Efficient Adaptation for Speech Technology

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the registrar's office.

Thèse n. xxxx 2025
présentée le xx octobre 2025
Faculté des sciences et techniques de l'ingénieur
Laboratoire de l'IDIAP
Programme doctoral en génie électrique
pour l'obtention du grade de Docteur ès Sciences
par

Haolin CHEN

acceptée sur proposition du jury :

Prof A. Popescu-Belis, président du jury
Prof P. Frossard, Dr Ph. N. Garner, directeurs de thèse
Dr M. Rajman, rapporteur
Prof S. King, rapporteur
Dr G. Lecorvé, rapporteur

Lausanne, EPFL, 2025

Acknowledgments

First and foremost, I would like to express my heartfelt gratitude to Professor Dong Wang from the Center for Speech and Language Technology (CSLT) at Tsinghua University. At a time when no one else was willing to take in an inexperienced undergraduate student like me, he kindly accepted me and introduced me to the field of speech processing, marking the beginning of my research journey in this area. I am also deeply thankful to Professor Chao Zhang, who shared the job posting with me when I was applying for PhD positions.

I am especially grateful to my supervisor, Phil. Despite having no precedent for accepting students directly from undergraduate studies, he gave me this opportunity, allowing me to embark on my doctoral study. Over the past four years, Phil has provided me with tremendous academic guidance, helping me grow from a beginner lacking formal research training into an independent researcher. He also offered invaluable advice and support in daily life, helping me find my footing in a foreign country. In research, Phil gave me the freedom to explore areas I was truly passionate about and was always there to help whenever I encountered difficulties. Such mentorship is rare and deeply appreciated. In addition, I would like to thank my thesis director, Pascal, for his administrative support and academic guidance, which ensured the smooth progress of my research.

I would also like to thank the SNSF NAST project for funding my doctoral research. I am sincerely grateful to all the staff and friends at Idiap, whose presence made my time in Martigny both enjoyable and enriching. Above all, I am profoundly thankful to my family for their invaluable emotional and material support throughout this journey.

Finally, I want to thank the members of my thesis jury for taking the time to review my thesis, participate in the defense, and provide valuable feedback.

Martigny, September, 2025 Haolin Chen

Abstract

Adapting generic models to specific domains or tasks, a process termed adaptation, has long been of interest in speech and language processing, particularly when target data are insufficient for training bespoke models from scratch. The pre-training fine-tuning paradigm has underpinned the development and application of such generic models, which are initially trained on extensive datasets before subsequent refinement on domain- or task-specific data. While recent large pre-trained models increasingly demonstrate in-context or zero-shot learning capabilities, adaptation remains crucial for significantly enhancing performance when more target data are available. Primarily motivated by the adaptation of text-to-speech synthesis (TTS) models, in this thesis, we investigate a series of adaptation techniques, including both TTS-specific methods and generic fine-tuning approaches, with particular emphasis on data efficiency, parameter efficiency, and generalizability.

The thesis begins by exploring the integration of diffusion models into adaptive TTS systems, motivated by the recent success of deep generative models in synthesizing realistic speech. Building on the Diffusion Transformer architecture, we utilize adaptive layer normalization to condition the diffusion network on text representations, which further enables parameter-efficient adaptation. Compared to convolutional counterparts, the proposed approach offers faster inference for general TTS tasks and outperforms transformer-based adaptive TTS models in terms of naturalness and speaker similarity under few-shot and few-parameter settings.

The second part shifts from ad hoc adaptation to generic parameter-efficient fine-tuning (PEFT) for TTS systems, which increasingly rely on large pre-trained models with strong zero-shot capabilities. Despite PEFT enabling efficient adaptation, catastrophic forgetting remains an issue, damaging the base model's generalizability. To mitigate this, we apply Bayesian transfer learning techniques to regularize PEFT with low-rank adaptation (LoRA) and preserve pre-training knowledge, utilizing diagonal and Kronecker-factored Laplace approximations. Experiments on language modeling and TTS demonstrate that catastrophic forgetting can be overcome by our methods without degrading fine-tuning performance, with Kronecker-factored approximation yielding superior pre-training knowledge preservation.

Continuing the exploration of Bayesian learning theory from the previous part, the final part of this thesis investigates the applications of variational inference to PEFT. Unlike Laplace approximation, variational inference frames posterior estimation as an online optimization problem, allowing for more flexible and expressive distributions. We first assess its effectiveness in improving predictive accuracy and calibration relative to Laplace-based methods. We then leverage its online posterior estimates to identify and prune redundant LoRA components, enabling automatic, layer-wise allocation of the parameter budget.

In summary, the thesis contributes to the advancement of adaptive TTS systems and offers Bayesian perspectives on enhancing generic adaptation techniques with respect to generalizability and efficiency. In particular, it provides a principled investigation of posterior estimation for adapted parameters using both Laplace approximation and variational inference, highlighting the advantages of Bayesian learning in fine-tuning.

