EPFL

Thesis n° 11271

# Towards quasi real-time simulations of district heating networks for an optimal sustainable design and control

Presented on 27th March 2026

School of Engineering
L'IDIAP Laboratory
Doctoral program in Electrical Engineering

for the award of the degree of Docteur ès Sciences (PhD)

by

**Roberto BOGHETTI**

Accepted on the jury's recommendation

Prof. S. Carrara, jury president
Dr J.-M. Odobez, Dr J. H. Kämpf, thesis directors
Prof. G. Schweiger, examiner
Dr G. Mavromatidis, examiner
Prof. O. Fink, examiner

École
polytechnique
fédérale
de Lausanne

2026

"We should play the part of a careful householder; we should increase what we have inherited. Much still remains to do, and much will always remain, and he who shall be born a thousand ages hence will not be barred from his opportunity of adding something further."

— Seneca, Epistulae Morales 64.7, trans. R. M. Gummere

To my parents . . .

# Acknowledgements

The writing of this thesis has been a long and challenging journey. It would not have been possible without the support, encouragement, and companionship of many people who have shaped these years in profound ways.

First and foremost, I would like to thank my advisors, Jerome and Jean-Marc. Jerome, you instilled in me the curiosity and ambition to pursue a PhD in the first place. You consistently encouraged me to follow the rewarding path rather than the easy one, and I am deeply grateful for the trust you placed in me from day one. You gave me the freedom to explore my ideas independently while filling every meeting with challenging, but insightful discussions. Jean-Marc, your constant support and thoughtful feedback were invaluable. Even when my explanations were unclear, you took the time to truly help me. The results presented in this thesis would not have been achieved without your guidance and advice.

I am also sincerely grateful to my jury members, Prof. Sandro Carrara, Prof. Olga Fink, Dr. Georgios Mavromatidis, and Prof. Gerald Schweiger, for taking the time to read this work and for their insightful feedback. I greatly appreciated the discussions during and after my oral defense; they sparked new ideas and provided meaningful perspective on how to move forward.

A substantial part of this thesis was carried out at Idiap, where I was welcomed with remarkable warmth from the very first day. Over time, Idiap became much more than a workplace, it became a second family.

I would first like to thank the past and present members of the Energy Informatics Group. Giuseppe, you were the first to welcome me and quickly became my reference point whenever I needed help. The time we spent together in Valais, skiing, hiking, climbing, or experimenting with new vegan recipes as neighbors, remains among my most cherished memories. Ahad, Ahmad, Ahmed, Andrea, Chantal, David, Dubon, Fahad, Giacomo, Gianrocco, Gregorio, Karin, Kin Ho, Leonardo, Marco, Mathieu, Matteo Bilardo, Matteo Piro, Maximilien, Ollie, Olivier, Rui — with each of you I shared unforgettable moments: endless fondues, passionate discussions, and laughter that made even the busiest weeks lighter.

Among the first friends I met at Idiap were Eklavya, Florian, Neha, and Pablo. You tirelessly invited me to dinners, sports sessions, and trips, reminding me that work is only part of the

journey. I am deeply grateful for every ski day, climbing session, and hike that helped restore balance during intense periods.

A special mention goes to the original “pre-Aubertians”: Anshul, Arya, Karl, Sergio, and Tilak. The long nights, shared dinners, jam sessions, and countless inside jokes during our time as neighbors are part of the story I will always carry with me.

But the list of friends who made these years so special is long: Ajinkya, Alejandra, Alice, Alina, Alessandro, Amir, Andrei, Ante, Barbara, Bogdan, Cem, Chloé, Darya, Enno, Esau, Fabio, François, Giuseppe, Haruki, Hasindri, Hatef, Isabelle, Julian, Laura (and Carlos!), Laurent, Louise, Luis, Mahdi, Marco, Maxime, Michiel, Mingchi, Mirko, Olena, Parsa, Pierre, Roberto, Shashi, Simon, Suhan, Yulia, Zohreh, and many others — thank you for the countless shared moments that made Idiap not just an exceptional place to work, but to grow.

I would also like to acknowledge Idiap’s administrative and technical staff for their constant support and kindness, especially Giorgia, Laura, Maxime, Qendresa, Vincent, and William. And William, thank you for introducing me to squash.

Mathew and Michael, I am deeply grateful for the opportunity to be a teaching assistant in your courses. Through that experience, I not only learned a lot, but also discovered how much I enjoy teaching.

A heartfelt thank you goes to Matteo Sorci. During your time at Idiap, you were a reference point for many of us: always encouraging, always providing a solution to every problem. I am grateful for your friendship and for the many intense, thoughtful, and sometimes wonderfully absurd moments we shared.

I would also like to thank my EPFL friends, particularly Giulia and Jan, for the unforgettable adventures, from Copenhagen to Sion; Vasileios and Hnat for the hard work and loud laughter; and Siping for the endless phone calls that were as therapeutic as they were entertaining.

Anne, thank you not only for your professionalism and efficiency, but also for the many conversations we shared over the years. Aude, although our discussions were few, they were meaningful, and your advice supported me at important moments during my PhD.

I am also grateful to my colleagues at Satom, especially Bastien, Mentor, and Raymond, for the warm welcome and stimulating discussions, despite my imperfect French.

None of this would have been possible without the unwavering moral support of my lifelong friends. Francesca, Guido, Lorenzo, and Ugo: although life has scattered us across different places, I felt you closer than ever during these years. You stood by me through difficult moments and celebrated every small victory. Your calls, messages, and energy never failed to bring a smile to my face. Barso, Boce, Burga, Emi, Favilli, Mauro, Sbrana, and Scarpe: you are my safe harbor whenever I return to Tuscany, and the source of some of the most unforgettable stories of my life. Your craziness and extraordinary wit continue to inspire me. Arianna, Elisabetta: thank you for all the summer apéros, the gossip, and the fond trips down memory lane. Sonia: thanks for being the driving force that keeps the old Unipi team together, and for always being there when I need to vent, skate, or just smile.

Speaking of wit, a heartfelt thank you to my former dodgeball teammates, and in particular Ale, Dani, and Massi. What started as an inside joke became something much bigger: Hydra Viareggio has grown tremendously, and new generations now compete at the world level thanks to your dedication. Coming back to play with you still brings me the same joy as our very first match.

I would also like to thank Stefano, for the memes, the laughter, the photography sessions, the hikes, and the Zurich visits.

And Giona, you have been an invaluable collaborator and have become an irreplaceable friend. I am grateful for all the calls, beers, and celebrations, both in Italy and in Switzerland.

During all these years, everything around me changed, yet my life had one constant: Chiara. You have been by my side for longer than I can remember, supporting me through every high and every low. No amount of words can fully capture the life we have shared, the places we have seen, and the experiences through which we have grown up together. As always, you have been my pillar. If I did not give up, if I found the strength to push that extra centimeter when it mattered most, it is because of you. Thank you.

Last, but certainly not least, I would like to thank my family. Andrea, despite the distance, you have always been there for me. I rarely call or text, yet you tirelessly make up for my silence. Feeling your support, and through you the support of our entire family, has meant far more to me than I have ever been able to express. I am deeply grateful for that.

One of my greatest regrets is not being able to share this milestone with my parents. Our story may not have been the easiest, but you are the very reason I am who I am today. Thank you, from the bottom of my heart, for your resilience, for your guidance, and for giving me the opportunity to be here, now.

*Martigny, 27 February 2026* R. B.

This work was conducted within the framework of the Eguzki project, as part of a research programme of the Swiss Federal Office of Energy (SFOE) under project number SI/502111, and was funded by the SFOE and the project partners OIKEN SA, ALTIS Groupe SA, SATOM SA, and RWB Valais SA.
It also benefited from support through the EPFLglobaLeaders programme, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 945363.

# Abstract

District Heating Networks (DHNs) are a key component of thermal energy system decarbonization, and their planning, design, and operation place conflicting demands on simulation tools with respect to accuracy, flexibility, and computational efficiency. This thesis proposes a hybrid modeling strategy that combines physics-based simulation with machine learning surrogates, selectively applied across different phases of the DHN lifecycle to balance these requirements.

For the detailed design phase, the thesis argues that high-fidelity, component-based physical modeling remains necessary to ensure accuracy, interpretability, and robustness. This is achieved through the development of PyDHN, an open-source Python framework emphasizing modularity and component flexibility. The framework is systematically verified against benchmarks, experimental data, and operational measurements from the DHN of Verbier (Switzerland), showing strong agreement despite measurement uncertainty. Although computationally intensive, PyDHN provides reliable reference solutions for engineering analysis and for generating high-quality training data.

The thesis then focuses on the operational phase, where computational speed becomes a primary constraint. A hybrid differentiable framework is introduced in which the hydraulic solver is replaced by a topology-specific Graph Neural Network (GNN) surrogate, while thermal dynamics are retained in a fully differentiable physical model optimized for GPU execution. A novel cycle-based graph representation allows the surrogate to capture global pressure–flow dependencies in meshed networks. Compared to fully physics-based simulations, the proposed approach reduces computational time by a factor of 100 while maintaining high predictive accuracy. Its differentiable structure further supports gradient-based optimization of operational control strategies, which outperform heuristic and gradient-free methods.

Finally, the thesis considers the pre-design phase, characterized by the need to rapidly explore alternative network layouts. A topology-agnostic GNN surrogate is developed and trained on a large dataset of synthetic networks. By incorporating physics-informed inductive biases, the model generalizes to unseen topologies and exhibits zero-shot transfer to the Verbier network.

While slightly less accurate than topology-specific surrogates, it provides sufficient fidelity for early-stage planning at minimal computational cost.

Overall, this thesis shows that hybrid physics–machine learning modeling offers a practical and scalable approach to DHN simulation across the entire lifecycle. Rather than replacing physical models, data-driven surrogates complement them, supporting real-time optimization and large-scale design exploration for sustainable thermal energy systems.

**Keywords**: district heating networks, machine learning, graph neural networks, optimization, surrogate modeling, sustainability, physics-informed machine learning, sciml.

# Résumé

Les réseaux de chaleur urbains (RCU) constituent un élément clé de la décarbonation des systèmes d'énergie thermique, et leur planification, conception et exploitation imposent des exigences contradictoires aux outils de simulation en termes de précision, de flexibilité et d'efficacité computationnelle. Cette thèse propose une stratégie de modélisation hybride combinant simulation physique et modèles de substitution par apprentissage automatique, appliqués de manière sélective aux différentes phases du cycle de vie des RCU afin d'équilibrer ces exigences.

Pour la phase de conception détaillée, la thèse soutient qu'une modélisation physique haute fidélité, basée sur les composants, reste nécessaire pour garantir la précision, l'interprétabilité et la robustesse. Ceci est réalisé par le développement de PyDHN, un framework open-source en Python mettant l'accent sur la modularité et la flexibilité des composants. Le framework est systématiquement vérifié par rapport à des benchmarks, des données expérimentales et des mesures opérationnelles du RCU de Verbier (Suisse), montrant une forte concordance malgré l'incertitude de mesure. Bien que coûteux en temps de calcul, PyDHN fournit des solutions de référence fiables pour l'analyse d'ingénierie et pour la génération de données d'entraînement de haute qualité.

La thèse se concentre ensuite sur la phase opérationnelle, où la rapidité de calcul devient une contrainte primordiale. Un cadre hybride différentiable est introduit, dans lequel le solveur hydraulique est remplacé par un modèle de substitution de type réseau de neurones sur graphes (GNN) spécifique à la topologie, tandis que la dynamique thermique est conservée dans un modèle physique entièrement différentiable optimisé pour l'exécution sur GPU. Une représentation novatrice du graphe basée sur les cycles permet au modèle de substitution de capturer les dépendances globales pression–débit dans les réseaux maillés. Par rapport aux simulations entièrement physiques, l'approche proposée réduit le temps de calcul d'un facteur 100 tout en maintenant une précision prédictive élevée. Sa structure différentiable permet en outre l'optimisation par gradient des stratégies de contrôle opérationnel, surpassant les méthodes heuristiques et sans gradient.

Enfin, la thèse considère la phase de pré-conception, caractérisée par la nécessité d'explo-

rer rapidement des configurations alternatives de réseau. Un modèle de substitution GNN agnostique à la topologie est développé et entraîné sur un vaste jeu de données de réseaux synthétiques. En intégrant des biais inductifs informés par la physique, le modèle se généralise à des topologies inédites et présente un transfert zero-shot vers le réseau de Verbier. Bien que légèrement moins précis que les modèles de substitution spécifiques à la topologie, il offre une fidélité suffisante pour la planification en phase préliminaire à un coût computationnel minimal.

Dans l'ensemble, cette thèse montre que la modélisation hybride physique–apprentissage automatique offre une approche pratique et extensible pour la simulation des RCU sur l'ensemble du cycle de vie. Plutôt que de remplacer les modèles physiques, les modèles de substitution basés sur les données les complètent, permettant l'optimisation en temps réel et l'exploration de conception à grande échelle pour des systèmes d'énergie thermique durables.

**Mots clefs** : réseaux de chaleur urbains, apprentissage automatique, réseaux de neurones sur graphes, optimisation, modélisation de substitution, durabilité, apprentissage automatique informé par la physique, sciml.

# Zusammenfassung

Fernwärmenetze (FWN) sind eine Schlüsselkomponente der Dekarbonisierung thermischer Energiesysteme, und ihre Planung, Auslegung und ihr Betrieb stellen widersprüchliche Anforderungen an Simulationswerkzeuge hinsichtlich Genauigkeit, Flexibilität und Recheneffizienz. Diese Dissertation schlägt eine hybride Modellierungsstrategie vor, die physikbasierte Simulation mit maschinellen Lern-Surrogaten kombiniert, welche gezielt in verschiedenen Phasen des FWN-Lebenszyklus eingesetzt werden, um diese Anforderungen auszubalancieren.

Für die Phase der Detailplanung argumentiert die Dissertation, dass eine hochgenaue, komponentenbasierte physikalische Modellierung weiterhin notwendig ist, um Genauigkeit, Interpretierbarkeit und Robustheit zu gewährleisten. Dies wird durch die Entwicklung von PyDHN erreicht, einem quelloffenen Python-Framework mit Schwerpunkt auf Modularität und Komponentenflexibilität. Das Framework wird systematisch anhand von Benchmarks, experimentellen Daten und Betriebsmessungen des Fernwärmenetzes von Verbier (Schweiz) verifiziert und zeigt trotz Messunsicherheiten eine hohe Übereinstimmung. Obwohl rechenintensiv, liefert PyDHN zuverlässige Referenzlösungen für ingenieurwissenschaftliche Analysen und für die Erzeugung hochwertiger Trainingsdaten.

Anschließend konzentriert sich die Dissertation auf die Betriebsphase, in der die Rechengeschwindigkeit zur primären Einschränkung wird. Es wird ein hybrides differenzierbares Framework eingeführt, in dem der hydraulische Löser durch ein topologiespezifisches Graph-Neural-Network-(GNN)-Surrogat ersetzt wird, während die thermische Dynamik in einem vollständig differenzierbaren physikalischen Modell beibehalten wird, das für die GPU-Ausführung optimiert ist. Eine neuartige zyklenbasierte Graphdarstellung ermöglicht es dem Surrogat, globale Druck-Durchfluss-Abhängigkeiten in vermaschten Netzen zu erfassen. Im Vergleich zu rein physikbasierten Simulationen reduziert der vorgeschlagene Ansatz die Rechenzeit um den Faktor 100 bei gleichzeitig hoher Vorhersagegenauigkeit. Seine differenzierbare Struktur unterstützt darüber hinaus die gradientenbasierte Optimierung betrieblicher Regelungsstrategien, die heuristischen und gradientenfreien Methoden überlegen sind.

Abschließend betrachtet die Dissertation die Vorentwurfsphase, die durch die Notwendigkeit gekennzeichnet ist, alternative Netzwerklayouts schnell zu erkunden. Ein topologieagnosti-

sches GNN-Surrogat wird entwickelt und auf einem großen Datensatz synthetischer Netzwerke trainiert. Durch die Integration physikinformierter induktiver Biases generalisiert das Modell auf unbekannte Topologien und zeigt einen Zero-Shot-Transfer auf das Verbier-Netzwerk. Obwohl etwas weniger genau als topologiespezifische Surrogate, bietet es ausreichende Genauigkeit für die Planung in frühen Phasen bei minimalem Rechenaufwand.

Insgesamt zeigt diese Dissertation, dass eine hybride Modellierung durch eine Kombination aus Physik und maschinellem Lernen einen praktischen und skalierbaren Ansatz für die FWN-Simulation über den gesamten Lebenszyklus bietet. Anstatt physikalische Modelle zu ersetzen, ergänzen datengetriebene Surrogate diese und unterstützen Echtzeitoptimierung sowie großskalige Entwurfsexploration für nachhaltige thermische Energiesysteme.

**Stichwörter**: fernwärmenetze, maschinelles lernen, graph-neuronale-netze, optimierung, surrogatmodellierung, nachhaltigkeit, physikinformiertes maschinelles lernen, sciml.

# Contents

**Acknowledgements** **i**

**Abstract (English/Français/Deutsch)** **v**

**List of figures** **xv**

**List of tables** **xix**

**List of Abbreviations** **xxi**

**1 Introduction** **1**

**Introduction** **1**

1.1 Motivation and Context . . . 1
1.1.1 Heating as the Main Driver of Building-related Emissions . . . 2
1.1.2 Swiss Pathways to Decarbonizing Heating . . . 3
1.2 District Heating . . . 4
1.2.1 Common Characteristics . . . 4
1.2.2 The Generational Trajectory of District Heating . . . 6
1.2.3 Barriers and Opportunities for District Heating in Switzerland . . . 8
1.3 The Role of Optimization . . . 10
1.3.1 Optimization Across the DHN Lifecycle . . . 12
1.4 Simulation Tools for DHNs . . . 14
1.5 Machine Learning . . . 17
1.6 Rationale for the Present Study . . . 18
1.7 Research Hypotheses . . . 19
1.8 Thesis Outline . . . 20

**2 Physics-based Modeling** **21**

2.1 Introduction . . . 21
2.2 State of the Art . . . 22
2.2.1 Physical Modeling . . . 22
2.3 Contribution . . . 24
2.4 Methodology . . . 25
2.4.1 Physical modeling . . . 25

2.5 Implementation and Verification . . . 31
2.5.1 Pipe Model . . . 32
2.5.2 DESTEST . . . 33
2.6 Quantifying the Computational Bottleneck . . . 42
2.6.1 The Hydraulic Iteration Bottleneck . . . 42
2.6.2 The Thermal Sequentiality Bottleneck . . . 43
2.7 Chapter Conclusion . . . 44

**3 Graph Neural Network Surrogates for Hydraulic Simulation 45**
3.1 Introduction . . . 45
3.2 State of the Art . . . 46
3.3 Contribution . . . 47
3.4 Methodology . . . 48
3.4.1 Problem Setup . . . 48
3.4.2 Surrogate Modeling . . . 50
3.4.3 Condensed Graph Representation . . . 51
3.4.4 Input–output Structure of the Surrogate Model . . . 52
3.4.5 Graph Neural Networks . . . 52
3.4.6 Surrogate Model Architecture . . . 54
3.4.7 Loss Function . . . 55
3.5 Data, Model Configuration, and Evaluation Protocol . . . 56
3.5.1 Network Topology and Training Domain . . . 57
3.5.2 Experimental Data Generation . . . 58
3.5.3 Model Configurations . . . 59
3.5.4 Evaluation Metrics . . . 59
3.5.5 Implementation Details . . . 60
3.6 Results . . . 60
3.7 Ablation Study: Loss Function Variants . . . 62
3.8 Discussion . . . 63
3.9 Chapter Conclusion . . . 64

**4 Hybrid Differentiable Modeling: Real-World Validation and Application 67**
4.1 Introduction . . . 67
4.2 State of the Art . . . 68
4.2.1 Overview of Operational Control Strategies . . . 68
4.2.2 Gradient-Based DHN Optimization . . . 70
4.3 Contributions . . . 70
4.4 Dynamic Thermal Model . . . 71
4.4.1 Analytical Solution of the Pipe Source Problem . . . 72
4.4.2 Solution of the Pipe Advection Problem . . . 74
4.4.3 Model Boundary Conditions and Overall System Solution . . . 78
4.4.4 Implementation Details and Differentiability. . . . 78
4.5 Verification of the Proposed Workflows . . . 80

4.5.1 Case Study and Data Collection . . . 81
4.5.2 Simulation Specifications . . . 82
4.5.3 Verification Procedure . . . 83
4.5.4 Results . . . 84
4.5.5 Limitations of the Verification Study . . . 88
4.6 Application: Gradient-Based Control Optimization . . . 89
4.6.1 Control Strategies . . . 89
4.6.2 Optimization Approach . . . 89
4.6.3 Results and Discussion . . . 90
4.7 Chapter Conclusion . . . 91

**5 Towards Generalizable Surrogate Models: Topology-Agnostic Learning 93**
5.1 Introduction . . . 93
5.2 Contributions . . . 94
5.3 Methodology . . . 94
5.3.1 Problem Setup . . . 94
5.3.2 Problem Space Definition . . . 95
5.3.3 Synthetic Data Generation Framework . . . 97
5.3.4 Input-Output Structure of the Surrogate Models . . . 100
5.3.5 Solver-Inspired Architecture . . . 102
5.3.6 Data Augmentation . . . 107
5.4 Data, Model Configuration, and Evaluation Protocol . . . 108
5.4.1 Dataset Composition . . . 108
5.4.2 Model Configurations . . . 109
5.4.3 Evaluation Protocol . . . 110
5.4.4 Implementation Details . . . 110
5.5 Results and Discussion . . . 110
5.5.1 In-Distribution Generalization . . . 111
5.5.2 Real-World Generalization . . . 114
5.5.3 Computational Efficiency . . . 116
5.6 Exploratory Studies . . . 117
5.6.1 Unsupervised Training . . . 117
5.7 Chapter Conclusion . . . 118

**6 Conclusions and Outlook 119**
6.1 Limitations, Transferability, and Future Directions . . . 121
6.2 Concluding Remarks . . . 124

**Bibliography 137**

**Curriculum Vitae 139**

# List of Figures

1.1 Schema of a meshed 3rd generation DHN. The flow directions outside the internal loop are known. . . . . 5

2.1 Example of a DHN and its graph representation. **Left:** GIS-style DHN layout, where supply and return lines are represented as a single connection. The heating plant is labeled HS, substations as S, and junctions with lowercase letters. **Right:** Graph representation used in this work, where supply and return lines are explicitly separated. . . . . 26

2.2 Cycle-based decomposition of mass flows. **Left:** Cycles of the graph in Figure 2.1. Boundary cycle flows are fully determined by the mass flow given at their unique substation. **Right:** Superposition principle for cycle $c_4$. Edge mass flows are obtained by summing signed contributions from the cycle flows $\tilde{m}_i$. The contribution of $\tilde{m}_4$ is shown in bold, signs reflect the cycle-edge incidence matrix. 29

2.3 Verification of the aerial pipe model against measured data. The shaded area shows the measurement range, taking into account sensor tolerance. The simulated output of the proposed model closely tracks the real-world measurements. 33

2.4 Layout of the radial network used in the DESTEST Common Exercises. The node $i$ is the source, and the numbered circles are the consumers. . . . . 34

2.5 Mass flow tracking performance at Substation 1. **Top:** The solid line represents the common mass flow demand profile used for all buildings in DESTEST CE1, serving as the boundary condition for the simulation. The dashed line shows the actual mass flow in the reference Modelica simulation. **Bottom:** The control error (actual minus setpoint) illustrates that the Modelica controller tends to undershoot response, visible as distinct negative spikes, whenever the demand changes rapidly. . . . . 35

2.6 Results of the DESTEST CE0. Total load and heat loss. . . . . 37

2.7 Results of the DESTEST CE0. Mass flow and pressure. . . . . 37

2.8 Results of the DESTEST CE0. Supply temperatures. . . . . 38

2.9 Results of the DESTEST CE0. Return temperatures. . . . . 39

2.10 Temperature dynamics at Substation 1 (DESTEST CE1). During zero-demand periods, both models show cooling on the supply side (**top**), with PyDHN (solid blue) exhibiting a steeper decay than the reference (dashed red). On the return side (**bottom**), the Modelica model maintains a quasi-steady temperature, potentially computed inside the substation rather than at the bypass level, whereas PyDHN simulates the natural cooling of stagnant fluid. The periodic spikes in the PyDHN return profile reflect the activation of the bypass, transferring hot supply water directly to the return line. . . . . 41

2.11 Heat loss dynamics for the return pipe segment between nodes $i$ and $h$. During almost no-flow intervals, the proposed tool (solid blue) simulates a rapid decay to near-zero heat loss, followed by a high-intensity spike representing the thermal load of reheating the cooled pipe wall. The reference tool (dashed red) instead exhibits an initial sudden drop and rebound, followed by a gradual decay that sustains a higher rate of heat loss. . . . . 42

3.1 Overview of the simulation pipeline and relevant sections. The network topology is used to extract a cycle basis and define the structure and edge features of the condensed graph, while static pipe properties provide its node attributes. These, together with the boundary flows computed at each time step from hydraulic boundary conditions, form the input to the Surrogate Model (SM in the figure). The model predicts internal cycle flows, used to reconstruct full pipe mass flows $\dot{\boldsymbol{m}}^{\text{pred}}$, which are then passed, along with thermal boundary conditions, to a physics-based thermal model. . . . . 49

3.2 Construction of the condensed graph. **Left**: Original network graph showing a producer (HS), substations (S), and junctions (lowercase letters). Center: Boundary mass flows are computed based on boundary conditions, with color intensity qualitatively representing flow magnitude (white indicating zero flow). **Right**: In the condensed graph, each cycle is a virtual node connected to its constituent pipe nodes, which retain boundary flow magnitude as a feature. Edges are assigned an orientation attribute. . . . . 50

3.3 **Left:** Original DyResGen block. **Right:** Our implementation with ELU. The plus sign $\bigoplus$ indicates element-wise summation. . . . . 54

3.4 Overview of the proposed GNN architecture. Inputs are processed to create scaled node features ($\boldsymbol{X}_{\text{scaled}}$), an adjacency matrix ($\boldsymbol{A}'$), and edge features ($\boldsymbol{e}$). In Stage 1, an encoder-decoder model produces an initial cycle flow prediction ($\tilde{\boldsymbol{m}}_1^{(1)}$). This prediction is then processed by a Flow Aggregation Layer and combined with the original inputs to form a new feature matrix for Stage 2, which outputs a final, refined prediction ($\tilde{\boldsymbol{m}}_1^{\text{pred}}$). . . . . 55

3.5 The DHN of Verbier. Heat plants are shown in green, while the substations with pressure measurements are in blue and aerial pipes are colored in red. The central section of the network containing the internal loops is highlighted in blue. 57

3.6 Selected internal cycles for the supply (**left**) and return (**right**) lines with their respective IDs and directions. The section of the network with these cycles is highlighted in Figure 3.5. . . . . 59
3.7 Overview of the model accuracy on the test set for the supply and return lines under different maximum residual thresholds. . . . . 62
3.8 Distribution of the residual error on the test set for different ranges of maximum circulating mass flow. . . . . 63

4.1 Schematic representation of the pipe model. **Left**: Illustration of the Lagrangian modeling approach, which tracks discrete water volumes of varying sizes along the pipe. The cross-section distinguishes the central water core (dark blue) from the surrounding viscous sublayer (light blue), the thickness of which is increased for visual clarity. **Right**: Schematic representation of the two-node lumped parameter thermal model. The diagram illustrates the radial heat transfer across concentric layers of the pipe system. The thermal circuit consists of two capacitive nodes: the water core temperature ($\theta_{wc}$) and the stationary boundary temperature ($\theta_B$), connected to the external environment temperature ($\theta_{ext}$). . 72
4.2 Illustration of the pipe model: advection and outlet temperature computation. Here, the incoming volume is less than the total pipe volume ($\alpha = 0$), and the outlet temperature is computed as the volume-weighted average temperature of the portion of water highlighted by the red dashed border. For visual clarity, the distinction between the water core and viscous sublayer is not depicted. . . . . 75
4.3 The DHN of Verbier. The producers are highlighted in green, while the substations with pressure measurements are in blue. Aerial pipes are highlighted in red. . . . . 81
4.4 Daily variations in monitoring data, showing mean daily values and ranges. **Left**: total power demand (kW) and air temperature (°C). **Right**: mass flow ($kg\,s^{-1}$) and supply temperature (°C) at the MAIN plant. Shaded regions indicate the variability. . . . . 83
4.5 Modeling error on pressure. **Left**: distribution of the absolute error on pressure difference **Right**: relative error after the correction. The dashed red lines indicate the thresholds of ±1.2%. . . . . 85
4.6 Measured and simulated return temperature of two producers with bars showing the absolute error in K. The bars are colored in blue if the error with sign is below −2 °C, red if above 2 °C, and green if in between these values. The blue shaded area around the measurements shows the threshold of ±2 °C, while the red shaded areas show the time-steps where monitoring data was missing. . . . . 86

5.1 Examples of graphs generated with the proposed procedure. All graphs feature 6 cycles and 80 pipes. The initial cycle, around which the rest of the graph is built, is highlighted in red. Randomly selected source and sink nodes, representing the connections to branches, are colored in red and blue respectively. . . . . 97

5.2 Illustration of the topological synthesis procedure. The target graph consists of 3 cycles with an edge budget allocation of [6,5,2]. **1.** A base cycle is constructed using the first 6 edges. **2.** A second cycle is formed by inserting a chain of 5 edges into the existing structure. **3.** Similarly, the final cycle is created by connecting two existing nodes with a chain of 2 edges. . . . . 98
5.3 Schematic of the pipe sizing step. **Left:** Node assignment. Nodes are designated as sources (red) or sinks (blue), representing interfaces with external branches (dotted lines) where production or consumption occurs. **Right:** Flow estimation heuristic and pipe sizing. Sink demands are propagated back to sources along shortest paths. The resulting aggregated flow intensity (visualized by edge shade) determines the minimum viable diameter for each pipe segment, from which the final diameter is stochastically sampled (visualized by edge thickness). . . . 99
5.4 High-level overview of the recurrent Solver-Inspired architecture. . . . . 102
5.5 Cumulative distribution of the accuracy on the test set for different residual thresholds. The x-axis reports the residual threshold (up to 500 Pa), while the y-axis shows the percentage of samples whose maximum cycle residual does not exceed the threshold. More than 70% of samples exhibit a maximum residual below 50 Pa, highlighting the strong concentration of errors at low magnitudes and the presence of a long but sparse tail. . . . . 113
5.6 Analysis of outliers (> 1000 Pa): heatmap showing the count of outliers as a function of topological coplexity (number of cycles vs. number of pipes). . . . . 114
5.7 Analysis of the residual outliers ($> 1000 Pa$) in the test set. **Left:** Heatmap showing the average maximum cycle residual (Pa) plotted against the maximum cycle stiffness and boundary flow. High residuals are concentrated in the region of high stiffness and high flow. **Right:** Percentage of outliers in each bin. A problematic zone is clearly delimited at the intersection of high hydraulic resistance and high operational load. . . . . 115

# List of Tables


1.1 Comparative Analysis of District Heating Generations. . . . . 8

1.2 Summary of optimization studies on DHNs and stated benefits . . . . . 11

2.1 RMSE between the measured outlet temperature and the simulation output of the proposed model and several reference tools. Best results for each case are highlighted in bold. . . . . 33

2.2 Results of the proposed framework on DESTEST CE0. The value reported is the NMBE with respect to the reference value. . . . . 36

2.3 Results of the proposed framework on DESTEST CE1. The value reported is the NMBE with respect to the reference work. . . . . 40

3.1 Validation set performance over 10 runs for each tested architecture. Values reported as mean ± std (best) over the different seeds. Best values for each KPI are highlighted in bold. . . . . 60

3.2 Performance of the best run (model *ELU-2-7l*) on the test set: results are shown for the supply and return lines separately. Values reported as mean ± std within the set. . . . . 61

3.3 Ablation study on the loss function. Comparison between regression-only loss (*Regression*), fixed-weight loss (reported as $w_1$, $w_2$), and dynamic weighting (*Ours*). Validation set performance over 10 runs reported as mean ± std (best). Best values in bold. . . . . 64

4.1 Performance comparison between the hybrid method and PyDHN against monitoring data: inlet temperatures. . . . . 87

4.2 Performance of each curve optimized with our method and DE. Total heat loss differences relative to manual control (in kWh and %) are reported, along with the mean ($\mu$) and standard deviation ($\sigma$) of supply temperatures, shown as (MAIN, SEC1). Bold values indicate the lowest heat loss for each variant; no bold means manual control performed best. Curve types: $C$ (constant), $L$ (linear), $p$ (preheating), $s$ (setback). . . . . 91

5.1 Definition of the problem space boundaries, organized by category. . . . . 96

5.2 Composition of the synthetic training dataset. Complex topologies are oversampled to ensure the model learns robust update rules for high-interference scenarios. . . . . 109
5.3 Validation set performance statistics (10 seeds). We report **Mean** ± **Std** and the **(Best)** observed value across seeds. **Acc**: Percentage of samples with residuals < 100 Pa. $\mathbf{MAE_m}$: Mean Absolute Error on pipe mass flow. . . . . 111
5.4 Validation set hydraulic consistency. $\mathbf{Res_c}$: Average cycle pressure residual. $\mathbf{Res_m}$: Maximum pressure residual observed across the entire dataset. Note that for $\mathrm{Res_m}$ values are given in kPa. Mean and standard deviation are taken across random seeds. . . . . 112
5.5 Test set metrics of the selected best-performance model. **Acc**: Percentage of samples with cycle residuals < 100 Pa. **MAE**: Mean Absolute Error on pipe mass flow. $\mathbf{Res}_c$: Average cycle pressure residual. $\mathbf{Res}_s$: Average of the per-sample maximum cycle residual. $\mathbf{Res}_m$: Maximum pressure residual observed across the dataset. . . . . 112
5.6 Test set cycle pressure residuals as a function of the number of cycles in the network. Statistics are computed over individual samples for the selected Solver-Inspired GNN model. . . . . 114
5.7 Zero-shot performance on Verbier DHN monitoring data. The model was applied directly without any fine-tuning. **Acc**: Percentage of samples with cycle residuals < 100 Pa. **MAE**: Mean Absolute Error on pipe mass flow. $\mathbf{Res}_c$: Average cycle pressure residual. $\mathbf{Res}_s$: Average of the per-sample maximum cycle residual. $\mathbf{Res}_m$: Maximum pressure residual observed across the dataset. . . . . 115
5.8 Computational efficiency statistics (10 seeds). **Training Time** is the time taken to complete 100 training epochs. **Convergence Epoch** denotes the training percentage where the maximum validation accuracy was achieved. We report **Mean** ± **Std** and the **(Best)** observed value, where lower is better for both metrics. 116
5.9 Comparison of model performance between the supervised baseline (SL) and the physics-only (unsupervised, UL) training strategy on the test set. . . . . 117

# List of abbreviations

| | |
|---|---|
| **1GDH** | First Generation District Heating |
| **2GDH** | Second Generation District Heating |
| **3GDH** | Third Generation District Heating |
| **4GDH** | Fourth Generation District Heating |
| **5GDHC** | Fifth Generation District Heating and Cooling |
| **6GDHC** | Sixth Generation District Heating and Cooling |
| **Acc** | Accuracy |
| **BC** | Boundary Conditions |
| **CAGR** | Compound Average Growth Rate |
| **CAPEX** | Capital Expenditures |
| **CE** | Common Exercise |
| **CHP** | Combined Heat and Power |
| **CPU** | Central Processing Unit |
| **CVRMSE** | Coefficient of Variation of the Root Mean Square Error |
| **DACVRMSE** | Daily Amplitude Coefficient of Variation of the RMSE |
| **DAE** | Differential-Algebraic Equation |
| **DE** | Differential Evolution |
| **DESTEST** | District Energy Simulation Test |
| **DETEC** | Federal Department of the Environment, Transport, Energy and Communications |
| **DH** | District Heating |
| **DHN** | District Heating Network |
| **DHW** | Domestic Hot Water |
| **DN** | Nominal Diameter (*Diamètre Nominal*) |
| **ELU** | Exponential Linear Unit |
| **EnG** | Energy Act |
| **FEM** | Finite Element Methods |
| **FMI** | Functional Mock-up Interface |
| **FOEN** | Swiss Federal Office for the Environment |
| **GENConv** | Generalized Graph Convolution |
| **GHG** | Greenhouse Gas |
| **GIS** | Geographic Information System |
| **GNN** | Graph Neural Network |

| | |
|---|---|
| **GPU** | Graphics Processing Unit |
| **HCVRMSE** | Hourly-averaged Coefficient of Variation of the RMSE |
| **HIU** | Heat Interface Unit |
| **IBPSA** | International Building Performance Simulation Association |
| **IEA** | International Energy Agency |
| **IPCC** | Intergovernmental Panel on Climate Change |
| **KIG** | Climate and Innovation Act |
| **KPI** | Key Performance Indicator |
| **LHS** | Latin Hypercube Sampling |
| **LN** | Layer Normalization |
| **LTI** | Linear Time-Invariant |
| **MAE** | Mean Absolute Error |
| **MAPE** | Mean Absolute Percentage Error |
| **MAS** | Multi-Agent Systems |
| **MINLP** | Mixed Integer Non-Linear Problems |
| **ML** | Machine Learning |
| **MLP** | Multilayer Perceptron |
| **MPC** | Model Predictive Control |
| **MPS** | Message Passing Stack |
| **NMBE** | Normalized Mean Bias Error |
| **NN** | Neural Network |
| **ODE** | Ordinary Differential Equation |
| **OPEX** | Operational Expenditures |
| **PID** | Proportional-Integral-Derivative |
| **PINN** | Physics-Informed Neural Network |
| **PReLU** | Parametric Rectified Linear Unit |
| **ReLU** | Rectified Linear Unit |
| **RMSE** | Root Mean Square Error |
| **RNN** | Recurrent Neural Network |
| **SFOE** | Swiss Federal Office of Energy |
| **SL** | Supervised Learning |
| **TRNSYS** | TRaNsient SYStem |
| **UL** | Unsupervised Learning |

# 1 Introduction

## 1.1 Motivation and Context

Anthropogenic climate change poses an urgent global challenge, as it intensifies extreme weather events, accelerates sea-level rise, and threatens large-scale ecosystem stability (Masson-Delmotte et al., 2021). Its primary driver is the accumulation of Greenhouse Gases (GHGs) in the atmosphere, which amplify the natural greenhouse effect and raise the Earth's mean surface temperature (Masson-Delmotte et al., 2021). The magnitude of this warming is strongly correlated with cumulative $CO_2$ emissions and other long-lived GHGs, implying that continued emissions will result in further temperature increases (Masson-Delmotte et al., 2021). To counteract this trajectory, the international community adopted the Paris Agreement in 2015, with nearly all nations committing to limit the rise in global mean temperature to well below 2 °C and to pursue efforts to restrict it to 1.5 °C above pre-industrial levels (UNFCCC, 2016). This threshold has been identified as a critical boundary beyond which the likelihood of severe and potentially irreversible impacts increases sharply. Achieving stabilization at these levels requires a rapid reduction in GHG emissions and a systemic transformation of the global energy infrastructure (Masson-Delmotte et al., 2021; Rogelj et al., 2015).

In line with these international commitments, Switzerland has set its own ambitious targets, most notably through its Energy Strategy 2050 and the Climate and Innovation Act ("Bundesgesetz über die Ziele im Klimaschutz, die Innovation und die Stärkung der Energiesicherheit", 2022), with the national goal of achieving net-zero GHG emissions by 2050. Within this national strategy, the building sector is a critical lever.

According to a report commissioned by the Swiss Federal Office of Energy (SFOE) (Kemmler & Vu, 2024), energy demand in buildings reached 290.9 PJ in 2023, representing around 41% of the internal final energy consumption (or 38% of the total). Although this value marked a slight increase compared to the 286.1 PJ of the previous year, largely driven by colder weather (+1.8% heating degree days) and reduced solar radiation (-7.8%), the long-term trend is positive. Over the 2010-2023 period, building energy consumption decreased substantially by 107.8 PJ, averaging an annual reduction of approximately 1.7% (Compound Average Growth Rate, CAGR,

of -2.4%). This downward trajectory in Switzerland contrasts significantly with the steady upward trend reported globally, where energy demand increased by an average of 0.9% per year during the same period (International Energy Agency, 2024).

Despite this remarkable trend, building energy demand remains high, and the associated GHG emissions are correspondingly significant. According to the Swiss Federal Office for the Environment (FOEN), the building sector (comprising households and services) was responsible for 9.05 $MtCO_2$-eq in 2023. This represents 22.2% of Switzerland's total GHG emissions (Federal Department of the Environment, Transport, Energy and Communications & Federal Office for the Environment, 2025). These emissions primarily stem from the combustion of fossil fuels (mainly heating oil and natural gas) for space heating and hot water preparation. While the sector's emissions have decreased substantially since 1990 (when they amounted to 16.72 $MtCO_2$-eq), buildings remain a key focus area for achieving national climate goals.

### 1.1.1 Heating as the Main Driver of Building-related Emissions

Within the Swiss building sector's energy consumption profile, the provision of space heating and Domestic Hot Water (DHW) are the primary drivers. In 2023, these two end-uses combined accounted for 86.7% (252.2 PJ) of the total final energy consumed in Swiss buildings (households, services, and industry building components). Addressing the energy demand for these thermal applications is therefore a core challenge to decarbonizing the Swiss built environment.

**Space heating** is the largest single energy end-use in Swiss buildings, responsible for 71.2% (207.0 PJ) of the sector's consumption in 2023 (Kemmler & Vu, 2024). Historically and currently, a significant portion of this demand is met through the direct combustion of fossil fuels, making space heating a principal source of the sector's direct GHG emissions. While the trend is positive, with total space heating consumption declining by 21.2% between 2000 and 2023 despite a substantial increase in heated floor area (+31.7%), the reliance on fossil fuels remains a key challenge. In 2023, heating oil and gas still supplied 55.6% of the end energy for space heating in households.

**DHW** represents the second major thermal end-use in Swiss buildings, accounting for 15.5% (45.2 PJ) of building energy consumption in 2023. Unlike space heating, where demand has clearly decreased due to better insulation and efficiency measures, the energy required for DHW has shown less reduction, decreasing only by 9.0% between 2000 and 2023. This relative stability means that as buildings become more thermally efficient, DHW accounts for a progressively larger share of the remaining heat demand. A key technical challenge lies in the required supply temperatures: while modern, low-temperature space heating systems can operate efficiently at temperatures around 35 °C to 40 °C, DHW systems typically require 60 °C to ensure hygienic safety by preventing the growth of pathogens like Legionella (Graf et al., 2024). This can limit the efficiency or direct applicability of certain renewable heat sources like heat pumps operating optimally at low temperatures.

**Space cooling**, while still representing a much smaller share of building energy consumption compared to heating (approximately 2.1% or 6.0 PJ in 2023), is a rapidly growing end-use in Switzerland. Its consumption increased by over 50% between 2000 and 2023. This trend is likely driven by factors including warmer summer temperatures (Cooling Degree Days in 2023 were significantly above the 2000-2023 average), increasing standards for comfort in service sector buildings, and potentially rising adoption in residential buildings. As cooling demand is met almost exclusively by electricity, its growth contributes to increasing electricity consumption in buildings.

### 1.1.2 Swiss Pathways to Decarbonizing Heating

Switzerland's approach to tackling emissions from building heating aligns with global strategies but is specifically shaped by its national context and a multi-layered policy framework. The Confederation sets overarching national targets (e.g., Net-Zero by 2050) via key legislation like the Energy Act (EnG) (Die Bundesversammlung der Schweizerischen Eidgenossenschaft, 2016), $CO_2$ Act (The Federal Assembly of the Swiss Confederation, 2011), and Climate and Innovation Act (KIG) ("Bundesgesetz über die Ziele im Klimaschutz, die Innovation und die Stärkung der Energiesicherheit", 2022), and provides significant funding through mechanisms such as the $CO_2$ levy-funded Buildings Programme and a CHF 2 billion KIG impulse fund for heating system replacement (Swiss Federal Office of Energy & Federal Office for the EnvironmentFederal Office for the Environment (FOEN), 2025; Swiss Federal Office of Energy (SFOE), 2023). Primary regulatory authority for building energy use resides with the cantons, which implement federal goals through cantonal energy laws, largely harmonized via the MuKEn framework ("Mustervorschriften der Kantone im Energiebereich (MuKEn)", 2025). The overall strategy pursues a dual approach: reducing thermal demand and transitioning energy supply. This is achieved primarily through three pillars:

1. **Reduction of Thermal Demand:** The most fundamental step is to minimize the amount of energy required for heating through improvements in the building envelope. Key measures include enhanced insulation, high-performance windows, and airtight construction. Despite progress towards targets like the Energy Strategy 2050's aim for reduced per capita consumption, accelerating the deep renovation rate of existing buildings remains a significant challenge (Swiss Federal Office of Energy (SFOE), 2023).

2. **Electrification of Heating:** Transitioning heat supply relies heavily on Switzerland's largely $CO_2$-free electricity grid, with electric heat pumps identified as a key technology (*Energy Strategy 2050 Monitoring Report 2019 Abridged Version*, 2019; "Mustervorschriften der Kantone im Energiebereich (MuKEn)", 2025; Swiss Federal Office of Energy (SFOE), 2023). Electrification allows the direct use of renewable energy, enhances system integration, and supports flexibility through smart grids (Heinen et al., 2018; Schill & Zerrahn, 2020; X. Zhang et al., 2019). However, the resulting increase in winter electricity demand, when domestic renewable generation is lower and the reliance on

imports is higher, poses a challenge for the security of supply (Swiss Federal Office of Energy (SFOE), 2023).

3. **Collective Solutions (District Heating, DH):** In densely populated areas, District Heating Networks (DHNs) are prioritized for systemic decarbonization, leveraging diverse low-carbon sources like waste heat, biomass, ambient heat (e.g., from lakes) via large heat pumps, and potentially geothermal energy (*Potenzial von fernwärme- und fernkälteanlagen*, 2021; Swiss Federal Office of Energy (SFOE), 2023). Although currently serving a smaller market share (approximately 6% of heating/hot water energy in 2019), the goal is to more than double the heat delivered via DH by 2050 (from 8.4 TWh to 17-22 TWh) (*Potenzial von fernwärme- und fernkälteanlagen*, 2021).

These pillars are highly synergistic. For instance, deep retrofits reduce the required capacity of both individual heat pumps and connections to district heating networks, lowering capital costs and at the same time providing a challenge for long-term planning (Hansen et al., 2019; Popovski et al., 2023). Furthermore, DHNs integrated with large-scale heat pumps and thermal energy storage can provide valuable flexibility to the broader energy system. This coupling between the heating and electricity sectors can support higher shares of variable renewable energy: DH systems can absorb surplus electricity from wind and solar to generate and store heat, thereby helping to balance the grid and reducing the need for costly electricity storage or curtailment (*Climate Change 2022*, 2022; Fernqvist et al., 2023).

## 1.2 District Heating

Building on the discussion of the systemic role of DHNs, the following section provides an overview of these systems, introducing the associated terminology, their defining characteristics, historical evolution, and the technological shifts that distinguish different generations of these systems. The section will conclude with a discussion on the main barriers to their adoption.

### 1.2.1 Common Characteristics

While the specific technologies and configurations have evolved dramatically since their inception, DHNs can be conceptually understood through three principal subsystems: thermal energy production sources (*producers)*, a *distribution network*, and the end-user (*consumers*) interface (De la Cruz & Ugalde-Loo, 2021). The nature of these components and the synergy between them have been redefined with each generation, dictating the system's overall efficiency, reliability, and economic viability. Figure 1.1 shows a high-level overview of the main concepts.

**Producers** provide the thermal energy for the network. Historically, this function was dominated by large, *centralized* plants housing boilers or Combined Heat and Power (CHP) units.

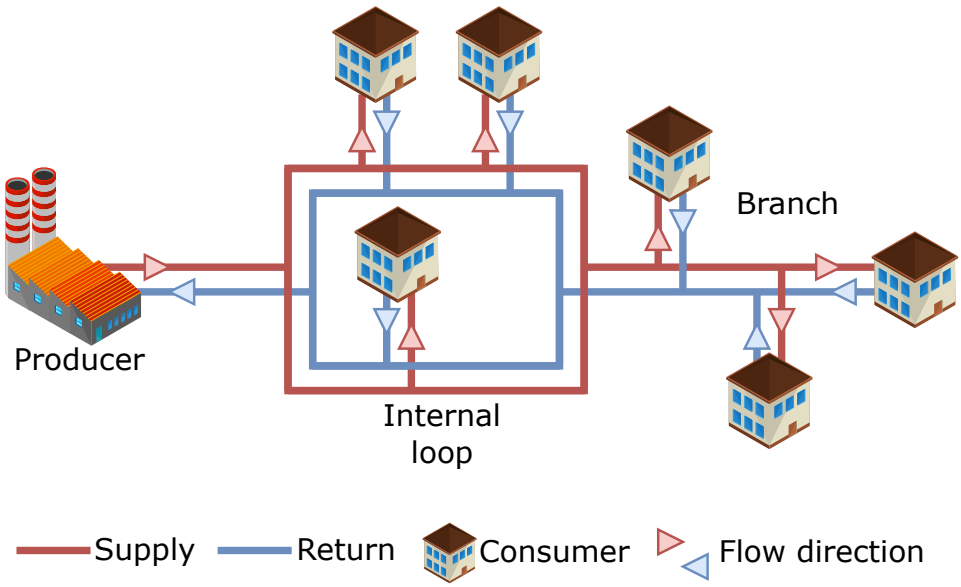



Figure 1.1: Schema of a meshed 3rd generation DHN. The flow directions outside the internal loop are known.

While this model persists, modern networks increasingly incorporate a portfolio of *distributed* sources, such as geothermal wells, solar thermal arrays, and interfaces to capture waste heat, converging towards a completely *decentralized* architecture.

**The Distribution Network** functions as the circulatory system, transporting thermal energy from the production sources to the end-users. It consists of a vast, interconnected network of underground pipes. In nearly all modern systems, this is a closed-loop configuration comprising two parallel pipelines: a *supply line* that carries the heated working fluid to the consumers, and a *return line* that brings the fluid back at a lower temperature to be reheated. The physical layout, or *topology*, of this network can vary. A *branched* or *tree* topology is common for its simplicity and lower initial cost, but it is vulnerable to single-point failures. More robust *ring* or *meshed* topologies, characterized by their *internal loops*, provide multiple paths for fluid flow, increasing supply resilience and hydraulic stability at the cost of higher capital investment. A primary challenge in network operation is mitigating *thermal losses* to the surrounding ground, which are a direct function of the temperature difference between the working fluid and the soil, the pipe diameter, and the quality of the pipe insulation.

**The Consumer Interface** forms the boundary between the district network and the individual building's internal system. This component has evolved significantly from direct steam condensation in radiators with simple steam traps to manage condensate to decentralized heat pumps. Across modern generations, this interface typically contains control valves, pumps, and meters for regulation and billing.

### 1.2.2 The Generational Trajectory of District Heating

The concept of *generations* provides a structured framework for understanding the historical development of DH, with each successive generation driven by the need to overcome the limitations of its predecessor, primarily by reducing operating temperatures to improve efficiency and integrate new heat sources (Lund et al., 2014). In particular, five generations have been developed, with recent proposals around a sixth generation. Importantly, these generations coexist, as older generations are still being built and operated. In the case of later generations, their development is progressing in parallel. A summary of the existing generations and their main characteristics is given in Table 1.1.

**First Generation (1GDH)** systems, emerging in the 1880s, were the first form of district energy. They provided heat using steam at very high temperatures, often exceeding 200 °C, as the carrier. This steam was distributed through pipes housed in large, site-built concrete ducts (Lund et al., 2014). While revolutionary for their time, these systems were limited by extremely high thermal losses, significant safety risks associated with high-pressure steam, and rapid material degradation.

**Second Generation (2GDH)** networks, developed from the 1930s onwards, replaced steam with pressurized hot water. Operating at supply temperatures typically above 100 °C, these systems were safer, more reliable, and more efficient than their steam-based counterparts (Lund et al., 2014). Many 2GDH systems were constructed in the post-war era, often fueled by coal or oil, and their development was mainly driven by the prominent need of reducing fuel consumption and increasing thermal comfort through CHP plants. However, their high operating temperatures still result in substantial heat losses and severely limit their ability to integrate low-grade renewable or waste heat sources.