Keywords: speech synthesis, deep generative models, adaptation, parameter-efficient fine-tuning, Bayesian transfer learning, Laplace approximation, variational inference

Résumé

L'adaptation de modèles génériques à des domaines ou à des tâches spécifiques, un processus appelé adaptation, suscite depuis longtemps un intérêt dans le traitement de la parole et du langage, en particulier lorsque les données cibles sont insuffisantes pour entraîner des modèles sur mesure depuis zéro. Le paradigme d'apprentissage par pré-entraînement puis affinement (fine-tuning) a sous-tendu le développement et l'application de ces modèles génériques, qui sont d'abord entraînés sur de vastes ensembles de données avant d'être affinés sur des données spécifiques à un domaine ou à une tâche. Bien que les récents modèles de grande taille pré-entraînés démontrent de plus en plus des capacités d'apprentissage en contexte ou sans entraînement préalable (zero-shot), l'adaptation reste cruciale pour améliorer significativement les performances lorsque davantage de données cibles sont disponibles. Principalement motivée par l'adaptation des modèles de synthèse texte-vers-parole (TTS), cette thèse explore une série de techniques d'adaptation, incluant à la fois des méthodes spécifiques au TTS et des approches génériques d'affinement, avec un accent particulier sur l'efficacité en termes de données, l'efficacité paramétrique, et la généralisabilité.

La thèse commence par explorer l'intégration des modèles de diffusion dans les systèmes TTS adaptatifs, motivée par le succès récent des modèles génératifs profonds dans la synthèse de parole réaliste. En s'appuyant sur l'architecture Diffusion Transformer, nous utilisons une normalisation adaptative des couches (adaptive layer norm) pour conditionner le réseau de diffusion sur des représentations textuelles, ce qui permet une adaptation efficace en termes de paramètres. Comparée aux approches convolutionnelles, l'approche proposée offre une inférence plus rapide pour les tâches TTS générales et surpasse les modèles TTS adaptatifs à base de transformers en termes de naturel et de similarité de locuteur dans des contextes à faible nombre d'exemples et de paramètres.

La seconde partie passe d'une adaptation ad hoc à un affinement générique efficace en paramètres (PEFT) pour les systèmes TTS, qui reposent de plus en plus sur des modèles pré-entraînés de grande taille avec de fortes capacités zero-shot. Bien que le PEFT permette une adaptation efficace, l'oubli catastrophique reste un problème, nuisant à la généralisabilité du modèle de base. Pour y remédier, nous appliquons des techniques d'apprentissage transféré bayésien afin de régulariser le PEFT avec une adaptation à faible rang (LoRA) et de préserver les connaissances issues du pré-entraînement, en utilisant des approximations de Laplace diagonales et factorisées de Kronecker. Les expériences en modélisation du langage et en TTS démontrent que l'oubli catastrophique peut être évité par nos méthodes sans dégrader les performances d'affinement, l'approximation factorisée de Kronecker assurant une meilleure préservation des connaissances acquises lors du pré-entraînement.

Poursuivant l'exploration de la théorie bayésienne initiée dans la partie précédente, la dernière partie de cette thèse étudie les applications de l'inférence variationnelle au PEFT. Contrairement à l'approximation de Laplace, l'inférence variationnelle reformule l'estimation du postérieur comme un problème d'optimisation en ligne, permettant des distributions plus flexibles et expressives. Nous évaluons d'abord son efficacité en termes d'amélioration de la précision prédictive et de la calibration par rapport aux méthodes basées sur Laplace. Nous exploitons ensuite ses estimations postérieures en ligne pour identifier et élaguer les composantes LoRA redondantes, permettant une allocation automatique et couche-par-couche du budget de paramètres.

En résumé, cette thèse contribue à l'avancement des systèmes TTS adaptatifs et propose des perspectives bayésiennes pour améliorer les techniques d'adaptation générales en termes de généralisabilité et d'efficacité. En particulier, elle offre une analyse rigoureuse de l'estimation a posteriori des paramètres adaptés en utilisant à la fois l'approximation de Laplace et l'inférence variationnelle, mettant en évidence les avantages de l'apprentissage bayésien pour l'ajustement fin.

Mots-clés : synthèse vocale, modèles génératifs profonds, adaptation, affinement efficace en paramètres, apprentissage transféré bayésien, approximation de Laplace, inférence variationnelle

List of Figures

List of Tables

Acronyms

ACC	Accuracy.
adaLN	Adaptive Layer Normalization.
AdaLoRA	Adaptive Low-Rank Adaptation.
AM	Acoustic Modeling.
ASR	Automatic Speech Recognition.
BNN	Bayesian Neural Network.
CER	Character Error Rate.
CNN	Convolutional Neural Network.
DCR	Degradation Category Rating.
DDPM	Denoising Diffusion Probabilistic Models.
DGM	Deep Generative Model.
DiT	Diffusion Transformer.
ECE	Expected Calibration Error.
ELBO	Evidence Lower Bound.
EWC	Elastic Weight Consolidation.
FIM	Fisher Information Matrix.
FT	Fine-tuning.
G2P	Grapheme-to-Phoneme.
GAN	Generative Adversarial Network.
GLUE	General Language Understanding Evaluation.
GPU	Graphics Processing Unit.
GT	Ground Truth.
ITU	International Telecommunication Union.
IVON	Improved Variational Online Newton.