**Third Generation (3GDH)** represents the current, mature choice for large-scale DH deployment. Emerging in the 1970s, largely in response to the oil crises, 3GDH systems were designed with energy efficiency as a primary goal. Their defining characteristics include supply temperatures below 100 °C (typically in the 80 °C to 90 °C range), the widespread adoption of highly effective factory-built, pre-insulated bonded pipes, and the use of compact, standardized *substations* at the consumer interface (Lund et al., 2014). This combination of technologies drastically reduced heat losses, improved operational control, and lowered installation costs. As a result, 3GDH became the dominant paradigm, and it is today the **currently most diffused generation** of DH, forming the backbone of the vast majority of existing networks across Europe and in Switzerland (Caputo et al., 2021; Chambers et al., 2019; De la Cruz & Ugalde-Loo, 2021). Notably, 3GDH can use waste heat, solar and geothermal sources (Socci et al., 2024; Sorknæs et al., 2022).

Driven by the urgent need for decarbonization and the integration of fluctuating renewable energy sources, two new generations of DHC are being developed and deployed in parallel since recent times, representing a further shift towards lower operating temperatures (Lund et al., 2021).

**Fourth Generation District Heating (4GDH)** is an evolutionary advancement of the 3GDH concept. Its primary objective is to allow a fully sustainable energy system by operating at lower supply temperatures, typically between 50 °C to 70 °C (Lund et al., 2014). This temperature reduction unlocks the potential to integrate a wide array of low-grade heat sources that would be less efficient, or even incompatible, with 3GDH systems. These sources include solar thermal, shallow geothermal, and waste heat recovered from data centres, supermarkets, and industrial processes (Lund et al., 2021; Sorknæs et al., 2022). Furthermore, 4GDH is designed to be an integral component of future smart energy systems. The thermal mass of the network or heat storage tanks can be used for energy storage, providing flexibility to the electrical grid by absorbing surplus renewable electricity (Averfalk et al., 2017; Zheng et al., 2018). The transition from 3GDH to 4GDH, while offering substantial system-level benefits, requires that connected buildings have high-performance thermal envelopes and are equipped with low-temperature heat emitters (e.g., floor heating).

**Fifth Generation District Heating and Cooling (5GDHC)**, in contrast, represents a more substantial paradigm shift. It is not a direct successor to 4GDH but rather a parallel development with a fundamentally different architecture (Boesten et al., 2019; Lund et al., 2021). 5GDHC networks operate as low-temperature thermal grids with water at near-ambient ground temperatures (e.g., 10 °C to 25 °C). The core innovation is the decentralization of temperature boosting. Each connected building is equipped with its own heat pump, which acts as a substation. For heating, the heat pump extracts low-temperature heat from the network and raises it to the required level. For cooling, it rejects heat from the building into the network. This architecture creates a decentralized, bidirectional system where buildings can be both consumers and producers (*prosumers*) of thermal energy. The key advantage is the ability to balance thermal loads within the network; for instance, the waste heat from a building being cooled can be directly utilized by a neighboring building that requires heating. This internal heat recovery can lead to very high system efficiencies.

The divergence between 4GDH and 5GDHC paradigms highlights a fundamental choice in the design of future urban energy infrastructure: a centrally optimized, top-down system versus a decentralized, transactive, bottom-up system. The superiority of one over the other is highly context-dependent. For example, recent thermoeconomic analyses of residential districts concluded that, in the considered cases, 4GDHC system outperformed a 5GDHC system in terms of primary energy savings and $CO_2$ emission reductions (Calise et al., 2023; Gudmundsson et al., 2022). Common reasons are the lack of significant simultaneous heating and cooling demand in a purely residential area, which is the primary condition under which 5GDHC's load-balancing benefits are realized, and the significant competitive advantage provided by the economy of scale in centralized 4GDH systems, where the centralized sources benefit from access to less costly input energy. This suggests that future urban landscapes may feature a mosaic of DHC solutions, with 4GDH being more suitable for homogeneous residential areas and 5GDHC thriving in dense, mixed-use urban contexts, in parallel to existing and newly built 3GDH.

**Sixth Generation (6GDHC)** networks are a novel concept that is currently in the early stages of laboratory research. First proposed in 2010 (Weber & Favrat, 2010), 6GDHC moves beyond single-phase water and uses $CO_2$ as working fluid. A recent experimental design used two separate lines carrying $CO_2$ in liquid and vapour forms (Lima, 2025).The theoretical advantage lies in leveraging the latent heat of vaporization of $CO_2$, which could offer extremely high energy density and allow for more efficient heat transfer and smaller infrastructures. However, this concept is at a low technology readiness level, facing significant design and material challenges related to high pressures and two-phase flow control, despite the successful realization of small-scale demonstrations.

Table 1.1: Comparative Analysis of District Heating Generations.

| Gen. | Period | Heat Carrier | Supply Temp. | Key Characteristics & Concept |
|---|---|---|---|---|
| **1GDH** | ~1880s | Steam | > 150 °C | Centralized steam generation. High thermal losses. Considered obsolete. |
| **2GDH** | ~1930s | Water | > 100 °C | Centralized pressurized hot water from CHP. High heat losses. |
| **3GDH** | ~1970s | Water | < 100 °C | **Most diffused.** Standardized, pre-insulated pipes, and HIUs. |
| **4GDH** | ~2010s | Water | 50 °C to 70 °C | Low-temperature PHW. Integrates low-grade renewables & waste heat. Centralized. |
| **5GDHC** | ~2010s | Water / Brine | 10 °C to 25 °C | Ambient-temperature loop. Decentralized heat pumps (prosumers). Bidirectional heat recovery. |
| **6GDHC** | Concept | $CO_2$ | Multi-phase | Experimental concept using two-phase $CO_2$ as a high-density carrier. |

### 1.2.3 Barriers and Opportunities for District Heating in Switzerland

Despite its strategic importance for decarbonizing dense urban areas, the expansion and operation of DHNs face considerable challenges, highlighted by a comprehensive report commissioned by the Swiss Federal Department of the Environment, Transport, Energy and Communications (DETEC) (Weisskopf et al., 2022). These span economic, infrastructural, and governance domains, and overcoming them is crucial to unlocking DH's full potential within the national energy transition.

The most significant barrier is economic, stemming from the high capital intensity of DHNs and their fundamental reliance on achieving high connection densities to be viable. These networks require substantial upfront investments with long payback periods and face a high-risk

profile due to the planning uncertainties on customer adoption rates and future consumption development. This core economic vulnerability is challenged from two directions simultaneously: by the existing fossil fuel infrastructure and by the rise of decentralized renewable solutions.

On one front, DHNs must compete directly with established natural gas grids. In areas already well-served by gas, the perceived cost-competitiveness of existing, often amortized, connections is a major barrier to customer acquisition. This leads to direct competition that can undermine the business case for both infrastructures if operated in parallel. Furthermore, it introduces significant financial and organizational difficulty related to coordinating the strategic phase-out of gas and managing the resulting stranded assets.

On the second front, the DHN business case is progressively eroded by decentralized renewable solutions, mainly individual heat pumps. The report highlights a critical time urgency, as every property owner who installs an individual system represents an irreversible loss to the potential customer base. This market fragmentation is often exacerbated by competing subsidies, where public funds may support both individual solutions and DHN connections within the same designated area, steering the policy's intended effect.

The competition of heat pumps, while advancing the replacement of fossil fuels, threatens to lock in a sub-optimal decarbonization pathway. The risk is to lose the unique, large-scale benefits of DHNs: their ability to leverage diverse local and low-carbon heat sources and provide system-level thermal flexibility, which are crucial for mitigating the grid-integration challenges of mass electrification (Swiss Federal Office of Energy (SFOE), 2023).

Governance and planning complexity represents another major hurdle. The successful deployment of DH necessitates integrated, long-term municipal and cantonal heat planning. However, achieving the required coordination between diverse actors, municipalities (which may or may not own the gas utility), gas suppliers, potential DH operators (which can be public, private, or mixed entities), electricity grid companies, and property owners, is often difficult and resource-intensive. Aligning the strategic decommissioning of gas network segments with the build-out schedule of new thermal networks requires careful orchestration, robust institutional frameworks, and clear communication strategies, which are still evolving in many regions. This complexity is further amplified by the site and situation-specific nature of legal frameworks, such as for securing right-of-way, and a recognized lack of skilled specialists within public administrations to manage such multifaceted, long-term infrastructure projects.

These severe economic and logistical challenges directly threaten the viability of DHNs and thus the success of the systemic heat transition. To overcome these barriers, projects must be made more economically competitive and their planning risks significantly reduced. However, the high complexity of these projects is a key obstacle, often exceeding the personnel resources and technical capacities of the planners (Weisskopf et al., 2022). Advanced analytical approaches are therefore essential to address this complexity, allowing planners to mitigate investment risks, lower lifecycle costs, and accelerate deployment. This directly translates

into the need for fast, yet accurate simulation and optimization methods. Such tools offer crucial capabilities for better navigating long-term uncertainties (e.g., connection density, consumption development), designing more efficient networks, and reducing operating costs in this challenging environment. The next section will provide an overview of the current landscape for these methods, including their potential and open challenges.

## 1.3 The Role of Optimization

As highlighted in the previous sections, the adoption and competitiveness of DHNs are hindered by their high capital costs and financial uncertainties. Overcoming this barrier requires a systematic reduction in planning, design, and operating costs.

A primary lever to achieve this is *optimization*. Numerous studies have demonstrated that a largely untapped potential for both cost and emission reductions exists. Table 1.2 provides an excerpt of the relevant literature highlighting the significant benefits reported. Notably, the inclusion of several real-world case studies underscores that this potential is practically achievable in existing and new networks.

These studies also highlight a growing complexity and uncertainty of the modern energy context, which makes formulating and solving optimization problems substantially harder. This complexity arises from both external and internal factors. Externally, long-term planning is complicated by significant uncertainties from climate change, sector coupling, heat demand reduction, competing technologies, and the integration of intermittent heat sources. Internally, the transition toward low-temperature, multi-source networks with prosumers, distributed storage, and dynamic control strategies demands far more sophisticated coordination.

Concurrently, the increasing availability of digital technologies and sensor data is reshaping how DHNs can be monitored and controlled. Both the International Energy Agency (IEA) and the Intergovernmental Panel on Climate Change (IPCC) emphasize that digitalization, through advanced modeling, data integration, and automated control, is essential to unlock the flexibility and efficiency needed in future integrated energy systems (*Climate Change 2022*, 2022; IEA, 2023). Optimization acts as the core decision-making layer that leverages this digitalization, converting data and models into actionable strategies (Schmidt, 2021).

However, realizing this potential requires a multifaceted approach. The diversity of optimization tasks across the DHN lifecycle demands models that balance speed, accuracy, and flexibility in different ways. As a results, an ecosystem of different approaches is needed, comprising a family of models tailored to the specific requirements of the problem at hand. The following subsection explores this concept by analyzing the distinct optimization needs and trade-offs at each stage of the DHN lifecycle.

Table 1.2: Summary of optimization studies on DHNs and stated benefits

| Ref. | Goal | Results |
|---|---|---|
| (Friebe et al., 2023) | Finding optimal strategy between DH and individual supply for a municipality using an archetype-based approach. | On the synthetic case study DH showed a lower levelized cost of heating in **94%** of variants, with savings of **0.002–0.135 €/kWh** depending on energy prices and emission targets. |
| (Salenbien et al., 2023) | Cost-optimizing the topology of a DHN under 3 different heat supply scenarios using real metered demands. | The approach found a solution **5%** less costly (M€29.68 vs M€31.30) than the intuitive low-temperature source option. |
| (Chen et al., 2023) | Optimizing a real DHN topology (layout, pipe size, source location) under load uncertainty. | Construction cost reduced by **6.24%** and heat loss by **5.30%** vs. the existing system, while source location showed a potential **36.30%** cost difference between best and worst cases. |
| (Pesola, 2023) | Cost-optimizing the integration of HPs in a Finnish DHN, assessing ancillary service and open DH market participation. | Compared to historical data, optimal hybrid operation reduced OPEX by **24%** vs. DH-only. CAPEX made the investment uneconomical. |
| (Sartor & Dewallef, 2018) | Assessing the techno-economic feasibility of adding heat storage (tank or using the DHN) to an existing biomass CHP-DHN. | Without subsidies, optimal storage reduced cost of heat by **0.2–0.25 €/MWh** and $CO_2$ by **250–295 t/yr** (DHN-tank case), but tank payback was long (13 yrs). With subsidies, storage was not viable. |
| (H. Li et al., 2021) | Evaluating short-term (Water Tank, WT) and long-term (Borehole, BTES) storage in a real DHN using waste heat to mitigate supply/demand mismatch and reduce peaks. | WT shaved peak load by **31%** and saved costs by **5%** (payback < 15 yrs if eff > 80%) but did not solve mismatch. BTES increased waste heat use to **96%**, cut $CO_2$ by **8%**, saved costs by **6%**, but payback was > 17 yrs. |
| (Saloux et al., 2023) | Optimizing multi-boiler sequencing in two real Canadian DHNs through AI-based MPC (with field test). | Reduced natural gas use by **1.3%** (Site 2) and **2.8%** (Site 1) over 2-3 months, leading to 45-77 $tCO_2$-eq avoided and $10k-$20k CAD saved. |
| (Lu et al., 2021) | Cost-optimizing consumer-side control of a real DHN based on real-time indoor temperatures. | The optimized strategy reduced daily operation costs by up to **38.18%** (average **27.86%**) compared to the existing experience-based operation, while maintaining thermal comfort. |
| (de Giuli et al., 2024) | Cost-optimizing the supply temperature of a toy DHN with a surrogate model-based MPC. | Daily costs reduced by **5.9%** and heat loss by **67.5%** vs. rule-based control. |

### 1.3.1 Optimization Across the DHN Lifecycle

Optimization is applied across the entire DHN lifecycle, from initial planning and design to ongoing operation and control. Central to these optimization procedures is a model that simulates the consequences of key decisions, such as selecting pipe diameters or adjusting supply temperatures. These models invariably require a tradeoff among speed, accuracy, and the flexibility to accommodate detailed components like heat storage tanks. Since the ideal balance of these characteristics differs significantly by lifecycle stage, we adopt a three-phase classification to systematically analyze the distinct modeling needs and optimization objectives for each phase.

**(i) Preliminary Design and Planning** In the preliminary design phase, the primary goal is to rapidly explore a wide range of design possibilities to identify promising candidates for further analysis (Calixto et al., 2024; Guo et al., 2024; Hermans et al., 2023). This is particularly challenging when optimizing the pipe routing (topology) and sizing, where the combination of discrete choices (e.g., pipe placement) and the non-linear thermal-hydraulic physics render the resulting problem potentially intractable for large networks without significant simplifications (Wack, Serra, et al., 2023). Furthermore, due to the non-linear and potentially non-convex nature of the problem, optimization algorithms might converge to different local optima depending on the approach and initialization, potentially requiring multiple runs with varied parameters (Wack, Serra, et al., 2023). This complexity is further exacerbated by the need to evaluate numerous scenarios, often necessary due to high intrinsic uncertainty (e.g., in future heat demand), with long planning horizons (Chen et al., 2023; Salenbien et al., 2023). Consequently, the emphasis during this early stage is typically placed on computational speed over achieving high fidelity, although care must be taken to retain sufficient detail on key aspects, such as complex heat dynamics, to ensure the models capture essential system behavior and allow for meaningful comparisons (Calixto et al., 2024; Hermans et al., 2023; Pizzolato et al., 2018). Since detailed systems are typically specified in later design phases, model flexibility is a low priority, and ideal components are typically used.

**(ii) Detailed Design** Following the preliminary phase, detailed design focuses on refining the selected configurations and finalizing component specifications. This stage demands higher model fidelity to accurately predict system performance and interactions of the small subset of options filtered during the previous phase (Hiris et al., 2022; Höffner & Glombik, 2024; Salvestroni et al., 2021). Dynamic simulation becomes essential, often employing general-purpose tools to capture the transient behavior of the network and its components over relevant time scales (Knudsen et al., 2021; H. Li et al., 2021). Models incorporate more specific representations of chosen technologies, such as stratified thermal storage tanks, detailed heat pump performance maps or correlations, specific solar collector types, and potentially more granular network hydraulics (Delubac et al., 2021; H. Li et al., 2021; Salvestroni et al., 2021). Optimization objectives shift towards minimizing lifecycle costs (considering both investment

and detailed operational expenditures), maximizing the utilization of renewable or waste heat sources, and reducing environmental impacts like $CO_2$ emissions, while strictly satisfying operational constraints such as supply temperatures and peak load coverage (Delubac et al., 2021; H. Li et al., 2021; Nakama et al., 2023). Optimizing design variables, sometimes simultaneously with initial system operations, becomes crucial for evaluating the true potential of technologies like thermal storage (Delubac et al., 2021; Knudsen et al., 2021; Nakama et al., 2023). While computational demands increase due to model complexity and dynamics, techniques such as using representative "typical days" or optimization-based data screening can help manage the simulation load (Delubac et al., 2021; Nakama et al., 2023). Compared to the preliminary phase, the priority shifts from computational speed towards accuracy and the flexibility to model specific, chosen components in detail.

**(iii) Operational Control** Once the network is built, the optimization focus shifts to real-time operation and scheduling. The primary objective becomes minimizing operational expenditures (e.g., fuel and electricity costs) and associated GHG emissions, while dynamically matching heat supply to time-varying demand and guaranteeing thermal comfort (Lu et al., 2021; Quaggiotto et al., 2021; Saloux et al., 2023). Traditional rule-based or PID controllers often struggle with the system's complex dynamics, including transport delays and coupled thermal-hydraulic behaviors, leading to sub-optimal performance (Buffa et al., 2021; Lu et al., 2021). Advanced strategies, predominantly Model Predictive Control (MPC), are employed to address this complexity (Buffa et al., 2021; Quaggiotto et al., 2021; Saloux et al., 2023). MPC relies on a dynamic model of the system and forecasts of disturbances, such as heat load, weather, and energy prices, to compute an optimal operational plan over a receding horizon (Buffa et al., 2021; Quaggiotto et al., 2021; Saloux et al., 2023). A key challenge for MPC implementation is the trade-off in model fidelity. The models must be computationally **fast** to be suitable for real-time optimization (de Giuli et al., 2024; Quaggiotto et al., 2021), yet highly **accurate** in predicting the system's thermal-hydraulic dynamics, as precise management of heat delivery and thermal inertia is essential (Buonomano et al., 2023; Lu et al., 2021). Complex physical models are often computationally prohibitive (de Giuli et al., 2024), leading to the use of simplified linear models (Ghilardi et al., 2021; Quaggiotto et al., 2021) or emerging data-driven models such as Physics-Informed Neural Networks (PINNs), potentially with embed topological knowledge, to balance accuracy and speed (de Giuli et al., 2024; Saloux et al., 2023). Unlike the design phases, model **flexibility** to test different components is not a requirement, as the system hardware is already defined (Lu et al., 2021; Saloux et al., 2023).

These three distinct lifecycle phases highlight that no single simulation model or optimization method is universally optimal. The ideal trade-off between computational speed, model fidelity, and flexibility is highly dependent on the specific task. To translate this optimization potential from research into practice and accelerate the adoption of optimized strategies, empowering engineers and planners with accessible, open-source, and modular tools is critical (Delubac et al., 2021), especially in the context of high urgency that the Swiss DHN market is facing.

The following section reviews the state-of-the-art in existing DHN simulation tools and introduces the critical gaps that motivate this thesis.

## 1.4 Simulation Tools for DHNs

A wide array of modeling tools for DHNs is available, possessing distinct modeling paradigms, capabilities, and limitations, particularly concerning simulation speed, accuracy, scalability, flexibility, and the ability to model complex network topologies (Höffner & Glombik, 2024; Schweiger et al., 2018). Furthermore, they have different level of support for co-simulation with other tools, for which the Functional Mock-up Interface (FMI) represents the standard interface (Blochwitz et al., 2012). This section analyzes the most common modeling paradigms and respective tools, based on recent comprehensive reviews, to identify critical gaps in the current software landscape.

A first group of models, frequently used in optimization studies, is that of **general-purpose modeling tools**. These can follow two different paradigms (Schweiger et al., 2018):

- **Causal (or block-based) tools** operate on an input-output relationship. Components are pre-defined blocks where the flow of calculation is fixed (Schweiger et al., 2018).
- **Acausal (or equation-based) tools** use declarative, equation-based languages. This approach allows the modeler to describe the physical system using differential algebraic equations, and the solver determines the calculation order (Kuntuarova et al., 2024; Schweiger et al., 2018).

**TRNSYS** (TRaNsient SYStem) (Klein, 1988) is a graphical, causal simulation environment, widely used for thermal engineering, particularly for buildings and HVAC systems (Castaldo & Pisello, 2018; Kuntuarova et al., 2024; Schweiger et al., 2018). This is a *commercial* tool. It features a large standard library and is extensible with commercial libraries (e.g., TESS) and custom components (called "Types"). The primary limitation of TRNSYS stems from its causal paradigm. Modeling meshed networks with bidirectional flow is inherently difficult and requires workarounds. Acausal tools are better suited for this topology (Schweiger et al., 2018). **TRNSYS-TUD**, a custom implementation from the Technical University Dresden, solves this by using a proprietary hydraulic solver specifically developed for large meshed DHNs (Paulick et al., 2018). Existing pipe models span from the massless Type 31 and Type 709 to the advanced Type 604, or custom implementations like the Type 709cc tested by Schweiger et al. (2018), that include the pipe capacity and are therefore better suited for dynamic transient simulations. While TRNSYS can be coupled with other tools, it does not officially provide FMI support; however, an open-source solution for FMI co-simulation (as a slave) is available (Höffner & Glombik, 2024; Schweiger et al., 2018).

**Simulink** (“Simulink - Simulation and Model-Based Design”, 2025), part of the *commercial*

MATLAB environment, is a causal tool, and as such it shares the same fundamental difficulties as TRNSYS in modeling meshed networks (Schweiger et al., 2018). It is widely used for automatic control, digital signal processing, and power system simulation (Schweiger et al., 2018). However, the availability of validated components for detailed building and DHN simulations is limited (Höffner & Glombik, 2024; Schweiger et al., 2018). While some are provided in the block library **Carnot**, validation studies have shown that its pipe models (e.g., finite volume) may struggle to accurately capture both the transport delay and the thermal damping of a dynamic temperature wave simultaneously (Beckenbauer et al., 2017). Furthermore, studies have shown the computational performance of Simulink for building simulation is significantly worse than acausal tools (Schweiger et al., 2018).

**Modelica** (Fritzson, 1998) is an *open-source, acausal, object-oriented modeling language*. It is supported by both open source (**OpenModelica** (Fritzson et al., 2020)) and commercial (**Dymola** (Elmqvist et al., 1996), **SimulationX** (I., 2025)) simulation environments (Kuntuarova et al., 2024; Schweiger et al., 2018), however, modeling complex systems frequently requires the latter (Hirsch & Nicolai, 2022). Its equation-based nature makes it well-suited for modeling DHNs (Kuntuarova et al., 2024), for which several open-source, community-developed components exists. In particular, the **Modelica IBPSA Library** (Wetter et al., 2019) provides base models that are at the core of other initiative such as the **DisHeatLib** (Leitner et al., 2019), **Buildings** (Wetter et al., 2014), **AixLib** (Maier et al., 2023) and **IDEAS** (Jorissen et al., 2018) libraries. Additionally, other libraries such as **DCOL** (Febres et al., 2017) (open-source) and **TPL** (Schweiger et al., 2017) (commercial) provide further alternatives. However, the acausal paradigm introduces a significant limitation in scalability and speed. The flattening process, where all system equations are globally reduced before simulation, can become a severe performance bottleneck for large and complex district-scale models (Höffner & Glombik, 2024; Kuntuarova et al., 2024; Schweiger et al., 2018). In fact, performance benchmarks have shown that Modelica-based simulations for large networks can be up to 75 times slower than specialized solvers, rendering them impractical for the iterative loops required in large-scale optimizations (Hirsch & Nicolai, 2022).

**IDA ICE** is a *commercial* dynamic building simulation tool that also uses an acausal, equation-based paradigm (Kuntuarova et al., 2024; Schweiger et al., 2018). Unlike Modelica's global flattening, IDA ICE uses pre-compiled component models and a Differential-Algebraic Equation (DAE) solver, which has been shown to significantly exceed the performance of Dymola and Simulink, especially as model size increases (Sahlin et al., 2019; Schweiger et al., 2018). Its primary limitation is its *closed ecosystem*. According to Schweiger et al. (2018), its support for Modelica is limited and in particular it is not compatible with many commonly used district-level libraries. Furthermore, it lacks support for the FMI standard. This limitation persists: an analysis of the official software changelogs up to July 2025 (version 5.1.1) confirms that neither FMI support nor full Modelica compatibility have been implemented (EQUA, 2025).

Beyond general-purpose tools, several specialized DHN simulation tools have also been

developed and discussed in the literature. However, these are generally less established in research practice, and fewer studies employ them as the modeling layer for optimization tasks (Kuntuarova et al., 2024).

Many of these are *commercial and closed-source* tools (Brown et al., 2022; Kuntuarova et al., 2024), such as **STANET** (Fischer-Uhrig Engineering GmbH, 2025), **TERMIS** (now EcoStruxure) (Schneider Electric, 2025), **NEPLAN** ("PSI Neplan", 2025), **DistrictLab-H** (Baviere et al., 2023) and **NetSim** ("NetSim - Grid Simulation System", 2025). Due to their closed-source nature, these tools are often less transparent than open-source alternatives, especially concerning the underlying equations, are restricted in scope and are less suitable for community-driven initiative to extend their model libraries and for reproducible research. Furthermore, they often lack flexible APIs and cannot therefore be easily integrated into custom optimization procedures or used to test advanced control strategies (Åkesson et al., 2010; Hilpert et al., 2018; Kuntuarova et al., 2024).

Open-source, specialized tools on the other hand, while very fragmented, show a potential to overcome most of these issues.

Among these, **SIM-VICUS** (Hirsch & Nicolai, 2022) is a recently developed dynamic simulation software for buildings and district energy systems. The network part targets large 5GDHC networks, accurately modeling temperature-dependent fluid viscosity. It supports the FMI standard and it offers faster performance than Modelica while maintaining high accuracy. In particular, studies report simulation times in the order of hours for a single year using an adaptive time-step size, with the network model being the primary bottleneck (Hirsch & Nicolai, 2022; Hirsch & Paepcke, 2022). SIM-VICUS was recently transitioned to a commercial product, with the latest open-source version being currently still available, although its future is uncertain (Jansen et al., 2025).

Similarly, **CitySim** (Robinson et al., 2009) addresses the combined simulation of both 3GDH networks and the connected buildings, although it uses a steady-state approach, and supports the FMI standard. At present, it does not offer a structured solution to integrate custom components.

A notable subset of these open-source, specialized tools consists of libraries built natively in **Python**. This approach is valued for its flexibility; Kudela et al. (2020) argue that Python (alongside Modelica) is the best-suited language for promoting a collaborative, bottom-up approach to component creation. As a general-purpose scripting language, Python also allows models to be easily integrated into custom optimization loops, facilitates rapid prototyping and testing of different approaches, and can potentially interface with other tools.

However, like many specialized tools, the current ecosystem of Python-native libraries is lacking crucial features for dynamic, transient DHN simulation. Popular libraries such as **Pandapipes** (Lohmeier et al., 2020), **DHNx** (Röder et al., 2021), and **TESpy** (Witte & Tuschy, 2020) are all based on steady-state models. Furthermore, among these examples, only Pandapipes

supports meshed network topologies, and only TESpy allows custom components without direct modifications to the source code.

This review reveals a clear, two-fold gap in the existing tool landscape, corresponding to the distinct optimization needs of the DHN lifecycle.

First, for preliminary design and real-time operational optimization (Phases i and iii), a fundamental computational bottleneck exists. These tasks might demand thousands of iterative simulations, a process for which high-fidelity dynamic models, even the most performant, are inherently too slow. Furthermore, physics-based tools often preclude the use of efficient gradient-based optimization methods. This necessitates a different approach. Machine Learning (ML) has emerged as a powerful solution for creating high-speed, high-fidelity surrogate models in several engineering disciplines, however the landscape of ML for dynamic DHN simulation remains severely under-researched.

Second, for detailed design (Phase ii), a high-fidelity, flexible, and transparent physics-based tool is essential. Crucially, such a tool is also the prerequisite for generating the high-quality, controlled training data needed to develop the ML surrogates for the other phases. This data generation tool must be dynamic, open-source, and allow for easy, bottom-up creation of custom components and rapid prototyping, with precise control.

This thesis addresses both challenges in an integrated manner. It first introduces a new object-oriented, dynamic, and Python-native DHN model, designed for flexibility, transparency, and full programmatic control. This model then serves as a foundation for developing and training ML-based surrogate models, suited for fast, gradient-compatible optimization of both design and operations. The next section introduces the ML foundations needed as prerequisite for this second contribution.

## 1.5 Machine Learning

The preceding sections highlighted a fundamental trade-off between computational speed and accuracy in physics-based simulation tools. To address this limitation, **surrogate models** (Alizadeh et al., 2020) are developed as computationally efficient approximations of complex simulations. Supervised ML has emerged as a state-of-the-art framework for constructing such surrogates.

In its standard formalization (Bronstein et al., 2021), supervised ML is the process of learning an unknown mapping $f : \mathcal{X} \to \mathcal{Y}$ that links an input domain $\mathcal{X}$ to an output domain $\mathcal{Y}$. This is achieved through a collection of $N$ input–output samples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, which represent observations drawn from an underlying data-generating process. Rather than searching directly for $f$, the learning algorithm explores a parametrized family of candidate functions $\mathcal{F} = f_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta$, referred to as the **hypothesis space**. The task then reduces to identifying the parameters $\tilde{\boldsymbol{\theta}}$ that minimize a chosen **loss function**, quantifying the training objective

(e.g., discrepancy between predicted and true outputs). In practice, deep Neural Networks (NNs) provide a highly expressive realization of $\mathscr{F}$, capable of capturing intricate nonlinear relationships and serving as effective universal function approximators for surrogate modeling tasks (Hornik et al., 1989).

A naive approach is to begin with a selection of general, unconstrained functions. This generality is a critical weakness in engineering applications: it is not only **data-inefficient**, but it is also prone to finding **non-physical solutions** that, while fitting the training data, violate fundamental laws and fail to generalize (Desai, 2021; Meng et al., 2025; Singha, 2025). While research has shown that the lack of inductive bias can be compensated, this usually requires larger scales (Bachmann et al., 2023).

The solution is to constrain this search by embedding prior knowledge into the model's design. This set of assumptions is its **inductive bias**. Instead of a general-purpose model, we can employ biases tailored to the system's known properties. A **relational bias**, for example, assumes that the system's topology is critical. A **Graph Neural Network (GNN)** embodies this bias, as it is fundamentally structured to operate on graph data and respects the connectivity of the pipe network, immediately discarding all functions that do not (Battaglia et al., 2018; Ringsquandl et al., 2021). A **physics-informed bias** can also be applied, directly embedding known physical laws (e.g., conservation equations) into the model's architecture or training objective (Thangamuthu et al., 2022). This constrains the search to a much smaller, physically-plausible subspace of $\mathscr{F}$, yielding models that are more robust, data-efficient, and more likely to generalize correctly to unseen data.

This ML-based surrogate approach can directly address the speed-accuracy dilemma. Depending on the choice of $\mathscr{F}$, the resulting model can be **fast** at inference and, due to its strong biases, **accurate** (Alizadeh et al., 2020). Critically, the model is also fully **differentiable**. This property unlocks the use of highly efficient **gradient-based optimization** algorithms, which are rarely used at present for solving the complex, high-dimensional optimization problems inherent to DHN optimization.

## 1.6 Rationale for the Present Study

The preceding sections have established that overcoming the high capital and operational costs of DHNs is essential for achieving the targets of the Swiss Energy Strategy. The most significant and immediate opportunity for large-scale energy savings and emission reductions lies in improving the performance of the vast stock of existing 3GDH systems, often operating sub-optimally, as well as in guiding the efficient design of new extensions.

Unlocking this potential requires advances in how DHNs are optimized across their entire lifecycle, from early planning to operation. However, simulation models that underpin these optimization processes face inherent trade-offs among speed, accuracy, and flexibility, which limit their usability in different decision contexts. High-fidelity physics-based tools, while

accurate and flexible, are computationally expensive for rapid design exploration or real-time applications. This is an acceptable trade-off for design optimization, but existing open-source frameworks are generally limited in either simulation fidelity or flexibility.

Conversely, surrogate models are faster at the expense of generality, and are therefore more suitable for pre-design and operational optimization. ML-based surrogates in particular have shown remarkable success in overcoming the speed-accuracy trade-off in other domains of engineering and physics. Yet, their adoption in DHN modeling remains limited, with few studies being systematically limited in scalability, transferability or comprehensiveness.

This gap motivates the present study, which aims to develop a set of models tailored to the specific requirements of the pre-design, design, and operational phases of DHNs lifecycle.

## 1.7 Research Hypotheses

This thesis addresses the aforementioned challenges by developing a set of models for the rapid simulation and efficient optimization of meshed DHNs. In particular, it focuses on providing suitable solutions for the distinct needs of the **pre-design**, **design**, and **operation** phases of a network's lifecycle. The work is structured to demonstrate the following three core hypotheses:

1. **For the pre-design phase, a topology-agnostic GNN, trained on a diverse range of synthetic networks, can provide rapid estimations to accelerate high-level planning.**

   In the earliest planning stages, multiple network layouts must be considered quickly. We hypothesize that a GNN surrogate model with physics-inspired inductive biases can generalize mass flow predictions in meshed DHNs to unseen network topologies, converging below the residual threshold of physics-based solvers in the majority of cases, while maintaining order-of-magnitude speedup. While not matching the precision of a topology-specific surrogate, its speed and flexibility make it ideal for initial feasibility studies and high-level topological optimization.

2. **For the design phase, a detailed, component-based physical simulation framework provides the necessary flexibility and fidelity for the evaluation of pre-design choices.**

   A high-fidelity physical model is needed for the design phase, where accuracy and flexibility are the main requirements. This approach allows for the detailed evaluation of specific network designs and the integration of custom components, which is critical for minimizing long-term investment and operational costs.

3. **For the operational phase, a specialized GNN-based model, trained on a fixed network topology, can act as a high-speed, high-accuracy surrogate model to allow real-time optimization.**

   Once a network is built, its topology is fixed. We hypothesize that a GNN surrogate trained on a fixed network topology can predict mass flow distributions in meshed

DHNs, converging below the residual threshold of physics-based solvers in more than 95% of test cases, while being faster by at least an order of magnitude. Coupled with an optimized Lagrangian thermal model, the resulting hybrid framework can reproduce real-world thermal dynamics with negligible deviations compared to fully physics-based simulations.

## 1.8 Thesis Outline

The remainder of this thesis is structured to systematically address the three core hypotheses. While the chapters follow a different order than the hypotheses were introduced, this order is chosen to build a foundational narrative where each contribution serves as a prerequisite for the next. The structure begins with the detailed physical model, which then allows the development of the specialized and, finally, the generalized surrogate models.

- **Chapter 2** details the development and verification of an open-source, high-fidelity, physics-based simulation framework. This chapter directly addresses **Hypothesis 2 (Design)**, establishing a flexible and accurate model for detailed engineering. This framework will also serve to provide the validated ground truth and the data generation engine required for the surrogate models developed in subsequent chapters.
- **Chapter 3** introduces a novel GNN-based surrogate model for simulating the hydraulics of specific DHN topologies. This model achieves significant computational speedups while maintaining comparable accuracy to the physics-based framework, at the cost of flexibility. Building upon the foundation laid in Chapter 2, it allows the near real-time simulation of full-scale meshed networks, paving the way for operational optimization.
- **Chapter 4** extends the hydraulic surrogate with a fast, differentiable Lagrangian thermal model. The combination of these two components directly addresses **Hypothesis 3 (Operation)**. Their accuracy and generalization are assessed on a real-world case study using monitoring data, and the Lagrangian model's applicability is demonstrated through the gradient-based optimization of typical control parameters. This chapter thus validates the proposed methods for both design and operational optimization.
- **Chapter 5** explores the path toward a generalizable, topology-agnostic surrogate model. With the case-specific models validated, this chapter focuses on **Hypothesis 1 (Pre-design)**, examining how the GNN framework can be trained on diverse synthetic networks to provide rapid, high-level performance estimates across arbitrary topologies, a key enabler for efficient pre-design analysis.
- **Chapter 6** concludes the thesis by summarizing the main findings and contributions, revisiting how the three core hypotheses were addressed, and outlining promising directions for future research.

# 2 Physics-based Modeling

*The content of this chapter is primarily drawn from the following publications. The material has been revised and expanded for this thesis; in particular, the DESTEST results have been updated to reflect the current pipe model used throughout the remainder of this work.*

*Boghetti, R., & Kämpf, J. H. (2024). Verification of an open-source Python library for the simulation of district heating networks with complex topologies.* Energy, 290*, 130169.*

*Boghetti, R., Peronato, G., & Kämpf, J. H. (2023). Verification of PyDHN-a Python library for the thermo-hydraulic simulation of district heating networks-through the DESTEST. In* Building Simulation 2023 *(Vol. 18, pp. 2840-2846). IBPSA.*

## 2.1 Introduction

As established in Chapter 1, the central goal of this thesis is to propose and evaluate simulation methods specifically suited to the distinct lifecycle phases of DHNs. This chapter addresses the **design phase**, where the detailed evaluation and sizing of specific technologies demands the highest degree of physical fidelity and modeling flexibility. Consequently, the primary objective of this chapter is to develop a high-fidelity, physics-based modular simulation framework that directly addresses the second core hypothesis of this work.

Being the most accurate, physics grounded model of the three proposed, this framework additionally serves a dual purpose that extends beyond the design stage and underpins the rest of this thesis. First, it will function as the ground truth benchmark against which the accuracy of the accelerated, data-driven models developed in subsequent chapters will be validated. Second, it will serve as the data generation engine required to train these models.

This second role is particularly crucial, as relying solely on historical operational data is often

too limiting. In fact, such datasets are typically sparse, lacking measurements of key variables within the distribution network, and are limited to the narrow range of conditions under which the network has been operated. To train robust surrogate models capable of generalizing across diverse scenarios, a synthetic data source that provides complete, consistent, and varied data is needed.

In fulfilling these roles, the framework developed herein is foundational: it provides the necessary simulation tool for detailed DHN design, establishes the standard for validating subsequent hypotheses, and identifies the computational bottlenecks that motivate the very search for the accelerated methods this thesis proposes.

## 2.2 State of the Art

This section reviews the state of the art in simulation methods for 3GDH networks to contextualize the chapter's contributions. Section 1.4 previously presented an overview of the existing tools that integrate these methods and are critical for their practical adoption.

### 2.2.1 Physical Modeling

Most state-of-the-art methods for the simulation of 3GDH networks model the problem as the flow of an incompressible fluid with constant properties through 1-dimensional subsequent elements connected by junctions. This allows the formalization of the network as a directed graph, in which pipes are encoded as edges while, depending on the method, other elements such as consumers and producers are encoded as either edges or vertices. As a consequence of the hypothesis of incompressible fluid with constant properties, the hydraulic and thermal modeling are in practice frequently decoupled, as their interaction has a small effect on the results. Hydraulic simulations are typically conducted in a steady-state manner, as pressure waves propagate in the network at a speed around $1000\,\mathrm{m\,s^{-1}}$, whereas thermal simulations are usually dynamic, accounting for temperature variations that propagate at the water flow velocity (Menapace et al., 2020). In some cases, a steady-state model is instead used for the thermal part, for example in simulations aimed at topological optimization of the piping network (Wack, Baelmans, et al., 2023).

#### Hydraulic Behavior

The first method to simulate the hydraulic behavior of meshed water networks was proposed by Hardy Cross (Cross, 1936), and consisted in iteratively correcting the flows in each internal loop. More recently, matrix-based methods were introduced, allowing to find the pressure distribution in the network with higher computational efficiency. They can be divided into two categories as identified in (Osiadacz & Pienkosz, 1988), namely the node methods and the loop methods. According to Osiadacz and Pienkosz (1988), loop methods are to be preferred,

since they have better convergence characteristics, are more computationally efficient and are less sensitive to initial conditions. In both approaches, the resulting system of equations is commonly solved using root-finding algorithms such as the Newton-Raphson method. At present, most simulation tools use a node method, with the notable exception of CitySim.

For many optimization problems, the mass flows at the network's peripheral elements are considered boundary conditions. This is the case, for example, when consumers have a known energy demand. In these situations, if the network is radial, the resulting system of equations is determined, and flowrates in the rest of the network can be immediately found without the need for iterative solvers (Dénarié et al., 2023). This concept extends to meshed networks by splitting the topology into a tree and a co-tree. The decomposition reduces the governing system of equations into a smaller set expressed in terms of the co-tree flows, which become the independent variables of the problem (Osiadacz & Pienkosz, 1988; Rahal, 1995).

#### Thermal Behavior

A comprehensive overview of the main methods used for solving the dynamic **thermal behavior** is given in (Dancker & Wolter, 2021; Maurer et al., 2021). The approaches are either Euler-based (such as the element method and the node method (Benonysson, 1991)) where the evolution of temperature in fixed discretization points is evaluated, or Lagrange-based (such as the plug-flow method (Oppelt et al., 2016)), where volumes of water are tracked throughout the network. The latter are generally harder to implement, but have better performance with a lower computational complexity (Dancker & Wolter, 2021; Sartor et al., 2018). While most methods solve the nodal energy balance to find boundary conditions, some Lagrange-based works pass water volumes directly to the next element without aggregation at the node level (Oppelt et al., 2016). Euler-based methods are common in specialized dynamic simulation tools such as SIM-VICUS, while general purpose frameworks offer both choices.

Regarding model fidelity, Capone et al. (2021) investigated the effect of different modeling assumptions by comparing four increasingly detailed pipe models. After evaluating them on a single pipe, they applied the two best-performing models to a full DHN simulation using a node method for the hydraulic part and an Eulerian-based approach for the thermal simulation. They found that the pipe's inner layer (steel, in their case) has a significant influence and should be considered, whereas the insulation's heat capacity can be neglected without loss of accuracy.

#### Integrated Models and Validation Studies

Several works exist in the literature using different combinations of these methods that have been verified on real networks. Tol (2021) proposed a matrix-based method to find end-user flow rates based on pumps head lifts and valve positions with a focus on radial networks. The method is based on the identification of closed loops between the heat plants and the

consumers in which the energy conservation is enforced. The method was tested on a real case study, although not compared to monitoring data. A similar approach based on the decomposition of a radial network in cycles was previously presented by Stevanovic et al. (2007), which discretized the network into a series of loops passing through the heat plant and each substation. Tests on a real radial network showed relative errors between the predicted and measured flow rates mostly below 5%, with the highest error being just below 10%. In order to avoid iterative processes to solve the momentum equations, Guelpa and Verda (2019) proposed to split meshed networks into a tree and a co-tree part to be solved separately. Using a linear regression model fit to the monitoring data they estimated the water flow rates in the eliminated branches based on the known flow rates in the heat plants. Using this information and the known energy demand of consumers as boundary conditions, they could then derive the mass flows in the tree part. The results of the simplified method, combined with a transient thermal model, on the simulation of a meshed network with several internal loops showed good agreement with those of a full model.

An important step towards both fast and accurate thermal simulations is the implementation of methods based on a Lagrangian approach, which has demonstrated good accuracy when used to simulate testbed pipes (Dénarié et al., 2019; Sartor & Dewalef, 2017). In this direction, Oppelt et al. (2016) proposed to use a loop method combined with a plug-flow model that keeps track of the volumes of water throughout the pipeline with no volume aggregation at node level. A similar model was then implemented by van der Heijde et al. (2017) and tested against monitoring data. The comparison of measured and simulated temperature profiles for selected substations showed a distribution of the relative difference centered at nearly 0°C with a maximum standard deviation of 1°C.

Finally, in a recent work, Dénarié et al. (2023) presented a tool for fast dynamic simulations of large-scale DHNs that was implemented in a GIS-based workflow. The hydraulic simulation is solved using the Hardy Cross method, while a hybrid approach is employed for the heat transmission. An accurate Lagrange-based approach (Dénarié et al., 2019) is used for pipes in radial networks, while a lumped capacity method is used for meshed networks. The model is verified on a real network for which hourly monitoring data was available. Results show an average error of -0.1°C with a standard deviation of 1.12°C on the return temperature at the production plant, and average errors between 0.39°C and -1.33°C with a maximum standard deviation of 6.30°C on the supply temperature of selected consumers.

## 2.3 Contribution

This review, combined with the tool overview in Section 1.4, highlights a clear gap in the DHN simulation landscape: the absence of a component-based, open-source framework within an application-agnostic environment that allows reproducible research and high-quality dataset generation for data-driven modeling. Furthermore, although highly accurate Lagrangian methods are available for thermal simulations, they are rarely implemented in open-source

tools suitable for meshed networks.

This chapter directly addresses this gap by introducing a novel, open-source simulation framework for DHNs, implemented as a Python package. It is specifically architected to support the detailed design and analysis of 3GDH systems, the most widespread generation, while providing a robust foundation for future research applications.

The principal contributions of this work are:

- **A Robust and Fast Hydraulic Solver**: we implement a computationally efficient hydraulic model based on a simplified loop method. This provides a fast and stable solution for the steady-state hydraulics of large, arbitrarily meshed networks.
- **The Integration of a High-Fidelity Lagrangian Thermal Model**: This work provides a robust and computationally efficient implementation of the advanced Lagrange-based thermal propagation model from Dénarié et al. (2019). While the core method existed, our contribution lies in its seamless integration within a full simulation framework for meshed networks that is scalable to large system and supports custom components.
- **An Open-Source, Extensible Framework**: The entire framework is released as an open-source Python package. Its bottom-up, object-oriented design allows researchers to easily define and integrate custom network components (e.g., new types of producers, consumers, or storage units), making it an ideal tool for both detailed engineering studies and for generating synthetic datasets to train machine learning models.

## 2.4 Methodology

This section outlines the main physical models used in this work (Section 2.4.1), including the network graph representation, the overall convergence loop, and the modeling of network components. Further verification of the model against monitoring data from a real DHN will be presented later in this thesis in Chapter 4.

### 2.4.1 Physical modeling

A decoupled simulation is carried out using a flexible approach that allows the use of both dynamic and steady-state models. The implementation described in this work employs a steady-state model for the hydraulic part and a Lagrange-based model for the thermal part, extending the work presented by Dénarié et al. (2023). As depicted in Figure 2.1, the DHN is considered to be made of several components, which are ideally placed within one of the two lines - supply or return - or between them (i.e., have one endpoint in each line). Within this work, the latter will be referred to as *leaf* components.

The working fluid is assumed to be incompressible, pressurized and to remain in liquid state

throughout the whole simulation. It flows within the network without a free surface at a flow rate $\dot{m}$ (kg s$^{-1}$), pressure $p$ (Pa) and temperature $\theta$ (°C). By default, water is used as fluid and its properties are considered to vary with temperature according to the relationships found in (Popiel & Wojtkowiak, 1998).

### Graph representation

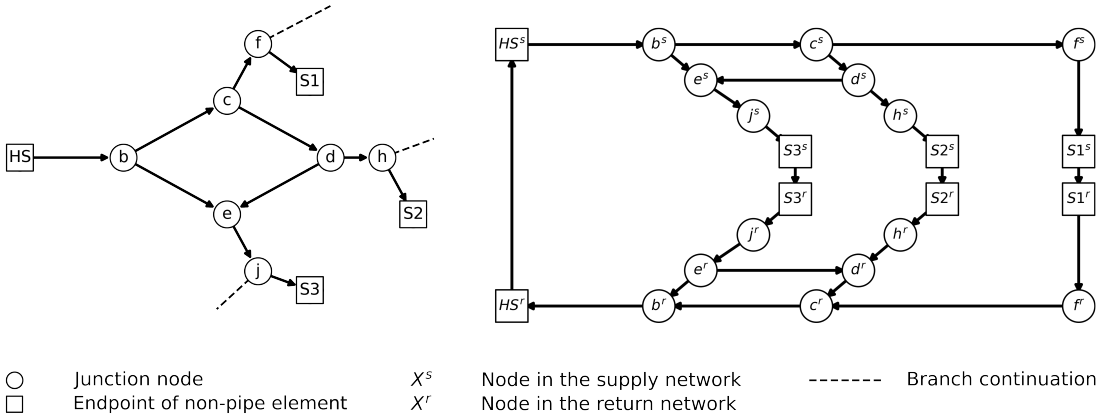



Figure 2.1: Example of a DHN and its graph representation. **Left:** GIS-style DHN layout, where supply and return lines are represented as a single connection. The heating plant is labeled HS, substations as S, and junctions with lowercase letters. **Right:** Graph representation used in this work, where supply and return lines are explicitly separated.

The DHN is mathematically represented as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}|$ nodes and $\mathcal{E}$ edges, where edges contain the different components of the network. In this work, the graph is constrained so that vertices can only have total degree 2 or 3 and open-loop parts are not allowed, as depicted in Figure 2.1. The topology of the network is then mapped using the vertex-edge incidence matrix $\boldsymbol{A} \in \mathbb{Z}^{|\mathcal{V}|\times|\mathcal{E}|}$, which encodes the relationship between vertices and edges, built such that each element $a_{ij} \in \boldsymbol{A}$ is 1 if $e_j$ in an incoming edge of vertex $v_i$, $-1$ if it is a leaving edge, and 0 otherwise.

We also encode the the network graph based on its *cycles* - closed paths in the graph where only the first and last nodes coincide. This is done by using a subset of $|\mathcal{C}| = |\mathcal{E}| - |\mathcal{V}| + 1$ cycles, called a *fundamental cycle basis*, with the property that all remaining cycles in the network graph are a linear combination of the members of this subset, which are called *fundamental cycles*. The corresponding *cycle-edge incidence matrix* $\boldsymbol{B} \in \mathbb{Z}^{|\mathcal{C}|\times|\mathcal{E}|}$ is then defined such that each element $b_{ij} \in \boldsymbol{B}$ is either 1 if cycle $c_i$ contains edge $e_j$ and their directions agree, $-1$ if they have opposite directions, 0 otherwise.

For the purpose of vector notation, we define a fixed but arbitrary ordering for these sets. This ordering creates a one-to-one mapping between the set elements and a range of indices. For example, the edges in $\mathcal{E}$ are mapped to the indices $\{1,\ldots,|\mathcal{E}|\}$. Consequently, a property $x$ associated with all edges can be represented as a vector $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}$, where the $j$-th component $x_j$

corresponds to the value of the property for the $j^{\text{th}}$ edge.

### Overall convergence

The convergence loop aims at finding the average mass flow, total pressure difference and temperature distribution in all the edges of the network graph given some boundary conditions in the leaf components. In the view of a more modular and efficient approach, the hydraulic and thermal parts of the simulation are decoupled. However, as the fluid properties are re-evaluated based on the temperature, the proposed tool allows for iterating through the two parts multiple times and as such coupling their results.