KFAC	Kronecker-Factored Approximate Curvature.
KL	Kullback-Leibler Divergence.
LA	Laplace Approximation.
LLA	Linearized Laplace Approximation.
LLLA	Last-Layer Laplace Approximation.
LLM	Large Language Model.
LoRA	Low-Rank Adaptation.
LSTM	Long Short-Term Memory.
MAP	Maximum A Posteriori.
MC	Monte Carlo.
MCD	Mel-Cepstral Distortion.
MFA	Montreal Forced Aligner.
MOS	Mean Opinion Score.
MUSHRA	Multiple Stimuli with Hidden Reference and Anchor.
NLL	Negative Log-Likelihood.
NLP	Natural Language Processing.
OOD	Out-of-Domain.
PEFT	Parameter-Efficient Fine-Tuning.
PESQ	Perceptual Evaluation of Speech Quality.
POS	Part-of-Speech.
PPL	Perplexity.
PT	Pre-trained.
RNN	Recurrent Neural Network.
RTF	Real-Time Factor.
SECS	Speaker Embedding Cosine Similarity.
SGM	Score-Based Generative Models.
SI-SDR	Scale-Invariant Signal-to-Distortion Ratio.
SMOS	Similarity Mean Opinion Score.
SNR	Signal-to-Noise Ratio.
STOI	Short-Time Objective Intelligibility.
SVD	Singular Value Decomposition.
TTS	Text-to-Speech.

VAE	Variational Autoencoder.
VI	Variational Inference.
WER	Word Error Rate.

Contents

1 Introduction

1.1 Motivation

The rapid advancement of speech technology has revolutionized human-computer interaction, giving rise to a wide range of applications, including voice assistants, audiobooks, and speech-to-speech translation. This progress can be primarily attributed to two interrelated factors: innovations in modeling techniques and the increasing scale of training data and model capacity. From a methodological perspective, advances in deep generative models, such as diffusion models (Ho et al., 2020; Song et al., 2021) and flow matching (Lipman et al., 2023), have enabled the synthesis of highly realistic speech signals; meanwhile, speech language models (Défossez et al., 2024; Xu et al., 2025) powered by autoregressive models such as transformers have demonstrated strong multi-functional capabilities on both generative and discriminative tasks when multimodal data is unified within a shared discrete token space. On the scaling front, the growth of training data has necessitated the development of increasingly large models, which, once exceeding certain thresholds, often exhibit emergent capabilities not present in smaller counterparts.

Independent of modeling methods, a persistent challenge in the development of speech technology is the need for customization and personalization across various applications, i.e., tailoring speech processing systems to individual users or specialized domains. For example, in text-to-speech synthesis (TTS), users may wish to generate speech in a particular voice, such as their own, or in a speaking style that is either underrepresented or entirely absent from the training data. Likewise, in automatic speech recognition (ASR), service providers often aim to extend the functionality of a general-purpose ASR system to better serve specific user groups, such as children or speakers of a particular language or dialect. These scenarios give rise to the challenge of adapting a generic model to a specific domain or task, a process commonly referred to as adaptation, especially in contexts where the available data is too limited to train a dedicated model from scratch. A key consideration in adaptation is efficiency, both in terms of data usage and parameter modification. The objective is to achieve high performance with limited adaptation data by updating only a small subset of the base model's parameters, thereby minimizing data requirements and computational overhead.

Funded by the Swiss National Science Foundation (SNSF) project, Neural Architectures for Speech Technology, we are primarily motivated by the adaptation of TTS systems to specific speaker identity, speaking style, and emotion. Adaptation methods are closely tied to model architectures; in the context of TTS, this primarily concerns the acoustic model. Early neural TTS acoustic models were characterized by sequence-to-sequence models with an encoder-decoder architecture (Shen et al., 2018; Ren et al., 2021a). Research in this era generally followed two directions: designing specialized methods to improve the generalization of base models across domains, and enhancing adaptation efficiency by minimizing data and parameter requirements. The first direction often involves domain-specific, ad hoc techniques due to the heterogeneity of model architectures (Wang et al., 2018; Hsu et al., 2019); we argue that the trend toward unified model architectures will reduce such complexity. The latter, which forms the basis of the initial phase of this thesis, focuses on identifying adaptable components or integrating dedicated modules to enable efficient adaptation (Chen et al., 2021; Huang et al., 2022b).

Nevertheless, recent developments in the field have motivated a reorientation of our research objectives. First, there has been a growing interest in general-purpose models shifting from task-specific ones, largely driven by innovations in modeling techniques. In TTS, large pre-trained models are now able to not only deliver human-level natural speech, but also support advanced functions such as zero-shot voice cloning and speech editing (Wang et al., 2023a; Li et al., 2023; Huang et al., 2024). Second, architectural unification across domains has led to the adoption of generic adaptation techniques (Ding et al., 2023a), many originating from natural language processing (NLP), replacing earlier ad hoc approaches. Consequently, the distinction among different adaptation targets has diminished, with adaptation framed as a general transfer learning problem. Third, the pre-training fine-tuning paradigm has become fundamental in model development, leveraging large-scale data for general model training, followed by task-specific refinement. These developments highlight the need to explore general adaptation methods suited to this new paradigm in the second phase of the thesis, among which parameter-efficient fine-tuning (PEFT) has emerged as a promising approach.