The hydraulic simulation is carried out using a modified version of the loop method. The problem is governed by the Kirchhoff's Voltage Law:

$$\sum_{j=1}^{|\mathscr{E}|} b_{ij} \Delta p_j = 0 \quad \forall i = 1, \ldots, |\mathscr{C}| \tag{2.1}$$

and the flow equation:

$$\Delta p_j = \phi_j(\dot{m}_j) \quad \forall j = 1, \ldots, |\mathscr{E}| \tag{2.2}$$

Knowing that:

$$\dot{m}_j = \sum_{i=1}^{|\mathscr{C}|} b_{ji}^T \tilde{m}_i \quad \forall j = 1, \ldots, |\mathscr{E}| \tag{2.3}$$

where $\tilde{m}_i$ ($\mathrm{kg\,s^{-1}}$) is the *fundamental mass flow* of the $i^{th}$ cycle in $\boldsymbol{B}$. Putting together eqs. (2.1) to (2.3) we obtain:

$$\sum_{j=1}^{|\mathscr{E}|} b_{ij} \phi_j(b_{ji}^T \tilde{m}_i) = 0 \quad \forall i = 1, \ldots, |\mathscr{C}| \tag{2.4}$$

Equation (2.3) states that the mass flow in one edge is the directed sum of the fundamental flows passing through that edges. As a consequence, if the $j^{th}$ edge is exclusive to the $i^{th}$ fundamental cycle, and $\dot{m}_j$ is known, then it must hold that $\tilde{m}_i = \dot{m}_j$. We assume that this is

true for a subset of $|\mathscr{C}^*| < |\mathscr{C}|$ leaf components in the network (within this work, the consumers and secondary producers) and that each fundamental cycle can contain at most one edge with known mass flow. Arranging the problem such that the cycles of indexes 1 to $|\mathscr{C}^*|$ are solvable in this way, the system of eq. (2.4) is reduced to:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{j=1}^{|\mathscr{E}|} b_{ij}\phi_j(b_{ji}^T \tilde{m}_i) = 0 \quad \forall\, i = |\mathscr{C}^*| + 1, \ldots |\mathscr{C}| \tag{2.5}$$

The system in eq. (2.5) is then solved iteratively using the Newton-Raphson method, which at each iteration $k+1$ finds a correction factor $\boldsymbol{\Delta x}^k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$ to the previous guess $\boldsymbol{x}^k$ by solving the linear system:

$$\boldsymbol{J}(\boldsymbol{x}^k)\boldsymbol{\Delta x}^k = -\boldsymbol{f}(\boldsymbol{x}^k)\,, \tag{2.6}$$

where $\boldsymbol{J}(\boldsymbol{x}^k)$ denotes the Jacobian matrix of the residual function $\boldsymbol{f}$ evaluated at the current estimate $\boldsymbol{x}^k$. At each step, the mass flows in the network edges are then computed using eq. (2.3), while the pressure differences in edges without an imposed mass flow are computed using eq. (2.2). For the remaining edges, the pressure difference is instead computed as the negative directed sum of pressure differences of the other edges in the corresponding fundamental cycle. This is necessary since fixing a function $\phi$ for the $|\mathscr{C}^*|$ edges with imposed mass flow would make the problem overdetermined.

This process is repeated until a predefined maximum number of iterations is reached or the following condition is satisfied:

$$\max_{|\mathscr{C}^*|+1 \le i \le |\mathscr{C}|} \left| f(x_i) \right| \le \delta \tag{2.7}$$

where $\delta$ is a parameter of the solver.

The method here described benefits from finding a cycle basis that maximizes $|\mathscr{C}^*|$, as this number is directly linked to the constraints used to reduce the solution search space. A fast way to compute such a cycle basis is the following. First, all the edges with leaf components are removed from the original network graph except for one, with the constraint that the mass flow in this edge is not fixed. Then, a spanning tree of this subgraph, treated as undirected, is computed. By adding back each edge of the original graph that is not in the spanning tree a cycle will close, that contains either one leaf component with imposed mass flow (*boundary cycles*) or none (*internal cycles*). The list of cycles found with this approach constitutes the

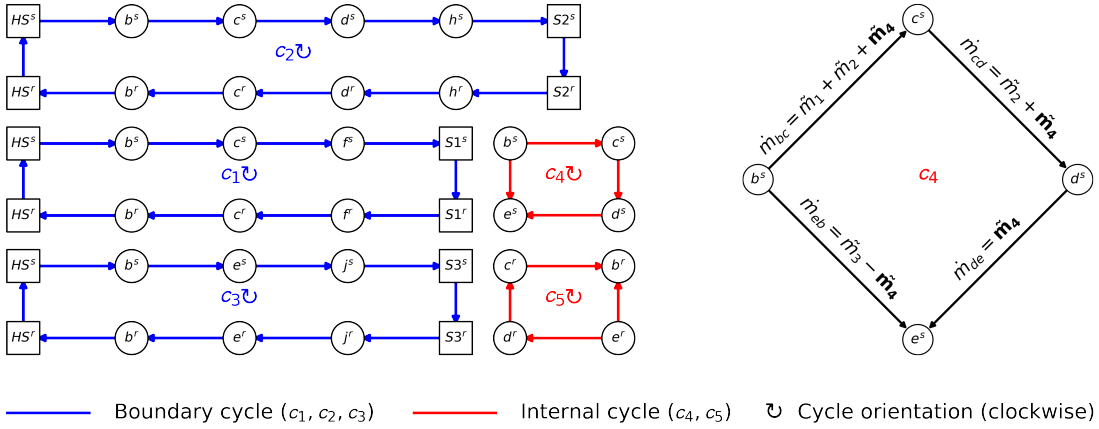



Figure 2.2: Cycle-based decomposition of mass flows. **Left:** Cycles of the graph in Figure 2.1. Boundary cycle flows are fully determined by the mass flow given at their unique substation. **Right:** Superposition principle for cycle $c_4$. Edge mass flows are obtained by summing signed contributions from the cycle flows $\tilde{m}_i$. The contribution of $\tilde{m}_4$ is shown in bold, signs reflect the cycle-edge incidence matrix.

fundamental cycle basis of the network graph. A depiction of this decomposition is given in Figure 2.2.

The thermal simulation is then carried out by solving the heat balance of nodes.

Defining the matrices $\boldsymbol{A}^+$ and $\boldsymbol{A}^-$ such that:

$$a^+_{ij} = \frac{|a_{ij}| + a_{ij}\mathrm{sgn}(\dot{m}_j)}{2}, \tag{2.8a}$$

$$a^-_{ij} = \frac{|a_{ij}| - a_{ij}\mathrm{sgn}(\dot{m}_j)}{2}. \tag{2.8b}$$

The system of equations for the thermal simulation can be expressed as (Wang et al., 2016; S. Zhang et al., 2021):

$$\sum_{j\in\mathscr{E}^+_i} a^+_{ij}\,|\dot{m}_j|\,c_{\mathrm{p},j}\theta_{\mathrm{out},j} = \sum_{j\in\mathscr{E}^-_i} a^-_{ij}\,|\dot{m}_j|\,c_{\mathrm{p},j}\theta_{\mathrm{in},j} \quad \forall i = 1,\dots,|\mathscr{V}|, \tag{2.9}$$

$$\theta_{\mathrm{out},j} = \psi_j(\theta_{\mathrm{in},j}) \quad \forall j \in \mathscr{E}^+_i \cup \mathscr{E}^-_i, \tag{2.10}$$

$$\theta_{\text{in},j} = \theta_i \quad \forall\, j \in \mathscr{E}_i^- \,, \tag{2.11}$$

where $\psi(\cdot)$ is a generic mapping from inlet to outlet temperature, $\mathscr{E}_i^+$ and $\mathscr{E}_i^-$ are respectively the set of indices of incoming and leaving edges of vertex $i$, $c_\text{p}$ ($\text{J}\,\text{kg}^{-1}\,\text{K}^{-1}$) is the specific heat capacity of the fluid and the subscripts *in* and *out* indicate respectively the values of the *inlet* and *outlet* temperature. Putting together eqs. (2.9) to (2.11), a system of $|\mathscr{V}|$ equations is found, that is then solved using the Newton-Raphson method previously described.

If not specified otherwise, in this work the maximum number of iterations for the hydraulic simulation is set to 100 and the convergence criterion to $\delta = 1\,\text{Pa}$ (other studies commonly employ a threshold of $\delta = 100\,\text{Pa}$ (Cognet, 2020)) while for the thermal simulation these values are set to 100 and $1 \times 10^{-12}\,\text{W}$ respectively. The fundamental mass flows are initialized as $1 \times 10^{-5}\,\text{kg}\,\text{s}^{-1}$ and the node temperature is initialized as 50 °C.

#### Components modeling

Pipes are modeled as hollow cylinders of length $\ell$ (m) with rugosity $\epsilon$ (m) and internal radius $r_0$ (m), covered by an insulation layer and a casing. A height difference $\Delta z$ (m) might occur between the two ends of the pipe. Pressure losses are then determined using the well known Darcy-Weisbach equation plus the hydrostatic pressure:

$$\Delta p = \frac{\ell f_D}{4\pi^2 \rho r_0^5} |\dot{m}| \dot{m} + \rho g \Delta z \tag{2.12}$$

where $f_D$ (-) is the friction factor and $g = 9.81\,\text{m}\,\text{s}^{-2}$ the gravitational acceleration. The friction factor is computed as a function of the Reynolds number $Re$ (-). For laminar flow, occurring at $Re \leq 2300$, the friction factor is computed as:

$$f_D = \frac{64}{Re} \tag{2.13}$$

For turbulent flow ($Re \geq 4000$) Haaland equation, valid for $4000 \leq Re \leq 10^8$ and $10^{-6} \leq \epsilon/(2r_0) \leq 0.05$, is applied (Zeghadnia et al., 2019):

$$f_D = \left\{ 1.8 \log_{10} \left[ \left( \frac{\epsilon}{7.4 r_0} \right)^{1.11} + \frac{6.9}{Re} \right] \right\}^{-2} \tag{2.14}$$

Finally, for the transition regimen, occurring within $2320 < Re < 4000$, the relationship given by Hafsi (2021) is used.
For the thermal part, the two-capacity model described in (Dénarié et al., 2019) is used. In order to increase the flexibility of the model, the linear heat-transfer coefficient for the insulation $h_{\text{ins}}$ is replaced with that for the combination of insulation, casing and external convection $h_{\text{out}}$ ($\text{W m}^{-1}\,\text{K}^{-1}$):

$$h_{\text{out}} = \frac{1}{\frac{1}{h_{\text{ins}}} + \frac{1}{h_{\text{cas}}} + \frac{1}{h_{\text{ext}}}} \tag{2.15}$$

$$h_{\text{cas}} = \frac{2\pi k_{\text{c}}}{\ln\left(r_{\text{cas}}/r_{\text{ins}}\right)} \tag{2.16}$$

Where $r_{\text{cas}}$ and $r_{\text{ins}}$ are the radii of casing and insulation respectively, while $k_{\text{cas}}$ ($\text{W m}^{-1}\,\text{K}^{-1}$) is the thermal conductivity of the casing material.

The outlet pipe temperature is then computed as the weighted average of the temperatures in the leaving volumes.

In the present work, the pressure difference in producers and consumers is either imposed or computed as the negative sum of the other elements in the respective cycle, as the mass flow is imposed based on measurements. The outlet temperature of producers is also imposed according to measurements, while that of consumers is computed from the simulated inlet temperature by imposing the measured temperature difference.

## 2.5 Implementation and Verification

The thermal-hydraulic model described in the previous chapter has been implemented as an open-source Python library (PyDHN ("Idiap/Pydhn: Python Library for Modeling, Simulating, and Analyzing District Heating Networks", 2023)). To ensure the framework's accuracy and robustness, a comprehensive, two-stage verification process was undertaken.

This process begins at the component level (Section 2.5.1), where the core thermal pipe model is verified in isolation. This is achieved by comparing its simulation results against high-resolution experimental data from a well-documented aerial pipe experiment. This initial step confirms the fundamental accuracy of the thermal implementation and benchmarks its performance against established commercial tools, particularly under dynamic conditions.

Then, the integrated framework is assessed against the standardized IBPSA Project 1 DESTEST benchmark (Section 2.5.2). This second stage verifies the model's ability to perform simula-

tions of a complete, albeit simplified, radial DHN in both steady-state and dynamic conditions.

A final verification on a real-world, meshed network, including a comparison with other models proposed in this thesis, is then presented in Chapter 4.

### 2.5.1 Pipe Model

The first step in validating the proposed simulation framework is to verify the accuracy of the thermal pipe model in isolation. To this end, the model is evaluated against the high-resolution, open-access experimental dataset of Schweiger et al. (2018), which includes benchmark results from widely used simulation tools. This verification step aims to both confirm the correctness of the implementation and justify the model's selection over alternative approaches.

#### Experimental Setup

The original experiment consisted of a 60.3 m insulated aerial pipe where water was heated from 27 °C to 74 °C at a constant flow rate. The outlet temperature was recorded with one-second granularity.

To replicate these conditions, the model was configured with the pipe's known physical characteristics, assuming an internal pipe rugosity of 0.007 mm and an external heat transfer coefficient of $9.35\,\mathrm{W\,m^{-2}\,K^{-1}}$. The performance of the model is evaluated by comparing its simulated outlet temperature against the measured data and the results from several established simulation tools used in the original study (Dymola, Simulink, TRNSYS, IDA-ICE). Following the original study's methodology, the analysis focuses on three key time frames, using the Root Mean Square Error (RMSE) as the primary metric:

- **Case a (Overall):** The entire 30-minute experiment.
- **Case b (Dynamic):** A 100-second period with a steep temperature ramp.
- **Case c (Steady-State):** A 300-second period of stable, high-temperature operation.

#### Results

The results, shown in Figure 2.3 and Table 2.1, demonstrate a high degree of accuracy for the proposed model across all evaluated conditions.

Visually, the simulated outlet temperature aligns well with the experimental measurements, accurately capturing the thermal dynamics during both stable and transient phases.

Quantitatively, the model's performance is superior to the benchmarked tools, particularly under challenging dynamic conditions. As detailed in Table 2.1, the proposed model achieves the **lowest RMSE** for the overall experiment (0.16 K in Case a) and, most notably, for the highly

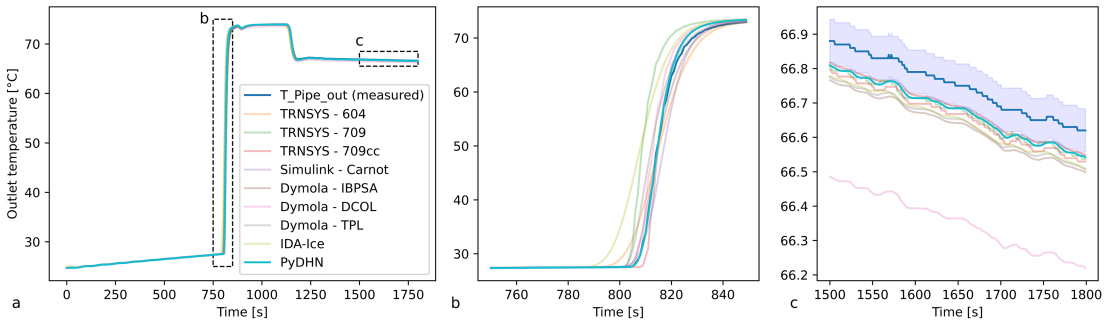



Figure 2.3: Verification of the aerial pipe model against measured data. The shaded area shows the measurement range, taking into account sensor tolerance. The simulated output of the proposed model closely tracks the real-world measurements.

dynamic phase (0.59 K in Case b). In steady-state conditions (Case c), its performance is on par with the best-performing commercial tools, achieving a joint-lowest RMSE of 0.07 K.

Table 2.1: RMSE between the measured outlet temperature and the simulation output of the proposed model and several reference tools. Best results for each case are highlighted in bold.

| | Ours | IDA ICE | Dymola | | | Simulink | TRNSYS | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | DCOL | IBPSA | TPL | Carnot | 604 | 709 | 709 cc |
| a. | **0.16** | 1.55 | 0.57 | 0.23 | 0.61 | 0.50 | 0.53 | 1.46 | 0.31 |
| b. | **0.59** | 6.49 | 2.02 | 0.91 | 2.52 | 2.07 | 2.23 | 6.12 | 1.27 |
| c. | **0.07** | 0.11 | 0.40 | 0.12 | 0.11 | **0.07** | **0.07** | 0.08 | 0.09 |

Furthermore, the model's maximum absolute error throughout the entire simulation was only 1.67 K, a significant improvement over the next-best reference tool, which had a maximum error of 3.43 K. This initial verification confirms that the core thermal model is not only physically sound but also highly accurate and suitable for aerial pipes, providing a solid foundation for the full network simulations that follow.

### 2.5.2 DESTEST

To further assess the framework's capabilities, the model is evaluated against the District Energy Simulation Test (DESTEST) benchmark, a standardized test developed as part of IBPSA Project 1 (Saelens et al., 2019). This verification specifically targets the model's ability to perform both steady-state and dynamic simulations of a toy DHN through two distinct exercises.

#### Experimental Setup

The core of the DESTEST is an **inter-model comparison protocol**, where the outputs of different simulation tools are compared across several Common Exercises (CEs) numbered

from 0 to 3. The presented model is tested on CE0, a steady-state exercise, and CE1, a dynamic exercise.

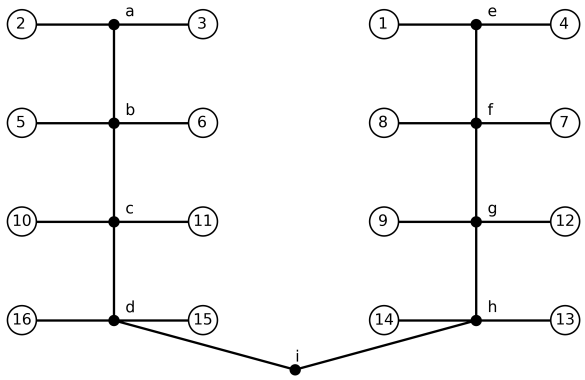



Figure 2.4: Layout of the radial network used in the DESTEST Common Exercises. The node $i$ is the source, and the numbered circles are the consumers.

For both CEs, the case study is the toy radial network depicted in Figure 2.4. It consists of 16 single-family dwellings fed by a single heat source. The pipes are defined as hollow cylinders with two layers: a crosslinked polyethylene core and a surrounding insulation. The specific pipe diameters, lengths, and material properties are provided by the CE protocol.

The heat-carrier fluid is assumed to have constant properties, independent of temperature or pressure, corresponding to pure water at 50 °C. The values used are: density ($\rho = 988\,\mathrm{kg\,m^{-3}}$), specific heat capacity ($c_\mathrm{p} = 4180\,\mathrm{J\,kg^{-1}\,K^{-1}}$), thermal conductivity ($\lambda = 0.64\,\mathrm{W\,m^{-1}\,K^{-1}}$), and kinematic viscosity ($\nu = 5.5 \times 10^{-7}\,\mathrm{m^2\,s^{-1}}$ ).

The simulations are based on several thermal assumptions. The heat source is ideal, providing a constant supply temperature of 70 °C. Each dwelling substation is also modeled as ideal, enforcing a fixed temperature difference of 30 K between its supply and return. The soil is not explicitly modeled; instead, a constant temperature of 10 °C is imposed as the boundary condition at the external pipe surface. Moisture transport and related phenomena in the surrounding ground are not considered.

Hydraulic assumptions are also made to simplify the model. Pressure losses are calculated solely based on pipe friction, while singular losses are neglected. All components are assumed to lie at the same elevation, eliminating hydrostatic effects. Furthermore, both the substations and the heat plant are considered hydraulically ideal, meaning they introduce no pressure drop.

**Network Common Exercise 0 (CE0)** The first exercise, CE0, focuses on a steady-state simulation, with the goal of assessing the correctness of the fundamental equations for pressure

drop and heat loss calculation. The exercise imposes a uniform mass flow of $553\,\mathrm{kg\,h^{-1}}$ at the heat source. As the proposed framework employs a quasi-dynamic thermal model, the simulation is run until steady-state conditions are reached, which aligns with the benchmark's requirements for dynamic tools.

The verification is conducted by comparing 18 key physical indicators, including mass flow, pressure drop, fluid temperature, heat loss and heat load against the results of reference tools (Modelica Buildings library, TRNSYS, and SIM-VICUS). Although the results are single steady-state values, the Normalized Mean Bias Error (NMBE) is used as the primary Key Performance Indicator (KPI) for consistency with the benchmark's methodology. The DESTEST protocol provides a comparison tool that automates this KPI computation and the generation of related plots.

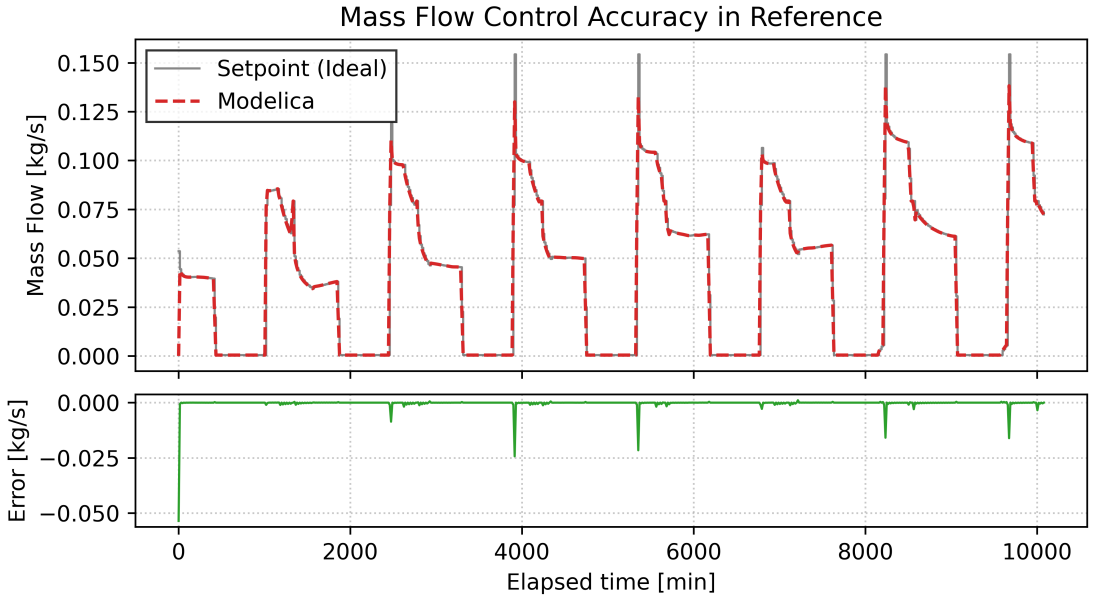



Figure 2.5: Mass flow tracking performance at Substation 1. **Top:** The solid line represents the common mass flow demand profile used for all buildings in DESTEST CE1, serving as the boundary condition for the simulation. The dashed line shows the actual mass flow in the reference Modelica simulation. **Bottom:** The control error (actual minus setpoint) illustrates that the Modelica controller tends to undershoot response, visible as distinct negative spikes, whenever the demand changes rapidly.

**Network Common Exercise 1 (CE1)** The second exercise, CE1, extends the analysis to a dynamic simulation over a 7-day period. The goal is to assess the correct transient modeling of heat losses, temperature distribution, and pressure distribution under more realistic, time-dependent conditions. While using the same network topology and component assumptions as CE0, CE1 introduces a cyclic load profile for all 16 buildings, updated every 10 minutes (Figure 2.5). Given this power schedule, each substation adjusts its mass flow rate to maintain

a constant 30 K temperature difference. To avoid unrealistic return temperatures during start-up after zero-demand periods, the outlet temperature is constrained to remain above 28 °C whenever the substation is active.

For the comparison, 9 KPIs are evaluated at 15-minute intervals, but the official DESTEST comparison data is currently not available through the comparison tool. The evaluation is therefore conducted against the only available reference data, a single study using the Modelica Buildings library and in particular the plug flow pipe model, with a total of 8 KPIs reported. This means the reference is not an average computed from multiple submissions, limiting its use as a formal benchmark. Therefore, while the NMBE is still calculated against this single reference for consistency, the primary evaluation is based on the visual and qualitative agreement between the simulated time-series data from the proposed model and the reference tool.

Additional choices were made for the simulation setup, informed in part by this comparison context. As the DESTEST documentation does not prescribe a specific time step size, and since minimum fluid traveling time in any pipe is around 25 s, the simulation is run with a time step of 20 s. A three-hour preconditioning period is also simulated to mitigate the impact of arbitrary initial conditions. Finally, to ensure a consistent comparison with the aforementioned Modelica reference model, a minimum flow rate of 1.77 $\mathrm{kg\,h^{-1}}$ at substations is implemented, similar to a bypass. This is not a requirement for the proposed framework, but it mimics the conditions of the reference tool, which requires a minimum flow for numerical stability.

#### Results CE0

The list of errors for all KPIs is given in Table 2.2. The plots generated by the comparison tool are given in Figs. 2.6 to 2.9.

Table 2.2: Results of the proposed framework on DESTEST CE0. The value reported is the NMBE with respect to the reference value.

**Results for nodes**

| KPI | Node | Supply | Return |
|---|---|---|---|
| $\dot{m}$ | $i$ | -0.07 | - |
| $T$ | $i$ | 0.00 | -0.32 |
| $T$ | $h$ | 0.00 | -0.26 |
| $T$ | $g$ | 0.00 | -0.33 |
| $T$ | $f$ | 0.00 | -0.38 |
| $T$ | $e$ | 0.00 | -0.40 |
| $T$ | $1$ | -0.01 | -0.40 |
| $\dot{Q}$[1] | $i$ | 0.63 | - |

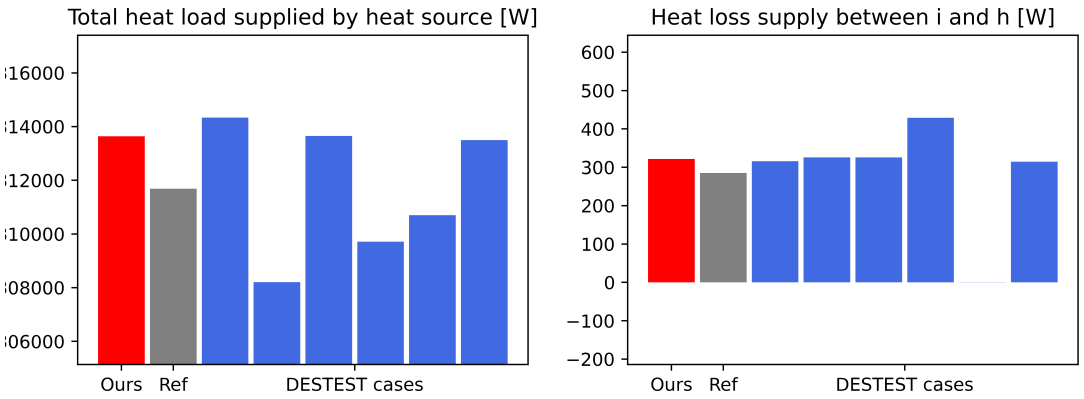
[1] *Total heat load.*

**Results for edges**

| KPI | Edge | Supply | Return |
|---|---|---|---|
| $\Delta p$ | $i$-$e$ | -1.04 | - |
| $\Delta p$ | $a$-$i$ | - | -3.55 |
| $\Delta p$ | $i$-$h$ | - | -5.18 |
| $\dot{Q}$[2] | $i$-$h$ | 12.75 | - |

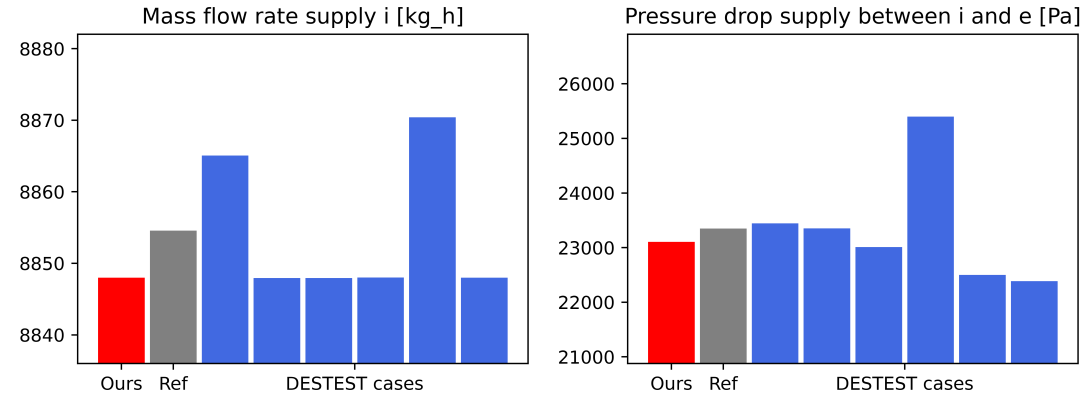
[2] *Heat loss.*

The test reported an accuracy grade on the reference values of 87.62%, in line with the highest

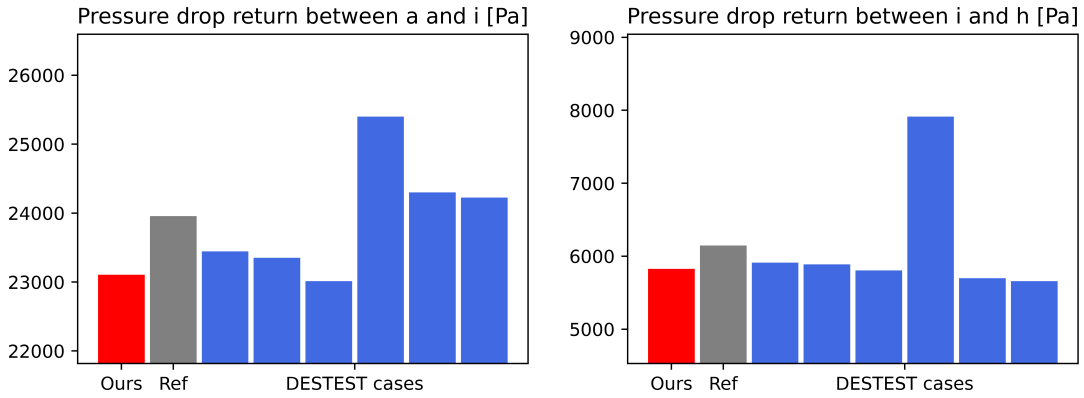



(a) (b)

Figure 2.6: Results of the DESTEST CE0. Total load and heat loss.




(a) (b)




(c) (d)

Figure 2.7: Results of the DESTEST CE0. Mass flow and pressure.

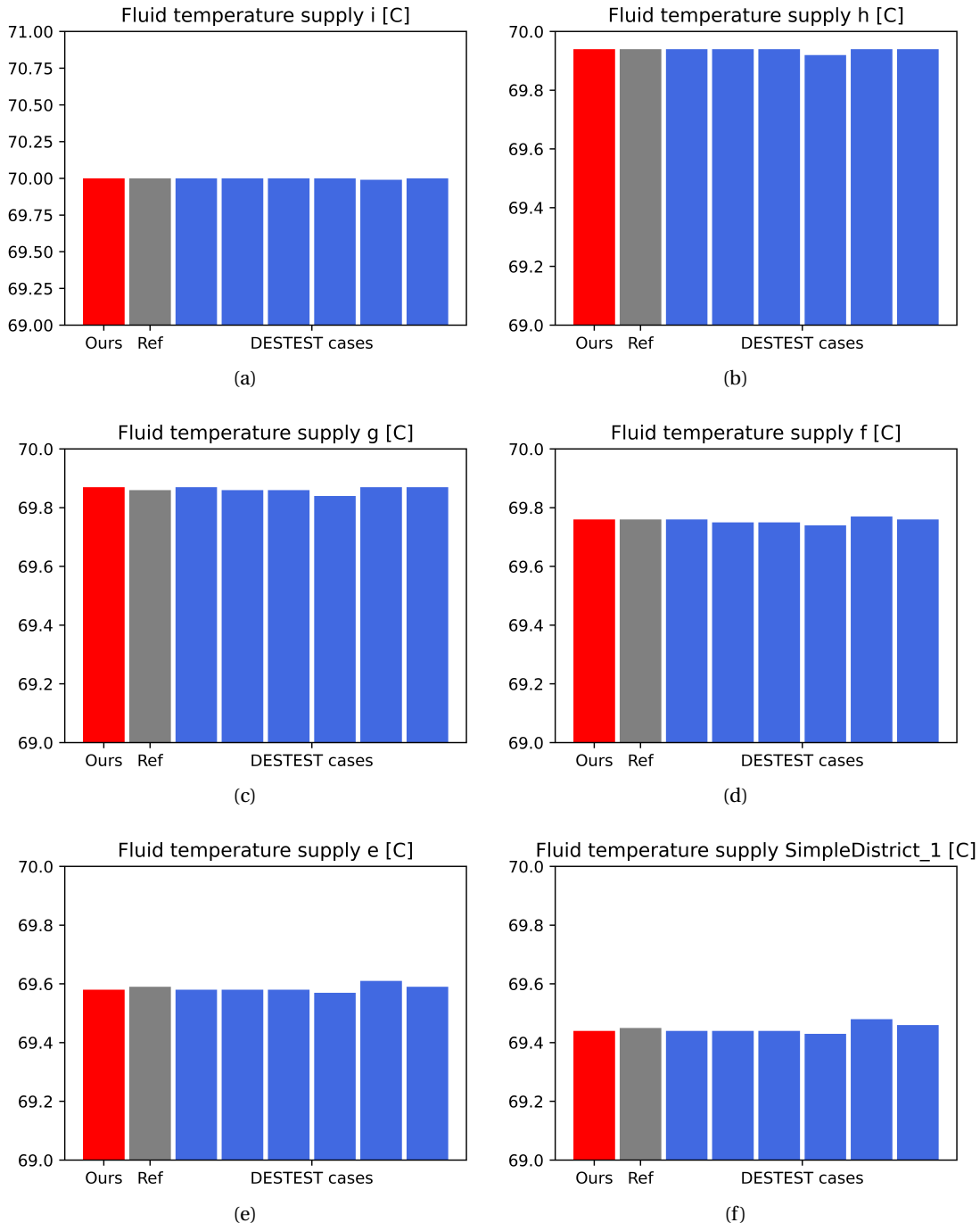



Figure 2.8: Results of the DESTEST CE0. Supply temperatures.

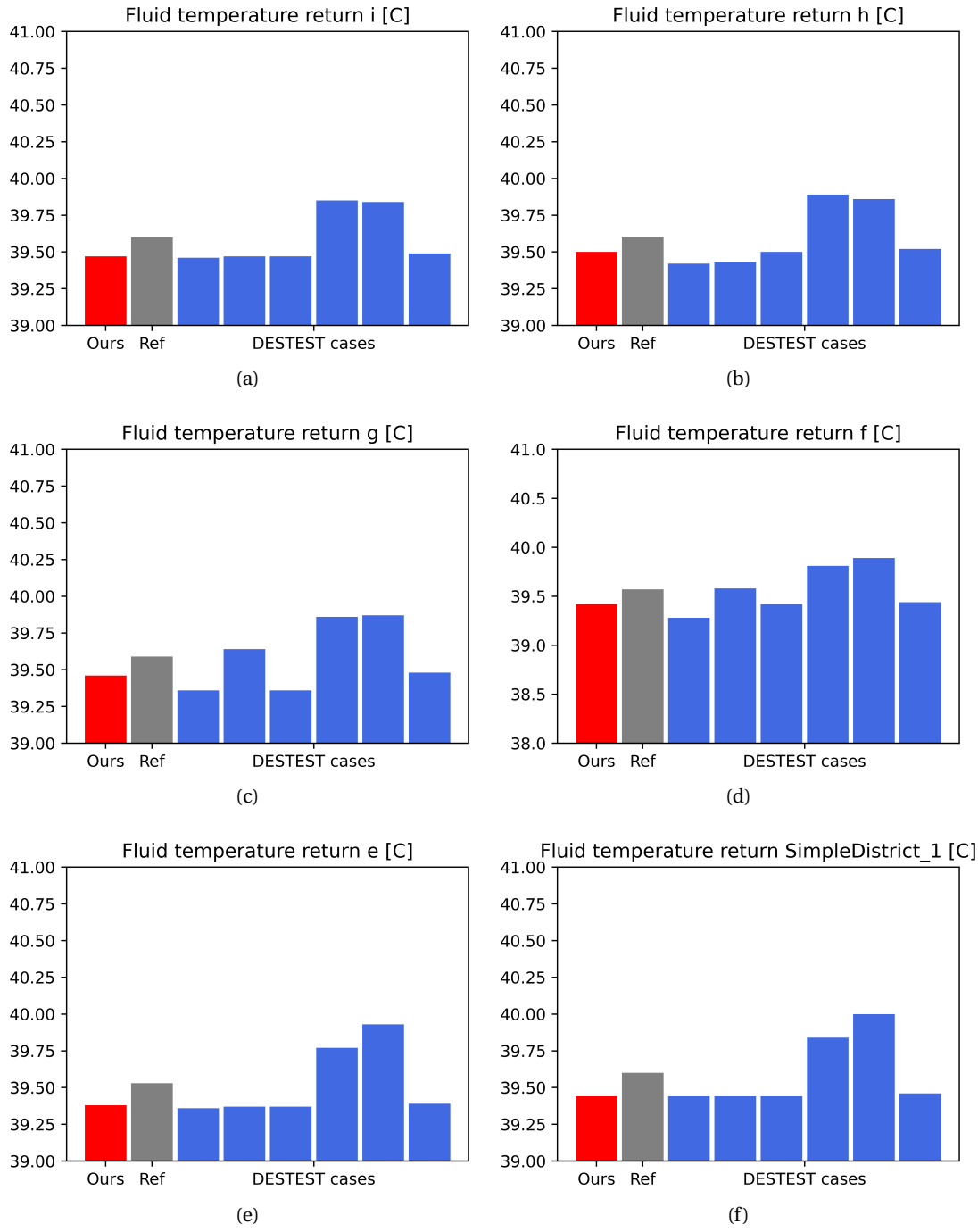



Figure 2.9: Results of the DESTEST CE0. Return temperatures.

result among the reference tools (85.17% for TRNSYS). All the considered KPIs are within the upper and lower bounds of the reference tools. The highest deviations from the reference value were found in KPIs related to the hydraulic part of the simulation. In particular, NMBEs of -1.04%, -3.55% and -5.18% were obtained for the pressure drops between nodes *i* and *e* in the supply network and between nodes *a* and *i* and *i* and *h* in the return network. These errors can be explained by the large difference in results among the reference tools, of which the reference value is an average. Between nodes *i* and *h* (see Figure 2.7d), for instance, the outputs of the reference tools ranged from 5658 Pa to 7913 Pa, with just two results above 6000 Pa. The reference value was then 6146 Pa, on which the reference tool with the closer output had a NMBE of 3.81%.

**Results CE1**

Table 2.3: Results of the proposed framework on DESTEST CE1. The value reported is the NMBE with respect to the reference work.

**Results for nodes**

| KPI | Node | Supply | Return |
|---|---|---|---|
| $\dot{m}$ | *1* | 0.68 | - |
| $T$ | *i* | - | 1.03 |
| $T$ | *1* | -7.62 | -4.31 |
| $\dot{Q}^{1}$ | *1* | -1.61 | - |

[1] *Delivered heat.*

**Results for edges**

| KPI | Edge | Supply | Return |
|---|---|---|---|
| $\Delta p$ | *i-h* | - | 14.37 |
| $\dot{Q}^{2,3}$ | *i-h* | - | -6.08 |
| $\dot{Q}^{2}$ | *all pipes* | 2.47 | |

[2] *Heat loss.*

[3] *Reference reports return instead of supply.*

The results of the proposed model for CE1 are presented in Table 2.3. The reference tool did not always achieve the required setpoints (Figure 2.5). For Substation 1, where several KPIs are evaluated, the peak loads were not met in the reference tool results, likely due to the use of a continuous controller. In contrast, our approach employed an ideal consumer model, which imposed the exact setpoints. As a result, the NMBE between the two tools was 0.68% for the mass flow and -1.61% for the actual delivered heat at the substation, leading to slightly different boundary conditions in the two experiments.

A significant difference in temperature at the substation nodes was found, with a NMBE of -7.62% and -4.31% on the inlet and outlet temperature respectively. The largest temperature error occurred during periods of zero load: while the reference model maintained a return temperature of approximately 40 °C, the temperature in our model dropped rapidly due to water stagnation (Figure 2.10). This likely reflects a difference in what each model reports: the reference data appear to use the secondary-side return temperature at the substation, whereas our model evaluates the temperature of the primary-side water directly in the connecting pipe (i.e., the flow exiting the bypass) as the substation is considered ideal and the secondary is not modeled, following the DESTEST prescriptions. The temperature in the supply node decreased in both tools, although somewhat faster in our model. This difference is partly attributable to the boundary-condition mismatch noted above; however, its magnitude suggests additional

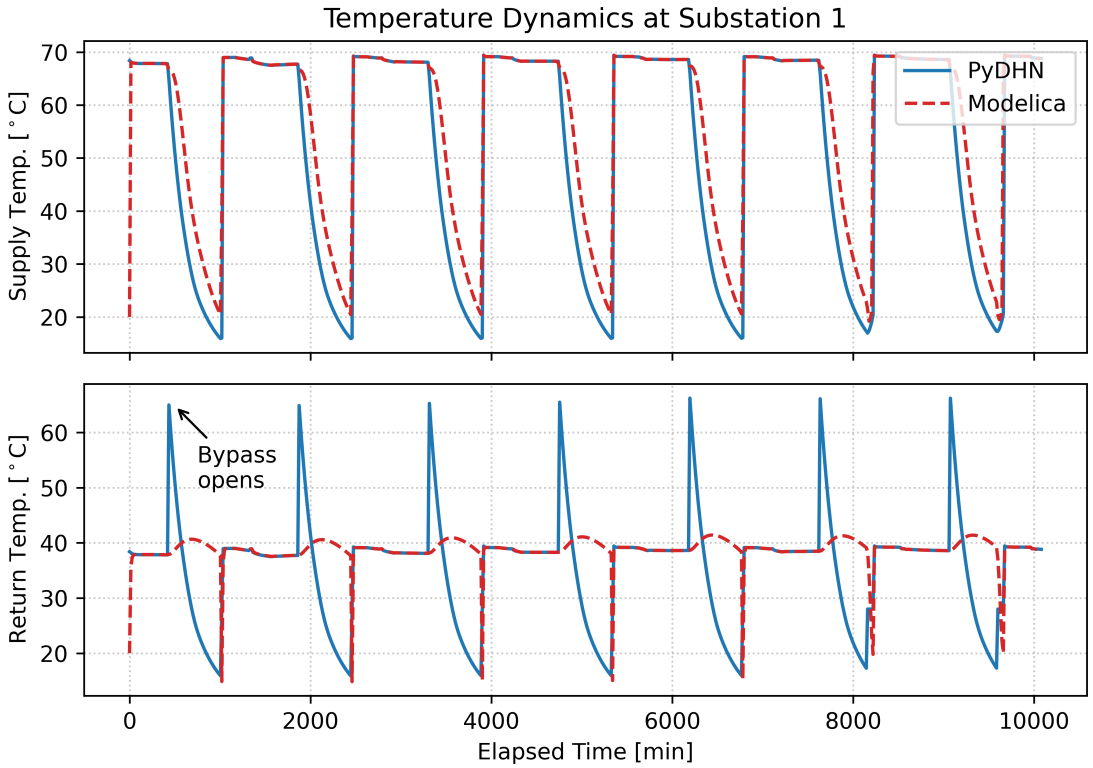



Figure 2.10: Temperature dynamics at Substation 1 (DESTEST CE1). During zero-demand periods, both models show cooling on the supply side (**top**), with PyDHN (solid blue) exhibiting a steeper decay than the reference (dashed red). On the return side (**bottom**), the Modelica model maintains a quasi-steady temperature, potentially computed inside the substation rather than at the bypass level, whereas PyDHN simulates the natural cooling of stagnant fluid. The periodic spikes in the PyDHN return profile reflect the activation of the bypass, transferring hot supply water directly to the return line.

contributing factors, such as the different pipe flow models (the two-node formulation used here is expected to behave more accurately at low Reynolds numbers because of the more prominent influence of boundary-layer effects (Dénarié et al., 2019)) and potentially other implementation differences, for example, pipe heat capacity parameters, which could not be verified because the reference simulation script is not publicly available. When periods with zero load are excluded, the NMBE reduces substantially for both inlet (-0.63 %) and outlet (0.56 %) temperatures.

Pressure loss values for this consumer were not provided in the reference dataset; however, the discrepancy can be assessed for the pipe connecting nodes $i$ and $h$ in the return network. Here, an NMBE of 14.37% was obtained, with our model simulating higher pressure losses than Modelica Buildings, despite the close agreement between the two tools in CE0. This deviation can be partly explained by the difference in mass flow during load peaks: since the proposed model yields a higher mass flow, correspondingly higher pressure losses are expected, as they

scale approximately with the square of the flow rate.

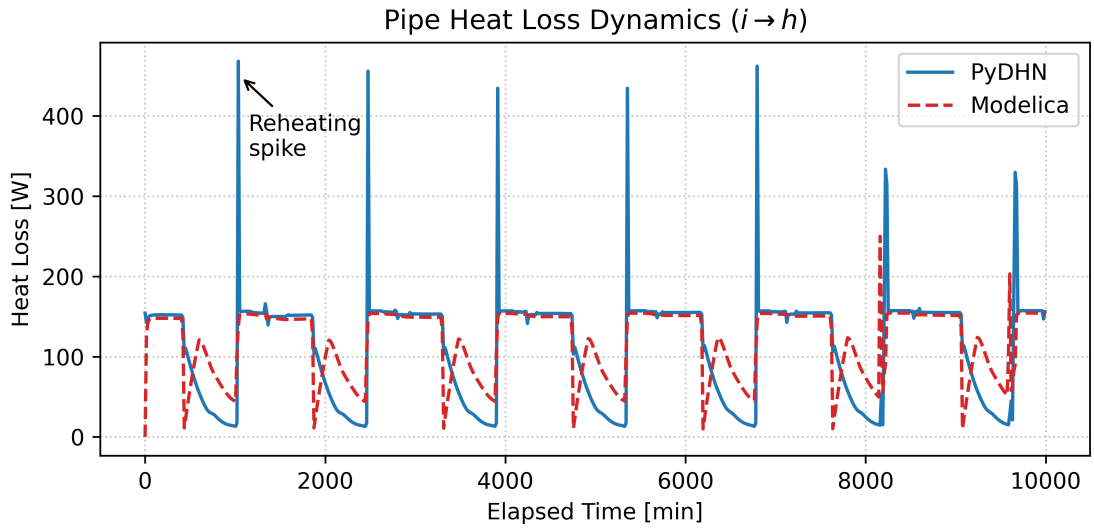



Figure 2.11: Heat loss dynamics for the return pipe segment between nodes $i$ and $h$. During almost no-flow intervals, the proposed tool (solid blue) simulates a rapid decay to near-zero heat loss, followed by a high-intensity spike representing the thermal load of reheating the cooled pipe wall. The reference tool (dashed red) instead exhibits an initial sudden drop and rebound, followed by a gradual decay that sustains a higher rate of heat loss.

Heat losses in the same reference pipe were also evaluated among the physical indicators (NMBE of -6.08%): in this case, PyDHN shows a smoother decay in heat loss during stagnation periods followed by sharp peaks when demand resumes, representing the energy load required to reheat the cooled pipe wall (Figure Figure 2.11). The reference tool, conversely, exhibits attenuated spikes, consistent with a higher thermal inertia (e.g., steel) retaining heat across the short bypass intervals. The heat losses in the network follow similar profiles but had a lower discrepancy of 2.47%. Overall, the average absolute NMBE of the tested tool against Modelica Buildings is 4.77% (3.09% if the no-demand periods are excluded).

## 2.6 Quantifying the Computational Bottleneck

Despite its flexibility and accuracy, the proposed framework has two structural bottlenecks that limit its applicability in pre-design and operational optimization tasks. These bottlenecks motivate the accelerated models developed in the following chapters.

### 2.6.1 The Hydraulic Iteration Bottleneck

Hydraulic calculations are steady-state: for fixed boundary conditions, each time step requires solving a nonlinear algebraic system. When boundary conditions do not depend on previous hydraulic or thermal states, hydraulic steps at different times are independent and can, in

principle, be evaluated in parallel.

This independence is lost in some optimization tasks. For example, if substation heat demand is computed dynamically based on the effective heat delivered at the previous step, the hydraulic state becomes sequential, even though the underlying equations are steady-state.

Regardless of whether such sequential feedback is present, each hydraulic step is intrinsically expensive. Solving the nonlinear system typically requires several Newton–Raphson iterations, each involving the evaluation of component outlet temperatures, Jacobian assembly, and linear system solves. This constitutes a *horizontal* bottleneck: each step is computationally heavy, even when steps can be parallelized.

Crucially, sequential dependencies at the time-step level do not prevent parallelism at the scenario level. Pre-design and planning studies often require thousands of independent trajectories; even if each trajectory must advance step-by-step, many scenarios can be computed concurrently. A surrogate or accelerated physics-based model that enables high-throughput scenario parallelism therefore offers significant advantages.

The hydraulic surrogate models introduced in the following chapters address this horizontal bottleneck directly, replacing the iterative compuations while supporting both trajectory-level and scenario-level parallelism.

### 2.6.2 The Thermal Sequentiality Bottleneck

The thermal model shares the same horizontal bottleneck as the hydraulic simulation whenever component functions $\psi$ are nonlinear, as is the case for the Lagrangian pipe model in its original formulation.

More importantly, the thermal simulation has an additional *vertical* bottleneck: the process is intrinsically autoregressive. The temperature distribution at a time step depends on the water volumes transported from the previous step and on mixing outcomes at all internal nodes. Thus, thermal steps within a trajectory cannot be easily parallelized.

The horizontal cost of the thermal loop can be reduced. Although the framework presented in this chapter allows general nonlinear $\psi$ functions to maximize flexibility, many components used in practice exhibit linear thermal behavior.

If the Lagrangian pipe model is reformulated so that its $\psi$ function is linear, the entire nodal energy balance system becomes linear as well. With the components used in this work, the thermal loop is already computationally light; once linearized, it reduces to a sequence of sparse matrix operations. In this case, surrogate models are unnecessary for accelerating the horizontal dependency, as a physics-based simulation can achieve similar computational speed.

A second limitation of classical Lagrangian formulations is their lack of differentiability (Maurer

et al., 2021). While most components and the thermal loop are differentiable, the pipe update is not, which prevents the full thermal model from being used in gradient-based optimization.

In later chapters, we introduce a reformulated Lagrangian pipe model that is both differentiable and linear, eliminating these obstacles without sacrificing fidelity. Accordingly, the entire thermal simulation is reformulated as a GPU-accelerated, differentiable model that also supports scenario-level parallelism.

## 2.7 Chapter Conclusion

This chapter presented the foundational simulation framework proposed for the detailed design and analysis of meshed DHNs. We introduced a novel, open-source Python library that integrates a fast hydraulic loop method with a high-fidelity Lagrangian thermal model. The modular architecture allows for the seamless integration of custom components, filling a critical gap in the current landscape of flexible, open-source simulation tools.

The reliability of the framework was verified through a two-stage process. At the component level, the thermal pipe model demonstrated superior accuracy against experimental aerial pipe data, achieving an RMSE below 0.6 K in highly dynamic conditions and outperforming several established commercial tools. At the network level, the framework showed strong agreement with reference solutions in the IBPSA DESTEST benchmark, validating its capability to handle both steady-state and transient scenarios in radial topologies.

Crucially, this physics-based model serves as the *ground truth* engine for the remainder of this thesis. It allows the generation of the high-quality, consistent synthetic datasets necessary to train the data-driven models proposed in subsequent chapters.

However, the performance analysis presented in Section 2.6 highlighted distinct computational barriers. While accurate, the reliance on iterative solvers for non-linear hydraulic systems creates a horizontal bottleneck that hinders applications requiring large-scale scenario evaluations, such as real-time optimization or uncertainty quantification. This limitation directly motivates the work of the following chapter, which proposes a Geometric Deep Learning approach to surrogate the hydraulic solver, aiming to retain physical consistency while drastically reducing computational cost.

# 3 Graph Neural Network Surrogates for Hydraulic Simulation

*The content of this chapter is primarily drawn from the following publications.*

*Boghetti, R., Odobez, J. M., & Kaempf, J. H. (2025). Graph Neural Network-Based Surrogate Modeling for Fast and Scalable Simulations of Meshed District Heating Networks.* Preprint Available at SSRN 5576535.

## 3.1 Introduction

As highlighted in Chapter 2, high-fidelity physics-based models provide the necessary flexibility for the detailed design of DHNs. However, the performance analysis in Section 2.6 identified the computational cost of the hydraulic simulation as an unsolved limitation for applications where simulation speed is critical. In meshed topologies, the interdependence of pressure and mass flow requires the solution of a system of non-linear equations via iterative numerical methods (e.g., Newton-Raphson). While robust, these iterative solvers constitute a significant **computational bottleneck**, rendering the model too slow for time-critical applications such as real-time control, or for tasks requiring large-scale scenario parallelism, such as uncertainty quantification and pre-design optimization.

To overcome this latency, the engineering domain has increasingly turned to ML to create data-driven surrogate models. By approximating the input-output relationship of the physical solver, these surrogates can infer system states orders of magnitude faster than numerical methods. However, existing ML approaches for DHNs typically suffer from **limited generalizability and scalability**. Standard architectures (e.g., Multi-Layer Perceptrons, MLPs) rely on fixed-size inputs, lack permutation invariance, and fail to explicitly encode the underlying graph structure. These limitations inherently tie the model to a specific network topology. Consequently, applying these models to a new DHN or altering an existing structure necessitates architectural reconfiguration and full retraining. Furthermore, while some architectures are theoretically transferable to new topologies, they generally fail to reproduce the hydraulic

dynamics of meshed networks, where the underlying physics is driven by global topology rather than just local connectivity.

This chapter takes a first step towards addressing these challenges by proposing a novel **hydraulic surrogate modeling framework** based on Geometric Deep Learning. We employ an inherently topology-agnostic GNN architecture that exploits the cycle-based formulation to embed physical inductive biases directly into the model, thereby ensuring a global view of the hydraulic problem. While this chapter validates the architecture's physical accuracy on a reference network, its structural flexibility lays the groundwork for cross-topology generalization explored in Chapter 5.

## 3.2 State of the Art

Among early works, Guelpa and Verda (2019) proposed a simplified model for the hydraulic simulation of a meshed DHN. They removed one pipe from each internal loop of the network, effectively decomposing its topology into a tree and a co-tree structure, which are then solved separately. The co-tree is solved first: the mass flows in the removed pipes are estimated using a linear model based on the processed mass flows at the heat plants. These estimated flows are then used as additional boundary conditions to solve the tree part. While achieving three orders of magnitude speed-up, this approach is inherently network-specific and cannot generalize, as the linear model learns relationships valid only for that specific topology and data distribution. If the network layout is altered, or a different set of pipes is removed for the decomposition, the entire model must be rebuilt.

Rodrigue et al. (2024) proposed to reduce the DHN topology by replacing predefined clusters of substations with single nodes, and estimate the temperatures of connecting pipe water using Artificial Neural Networks, while modeling remaining dynamics physics-based. Applied to a synthetic network, the proposed hybrid approach reduced simulation time by 27%, with negligible power estimation errors at heat sources. However, performance gains are not on the same magnitude as other methods due to partial physics-based reliance, and pipe-level outputs in clusters are not computed.

Boussaid et al. (2023) investigated the use of a surrogate model based on GNNs (Scarselli et al., 2009) with graph attention (Veličković et al., 2018) for predicting the return temperature of nodes in a toy network with 1 heat plant and 6 consumers. The model achieved a Root Mean Square Error (RMSE) of 0.3 K on the test data while reducing the computational time by a factor of $10^4$. A followup work Boussaid et al., 2024 extended the model to predict the nodal temperature and pressure for both the supply and return nodes, improving the architecture with recurrent layers (Gated Recurrent Units, (Cho et al., 2014)), graph convolutional layers (Kipf & Welling, 2017), and skip connections. The new architecture achieved a RMSE below 0.5 K on the temperature prediction and an average RMSE of 0.002 and 0.06 bar (200 and 6000 Pa) on the pressure estimation of supply and return nodes respectively. Scaling the model to larger networks showed increased prediction error with node count growth. Increasing

the complexity of the model partly mitigated the loss of performance, at the cost of inference speed. For a 100-node DHN simulated over 54 days with 20-minute time steps, the inference time was approximately 42 seconds, or around 92 steps per second. A notable limitation is that the GNN architecture provides only localized network views, inadequate for simulating large meshed networks where hydraulic behavior depends on entire loop topology, as evidenced by the relatively high errors reported for pressure estimation. While the size of the local view could be expanded by adding more GNN layers, this would come at the cost of slower inference and a potential loss of detail in the learned representations. Finally, de Giuli et al. (2024) proposed PI-RNN, an architecture interconnecting distinct Recurrent Neural Networks (RNNs) to form a graph resembling the DHN topology, requiring separate RNNs for each consumer (load) plus one for the return line. While promising on the nine-node synthetic example, this approach is highly topology-coupled, making scaling or adaptation difficult without re-designing the entire model.

## 3.3 Contribution

The literature review reveals a gap in surrogate models for hydraulic simulation of meshed DHNs, where pressure and flow distribution is influenced by the network as a whole. Existing approaches suffer from three main limitations: they are often topology-dependent and must be redesigned when the layout changes; some provide only partial predictions, omitting pipe-level or element-wise outputs; and most scale poorly to large networks. GNN-based surrogate models offer a promising alternative, as they naturally adapt to topology and can provide full-network predictions, but their limited receptive field prevents capturing the long-range dependencies of hydraulic physics.

This chapter addresses the above limitation by introducing a novel surrogate model for estimating mass flows in meshed DHNs that has a global view of the network. The model reduces reliance on iterative corrections and achieves a throughput of thousands of time steps per second.

Although the proposed GNN architecture is designed to be topology-agnostic, a feature that theoretically allows for transferability across different networks (e.g., via generalization or few-shot adaptation), this chapter focuses specifically on its application to operational optimization. In this context, the network topology remains fixed, and the focus is on prediction fidelity and speed. Therefore, we train and evaluate the model on a specific target network to achieve the minimal error margins necessary for control tasks. However, this baseline architecture serves as a starting point for the surrogate model proposed in Chapter 5. There, the same architectural principles will be adapted to the pre-design phase, where the priority shifts towards generalizing to unseen topologies, albeit with a trade-off in accuracy.

We first test the surrogate hydraulic model on unseen synthetic data. Then, in Chapter 4, we couple it with a GPU-optimized, Lagrangian thermal model and evaluate the combined system on real-world monitoring data, comparing its performance to that of PyDHN.

While this work focuses on DHNs, the approach can be readily adapted to cooling networks.

The specific contributions of this chapter are:

- The formulation of a novel **condensed graph representation** that explicitly models cycle-pipe interactions, allowing the GNN to capture global hydraulic dependencies in meshed networks.
- The design of a **two-stage GNN architecture** that employs a coarse-to-fine refinement strategy allowing robust prediction accuracy while retaining a limited depth.
- A **physics-informed training strategy** utilizing a composite loss function with dynamic weighting to enforce Kirchhoff's second law and ensure physical consistency.
- A resulting throughput of **2187 steps/s** with a **97.34% convergence rate**, reducing computation time by nearly two orders of magnitude compared to physics-based solvers without compromising fidelity.

## 3.4 Methodology

This section details the methodology used to solve the hydraulic simulation task. We begin by defining the problem setup and constraints in Section 3.4.1. Following this, we present the high-level workflow of the proposed surrogate model in Section 3.4.2, serving as a roadmap for the subsequent sections which detail the model's specific architectural components and training procedures (Section 3.4.3 to Section 3.4.7).

### 3.4.1 Problem Setup

The hydraulic state of a DHN is only partially observable, with sensors typically at producers and consumers, but not within the pipe network. The simulation task is thus to infer the complete network state from this limited set of measurements. To do so, we assume that we are given two categories of information:

- **Time-Invariant Structure:** The physical layout of the network is fixed. This includes the pipe properties (length, diameter, roughness) and the graph topology defined by the incidence matrices.
- **Time-Variant Boundary Conditions:** observations at producers and consumers define the hydraulic problem at each time step, fully constraining the system. These are: (i) mass flow rates at consumers and secondary producers, (ii) pressure difference across the main producer.

Mathematically, these inputs are mapped to the network state by solving the system of nonlinear equations derived in Section 2.4.1. By decomposing the cycle basis into *boundary cycles*

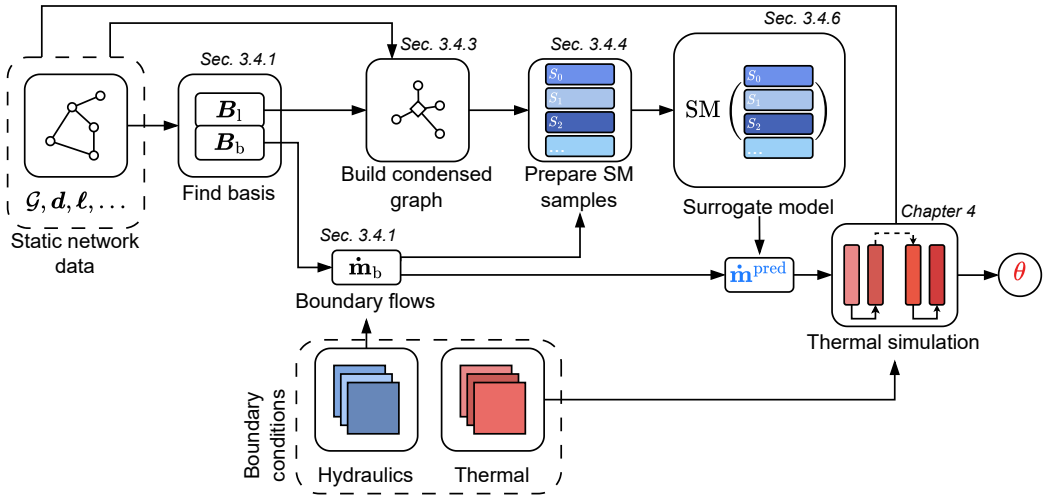



Figure 3.1: Overview of the simulation pipeline and relevant sections. The network topology is used to extract a cycle basis and define the structure and edge features of the condensed graph, while static pipe properties provide its node attributes. These, together with the boundary flows computed at each time step from hydraulic boundary conditions, form the input to the Surrogate Model (SM in the figure). The model predicts internal cycle flows, used to reconstruct full pipe mass flows $\dot{\boldsymbol{m}}^\mathrm{pred}$, which are then passed, along with thermal boundary conditions, to a physics-based thermal model.

and *internal cycles*, we can partition the problem variables. Let $\mathscr{C}_\mathrm{b}$ and $\mathscr{C}_\mathrm{l}$ denote the sets of boundary and internal cycles, respectively. We partition the cycle-edge incidence matrix $\boldsymbol{B}$ and the cycle mass flow vector $\tilde{\boldsymbol{m}}$ as follows:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{B}_\mathrm{l} \\ \boldsymbol{B}_\mathrm{b} \end{bmatrix}, \quad \tilde{\boldsymbol{m}} = \begin{bmatrix} \tilde{\boldsymbol{m}}_\mathrm{l} \\ \tilde{\boldsymbol{m}}_\mathrm{b} \end{bmatrix} \tag{3.1}$$

where $\boldsymbol{B}_i \in \{-1, 0, 1\}^{|\mathscr{C}_i| \times |\mathscr{E}|}$ and $\tilde{\boldsymbol{m}}_i \in \mathbb{R}^{|\mathscr{C}_i|}$. The vector $\tilde{\boldsymbol{m}}_\mathrm{b}$ is fully determined by the boundary conditions. The task is therefore to find the unknown internal cycle flows $\tilde{\boldsymbol{m}}_\mathrm{l}$ that satisfy the pressure balance around the internal loops. Substituting the partitioned variables into Equation (2.4) yields the governing equation for the internal state:

$$\boldsymbol{B}_\mathrm{l} \boldsymbol{\phi}\left(\boldsymbol{B}_\mathrm{b}^\top \tilde{\boldsymbol{m}}_\mathrm{b} + \boldsymbol{B}_\mathrm{l}^\top \tilde{\boldsymbol{m}}_\mathrm{l}\right) = \boldsymbol{0} \tag{3.2}$$

Solving Equation (3.2) for $\tilde{\boldsymbol{m}}_\mathrm{l}$ allows for the reconstruction of the mass flow in every edge of the network via superposition.

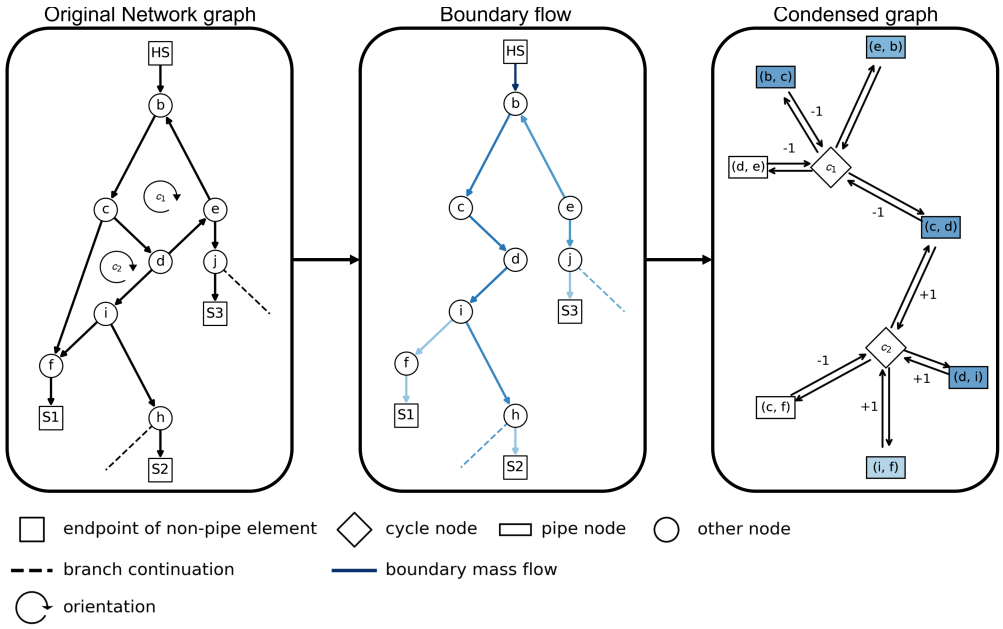



Figure 3.2: Construction of the condensed graph. **Left**: Original network graph showing a producer (HS), substations (S), and junctions (lowercase letters). Center: Boundary mass flows are computed based on boundary conditions, with color intensity qualitatively representing flow magnitude (white indicating zero flow). **Right**: In the condensed graph, each cycle is a virtual node connected to its constituent pipe nodes, which retain boundary flow magnitude as a feature. Edges are assigned an orientation attribute.

### 3.4.2 Surrogate Modeling

This section details the GNN-based surrogate model developed to solve the hydraulic problem introduced in Chapter 2 without costly Newton–Raphson iterations. The overall workflow is summarized in Figure 3.1. In what follows, we assume that the cycle basis for the network graph is constructed according to the procedure given in Section 2.4.1. For a given set of boundary conditions, the process is as follows:

1. Derive the boundary cycle flows $\tilde{\boldsymbol{m}}_\mathrm{b}$ from the known mass flows at substations and secondary plants, and compute the corresponding boundary edge flows: $\dot{\boldsymbol{m}}_\mathrm{b} = \boldsymbol{B}_\mathrm{b}^\top \tilde{\boldsymbol{m}}_\mathrm{b}$ (Figure 3.2, center).

2. Construct a condensed graph $\mathcal{G}'$, whose structure is described by the vertex-edge incidence matrix $\boldsymbol{A}'$ along with vectors of edge features $\boldsymbol{e}$, and node features $\boldsymbol{\ell}'$ and $\boldsymbol{d}'$. In line with Chapter 2, we assume a fixed, identical roughness for all pipes. In $\mathcal{G}'$, each of the $N_\mathrm{V}$ nodes corresponds either to a pipe or to an internal cycle, making the cycle–pipe interactions within internal loops explicit (Figure 3.2, right).

3. Prepare training/inference samples $S = (\boldsymbol{A}', \boldsymbol{e}, \boldsymbol{X})$, with $\boldsymbol{X} = \left[\boldsymbol{x}_u^\top\right]_{u=1}^{N_\mathrm{V}} \in \mathbb{R}^{N_\mathrm{V}\times 3}$, where each node feature vector is defined as $\boldsymbol{x}_u = [\dot{m}'_{b,u}, d'_u, \ell'_u]^\top$. The vector $\dot{\boldsymbol{m}}'_\mathrm{b}$ is adapted from $\dot{\boldsymbol{m}}_\mathrm{b}$

to match the cycle-based representation of the condensed graph $\mathcal{G}'$.

4. For training and evaluation, compute the ground-truth internal cycle flows $\tilde{\boldsymbol{m}}_\mathrm{l}$ through the fully physics-based simulation described in Section 2.4.1. These are not available during deployment.

5. Use the surrogate model to estimate the internal cycle flows $\tilde{\boldsymbol{m}}_\mathrm{l}^\mathrm{pred}$ from a tuple $S$.

6. Reconstruct the edge mass flow vector for the whole network: $\dot{\boldsymbol{m}}^\mathrm{pred} = \dot{\boldsymbol{m}}_\mathrm{b} + \boldsymbol{B}_\mathrm{l}^\top \tilde{\boldsymbol{m}}_\mathrm{l}^\mathrm{pred}$.

The remainder of this section details the components of this workflow. First, we define the data representation, covering the condensed graph construction (Section 3.4.3) and the input–output formulation (Section 3.4.4). Second, we describe the learning framework, including the background on GNNs (Section 3.4.5), the specific network architecture (Section 3.4.6), and the physics-informed loss function (Section 3.4.7).

### 3.4.3 Condensed Graph Representation

To emphasize the functional relationships relevant to the hydraulic problem, we adopt a condensed representation based on the cycle formulation introduced in Section 2.4.1. Unlike the physical network graph (Figure 2.1, left), this representation excludes elements not involved in predicting internal cycle flows: producers, consumers, external pipes, and direct pipe–pipe connections. Instead, each internal cycle is represented by a virtual node connected to its associated pipes. Beyond reducing data complexity and simplifying learning, this formulation also alleviates limitations of GNNs related to receptive field size and oversmoothing, as discussed in Section 3.4.5.

The condensed representation follows two main principles:

1. The solution to the hydraulic problem can be formalized as the superposition of (i) the boundary edge flows $\dot{\boldsymbol{m}}_\mathrm{b}$, known, and (ii) the internal loop flows $\dot{\boldsymbol{m}}_\mathrm{l}$, unknown.

2. Any valid mass flow vector, $\dot{\boldsymbol{m}}$, can be expressed as a linear combination of cycle flows (see Equation (2.3)).

Following these principles, the condensed graph $\mathcal{G}'$ is derived from the internal loop cycle matrix $\boldsymbol{B}_\mathrm{l}$ and the static properties $\ell$ and $\boldsymbol{d}$, and represented by the tuple $(\boldsymbol{A}', \boldsymbol{e}, \ell', \boldsymbol{d}')$. Its construction proceeds as follows:

1. Define $\mathcal{G}'$ structure with one vertex per internal loop pipe and one per internal cycle, connected by edges drawn from pipe vertices to the cycle vertices that include them. This yields a bipartite graph, encoded by the adjacency matrix $\boldsymbol{A}'$, with $N_\mathrm{p}$ pipe and $N_\mathrm{c}$ cycle vertices, for a total of $N_\mathrm{V} = N_\mathrm{p} + N_\mathrm{c}$ vertices.

2. Define the vectors of static vertex properties $\boldsymbol{\ell}'$ and $\boldsymbol{d}'$, using the original value for pipe vertices while assigning zero values to cycle vertices.

3. Construct the edge feature vector $\boldsymbol{e}$ by stacking scalar values $e_{uv} \in \{-1, 1\}$ that encode the orientation of each pipe within its associated cycles, as defined by the corresponding entries of the matrix $\boldsymbol{B}_\mathrm{l}$.

### 3.4.4 Input–output Structure of the Surrogate Model

Given the condensed graph, its static properties, and a set of boundary conditions, we generate training samples consisting of model inputs and corresponding ground-truth targets. Note that targets are only available during training and evaluation; they are not accessible at inference time.

First, we construct the boundary edge flow vector for the condensed graph $\dot{\boldsymbol{m}}'_\mathrm{b}$, by assigning values from $\dot{\boldsymbol{m}}_\mathrm{b}$ to the $N_\mathrm{p}$ pipe vertices, and zero to the $N_\mathrm{c}$ cycle vertices. The node feature matrix $\boldsymbol{X}$ is then constructed by stacking a feature vector $\boldsymbol{x}_u = [\dot{m}'_{\mathrm{b},u}, d'_u, \ell'_u]^\top$ for each node $u$. The learning target is then the vector of internal cycle flows $\tilde{\boldsymbol{m}}_\mathrm{l} \in \mathbb{R}^{N_\mathrm{c}}$, defined over the cycle nodes. Each training sample thus consists of the tuple $S = (\boldsymbol{A}', \boldsymbol{X}, \boldsymbol{e}, \tilde{\boldsymbol{m}}_\mathrm{l})$, where $\boldsymbol{e}$ encodes edge orientations as described in the previous section. Note however that each training sample only differs in the values of $\dot{m}'_{\mathrm{b},u}$ in the input, and $\tilde{\boldsymbol{m}}_\mathrm{l}$ at the output, all other variables being defined according to the DHN and graph structure.

Because the supply and return lines do not share any pipe, $\mathcal{G}'$ is naturally disconnected, forming two independent subgraphs. While physically identical, we compute their cycle bases independently, which makes their corresponding portions of the global adjacency matrix ($\boldsymbol{A}'$) and edge-feature vector ($\boldsymbol{e}$) different. We treat these structurally distinct subgraphs as separate samples to augment our training data and to verify the approach's robustness to variations in the graph construction algorithm. For notational simplicity, however, we will refer to these inputs with the unified symbols $\boldsymbol{A}'$ and $\boldsymbol{e}$ in the following sections.

### 3.4.5 Graph Neural Networks

GNNs are deep learning models for graph-structured data. They are broadly divided into *spectral* methods, which rely on the graph Laplacian and operate in a frequency-like domain, and *spatial* methods, which define operations directly on nodes and their neighborhoods. In particular, the latter use a *message-passing* mechanism, where nodes aggregate information from their neighbors to update their own representations.

In this work we propose to use spatial GNNs for our task. The main motivation is that in our condensed graph, each cycle vertex has to aggregate information from its connected pipes, which can naturally and conceptually be captured by message-passing operations. In addition, spatial methods are scalable, flexible, and well suited to *multi-graph settings*, where

the model is trained across different network topologies, allowing future investigations into model transferability and generalization.

The inference process in spatial GNNs is done by iterating message-passing operations acting on some hidden representation, which is achieved by stacking several message-passing layers in the neural network. Such iterations allow the model to capture broader dependencies between nodes and identify more expressive patterns. More formally, each iteration or layer is working as follows. We denote by $\boldsymbol{z}_u^t$ the hidden representation associated with node $u$ at layer $t$, where $\boldsymbol{z}_u^0$ denotes the initial embedding commonly obtained by applying a learnable transformation (e.g., a linear layer) to the raw input features $\boldsymbol{x}_u$. The general message-passing framework Gilmer et al. (2017) defines the update from layer $t$ to $t+1$ as:

$$\boldsymbol{z}_u^{t+1} = U_{\boldsymbol{\Theta}}\left(\boldsymbol{z}_u^t, \bigoplus_{v \in \mathcal{N}(u)} M_{\boldsymbol{\Theta}}\left(\boldsymbol{z}_u^t, \boldsymbol{z}_v^t, \boldsymbol{e}_{vu}\right)\right) \tag{3.3}$$

Here, $\mathcal{N}(u)$ is the set of neighbors of node $u$. The message function $M_{\boldsymbol{\Theta}}$ computes messages passed from neighboring nodes $v$ to node $u$, optionally incorporating edge features $\boldsymbol{e}_{vu}$. The aggregation operator $\bigoplus$ combines these messages, usually with a permutation-invariant function (e.g., summation), and the update function $U_{\boldsymbol{\Theta}}$ integrates the result with the node's previous representation to produce $\boldsymbol{z}_u^{t+1}$. Message-passing GNNs have a limited receptive field: each message-passing layer allows a node to incorporate information from its immediate neighbors; stacking $t$ layers extends the receptive field to nodes up to $t$ hops away. While increasing the receptive field is desirable in many applications, stacking multiple message-passing layers quickly leads to *oversmoothing*, a phenomenon in which node representations become increasingly similar across the graph, degrading performance in tasks requiring fine-grained distinctions (Rusch et al., 2023).

To mitigate oversmoothing, we adopt the approach proposed by G. Li et al. (2020), which combines improved message aggregation with architectural features designed for deep GNNs. Specifically, we use the DyResGEN block, based on the GENeralized Graph Convolution (GENConv), which defines the message-passing update as:

$$\boldsymbol{z}_u^{t+1} = \mathrm{MLP}\left(\boldsymbol{z}_u^t + \bigoplus_{v \in \mathcal{N}(u)} \left(\mathrm{ReLU}\left(\boldsymbol{z}_v^t + \boldsymbol{\Theta}_e \boldsymbol{e}_{vu}\right) + \epsilon\right)\right) \tag{3.4}$$

In the above equation, ReLU is the rectified linear activation function (Fukushima, 1969), $\boldsymbol{\Theta}_e \in \mathbb{R}^{|\boldsymbol{z}_v^t| \times |\boldsymbol{e}_{vu}|}$ is a learnable weight matrix used to project edge features into the same dimensionality as node embeddings, $\epsilon = 10^{-7}$ is a small positive constant for numerical stability, and the update function, MLP, is a multi-layer perceptron (see (G. Li et al., 2020) for architectural details). The aggregation operation $\bigoplus$ in GENConv is a generalized softmax-weighted summation with a learnable temperature parameter. The DyResGEN block further extends this GENConv formulation by adding residual connections and dynamic normalization. The full block consists of the following sequence: BatchNorm (Ioffe & Szegedy, 2015) → ReLU → GENConv → residual addition with the input (see Figure 3.3, left).

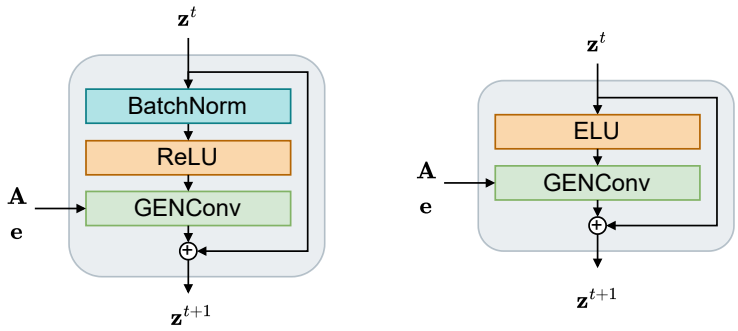



Figure 3.3: **Left:** Original DyResGen block. **Right:** Our implementation with ELU. The plus sign $\bigoplus$ indicates element-wise summation.

### 3.4.6 Surrogate Model Architecture

The surrogate model builds on the GNN framework introduced above and predicts internal cycle mass flows, $\tilde{\boldsymbol{m}}_{\mathrm{l}}$, using the condensed graph representation $\mathscr{G}'$ (Section 3.4.3). Inputs include the graph structure ($\boldsymbol{A}'$), edge features ($\boldsymbol{e}$), and a node feature matrix ($\boldsymbol{X}$). $\boldsymbol{X}$ combines static pipe properties ($\boldsymbol{d}'$, $\boldsymbol{\ell}'$) with dynamic boundary mass flows ($\dot{\boldsymbol{m}}'_{\mathrm{b}}$). Features for cycle nodes are initialized to zero. For stability and faster convergence, $\boldsymbol{X}$ is column-wise scaled to $\boldsymbol{X}_{\mathrm{scaled}}$ by the maximum absolute value over the training set. The graph structure $\boldsymbol{A}'$ and directions $\boldsymbol{e}$ are fixed for each line (e.g., supply), while only the boundary mass flows vary across samples. The architecture (Figure 3.4) consists of two encoder–decoder stages trained jointly end-to-end, where the second stage functions as a residual correction mechanism, refining the prediction of the first stage. In Stage 1, the encoder projects $\boldsymbol{X}_{\mathrm{scaled}}$ into a latent space via a linear layer and processes it with a Series of seven Message-Passing blocks (MPS). Each block is based on DyResGEN but modified by replacing the original BatchNorm–ReLU pre-activation with Exponential Linear Units (ELUs) (Figure 3.3, right), following evidence of improved performance (Clevert et al., 2016) confirmed in our preliminary tests. Message direction alternates between blocks: odd blocks propagate from pipes to cycles, even blocks from cycles to pipes, with separate parameters for each block. Since overlapping cycles exchange information only through common pipes, two layers are required for one cycle-to-cycle update; in our case study, five layers are needed for the most distant cycles, so we use seven to ensure full propagation without excessive depth or oversmoothing. The encoder produces context-aware node embeddings, which the decoder (a two-layer MLP with ELU activation) maps to an initial prediction of cycle flows, $\tilde{\boldsymbol{m}}_{\mathrm{l}}^{(1)}$.

This intermediate prediction is then processed by a Flow Aggregation Layer, which projects the cycle flows to pipe flows (analogous to Equation (2.3)) and sums them with the corresponding boundary flows. The resulting vector is then batch-normalized, zero-filled at cycle nodes, and concatenated with $\boldsymbol{X}_{\mathrm{scaled}}$. This new feature matrix forms the input to Stage 2, which shares the same structure as Stage 1 but with independent parameters. Its decoder outputs a correction term which is summed with the Stage 1 prediction to yield the final estimate $\tilde{\boldsymbol{m}}_{\mathrm{l}}^{\mathrm{pred}}$.

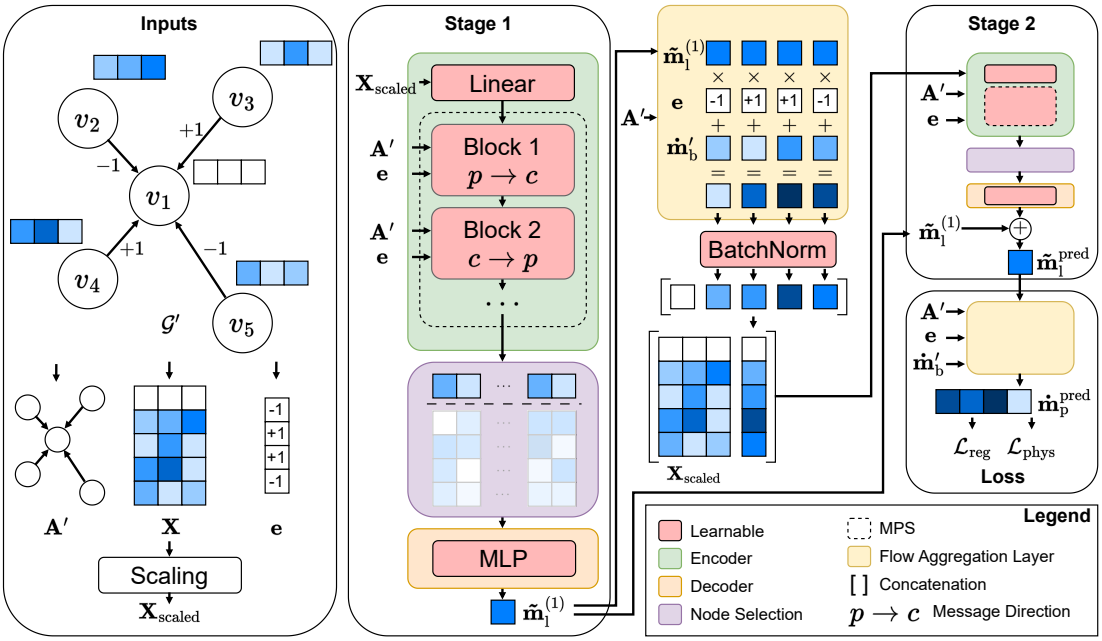



Figure 3.4: Overview of the proposed GNN architecture. Inputs are processed to create scaled node features ($\boldsymbol{X}_{\text{scaled}}$), an adjacency matrix ($\boldsymbol{A}'$), and edge features ($\boldsymbol{e}$). In Stage 1, an encoder-decoder model produces an initial cycle flow prediction ($\tilde{\boldsymbol{m}}_{\text{l}}^{(1)}$). This prediction is then processed by a Flow Aggregation Layer and combined with the original inputs to form a new feature matrix for Stage 2, which outputs a final, refined prediction ($\tilde{\boldsymbol{m}}_{\text{l}}^{\text{pred}}$).

### 3.4.7 Loss Function

The model is trained by minimizing a composite loss function that combines a supervised regression term ($\mathscr{L}_{\text{reg}}$) with a physics-informed penalty ($\mathscr{L}_{\text{phys}}$) in a weighted sum:

$$\mathscr{L} = w_1(n_e) \cdot \mathscr{L}_{\text{reg}} + w_2(n_e) \cdot \mathscr{L}_{\text{phys}} \tag{3.5}$$

where $n_e$ is the current training epoch, and $w_1(n_e)$ and $w_2(n_e)$ are dynamic weights that balance the two objectives. Both loss components operate on the predicted mass flows of the subset of pipe nodes in the condensed graph, denoted with the subscript $p$, $\dot{\boldsymbol{m}}_{\text{p}}^{\text{pred}}$. These are reconstructed from the model's cycle-flow output, $\tilde{\boldsymbol{m}}_{\text{l}}^{\text{pred}}$ as $\dot{\boldsymbol{m}}_{\text{p}}^{\text{pred}} = \boldsymbol{B}_{\text{p}}^{\top} \tilde{\boldsymbol{m}}_{\text{l}}^{\text{pred}}$. Here, $\boldsymbol{B}_{\text{p}} \in \mathbb{R}^{N_c \times N_p}$ is a sub-matrix of $\boldsymbol{B}_{\text{l}}$ containing only the columns corresponding to the $N_p$ internal loop pipes.

The regression loss is then defined as the mean squared error between predicted and ground-truth pipe-level mass flows:

$$\mathscr{L}_{\text{reg}} = \frac{1}{N_{\text{s}} N_{\text{p}}} \sum_{i=1}^{N_{\text{s}}} \sum_{j=1}^{N_{\text{p}}} \left( \dot{m}_{\text{p},ij}^{\text{pred}} - \dot{m}_{\text{p},ij} \right)^2 \tag{3.6}$$

Here, $N_{\text{s}}$ is the number of samples in the batch. The physics loss term reflects Kirchhoff's

second law (Equation (2.1)) and it is defined as a log-squared penalty on the cycle-level residuals:

$$\mathscr{L}_{\text{phys}} = \frac{1}{N_{\text{s}} N_{\text{c}}} \sum_{i=1}^{N_{\text{s}}} \sum_{j=1}^{N_{\text{c}}} \left( \ln\left(1 + \frac{|r_{ij}|}{k}\right)\right)^2 \tag{3.7}$$

Here, $r_{ij}$ denotes the pressure residual of internal cycle $j$ in sample $i$, computed from the predicted pipe flows. Specifically,

$$r_{ij} = \left[ \boldsymbol{B}_{\text{p}} \boldsymbol{\phi}(\dot{\boldsymbol{m}}_{\text{p},i}^{\text{pred}}, \boldsymbol{d}'_{\text{p}}, \boldsymbol{\ell}'_{\text{p}}) \right]_j . \tag{3.8}$$

The normalization constant $k = \delta/(e-1)$ is chosen such that residuals smaller than the threshold $\delta = 100$ Pa (matching the tolerance used in physics solvers such as (Robinson et al., 2009)) maps to below 1 before squaring. The goal is to introduce an inflection point in the loss function: residuals below $\delta$ contribute less aggressively, while larger violations remain strongly penalized. As pressure residuals can span several orders of magnitude, we use the natural logarithm to ensure training stability by dampening the effect of large squared errors.

To facilitate convergence, we give more importance to the regression loss in early training, and progressively shift toward enforcing physical consistency. This is achieved through a sigmoid-based schedule centered at the midpoint of training:

$$w_2(n_e) = \sigma\left(20 \cdot \frac{n_e - \frac{N_e}{2}}{N_e}\right), \quad w_1(n_e) = 1 - w_2(n_e) \tag{3.9}$$

where $N_e$ is the total number of epochs and $\sigma(x) = \frac{1}{1+\exp(-x)}$. This schedule ensures a smooth transition from data-driven learning to physics-guided refinement.

## 3.5 Data, Model Configuration, and Evaluation Protocol

This section details the experimental framework used to evaluate the proposed approach for internal loop flow prediction in a real-world DHN. We first describe the case study network topology (Section 3.5.1). We then explain the procedure used to generate synthetic training, validation, and testing datasets to have a broad coverage of operational conditions (Section 3.5.2). Subsequently, we outline the model configurations adopted for benchmarking and describe the training protocol and hyperparameters (Section 3.5.3). We then define the evaluation metrics: a core set used for model selection, and additional set to assess robustness and constraint satisfaction of the best performing model (Section 3.5.4). Finally, we provide relevant implementation details (Section 3.5.5).

### 3.5.1 Network Topology and Training Domain

To evaluate the approach on a realistic problem space, we utilize the topology of a real-world DHN located in Verbier, Switzerland. While a comprehensive description of the site, its specific operational challenges, and the analysis of monitoring data will be reported in Section 4.5.1 when discussing the performance of the proposed models against monitoring data, this section defines the structural properties and physical bounds necessary for generating the synthetic training dataset.

The selected network topology serves 165 substations and is powered by three heat plants. The primary plant (MAIN) covers the baseload, while two secondary plants (SEC1 and SEC2) provide additional capacity when demand exceeds the main plant's output. Structurally, both the supply and return lines feature a complex meshed topology characterized by six overlapping internal loops. Figure 3.5 illustrates the network layout, highlighting the central meshed section, where we solve the hydraulic coupling via the surrogate model.

Since no reliable data on the elevation of the network is available, we disregard any hydrostatic pressure differences assuming equal pipe heights.

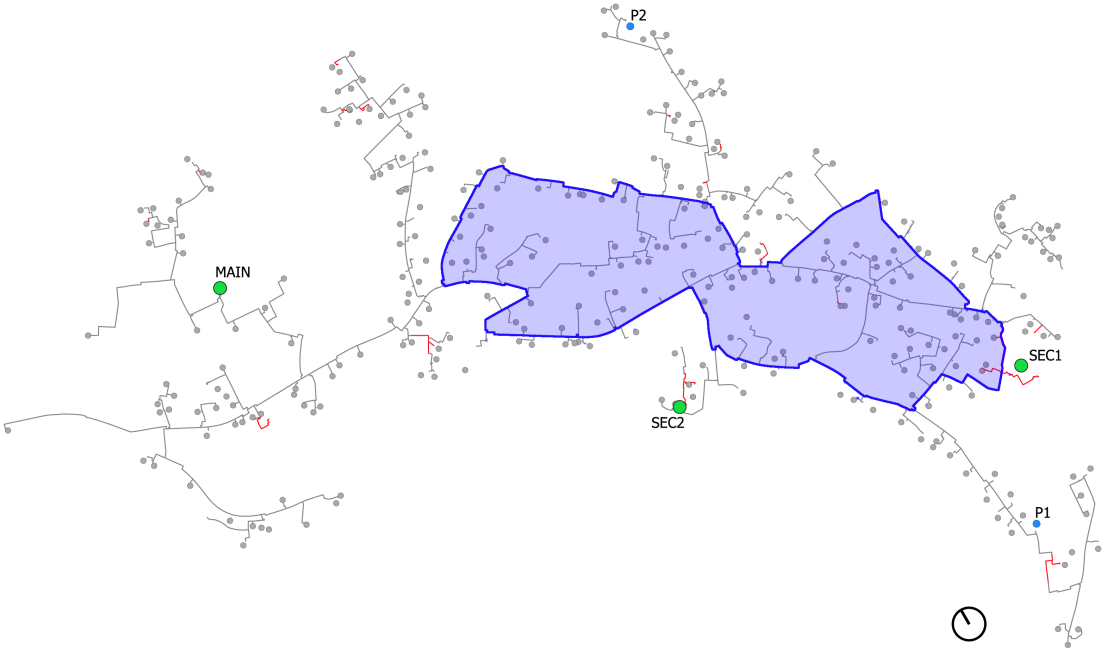



Figure 3.5: The DHN of Verbier. Heat plants are shown in green, while the substations with pressure measurements are in blue and aerial pipes are colored in red. The central section of the network containing the internal loops is highlighted in blue.

### 3.5.2 Experimental Data Generation

Our goal is to estimate internal loop flows under varying boundary conditions, specifically in the central meshed section of the network. Reliance solely on historical monitoring data is insufficient for training a robust surrogate, as such data typically covers only a narrow subset of possible operating conditions and lacks direct measurements of mass flows within the distribution pipes. Therefore, we generate training, validation, and testing datasets through systematic sampling of boundary conditions within the network's known installed capacities, utilizing physics-based simulations to generate the ground truth.

Since the supply and return lines share identical topology and properties, corresponding pipes in the two lines will have the same mass flow under a given set of boundary conditions. Therefore, it is sufficient to explicitly generate boundary conditions that ensure broad coverage on the supply side, implicitly covering the same problem space on the return. We begin by defining the operational bounds. We calculate the minimum and maximum total mass flow rates for each of the 37 branches ramifying from the internal loops, excluding the branch containing the MAIN heat plant. The maximum mass flow rate is estimated based on the branch's cumulative peak power (as provided by the DHN operator), assuming a standard design temperature difference of 30 K at substations. For a minority of substations where historical monitoring indicated flows exceeding this design estimate, the peak mass flow rate is set to the maximum measured value plus a 5% margin. The minimum mass flow rate for each branch is set to 0. In branches containing a secondary heat plant, this minimum is adjusted by subtracting the plant's maximum potential output (computed from peak power and $\Delta T = 30\,\mathrm{K}$) to account for potential reverse flows. We then use Latin Hypercube Sampling (LHS) to draw synthetic scenarios between these bounds, providing a uniform coverage of the high-dimensional input space that spans the full range of plausible operating conditions. Mass flow in the remaining branch containing the MAIN heat plant is adjusted to ensure mass balance for each scenario.

Finally, we evenly distribute each branch mass flow among its non-pipe elements as boundary conditions and simulate each scenario using PyDHN, with a convergence threshold of 1 Pa. This process is repeated to create separate sets of values for training (150,000 cases), validation (50,000 cases), and testing (50,000 cases). Each case is split into two condensed graphs, as described in Section 3.4.4. The subsets of chosen internal cycles for the supply and return line are shown in Figure 3.6. With this specific cycle configuration, the farthest cycles are separated by a distance of 3 (i.e., three hops are required to traverse from one cycle to another; for example, cycles 3 and 4). Considering the model architecture described in Section 3.4.6, which alternates message passing between pipe and cycle vertices across layers, at least 5 layers are necessary to ensure information exchange between all cycles.

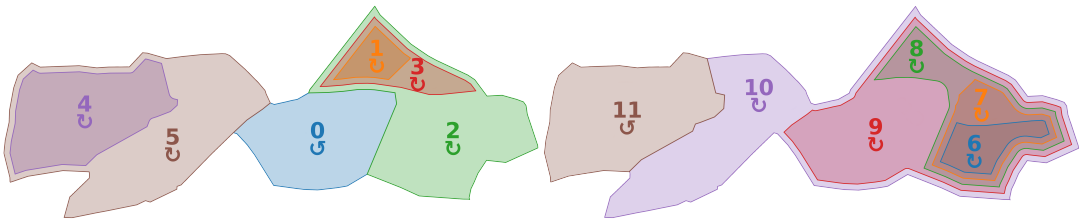



Figure 3.6: Selected internal cycles for the supply (**left**) and return (**right**) lines with their respective IDs and directions. The section of the network with these cycles is highlighted in Figure 3.5.

### 3.5.3 Model Configurations

The proposed reference (*ELU-2-7l*) architecture, described in Section 3.4.6, is a two-stage GNN composed of two stacked sub-models with 7 message-passing layers each. To better understand the contribution of different architectural choices, we compare the proposed model with a set of variants, denoted as *[activation]-[#stages]-[#layers]*. We vary:

- **Activation function:** To isolate the benefit of our proposed ELU activation, we test variants that use the ReLU and BatchNorm combination, in line with the original DyResGEN formulation.
- **Number of stages:** To study the effect of staging on oversmoothing, we compare our two-stage architecture against single-stage variants.
- **Model depth (single-stage only):** To evaluate the impact of depth, we test single-stage models with 7 and 15 layers. The 15-layer model has a comparable number of parameters and message-passing operations to our two-stage model, providing a direct comparison.

All models use a hidden dimensionality of 256 for node embeddings and are trained using the AdamW optimizer (Loshchilov & Hutter, 2019) for 100 epochs. The learning rate is initialized at 0.0001 and decayed by a factor of 10 every 10 epochs. We use a batch size of 32 and a weight decay factor of 0.01. To account for variability due to initialization, each configuration is run 10 times with different random seeds.

### 3.5.4 Evaluation Metrics

We evaluate model performance using two groups of metrics: one for model selection, evaluated on the validation set, and a second set for further analysis of the best-performing model, evaluated on the test set. The following metrics are used for model selection:

1. **Accuracy (Acc)**: the percentage of test samples for which all cycle residuals in Equation (3.2) are below 100 Pa.

2. **Mean Absolute Error on pipe-level mass flow ($\mathbf{MAE_m}$)**: the average prediction error in mass flow across all pipes and samples.

3. **Average cycle residual ($\mathbf{Res_c}$)**: the average absolute residual (Pa) per cycle, averaged over all cycles and samples.

For each model configuration, we retain the parameters corresponding to the epoch with the highest *Acc*, and report the average and best performance over the 10 runs. After identifying the best-performing architecture based on *Acc*, we further characterize its robustness using:

4. **Sample-level maximum residual ($\mathbf{Res_s}$)**: the average of the sample-wise maximum absolute cycle residuals, taken over the entire test set.

5. **Global maximum residual ($\mathbf{Res_m}$)**: the single highest absolute cycle residual observed across all test samples.

### 3.5.5 Implementation Details

All experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3090 and 16 GB of RAM. The models were implemented in PyTorch 2.1.2 (Ansel et al., 2024) and PyTorch Geometric 2.5.0 (Fey & Lenssen, 2019).

## 3.6 Results

We compare variants of the proposed GNN architecture along two dimensions: (i) single-stage (7 and 15 layers) vs. two-stage design, and (ii) ELU-based vs. ReLU-based blocks. Table 3.1 summarizes the results for the six configurations described in Section 3.5.3 according to the metrics detailed in Section 3.5.4.

Table 3.1: Validation set performance over 10 runs for each tested architecture. Values reported as mean ± std (best) over the different seeds. Best values for each KPI are highlighted in bold.

| Model | $\mathbf{MAE_m}$ ($kg\,s^{-1}$) ↓ | $\mathbf{Res_c}$ (Pa) ↓ | Acc (%) ↑ |
|---|---|---|---|
| *ELU-2-7l* | **0.0040**±0.0006 (**0.0037**) | **23.72**±3.99 (**17.95**) | 86.17±9.92 (**96.94**) |
| *ELU-1-7l* | 0.0165±0.0004 (0.0162) | 86.52±6.39 (79.13) | 12.67±2.61 (16.46) |
| *ELU-1-15l* | 0.0126±0.0009 (0.012) | 74.66±10.31 (65.38) | 20.5±5.26 (27.07) |
| *ReLU-2-7l* | 0.0060±0.0004 (0.0056) | 26.41±1.94 (24.09) | **88.9**±2.06 (91.46) |
| *ReLU-1-7l* | 0.0232±0.0020 (0.0214) | 190.92±17.22 (166.08) | 1.55±0.73 (3.11) |
| *ReLU-1-15l* | 0.0272±0.0043 (0.0196) | 232.66±54.43 (127.51) | 1.48±1.38 (5.29) |

The *ELU-2-7l* variant outperformed, on average, all other tested architectures, with the best model estimating the hydraulic state of the network with a maximum residual of 100 Pa on 97% of the validation cases. However, compared to its *ReLU* counterpart, accuracy exhibited

a wider variation, ranging from 67.11% to 96.94%, whereas with *ReLU-2-7l* it ranged from 83.83% to 91.46%. Despite this, the estimation of mass flow was closer to that of physics-based simulations and the average residuals were lower. Interestingly, architectures that employ a two-stage refinement process demonstrated significantly higher accuracy compared to single-stage architectures, even when using the same total number of message-passing layers. In fact, simply increasing the number of layers in the single-stage model only improved performance by a modest margin, and even degraded average performance in the *ReLU* variant. This is consistent with our earlier analysis (see Section 3.4.6) showing that at least five layers are needed for communication between the most distant cycles. Adding more layers without architectural separation may lead to oversmoothing rather than increased expressiveness.

An overview of the performance of the best training run on the test set is given in Table 3.2. The model achieves comparable performance on the test set as it did on the validation set,

Table 3.2: Performance of the best run (model *ELU-2-7l*) on the test set: results are shown for the supply and return lines separately. Values reported as mean ± std within the set.

| **Subset** | **Acc (%)** ↑ | **MAE$_\mathbf{m}$** ($\mathrm{kg\,s^{-1}}$) ↓ | **Res$_c$ (Pa)** ↓ | **Res$_s$ (Pa)** ↓ | **Res$_m$ (Pa)** ↓ |
|---|---|---|---|---|---|
| Supply | 97.34 | 0.0034±0.0017 | 16.82±18.31 | 42.00±23.75 | 686.26 |
| Return | 96.42 | 0.0040±0.0024 | 19.03±19.26 | 45.43±25.17 | 1099.30 |

with an overall accuracy of 96.89%. This accuracy value is averaged over the supply and return lines, which, although physically identical, are represented using different cycle bases. As a result, the corresponding condensed graphs differ in structure, providing distinct inputs to the GNN despite encoding the same underlying physics (see Section 3.5.2). This setup allows us to assess the robustness of the model to variations in cycle basis selection. In practical applications, however, where the supply and return networks are symmetric, the model could be trained and evaluated on just one side, with flows for the other inferred directly. At test time, this also means that the model's effective accuracy could be approximated by selecting the best-performing cycle decomposition, rather than averaging across both. We observed slightly different model performance across the two lines: while in both cases the accuracy was above 95%, the model exhibits slightly better values for all metrics on the supply line.

A closer look at the model accuracy under different acceptance thresholds on physics residuals is given in Figure 3.7. We observe that most cases have a maximum residual below 50 Pa, including more than 70% of supply line samples. In general, cycles with ID 4 in the supply line and ID 7 in the return line most frequently exhibited the highest pressure residuals among all cycles, though no clear pattern or explanation was identified.

To better understand when the surrogate model tends to perform worse, we analyze how the magnitude of residuals varies across different flow conditions. This helps us identify whether performance degradation correlates with extreme operating regimes, such as high or low circulating mass flow. Examining the mass flow distribution in the network (Figure 3.8), we find that the residuals are, on average, higher when the mass flow is close to its lower or upper

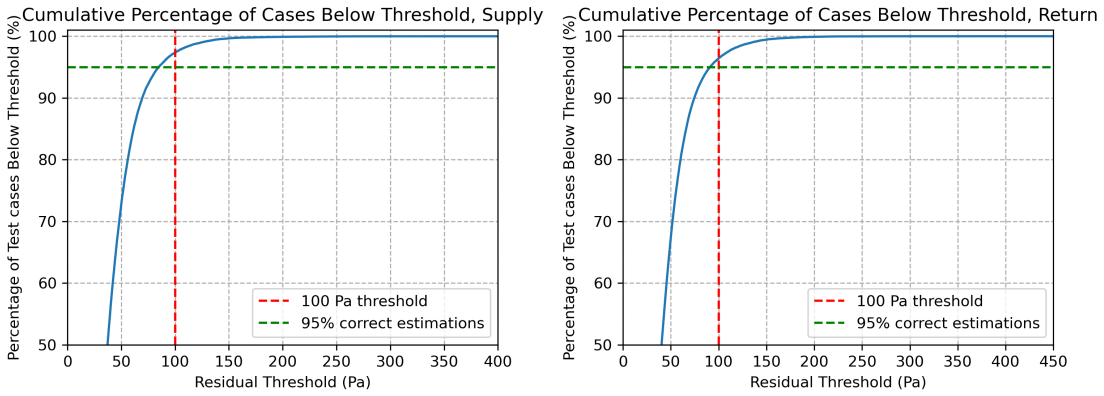



Figure 3.7: Overview of the model accuracy on the test set for the supply and return lines under different maximum residual thresholds.

limit, and particularly for values above 40 kg/s. This is likely because, although boundary conditions for each branch were sampled uniformly (LHS), extreme total flows require the rare configuration of many branches operating simultaneously at their limits. Additionally, since pressure losses scale roughly with the square of the mass flow, modulated by a flow-dependent friction factor, even small relative errors in predicted mass flows can lead to large absolute residuals under high-flow conditions.

The most accurate variant achieved an average throughput of **2187 steps/s**, representing a speedup of approximately **85**× compared to the PyDHN solver (measured on pure solver execution time, excluding overheads).

## 3.7 Ablation Study: Loss Function Variants

To assess the impact of the proposed loss design, we conduct an ablation study evaluating two simplified alternatives:

1. A model trained using only the regression loss $\mathscr{L}_{\text{reg}}$, without any physical residual supervision.

2. Models trained with fixed, equal weights for the regression and physics loss components throughout the full training, using different weight combinations.

Both variants use the same architecture as the reference model (*ELU-2-7l*), trained under identical conditions over 10 random seeds. We omit a physics-only variant, since it failed to produce meaningful results. Table 3.3 reports the average and best performance across seeds on the validation set. To reduce the impact of outliers, the averages are computed after discarding the two best and worst runs for each variant.