PEFT techniques aim to adapt large pre-trained models to new tasks or domains by modifying a small fraction of parameters or adding lightweight components while keeping most of the model frozen. This reduces computation, memory, and storage costs, enabling efficient customization on low-resource devices and simplifying the deployment of multiple task-specific variants from a shared base model. While PEFT has significantly enhanced adaptation efficiency, it still presents several problems requiring further investigation, many of which are inherent to transfer learning. A first concern is the potential loss of generalizability: the model may lose much of the knowledge it gained during pre-training. This loss can adversely affect the model's ability to generalize to unseen data, and is even more unfavorable on modern large pre-trained models that are usually multi-functional by training on a diverse range of tasks and data. A second issue is model overconfidence: given limited adaptation data, the model may produce erroneous predictions with disproportionately high confidence, thereby undermining reliability and posing risks in real-world applications. Finally, instead of relying on predefined strategies for parameter modification, automatically identifying the most critical modules or layers for adaptation can further optimize performance and efficiency. In light of these challenges, this thesis seeks to develop a unified and theoretically grounded framework for model adaptation.

1.2 Main Contributions

This thesis contains three broad contributions across two primary phases, each addressing key challenges from ad hoc to generalized approaches.

The first phase of this thesis is situated in the era of encoder-decoder models for acoustic modeling in TTS, during which the integration of deep generative models such as flow and diffusion models as decoders has substantially improved the quality and naturalness of synthesized speech. Within this context, we aim to design an architecture that not only generates high-quality, natural-sounding speech but also enables efficient adaptation in low-resource settings, both in terms of data and model parameters. Motivated by the success of diffusion models in synthesizing realistic speech, we investigate how diffusion can be included in adaptive TTS systems. Inspired by the adaptable layer norm modules for transformer, we adapt the Diffusion Transformer architecture as a new backbone of diffusion models for acoustic modeling. Specifically, the adaptive layer norm is used to condition the diffusion process on text representations, which further enables parameter-efficient adaptation. We show the new architecture to be a faster alternative to its convolutional counterpart for general TTS, while demonstrating a clear advantage on naturalness and similarity over the transformer for few-shot and few-parameter adaptation. To formally evaluate our system against state-of-the-art approaches, we submitted an entry to the Blizzard Challenge 2023 which focused on French TTS. Our submission utilized the proposed model, with an additional focus on text analysis specifically addressing liaisons and heterophonic homographs. Formal evaluations ranked our system favorably among competitors, demonstrating its ability to achieve state-of-the-art performance in terms of synthesis quality and naturalness.

The second phase of this thesis transitions from model-specific adaptation techniques to more general PEFT frameworks. The first focus within this phase addresses the issue of catastrophic forgetting, where fine-tuning undermines the pre-trained model's inherent capabilities. In TTS, this issue manifests as a loss of zero-shot synthesis performance, eventually compromising generalizability and overall synthesis quality. To overcome catastrophic forgetting, we investigates the application of Bayesian transfer learning within the PEFT paradigm. At the core of this approach is the estimation of the posterior distribution over pre-trained model parameters using the Laplace approximation. This posterior distribution acts as a regularizer during adaptation, guiding updates in a manner that preserves the information acquired during pre-training. We demonstrate that existing Bayesian transfer learning techniques can be applied to PEFT to prevent catastrophic forgetting provided that the parameter shift is differentiable and therefore amenable to gradient-based optimization. In a principled series of experiments on language modeling and speech synthesis tasks, we utilize established Laplace approximations, including diagonal and Kronecker-factored approaches, to regularize PEFT with low-rank adaptation (LoRA) and compare their performance in pre-training knowledge preservation. Our results demonstrate that catastrophic forgetting can be overcome by our methods without degrading the fine-tuning performance, and using the Kronecker-factored approximation produces a better preservation of the pre-training knowledge than the diagonal ones.

Continuing the exploration of Bayesian learning theory, the final component of the thesis examines the applications of variational inference to PEFT. Sharing the ultimate goal of learning parameter distributions with Laplace approximation, variational inference formulates posterior estimation as an optimization problem, allowing for the learning of more expressive and accurate posterior along the training process. Utilizing Improved Variational Online Newton (IVON), a state-of-the-art variational inference optimizer, we first assess its effectiveness in improving predictive accuracy and calibration relative to Laplace-based methods. By sampling from the learned parameter distribution during inference, both IVON and Laplace-based method are shown to significantly improve calibration and reduce overconfidence. We then leverage IVON's online posterior estimates to identify and prune redundant LoRA components, enabling automatic, layer-wise allocation of parameter budget. This not only enhances performance and efficiency but also offers a Bayesian interpretation of importance scoring strategies commonly used for parameter selection in PEFT.

1.3 Thesis Outline

This thesis is organized into six chapters, with the main contributions presented in Chapters 3 to 5. The current chapter introduces the motivation, contributions, and structure of the thesis.

Chapter 2 provides the necessary background, including an overview of deep generative models foundational to modern TTS systems, the evolution of neural TTS architectures, a summary of parameter-efficient fine-tuning (PEFT) techniques, and a review of the datasets and evaluation metrics used throughout the thesis.

Chapter 3 presents the first contribution: the integration of the Diffusion Transformer architecture for adaptive TTS using adaptive layer normalization. This design enables both data-efficient and parameter-efficient adaptation. The chapter also details our submission to the Blizzard Challenge 2023 and reports the corresponding evaluation results.

Chapter 4 introduces the second contribution. It includes a thorough mathematical derivation of the Bayesian transfer learning theory using Laplace approximation, which provides a unified framework for overcoming catastrophic forgetting in PEFT. The chapter validates the approach through systematic experiments in language modeling and speaker adaptation for TTS.

Chapter 5 describes the third contribution, which explores the applications of variational inference in PEFT for predictive uncertainty estimation and parameter importance estimation. The first part compares variational inference with Laplace-based methods in terms of improving predictive accuracy and calibration. The second part leverages online posterior estimates to guide parameter selection and improve adaptation performance and efficiency.