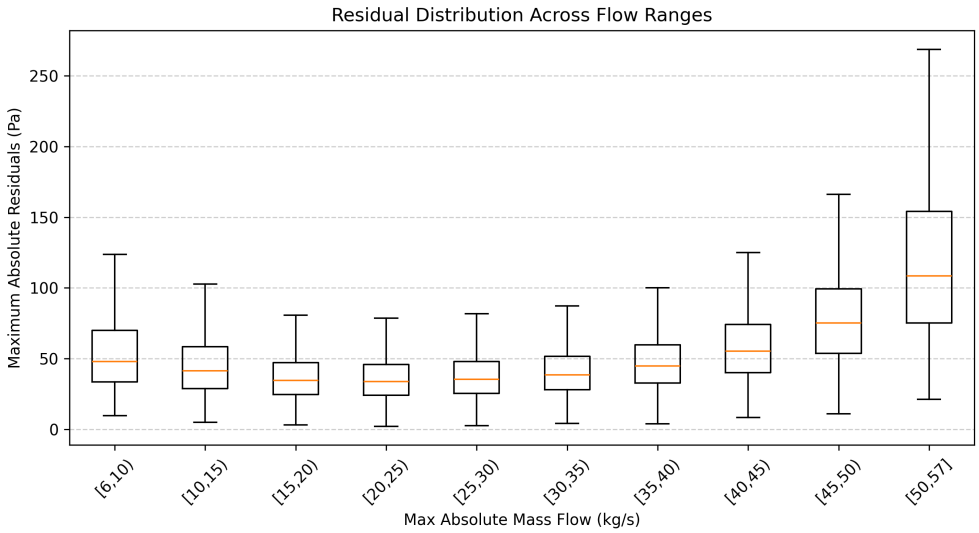



Figure 3.8: Distribution of the residual error on the test set for different ranges of maximum circulating mass flow.

The regression-only model achieves the lowest average and best $MAE_m$, but has a noticeably lower physical accuracy. While the regression loss minimizes absolute flow differences uniformly across the network, this does not translate directly into improved physical consistency. Pressure losses depend nonlinearly on pipe diameter, and linearly on length, meaning that identical flow errors can result in significantly different residuals across cycles. The physics-based loss accounts for these effects, making it a more targeted objective for improving physical fidelity, even if the overall $MAE_m$ is slightly higher. At the same time, gradually shifting training focus from data consistency to physical consistency appears beneficial for overall performance, allowing the model to first learn the general structure of the outputs before enforcing domain constraints more strictly. In fact, while the average accuracy is slightly lower than with the best fixed weight combination (by approximately 1.5%) when using the weight scheduler, the average cycle residuals are consistently reduced, indicating an overall improvement on physical consistency.

## 3.8 Discussion

The performance of the proposed model on a wide problem space, despite the differing representations of the supply and return lines, suggests promising generalization capabilities. In this context, we argue that the minor discrepancies between the supply and return lines are too small to indicate any fundamental limitations of the model. Notably, residual errors tend to be higher at extreme flow conditions. A potential strategy to alleviate this issue is to generate more training samples within this problem space. Alternatively, non-converged predictions could serve as initial guesses for iterative solvers, potentially accelerating their convergence.

Table 3.3: Ablation study on the loss function. Comparison between regression-only loss (*Regression*), fixed-weight loss (reported as $w_1$, $w_2$), and dynamic weighting (*Ours*). Validation set performance over 10 runs reported as mean ± std (best). Best values in bold.

| **Variant** | **MAE$_m$** (kg s$^{-1}$) ↓ | **Res$_c$** (Pa) ↓ | **Acc (%)** ↑ |
|---|---|---|---|
| *Regression* | **0.0029**±0.0001 (**0.0027**) | 23.09±1.11 (19.90) | 83.08±2.97 (90.24) |
| *0.01, 0.99* | 0.0085±0.0007 (0.0071) | 27.53±1.17 (24.12) | 89.74±0.82 (93.89) |
| *0.1, 0.9* | 0.0080±0.0008 (0.0085) | 27.81±0.51 (26.10) | 88.75±1.61 (91.86) |
| *0.3, 0.7* | 0.0080±0.0004 (0.0085) | 26.78±0.83 (26.32) | 90.56±0.52 (92.11) |
| *0.5, 0.5* | 0.0079±0.0004 (0.0072) | 26.66±0.97 (23.78) | 90.75±0.78 (92.83) |
| *0.7, 0.3* | 0.0076±0.0005 (0.0073) | 25.57±1.25 (24.91) | **91.85**±1.05 (93.64) |
| *0.9, 0.1* | 0.0071±0.0001 (0.0065) | 25.55±2.06 (21.99) | 91.68±3.52 (94.65) |
| *0.99, 0.01* | 0.0056±0.0002 (0.0055) | 24.92±3.07 (21.72) | 89.95±7.26 (95.34) |
| *Ours* | 0.0038±0.0001 (0.0037) | **22.79**±1.30 (**17.95**) | 89.23±3.05 (**96.94**) |

Based on the results, the proposed surrogate model shows significant potential for real-time optimization, achieving good agreement with physics-based methods while being approximately 85 times faster in our setup. Importantly, our approach can simulate multiple scenarios simultaneously, allowing more efficient operational optimization.

Consistent with the trade-offs discussed in Chapter 1, the primary limitation of this surrogate model is modeling flexibility. Unlike the physics-based framework presented in Chapter 2, which allows for the modular integration of custom components, the surrogate model cannot incorporate new component types or physical behaviors without retraining. Furthermore, while the GNN architecture is inherently transferable, the trained model instance is specific to the case study. Further research is required to determine if a single model can generalize across different network layouts or handle the structural perturbations typical of design optimization.

However, as highlighted in Section 1.3.1, these limitations are acceptable for operational optimization, where the system hardware and topology are already defined and the priority lies in computational speed rather than component modularity

## 3.9 Chapter Conclusion

This chapter addressed the fundamental computational bottleneck inherent in the hydraulic simulation of meshed DHNs identified in Section 2.6.1. We introduced a GNN surrogate model that works on a novel condensed graph representation to capture the global dependencies of pressure and mass flow without relying on computationally expensive iterative numerical solvers.

The proposed framework demonstrates a significant advancement in simulation efficiency, achieving a throughput of **2187 steps/s**. This corresponds to a speedup of approximately **85**× compared to the physics-based PyDHN reference, while maintaining a high degree of

physical fidelity, with over 97% of test samples satisfying the convergence criteria within a tight tolerance.

Consistent with the trade-offs discussed in Section 3.1, the primary limitation of this approach is its topological specificity. While the GNN architecture is inherently transferable, the trained model parameters are specific to the fixed network layout. However, in the context of operational optimization, where the infrastructure is defined and the main requirements are simulation speed and accuracy, this specialization allows for higher fidelity predictions without the overhead of generalized training. The ability to evaluate thousands of hydraulic states per second opens new possibilities for real-time control and large-scale scenario analysis that were previously computationally prohibitive.

This hydraulic surrogate serves as the first component of the simulation workflow proposed in this thesis for the operational optimization of DHNs. In the following chapter, we will complement this work by developing a differentiable, GPU-accelerated thermal model. We will then integrate both components into a hybrid framework and apply it to the full real-world case study of Verbier, utilizing historical monitoring data to validate the system's end-to-end performance and demonstrate its application in gradient-based operational optimization.

# 4 Hybrid Differentiable Modeling: Real-World Validation and Application

*The content of this chapter is primarily drawn from the following publications.*

*Boghetti, R., & Kämpf, J. H. (2024). Verification of an open-source Python library for the simulation of district heating networks with complex topologies.* Energy, 290*, 130169.*

*Boghetti, R., & Kämpf, J. H. (2026). Optimizing Supply Temperature Control in District Heating Networks via Differentiable Dynamic Simulation and Gradient Descent.* Construction, Energy, Environment and Sustainability. ICCEES 2025. Lecture Notes in Civil Engineering.

*Boghetti, R., Odobez, J. M., & Kaempf, J. H. (2025). Graph Neural Network-Based Surrogate Modeling for Fast and Scalable Simulations of Meshed District Heating Networks.* Preprint Available at SSRN 5576535.

## 4.1 Introduction

The previous chapter introduced a fast and accurate surrogate model for simulating distribution network hydraulics, addressing one of the primary computational bottlenecks in DHN simulation. To make this surrogate useful for operational optimization, it must be paired with a thermal model that is equally efficient and, ideally, compatible with gradient-based optimization methods.

This chapter completes the proposed operational optimization framework by introducing a differentiable Lagrangian thermal model. In contrast to traditional thermal simulations, which often break differentiability and therefore cannot be embedded into gradient-based optimization algorithms (Maurer et al., 2021), our formulation enables end-to-end gradient propagation across multiple simulation time steps. We integrate this thermal component with the hydraulic surrogate from Chapter 3, yielding a unified, fast, and fully differentiable hybrid simulation framework.

We then assess the capabilities of this hybrid framework, alongside the physics-based model from Chapter 2, against monitoring data from the Verbier DHN.

Finally, we demonstrate the practical value of the differentiable framework in an operational control setting by optimizing supply heating curves via gradient descent to reduce heat losses while maintaining consumer comfort constraints.

## 4.2 State of the Art

Building on the review of physical modeling principles in Chapter 2 and the discussion of data-driven surrogates in Chapter 3, this section focuses on the state of control strategies in 3GDH networks. In particular, we examine why traditional rule-based control remains prevalent despite the advantages of advanced methods such as MPC, and we highlight the opportunities for improving current operational practice.

### 4.2.1 Overview of Operational Control Strategies

The control of supply temperature in 3GDH networks has historically relied on manual operation or rule-based strategies, most notably **heating curves** (also called temperature compensation curves), which define the supply temperature as a simple, and frequently linear, function of the outdoor air temperature (Chicherin, 2025; Laakkonen et al., 2017; Madsen et al., 1994; Nigro et al., 2025). While robust and simple to implement, these static rules fail to account for the network's thermal inertia and for time-varying demand patterns, often leading operators to increase supply temperatures to compensate for unmodeled factors (Bergsteinsson et al., 2022; Vandermeulen et al., 2018).

In practice, heating curves are tuned over long periods of operation, typically an entire year, with parameters adjusted slowly based on observed performance. This process results in extended periods of suboptimal control and unnecessary energy consumption. Moreover, when substantial changes occur in the network, the tuning must be repeated, making it difficult to adapt efficiently to evolving operating conditions. This operational reality was corroborated by findings from the Eguzki project (Bonvin et al., 2024) and validated by the consortium partners, including RWB Groupe, a specialized engineering firm with decades of experience in the design and management of thermal networks.

To address the limitations of traditional rule-based control, more advanced alternatives have been proposed in the literature, primarily categorized into operational optimization, model predictive control (MPC), multi-agent systems (MAS), and purely data-driven approaches. Operational optimization focuses on optimizing control over a short term, typically hours to days, based on a system model and predictions of demand, weather, and energy prices (Vandermeulen et al., 2018) . However, the large-scale dimension and non-linear dynamics inherent to detailed network models create a computational burden that often exceeds the

limits required for practical implementation (Nigro et al., 2025). This complexity is exacerbated by the fact that these tasks are often formulated as Mixed Integer Non-Linear Problems (MINLP), for which efficient solvers are lacking (Vandermeulen et al., 2018).

An advancement over offline optimization is MPC, an online method that applies optimal control profiles in a receding horizon fashion, utilizing feedback from the network to correct prediction errors and handle stochastic disturbances (Buffa et al., 2021). Although MPC has demonstrated significant cost savings and improved efficiency in studies, its widespread implementation is hindered by high computational demands and the difficulty of deriving precise system models that can predict unexpected disturbances (Buffa et al., 2021; Nigro et al., 2025). Furthermore, the development and maintenance of MPC require highly skilled personnel and significant investment, which may not be immediately repaid by operational savings (Buffa et al., 2021). Well-documented real-world demonstrations of MPC remain relatively scarce (Saloux et al., 2024).

Alternatively, Multi-Agent Systems (MAS) offer a decentralized control approach where agents represent subsystems, making them suitable for the modular and complex nature of thermal networks (Vandermeulen et al., 2018). While MAS can increase robustness and scalability, these systems often lack a strong central unit to effectively manage the overall network dynamics, and there is a current lack of standardization in communication protocols (Buffa et al., 2021; Vandermeulen et al., 2018).

Finally, purely data-driven methods, including Machine Learning (ML) and Reinforcement Learning (RL), have been explored for load forecasting and system optimization, offering the advantage of not requiring detailed physical models (Ma & Jørgensen, 2024; Mugnini et al., 2022). However, approaches like RL require onerous training processes and large, high-quality datasets to avoid poor performance when extrapolating beyond training conditions, and they often lack guarantees that hard operational constraints will be satisfied (Mugnini et al., 2022). Furthermore, the data available for training such models is often limited by the fact that operators are typically not allowed to install sensors within buildings or access internal heating systems, thereby restricting the availability of crucial building-side data (Guelpa et al., 2023). Ultimately, the transition to these intelligent technologies is constrained by technical barriers, high initial costs, and a shortage of workforce skills in data science and control engineering (Ma & Lygnerud, 2025).

Given that rule-based controls remain the practical standard for the vast majority of existing networks, improving the parameterization of heating curves via offline optimization constitutes a high-impact, low-barrier strategy. This approach enhances efficiency without necessitating a complete overhaul of the control infrastructure or the deployment of complex online optimization schemes. Consequently, this represents an ideal use case for our proposed differentiable thermal model. Following the model's validation, we will demonstrate its utility by applying it to the optimization of heating curves. To contextualize this approach, the following subsection provides an overview of gradient-based optimization methods applied

to DHNs.

### 4.2.2 Gradient-Based DHN Optimization

To allow the use of efficient gradient-based optimizers, the underlying simulation model must be differentiable: it should be possible to propagate derivatives from a cost function defined on the model's output to the input parameters. In the context of DHNs, gradient-based optimization has only been explored in few works that applied it to network design and topology optimization. In particular, Blommaert et al. (2020) and Wack, Baelmans, et al. (2023) utilized adjoint-based methods to optimize pipe routing and sizing for large-scale networks, demonstrating that gradient-based methods scale better than mixed-integer programming or heuristics for large design spaces. A key advantage of adjoint-based methods is that they compute gradients with a computational cost that is independent of the number of design variables, allowing them to scale well for large networks. Similarly, Pizzolato et al. (2018) applied topology optimization for robust network design, and Lambert and Spliethoff (2024) employed a two-phase non-linear programming approach for expansion planning. However, these works primarily target the static design phase (sizing, topology) rather than dynamic operation (temperature control).

A fundamental limitation of these frameworks is their reliance on steady-state thermal models. By focusing on specific design points or worst-case scenarios to simplify the physics, these approaches neglect the transient thermal behavior and time delays inherent to fluid transport. As discussed in Chapter 2, thermal dynamic simulations would offer a more robust approach, especially for operational optimization. Among these, Lagrangian models are often preferred for their speed and accuracy (Dénarié et al., 2019). Yet, standard implementations of Lagrangian models rely on discrete logic operations, such as handling step-functions in temperature profiles, that break the computational graph, rendering the model non-differentiable (Maurer et al., 2021). Consequently, there is a lack of frameworks that combine the accuracy of Lagrangian thermal tracking with the differentiability required for modern gradient-based operational optimization. This chapter addresses this gap by reformulating the Lagrangian approach to be fully compatible with automatic differentiation, allowing the efficient optimization of control parameters.

## 4.3 Contributions

The preceding chapter established a fast and topology-agnostic surrogate model for network hydraulics, resolving a significant computational bottleneck in DHN simulation. However, for this surrogate to be effectively utilized in operational optimization, it must be coupled with a thermal model that is not only computationally efficient but also fully compatible with modern gradient-based optimization algorithms.

As discussed in the state of the art, the application of gradient-based optimization in DHNs

has predominantly relied on steady-state models to ensure differentiability, inherently neglecting the system's thermal inertia. While dynamic models capture these transients, they are rarely employed in gradient-based frameworks due to computational costs or formulation constraints. In particular, Lagrangian approaches are favored due to their speed and accuracy. However, their standard implementations depend on discrete logic operations that break the computational graph, rendering them incompatible with automatic differentiation.

This chapter bridges this gap by introducing a fully differentiable Lagrangian thermal model. We integrate this thermal component with the hydraulic surrogate from Chapter 3 to form a hybrid differentiable simulation framework.

We then proceed to a comprehensive validation using real-world monitoring data from the Verbier district heating network. We first verify the fully physics-based model introduced in Chapter 2 to confirm its validity as ground truth reference, and subsequently assess the performance of the proposed hybrid framework. Finally, we apply this framework to a practical control problem: the optimization of supply temperature heating curves. We show that the differentiable nature of the model allows for the direct tuning of control parameters via gradient descent, outperforming both manual baselines and gradient-free evolutionary algorithms.

The specific contributions of this chapter are:

- The formulation of a **GPU-optimized differentiable Lagrangian thermal model** that resolves the non-differentiability of standard discrete-event tracking by solving the heat transfer analytically and handling advection via smooth tensor operations. This completes the **hybrid simulation framework** introduced in Chapter 3.
- The **verification of the fully physics-based model** from Chapter 2 against monitoring data from the meshed network of Verbier, demonstrating its accuracy and establishing it as a reliable baseline for evaluating surrogate approaches.
- The comprehensive **validation of the hybrid framework** on the same network, confirming its ability to reproduce real-world thermal transients with high fidelity.
- A demonstration of **gradient-based control optimization**, showing that heating curve parameters can be optimized via backpropagation through the proposed dynamic simulation to minimize heat losses while satisfying consumer comfort constraints.

## 4.4 Dynamic Thermal Model

This section presents a differentiable Lagrangian thermal model that builds upon the framework introduced in Chapter 2. The main modification lies in the Lagrangian pipe formulation: while based on the work of Dénarié et al. (2019), we have adapted the governing equations to ensure differentiability and a linear dependence of the outlet temperature on the inlet

temperature. This allows the global network system to be solved directly via a linear system of equations, eliminating the need for the iterative convergence loops required by the original reference.

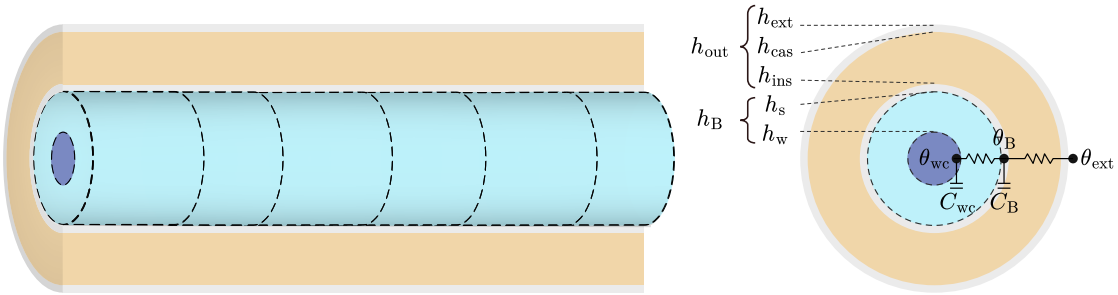



Figure 4.1: Schematic representation of the pipe model. **Left**: Illustration of the Lagrangian modeling approach, which tracks discrete water volumes of varying sizes along the pipe. The cross-section distinguishes the central water core (dark blue) from the surrounding viscous sublayer (light blue), the thickness of which is increased for visual clarity. **Right**: Schematic representation of the two-node lumped parameter thermal model. The diagram illustrates the radial heat transfer across concentric layers of the pipe system. The thermal circuit consists of two capacitive nodes: the water core temperature ($\theta_{wc}$) and the stationary boundary temperature ($\theta_B$), connected to the external environment temperature ($\theta_{ext}$).

At the component level, each pipe is modeled as a sequence of water volumes of different size, resulting from the incoming mass flow at each time step (Figure 4.1, left). A two-capacity model is used where one node represents the central turbulent water core and a second node lumps the viscous boundary layer and the pipe wall, insulation and casing (Figure 4.1, right). Consistent with the original work, we refer to this second node as the stationary boundary. This results in a system of inhomogeneous hyperbolic conservation laws that is commonly split into a source and advection problem, that, unlike the original work, we solve in this order for simplicity of implementation.

The remainder of this section is organized as follows. First, we describe the solution of the source (Section 4.4.1) and advection (Section 4.4.2) subproblems. Then, we provide details on the boundary conditions and the updated global solution strategy (Section 4.4.3), and finally discuss implementation and differentiability details (Section 4.4.4).

### 4.4.1 Analytical Solution of the Pipe Source Problem

For a given control volume, the temperature evolution of the water core ($\theta_{\text{wc}}$) and the stationary boundary ($\theta_{\text{B}}$) is governed by the coupled system of Ordinary Differential Equations (ODEs):

$$\begin{bmatrix} \frac{d\theta_{\text{wc}}}{dt} \\ \frac{d\theta_{\text{B}}}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{h_{\text{B}}}{C_{\text{wc}}} & \frac{h_{\text{B}}}{C_{\text{wc}}} \\ \frac{h_{\text{B}}}{C_{\text{B}}} & -\frac{h_{\text{B}}+h_{\text{out}}}{C_{\text{B}}} \end{bmatrix} \begin{bmatrix} \theta_{\text{wc}} \\ \theta_{\text{B}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{h_{\text{out}}}{C_{\text{B}}}\theta_{\text{ext}} \end{bmatrix} \tag{4.1}$$

Compactly defined as:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{K}\boldsymbol{\theta} + \boldsymbol{b} \tag{4.2}$$

Here, $C_{\text{wc}}$ and $C_{\text{B}}$ ($\text{J m}^{-1}\,\text{K}^{-1}$) represent the thermal capacities of the water core and the stationary boundary, respectively. The linear heat transfer coefficient $h_{\text{B}}$ ($\text{W m}^{-1}\,\text{K}^{-1}$) is defined according to the original work from Dénarié et al. (2019), while $h_{\text{out}}$ follows Equation (2.16). We assume that $C_{\text{wc}}, C_{\text{B}} > 0$ and $h_{\text{B}}, h_{\text{out}} > 0$. Note that $C_{\text{wc}}$, $C_{\text{B}}$ and $h_{\text{B}}$ (via the boundary layer) all depend on temperature-varying water properties (density, specific heat, viscosity, conductivity). Consequently, the thermal dynamics in Equation (4.1) are strictly non-linear.

To allow an efficient analytical solution of the system, we assume the fluid properties remain constant over the time interval $\Delta t$. In particular, at time step $\tau \geq 1$, we evaluate these properties using the water core temperature $\theta_{\text{wc}}^{(\tau-1)}$ from the previous step and the relationships from (Popiel & Wojtkowiak, 1998).

With this simplification, the system in Equation (4.1) is an inhomogeneous piecewise Linear Time-Invariant (LTI) system over the discretization interval $\Delta t$. We solve this system analytically using eigenvalue decomposition. This approach is preferred over numerical integration (e.g., Runge-Kutta) to ensure numerical precision and stable gradient propagation. The exact solution at time step $\tau$ is given by the superposition of a particular solution ($\boldsymbol{\theta}_p$) and the general solution to the complementary equation:

$$\boldsymbol{\theta}^{(\tau)} = \boldsymbol{\theta}_{\text{p}} + c_1 \mathbf{v}_1 e^{\lambda_1 \Delta t} + c_2 \mathbf{v}_2 e^{\lambda_2 \Delta t} \tag{4.3}$$

Here, the terms $\lambda_{1,2}$ and $\mathbf{v}_{1,2}$ correspond to the eigenvalues and eigenvectors of the matrix $\boldsymbol{K}$, respectively, and $c_1$ and $c_2$ are scalar constants. To compute the solution efficiently, we exploit the low dimensionality of the system. The eigenvalues $\lambda_{1,2}$ are defined as the roots of the characteristic polynomial:

$$\det(\boldsymbol{K} - \lambda \boldsymbol{I}) = 0 \tag{4.4}$$

where $\det(\cdot)$ denotes the determinant and $\boldsymbol{I}$ is the identity matrix. Given that $\boldsymbol{K}$ is a $2 \times 2$ matrix, this expands to the quadratic equation:

$$\lambda^2 - \text{tr}(\boldsymbol{K})\lambda + \det(\boldsymbol{K}) = 0 \tag{4.5}$$

where $\text{tr}(\cdot)$ denotes the trace of the system matrix. The eigenvalues are thus obtained analytically via the quadratic formula:

$$\lambda_{1,2} = \frac{\text{tr}(\boldsymbol{K}) \pm \sqrt{\text{tr}(\boldsymbol{K})^2 - 4\det(\boldsymbol{K})}}{2} \tag{4.6}$$

Under the physical constraints stated above ($C_{\text{wc}}, C_{\text{B}} > 0$ and $h_{\text{B}}, h_{\text{out}} > 0$), the diagonal entries of matrix $\boldsymbol{K}$ are non-positive, while the off-diagonal entries are non-negative. This ensures that

$\mathrm{tr}(\boldsymbol{K}) \leq 0$ and that the discriminant $\mathrm{tr}(\boldsymbol{K})^2 - 4\det(\boldsymbol{K}) \geq 0$, yielding real eigenvalues. Since each eigenvalue $\lambda_i$ satisfies the characteristic equation, the matrix $\boldsymbol{K} - \lambda_i \boldsymbol{I}$ is singular by definition. Consequently, its rows are linearly dependent, meaning that any non-zero row spans the same one-dimensional null space. The corresponding eigenvector $\mathbf{v}_i$ can thus be determined from any single row. To maximize numerical stability and avoid vanishing gradients, we identify the row with the largest $L^2$ norm and denote it as $\mathbf{u} = [u_1, u_2]^\top$. In practice, a small regularization constant ($\epsilon = 10^{-8}$) is added to the normalization factor to prevent numerical edge cases. The eigenvector is computed analytically as the normalized orthogonal complement of

$$\tilde{\mathbf{v}}_i = \begin{bmatrix} -u_2 \\ u_1 \end{bmatrix}, \quad \mathbf{v}_i = \frac{\tilde{\mathbf{v}}_i}{||\tilde{\mathbf{v}}_i||} \tag{4.7}$$

The particular solution $\boldsymbol{\theta}_\mathrm{p}$ corresponds to the steady-state condition where thermal losses balance external heat input. Setting $\dot{\boldsymbol{\theta}} = \mathbf{0}$ in Equation (4.1) and solving the resulting linear system yields $\boldsymbol{\theta}_\mathrm{p} = [\theta_\mathrm{ext}, \theta_\mathrm{ext}]^\top$, representing thermal equilibrium with the surrounding environment.

Finally, the scalar coefficients $c_1$ and $c_2$ are determined by projecting the deviation from the steady state at the previous time step $\tau - 1$ onto the eigenvector basis. This requires solving the linear system:

$$c_1\mathbf{v}_1 + c_2\mathbf{v}_2 = \boldsymbol{\theta}^{(\tau-1)} - \boldsymbol{\theta}_\mathrm{p} \tag{4.8}$$

By substituting these back into the general solution, the new temperatures can be found analytically. Note that the solution obtained here represents only the effect of thermal losses within each control volume over $\Delta t$. It does not yet account for advection along the pipe. For this reason, we denote the resulting state by $\hat{\boldsymbol{\theta}}^{(\tau)} = [\hat{\theta}_\mathrm{wc}^{(\tau)}, \hat{\theta}_\mathrm{B}^{(\tau)}]^\top$, emphasizing that it is an intermediate temperature update that will be further advected in the subsequent step of the simulation.

Given the updated water core temperature, we can compute the thermal loss as:

$$Q_\mathrm{loss}^{(\tau)} = \left(\theta_\mathrm{wc}^{(\tau-1)} - \hat{\theta}_\mathrm{wc}^{(\tau)}\right) \rho^{(\tau-1)} V c_\mathrm{p}^{(\tau-1)} \frac{1}{c}, \tag{4.9}$$

where $V$ is the volume of the water core segment and $c = 3600\,\mathrm{J\,W\,h^{-1}}$.

### 4.4.2 Solution of the Pipe Advection Problem

After solving the source problem for each water volume, we address the advection of water through the pipe network.

The advection step accounts for the spatial displacement of water volumes due to mass flow, updating both the position and the thermal state of the water core. The inputs to this step are the volume configuration at time $\tau - 1$ (which has not yet moved) and the thermally updated temperatures resulting from the source problem.

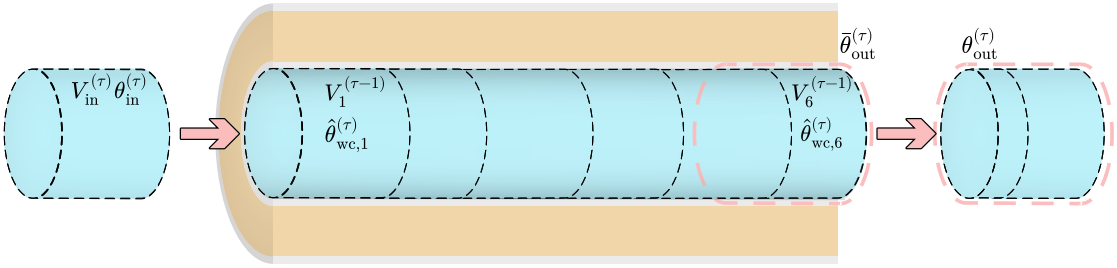



Figure 4.2: Illustration of the pipe model: advection and outlet temperature computation. Here, the incoming volume is less than the total pipe volume ($\alpha = 0$), and the outlet temperature is computed as the volume-weighted average temperature of the portion of water highlighted by the red dashed border. For visual clarity, the distinction between the water core and viscous sublayer is not depicted.

Consider a pipe of maximum volume capacity $V_{\text{max}}$ and let $\dot{m}^{(\tau)} > 0$ denote its mass flow rate over the interval $[\tau-1, \tau]$. Without loss of generality, we first describe the advection procedure for strictly positive mass flow; the treatment of zero or reversed flow is addressed subsequently. Let $N_{\text{wv}}$ denote the number of water volumes at the start of the advection step, i.e., $N_{\text{wv}} = N_{\text{wv}}^{(\tau-1)}$. The pipe state is represented by a dynamic sequence of $N_{\text{wv}}$ water volumes, characterized by:

- Volume: $V_i^{(\tau-1)} \in \mathbb{R}_+$ for $i = 1, \ldots, N_{\text{wv}}$ from the previous time step.
- Updated Water core temperature: $\hat{\theta}_{\text{wc},i}^{(\tau)} \in \mathbb{R}$ for $i = 1, \ldots, N_{\text{wv}}$, from the source solution $\hat{\boldsymbol{\theta}}^{(\tau)}$.
- Updated Wall temperature: $\hat{\theta}_{\text{B},i}^{(\tau)} \in \mathbb{R}$ for $i = 1, \ldots, N_{\text{wv}}$, from the source solution $\hat{\boldsymbol{\theta}}^{(\tau)}$.

Subject to the constraint $\sum_{i=1}^{N_{\text{wv}}} V_i^{(\tau-1)} = V_{\text{max}}$.

At each discrete time step of duration $\Delta t$, advection proceeds as follows. A new water volume enters the pipe inlet with volume:

$$V_{\text{in}}^{(\tau)} = \frac{|\dot{m}^{(\tau)}|\Delta t}{\rho(\theta_{\text{sn}}^{(\tau-1)})}, \tag{4.10}$$

Here, the density of the inlet water volume $\rho$ is evaluated using the temperature of the pipe starting node from the previous time step, $\theta_{\text{sn}}^{(\tau-1)}$. This decouples the network-wide nodal temperature calculation from the advection step, maintaining linearity in the overall system. Given the weak temperature dependence of liquid water density and the continuous nature of thermal dynamics, the error introduced by this one-step lag is negligible for sufficiently small $\Delta t$.

We then construct an augmented sequence by prepending the new volume:

$$\tilde{\boldsymbol{V}}^{(\tau)} = [V_{\text{in}}^{(\tau)}, V_1^{(\tau-1)}, \ldots, V_{N_{\text{wv}}}^{(\tau-1)}]^\top \in \mathbb{R}_+^{N_{\text{wv}}+1}, \tag{4.11}$$

$$\tilde{\boldsymbol{\theta}}_{\text{wc}}^{(\tau)} = [\theta_{\text{in}}^{(\tau)}, \hat{\theta}_{\text{wc},1}^{(\tau)}, \ldots, \hat{\theta}_{\text{wc},N_{\text{wv}}}^{(\tau)}]^\top \in \mathbb{R}^{N_{\text{wv}}+1}. \tag{4.12}$$

Where $\theta_{\text{in}}^{(\tau)}$ is the inlet temperature at time step $\tau$. Water exceeding the pipe volume is expelled from the outlet. Given the cumulative volume at position $j$ in the augmented sequence:

$$S_j^{(\tau)} = \sum_{i=1}^{j} \tilde{V}_i^{(\tau)}, \quad j = 1, \ldots, N_{\text{wv}} + 1, \tag{4.13}$$

the excess volume for each element $j$ is computed as:

$$E_j^{(\tau)} = \min\left(\max(S_j^{(\tau)} - V_{\max}, 0), \tilde{V}_j^{(\tau)}\right), \quad j = 1, \ldots, N_{\text{wv}} + 1, \tag{4.14}$$

which ensures $0 \leq E_j^{(\tau)} \leq \tilde{V}_j^{(\tau)}$.

The outlet temperature is computed as the volume-weighted average of the expelled portion:

$$\theta_{\text{out}}^{(\tau)} = \frac{\sum_{j=1}^{N_{\text{wv}}+1} E_j^{(\tau)} \tilde{\theta}_{\text{wc},j}^{(\tau)}}{\sum_{j=1}^{N_{\text{wv}}+1} E_j^{(\tau)}}. \tag{4.15}$$

The updated volumes for the time step $\tau$ are:

$$V_j^{(\tau)} = \tilde{V}_j^{(\tau)} - E_j^{(\tau)}, \quad j = 1, \ldots, N_{\text{wv}} + 1. \tag{4.16}$$

Only non-zero volumes (along with their corresponding temperatures) are retained in the sequence, yielding $N_{\text{wv}}^{(\tau)} \leq N_{\text{wv}} + 1$.

Note that the volume-weighted average of Equation (4.15) can be ideally split into two parts: one that depends on $\theta_{\text{in}}^{(\tau)}$ and one that does not. In particular, it can be written as:

$$\theta_{\text{out}}^{(\tau)} = \alpha^{(\tau)} \theta_{\text{in}}^{(\tau)} + (1 - \alpha^{(\tau)}) \bar{\theta}_{\text{out}}^{(\tau)}, \tag{4.17}$$

$$\alpha^{(\tau)} = \max\left(0, \frac{V_{\text{in}}^{(\tau)} - V_{\max}}{V_{\text{in}}^{(\tau)}}\right), \tag{4.18}$$

where $\bar{\theta}_{\text{out}}^{(\tau)}$ represents the volume-weighted average temperature of the portion of outflow originating from water already in the pipe at the start of the time step. The process is illustrated in Figure 4.2.

Since both $\bar{\theta}_{\text{out}}^{(\tau)}$ and $\alpha^{(\tau)}$ do not depend on $\theta_{\text{in}}^{(\tau)}$ under the assumptions made, this is effectively a linear model.

### Stationary Boundary Update

While the water core advects with the flow, the stationary boundary remain fixed in space. However, because our model discretizes the pipe based on the moving water volumes, the numerical grid describing the stationary boundary shifts at every time step. Consequently, the temperature profile of the stationary boundary must be remapped from the spatial configuration at time $\tau-1$ to the new configuration at time $\tau$. We formulate this as a 1D conservative remapping problem. Let the spatial domain of the pipe be defined by the coordinate $x \in [0, \ell]$. The locations of the water volumes at the previous and current time steps define two distinct grids, denoted by the intervals $\Omega_j^{(\tau-1)}$ and $\Omega_i^{(\tau)}$, respectively:

$$\Omega_j^{(\tau-1)} = [x_{j-1}^{(\tau-1)}, x_j^{(\tau-1)}), \quad \text{with } x_j^{(\tau-1)} = \sum_{k=1}^{j} V_k^{(\tau-1)}, \tag{4.19}$$

$$\Omega_i^{(\tau)} = [x_{i-1}^{(\tau)}, x_i^{(\tau)}), \qquad \text{with } x_i^{(\tau)} = \sum_{k=1}^{i} V_k^{(\tau)}. \tag{4.20}$$

Since the physical wall temperature distribution is invariant to the grid movement, the temperature $\theta_{\mathrm{B},i}^{(\tau)}$ of the $i$-th new volume is computed as the weighted average of the temperatures from the overlapping intervals in the previous grid. The overlap length between a new interval $i$ and an old interval $j$ is given by:

$$L_{ij} = \max\Big(0, \min(x_i^{(\tau)}, x_j^{(\tau-1)}) - \max(x_{i-1}^{(\tau)}, x_{j-1}^{(\tau-1)})\Big). \tag{4.21}$$

This formula corresponds to the intersection length $|\Omega_i^{(\tau-1)} \cap \Omega_j^{(\tau)}|$ and ensures that only physically overlapping regions contribute to the temperature update. The new boundary temperatures are then obtained via the linear projection:

$$\theta_{\mathrm{B},i}^{(\tau)} = \sum_j \frac{L_{ij}}{V_i^{(\tau)}} \theta_{\mathrm{B},j}^{(\tau-1)}. \tag{4.22}$$

In matrix form, this is expressed as $\boldsymbol{\theta}_{\mathrm{B}}^{(\tau)} = \boldsymbol{M}\boldsymbol{\theta}_{\mathrm{B}}^{(\tau-1)}$, where the entries of the mixing matrix are $\boldsymbol{M}_{ij} = L_{ij}/V_i^{(\tau)}$.

### Handling reversed and zero flow.

When $\dot{m}^{(\tau)} < 0$, the flow direction is reversed, and the advection procedure is applied by reordering the volume sequence in reverse spatial order before computing inlet and outlet dynamics, then restoring the original ordering. For $\dot{m}^{(\tau)} = 0$, no advection occurs: volumes remain spatially fixed while undergoing thermal evolution according to the source problem. The outlet temperature is then defined as the temperature of the rightmost water volume (nearest to the nominal outlet).

### 4.4.3 Model Boundary Conditions and Overall System Solution

The model boundary conditions follow the linear component formulations introduced in Chapter 2. Specifically, a producer is treated as a slack node where the outlet temperature is fixed to a setpoint $\theta_{set}^{(\tau)}$:

$$\theta_{\text{out}}^{(\tau)} = \theta_{set}^{(\tau)}. \tag{4.23}$$

A consumer is modeled as an edge that imposes a temperature drop $\Delta\theta_{set}^{(\tau)}$ between the inlet (supply) and outlet (return) nodes:

$$\theta_{\text{out}}^{(\tau)} = \theta_{\text{in}}^{(\tau)} - \Delta\theta_{set}^{(\tau)}. \tag{4.24}$$

Combining these conditions with the formulation of the Lagrangian pipe model in Equation (4.17), we obtain that all the transfer functions $\psi_j$ in Equation (2.10) are strictly linear with respect to the current inlet temperatures. Consequently, the general system described in Equations (2.9) to (2.11) simplifies to a sparse linear system of the form $\boldsymbol{Ax} = \boldsymbol{b}$. Unlike the physics-based model presented in Chapter 2, which requires iterative Newton-Raphson solvers, this system is linear and is solved directly. In particular, we solve it via LU factorization to preserve differentiability.

Note that if non-slack nodes with zero mass flow are present, **A** is singular. Therefore, the equations of such nodes are excluded from the linear system and the corresponding nodes are assigned the average value of the temperatures of immediately adjacent water volumes in pipes.

Also note that, to retain linearity, the consumer model in Equation (4.24) is not constrained, therefore it does not inherently prevent unreasonable results (e.g., $\theta_{out} < \theta_{soil}$). Physical limits can be strictly enforced, and non-linear functions allowed, by decoupling the system. In particular, given the topological assumptions with which the network graph is built (see Section 2.4.1), this involves solving the supply network first, calculating consumer outlets with explicit saturation logic or non-linear functions, and subsequently solving the return network.

### 4.4.4 Implementation Details and Differentiability.

**Initial conditions.**

At $\tau = 0$, the pipe is initialized with a single Lagrangian water volume equal to the total pipe capacity. The initial temperature field is uniform, with all water core and static boundary nodes set to 50 °C.

A pre-conditioning simulation is performed prior to the experiments, typically with fixed boundary conditions. This phase serves to replace the initial single-volume condition with a refined temperature distribution consistent with the specific network topology and boundary

settings.

**Differentiability.**

All operations are designed to maintain computational graph connectivity throughout the simulation for automatic differentiation frameworks.

The analytical solution of the **source problem** ensures stable gradient propagation. Since the thermal parameters are strictly positive ($C_{\text{wc}}, C_{\text{B}}, h_{\text{B}}, h_{\text{out}} > 0$), the characteristic polynomial's discriminant is always positive. This guarantees distinct real eigenvalues ($\lambda_1 \neq \lambda_2$), avoiding the gradient explosion associated with degenerate eigenvalues in automatic differentiation frameworks. Furthermore, the physical disparity between the thermal capacities and heat transfer rates ensures the eigenvalues remain well-separated, maintaining numerical precision. Additionally, the analytical eigenvector implementation explicitly selects the matrix row with the largest magnitude for computation, preventing numerical instability when off-diagonal elements are small.

To preserve differentiability during **advection**, we replace standard array indexing with element-wise selection based on boolean masks. This ensures gradient flow is maintained even through discrete logic such as flow reversal or volume shuffling. While the excess volume computation in Equation (4.14) employs nested min/max operations, automatic differentiation frameworks propagate valid subgradients through the active branches, preventing singularities.

The **global system** is solved through LU factorization, which allows for smooth backpropagation through the solver. To avoid numerical singularities, we explicitly omit equations for nodes with zero mass flow, solving the linear system only for the active components of the graph. The temperatures of inactive nodes are assigned via neighbor averaging, preserving the connectivity of the computational graph without compromising the stability of the factorization.

Finally, potential **singularities during division operations** are explicitly handled. In the outlet temperature calculation (Equation (4.15)), the division is numerically safe because the denominator (total expelled volume) is strictly positive when $\dot{m}^{(\tau)} > 0$; for zero-mass flow, the logic branches to a direct assignment, bypassing the division entirely. Similarly, during the stationary wall update, the division by $V_i^{(\tau)}$ is secured by explicitly masking out zero-length padding volumes before the normalization step.

**Implementation and Computational Efficiency**

The model is implemented in PyTorch to take advantage of GPU acceleration and automatic differentiation. To maximize parallel throughput, **operations are vectorized across the entire network**, treating all pipes simultaneously as a single batch. This approach naturally allows multiple independent DHNs to be simulated in parallel as part of a single, disconnected graph.

We represent the network state using dense tensors of shape $[N_{\text{pipes}}, N_{\max}^{(\tau)}]$, where the dimension $N_{\max}^{(\tau)}$ dynamically adapts to the maximum number of volumes in any pipe at the current time step. This dynamic sizing ensures memory usage remains proportional to the physical state of the system rather than a worst-case fixed allocation. While not strictly necessary for the network sizes considered in this work, and therefore not implemented, this formulation supports the merging of adjacent volumes to bound $N_{\max}^{(\tau)}$, offering a potential trade-off between memory footprint and spatial resolution for larger-scale systems.

A key feature of this implementation is the separation of memory and compute workloads. While structural operations, such as advection via parallelized cumulative sums and wall remapping, operate on the full padded tensors, the computationally intensive physics (e.g., eigenvalue decomposition) is executed exclusively on active elements via boolean masking. This ensures that the cost of complex physics scales linearly with the actual number of fluid volumes, independent of the padding.

To preserve the computational graph for automatic differentiation, we replace non-differentiable in-place mutations with custom masked assignment primitives. This applies universally to discrete state changes, such as volume injection or the virtual topology flipping used to simulate reversed flow, which is handled by reordering tensor indices prior to advection.

Regarding **static boundary temperature remapping**, we deliberately choose a dense broadcasting formulation over a sparse implementation. Although computing the intersection of all volume pairs implies a theoretical complexity of $\mathcal{O}(N_{\text{pipes}} \cdot (N_{\max}^{(\tau)})^2)$, this approach allows the use of highly optimized dense matrix kernels. For typical problem sizes ($N_{\max}^{(\tau)} \approx 10^2$), the quadratic scaling overhead is empirically negligible compared to the high latency of managing sparse indices on the GPU.

Finally, when gradients are required over long time horizons, we utilize **gradient checkpointing** to trade computation for memory. By recomputing the dense intermediate activations of the physical solver during the backward pass, we avoid storing the massive history of intermediate operations. This reduces the memory footprint to effectively just the sequence of state snapshots, without introducing approximations or degrading gradient accuracy.

## 4.5 Verification of the Proposed Workflows

The verification is aimed at assessing the applicability and performance of the two workflows proposed thus far: the physics-based model of Chapter 2 and the combination of the GNN hydraulic model of Chapter 3 with the thermal model introduced in this section. To this end, we conduct a comparison of the simulation results with the monitoring data of the Verbier DHN.

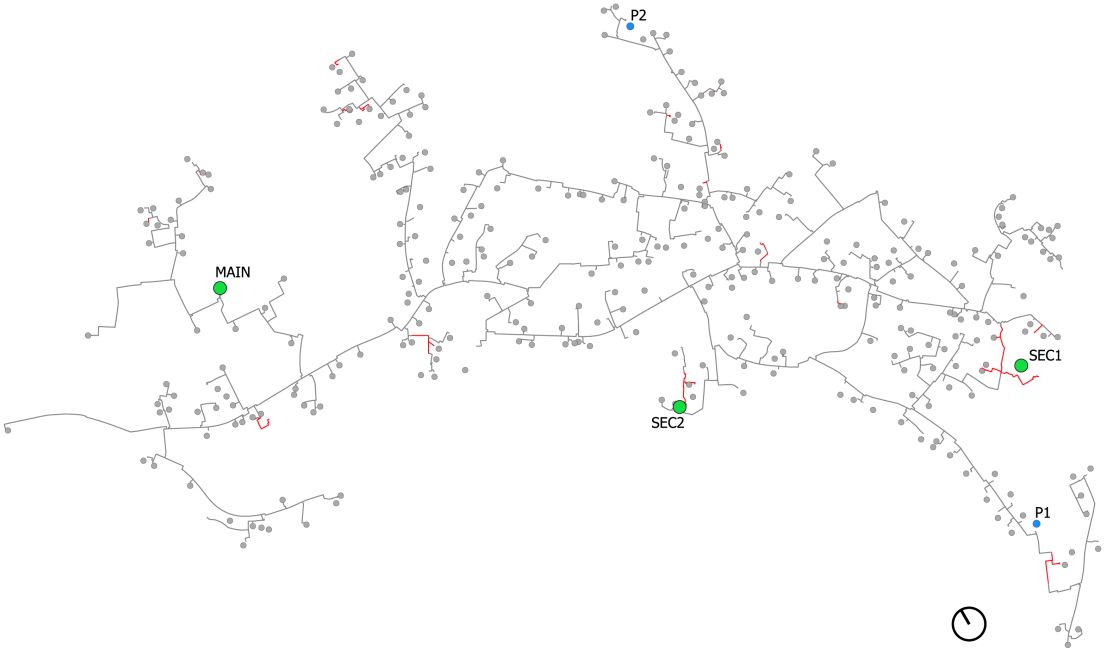



Figure 4.3: The DHN of Verbier. The producers are highlighted in green, while the substations with pressure measurements are in blue. Aerial pipes are highlighted in red.

### 4.5.1 Case Study and Data Collection

The considered case study is the network of Verbier briefly introduced in Section 3.5.1. At the time of the experiment, the DHN (Figure 4.3) is connected to 165 active substations and 3 producers (MAIN, SEC1 and SEC2). Among the active substations, 35 are exclusively dedicated to the defrosting of access ramps to underground parkings. The distribution network is of meshed type with 6 internal loops and has a total of 28 km of pipes, including dead branches, with an estimated altitude difference between the lowest and highest point of 126 m. Most pipes are in direct contact with the ground at an average depth of 0.8 m, however some pass through building basements or parking lots, for which the aerial pipe model was used. Furthermore, in a few cases, the connections between the network and the customers' substations are to be refurbished and are hence lacking the insulation layer, while the list of these occurrences is unfortunately not available.

The available monitoring data consists of average measurements (volumetric flow, temperature and pressure) and/or calculated data (power and energy) from the energy provider with a time resolution of 15 minutes, or 15 seconds in a few cases, collected between Tuesday, January 10 and Sunday, January 22, 2022. Since the location is a popular mountain resort, the analyzed period is within the ski season, hence high energy demand is expected. The data was originally collected using sensors set to register a value each time it deviates from the previous measurement more than a fixed threshold, and was then aggregated. Measurements are available for each leaf component, with the exception of pressure that was only measured

in the central plants and in two substations amongst the furthest consumers, highlighted in Figure 4.3 as P1 and P2. The temperature sensors are of class B according to the international standard IEC 60751, which equals to a tolerance of $\pm(0.3+0.005\cdot\theta)$°C at a given temperature $\theta$ (°C). The flow sensors are of class 2 (European standard EN 1434), and therefore have a tolerance of $\pm(2+0.02\cdot Q_p/Q)$%, up to a maximum of ±5%, where $Q_p/Q$ is the ratio between the maximum and current flow. Finally, the accuracy of the pressure sensors is ±0.3%. The mass flow values of SEC2 are not directly measured, but rather calculated from other available sensors by the network provider.

A quality screening was conducted on the data after averaging the measurements from the few sensors with 15-second granularity on a 15-minute basis. To this end, considering a closed network, the sum of mass flow in substations is expected to be equal to the sum of mass flow in heating stations. Over the 926 time-steps with available measurements, an average error of 2.43%, corresponding to a RMSE of 4.99 $\mathrm{m^3\,h^{-1}}$, was found. Furthermore, around half of the measurements are missing between January 16 and the end of the monitoring period, and are therefore excluded from the verification as they would have a too significant impact on the dynamics of the network. Missing measurements in the remaining period were filled with their previous values. Since SEC2 was only active for only 7 non-consecutive hours during the considered period, and with several missing data-points, results for this heating stations will not be considered.

An overview of the control strategies employed during the monitoring period is given in Figure 4.4. Unlike conventional setups where the supply temperature is regulated based on air temperature, here it is manually set. The demand peaks at around 9 AM, which is met by increasing the mass flow rate and, consequently, lowering the supply temperature. Around 10 PM, as demand decreases, the mass flow rate is reduced, and the supply temperature is increased to compensate for the longer travel time and prevent excessively low temperatures at consumer substations. A different control strategy is used for SEC1, where the temperature and mass flow are kept almost constant at 82 °C and 5 $\mathrm{kg\,s^{-1}}$ respectively.

### 4.5.2 Simulation Specifications

Simulations are run using a step-size of 60 seconds with imposed boundary conditions to the leaf components according to the monitoring data. In particular, we imposed measured values of the pressure difference at the main producer, the mass flow of SEC1 and SEC2, the outlet temperature of all producers and the mass flows and temperature differences of the consumers. The convective heat transfer coefficient for external convection in aerial pipes is estimated using the relationships for forced convection in (Çengel, 2003) considering an air temperature of 18 °C and velocity of 0.4 $\mathrm{m\,s^{-1}}$ (Matthews et al., 1989). The soil thermal conductivity and diffusivity were taken as their average values for the Canton of Valais from (Assouline et al., 2019) as respectively 0.6 $\mathrm{W\,m^{-1}\,K^{-1}}$ and 0.0864 $\mathrm{m^2\,s^{-1}}$, while an average temperature of −2.23 °C was computed with the relationship proposed by Kusuda and Achenbach (1965)

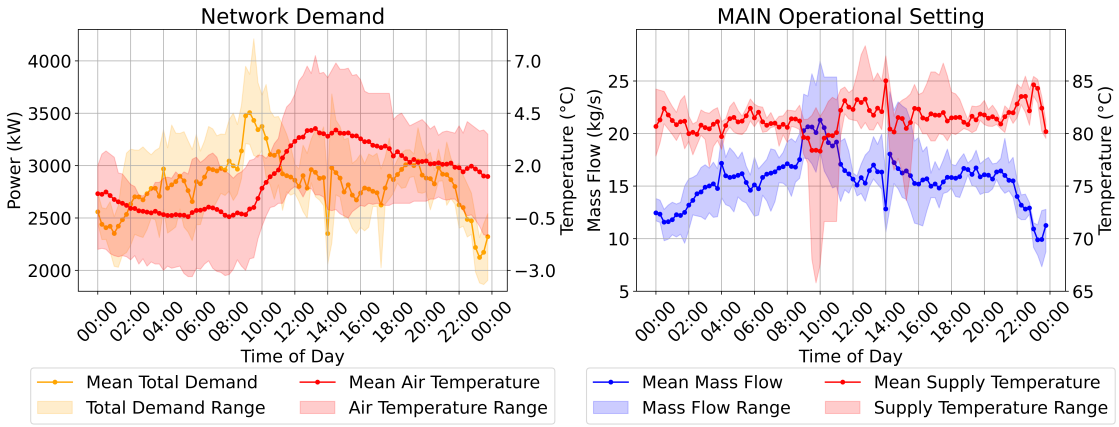



Figure 4.4: Daily variations in monitoring data, showing mean daily values and ranges. **Left**: total power demand (kW) and air temperature (°C). **Right**: mass flow ($kg\,s^{-1}$) and supply temperature (°C) at the MAIN plant. Shaded regions indicate the variability.

using air-temperature data from the nearby weather station from MeteoSwiss (Montagnier, Val de Bagnes), which is close to Verbier but at a lower altitude (839 m asl). Finally, since the height differences of the different elements in the network were not known with enough accuracy, all elements were assumed to be at the same altitude, therefore neglecting the hydrostatic pressure difference.