Chapter 6 concludes the thesis and outlines directions for future research.

Note: To improve the clarity and fluency of the written text, large language models, including OpenAI's ChatGPT, Google's Gemini, and DeepSeek, were employed during the writing process of this thesis. These tools were used solely for language refinement, such as enhancing grammar, style, and phrasing.

2 Background

2.1 Deep Generative Models

2.1.1 Variational Autoencoder

Variational Autoencoder (VAE) (Kingma and Welling, 2014) is a type of deep generative model that operate within the framework of probabilistic graphical models and variational inference. The primary objective of a VAE is to learn the underlying probability distribution $p(\mathbf{x})$ of training data, enabling both the generation of new data samples resembling the training data and the learning of meaningful low-dimensional latent representations.

VAEs achieve this by positing a generative process involving unobserved, continuous latent variables $\mathbf{z}$. It is assumed that the data $\mathbf{x}$ is generated from $\mathbf{z}$ according to some conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$, often referred to as the decoder, parameterized by $\boldsymbol{\theta}$. The latent variables themselves are assumed to follow a prior distribution $p(\mathbf{z})$, typically chosen to be a simple distribution like the standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The marginal likelihood of the data is then given by the integral:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})\, d\mathbf{z} \tag{2.1}$$

Directly maximizing this marginal likelihood $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ with respect to $\boldsymbol{\theta}$ is generally intractable for complex neural networks used for $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$. VAEs address this challenge by introducing an encoder, denoted by $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, parameterized by $\boldsymbol{\phi}$, which serves as an approximation to the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. Instead of maximizing the marginal log-likelihood directly, VAEs optimize a lower bound known as the Evidence Lower Bound (ELBO), $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x})$, derived using variational principles:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})] - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) || p(\mathbf{z})) \tag{2.2}$$

Here, $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]$ represents the expected reconstruction log-likelihood under the approximate posterior. This term encourages the decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ to accurately reconstruct the input data $\mathbf{x}$ from latent representations $\mathbf{z}$ sampled according to the encoder's output distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. The second term, $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$, is the Kullback-Leibler (KL) divergence between the approximate posterior and the prior $p(\mathbf{z})$. This term acts as a regularizer that encourages the distribution of encoded representations $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ for a given $\mathbf{x}$ to remain close to the prior distribution $p(\mathbf{z})$, thereby promoting a structured latent space.

The gap between the true log-likelihood and the ELBO is the KL divergence between the approximate and true posterior: $\log p_{\boldsymbol{\theta}}(\mathbf{x}) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))$. Maximizing the ELBO thus corresponds to simultaneously maximizing the reconstruction likelihood and minimizing the divergence between the approximate posterior and the prior, which implicitly minimizes the divergence between the approximate and the true posterior.

Both the encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and the decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ are typically implemented using deep neural networks, which are jointly optimized using gradient descent on the negative ELBO. A key technique enabling such optimization is the reparameterization trick, which allows gradients to backpropagate through the sampling process from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. For instance, if $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x}))$, a sample $\mathbf{z}$ can be drawn as $\mathbf{z} = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}) + \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{x}) \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ denotes element-wise multiplication. This reparameterization makes the expectation term in the ELBO differentiable with respect to $\boldsymbol{\phi}$.

2.1.2 Normalizing Flow

Flow-based generative models explicitly model the data distribution by leveraging normalizing flows. A normalizing flow applies a sequence of invertible transformations to map a simple prior distribution $p(\mathbf{z})$ to a complex data distribution $p(\mathbf{x})$, using the change-of-variable law of probabilities. These invertible functions, denoted by $\mathbf{f}$, are referred to as flow steps:

$$\mathbf{x} = \mathbf{f}_1 \circ \mathbf{f}_2 \circ \ldots \mathbf{f}_K(\mathbf{z}) \tag{2.3}$$

Thanks to the invertibility of each flow step, the exact log-likelihood of data can be computed analytically via the change-of-variable formula:

$$\begin{aligned} \log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \sum_{i=1}^{K} \log\left|\det\left(\mathbf{J}\left(\mathbf{f}_i^{-1}(\mathbf{x})\right)\right)\right| \\ \mathbf{z} &= \mathbf{f}_K^{-1} \circ \mathbf{f}_{K-1}^{-1} \circ \ldots \mathbf{f}_1^{-1}(\mathbf{x}) \end{aligned} \tag{2.4}$$

Here, $\mathbf{J}$ denotes the Jacobian matrix of the inverse transformation $\mathbf{f}_i^{-1}(\mathbf{x})$. In practice, the flow steps are parameterized by neural networks, and the model is trained by minimizing the negative log-likelihood of the data.

The key design considerations in constructing normalizing flows are twofold: first, each $\mathbf{f}$ must be invertible and differentiable to ensure tractable computation of the transformed density; second, the Jacobian determinant must be computationally efficient to evaluate. Prominent examples of flow-based architectures include NICE (Dinh et al., 2015), RealNVP (Dinh et al., 2017), and Glow (Kingma and Dhariwal, 2018), which are specifically designed to allow efficient computation of both the inverse mapping and the Jacobian determinant in a single forward pass. Normalizing flows offer a key advantage over VAEs by providing exact likelihood calculation, which leads to better likelihood estimates and generation quality, avoiding issues like the blurry reconstructions and posterior collapse commonly observed in VAEs.

2.1.3 Diffusion Model

Diffusion models refer broadly to two classes of generative models: Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020), formulated via Markov chains, and Score-based Generative Models (SGM) (Song et al., 2021), based on stochastic differential equations. Due to space constraints, we focus on DDPM, which offers a more probabilistically grounded formulation.

DDPM consists of two Markov processes: a forward diffusion process that gradually adds noise to the data, and a reverse process that reconstructs data from noise. The forward process transforms a clean data point $\mathbf{x}_0$ into a Gaussian noise sample $\mathbf{x}_T$ over T steps using a predefined noise schedule $\beta_t \in \beta_1, \ldots, \beta_T$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \quad q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \tag{2.5}$$

Defining $\alpha_t = 1-\beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, one can derive a closed-form expression:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right) \tag{2.6}$$

This allows sampling $\mathbf{x}_t$ at any timestep directly from $\mathbf{x}_0$ without iterating through intermediate steps.

The reverse process reconstructs data by gradually denoising $\mathbf{x}_T$ back to $\mathbf{x}_0$ using parameterized Gaussian transitions:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \Sigma_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right), \quad p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \tag{2.7}$$

with $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The reverse transition probability $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ can be parameterized by a neural network $\boldsymbol{\theta}$, and is analytically tractable when conditioned on $\mathbf{x}_0$:

$$\begin{aligned} q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) &= N\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right) \\ \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t, \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{aligned} \tag{2.8}$$

Similar to VAEs, the training objective is to maximize the data likelihood via the evidence lower bound (ELBO). The loss is given by:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_q\left[\mathrm{KL}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right) + \sum_{t=2}^{T} \mathrm{KL}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_{\boldsymbol{\theta}}\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right) - \log p_{\boldsymbol{\theta}}\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)\right] \tag{2.9}$$

Every KL term in the loss function calculates the distance between two Gaussian distribution thus can be computed in closed form. Note that the first term is a constant and not parameterized. By setting $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)$ as a constant and reparameterizing $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right)$ from $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, Ho et al. (2020) demonstrate the problem of learning $\tilde{\boldsymbol{\mu}}_t$ can be converted to estimating the Gaussian noise $\boldsymbol{\epsilon}$ with neural network $\boldsymbol{\theta}$. The loss is then to minimize the difference between the true noise ϵ and the estimated noise:

$$\mathcal{L}_{\text{simple}}(\boldsymbol{\theta}) = \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right\|^2\right] \tag{2.10}$$

Diffusion models generally outperform normalizing flows in generating high-fidelity and diverse samples due to their ability to model complex, multi-modal distributions more effectively. While normalizing flows require invertible architectures and a fixed dimensionality, which can limit their expressiveness, diffusion models employ a multi-step denoising process that gradually transforms noise into data, offering greater flexibility and robustness in capturing intricate data structures.

2.2 Neural Text-to-Speech Architectures

2.2.1 Overview

A neural network-based TTS system generally consists of three main components: text analysis, acoustic model, and vocoder, as is shown in Figure 2.1.

1. The text analysis module transforms input text into linguistic features. For neural TTS, it is largely simplified to text normalization and grapheme-to-phoneme (G2P) conversion: the module first converts character input to standardized word format (e.g. “2025” normalized to “twenty twenty-five”), and then obtains the corresponding phoneme sequence in the second step (e.g. “modern” converted to “M AA1 D ER0 N”). Traditionally, both steps are implemented using rule-based systems and lexicon lookups.

Figure 2.1: Main components of neural TTS system.

Neural network-based methods have been introduced later, framing text normalization and G2P conversion as sequence-to-sequence learning tasks. With more recent language model-based TTS systems, these steps are increasingly abstracted away. Such models typically tokenize input text at the word or character level, allowing the acoustic model to directly learn the mapping from textual input to speech features without explicit intermediate representations.

2. The acoustic model converts linguistic features into acoustic representations, serving as a core component in a TTS system. Traditionally, the most widely used representation has been the mel-spectrogram, which is a continuous, low-dimensional feature that captures pitch, energy, and timbral information using conventional signal processing techniques. However, recent advances in speech representation learning, particularly in tokenization and quantization methods (Hsu et al., 2021; Chen et al., 2022; Zeghidour et al., 2022; Défossez et al., 2023), have enabled the transformation of continuous speech signals into discrete tokens. These discrete representations facilitate unified, multi-modal modeling of text and speech within a single architecture. The design of the acoustic model varies depending on the specific application, the type of acoustic representation employed, and the structure of the overall TTS pipeline. Nevertheless, the central task of acoustic modeling is typically framed as simply a sequence-to-sequence learning problem. Broadly, acoustic models can be categorized into two main types: encoder-decoder models and speech language models.

3. The vocoder synthesizes an intelligible audio waveform from acoustic features. Vocoders can be broadly categorized based on their architectures, including CNN- (van den Oord et al., 2016), RNN- (Kalchbrenner et al., 2018), GAN- (Kong et al., 2020), flow- (Prenger et al., 2019), and diffusion-based (Kong et al., 2021) models. Traditionally, they have been designed as general-purpose converters that transform a mel-spectrograms into a time-domain waveform. However, these architectures are readily adaptable to alternative acoustic representations, including discrete tokens, as such tokens can be easily mapped to continuous feature spaces via embedding lookups. Recent neural audio codecs (Zeghidour et al., 2022; Défossez et al., 2023) adopt an encoder-decoder architecture in which the decoder effectively functions as a vocoder, reconstructing waveform from discrete token sequences. Ultimately, the choice of vocoder architecture is closely tied to the nature of the acoustic representation used in the system.