### 4.5.3 Verification Procedure

#### Physics-based Model

First, we assess the accuracy of the physics-based model, crucially important as this model is used to generate ground-truth values for the other models in this thesis. The verification of the simulation results is carried out by comparing the simulated and measured values of two physical quantities: pressure and the temperature. The indicator chosen for verifying the hydraulic part is the pressure difference between the inlet and outlet nodes of SEC1, P1 and P2. Since the pressure values are affected by several factors, such as the sensors' exact position, singular pressure losses and age of the pipes, which are neglected in the presented model, we expect the pressure losses to be significantly underestimated by the simulation. In order to mitigate this effect, we make the assumption that neglecting these factors leads to a systematic error, unique for each measurement point. A constant correction value, computed as the average error between the measured and simulated pressure difference in the considered element, is therefore added to the simulated pressure difference before the comparison. Then, the Mean Absolute Percentage Error (MAPE) between the monitoring and simulated values is computed and used as verification metric. In particular, since the measured value depends on 4 sensors, each with an accuracy of ±0.3%, we use ±1.2% as a reference MAPE. Because the correction factor is estimated from the same monitoring period used for evaluation, this hydraulic assessment should be interpreted as bias-corrected verification rather than a strict

validation.

On the thermal side, the chosen indicator is the return temperature of the active producers. This indicator is frequently considered as it accumulates potential errors from the temperature calculation along the whole network. Similarly to the pressure difference, we established a benchmark accuracy value of $\pm 2\,^{\circ}\mathrm{C}$. This value takes into account the accuracy of the 4 temperature sensors traversed by the water flow in its path between the two ends of the heating stations, while neglecting other non quantifiable uncertainties.

#### Hybrid Framework

We repeat the experiment with the hybrid framework, although with a different time step size for the steady-state hydraulic simulation and the thermal simulation. The hydraulic simulation is run with a 15-minute time step, to match the resolution of the operational data, assuming constant flows within each interval. The thermal simulation uses a finer 60-second step size to capture faster heat dynamics. As the supply and return lines share layout and characteristics, we use the surrogate model to estimate mass flows only for the supply line and apply the same output to the return line. The full 12-day period is simulated, but results are only evaluated over the first 6 days with reliable data, totaling 1247 hydraulic steps, 18705 thermal steps, and 1440 pre-heating steps (24 hours) for the thermal case. Using the proposed method, the hydraulic simulation completes in 0.57 seconds (2187 steps/s) versus 50 seconds with PyDHN (solver time only), with a 97.75% convergence rate. The full thermal simulation, including pre-heating, completes in 3 minutes and 47 seconds (89 steps/s).

### 4.5.4 Results

#### Physics-based Model

In order to assess the suitability of the presented models to simulating a meshed network, a simulation of the DHN of Verbier has been conducted for the period January 10 to January 15, 2022 (8640 60-second time-steps). In what follows, we consider a step size of 15 minutes, and consequently, the simulation results were re-sampled to a 15-minute interval to align with the granularity of monitoring data. For a total of 19 time-steps within the considered period, of which at most 6 are consecutive, the monitoring data was missing, and boundary conditions from the closest previous measurement were used.

An overview of the error used to compute the correction factors, as well as the distribution of the relative error on the pressure after correction is given in Figure 4.5. A positive error indicates that the simulated pressure difference is above the measured value and as a consequence that the pressure losses through the network are underestimated. After applying the correction, the MAPE was 2.45% for SEC2, 4.31% for P1 and 3.34% for P2. While in all cases this error is higher than the tolerance of the sensors, it is still within a reasonable range considering that a large share of uncertainty could not be accounted for by using a constant shift of the output

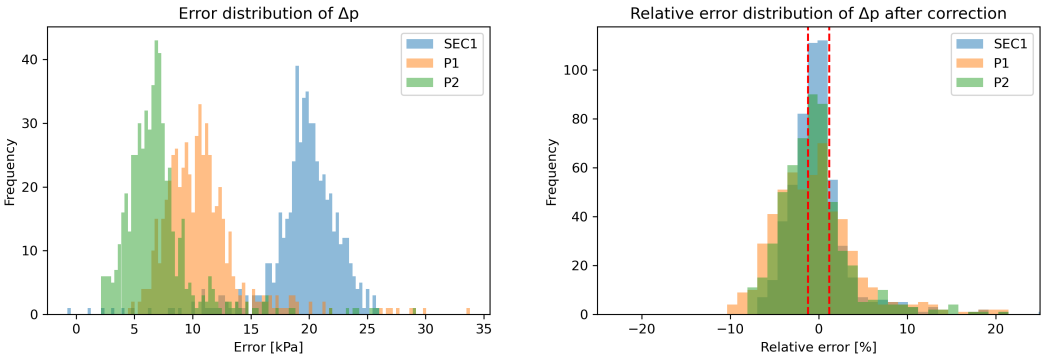



Figure 4.5: Modeling error on pressure. **Left**: distribution of the absolute error on pressure difference **Right**: relative error after the correction. The dashed red lines indicate the thresholds of ±1.2%.

values, as for example that introduced by the 15-minute averaging. The graph on the right in Figure 4.5 shows the distributions of the error with sign. These distributions have a mean of 0.11% for SEC1, 0.38% for P1 and 0.15% for P2, with standard deviations of 4.01%, 8.74% and 6.39% respectively.

The error on the return temperature at producers is given in Figure 4.6. The simulated temperature is for the most part within the threshold of ±2 °C from the target value, with an RMSE of 0.57 K for MAIN and 0.83 K for SEC1.

An error outside the identified thresholds is found in 5 time steps (0.9 % of the simulated horizon) for MAIN and 10 for SEC1 (1.8%). In all these cases except 3 the simulated temperature was higher than the measured one. The error distribution have in fact means of 0.08 °C and 0.56 °C with standard deviations of 0.43 °C and 0.71 °C for MAIN and SEC1 respectively. The low but consistent overestimation of the return temperatures might be due to several factors. The main reasons might lie in the uncertainties linked to the network characteristics, such as the unknown location of uninsulated pipes, the exact position of sensors and the effective performance of the insulation, which decreases with time. Another factor might be the use of average soil characteristics to compute its temperature that are not representative of the precise location and period of the year, and do not consider a potentially frozen soil. Missing data also had a notable effect on the results, especially when measurements were missing for subsequent time-steps. With reference to Figure 4.6, this effect is particularly evident towards the end of the simulation period. Here, the last two blocks of missing measurements (red shades in the figure) last 30 and 45 minutes respectively, while the water velocity is high and most of its volume returns to MAIN within 2 hours. During this period the true return temperature has an inversion of trend, possibly due to a decline in supply temperature, that might have happened in the missing data. A similar effect can be seen in SEC1 when, on Jan 13, the largest group of 6 consecutive missing 15-minutes measurements happens. Here the water velocity is lower, and the time taken for most of it to traverse the network is between 3 and 4 hours. In fact, the missing data coincides with a point in time were the demand and

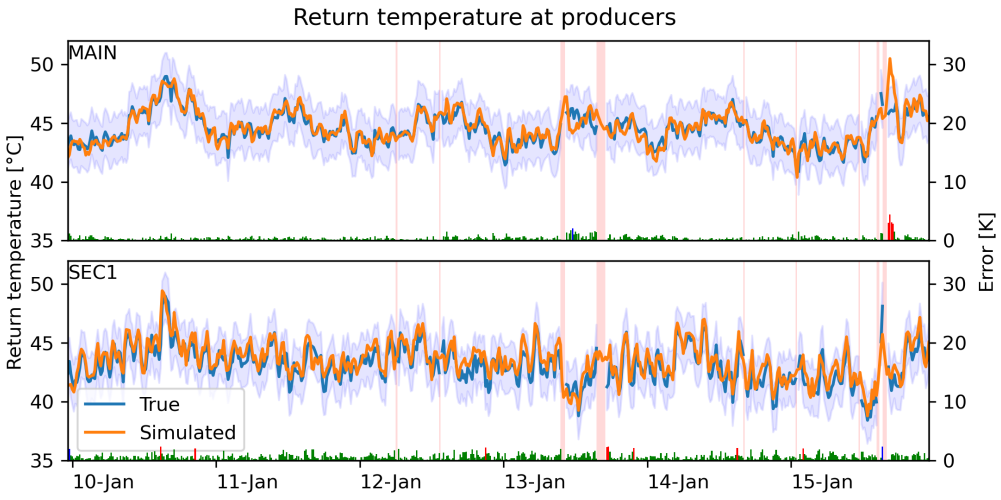



Figure 4.6: Measured and simulated return temperature of two producers with bars showing the absolute error in K. The bars are colored in blue if the error with sign is below −2 °C, red if above 2 °C, and green if in between these values. The blue shaded area around the measurements shows the threshold of ±2 °C, while the red shaded areas show the time-steps where monitoring data was missing.

the supply temperatures are decreasing. As the measurements become available again, the demand and supply temperatures have increased again, and in two immediate time-steps an error higher than the threshold occur. After 4 hours the error increases again for 1 time-step, which might be still related to this large gap in monitoring data. As a mean of comparison with previous works, we also report the error on the supply temperature of the two investigated consumers. For P1, the mean error is 1.41 K, with a standard deviation of 1.34 K. For P2, the mean error is instead 0.05 K with a standard deviation of 1.24 K. These results confirm that even in peripheral substations the model tends on average to underestimate the heat losses, at least for the considered case study. However, these values are well in line with the errors found in other works as reported in Section 2.2.1 and mostly within the tolerance of installed sensors.

Finally, in terms of simulation speed, the implemented strategy for hydraulic simulation was compared with the standard loop method. On the investigated case study, our method was consistently faster, reducing the solving time of the hydraulic part, on average, by 17.5%.

### Hybrid Framework

To validate the hybrid framework, we assess its ability to replicate the thermal dynamics observed in the monitoring data. Here, the hybrid model combines the GNN surrogate for hydraulics (Chapter 3) with the proposed Differentiable Lagrangian model for thermal dynamics. We compare its performance against the previously verified PyDHN baseline.

Table 4.1 summarizes the hybrid framework performance against monitoring data, focusing on inlet temperatures of MAIN, SEC1, P1, and P2. Note that for producers, the inlet corresponds to the network return temperature, whereas for consumers, it corresponds to their supply temperature. Following the recommendations of Johra et al. (2021), we report multiple metrics to capture both accuracy and variability, even though reference measurements are available. In addition to the Mean Absolute Error (MAE), we compute the Normalized Mean Bias Error (NMBE) to quantify systematic bias, and three forms of the Coefficient of Variation of the Root Mean Square Error (CVRMSE): standard, hourly-averaged (HCVRMSE), and daily amplitude (DACVRMSE).

| | **Method** | **NMBE** | **RMSE** | | | **MAE** |
|---|---|---|---|---|---|---|
| | | (≈ 0) | CV (↓) | HCV (↓) | DACV (↓) | (↓) |
| **MAIN** | PyDHN | 0.20 | 1.41 | 1.09 | 34.49 | 0.45 |
| | Hybrid | 0.36 | 1.38 | 1.07 | 35.71 | 0.43 |
| **SEC1** | PyDHN | 0.94 | 2.14 | 1.59 | 11.46 | 0.71 |
| | Hybrid | 0.94 | 2.21 | 1.61 | 13.69 | 0.74 |
| **P1** | PyDHN | 1.79 | 2.55 | 2.17 | 86.91 | 1.59 |
| | Hybrid | 1.85 | 2.59 | 2.20 | 88.85 | 1.63 |
| **P2** | PyDHN | 0.08 | 1.88 | 1.41 | 35.98 | 0.82 |
| | Hybrid | 0.17 | 1.91 | 1.45 | 36.2 | 0.83 |

Table 4.1: Performance comparison between the hybrid method and PyDHN against monitoring data: inlet temperatures.

The two approaches, the fully physics-based PyDHN and the hybrid framework, show similar performance, with error metrics falling within the expected range of modeling and measurement uncertainty. Both models generally have small NMBE, with the proposed approach showing slightly higher values than PyDHN. This may suggest a minor systematic bias, possibly due to the GNN training process or limitations in generalizing across different conditions.

The performance of the proposed surrogate hydraulic model on both synthetic and monitoring data, despite the differing representations of the supply and return lines, suggests promising generalization potential. In this context, we argue that the minor discrepancies between the supply and return lines are too small to indicate any fundamental limitations of the model. Notably, residual errors tend to be higher at extreme flow conditions. A potential strategy to alleviate this issue is to generate more training samples within this problem space. Alternatively, non-converged predictions could serve as initial guesses for iterative solvers, potentially accelerating their convergence.

Based on the results, the proposed hybrid framework shows significant potential for real-time optimization, achieving good agreement with physics-based methods while being approximately 85 times faster in our setup (under 4 minutes for a 12-day simulation with a 60-second

step size). This speed is comparable to similar studies proposing fully ML-driven surrogates and testing on smaller networks (Boussaid et al., 2024), with the added benefits of an accurate hydraulic model for meshed networks and a physics-based thermal model requiring no training. Importantly, our approach can simulate multiple DHN layouts or scenarios simultaneously, allowing efficient design and operational optimization, provided the hydraulic model is trained on all considered layouts.

Its main limitation is flexibility: while the architecture is transferable, further research is needed to determine if a single model can generalize across different network layouts, including minor perturbations as used in design optimization. This will be investigated in Chapter 5.

### 4.5.5 Limitations of the Verification Study

While the verification demonstrates the accuracy and computational efficiency of the proposed models, certain methodological limitations must be acknowledged when interpreting the performance metrics.

A primary constraint arises from the availability and quality of the monitoring data. Missing measurements were imputed using a forward-fill approach which inherently smooths dynamic transients in the network. However, because these missing data points constituted only a small minority of the evaluated time steps, this artificial dampening is expected to have a negligible impact on the overall error statistics.

Another limitation comes from the fact that the pressure correction factors were calibrated and evaluated over the exact same time window. Consequently, the reported pressure metrics represent a bias-corrected assessment rather than a strict out-of-sample validation, which could result in an overestimation of the model's predictive accuracy on unseen data.

Finally, the scope of this validation is confined to a specific two-week winter period on a single, albeit complex, network topology. This narrow focus is a direct consequence of the well-known difficulty in securing reliable, high-resolution monitoring data from operational DHNs. Although this dataset successfully proves the framework's viability for high-demand, cold-weather scenarios, its external validity remains an open question. Assessing how the model translates to different seasons, alternative operational regimes, or entirely distinct network architectures is a necessary next step, contingent upon the availability of broader ground-truth datasets. While these data constraints do not undermine the comparative analysis between the physics-based and hybrid models presented here, they do define the boundary conditions of the current findings and highlight the ongoing need for accessible operational data in future validation efforts.

## 4.6 Application: Gradient-Based Control Optimization

### 4.6.1 Control Strategies

We consider two base heating curves: a constant model and a piecewise linear model. For both, we test variants with preheating, night setback, or both. The piecewise linear model is controlled by 4 (learnable) parameters, $\theta_a, \theta_b, \theta_c, \theta_d$:

$$\theta_{\text{out}} = f_l(\theta_{\text{air}}) = \begin{cases} \theta_c, & \theta_{\text{air}} < \theta_a \\ \theta_c + \frac{(\theta_{\text{air}} - \theta_a)(\theta_d - \theta_c)}{\theta_b - \theta_a}, & \theta_a \leq \theta_{\text{air}} \leq \theta_b \\ \theta_d, & \theta_{\text{air}} > \theta_b \end{cases} \tag{4.25}$$

with the requirements that $\theta_a < \theta_b$ and $\theta_d < \theta_c$. The models with preheating have 3 additional parameters: the starting time $t_0$, the duration $\mathcal{T}$, and the temperature difference $\Delta$. Given $t_1 = t_0 + \mathcal{T}$ and assuming $t_0 < t_1$:

$$\theta_{\text{out}} = f_l(\theta_{\text{air}}) + \Delta^* , \tag{4.26}$$

$$\Delta^* = \Delta \left[ \sigma\left(k\left(t - t_0\right)\right) - \sigma\left(k\left(t - t_1\right)\right) \right] . \tag{4.27}$$

Here $\sigma$ is the sigmoid function and $k = 100$. $\Delta^*$ is used to avoid discontinuity induced by a sudden change of temperature. For daily transitions, $t$ is interpreted over a continuous daily cycle. The same formulation is used for setback, with the temperature delta constrained to be negative.

### 4.6.2 Optimization Approach

Given a control strategy, we seek to find its optimal sets of parameters, for MAIN and SEC1, and compare the performance of the optimized strategy against both the manually controlled supply temperatures used during the monitoring period and a gradient-free baseline. We frame this optimization problem as follows:

$$\min_P \left( \sum_{t=1}^{T} \sum_{i \in \mathscr{E}_p} Q_{\text{loss},i}^{(t)}(P) + \beta \sum_{t=1}^{T} \sum_{j \in \mathscr{E}_{c \setminus r}} \mathscr{Q}_j^{(t)}(P) \right) \tag{4.28}$$

$$\mathscr{Q}_j^{(t)} = \begin{cases} c_{\text{p}} \dot{m}_j^{(t)} \dfrac{\Delta t}{h} \delta_j^{(t)}, & \delta_j^{(t)} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.29}$$

$$\delta_j^{(t)} = \theta_{\text{ret},j}^{\min} - \theta_{\text{ret},j}^{(t)} \tag{4.30}$$

Here, $c_\text{p} = 4180\,\text{J}\,\text{kg}^{-1}\,\text{K}^{-1}$, $\Delta t$ (s) is the time-step size, $\mathscr{E}_p$ represents the set of pipes where energy losses $Q_{\text{loss},i}^{(t)}(P)$ are evaluated, while $\mathscr{E}_{c\setminus r}$ denotes the subset of consumers excluding ramps. The penalty term $\mathscr{Q}_j^{(t)}(P)$ accounts for violations of the minimum return temperature at consumers. The deviation $\delta_j^{(t)}$ (K) measures how much the simulated return temperature $\theta_{\text{ret},j}^{(t)}$ (°C) falls below the reference value $\theta_{\text{ret},j}^{\min}$ (°C), estimated as the minimum return temperature observed during the monitoring period.

Since the simulation model (Section 4.4) and the heating curves (Section 4.6.1) are differentiable, we can adjust the parameters $P$ using gradient descent. At each $k$-th iteration of gradient descent, the full trajectory of $T$ time steps is evaluated with a fixed set of parameters $P^{(k)}$, which is then updated as:

$$P^{(k+1)} = P^{(k)} - \eta \nabla \mathscr{L}(P^{(k)}), \tag{4.31}$$

where $\mathscr{L}$ is the objective function to minimize in Equation (4.28) and $\eta$ is the learning rate. In addition to the $T$ time steps, an initial period of 24 hours is also simulated at each iteration using the conditions of the first time step to initialize the temperatures of water volumes and stationary boundaries.

For each heating curve variant we perform 50 steps of gradient descent with an initial $\eta$ of 0.01. If the loss does not improve after 5 iterations, $\eta$ is reduced by half, up to a minimum of 0.0001. To prevent instabilities from large parameter updates in a single step, the update of temperature-related parameters is clipped to the range [-1, 1]. The $\beta$ factor in Equation (4.28) is chosen as 1000. Further analysis is conducted using the optimal parameter set found for each heating curve.

### 4.6.3 Results and Discussion

Results are shown in Table 4.2, alongside those obtained using a gradient-free method, Differential Evolution (DE). Under manual control, total heat losses amounted to 42 921 kWh, with average supply temperatures of 81.4 °C (MAIN) and 81.7 °C (SEC1), and standard deviations of 2.2 °C and 1.3 °C, respectively. For DE, we used SciPy's Virtanen et al., 2020 implementation with default settings and ran each optimization for 40 minutes, the maximum time our method took across all heating curves. As DE does not require gradient computations, function evaluations are faster, reaching around 210 evaluations per run.

Our gradient-based approach outperformed DE in most cases, although a more rigorous study is needed to assess whether this holds across different initializations and DE settings. The performance gap widens as the number of parameters increases, suggesting that DE struggles

Table 4.2: Performance of each curve optimized with our method and DE. Total heat loss differences relative to manual control (in kWh and %) are reported, along with the mean ($\mu$) and standard deviation ($\sigma$) of supply temperatures, shown as (MAIN, SEC1). Bold values indicate the lowest heat loss for each variant; no bold means manual control performed best. Curve types: *C* (constant), *L* (linear), *p* (preheating), *s* (setback).

| **Curve** | **Loss Diff. (kWh)** | | **Temp. $\mu$ (°C)** | | **Temp. $\sigma$ (°C)** | |
|---|---|---|---|---|---|---|
| | Ours | DE | Ours | DE | Ours | DE |
| *C* | **-90** (-0.2%) | 16 (0.0%) | (81.0, 82.0) | (81.5, 81.1) | (0.0, 0.0) | (0.0, 0.0) |
| *C+p* | **-283** (-0.7%) | 284 (0.7%) | (80.8, 82.5) | (81.7, 82.1) | (2.6, 2.8) | (1.1, 1.1) |
| *C+s* | 427 (1.0%) | 278 (0.6%) | (82.0, 82.6) | (80.6, 86.3) | (2.8, 2.2) | (0.9, 1.5) |
| *C+p+s* | 438 (1.0%) | 1025 (2.4%) | (80.5, 88.0) | (84.0, 79.0) | (2.4, 1.5) | (2.3, 1.4) |
| *L* | 526 (1.2%) | 343 (0.8%) | (82.3, 82.3) | (82.5, 79.6) | (1.1, 1.1) | (0.0, 0.0) |
| *L+p* | 69 (0.2%) | **-39** (-0.1%) | (81.6, 82.1) | (81.4, 80.9) | (2.3, 2.5) | (0.4, 0.7) |
| *L+s* | 105 (0.2%) | 1330 (3.1%) | (81.0, 83.7) | (84.3, 80.8) | (1.0, 0.7) | (1.0, 3.2) |
| *L+p+s* | **-82** (-0.2%) | 1006 (2.3%) | (81.2, 82.9) | (81.8, 87.0) | (3.1, 3.0) | (2.4, 4.1) |

with higher-dimensional parameter spaces within the fixed evaluation budget. The lowest losses were achieved using a simple constant curve with preheating, a strategy that closely mirrors the manual operations observed during the monitoring period and used as boundary conditions. In this configuration, the supply temperature is held at 80.0 °C for MAIN and 81.7 °C for SEC1. preheating is activated at 10 PM for both plants, increasing the temperature by 9 K to 10 K for about 2 hours. This compensates for the decrease in mass flow and replicates the supply temperature rise observed under manual control.

In general, optimized curves consistently initiate preheating around sharp drops in MAIN's mass flow, at 11 AM, 2 PM, or 10 PM, rather than during morning peaks. Setback variants performed poorly, with the optimizer minimizing their effect. These trends suggest that gradient descent effectively navigates the model's solution space, confirming its viability.

## 4.7 Chapter Conclusion

This chapter completed the hybrid differentiable modeling framework introduced in Chapter 3 by developing a fully differentiable Lagrangian thermal model that integrates seamlessly with the GNN-based hydraulic surrogate. The central contribution is a reformulation of the classical Lagrangian scheme that preserves computational graph connectivity throughout the simulation, allowing end-to-end gradient propagation without compromising the speed and numerical advantages inherent to Lagrangian approaches. By optimizing the implementation for GPU execution, the model achieves simulation speeds on par with state-of-the-art ML–based surrogates (89 steps/s), while retaining the benefits of a physics-driven formulation: accuracy, interpretability, robustness to out-of-distribution operating conditions, and full transferability to any DHN without retraining.

We validated both the physics-based framework of Chapter 2 and the hybrid framework using monitoring data from the meshed Verbier DHN. The physics-based model achieved return-temperature errors within the sensors' combined tolerance of ±2 °C, with RMSE values of 0.57 K for MAIN and 0.83 K for SEC1. This establishes it as a reliable ground-truth reference for subsequent analyses. The hybrid framework delivered comparable accuracy while being able to simulate a 12-day horizon at 60-second resolution in under four minutes.

We then demonstrated the practical utility of the differentiable thermal model through the optimization of supply-temperature control curves. Using gradient descent, we identified optimal heating-curve parameters and benchmarked them against both manual operation and a gradient-free optimization approach, again using Verbier as a case study. The gradient-based method achieved heat-loss reductions of up to 0.7% while maintaining consumer comfort constraints. Although this reduction may appear modest, it is notable given that the monitoring data reflect an operational regime in which manual control prioritized mass-flow adjustments while keeping supply temperature nearly constant, a strategy that inherently limits the achievable performance of any control curve solely parameterized by outdoor temperature. Moreover, as parameter dimensionality increased, the gradient-based approach consistently outperformed gradient-free optimization, highlighting the value of differentiability for real-world control design.

The main limitation of the current hybrid framework stems from the topological specificity of the GNN hydraulic surrogate: while the architecture is transferable, the trained parameters are tied to a particular network layout. This is acceptable for operational optimization on an existing system but restrictive for design studies requiring network reconfiguration. The next chapter addresses this constraint by introducing a topology-agnostic GNN formulation capable of generalizing across heterogeneous DHN structures. This extension broadens the applicability of the framework to planning and design tasks and also simplifies model training for control optimization, since the generalizable model can be fine-tuned to a specific case study, thereby reducing both the amount of required training data and the overall training time.

# 5 Towards Generalizable Surrogate Models: Topology-Agnostic Learning

*The content of this chapter is original and published for the first time in this thesis.*

## 5.1 Introduction

The previous chapters introduced novel frameworks for the simulation and optimization of DHNs. Chapter 2 introduced a high-fidelity physics-based model suitable for detailed design, while Chapter 3 proposed a GNN-based surrogate that, coupled with a differentiable, GPU-optimized Lagrangian model (Chapter 4), have the potential to accelerate simulations by orders of magnitude. However, a fundamental limitation persists in this workflow: the surrogate model was designed and validated primarily for a specific, fixed topology.

As outlined in Hypothesis 1 (Section 1.7), the pre-design and planning phases of a DHN lifecycle present a distinct set of challenges. In these early stages, the network topology is not fixed; rather, it is a variable to optimize. To do so, planners need a way to rapidly evaluate numerous potential layouts, or partial modifications such as adding loops, extending branches, or interconnecting existing clusters. Retraining a specialized surrogate for every topological variation is computationally prohibitive and practically unjustified.

This chapter addresses this gap by investigating topology-agnostic surrogate models capable of generalizing to unseen network structures. To this end, we generate a large-scale synthetic dataset of diverse DHN topologies and evaluate two distinct architectures. The first is the GNN of Chapter 3, adapted to a possibly larger distance between overlapping cycles. While this model achieves low average errors, we observe some limitations in its practical deployment: training is computationally expensive, particularly as network depth increases, and the model exhibits high maximum error peaks. This suggests that the model may be relying at least partially on dataset-specific shortcuts.

To overcome these limitations, we propose a novel architecture inspired by the structure of numerical hydraulic solvers. This new approach introduces several inductive biases in the model, such as the separation of static properties from dynamic states and the use of recurrent weight

sharing to mimic iterative convergence. We hypothesize that these architectural constraints will encourage the learning of generalized update rules rather than dataset-specific patterns.

We conclude the chapter by presenting a series of exploratory experiments demonstrating the model's versatility and use-cases.

## 5.2 Contributions

The specific contributions of this chapter are:

- **Procedural Data Generation:** We present a pipeline for generating thousands of hydraulically plausible, synthetic meshed networks to uniformly cover a vast problem space without relying on real-world data availability.
- **Architectural Inductive Biases:** We introduce a modified GNN architecture that embeds physical symmetries directly into the message-passing mechanism, allowing the model to adapt to unseen graph structures efficiently. The resulting model achieves an accuracy of more than 90% on test data, with average residuals of 38 Pa.
- **Zero-Shot Transfer Evaluation:** We test the general model on monitoring data from the real-world Verbier network (introduced in Section 4.5.1), which the model has never seen during training, to assess its out-of-distribution generalization capabilities. Results show promising performance, with average residuals of 52 Pa.

## 5.3 Methodology

This section details the methodology developed to extend the surrogate modeling framework, previously limited to fixed structures, to handle arbitrary network topologies. We begin by formally introducing the general problem formulation (Section 5.3.1) and the specific design and operational bounds chosen to constrain the problem space for this study (Section 5.3.2). Following this definition, we describe the generative framework used to synthesize physically consistent training data (Section 5.3.3) and the workflow to process it, starting from the input-output structure of the considered models (Section 5.3.4) and the novel architecture proposed (Section 5.3.5). Finally, we outline a data augmentation strategy used for the model training (Section 5.3.6).

### 5.3.1 Problem Setup

The fundamental objective remains the same as defined in Chapter 3: to predict the internal cycle flows, $\tilde{m}_\mathrm{l}$, which resolve the hydraulic state of a meshed DHN without iterative numerical solvers. However, the problem domain is significantly expanded.

In the case-specific model of Chapter 3 (Section 3.4.1), the network topology and associated

features were constant, i.e., $\boldsymbol{A}'$, $\boldsymbol{d}'$, $\boldsymbol{\ell}'$ and $\boldsymbol{e}$ did not change across samples, except for minor variations between the supply and return lines. This can be formalized as finding a mapping $f_\theta : \mathscr{X}_{\text{BC}} \rightarrow \mathscr{Y}$ for a fixed graph $\mathscr{G}'$, where $\mathscr{X}_{BC}$ represented the space of variable boundary conditions (imposed by substation loads) and $\mathscr{Y}$ the hydraulic state of the internal loops of the DHN.

Under the generalized formulation proposed in this chapter, the network structure is itself a varying input. The task is then to approximate a universal mapping $F_\theta : (\mathscr{G}', \mathscr{X}_{\text{BC}}) \rightarrow \mathscr{Y}$, capable of operating on any physically plausible meshed network topology drawn from a distribution $\mathscr{D}$, within pre-defined problem space constraints.

Formally, a data sample $S_i$ is no longer just a set of boundary conditions but a tuple encompassing the entire physical system configuration: $S_i = (\boldsymbol{A}'_i, \boldsymbol{X}_i, \boldsymbol{e}_i, \tilde{\boldsymbol{m}}_{\text{l},i})$. Consequently, the surrogate model must learn a significantly larger problem space, while ideally retaining a similar computational speed to make it advantageous over traditional iterative solvers.

### 5.3.2 Problem Space Definition

To train a topology-agnostic surrogate, we must explicitly bound the problem space to ensure the training distribution $\mathscr{D}$ is both tractable for learning and representative of real-world systems. Unlike the fixed-topology model of Chapter 3, the domain now encompasses the combined space of (i) **network topologies**, (ii) **physical characteristics**, and (iii) **operational states**. This represents a significantly larger problem space than what existing approaches have explored.

We define boundaries across these three aspects to envelope the complexity of typical mid-sized networks. In particular, we anchor the problem space to the Verbier network. This provides a concrete reference point with available test data and represents a realistic operational scale for Swiss DHNs. However, we extend beyond Verbier's exact specifications to create a margin for generalization.

Notably, we over-dimension certain constraints, particularly pressure loss limits, to ensure robustness during optimization procedures. In iterative design exploration, a solver may temporarily traverse high-friction or suboptimal states before converging to feasible solutions. A model trained exclusively on nominal operating conditions would fail in these regimes, undermining its utility for planning applications.

A summary of these bounds is provided in Table 5.1.

#### Topological Constraints

The primary challenge in achieving topology-agnostic learning lies in the combinatorial explosion of possible graph structures. The bounds must ensure that degenerate or unrealistic

Table 5.1: Definition of the problem space boundaries, organized by category.

| Category | Parameter | Constraint / Range |
|---|---|---|
| **Topology** | Network Scale | Max. 200 pipes (within internal loops) |
| | Meshing Complexity | Max. 6 overlapping internal loops |
| | Min. Cycle Girth | $N_{\text{girth}} \geq 10$ edges |
| | Cycle Overlap | $\geq 1$ unique edge; $\geq 1$ shared edge |
| **Physical** | Pipe Lengths | 0.5 m – 100 m |
| | Pipe Diameters | DN20 – DN500 (discrete sizes) |
| | Max. Pressure Loss | $\leq 350\,\text{Pa}\,\text{m}^{-1}$* |
| **Operational** | Total Network Demand | $\leq 50\,\text{kg}\,\text{s}^{-1}$ |
| | Min. Branch Flow | $\geq 0.228\,\text{kg}\,\text{s}^{-1}$ |
| | Supply/Demand Structure | 1–3 source nodes; 20–90% sink-only nodes |

*Exceeds typical operational limits ($100-200\,\text{Pa}\,\text{m}^{-1}$) to provide optimization margin.

topologies, which never occur in practice due to spatial, economic, and hydraulic constraints, are avoided. We therefore impose graph-level bounds that reflect the nature of real DHNs:

- **Cycle Girth:** We enforce a minimum cycle length of $N_{\text{girth}} = 10$ edges per loop. This prevents unrealistically tight cycles, such as small polygons or pipes that loop back on themselves within short distances. In practice, DHN loops emerge from the need to ensure supply redundancy across spatially distributed loads; tight loops provide no hydraulic benefit and are economically inefficient.
- **Cycle Overlap:** Each cycle must contain at least $N_{\text{unique}} = 1$ edge that belongs exclusively to that cycle, and at least 1 edge shared with another cycle. These constraints ensure topological validity: loops are neither completely disjoint nor entirely coincident.
- **Network Scale and Complexity:** We limit networks to a maximum of 200 pipes within the meshed portion and at most 6 overlapping internal loops. This captures the range from simple ring topologies to moderately complex meshed structures. Networks exceeding 6 loops typically represent large metropolitan systems or interconnected district clusters, which remain outside the current scope.

#### Physical Constraints

To ensure the surrogate model can generalize to diverse design practices, we use broad intervals for the physical constraints:

- **Pipe Dimensions:** Pipe lengths are sampled between 0.5 m and 100 m. Diameters are restricted to discrete standard nominal sizes (DN20 to DN500), reflecting the fact that

commercial piping is not a continuous variable.

- **Pressure Loss Limits:** We set an upper bound for pressure loss at 350 $\mathrm{Pa\,m^{-1}}$. This value is higher than standard design guidelines (typically 100 $\mathrm{Pa\,m^{-1}}$ to 200 $\mathrm{Pa\,m^{-1}}$ (Jie et al., 2020)) to ensure model robustness during optimization, where the search algorithm may temporarily traverse suboptimal, high-friction regions.

#### Operational Constraints

The operational constraints are chosen to cover a wide range of mid-sized networks:

- **Supply and Demand Nodes:** We set the proportion of sink nodes (branches with only consumers) between 20% and 90%; number of source nodes (branches with both consumers and producers) between 1 and 3.
- **Hydraulic Load Magnitude:** The total mass flow is capped at 50 $\mathrm{kg\,s^{-1}}$. We enforce a lower bound on branch peak flows (0.228 $\mathrm{kg\,s^{-1}}$) to avoid false sinks.

### 5.3.3 Synthetic Data Generation Framework

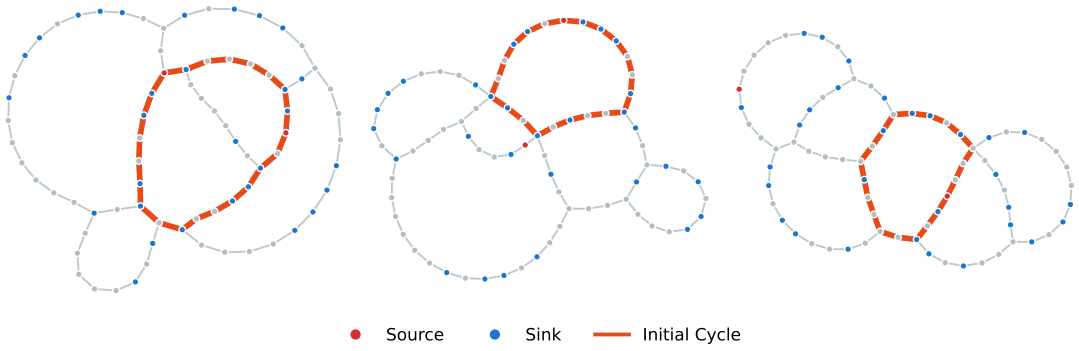



Figure 5.1: Examples of graphs generated with the proposed procedure. All graphs feature 6 cycles and 80 pipes. The initial cycle, around which the rest of the graph is built, is highlighted in red. Randomly selected source and sink nodes, representing the connections to branches, are colored in red and blue respectively.

To train a topology-agnostic model, we require a dataset that reflects the structural diversity of real-world systems. Since real-world DHN data is often proprietary, sparse and limited to specific topologies, we developed a procedural generation pipeline in Python. This pipeline synthesizes hydraulically valid network instances ($\mathcal{G}_i'$) and associated hydraulic states ($\tilde{\boldsymbol{m}}_{\mathrm{l},i}, \dot{\boldsymbol{m}}_{\mathrm{b},i}'$) within the problem space defined in Section 5.3.2.

To ensure homogeneous coverage of this high-dimensional space, we employ a stratified sampling approach across three distinct stages: (1) **topology generation**, (2) **sizing**, and (3)

**operational sampling**. The generation process is parallelized across multiple CPU cores to efficiently handle the computational throughput. Figure 5.1 shows three examples of generated layouts.

### Topology Generation

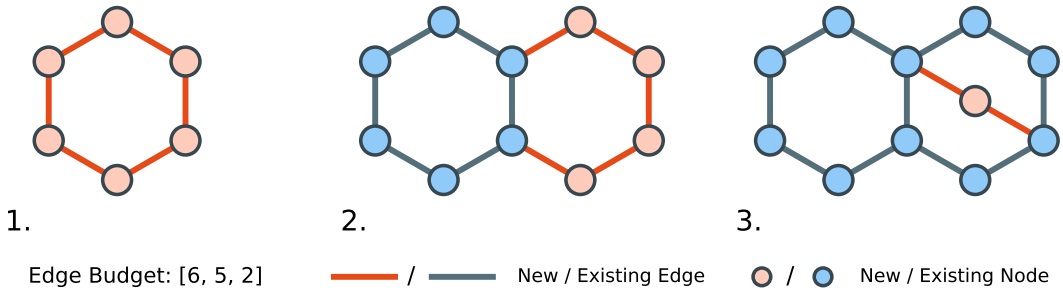



Figure 5.2: Illustration of the topological synthesis procedure. The target graph consists of 3 cycles with an edge budget allocation of $[6,5,2]$. **1.** A base cycle is constructed using the first 6 edges. **2.** A second cycle is formed by inserting a chain of 5 edges into the existing structure. **3.** Similarly, the final cycle is created by connecting two existing nodes with a chain of 2 edges.

The first stage constructs the graph skeleton $\mathscr{G}'$. We employ an incremental construction algorithm designed to mimic the internal loop structure of meshed DHNs, progressively adding overlapping cycles while strictly enforcing planarity and minimum girth constraints. The procedure is illustrated in Figure 5.2.

Given a target number of loops $N_c$ and a global pipe edge budget $N_{\text{pipes}}$, the synthesis proceeds as follows:

1. **Edge Budget Allocation:** The total edge budget is distributed stochastically across the loops. We ensure a baseline complexity by assigning the minimum required edges ($N_{\text{girth}}$) to the first loop and at least $N_{\text{unique}}$ edges to each subsequent loop. The remaining budget is distributed among all loops via a multinomial distribution.

2. **Base Cycle Construction:** The process initializes with a simple fundamental cycle using the allocated edge count for the first loop.

3. **Iterative Expansion:** Subsequent loops are inserted onto the existing structure. The algorithm selects two distinct existing nodes at a minimum distance of $N_{\text{overlap}}$ edges and connects them via a new chain of edges with a length equal to the cycle's allocated budget. This forms a new cycle that overlaps with the current graph.

4. **Constraint Verification:** Before committing an addition, we verify that the new cycle satisfies planarity and minimum girth requirements. If a violation is detected, the candidate addition is rejected, and a new pair of connection nodes is sampled. If valid connection points cannot be found after a set number of retries, the edge allocation is redrawn.

**Sizing**

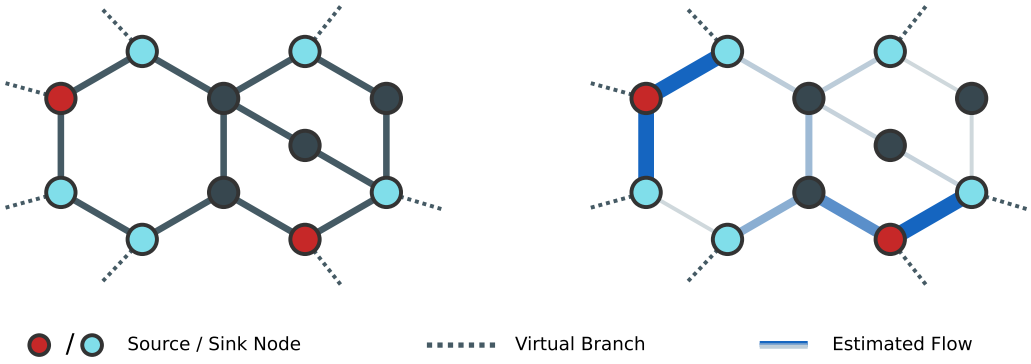



Figure 5.3: Schematic of the pipe sizing step. **Left:** Node assignment. Nodes are designated as sources (red) or sinks (blue), representing interfaces with external branches (dotted lines) where production or consumption occurs. **Right:** Flow estimation heuristic and pipe sizing. Sink demands are propagated back to sources along shortest paths. The resulting aggregated flow intensity (visualized by edge shade) determines the minimum viable diameter for each pipe segment, from which the final diameter is stochastically sampled (visualized by edge thickness).

Once the layout is fixed, the graph needs to be populated with physical attributes, specifically pipe diameters and lengths. The objective of this stage is to produce systems that are hydraulically consistent with real-world constraints while maximizing the diversity of the training distribution. This requires establishing nominal peak loads to size the infrastructure appropriately. The main steps are shown in Figure 5.3.

We first categorize nodes as either sources, sinks, or junctions. In this representation, source nodes denote connections to branches where both producers and consumers are present, whereas sink nodes denote connections to branches containing only consumers. To define the sizing criteria, we sample a total network design peak from a uniform distribution between the theoretical minimum (bounded by branch flow limits) and the global maximum defined in Table 5.1. This total load is distributed among sources and sinks using a log-uniform distribution. This heavy-tailed distribution was chosen to capture the structural heterogeneity of urban networks, which typically feature many small residential substations alongside a

sparse number of high-demand commercial consumers.

Determining appropriate diameters for a meshed network is non-trivial, as flow paths are indeterminate without a full hydraulic solve. To approximate realistic design choices efficiently, we employ a multi-path heuristic:

1. **Flow Estimation:** For every sink, we identify the 3 shortest paths to the nearest source. The demand is distributed across these paths inversely proportional to their length. This mimics the physical tendency of fluid to distribute across different routes, providing a more balanced flow estimate than naive shortest-path routing.
2. **Diameter Selection:** Using the estimated mass flow for each pipe, we select a diameter from the standard discrete set (DN20–DN500). The selection criterion ensures that the pressure drop per meter remains below the maximum threshold ($\Delta p/\ell \leq \Delta p_{max}$). To introduce variance and reflect the common engineering practice of oversizing for future expansion, we do not strictly select the smallest viable diameter. Instead, we randomly sample from the set of all diameters that satisfy the pressure constraint.
3. **Length Selection:** Finally, pipe lengths are assigned uniformly from the defined physical limits (0.5 m–100 m).

#### Operational Sampling and Solution

The final stage generates the training pairs. With the physical network frozen, we generate multiple operational scenarios by varying the boundary conditions relative to the design peaks.

We follow the same logic used in Chapter 3. We utilize LHS to ensure uniform coverage of the operational space. For each sampled boundary condition vector, we solve for the hydraulic state using the loop method of PyDHN introduced in Chapter 2 with a convergence criterion of 1 Pa. To maximize variability, if the simulation does not strictly converge but reaches a maximum absolute residual below 100 Pa, the sample is considered valid and retained in the dataset.

If a generated topology fails to produce valid solutions for any sampled boundary conditions (suggesting an ill-posed physical design), the entire graph is discarded, and the pipeline restarts from the topological generation phase.

### 5.3.4 Input-Output Structure of the Surrogate Models

#### Baseline Architecture

The baseline architecture from Chapter 3, originally designed for fixed-topology operation, treats the condensed graph as a single homogeneous structure where all nodes (both pipes

and cycles) share a unified feature space.

Given a sample $i$, each node $u$ in $\mathcal{G}_i'$ is assigned a feature vector $\boldsymbol{x}_u = [\dot{m}'_{\mathrm{b},u}, d'_u, \ell'_u]^\top$ where pipe nodes receive their actual geometric properties and boundary flows, while cycle nodes are assigned zero-vectors. The complete node feature matrix is $\boldsymbol{X}_i$. A training sample for the baseline model thus consists of $S_i^{\mathrm{base}} = (\boldsymbol{A}'_i, \boldsymbol{X}_i, \boldsymbol{e}_i, \tilde{\boldsymbol{m}}_{\mathrm{l},i})$, where $\tilde{\boldsymbol{m}}_{\mathrm{l},i} \in \mathbb{R}^{N_{\mathrm{c},i}}$ serves as the learning target.

To stabilize training and accelerate convergence, $\boldsymbol{X}_i$ is column-wise scaled to $\boldsymbol{X}_{\mathrm{scaled},i}$ by dividing each feature by its maximum absolute value observed across the entire training set. The model predicts internal cycle flows $\tilde{\boldsymbol{m}}_{\mathrm{l},i}^{\mathrm{pred}}$ directly. These are then used to reconstruct the complete network mass flow distribution via superposition: $\dot{\boldsymbol{m}}_i^{\mathrm{pred}} = \dot{\boldsymbol{m}}_{\mathrm{b},i} + \boldsymbol{B}_1^\top \tilde{\boldsymbol{m}}_{\mathrm{l},i}^{\mathrm{pred}}$.

While this formulation proved effective for the fixed-topology case study in Chapter 3, it treats all features uniformly within a single vector. When extending to topology-agnostic learning, explicitly distinguishing between pipe and cycle nodes, and separating time-invariant network properties (diameter, length) from time-variant operational states (boundary flows) may provide a useful inductive bias, potentially facilitating the model's ability to transfer learned patterns across different network configurations.

### Proposed Architecture

Unlike the baseline's homogeneous formulation, we treat $\mathcal{G}_i'$ as a heterogeneous graph where pipe and cycle node representations are explicitly separated. Furthermore, we move beyond a simple bipartite structure by introducing a set of inter-cycle edges. The goal is to provide cycles with direct information about their topological neighbors and to create a shortcut to propagate signals across nodes more efficiently. In particular, we construct a cycle-to-cycle graph where two cycle nodes are connected if they share at least one common pipe. This adjacency is derived from the cycle matrix $\boldsymbol{B}'_{\mathrm{l},i}$ through the matrix product: $\boldsymbol{A}_{\mathrm{cc},i} = \mathrm{sgn}\left(|\boldsymbol{B}'_{\mathrm{l},i}| \cdot |\boldsymbol{B}'_{\mathrm{l},i}|^\top\right) - \boldsymbol{I}$. We modify the data pipeline to support gradient-based design optimization applications (e.g., inverse design). Instead of feeding pre-scaled feature matrices to the model, we pass **raw physical attributes**. This ensures that the mapping from design variables, such as diameter, to the input features remains part of the differentiable computational graph (see Section 5.3.5). Pre-computation is restricted solely to discrete topological properties which are non-differentiable with respect to continuous design parameters. In particular, for each pipe $u$, we pre-compute the **node degree** $\delta_{p,u}$. Similarly, for a cycle node $v$, we pre-compute its **girth** (number of constituent pipes) $\delta_{\mathrm{g},v}$ and **inter-cycle degree** $\delta_{\mathrm{cc},v}$.

Accordingly, the input sample $S_i^{\mathrm{prop}}$ consists of raw attribute vectors and discrete adjacency matrices rather than processed feature tensors:

$$S_i^{\mathrm{prop}} = \left(\boldsymbol{A}'_i, \boldsymbol{A}_{\mathrm{cc},i}, \boldsymbol{d}'_i, \boldsymbol{\ell}'_i, \dot{\boldsymbol{m}}'_{\mathrm{b},i}, \boldsymbol{\delta}_{\mathrm{p},i}, \boldsymbol{\delta}_{\mathrm{g},i}, \boldsymbol{\delta}_{\mathrm{cc},i}, \boldsymbol{e}_i, \tilde{\boldsymbol{m}}_{\mathrm{l},i}\right) \tag{5.1}$$

### 5.3.5 Solver-Inspired Architecture

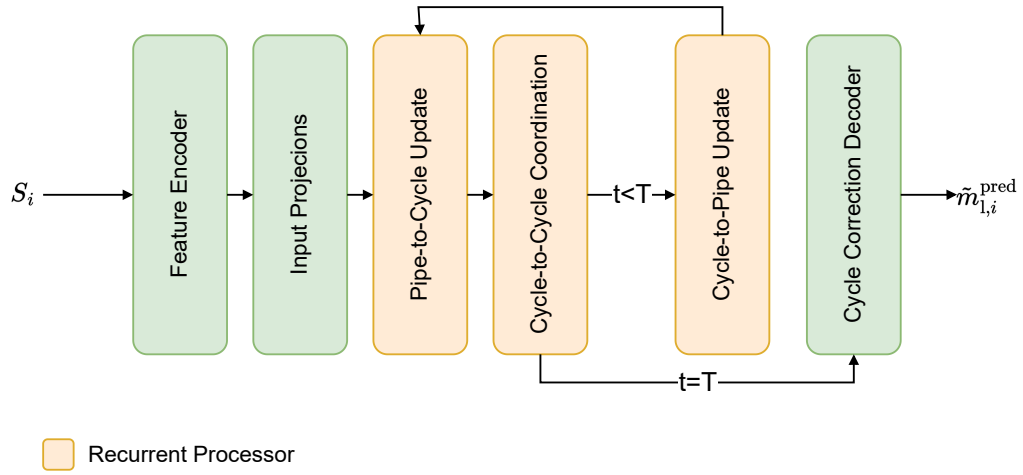



Figure 5.4: High-level overview of the recurrent Solver-Inspired architecture.

We seek to reduce the hypothesis space by constraining the model through physics-inspired inductive biases. Consistent with this effort, the surrogate model architecture is designed to explicitly reflect the separation between the time-invariant network structure and the time-variant hydraulic state described in Section 3.4.1. Standard GNNs typically merge all node features into a single hidden representation vector. However, this conflates the problem definition (geometry) with the solution (flow). Analogous to iterative numerical solvers, where geometric properties remain fixed while mass flows update at each step, we propose a *Solver-Inspired* architecture, of which a high-level overview is given in Figure 5.4. This scheme maintains two distinct representations for each node: a **static context**, which encodes fixed constraints, and a **dynamic state**, which evolves to represent the hydraulic solution.

The architecture is organized into four sequential processing stages, which serve as a guide for the remainder of this section:

1. **Differentiable Feature Encoding:** Raw inputs are pre-processed into feature vectors, maintaining differentiability with respect to design variables.
2. **Input Projections:** These features are projected into high-dimensional latent embeddings, initializing the static contexts and dynamic states.
3. **Recurrent Processor:** The core reasoning phase, which iterates over $T$ discrete solving steps ($t = 1 \dots T$). To mimic a convergent solver, all $t$ iterations share the same weights (recurrent architecture). Within each step, information flows through three sub-blocks:
    - *Pipe-to-Cycle Aggregation Block:* The current states of the pipes are aggregated to compute a residual signal for each loop.
    - *Cycle-to-Cycle Coordination Block:* Cycles exchange information with their neighbors to coordinate their local updates based on shared resistance.