The acoustic model–vocoder paradigm underlies most TTS systems, including those based on language models. Although fully end-to-end models exist that directly generate speech waveform from text, they typically integrate the functions of both the acoustic model and the vocoder into a unified architecture. For instance, VITS (Kim et al., 2021) incorporates a

VAE, a normalizing flow, and a GAN-based vocoder into a single model, effectively blurring the distinction between the acoustic model and vocoder. In this setup, the acoustic features are represented as latent variables of the VAE instead of mel-spectrograms. Regardless of the overall system design, the acoustic model remains the core component responsible for determining both the content and style of the generated speech, and therefore is the primary focus of our TTS research. In the remainder of this section, we will examine two key approaches of acoustic modeling in detail: encoder-decoder models and speech language models.

2.2.2 Encoder-Decoder Models

A typical encoder-decoder acoustic model comprises two main components: a text encoder, which transforms the linguistic input, such as words, phonemes, or characters, into fixed-dimensional representations, and a decoder, which sequentially generates acoustic features from these representations. Both the encoding and decoding stages can be framed as sequence-to-sequence modeling tasks, and are commonly implemented using architectures such as CNNs, RNNs, or transformers.

A key challenge in this framework is the inherent length mismatch between linguistic and acoustic sequences: for example, determining how many mel-spectrogram frames should correspond to a single phone. Two paradigms that address this challenge are Tacotron 2 (Shen et al., 2018) and FastSpeech 2 (Ren et al., 2021a). Tacotron 2 follows an autoregressive approach: the text encoder is a bidirectional RNN, while the decoder is a unidirectional RNN that learns alignment between text representations and mel-spectrogram frames via an attention mechanism. Similarly, Transformer TTS (Zheng et al., 2020) replaces the RNNs with transformer blocks while retaining the autoregressive nature of the decoder. In contrast, FastSpeech 2 adopts a non-autoregressive approach, utilizing feed-forward transformer blocks in both the encoder and decoder. It introduces a variance adapter between the encoder and decoder, which explicitly predicts the duration of each phone. During inference, the encoder outputs are expanded based on the predicted durations before being passed to the decoder. This adds another requirement of obtaining the alignment, which can be achieved either through external forced alignment tools prior to training, such as the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017), or by learning the alignment dynamically during training using algorithms like monotonic alignment search (Kim et al., 2020).

Further research in the era of encoder-decoder acoustic models focused on improving controllability, expressiveness, and naturalness. Enhancements in controllability and expressiveness are typically introduced in the stages preceding the decoder, where the model learns to encode not only linguistic content but also speaker-specific and stylistic variations. This includes fine-grained control over prosodic features such as pitch, energy, and speaking rate, often achieved through explicit conditioning or learned latent representations (Min et al., 2021; Huang et al., 2022b). Approaches such as global style tokens (Wang et al., 2018), variational autoencoders (Hsu et al., 2019), and reference encoders (Wu et al., 2022; Huang et al., 2022b) have proven effective in capturing and reproducing diverse speaker identities and speaking styles. In contrast, improving naturalness has largely relied on advances in decoder architectures, particularly through the adoption of deep generative models. Flow (Valle et al., 2021; Kim et al., 2020), diffusion (Jeong et al., 2021; Popov et al., 2021), and flow matching models (Mehta et al., 2024) have emerged as powerful alternatives to traditional decoder architectures, capable of modeling the complex distributions of acoustic features with greater precision. These models not only enhance the perceptual quality of synthesized speech by avoiding oversmoothing of mel-spectrograms and artifacts, but also produce more nuanced and natural prosody by modeling complex dependencies across the entire utterance.

2.2.3 Speech Language Models

Recent advances in speech language models are fundamentally enabled by the development of neural audio codecs, which enable efficient discretization of continuous waveform into compact, learnable token sequences. These codec models, such as Soundstream (Zeghidour et al., 2022) and EnCodec (Défossez et al., 2023), decompose speech into discrete or quantized representations, allowing language models to process speech as sequences akin to text. By leveraging tokenized speech representations, speech language models treat speech synthesis as a conditional language modeling problem, where autoregressive architectures such as a transformer generate speech tokens guided by textual input and reference audio. VALL-E (Wang et al., 2023a), a representative model in this paradigm, employs a hierarchical pipeline that generates coarse acoustic tokens first with an autoregressive transformer, followed by residual token predictions using a non-autoregressive transformer. Following this framework, subsequent work aimed to improve cross-lingual capability (Zhang et al., 2023c), alignment accuracy (Xin et al., 2024; Song et al., 2025), and generation efficiency (Chen et al., 2024).