- *Cycle-to-Pipe Feedback Block:* The updated loop states are broadcast back to the constituent pipes to refine the flow estimates for the next iteration.

Crucially, for the final iteration $t = T$, the feedback step is skipped to pass the cycle state directly to a decoder.

4. **Cycle Correction Decoder:** The final cycle states are mapped to scalar flow corrections.

Note that we adopt a slightly different convention than the one used in Chapter 3, where each block $t$ defined a single message-passing direction. Here, within each step $t$, the information flows through all the three distinct blocks of the recurrent processor.

The following subsections detail the specific mechanics of these stages.

Unless otherwise specified, all MLPs in the architecture share the same structure: `Linear` → `PReLU` → `Linear`. We adopt the `PReLU` activation (He et al., 2015) because the architecture acts as a residual solver, where error signals are signed quantities that must propagate in both positive and negative directions.

#### Differentiable Feature Encoding

To facilitate tasks such as the inverse design of pipe diameters, we implement feature engineering as a differentiable layer within the model's forward pass. For a given input sample, the model first computes the **geometric resistance factor** $\zeta_u$ for each pipe $u$ using raw physical units, and the aggregate **cycle stiffness** $\tilde{\zeta}_v$ for each cycle $v$. The goal is to analytically capture the sensitivity of the cycle to changes in mass flow. As the value of cycle stiffness can span multiple orders of magnitude, we employ log-scaling for numerical stability:

$$\zeta_u = \frac{\ell'_u}{{d'_u}^5}, \qquad \tilde{\zeta}_v = \ln\left(1 + \sum_{k \in \mathcal{N}(v)} \zeta_k\right) \tag{5.2}$$

Using these derived quantities, the model constructs three distinct feature vectors, separating static from dynamic information. We denote the maximum value of a feature $z$ over the training dataset as $\max_{\text{gl}}(z)$ and its maximum value within the current graph sample as $\max_{\text{gr}}(z)$.

**1. Pipe static feature vector ($\boldsymbol{x}_{\text{ps}}$):**

$$\boldsymbol{x}_{\text{ps},u} = \left[ \frac{d'_u}{\max_{\text{gl}}(d')}, \quad \frac{\ell'_u}{\max_{\text{gr}}(\ell')}, \quad \frac{\delta_{\text{p},u}}{\max_{\text{gl}}(\delta_{\text{p}})}, \quad \frac{\delta_{\text{p},u}}{\max_{\text{gr}}(\delta_{\text{p}})} \right]^\top \tag{5.3}$$

Note that we scale $\ell'_u$ by the graph maximum. This takes advantage of the fact that pressure loss scales linearly with the pipe length.

**2. Pipe dynamic feature vector ($x_{\mathrm{pd}}$):**

$$\boldsymbol{x}_{\mathrm{pd},u} = \left[ \frac{|\dot{m}'_{\mathrm{b},u}|}{\max_{\mathrm{gr}}(|\dot{m}'_{\mathrm{b}}|)}, \quad \frac{\max_{\mathrm{gr}}(|\dot{m}'_{\mathrm{b}}|)}{\max_{\mathrm{gl}}(|\dot{m}'_{\mathrm{b}}|)} \right]^{\top} \tag{5.4}$$

Here, we decompose the boundary mass flow into a *shape* and a *scale*. We utilize the absolute magnitude $|\dot{m}'_{\mathrm{b},u}|$. This exploits the symmetry of hydraulic friction, where a flow of $+\dot{m}$ generates the same magnitude of pressure loss as $-\dot{m}$. By restricting the feature space to positive values, we relieve the neural network from learning this symmetry. However, to preserve the physical consistency of the loop orientation relative to the absolute flow magnitudes, we must align the edge orientations with the new directions of the boundary conditions. We therefore modify the orientation scalars $e_{uv}$ such that:

$$e_{uv} \leftarrow e_{uv} \cdot \mathrm{sgn}(\dot{m}'_{\mathrm{b},u}) \tag{5.5}$$

This ensures that the cycle corrections predicted by the model are consistent with the original coordinate system without requiring inverse transformations at the output stage.

**3. Cycle static feature vector ($x_{\mathrm{cs}}$):**

$$\boldsymbol{x}_{\mathrm{cs},v} = \left[ \frac{\tilde{\zeta}_v}{\max_{\mathrm{gl}}(\tilde{\zeta}_v)}, \quad \frac{\tilde{\zeta}_v}{\max_{\mathrm{gr}}(\tilde{\zeta}_v)}, \quad \frac{\delta_{\mathrm{g},v}}{\max_{\mathrm{gl}}(\delta_{\mathrm{g}})}, \quad \frac{\delta_{\mathrm{g},v}}{\max_{\mathrm{gr}}(\delta_{\mathrm{g}})}, \quad \frac{\delta_{\mathrm{cc},v}}{\max_{\mathrm{gl}}(\delta_{\mathrm{cc}})}, \quad \frac{\delta_{\mathrm{cc},v}}{\max_{\mathrm{gr}}(\delta_{\mathrm{cc}})} \right]^{\top} \tag{5.6}$$

Finally, to bias the message passing aggregation, we assign scalar weights to the graph edges that reflect hydraulic importance. For the bipartite pipe-to-cycle graph, the edge weight $w_{uv}$ represents the fractional contribution of pipe $u$ to the total resistance of cycle $v$:

$$w_{uv} = \frac{\zeta_u}{\sum_{k \in \mathcal{N}(v)} \zeta_k} \tag{5.7}$$

Crucially, these weights sum to 1 for every cycle ($\sum_u w_{uv} = 1$), effectively normalizing the loop's resistance distribution.

For the cycle-to-cycle interaction, we aim to capture an **interaction coefficient** between loops. Conversely, if two cycles $i$ and $j$ share a set of pipes $\mathcal{S}_{ij} = \mathcal{N}(i) \cap \mathcal{N}(j)$, the influence of neighbor $j$ on target $i$ is proportional to the stiffness of the shared interface relative to the target's total stiffness. This allows us to define the inter-cycle weights $\omega_{j \to i}$ analytically as the sum of the constituent pipe weights:

$$\omega_{j \to i} = \sum_{u \in \mathcal{S}_{ij}} w_{ui} \tag{5.8}$$

This formulation results in a **directed, asymmetric weighted graph** (hence, we use $\to$ to denote weight asymmetry). For example, if a small, resistive loop $i$ shares a pipe with a

large, low-resistance loop $j$, the coupling $\omega_{j\to i}$ will be large (the shared pipe dominates $i$'s behavior), while $\omega_{i\to j}$ will be small ($j$ is indifferent to $i$). Mathematically, this corresponds to a weighted one-mode projection of the bipartite graph, efficiently computable via sparse matrix multiplication.

### Input Projections

Pipe nodes are initialized with both static and dynamic embeddings using separate MLPs, while cycle nodes are initialized with a static topological embedding only, as their internal flow is initially zero:

$$\begin{aligned} \boldsymbol{s}_{\mathrm{p},u} &= \mathrm{MLP}_{\mathrm{ps}}(\boldsymbol{x}_{\mathrm{ps},u}), & \boldsymbol{z}^{(0)}_{\mathrm{p},u} &= \mathrm{MLP}_{\mathrm{pd}}(\boldsymbol{x}_{\mathrm{pd},u}) & \forall u \in \mathcal{V}'_{\mathrm{p}} \\ \boldsymbol{s}_{\mathrm{c},v} &= \mathrm{MLP}_{\mathrm{cs}}(\boldsymbol{x}_{\mathrm{cs},v}), & \boldsymbol{z}^{(0)}_{\mathrm{c},v} &= \mathbf{0} & \forall v \in \mathcal{V}'_{\mathrm{c}} \end{aligned} \tag{5.9}$$

The static contexts for pipes $\boldsymbol{s}_{\mathrm{p},u}$ and cycles $\boldsymbol{s}_{\mathrm{c},v}$ remain fixed throughout the network layers, that will only update the corresponding dynamic state, $\boldsymbol{z}^{(t)}_{\mathrm{pd},u}$ and $\boldsymbol{z}^{(t)}_{\mathrm{c},v}$.

### Pipe-to-Cycle Update

We update the cycle dynamic state through a custom message passing block designed to approximate the accumulation of head loss residuals around a loop. At iteration $t > 0$, for every pipe $u$ connected to cycle $v$, we first compute a message $\boldsymbol{m}^{(t)}_{u\to v}$ as:

$$\boldsymbol{m}^{(t)}_{u\to v} = \mathrm{MLP}_{\mathrm{fric}}\left(\left[\boldsymbol{z}^{(t-1)}_{\mathrm{p},u} \parallel \boldsymbol{s}_{\mathrm{p},u}\right]\right) \cdot w_{uv} \cdot e_{uv}, \tag{5.10}$$

where $[\cdot \parallel \cdot]$ indicates concatenation. Here, multiplying by $e_{uv}$ enforces a symmetry with respect to the relative pipe-cycle orientation. Then, the messages are aggregated by summation to enforce the additive nature of pressure drops in a closed loop:

$$\bar{\boldsymbol{m}}^{(t)}_{v} = \sum_{u\in\mathcal{N}(v)} \boldsymbol{m}^{(t)}_{u\to v}\,. \tag{5.11}$$

The result is a directed weighted sum of contributions explicitly mimicking the calculation of cycle residuals. Finally, the cycle dynamic state is updated using a gated residual mechanism. The aggregated residual $\bar{\boldsymbol{m}}^{(t)}_{v}$ is concatenated with the current cycle state to compute a candidate update vector $\boldsymbol{\Delta z}^{(t)}_{v}$:

$$\boldsymbol{\Delta z}^{(t)}_{v} = \mathrm{MLP}_{\mathrm{up}}\left(\left[\bar{\boldsymbol{m}}^{(t)}_{v} \parallel \boldsymbol{z}^{(t-1)}_{\mathrm{c},v}\right]\right). \tag{5.12}$$

Despite the weights summing to 1, the magnitude of the message sum can fluctuate significantly. To stabilize the subsequent processing, $\mathrm{MLP}_{\mathrm{up}}$ includes a **Layer Normalization** (LN) (Ba et al., 2016) step immediately after the first linear projection, ensuring the activation distribution remains consistent. To ensure stability, this update is gated by the cycle's static

context $\boldsymbol{s}_{\mathrm{c},v}$, allowing rigid loops to dampen their reaction to residuals via the gate

$$\boldsymbol{g}_v^{(t)} = \sigma\left(\mathrm{MLP}_{\mathrm{gate}}(\boldsymbol{s}_{\mathrm{c},v})\right), \tag{5.13}$$

where $\sigma$ is the sigmoid function. The final update is applied with a learnable step size $\alpha$:

$$\tilde{\boldsymbol{z}}_{\mathrm{c},v}^{(t)} = \boldsymbol{z}_{\mathrm{c},v}^{(t-1)} + \alpha \cdot \mathrm{LN}\left(\boldsymbol{\Delta z}_v^{(t)} \odot \boldsymbol{g}_v^{(t)}\right). \tag{5.14}$$

Note the application of Layer Normalization on the gated update vector. In deep recurrent architectures, repeated additions can lead to signal explosion. By normalizing the update vector before scaling it with $\alpha$, we effectively enforce a soft update, ensuring that no single iteration step is excessively large.

### Cycle-to-Cycle Coordination

Once cycle nodes are updated according to the messages from constituent pipes, we employ a coordination block operating on the cycle-to-cycle graph. Structurally, this block follows the design principles of the GENConv architecture, shifting the non-linear computation from the message generation to the node update phase.

First, we sum the states of neighboring cycles $\mathcal{N}_{\mathrm{cc}}(v)$, weighted by their coupling strength (shared stiffness) with the receiving node:

$$\tilde{\boldsymbol{m}}_{\mathrm{cc},v}^{(t)} = \sum_{k \in \mathcal{N}_{\mathrm{cc}}(v)} \tilde{\boldsymbol{z}}_{\mathrm{c},k}^{(t)} \cdot \omega_{k \to v}. \tag{5.15}$$

Then, we concatenate the neighborhood signal $\tilde{\boldsymbol{m}}_{\mathrm{cc},v}^{(t)}$, the current local state $\tilde{\boldsymbol{z}}_{\mathrm{c},v}^{(t)}$, and the static stiffness context $\boldsymbol{s}_{\mathrm{c},v}$. This combined vector is passed through an MLP to compute the state update:

$$\boldsymbol{z}_{\mathrm{c},v}^{(t)} = \tilde{\boldsymbol{z}}_{\mathrm{c},v}^{(t)} + \mathrm{MLP}_{\mathrm{coord}}\left(\left[\tilde{\boldsymbol{m}}_{\mathrm{cc},v}^{(t)} \,\|\, \tilde{\boldsymbol{z}}_{\mathrm{c},v}^{(t)} \,\|\, \boldsymbol{s}_{\mathrm{c},v}\right]\right). \tag{5.16}$$

By delaying the non-linear transformation until after aggregation, the model is encouraged to learn complex conditional logic for the update. The network can weigh the aggregated neighbor influence against the local stiffness context, effectively coordinating the flow correction based on the aggregated perturbation expected from the overlapping loops.

### Cycle-to-Pipe Update

For all intermediate iterations $t < T$, the updated cycle representations must be communicated back to the pipes to refine their dynamic state for the next iteration.

For a pipe $u$, the incoming message is the updated state of its connected cycles $v \in \mathcal{N}(u)$, multiplied by their relative direction $e_{uv}$. Unlike the pipe-to-cycle step, we do not weight this

message, as the cycle correction should apply uniformly to all its constituent pipes.

$$\tilde{\boldsymbol{m}}_{v\to u}^{(t)} = \sum_{v\in\mathcal{N}(u)} \boldsymbol{z}_{\mathrm{c},v}^{(t)} \cdot e_{uv}\,. \tag{5.17}$$

The pipe dynamic state is then updated. We employ the same structure as the cycle-to-cycle update, concatenating the aggregated correction signal $\tilde{\boldsymbol{m}}_{v\to u}^{(t)}$ with the previous pipe state $\boldsymbol{z}_{\mathrm{p},u}^{(t-1)}$:

$$\boldsymbol{z}_{\mathrm{p},u}^{(t)} = \boldsymbol{z}_{\mathrm{p},u}^{(t-1)} + \mathrm{MLP}_{\mathrm{br}}\left(\left[\tilde{\boldsymbol{m}}_{v\to u}^{(t)} \,\|\, \boldsymbol{z}_{\mathrm{p},u}^{(t-1)}\right]\right). \tag{5.18}$$

Note that unlike the cycle update, this step does not utilize a static gating context. This reflects the hierarchy of the model, according to which the complex decision-making should be handled at the cycle level, where stiffness is contextualized. Pipes apply the resolved corrections to their internal state, preparing the feedback for the next residual evaluation in iteration $t+1$.

#### Cycle Correction Decoder

After $T$ iterations are completed, a decoder block maps the final cycle state $\boldsymbol{z}_{\mathrm{c},v}^{(T)}$ to a predicted cycle flow correction $\tilde{m}_v^{\mathrm{pred}}$. This decoder is a simple MLP:

$$\tilde{m}_v^{\mathrm{pred}} = \mathrm{MLP}_{\mathrm{dec}}\left(\boldsymbol{z}_{\mathrm{c},v}^{(T)}\right) \cdot \max_{\mathrm{gr}}(|\dot{m}'_{\mathrm{b}}|). \tag{5.19}$$

The output of this MLP is finally scaled by the maximum absolute boundary flow value, to bias the model towards predictions that are within the specific case's order of magnitude.

#### Loss Function

We use the the composite loss function described in Section 3.4.7 for both the baseline and Solver-Inspired model.

### 5.3.6 Data Augmentation

A fundamental property of the cycle-based formulation is that the definition of the positive flow direction for any given internal loop is arbitrary. Whether a cycle is defined as clockwise or counter-clockwise affects the sign of the values in the incidence matrix $\boldsymbol{B}_{\mathrm{l}}$ (and consequently the edge features $\boldsymbol{e}$) and the sign of the resulting cycle mass flow $\tilde{m}_{\mathrm{l}}$, but it does not alter the final pipe mass flows.

Mathematically, if the orientation of a cycle node $v$ is reversed, the corresponding row in $\boldsymbol{B}_{\mathrm{l}}$ changes sign, and the scalar cycle flow becomes $-\tilde{m}_{\mathrm{l},v}$. Since the physical pipe flow is reconstructed via $\dot{\boldsymbol{m}}_{\mathrm{p}} = \boldsymbol{B}_{\mathrm{l}}^{\top}\tilde{\boldsymbol{m}}_{\mathrm{l}}$, the two negations cancel out, preserving the physical state.

To ensure the surrogate model learns this symmetry and does not overfit to the specific orientation produced by the graph construction algorithm, we introduce a physics-informed data augmentation strategy.

During training, we apply a stochastic transformation $\mathcal{T}_{\text{flip}}$ to each sample in the batch. For every internal cycle node $v$ in the condensed graph, we sample a Bernoulli random variable $b_v \sim \text{Bernoulli}(p)$, with $p = 0.5$. If $b_v = 1$, we invert the orientation of the cycle by simultaneously flipping the signs of its target variable and its incident edge features:

$$\tilde{m}_{\text{l},v} \leftarrow -\tilde{m}_{\text{l},v} \quad e_{uv} \leftarrow -e_{uv} \quad \forall u \in \mathcal{N}(v) \tag{5.20}$$

where $\mathcal{N}(v)$ represents the set of pipe nodes connected to cycle $v$. This augmentation effectively doubles the diversity of topological representations seen by the network, enforcing equivariance to cycle direction and improving the model's generalization robustness.

## 5.4 Data, Model Configuration, and Evaluation Protocol

This section details the experimental framework used to evaluate the proposed topology-agnostic approach. We first detail the composition of the synthetic training, validation, and testing datasets created according to the procedure previously presented, focusing on the sampling strategies adopted to handle the combinatorial complexity of the topological problem space (Section 5.4.1). Subsequently, we outline the model architectures compared in this study, including baselines and the proposed Solver-Inspired framework, along with the training protocol and hyperparameters (Section 5.4.2). We then define the evaluation protocol, which assesses performance across two distinct regimes: in-distribution generalization and zero-shot transfer to real-world infrastructure (Section 5.4.3). Finally, we provide relevant implementation details (Section 5.4.4).

### 5.4.1 Dataset Composition

To ensure unbiased evaluation, we utilize three distinct synthetic datasets for training, validation, and testing. These sets are strictly disjoint. For higher-complexity classes, this disjointness extends to the topological structure (adjacency), ensuring the model encounters unseen graph connectivities. For lower-complexity classes where the space of non-isomorphic topologies is finite and potentially saturated (e.g., single-loop networks), disjointness is maintained through unique combinations of pipe lengths and diameters. Each dataset includes samples for all possible combinations of cycle and pipe counts defined by the bounds established in Section 5.3.2.

#### Training Set

The training dataset is constructed by generating all valid combination of pipe and cycle counts using a complexity-weighted sampling strategy. This is necessary to mitigate the fact that the space of valid graph structures expands exponentially with the number of internal loops ($N_c$). In particular, we follow the schedule detailed in Table 5.2, where the number of unique topologies for each combination of pipes and cycle counts increases with the cycle count. For each unique topology, we include 5 distinct operational scenarios (boundary conditions). The low number of operational conditions sampled for each topology should encourage the model to learn universal patterns that are invariant to geometry, rather than memorizing graph-specific shortcuts. As a result, the training dataset is comprised of 1 176 600 samples.

Table 5.2: Composition of the synthetic training dataset. Complex topologies are oversampled to ensure the model learns robust update rules for high-interference scenarios.

| **Internal Loops** ($N_c$) | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **Unique Topologies** | 20 | 100 | 200 | 300 | 400 | 500 |

#### Validation and Test Sets

In contrast to the training set, the validation and test sets follow a uniform distribution across complexity classes. This is to ensure a balanced benchmark, where performance on simpler networks is weighted equally to performance on complex ones, preventing aggregated metrics from being skewed by the performance on high-complexity topologies. Specifically, we generate 20 unique graphs for every valid combination of pipe and cycle counts. As with the training set, we sample 5 operational scenarios for each graph. This results in a total of 99 600 samples for each dataset.

### 5.4.2 Model Configurations

To assess the improvements brought by the proposed Solver-Inspired architecture we evaluate three distinct model configurations. All models use a hidden dimensionality of 256 channels for node embeddings.

- **ELU-2-7l:** The reference architecture from Chapter 3, composed of two stages of 7 message-passing layers each. This baseline assesses whether the original fixed-topology architecture can be naively retrained to handle varying graphs.
- **ELU-2-11l:** An extension of the baseline architecture to two stages of 11 layers each. Since the generalized dataset contains networks with a larger maximum inter-cycle distance, this variant tests whether performance gains are simply a function of increased

receptive field depth.

- **Solver-Inspired GNN:** The proposed Solver-Inspired architecture described in Section 5.3.5. It is configured with a depth of $T = 6$ recurrent blocks. Since each block contains both a pipe-to-cycle and cycle-to-pipe update, this corresponds to a depth comparable to the 11 layers used in the ELU-2-11l baseline, if inter-cycle message passing is not considered.

All models are trained using the AdamW optimizer with a weight decay of $10^{-2}$ to reduce overfitting. Due to the larger training dataset size compared to Chapter 3, and to maintain reasonable training times, we use a batch size of 256. Accordingly, the learning rate is initialized at 0.0002 and decayed by a factor of 10 every 10 epochs. To account for variability due to initialization, each configuration is run 10 times with different random seeds.

### 5.4.3 Evaluation Protocol

We evaluate the framework using the metrics defined in Section 3.5.4 (Accuracy, $\text{MAE}_\text{m}$, $\text{Res}_\text{c}$, $\text{Res}_\text{m}$). The best model realization according to the accuracy on the validation set is further investigated on test data across two distinct evaluation regimes:

1. **In-Distribution (Synthetic):** We assess performance on the synthetic test set described in Section 5.4.1. This measures the model's ability to generalize to unseen topologies drawn from the same distribution as the training data.
2. **Zero-Shot Transfer (Monitoring Data):** We evaluate the model directly on the monitoring data from the Verbier network (introduced in Section 4.5.1) without any fine-tuning. This serves as an out-of-distribution test, as the real-world topology may contain structural patterns not perfectly captured by the synthetic data generator.

### 5.4.4 Implementation Details

All experiments were conducted on a system equipped with an NVIDIA GeForce RTX 3090 and 32 GB of RAM. The models were implemented in PyTorch 2.1.2 and PyTorch Geometric 2.5.0.

## 5.5 Results and Discussion

This section evaluates the proposed approach on both synthetic and real-world DHNs. We first assess generalization on synthetic topologies (Section 5.5.1) by comparing three architectural variants, selecting the best-performing model based on the accuracy metric, and reporting its performance on the synthetic test set. We then assess zero-shot transfer on monitoring data from the Verbier DHN (Section 5.5.2), a network not observed during training and not generated by the synthetic topology generator, and therefore potentially out of distribution,

to demonstrate the model's ability to generalize to unseen operational systems without fine-tuning. Finally, we analyze training and inference efficiency (Section 5.5.3) to evaluate practical feasibility.

### 5.5.1 In-Distribution Generalization

#### Validation Set Performance and Model Selection

We begin by comparing the three model configurations, ELU-2-7l, ELU-2-11l, and the proposed Solver-Inspired GNN, on the new synthetic datasets. To assess the stability of the architectures, we first report the performance distribution on the validation set across 10 random initialization seeds. Table 5.3 and Table 5.4 summarize these metrics.

Table 5.3: Validation set performance statistics (10 seeds). We report **Mean** ± **Std** and the **(Best)** observed value across seeds. **Acc**: Percentage of samples with residuals < 100 Pa. $\mathbf{MAE_m}$: Mean Absolute Error on pipe mass flow.

| | **Acc** [%] (↑) | | $\mathbf{MAE_m}$ [$\mathrm{kg\,s^{-1}}$] (↓) | |
|---|---|---|---|---|
| **Model** | **Mean ± Std** | **(Best)** | **Mean ± Std** | **(Best)** |
| ELU-2-7l | 63.48 ± 3.92 | (65.58) | 0.0172 ± 0.0002 | (0.0168) |
| ELU-2-11l | 75.47 ± 0.90 | (76.99) | 0.0106 ± 0.0003 | (0.0102) |
| Solver-Inspired GNN | **90.79 ± 0.46** | **(91.52)** | **0.0059 ± 0.0002** | **(0.0057)** |

As shown in Table 5.3, the Solver-Inspired GNN significantly outperforms the baselines. It achieves a mean accuracy of 90.79% with a low standard deviation of ±0.46%, indicating that the architecture is robust to initialization. In contrast, the shallow ELU-2-7l model shows high variance (±3.92%) and poor accuracy (63.48%), suggesting it struggles to converge to a stable solution on this complex dataset. This lower performance was partly expected due to the limited number of message-passing operations, which restricts the model's receptive field on DHNs with distant loop interactions. While the deeper ELU-2-11l improves accuracy to 75.47%, it still lags behind the proposed method by over 15 percentage points.

Table 5.4 further highlights the performance improvement brought by the Solver-Inspired structure, as well as the higher consistency. While all model exhibit extreme maximum residuals ($Res_m$), potentially due to fringe cases in the dataset, the baseline models exceed 1.9 MPa on average, with best-case scenarios still near 1 MPa. The Solver-Inspired GNN reduces this maximum error by an order of magnitude (mean 165 kPa) and brings the average cycle residual ($Res_c$) down to just 34.6 Pa, validating the effectiveness of strong inductive biases. Further analysis is conducted on the model realization that achieved the highest accuracy for each architecture.

Table 5.4: Validation set hydraulic consistency. **$\mathbf{Res_c}$**: Average cycle pressure residual. **$\mathbf{Res_m}$**: Maximum pressure residual observed across the entire dataset. Note that for $\mathrm{Res_m}$ values are given in kPa. Mean and standard deviation are taken across random seeds.

| | **$\mathbf{Res_c}$** [Pa] (↓) | | **$\mathbf{Res_m}$** [kPa] (↓) | |
|---|---|---|---|---|
| **Model** | **Mean ± Std** | **(Best)** | **Mean ± Std** | **(Best)** |
| ELU-2-7l | 182.9 ± 23.7 | (160.3) | 1954.6 ± 745.5 | (954.4) |
| ELU-2-11l | 125.1 ± 11.8 | (111.8) | 1900.9 ± 683.0 | (929.0) |
| Solver-Inspired GNN | **34.6 ± 1.4** | **(32.1)** | **165.4 ± 44.5** | **(125.9)** |

**Test Set Performance**

After selecting the best-performing Solver-Inspired GNN realization based on validation accuracy, we evaluate its generalization performance on the test set. In contrast to the validation results, which aggregate statistics across multiple random seeds, test-set metrics are computed over the dataset for a single trained model. As a result, the reported standard deviations reflect the intrinsic variability of the dataset rather than sensitivity to initialization.

Table 5.5: Test set metrics of the selected best-performance model. **Acc**: Percentage of samples with cycle residuals < 100 Pa. **MAE**: Mean Absolute Error on pipe mass flow. **$\mathbf{Res}_c$**: Average cycle pressure residual. **$\mathbf{Res}_s$**: Average of the per-sample maximum cycle residual. **$\mathbf{Res}_m$**: Maximum pressure residual observed across the dataset.

| **Acc [%]** (↑) | **MAE** [$\mathrm{kg\,s^{-1}}$] (↓) | **$\mathbf{Res}_c$ [Pa]** (↓) | **$\mathbf{Res}_s$ [Pa]** (↓) | **$\mathbf{Res}_m$ [kPa]** (↓) |
|---|---|---|---|---|
| 91.78 | 0.0058 ± 0.0376 | 37.96 ± 717.03 | 64.28 ± 985.73 | 194.38 |

Table 5.5 summarizes the predictive accuracy and hydraulic consistency on the test set. The model achieves an accuracy of 91.78%, closely matching the validation performance and confirming that the strong inductive biases introduced by the Solver-Inspired architecture translate well to unseen synthetic topologies. Similarly, the mass-flow prediction error remains low, with a mean absolute error of $5.8 \times 10^{-3}\,\mathrm{kg\,s^{-1}}$, indicating that the surrogate preserves quantitative accuracy beyond the training and validation datasets.

In terms of absolute cycle residual, the average value ($Res_c$) is 38 Pa, while the average per-sample maximum value is 64 Pa, indicating that even the worst cycle within a given network configuration typically remains below the 100 Pa accuracy threshold. This is further evidenced in Figure 5.5, where the relationship between residual threshold and accuracy is shown. Although the residual distributions exhibit large standard deviations, these are driven by a very small number of extreme cases rather than systematic errors. In particular, residuals exceeding 1 kPa occur in only 0.39% of test samples. The maximum observed residual across the entire test set ($Res_m$) reaches 194 kPa, but such cases are rare and do not reflect the typical

behavior of the model. This behavior will be further investigated at the end of this analysis.

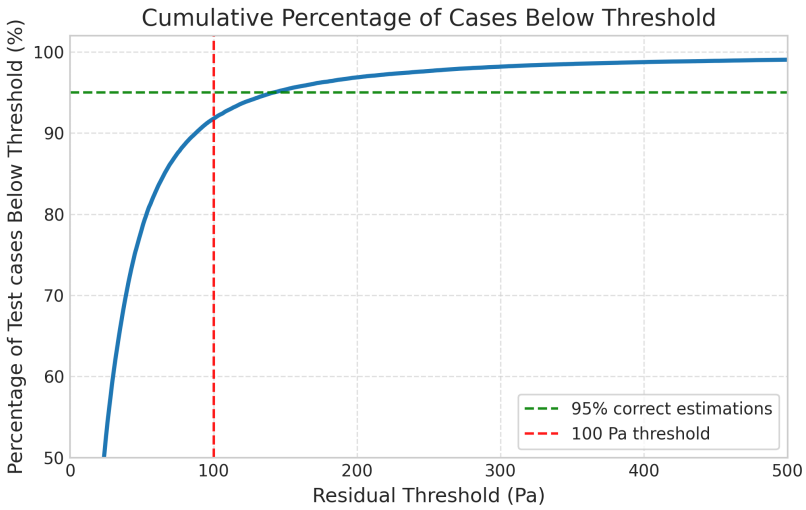



Figure 5.5: Cumulative distribution of the accuracy on the test set for different residual thresholds. The x-axis reports the residual threshold (up to 500 Pa), while the y-axis shows the percentage of samples whose maximum cycle residual does not exceed the threshold. More than 70% of samples exhibit a maximum residual below 50 Pa, highlighting the strong concentration of errors at low magnitudes and the presence of a long but sparse tail.

The model performance change with the number of cycles present in the graph, as reported in Table 5.6. Across networks with one to six cycles, the mean residual remains bounded between approximately 15 Pa to 60 Pa, indicating that the model effectively captures long-range loop interactions even in more complex topologies. Interestingly, the highest error is found in single-cycle cases. For the remaining cases, the model exhibit very low average residual when the graph contains 2 cycles, followed by a monotonic degradation as the number of cycles increases from 2 to 6. This behavior indicates that the proposed architecture scales relatively well with increasing topological complexity. The lower performance on single-cycle cases could be due to the intrinsically different behavior of the model for these cases, where the message-passing mechanism, especially in inter-cycle coordination, lacks the signal coming from other cycles. This singularity is further exacerbated by the lower number of single-cycle graphs in the training set compared to the rest of configurations.

Overall, these results demonstrate that the Solver-Inspired GNN not only achieves high predictive accuracy on unseen data, but also maintains a high degree of hydraulic consistency across a wide range of network configurations, including graphs with multiple interacting cycles.

#### Analysis of Residual Outliers

While the proposed model achieves high accuracy on the vast majority of test samples, a small fraction of cases exhibit significant physical inconsistencies. Figure 5.6 visualizes the

Table 5.6: Test set cycle pressure residuals as a function of the number of cycles in the network. Statistics are computed over individual samples for the selected Solver-Inspired GNN model.

| | $\mathbf{Res}_c$ [Pa] (↓) | | | | | |
|---|---|---|---|---|---|---|
| **Statistic** | **1 Loop** | **2 Loops** | **3 Loops** | **4 Loops** | **5 Loops** | **6 Loops** |
| Mean | 60.21 | 15.40 | 21.81 | 37.28 | 41.45 | 49.77 |
| Std | 1793.40 | 25.96 | 107.84 | 994.51 | 455.05 | 344.03 |

distribution of these outliers with respect to topological complexity. The heatmap shows

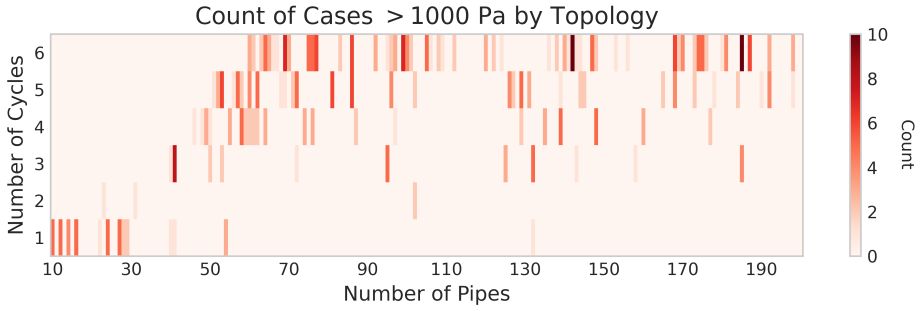
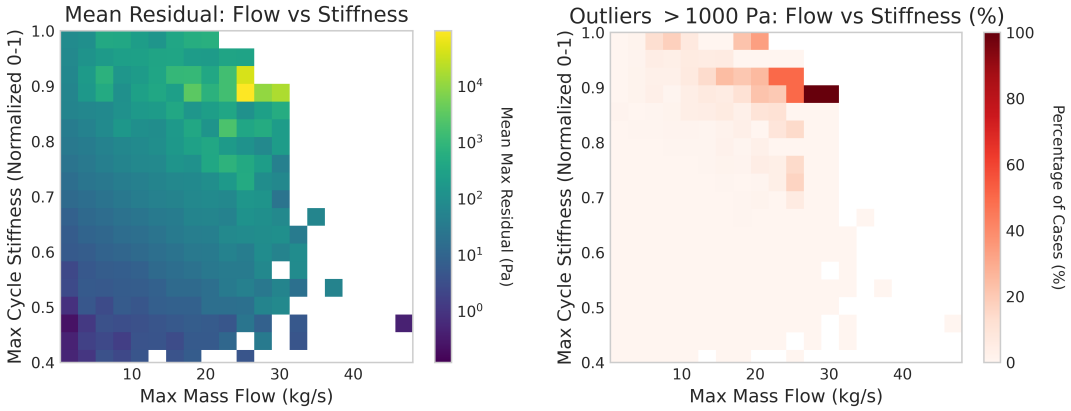



Figure 5.6: Analysis of outliers (> 1000 Pa): heatmap showing the count of outliers as a function of topological coplexity (number of cycles vs. number of pipes).

that outliers are not uniformly distributed. Instead, they cluster in specific configurations, primarily graphs with a low number of pipes (particularly those with a single cycle) and large 6-cycle graphs with several pipes. This is consistent with the previous discussion on the scaling of pressure residuals with the number of cycles, but further characterizes the problem, particularly for single-loop cases. Alongside topological complexity, Figure 5.7 further reveals that the degradation of model's performance scales smoothly with the combination of high cycle stiffness and high boundary mass flows. In particular, outliers are concentrated in a specific region of the problem space where both these quantities are at their limit. Hydraulically, high stiffness cycles correspond to loops with small diameters. In such loops, the pressure drop is extremely sensitive to small fluctuations in mass flow predictions. Consequently, even minor regression errors by the GNN, which are negligible in pipes with larger diameters, can translate into macroscopic violations of Kirchhoff's voltage law. This suggests that future improvements should focus on improving training data coverage and possibly enforcing stricter physics-informed constraints specifically for high-resistance loops.

### 5.5.2 Real-World Generalization

To assess the model's ability to generalize beyond the synthetic training distribution, we evaluate its zero-shot performance on monitoring data from the Verbier DHN introduced




(a) (b)

Figure 5.7: Analysis of the residual outliers ($> 1000 Pa$) in the test set. **Left:** Heatmap showing the average maximum cycle residual (Pa) plotted against the maximum cycle stiffness and boundary flow. High residuals are concentrated in the region of high stiffness and high flow. **Right:** Percentage of outliers in each bin. A problematic zone is clearly delimited at the intersection of high hydraulic resistance and high operational load.

in Section 4.5.1. This constitutes a out-of-distribution test: the Verbier network was never observed during training, its topology was not generated by the synthetic pipeline, and its operational patterns reflect real-world stochasticity rather than idealized sampling strategies.

Table 5.7 summarizes the model's performance across all available monitoring samples from the Verbier network.

Table 5.7: Zero-shot performance on Verbier DHN monitoring data. The model was applied directly without any fine-tuning. **Acc**: Percentage of samples with cycle residuals < 100 Pa. **MAE**: Mean Absolute Error on pipe mass flow. **Res**$_c$: Average cycle pressure residual. **Res**$_s$: Average of the per-sample maximum cycle residual. **Res**$_m$: Maximum pressure residual observed across the dataset.

| **Acc [%] (↑)** | **MAE** [kg s$^{-1}$] (↓) | **Res**$_c$ **[Pa] (↓)** | **Res**$_s$ **[Pa] (↓)** | **Res**$_m$ **[Pa] (↓)** |
|---|---|---|---|---|
| 51.12 | 0.0407 ± 0.0607 | 52.11 ± 66.70 | 129.16 ± 95.26 | 572.09 |

The model achieves an average cycle pressure residual of 52.11 Pa with a standard deviation of 66.70 Pa, indicating that the majority of predictions remain well within the 100 Pa threshold despite the potential distribution shift. Notably, the maximum observed residual of 572 Pa is substantially lower than the extreme outliers encountered in synthetic test data (194 kPa, see Table 5.5). This suggests that the design of the Verbier network, constrained by the need for redundancy and oversizing, occupies a more forgiving region of the problem space than where the high-error outliers in the synthetic distribution lie.

The zero-shot residuals on Verbier (52.11 Pa) are only marginally higher than the test-set performance (37.96 Pa), demonstrating robust transfer despite topological and operational differences. This close alignment reinforces the case that the physics-informed inductive biases embedded in the Solver-Inspired architecture successfully capture fundamental hydraulic principles that generalize across network instances.

Mass flow predictions exhibit a mean absolute error of $0.0407\,\mathrm{kg\,s^{-1}}$, approximately seven times higher than synthetic test performance ($0.0058\,\mathrm{kg\,s^{-1}}$, Table 5.5). While this degradation is non-negligible, it can be largely attributed to structural differences between synthetic and real-world design practices. In particular, oversized pipes in real DHNs exhibit lower pressure drops per unit flow, reducing the relative impact of small absolute errors in mass flow prediction.

To contextualize the observed $0.0407\,\mathrm{kg\,s^{-1}}$ error, we note that this corresponds to less than 1% of typical loop flows in the Verbier network. For pre-design applications, such as evaluating alternative network layouts or sizing expansion projects, this level of accuracy remains practically sufficient, provided that the corresponding pressure prediction error remains low.

### 5.5.3 Computational Efficiency

Beyond predictive accuracy and physical consistency, practical deployment of surrogate models requires computational efficiency during both training and inference. Table 5.8 compares the training cost and convergence speed of the models. The *Solver-Inspired GNN*

Table 5.8: Computational efficiency statistics (10 seeds). **Training Time** is the time taken to complete 100 training epochs. **Convergence Epoch** denotes the training percentage where the maximum validation accuracy was achieved. We report **Mean** ± **Std** and the **(Best)** observed value, where lower is better for both metrics.

| | **Training Time [h]** (↓) | | **Convergence Epoch** (↓) | |
|---|---|---|---|---|
| **Model** | **Mean ± Std** | **(Best)** | **Mean ± Std** | **(Best)** |
| ELU-2-7l | 17.88 ± 0.77 | (16.87) | 63.36 ± 16.96 | (32.32) |
| ELU-2-11l | 25.95 ± 2.18 | (24.12) | 54.95 ± 15.42 | **(28.28)** |
| Solver-Inspired GNN | **13.05 ± 0.45** | **(12.22)** | **53.03 ± 8.66** | (47.47) |

is the most computationally efficient. It requires on average only **13.05 hours** to complete 100 epochs in our setting, compared to 25.95 hours for the deep ELU-2-11l baseline. This efficiency likely stems from the lower complexity of operations in the model (e.g., sum vs softmax) and the lower net number of message-passing operations, as the new model only features one stage. Furthermore, the model converges faster, peaking at epoch 53, whereas the baselines require 55 to 63 epochs on average to reach maximum accuracy. This suggests that the physics-informed inductive biases provide a better initialization and gradient landscape

for the optimization process.

Concerning inference speed, the three models exhibit a similar throughput: in the new variable graph scenario, the Solver-Inspired architecture has an average throughput of **3689 samples/s**, on par with the ELU-2-7l model. The ELU-2-11l model has a slightly lower throughput at 3435 samples/s.

## 5.6 Exploratory Studies

### 5.6.1 Unsupervised Training

The results presented in Section 5.5.1 relied on a composite loss function, where the model was guided primarily by supervised labels during the early phases of training. In our previous work on fixed-topology surrogates (Chapter 3), we observed that training on the physics-based residual loss alone failed to converge to valid hydraulic states, failing to produce meaningful results. However, the dataset introduced in this chapter possesses significant topological and geometric diversity. This raises the question of whether this diversity can introduce new patterns that lead the model to learn the physical behavior of DHNs without explicit labels.

As a preliminary investigation, we retrained the Solver-Inspired GNN using only the physics-based component of the loss function (setting $w_1 = 0$ and $w_2 = 1$ in Equation (3.9) for the entire duration). All other hyperparameters and architectural details remained identical to the supervised configuration. After training 10 models with different random seeds and selecting the best candidate, we compare the performance of this approach against the standard supervised baseline on the validation set. Results are summarized in Table 5.9. Contrary to the

Table 5.9: Comparison of model performance between the supervised baseline (SL) and the physics-only (unsupervised, UL) training strategy on the test set.

| **Mode** | **Acc** [%] (↑) | **MAE** [$\mathrm{kg\,s^{-1}}$] (↓) | $\mathbf{Res_c}$ [Pa] (↓) | $\mathbf{Res}_s$ **[Pa] (↓)** | $\mathbf{Res_m}$ [kPa] (↓) |
|---|---|---|---|---|---|
| SL | **91.78** | **0.0058 ± 0.0376** | **37.96 ± 717.03** | **64.28 ± 985.73** | **194.4** |
| UL | 90.52 | 0.0084 ±0.0689 | 42.39 ±721.31 | 75.39 ±1119.75 | 212.9 |

fixed-topology case, the unsupervised model performs remarkably well. On the test set, the hydraulic consistency is comparable to the supervised baseline, with the mean cycle residual increasing only marginally from 38.0 Pa to 42.4 Pa. Furthermore, despite never observing the ground-truth flow values during training, the model autonomously recovers the correct flow distribution with a MAE of $0.0084\,\mathrm{kg\,s^{-1}}$. While this error is approximately 45% higher than the supervised baseline, it remains negligible in absolute terms, confirming that the model has successfully learned to solve the system of equations rather than memorizing data.

This switch to unsupervised learning does not come at the cost of training stability or computational efficiency. The unsupervised model converges to its optimal validation state slightly

faster, requiring on average **51.62 ± 0.82 epochs** compared to 53.03 ±8.66 for the supervised model. Similarly, the wall-clock training time for 100 epochs is slightly reduced, averaging **12.63 ± 0.33 hours** versus 13.05 ±0.45 hours for the supervised counterpart. This suggests that the physics-based loss provides a smooth and consistent gradient signal, avoiding the potential optimization conflicts associated with a multi-objective loss.

These results, although preliminary, are significant: they validate the hypothesis that provided the dataset is sufficiently diverse, the model can learn to function as a general-purpose solver without the need for computationally expensive ground-truth generation.

## 5.7 Chapter Conclusion

This chapter addressed the final major limitation of the data-driven simulation framework developed in this thesis: the restriction to fixed network topologies. By shifting the learning objective from mapping boundary conditions on a static graph to approximating the solution of flow and pressure distributions in arbitrary networks, we established a topology-agnostic surrogate capable of supporting the early-stage design and planning phases of DHNs.

To achieve this, we proposed a new model architecture designed to more closely mimic the iterative updates of numerical solvers. By explicitly separating static geometric properties from dynamic state variables and enforcing physics-informed inductive biases such as flow reversal symmetry and static/dynamic input separation, the proposed model demonstrated superior performance compared to the less structured baseline establihsed in Chapter 3. Trained on a large dataset of synthetic heterogeneous topologies, the model achieved an accuracy of 90.79% and a mean cycle residual of just 34.6 Pa, significantly outperforming the naive extension of the fixed-topology architecture.

We also verified the model's ability to generalize to out-of-distribution real-world systems. In a zero-shot transfer experiment on the Verbier monitoring data, the model yielded an average cycle residual of 52.11 Pa without any fine-tuning. This result suggests that the model has successfully learned some fundamental physical principles rather than memorizing dataset-specific patterns. While the analysis identified a specific weakness in handling high-stiffness loops under high-load conditions, the overall performance remains robust for the majority of applications.

With a computational throughput of over 3600 samples per second and the ability to differentiate with respect to design parameters, this generalized surrogate bridges the gap between accuracy and speed in surrogate models for design optimization of DHNs. It complements the operational models of Chapter 3 and Chapter 4, completing the thesis's vision of a comprehensive, differentiable simulation ecosystem capable of optimizing DHNs throughout their entire lifecycle, from the initial topological layout to real-time control.

# 6 Conclusions and Outlook

The overarching objective of this thesis was to improve the simulation and, consequently, optimization workflows used throughout the lifecycle of District Heating Networks (DHNs). A central premise was that no single modeling paradigm can simultaneously satisfy the conflicting requirements of speed, accuracy, and flexibility that arise across the phases of pre-design, detailed design, and operation. Instead, this work demonstrated that a structured combination of physics-based models and machine learning surrogates provides a coherent and effective solution, with each modeling approach deployed where it is most appropriate.

The thesis first investigated the detailed design phase, for which it established the necessity of a high-fidelity, component-based physical modeling framework (**Hypothesis 2**). This requirement was addressed through the development of PyDHN, an open-source Python library that integrates a robust hydraulic loop method with a Lagrangian thermal formulation. The framework was designed to prioritize physical consistency, modularity, and topological flexibility, allowing the evaluation of complex meshed networks while supporting custom components. Its validity was demonstrated through systematic verification against experimental data, established benchmarks, and real-world monitoring data from the Verbier DHN. The framework proved capable of accurately reproducing both hydraulic and thermal behavior in a real-world setting, despite asynchronous sensor measurements and missing data. At component level, the implemented thermal pipe model achieved performance comparable to, and in dynamic regimes better than, reference tools. On real monitoring data, after calibration of pressure losses, PyDHN matched sensor measurements with an error consistently within the reported tolerance bounds. While the computational cost of the physical solver remains prohibitive for real-time use, this limitation is consistent with the role of such models in detailed engineering studies, where accuracy, modularity, and interpretability outweigh execution speed. In this context, the results confirm that physics-based simulation remains the right choice for detailed design. Due to their accuracy, these models also provide the engine for generating reliable reference data, which is the cornerstone of creating surrogate models.

Building upon this foundation, the thesis subsequently addressed the operational phase,

where simulation speed becomes the dominant constraint (**Hypothesis 3**). As pointed out in the domain literature and further highlighted during the development of PyDHN, the hydraulic and thermal problems have distinct computational bottlenecks: per-step nonlinear solving cost in hydraulics, and sequential time dependence in thermal dynamics. This insight motivated the development of a hybrid operational framework in which the hydraulic solver is replaced by a Graph Neural Network (GNN) surrogate trained on data from a single network, while the thermal dynamics are modeled using a fully differentiable, GPU-optimized physical formulation. A key methodological contribution was the introduction of a cycle-based graph representation, allowing the surrogate to capture the global pressure-flow dependencies inherent to meshed networks. When coupled with the differentiable thermal model, the resulting framework achieved a nearly two orders of magnitude speedup compared to the fully physics-based reference, while maintaining a high level of agreement in predicted system behavior. Concretely, the hydraulic surrogate reached 2187 steps/s, corresponding to about 85× speedup over the PyDHN hydraulic solver, while converging to a solution within the same residual threshold in more than 97% of test cases. Beyond replication of the physical model, the differentiable nature of the framework allowed gradient-based optimization of operational control strategies, which proved more effective than both manual heuristics and gradient-free optimization methods. These results demonstrate that specialized, topology-specific surrogates can successfully bridge the gap between physical fidelity and real-time optimization, validating their suitability for operational decision support.

The final part of the thesis addressed the pre-design phase, where, in addition to high throughput, the model is required to work over a broad range of possible network topologies (**Hypothesis 1**). The optimization complexity lies in rapidly exploring a vast space of possible network layouts to support early-stage planning and feasibility assessment. To this end, a topology-agnostic GNN framework was developed and trained on a large procedurally generated dataset of hydraulically valid synthetic networks. By explicitly separating static geometric features from dynamic state variables and embedding physics-informed inductive biases into the architecture, the model learned to approximate hydraulic conservation laws across a wide range of unseen topologies. Its generalization capability was further demonstrated through zero-shot application to the network of Verbier, despite the absence of any real data during training. Although slightly less accurate than topology-specific surrogates, the model achieved sufficient fidelity for high-level assessments while offering high computational efficiency. In particular, the proposed model estimated mass-flow distributions in unseen, randomly generated meshed networks with mean cycle residuals in the order of a few tens of pascals (about 35 Pa), with residuals above 1000 Pa occurring only in 0.39% of samples. The average cycle residual remained low (52.11 Pa) also in zero-shot transfer to the network of Verbier. This confirms that generalized GNN surrogates can act as effective proxies for physics-based solvers in the earliest stages of network planning, allowing rapid iteration and informed selection of candidate designs.

In summary, this thesis presented four core novel contributions:

1. It introduced PyDHN, a validated open-source physics-based framework for meshed DHN simulations that integrates custom components in a modular architecture. PyDHN is systematically verified across component-level tests, intra-model benchmarks, and monitored data.

2. It proposed a topology-specific GNN surrogate for faster hydraulic simulation that preserves PyDHN-level fidelity while being almost two orders of magnitude faster. Its key elements are a condensed graph representation tailored to hydraulic functional relationships and an iterative GNN architecture.

3. It reformulated an existing Lagrangian thermal model into a linear, differentiable, GPU-optimized model. The resulting implementation can simulate 89 steps/s while preserving the guarantees of a physics-based approach. Based on this model, a gradient-based optimization of heating curves was demonstrated on the DHN of Verbier.

4. It improved the GNN model towards topology-agnostic pre-design exploration by introducing a data-generation pipeline designed to cover the topology problem space homogeneously, and by adding stronger physics-driven inductive biases within the GNN architecture. The resulting model retains the speed advantage with limited fidelity loss.

While these contributions demonstrate that hybrid physics–machine learning frameworks can substantially improve DHN simulation and optimization, a critical perspective is required on where the proposed methods create immediate value and where they remain exploratory. The strongest near-term value lies in engineering decision support, not in full automation of operational control or in regulatory replacement of high-fidelity physical studies. The remainder of this chapter examines the conditions under which the proposed methods transfer to broader contexts, identifies their main limitations, and outlines corresponding directions for future research.

## 6.1 Limitations, Transferability, and Future Directions

**Model scope and physical fidelity.** The surrogate workflows, including the reformulated Lagrangian implementation, were developed under idealized control assumptions for producers and consumers, without explicit modeling of valves, heat exchangers, or thermal energy storage. Effects such as dynamic pump behavior, valve control logic and fouling were likewise not represented. Extending the framework to include these elements would require substantial methodological effort but is necessary for broader operational deployment. Furthermore, all models were developed for networks and operating ranges typical of third-generation DHNs. Extending them to fourth- and especially fifth- or sixth-generation systems would demand significant modifications; in particular, if fluid incompressibility assumptions no longer hold, decoupling the hydraulic and thermal solvers becomes invalid.

The physics-based framework is, in principle, readily portable because it relies on first-principles conservation laws and modular component definitions. However, practical deployment requires detailed knowledge of the target DHN, including its topology, pipe characteristics, and control rules. Addressing the scope limitations above would directly widen the range of systems to which both the physical models and the derived surrogates can be applied.

**Data requirements and calibration quality.** A persistent challenge is the dependence on data. High-quality, well-aligned monitoring data remain essential for hydraulic calibration, where uncertainty in component characteristics, asset age, and sensor alignment can produce systematic deviations that propagate into surrogate training. Although to a lesser extent, data accuracy also impacts the thermal simulation, in particular due to uncertainty regarding the network characteristics. This issue is often exacerbated by a poor level of digitalization in DHN practice, as well as the lack of standardized formats for data collection and storage.

Concerning surrogate models, they require large datasets for training, which results in a significant time investment for data generation and labeling. Due to the iterative convergence of physics-based solvers, this effort results in a noisy target, where the simulated quantities do not represent exact solutions, but rather approximations with different levels of error. Furthermore, while the proposed GNN architectures and training procedures are designed to cover typical DHNs, their applicability ultimately depends on assumptions embedded in the underlying physical models and on the chosen bounds of the problem space. In this regard, the topology-agnostic surrogate should be interpreted as representing idealized conditions due to the uncalibrated nature of the training data: rarely met in existing systems, but reasonable for pre-design studies. Future work should therefore extend the proposed bounds and reduce dependence on large labeled datasets, for instance through unsupervised training on pressure residuals.

**Surrogate reliability and applicability.** Although strong performance was demonstrated across the intended use cases, surrogate models remain approximations of complex physical solvers. Their predictions cannot fully replace physics-based simulations in safety-critical or regulatory contexts, and even when physics residuals are monitored to objectively evaluate outputs, machine learning models retain an irreducible uncertainty that must be explicitly accounted for in operational use. In practice, this can be naturally addressed by treating unconverged predictions as initial guesses for iterative refinement through the Newton-Raphson method.

A further nuance concerns temporal coupling in hydraulics: in this thesis, hydraulic steps were treated as independent because setpoints were considered fixed at the start of the simulation trajectory. In practice, controls can depend on delayed thermal states, return temperatures, or unmet demand, and in such cases the speed advantage of machine-learning surrogates can diminish because part of the gain comes from per-step parallelism. This advantage is recovered in multi-scenario settings, such as when evaluating alternative topologies, controls, or boundary conditions. The proposed Lagrangian implementation also supports scenario-

level parallelization, adding to the robustness of the approach in these cases.

A related discussion, specific to control optimization, concerns the applicability of the proposed framework under different control paradigms, which vary considerably across regions and system ages. In some countries, district heating systems still operate under constant-flow and centralized temperature regulation with limited automatic balancing at building level, whereas elsewhere variable-flow control with automated substations and demand-side management is more common. Moreover, the degree of operator-side control depends on local regulatory and contractual frameworks: in some contexts, utilities can coordinate production, distribution, and demand management closely, while in others they have limited leverage over consumer-side behavior. Older systems, regardless of region, also tend to rely on simpler regulation strategies and less instrumented infrastructure, affecting both model assumptions and data availability for training. These differences mean that while the proposed hybrid framework is conceptually general, deployment in a new context must account for the specific control paradigm, regulatory setting, and data landscape of the target system, potentially requiring adaptations.

Future work in this direction should quantify the advantages of scenario-level parallelization, including the impact on gradient-free methods where entire populations can be evaluated simultaneously, and on gradient-based methods where independent optimization runs from different starting points can be explored in parallel. Furthermore, it should explore the framework's applicability in modern control settings, where demand-side management is increasingly prevalent.

**Optimization integration.** While the present work focuses primarily on simulation, the ultimate purpose of these models is optimization. DHN planning and operation involve multi-objective trade-offs among economic cost, emissions, equipment wear, and regulatory compliance. The gradient-based optimization demonstrated in this thesis is a first step, but deep integration of the proposed differentiable frameworks into multi-objective, stochastic, and robustness-oriented optimization pipelines remains an open challenge that would significantly increase their practical relevance. Pursuing this direction is among the most promising avenues for translating the computational gains achieved in this work into measurable operational and environmental impact.

**Usability and adoption.** Broad adoption of the proposed methods is unlikely without accessible interfaces and clear guidance for non-specialist users. An easy-to-use front-end is currently missing, and topology-specific surrogates in particular require dedicated machine-learning expertise for their development. Investing in tooling, documentation, and workflow integration is therefore a prerequisite for transitioning these methods from research prototypes to practical engineering tools.

## 6.2 Concluding Remarks

This thesis was motivated by a practical bottleneck in the decarbonization of heating: district heating networks are widely recognized as a key infrastructure for sustainable thermal energy supply, yet their deployment remains hindered by high capital costs, long asset lifetimes, and significant operational uncertainty. In this context, suboptimal design choices or overly conservative operation translate directly into avoidable expenditure. The set of models presented in this work aims at improving the ability to optimize these choices, making it feasible to evaluate more alternatives, without sacrificing physical grounding, before committing capital or operational strategies.

For engineering practice, this implies a shift from sparse, computationally constrained studies to structured iterative workflows. Engineers can first narrow large design spaces quickly, then concentrate high-fidelity analysis where it matters most, and finally apply optimization-informed control tuning during operation. This would result in reduced oversizing, improved thermal performance, and a shorter loop between the collection of monitoring data and the implementation of corrective measures.

For policy and planning, the value is primarily methodological. These models do not replace governance decisions, but they strengthen the analytical basis behind them: transparent comparison of investment scenarios, reproducible quantification of trade-offs among cost, efficiency, and emissions, and more robust stress-testing of decarbonization pathways. In a sector where infrastructure decisions are capital-intensive and long-lived, improving the robustness of the underlying analysis is itself a meaningful lever for progress.

In conclusion, this thesis provides advances in simulation and optimization of district heating systems that can positively impact decision-making at different levels and lifecycle stages, offering a practical pathway to support the transition to sustainable heating.

# Bibliography


Åkesson, J., Årzén, K. .-., Gäfvert, M., Bergdahl, T., & Tummescheit, H. (2010). Modeling and optimization with Optimica and JModelica.org—Languages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering*, *34*(11), 1737–1749. https://doi.org/10.1016/j.compchemeng.2009.11.011

Alizadeh, R., Allen, J. K., & Mistree, F. (2020). Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, *31*(3), 275–298. https://doi.org/10.1007/s00163-020-00336-7

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., … Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, *2*, 929–947. https://doi.org/10.1145/3620665.3640366

Assouline, D., Mohajeri, N., Gudmundsson, A., & Scartezzini, J.-L. (2019). A machine learning approach for mapping the very shallow theoretical geothermal potential. *Geothermal Energy*, *7*(1), 19. https://doi.org/10.1186/s40517-019-0135-6

Averfalk, H., Ingvarsson, P., Persson, U., Gong, M., & Werner, S. (2017). Large heat pumps in Swedish district heating systems. *Renewable and Sustainable Energy Reviews*, *79*, 1275–1284. https://doi.org/10.1016/j.rser.2017.05.135

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016, July). Layer Normalization. https://doi.org/10.48550/arXiv.1607.06450

Bachmann, G., Anagnostidis, S., & Hofmann, T. (2023). Scaling MLPs: A Tale of Inductive Bias. *Advances in Neural Information Processing Systems*, *36*, 60821–60840.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., … Pascanu, R. (2018, October). Relational inductive biases, deep learning, and graph networks. https://doi.org/10.48550/arXiv.1806.01261

Baviere, R., Vallee, M., Crevon, S., Vasset, N., & Lamaison, N. (2023). DISTRICTLAB-H: A new tool to optimize the design and operation of district heating and cooling networks.

*DHC Symposium 2023 - The 18th International Symposium on District Heating and Cooling*.

Beckenbauer, D., Ehrenwirth, M., Klärner, M., Zörner, W., & Cheng, V. (2017). Validation of a District Heating System Model and Simulation-Based Investigation of Bidirectional Heat Transport by Decentralised Solar Thermal Plants. *Proceedings of SWC2017/SHC2017*, 1–12. https://doi.org/10.18086/swc.2017.06.01

Benonysson, A. (1991). Dynamic modelling and operational optimization of district heating systems.

Bergsteinsson, H. G., Møller, J. K., Thilker, C. A., Guericke, D., Heller, A., Nielsen, T. S., & Madsen, H. (2022). Data-Driven Methods for Efficient Operation of District Heating Systems. In R. Garay-Martinez & A. Garrido-Marijuan (Eds.), *Handbook of Low Temperature District Heating* (pp. 129–163). Springer International Publishing. https://doi.org/10.1007/978-3-031-10410-7_6

Blochwitz, T., Otter, M., Åkesson, J., Arnold, M., Clauss, C., Elmqvist, H., Friedrich, M., Junghanns, A., Mauss, J., Neumerkel, D., Olsson, H., & Viel, A. (2012). Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models: 9th International Modelica Conference. *Proceedings of the 9th International Modelica Conference*, 173–184. https://doi.org/10.3384/ecp12076173

Blommaert, M., Wack, Y., & Baelmans, M. (2020). An adjoint optimization approach for the topological design of large-scale district heating networks based on nonlinear models. *Applied Energy*, *280*, 116025. https://doi.org/10.1016/j.apenergy.2020.116025

Boesten, S., Ivens, W., Dekker, S. C., & Eijdems, H. (2019). 5th generation district heating and cooling systems as a solution for renewable urban thermal energy supply. *Advances in Geosciences*, *49*, 129–136. https://doi.org/10.5194/adgeo-49-129-2019

Bonvin, M., Dewarrat, P., Kämpf, J., Ilazi, M., & Nafzger, A. (2024, June). *eguzki: Programme de simulation de réseaux de chauffage à distance basé sur l'intelligence artificielle pour la résolution rapide et prédictive de réseaux complexes bouclés* (Rapport final No. SI/502111-01). Office fédéral de l'énergie OFEN. Berne, Switzerland.

Boussaid, T., Rousset, F., Scuturici, V.-M., & Clausse, M. (2023). Evaluation of Graph Neural Networks as Surrogate Model for District Heating Networks Simulation. *36th International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS 2023)*, 3182–3193. https://doi.org/10.52202/069564-0286

Boussaid, T., Rousset, F., Scuturici, V.-M., & Clausse, M. (2024). Enabling fast prediction of district heating networks transients via a physics-guided graph neural network. *Applied Energy*, *370*, 123634. https://doi.org/10.1016/j.apenergy.2024.123634

Bronstein, M. M., Bruna, J., Cohen, T., & Veličković, P. (2021, May). Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. https://doi.org/10.48550/arXiv.2104.13478

Brown, A., Foley, A., Laverty, D., McLoone, S., & Keatley, P. (2022). Heating and cooling networks: A comprehensive review of modelling approaches to map future directions. *Energy*, *261*, 125060. https://doi.org/10.1016/j.energy.2022.125060

Buffa, S., Fouladfar, M. H., Franchini, G., Lozano Gabarre, I., & Andrés Chicote, M. (2021). Advanced Control and Fault Detection Strategies for District Heating and Cooling Systems—A Review. *Applied Sciences*, *11*(1), 455. https://doi.org/10.3390/app11010455

Bundesgesetz über die Ziele im Klimaschutz, die Innovation und die Stärkung der Energiesicherheit. (2022).

Buonomano, A., Forzano, C., Palombo, A., & Russo, G. (2023). Solar-assisted district heating networks: Development and experimental validation of a novel simulation tool for the energy optimization. *Energy Conversion and Management*, *288*, 117133. https://doi.org/10.1016/j.enconman.2023.117133

Calise, F., Cappiello, F. L., Cimmino, L., Dentice d'Accadia, M., & Vicidomini, M. (2023). A comparative thermoeconomic analysis of fourth generation and fifth generation district heating and cooling networks. *Energy*, *284*, 128561. https://doi.org/10.1016/j.energy.2023.128561

Calixto, S., Cozzini, M., Fedrizzi, R., & Manzolini, G. (2024). A New Method for the Techno-Economic Analysis and the Identification of Expansion Strategies of Neutral-Temperature District Heating and Cooling Systems. *Energies*, *17*(9), 2159. https://doi.org/10.3390/en17092159

Capone, M., Guelpa, E., & Verda, V. (2021). Accounting for pipeline thermal capacity in district heating simulations. *Energy*, *219*, 119663. https://doi.org/10.1016/j.energy.2020.119663

Caputo, P., Ferla, G., Belliardi, M., & Cereghetti, N. (2021). District thermal systems: State of the art and promising evolutive scenarios. A focus on Italy and Switzerland. *Sustainable Cities and Society*, *65*, 102579. https://doi.org/10.1016/j.scs.2020.102579

Castaldo, V. L., & Pisello, A. L. (2018). Uses of dynamic simulation to predict thermal-energy performance of buildings and districts: a review. *WIREs Energy and Environment*, *7*(1), e269. https://doi.org/10.1002/wene.269

Çengel, Y. A. (2003). *Heat Transfer: A Practical Approach*. McGraw-Hill.

Chambers, J., Narula, K., Sulzer, M., & Patel, M. K. (2019). Mapping district heating potential under evolving thermal demand scenarios and technologies: A case study for Switzerland. *Energy*, *176*, 682–692. https://doi.org/10.1016/j.energy.2019.04.044

Chen, J., Ding, L., Lv, H., Zhang, K., Hou, C., Lai, Z., & Lin, X. (2023). Topology optimization method of district heating system considering load uncertainty. *Energy Reports*, *10*, 4679–4691. https://doi.org/10.1016/j.egyr.2023.11.029

Chicherin, S. (2025). Conversion to Variable Flow Rate—Advanced Control of a District Heating (DH) System with a Focus on Operational Data. *Energies*, *18*(11), 2772. https://doi.org/10.3390/en18112772

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014, September). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. https://doi.org/10.48550/arXiv.1406.1078

Clevert, D.-A., Unterthiner, T., & Hochreiter, S. (2016, February). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). https://doi.org/10.48550/arXiv.1511.07289

*Climate change 2022: mitigation of climate change* (tech. rep.). (2022). IPCC. Geneva.

Cognet, P. (2020). *District Energy Network Simulation for the Evaluation of Solar Thermal Potential* [Master's thesis, EPFL].

Cross, H. (1936). Analysis of flow in networks of conduits or conductors. *University of Illinois. Engineering Experiment Station. Bulletin; no. 286*.

Dancker, J., & Wolter, M. (2021). Improved quasi-steady-state power flow calculation for district heating systems: A coupled Newton-Raphson approach. *Applied Energy*, *295*, 116930. https://doi.org/10.1016/j.apenergy.2021.116930

de Giuli, L. B., La Bella, A., & Scattolini, R. (2024). Physics-Informed Neural Network Modeling and Predictive Control of District Heating Systems. *IEEE Transactions on Control Systems Technology*, *32*(4), 1182–1195. https://doi.org/10.1109/TCST.2024.3355476

De la Cruz, I., & Ugalde-Loo, C. E. (2021). District Heating and Cooling Systems. In *Microgrids and Local Energy Systems* (pp. 91–126). IntechOpen.

Delubac, R., Serra, S., Sochard, S., & Reneaume, J.-M. (2021). A Dynamic Optimization Tool to Size and Operate Solar Thermal District Heating Networks Production Plants. *Energies*, *14*(23), 8003. https://doi.org/10.3390/en14238003

Dénarié, A., Aprile, M., & Motta, M. (2019). Heat transmission over long pipes: New model for fast and accurate district heating simulations. *Energy*, *166*, 267–276. https://doi.org/10.1016/j.energy.2018.09.186

Dénarié, A., Aprile, M., & Motta, M. (2023). Dynamical modelling and experimental validation of a fast and accurate district heating thermo-hydraulic modular simulation tool. *Energy*, *282*, 128397. https://doi.org/10.1016/j.energy.2023.128397

Desai, S. (2021). *Physics-informed neural networks for data-efficient learning* [Doctoral dissertation, University of Oxford].

Die Bundesversammlung der Schweizerischen Eidgenossenschaft. (2016, September). Energiegesetz (EnG).

Elmqvist, H., Brück, D., & Otter, M. (1996). Dymola-user's manual. *Dynasim AB, Research Park Ideon, Lund, Sweden*.

*Energy strategy 2050 monitoring report 2019 abridged version* (tech. rep.). (2019, November). Swiss Federal Office of Energy. Bern.

EQUA. (2025). Changelog.

Febres, J., Ubieta, E., Sterling, R., del Hoyo, I., & López, S. (2017, December). District Cooling Open Source Library (DCOL). https://doi.org/10.5281/zenodo.1215665

Federal Department of the Environment, Transport, Energy and Communications & Federal Office for the Environment. (2025, April). *Kenngrössen zur Entwicklung der Treibhausgasemissionen in der Schweiz 1990-2023* (tech. rep.). Federal Department of the Environment, Transport, Energy and Communications (DETEC) / Federal Office for the Environment (FOEN). Bern.

Fernqvist, N., Broberg, S., Torén, J., & Svensson, I.-L. (2023). District heating as a flexibility service: Challenges in sector coupling for increased solar and wind power production in Sweden. *Energy Policy*, *172*, 113332. https://doi.org/10.1016/j.enpol.2022.113332

Fey, M., & Lenssen, J. E. (2019, April). Fast Graph Representation Learning with PyTorch Geometric. https://doi.org/10.48550/arXiv.1903.02428

Fischer-Uhrig Engineering GmbH. (2025). *STANET*.

Friebe, M., Karasu, A., & Kriegel, M. (2023). Methodology to compare and optimize district heating and decentralized heat supply for energy transformation on a municipality level. *Energy*, *282*, 128987. https://doi.org/10.1016/j.energy.2023.128987

Fritzson, P. (1998). Modelica—A language for equation-based physical modeling and high performance simulation. In B. Kågström, J. Dongarra, E. Elmroth, & J. Waśniewski (Eds.), *Applied Parallel Computing Large Scale Scientific and Industrial Problems* (pp. 149–160). Springer. https://doi.org/10.1007/BFb0095332

Fritzson, P., Pop, A., Abdelhak, K., Ashgar, A., Bachmann, B., Braun, W., Bouskela, D., Braun, R., Buffoni, L., Casella, F., Castro, R., Franke, R., Fritzson, D., Gebremedhin, M., Heuermann, A., Lie, B., Mengist, A., Mikelsons, L., Moudgalya, K., … Östlund, P. (2020). The OpenModelica Integrated Environment for Modeling, Simulation, and Model-Based Development. *41*(4), 241–295. https://doi.org/10.4173/mic.2020.4.1

Fukushima, K. (1969). Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. *IEEE Transactions on Systems Science and Cybernetics*, *5*(4), 322–333. https://doi.org/10.1109/TSSC.1969.300225

Ghilardi, L. M. P., Castelli, A. F., Moretti, L., Morini, M., & Martelli, E. (2021). Co-optimization of multi-energy system operation, district heating/cooling network and thermal comfort management for buildings. *Applied Energy*, *302*, 117480. https://doi.org/10.1016/j.apenergy.2021.117480

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017). Neural Message Passing for Quantum Chemistry. *Proceedings of the 34th International Conference on Machine Learning*, 1263–1272.

Graf, C., Pärisch, P., Marszal-Pomianowska, A., Frandsen, M., Bendinger, B., & Cadenbach, A. (2024). Domestic hot water systems in well-insulated residential buildings: A comparative simulation study on efficiency and hygiene challenges. *Energy*, *313*, 133587. https://doi.org/10.1016/j.energy.2024.133587

Gudmundsson, O., Schmidt, R.-R., Dyrelund, A., & Thorsen, J. E. (2022). Economic comparison of 4GDH and 5GDH systems – Using a case study. *Energy*, *238*, 121613. https://doi.org/10.1016/j.energy.2021.121613

Guelpa, E., Capone, M., Sciacovelli, A., Vasset, N., Baviere, R., & Verda, V. (2023). Reduction of supply temperature in existing district heating: A review of strategies and implementations. *Energy*, *262*, 125363. https://doi.org/10.1016/j.energy.2022.125363

Guelpa, E., & Verda, V. (2019). Compact physical model for simulation of thermal networks. *Energy*, *175*, 998–1008. https://doi.org/10.1016/j.energy.2019.03.064

Guo, T., Chen, Y., & Kriegel, M. (2024). Novel multi-level optimization of district heating systems: Managing spatial scale and equipment portfolio design. *Energy and Buildings*, *317*, 114395. https://doi.org/10.1016/j.enbuild.2024.114395

Hafsi, Z. (2021). Accurate explicit analytical solution for Colebrook-White equation. *Mechanics Research Communications*, *111*, 103646.

Hansen, C. H., Gudmundsson, O., & Detlefsen, N. (2019). Cost efficiency of district heating for low energy buildings of the future. *Energy*, *177*, 77–86. https://doi.org/10.1016/j.energy.2019.04.046

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *Proceedings of the IEEE International Conference on Computer Vision*, 1026–1034.

Heinen, S., Mancarella, P., O'Dwyer, C., & O'Malley, M. (2018). Heat Electrification: The Latest Research in Europe. *IEEE Power and Energy Magazine*, *16*(4), 69–78. https://doi.org/10.1109/MPE.2018.2822867

Hermans, L., Haesen, R., Uytterhoeven, A., Peere, W., Boydens, W., & Helsen, L. (2023). Predesign of collective residential solar districts with seasonal thermal energy storage: Importance of level of detail. *Applied Thermal Engineering*, *226*, 120203. https://doi.org/10.1016/j.applthermaleng.2023.120203

Hilpert, S., Kaldemeyer, C., Krien, U., Günther, S., Wingenbach, C., & Plessmann, G. (2018). The Open Energy Modelling Framework (oemof) - A new approach to facilitate open science in energy system modelling. *Energy Strategy Reviews*, *22*, 16–25. https://doi.org/10.1016/j.esr.2018.07.001

Hiris, D. P., Pop, O. G., & Balan, M. C. (2022). Preliminary sizing of solar district heating systems with seasonal water thermal storage. *Heliyon*, *8*(2). https://doi.org/10.1016/j.heliyon.2022.e08932

Hirsch, H., & Nicolai, A. (2022). An efficient numerical solution method for detailed modelling of large 5th generation district heating and cooling networks. *Energy*, *255*, 124485. https://doi.org/10.1016/j.energy.2022.124485

Hirsch, H., & Paepcke, A. (2022). Detailed Modelling of Large District Heating and Cooling Network coupled to Ground Heat Exchanger. *BauSim Conference 2022*, *9*, 0–0. https://doi.org/10.26868/29761662.2022.38

Höffner, D., & Glombik, S. (2024). Energy system planning and analysis software—A comprehensive meta-review with special attention to urban energy systems and district heating. *Energy*, *307*, 132542. https://doi.org/10.1016/j.energy.2024.132542

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

I., G. E. S. (2025). SimulationX | System Simulation Software.

Idiap/pydhn: Python library for modeling, simulating, and analyzing district heating networks. (2023).

IEA. (2023, June). *Unlocking Smart Grid Opportunities in Emerging Markets and Developing Economies* (tech. rep.). OECD. https://doi.org/10.1787/3cd7a4b0-en

International Energy Agency. (2024). *Energy Efficiency 2024* (tech. rep.). IEA.

Ioffe, S., & Szegedy, C. (2015, March). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. https://doi.org/10.48550/arXiv.1502.03167

Jansen, D., Richter, V., Maier, L., Frisch, J., van Treeck, C., & Müller, D. (2025). Open-source framework for automated generation of building energy performance simulation

models and beyond from BIM Data. *Automation in Construction*, *179*, 106427. https://doi.org/10.1016/j.autcon.2025.106427

Jie, P., Zhao, W., Li, F., Wei, F., & Li, J. (2020). Optimizing the pressure drop per unit length of district heating piping networks from an environmental perspective. *Energy*, *202*, 117681. https://doi.org/10.1016/j.energy.2020.117681

Johra, H., Mans, M., Filonenko, K., De Jaeger, I., Saelens, D., & Tvedebrink, T. (2021). Evaluating different metrics for inter-model comparison of urban-scale building energy simulation time series. *Proceedings of Building Simulation 2021*, 1556–1563. https://doi.org/10.26868/25222708.2021.30410

Jorissen, F., Reynders, G., Baetens, R., Picard, D., Saelens, D., & Helsen, L. (2018). Implementation and verification of the IDEAS building energy simulation library. *Journal of Building Performance Simulation*, *11*(6), 669–688. https://doi.org/10.1080/19401493.2018.1428361

Kemmler, A., & Vu, P. (2024, October). *Analyse des schweizerischen Energieverbrauchs 2000–2023 nach Verwendungszwecken* (Synthesebericht). Swiss Federal Office of Energy. Bern.

Kipf, T. N., & Welling, M. (2017, February). Semi-Supervised Classification with Graph Convolutional Networks. https://doi.org/10.48550/arXiv.1609.02907

Klein, S. A. (1988). TRNSYS-A transient system simulation program. *University of Wisconsin-Madison, Engineering Experiment Station Report*, 38–12.

Knudsen, B. R., Rohde, D., & Kauko, H. (2021). Thermal energy storage sizing for industrial waste-heat utilization in district heating: A model predictive control approach. *Energy*, *234*, 121200. https://doi.org/10.1016/j.energy.2021.121200

Kudela, L., Chýlek, R., & Pospíšil, J. (2020). Efficient Integration of Machine Learning into District Heating Predictive Models. *Energies*, *13*(23), 6381. https://doi.org/10.3390/en13236381

Kuntuarova, S., Lickleder er, T., Huynh, T., Zinsmeister, D., Hamacher, T., & Perić, V. (2024). Design and simulation of district heating networks: A review of modeling approaches and tools. *Energy*, *305*, 132189. https://doi.org/10.1016/j.energy.2024.132189

Kusuda, T., & Achenbach, P. R. (1965). *Earth temperature and thermal diffusivity at selected stations in the United States* (tech. rep.). National Bureau of Standards Gaithersburg MD.

Laakkonen, L., Korpela, T., Kaivosoja, J., Vilkko, M., Majanne, Y., & Nurmoranta, M. (2017). Predictive Supply Temperature Optimization of District Heating Networks Using Delay Distributions. *Energy Procedia*, *116*, 297–309. https://doi.org/10.1016/j.egypro.2017.05.076

Lambert, J., & Spliethoff, H. (2024). A two-phase nonlinear optimization method for routing and sizing district heating systems. *Energy*, *302*, 131843. https://doi.org/10.1016/j.energy.2024.131843

Leitner, B., Widl, E., Gawlik, W., & Hofmann, R. (2019). A method for technical assessment of power-to-heat use cases to couple local district heating and electrical distribution grids. *Energy*, *182*, 729–738. https://doi.org/10.1016/j.energy.2019.06.016

Li, G., Xiong, C., Thabet, A., & Ghanem, B. (2020, June). DeeperGCN: All You Need to Train Deeper GCNs. https://doi.org/10.48550/arXiv.2006.07739

Li, H., Hou, J., Hong, T., Ding, Y., & Nord, N. (2021). Energy, economic, and environmental analysis of integration of thermal energy storage into district heating systems using waste heat from data centres. *Energy*, *219*, 119582. https://doi.org/10.1016/j.energy.2020.119582

Lima, R. (2025). 6th generation district heating and cooling networks: design, experimental results and future challenges. *CEES 2025 - International Conference on Construction, Energy, Environment and Sustainability*.

Lohmeier, D., Cronbach, D., Drauz, S. R., Braun, M., & Kneiske, T. M. (2020). Pandapipes: An Open-Source Piping Grid Calculation Package for Multi-Energy Grid Simulations. *Sustainability*, *12*(23), 9899. https://doi.org/10.3390/su12239899

Loshchilov, I., & Hutter, F. (2019, January). Decoupled Weight Decay Regularization. https://doi.org/10.48550/arXiv.1711.05101

Lu, M., Zhang, C., Zhang, D., Wang, R., Zhou, Z., Zhan, C., Zai, X., & Jing, Q. (2021). Operational optimization of district heating system based on an integrated model in TRNSYS. *Energy and Buildings*, *230*, 110538. https://doi.org/10.1016/j.enbuild.2020.110538

Lund, H., Østergaard, P. A., Nielsen, T. B., Werner, S., Thorsen, J. E., Gudmundsson, O., Arabkoohsar, A., & Mathiesen, B. V. (2021). Perspectives on fourth and fifth generation district heating. *Energy*, *227*, 120520. https://doi.org/10.1016/j.energy.2021.120520

Lund, H., Werner, S., Wiltshire, R., Svendsen, S., Thorsen, J. E., Hvelplund, F., & Mathiesen, B. V. (2014). 4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems. *Energy*, *68*, 1–11. https://doi.org/10.1016/j.energy.2014.02.089

Ma, Z. G., & Jørgensen, B. N. (2024). A Review of Intelligent Technologies in District Heating Systems. *Progress in Artificial Intelligence: 23rd EPIA Conference on Artificial Intelligence, EPIA 2024, Viana Do Castelo, Portugal, September 3–6, 2024, Proceedings, Part II*, 123–134. https://doi.org/10.1007/978-3-031-73500-4_11

Ma, Z. G., & Lygnerud, K. (2025). Digital Transformation of District Heating: A Scoping Review of Technological Innovation, Business Model Evolution, and Policy Integration. *Energies*, *18*(22), 5994. https://doi.org/10.3390/en18225994

Madsen, H., Sejling, K., Søgaard, H. T., & Palsson, O. P. (1994). On flow and supply temperature control in district heating systems. *Heat Recovery Systems and CHP*, *14*(6), 613–620. https://doi.org/10.1016/0890-4332(94)90031-0

Maier, L., Jansen, D., Wüllhorst, F., Kremer, M., Kümpel, A., Blacha, T., & Müller, D. (2023). AixLib: an open-source Modelica library for compound building energy systems from component to district level with automated quality management. *Journal of Building Performance Simulation*, *17*(2), 196–219. https://doi.org/10.1080/19401493.2023.2250521

Masson-Delmotte, V., Zhai, P., Pirani, A., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, Ö., Yu, R., & Zhou, B. (Eds.). (2021). *Climate*

*Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press. https://doi.org/10.1017/9781009157896

Matthews, T. G., Thompson, C. V., Wilson, D. L., Hawthorne, A. R., & Mage, D. T. (1989). Air velocities inside domestic environments: An important parameter in the study of indoor air quality and climate. *Environment International*, *15*(1), 545–550. https://doi.org/10.1016/0160-4120(89)90074-3

Maurer, J., Ratzel, O. M., Malan, A. J., & Hohmann, S. (2021). Comparison of discrete dynamic pipeline models for operational optimization of District Heating Networks. *Energy Reports*, *7*, 244–253. https://doi.org/10.1016/j.egyr.2021.08.150

Menapace, A., Boscheri, W., Baratieri, M., & Righetti, M. (2020). An efficient numerical scheme for the thermo-hydraulic simulations of thermal grids. *International Journal of Heat and Mass Transfer*, *161*, 120304. https://doi.org/10.1016/j.ijheatmasstransfer.2020.120304

Meng, C., Griesemer, S., Cao, D., Seo, S., & Liu, Y. (2025). When physics meets machine learning: a survey of physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, *1*(1), 20. https://doi.org/10.1007/s44379-025-00016-0

Mugnini, A., Ferracuti, F., Lorenzetti, M., Comodi, G., & Arteconi, A. (2022). Advanced control techniques for CHP-DH systems: A critical comparison of Model Predictive Control and Reinforcement Learning. *Energy Conversion and Management: X*, *15*, 100264. https://doi.org/10.1016/j.ecmx.2022.100264

*Mustervorschriften der Kantone im Energiebereich (MuKEn)* (Ausgabe 2025). (2025, August).

Nakama, C. S. M., Knudsen, B. R., Tysland, A. C., & Jäschke, J. (2023). A simple dynamic optimization-based approach for sizing thermal energy storage using process data. *Energy*, *268*, 126671. https://doi.org/10.1016/j.energy.2023.126671

NetSim - Grid Simulation System. (2025).

Nigro, L., La Bella, A., Casella, F., & Scattolini, R. (2025). Control-Oriented Modeling, Simulation, and Predictive Control of District Heating Networks. *IEEE Transactions on Automation Science and Engineering*, *22*, 7064–7079. https://doi.org/10.1109/TASE.2024.3460173

Oppelt, T., Urbaneck, T., Gross, U., & Platzer, B. (2016). Dynamic thermo-hydraulic model of district cooling networks. *Applied Thermal Engineering, 102*, 336–345.

Osiadacz, A. J., & Pienkosz, K. (1988). Methods of steady-state simulation for gas networks. *International Journal of Systems Science*, *19*(7), 1311–1321. https://doi.org/10.1080/00207728808547163

Paulick, S., Schroth, C., Guddusch, S., & Rühling, K. (2018). Resulting Effects On Decentralized Feed-In Into District Heating Networks – A Simulation Study. *Energy Procedia*, *149*, 49–58. https://doi.org/10.1016/j.egypro.2018.08.168

Pesola, A. (2023). Cost-optimization model to design and operate hybrid heating systems – Case study of district heating system with decentralized heat pumps in Finland. *Energy*, *281*, 128241. https://doi.org/10.1016/j.energy.2023.128241

Pizzolato, A., Sciacovelli, A., & Verda, V. (2018). Topology Optimization of Robust District Heating Networks. *Journal of Energy Resources Technology, 140*(2), 020905. https://doi.org/10.1115/1.4038312

Popiel, ., & Wojtkowiak, J. (1998). Simple formulas for thermophysical properties of liquid water for heat transfer calculations (from 0 C to 150 C). *Heat transfer engineering*, *19*(3), 87–101.

Popovski, E., Ragwitz, M., & Brugger, H. (2023). Decarbonization of district heating and deep retrofits of buildings as competing or synergetic strategies for the implementation of the efficiency first principle. *Smart Energy*, *10*, 100096. https://doi.org/10.1016/j.segy.2023.100096

*Potenzial von fernwärme- und fernkälteanlagen* (tech. rep.). (2021, December). The Federal Council. Bern.

PSI Neplan. (2025).

Quaggiotto, D., Vivian, J., & Zarrella, A. (2021). Management of a district heating network using model predictive control with and without thermal storage. *Optimization and Engineering*, *22*(3), 1897–1919. https://doi.org/10.1007/s11081-021-09644-w

Rahal, H. (1995). A co-tree flows formulation for steady state in water distribution networks. *Advances in Engineering Software*, *22*(3), 169–178. https://doi.org/10.1016/0965-9978(95)00020-W

Ringsquandl, M., Sellami, H., Hildebrandt, M., Beyer, D., Henselmeyer, S., Weber, S., & Joblin, M. (2021). Power to the Relational Inductive Bias: Graph Neural Networks in Electrical Power Grids. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1538–1547. https://doi.org/10.1145/3459637.3482464

Robinson, D., Haldi, F., Kämpf, J. H., Leroux, P., Perez, D., Rasheed, A., & Wilke, U. (2009). CitySim: Comprehensive micro-simulation of resource flows for sustainable urban planning. *Proceedings of Building Simulation 2009: 11th Conference of IBPSA*, *11*, 1083–1090. https://doi.org/10.26868/25222708.2009.1083-1090

Röder, J., Meyer, B., Krien, U., Zimmermann, J., Stührmann, T., & Zondervan, E. (2021). Optimal Design of District Heating Networks with Distributed Thermal Energy Storages – Method and Case Study. *International Journal of Sustainable Energy Planning and Management*, *31*, 5–22. https://doi.org/10.5278/ijsepm.6248

Rodrigue, D., Mabrouk, M. T., Pasdeloup, B., Meyer, P., & Lacarrière, B. (2024, April). Leveraging Artificial Neural Networks to Accelerate Dynamic Simulations of District Heating Networks Through Topology Reduction. https://doi.org/10.2139/ssrn.4799722

Rogelj, J., Luderer, G., Pietzcker, R. C., Kriegler, E., Schaeffer, M., Krey, V., & Riahi, K. (2015). Energy system transformations for limiting end-of-century warming to below 1.5 °C. *Nature Climate Change*, *5*(6), 519–527. https://doi.org/10.1038/nclimate2572

Rusch, T. K., Bronstein, M. M., & Mishra, S. (2023, March). A Survey on Oversmoothing in Graph Neural Networks. https://doi.org/10.48550/arXiv.2303.10993

Saelens, D., De Jaeger, I., Bünning, F., Mans, M., Vandermeulen, A., Van der Heijde, B., Garreau, E., Maccarini, A., Rønneseth, Ø., Sartori, I., et al. (2019). Towards a DESTEST: A

district energy simulation test developed in IBPSA project 1. *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, 3569–3577.

Sahlin, P., Skogqvist, P., Lebedev, A., Eriksson, L., & Grozman, P. (2019). On the Scalability of Equation-Based Building and District Simulation Models. *Building Simulation 2019*, 2584–2590. https://doi.org/10.26868/25222708.2019.210130

Salenbien, R., Wack, Y., Baelmans, M., & Blommaert, M. (2023). Geographically informed automated non-linear topology optimization of district heating networks. *Energy*, *283*, 128898. https://doi.org/10.1016/j.energy.2023.128898

Saloux, E., Runge, J., & Zhang, K. (2023). Operation optimization of multi-boiler district heating systems using artificial intelligence-based model predictive control: Field demonstrations. *Energy*, *285*, 129524. https://doi.org/10.1016/j.energy.2023.129524

Saloux, E., Runge, J., & Zhang, K. (2024). Field Implementation of a Predictive Control Strategy in District Heating Systems: A Tale of Two Demonstration Sites. In B. N. Jørgensen, L. C. P. da Silva, & Z. Ma (Eds.), *Energy Informatics* (pp. 314–327). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-48652-4_21

Salvestroni, M., Pierucci, G., Pourreza, A., Fagioli, F., Taddei, F., Messeri, M., & De Lucia, M. (2021). Design of a solar district heating system with seasonal storage in Italy. *Applied Thermal Engineering*, *197*, 117438. https://doi.org/10.1016/j.applthermaleng.2021.117438

Sartor, K., & Dewalef, P. (2017). Experimental validation of heat transport modelling in district heating networks. *Energy*, *137*, 961–968.

Sartor, K., & Dewallef, P. (2018). Integration of heat storage system into district heating networks fed by a biomass CHP plant. *Journal of Energy Storage*, *15*, 350–358. https://doi.org/10.1016/j.est.2017.12.010

Sartor, K., Thomas, D., & Dewallef, P. (2018). A comparative study for simulating heat transport in large district heating networks. *International Journal of Heat and Technology*, *36*(1), 301–308. https://doi.org/10.18280/ijht.360140

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Schill, W.-P., & Zerrahn, A. (2020). Flexible electricity use for heating in markets with renewable energy. *Applied Energy*, *266*, 114571. https://doi.org/10.1016/j.apenergy.2020.114571

Schmidt, D. (2021). Digitalization of district heating and cooling systems. *Energy Reports*, *7*, 458–464. https://doi.org/10.1016/j.egyr.2021.08.082

Schneider Electric. (2025). EcoStruxure District Energy.

Schweiger, G., Heimrath, R., Falay, B., O'Donovan, K., Nageler, P., Pertschy, R., Engel, G., Streicher, W., & Leusbrock, I. (2018). District energy systems: Modelling paradigms and general-purpose tools. *Energy*, *164*, 1326–1340. https://doi.org/10.1016/j.energy.2018.08.193

Schweiger, G., Larsson, P.-O., Magnusson, F., Lauenburg, P., & Velut, S. (2017). District heating and cooling systems – Framework for Modelica-based simulation and dynamic optimization. *Energy*, *137*, 566–578. https://doi.org/10.1016/j.energy.2017.05.115

Simulink - Simulation and Model-Based Design. (2025).

Singha, M. (2025, November). Learning Under Laws: A Constraint-Projected Neural PDE Solver that Eliminates Hallucinations. https://doi.org/10.48550/arXiv.2511.03578

Socci, L., Rocchetti, A., Verzino, A., Zini, A., & Talluri, L. (2024). Enhancing third-generation district heating networks with data centre waste heat recovery: analysis of a case study in Italy. *Energy*, *313*, 134013. https://doi.org/10.1016/j.energy.2024.134013

Sorknæs, P., Nielsen, S., Lund, H., Mathiesen, B. V., Moreno, D., & Thellufsen, J. Z. (2022). The benefits of 4th generation district heating and energy efficient datacentres. *Energy*, *260*, 125215. https://doi.org/10.1016/j.energy.2022.125215

Stevanovic, V. D., Prica, S., Maslovaric, B., Zivkovic, B., & Nikodijevic, S. (2007). Efficient numerical method for district heating system hydraulics. *Energy Conversion and Management*, *48*(5), 1536–1543. https://doi.org/10.1016/j.enconman.2006.11.018

Swiss Federal Office of Energy & Federal Office for the EnvironmentFederal Office for the Environment (FOEN). (2025, July). *Stand der Energie- und Klimapolitik in den Kantonen 2025* (tech. rep.). Bern.

Swiss Federal Office of Energy (SFOE). (2023, January). *Wärmestrategie 2050* (tech. rep.). Swiss Federal Office of Energy (SFOE). Bern.

Thangamuthu, A., Kumar, G., Bishnoi, S., Bhattoo, R., Krishnan, N. M. A., & Ranu, S. (2022). Unravelling the Performance of Physics-informed Graph Neural Networks for Dynamical Systems. *Advances in Neural Information Processing Systems*, *35*, 3691–3702.

The Federal Assembly of the Swiss Confederation. (2011, December). Federal Act on the Reduction of CO2 Emissions (CO2 Act).

Tol, H. İ. (2021). Development of a physical hydraulic modelling tool for District Heating systems. *Energy and Buildings*, *253*, 111512. https://doi.org/10.1016/j.enbuild.2021.111512

UNFCCC. (2016). The Paris Agreement.

van der Heijde, B., Fuchs, M., Ribas Tugores, C., Schweiger, G., Sartor, K., Basciotti, D., Müller, D., Nytsch-Geusen, C., Wetter, M., & Helsen, L. (2017). Dynamic equation-based thermo-hydraulic pipe model for district heating and cooling systems. *Energy Conversion and Management*, *151*, 158–169. https://doi.org/10.1016/j.enconman.2017.08.072

Vandermeulen, A., van der Heijde, B., & Helsen, L. (2018). Controlling district heating and cooling networks to unlock flexibility: A review. *Energy*, *151*, 103–115. https://doi.org/10.1016/j.energy.2018.03.034

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018, February). Graph Attention Networks. https://doi.org/10.48550/arXiv.1710.10903

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wack, Y., Baelmans, M., Salenbien, R., & Blommaert, M. (2023). Economic topology optimization of District Heating Networks using a pipe penalization approach. *Energy*, *264*, 126161. https://doi.org/10.1016/j.energy.2022.126161

Wack, Y., Serra, S., Baelmans, M., Reneaume, J.-M., & Blommaert, M. (2023). Nonlinear topology optimization of District Heating Networks: A benchmark of a mixed-integer and a density-based approach. *Energy*, *278*, 127977. https://doi.org/10.1016/j.energy.2023.127977

Wang, J., Zhou, Z., & Zhao, J. (2016). A method for the steady-state thermal simulation of district heating systems and model parameters calibration. *Energy Conversion and Management*, *120*, 294–305. https://doi.org/10.1016/j.enconman.2016.04.074

Weber, C., & Favrat, D. (2010). Conventional and advanced CO2 based district energy systems. *Energy*, *35*(12), 5070–5081. https://doi.org/10.1016/j.energy.2010.08.008

Weisskopf, T., Streit, D., Deschaintre, L., & Robineau, J.-L. (2022, August). *Beschleunigung des ausbaus thermischer netze: Projektbericht* (tech. rep.). Federal Department of the Environment, Transport, Energy and Communications (DETEC). Bern.

Wetter, M., van Treeck, C., Helsen, L., Maccarini, A., Saelens, D., Robinson, D., & Schweiger, G. (2019). IBPSA Project 1: BIM/GIS and Modelica framework for building and community energy system design and operation – ongoing developments, lessons learned and challenges. *IOP Conference Series: Earth and Environmental Science*, *323*(1), 012114. https://doi.org/10.1088/1755-1315/323/1/012114

Wetter, M., Zuo, W., Nouidui, T. S., & Pang, X. (2014). Modelica Buildings library. *Journal of Building Performance Simulation*, *7*(4), 253–270. https://doi.org/10.1080/19401493.2013.765506

Witte, F., & Tuschy, I. (2020). TESPy: Thermal Engineering Systems in Python. *Journal of Open Source Software*, *5*(49), 2178. https://doi.org/10.21105/joss.02178

Zeghadnia, L., Robert, J. L., & Achour, B. (2019). Explicit solutions for turbulent flow friction factor: A review, assessment and approaches classification. *Ain Shams Engineering Journal*, *10*(1), 243–252. https://doi.org/10.1016/j.asej.2018.10.007

Zhang, S., Gu, W., Lu, H., Qiu, H., Lu, S., Wang, D., Liang, J., & Li, W. (2021). Superposition-principle based decoupling method for energy flow calculation in district heating networks. *Applied Energy*, *295*, 117032. https://doi.org/10.1016/j.apenergy.2021.117032

Zhang, X., Strbac, G., Shah, N., Teng, F., & Pudjianto, D. (2019). Whole-System Assessment of the Benefits of Integrated Electricity and Heat System. *IEEE Transactions on Smart Grid*, *10*(1), 1132–1145. https://doi.org/10.1109/TSG.2018.2871559

Zheng, J., Zhou, Z., Zhao, J., & Wang, J. (2018). Integrated heat and power dispatch truly utilizing thermal inertia of district heating network for wind power integration. *Applied Energy*, *211*, 865–874. https://doi.org/10.1016/j.apenergy.2017.11.080

# Roberto Boghetti

roberto.boghetti@epfl.ch | github.com/rboghe | linkedin.com/in/roberto-boghetti/

## Education

2026 **Ph.D. in Electrical Engineering**
*École polytechnique fédérale de Lausanne (EPFL)* Lausanne, CH
Ph.D. candidate in the Electrical Engineering Doctoral program.
EPFLglobaLeaders Fellow (MSCA).

2020 **M.Eng. + M.Arch. in Building Engineering and Architecture**
*University of Pisa* Pisa, IT

## Work Experience

2021 – Present **Research Assistant**
*EPFL & Idiap Research Institute* Lausanne & Martigny, CH
Ph.D. position on developing novel graph learning models for the simulation of meshed district heating networks. Advised by Dr. Jean-Marc Odobez and Dr. Jérôme Kämpf. Key contributions:

- Designed a physics-informed, symmetry-preserving GNN for hydraulic state estimation in district heating networks; reached solver-level accuracy on unseen topologies with 85× speedup.
- Built and maintained an open-source library (PyDHN) used commercially to optimize 20+ energy grids.
- Formulated a novel differentiable Lagrangian pipe model; integrated into full grid simulation for gradient-based control optimization.

2023 **Secondment**
*Satom SA* Monthey, CH
Industry secondment in the Innovation & Development team focused on building data-driven pipelines for long-term energy planning under different renovation and climate change scenarios. Main achievements:

- Built a scalable ETL pipeline aggregating 10+ heterogeneous data sources.
- Developed multi-horizon forecasting models achieving <5% annual error on historical validation data; deployed as standalone software informing the company's long-term capital allocation and investment strategy (2024–2050).
- Performed energy grid extension analysis and thermal energy storage integration assessments to support design planning.

2020 – 2021 **Research Intern**
*Idiap Research Institute* Martigny, CH
Research position spanning multiple projects at the intersection of machine learning and energy systems.

- Developed gradient boosting models for real-time predictive control of lights and blinds in buildings; deployed to production in 3 office spaces achieving up to 75% energy savings.
- Built data workflows for 5 district heating simulation tools; performed systematic calibration of energy demand and sensitivity analysis.

2019 **Research Trainee**
*Idiap Research Institute* Martigny, CH
Master's thesis research on urban-scale building energy demand prediction. Main achievements:

- Benchmarked 10 ML algorithms for energy forecasting with novel input features describing urban morphology; outperformed uncalibrated physics-based models.
- Built a parallel computing pipeline for urban-scale energy modelling, cutting simulation time by 97%.

## Teaching & Supervision

### *Teaching Assistantships – EPFL*

Autumn 2024 **EE-554: Automatic Speech Processing**
*Lecturer: Mathew Magimai Doss*

Spring 2024 **EE-311: Fundamentals of Machine Learning** – Lead TA
*Lecturer: Michael Liebling*

Spring 2024 **EE-559: Deep Learning** – Guest TA
Prepared and gave lab session on Graph Neural Networks
*Lecturer: Andrea Cavallaro*

Spring 2022 **EE-311: Fundamentals of Machine Learning**
*Lecturer: Michael Liebling*

### *Guest Lectures*

2021 **Machine Learning for Energy Applications**
*Workshop "Sustainability of Urban Landscapes", Polytechnic University of Turin*
Guest lecture and hands-on lab on machine learning for energy applications.

### *Student Supervision*

Autumn 2023 **Semester Project – EPFL**
*Xiyun Fu: "An exploration on the use of GNNs for the simulation of a District Heating Network"*

2021 **Master's Thesis – University of Pisa**
*Chiara Apruzzese: "Impact of indoor environmental quality in European hospitality facilities: a study for IEQ assessment through text mining"*
Co-supervised external thesis. Original title in Italian.

## Projects

2026 – Present **M-POWER – AI Co-Pilot for Participatory Urban Energy Planning**
*Martigny Research Grant*

2024 – 2025 **IEA EBC Annex 84 – Demand Management of Buildings in Thermal Networks**
*International Energy Agency, Energy in Buildings and Communities Programme*

2023 – 2024 **Perspectives on AI – Workshop Series**
*Idiap Research Institute*

2020 – 2024 **Eguzki – AI-Based District Heating Network Simulation**
*Funded by the Swiss Federal Office of Energy (SFOE)*

2020 – 2023 **LUCIDELES – User-Centric Intelligent Daylight and Electric Lighting for Energy Saving**
*Funded by the Swiss Federal Office of Energy (SFOE)*

## Recent Talks

Apr. 2025 **GNNs: Successes and Challenges in the Analysis of Energy Grids**
*PyData Valais* Idiap, Martigny, Switzerland

Apr. 2025 **GNNs: Intuition, Misconceptions, and Real-World Challenges in Energy Systems**
*Invited talk at Chiesi* Chiesi, Parma, Italy

Mar. 2023 **Graph Cycles and How They Can Keep You Warm Next Winter**
*Idiap Tuesday Afternoon Meeting Series* Idiap, Martigny, Switzerland

## Peer Review

35 reviews for 4 journals: *Energy and Buildings* (21), *Data in Brief* (6), *MethodsX* (5), *Environmental Modelling & Software* (3).

## Skills & Languages

**ML & Deep Learning:** PyTorch, PyTorch Geometric, TensorFlow/Keras, Scikit-learn, XGBoost, LightGBM, Graph Neural Networks, CNNs, Physics-Informed ML, Reinforcement Learning

**Data & Computing:** Python, SQL, Pandas, NumPy, NetworkX, GeoPandas, PostgreSQL, Apache Spark, Docker, Slurm, Git, Weights & Biases, CI/CD Pipelines

**Languages:** Italian (native), English (fluent), French (basic)

## Selected Publications


[1] **R. Boghetti**, J.-M. Odobez, J. H. Kämpf. "Graph Neural Network-Based Surrogate Modeling for Fast and Scalable Simulations of Meshed District Heating Networks." *Available at SSRN 5576535*, 2025.

[2] **R. Boghetti**, J. H. Kämpf. "Optimizing Supply Temperature Control in District Heating Networks via Differentiable Dynamic Simulation and Gradient Descent." *International Conference on Construction, Energy, Environment and Sustainability (CEES)*, Springer Nature Singapore, 2025.

[3] G. Lamberti, **R. Boghetti**, et al. "Analysing the potential of open hotel review databases for IEQ assessment: a text mining approach." *Building Research & Information*, 53(1–2):261–279, 2025.

[4] **R. Boghetti**, J. H. Kämpf. "Verification of an open-source Python library for the simulation of district heating networks with complex topologies." *Energy*, 290:130169, 2024.

[5] G. Lamberti, **R. Boghetti**, et al. "Development and comparison of adaptive data-driven models for thermal comfort assessment and control." *Total Environment Research Themes*, 8:100083, 2023.

[6] M. Papinutto, **R. Boghetti**, et al. "Saving energy by maximising daylight and minimising the impact on occupants: An automatic lighting system approach." *Energy and Buildings*, 268:112176, 2022.

[7] C. Basurto, **R. Boghetti**, et al. "Implementation of machine learning techniques for the quasi real-time blind and electric lighting optimization in a controlled experimental facility." *Journal of Physics: Conference Series*, 2042(1), IOP Publishing, 2021.

[8] **R. Boghetti**, et al. "Understanding the performance gap: A machine learning approach on residential buildings in Turin, Italy." *Journal of Physics: Conference Series*, 1343(1), IOP Publishing, 2019.