Beyond the VALL-E paradigm, newer models have targeted improvements in quality, efficiency, and controllability of TTS systems. While most LLM-based TTS models rely on discrete tokenization via neural audio codecs, some studies have explored continuous representations within autoregressive frameworks to overcome limitations in audio quality (Meng et al., 2024). For efficiency, techniques such as generating multi-level codebook tokens in a single pass by generating multiple tokens simultaneously at a step have been proposed (Copet et al., 2023). Alternatively, models like SparkTTS (Wang et al., 2025) eliminate the need to generate high-level codebook tokens by first producing fixed-length global tokens that encode speaker attributes, followed by semantic tokens that capture linguistic content. On the controllability front, the multi-modal nature of speech language models enables speech generation to be guided by natural language prompts. For instance, VoiceCraft (Peng et al., 2024) employs neural codec language models and specialized architectures to support precise, text-guided speech editing. Similarly, InstructSpeech (Huang et al., 2024) uses multi-task LLMs trained on paired natural language instructions and speech data to allow fine-grained control over both semantic content and prosodic attributes. More recently, multi-task training has enabled speech language models to unify speech understanding and generation within a single multimodal architecture (Défossez et al., 2024; Xu et al., 2025). These advancements push the field toward highly versatile and general-purpose speech processing systems.

2.3 Parameter-Efficient Fine-Tuning

2.3.1 Overview

Parameter-efficient fine-tuning (PEFT) techniques aim to adapt large pre-trained models to new tasks or domains while minimizing the number of trainable parameters. Instead of updating the entire model, these methods focus on modifying only a small subset of inherent parameters or adding lightweight components while keeping the bulk of the pre-trained model frozen. PEFT not only reduces computation, memory, and storage costs, enabling efficient model customization on low-resource devices, but also facilitates sharing and deployment of multiple specialized model variants derived from a single base model.

Depending on whether the focus is on fine-tuning newly added modules or modifying the intrinsic parameters of a pre-trained model, PEFT techniques can be broadly categorized into addition-based methods and reparameterization-based methods. In the remainder of this section, we introduce several representative techniques from each category.

2.3.2 Addition-Based Methods

Addition-based methods introduce lightweight modules or input modifications to the model while keeping the majority of the pre-trained parameters frozen, which can be further categorized into adapter-based methods and prompt-based methods.

Adapter-Based Methods

Adapter modules (Houlsby et al., 2019; Pfeiffer et al., 2020) are lightweight, trainable components inserted into the transformer architecture. Typically, each adapter consists of a two-layer feed-forward network that forms a bottleneck structure: a down-projection $\mathbf{W}_{\text{down}} \in \mathbb{R}^{d\times r}$ reduces the dimensionality of the hidden representations from the model's hidden size d to a lower-dimensional latent space of rank $r \ll d$, followed by a nonlinearity, and an up-projection $\mathbf{W}_{\text{up}} \in \mathbb{R}^{r\times d}$ that restores the original dimensionality. Formally, given a hidden state $\mathbf{h} \in \mathbb{R}^{d}$, the adapter output is computed as: $\text{Adapter}(\mathbf{h}) = \mathbf{W}_{\text{up}}\sigma(\mathbf{W}_{\text{down}}\mathbf{h}) + \mathbf{h}$, where $\sigma(\cdot)$ denotes the activation function, and the residual connection ensures compatibility with the original model behavior. Adapters can be flexibly inserted at various points within the transformer layer, such as between the self-attention and feed-forward modules, or within residual connections.

Apart from bottleneck adapters, there exist other adapter-based techniques aiming to improve parameter efficiency by introducing inductive biases into adapter layers. For instance, Compacter (Mahabadi et al., 2021a) proposes a method combining hypercomplex multiplication with parameter sharing: the original linear layer is parameterized as a sum of Kronecker products of two smaller matrices. Hyperformer (Mahabadi et al., 2021b) learns adapter parameters by generating them using shared hypernetworks, assuming that there is shared knowledge across layers and tasks.

Prompt-Based Methods

Prompt-based methods modify the model input or internal activations to guide task-specific behavior without altering the original model weights. Two representative methods are prompt tuning (Lester et al., 2021) and prefix tuning (Li and Liang, 2021). Prompt tuning learns a sequence of continuous task-specific embeddings (soft prompts) that are prepended to the input tokens. These embeddings are optimized during training and serve as a lightweight mechanism for conditioning the model. Prefix tuning extends this approach by optimizing continuous vectors that are prepended to the key and value matrices at each transformer layer. Compared to prompt tuning, prefix tuning directly influences the attention mechanism, thereby providing a more expressive form of conditioning.

2.3.3 Reparameterization-Based Methods

Reparameterization-based methods directly alter the parameterization of the pre-trained model without any architectural modifications, either by modifying a subset of existing parameters or by expressing changes in a compact and structured form.

Low-Rank Adaptation

Low-rank adaptation (LoRA) (Hu et al., 2022) hypothesizes that the updates to model parameters during fine-tuning lie in a low-dimensional subspace, i.e., the weight modifications exhibit low intrinsic rank. Accordingly, instead of updating the full weight matrices, LoRA introduces a pair of trainable low-rank matrices $\mathbf{A} \in \mathbb{R}^{d_o \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times d_i}$ to approximate the weight update as $\Delta\mathbf{W} \approx \frac{\alpha}{r}\mathbf{AB}$, where $r \ll \min(d_o, d_i)$ and α is a scaling factor that controls the update magnitude. This parameterization is applied to the weight matrices of the self-attention layers, typically the query and value projections, while the original weights remain frozen. During inference, the low-rank approximation AB is reconstructed and added to the corresponding frozen weight matrix, yielding the adapted weights $\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W}$, with minimal overhead in both storage and computation.

Variants of LoRA can be broadly categorized by their focus on improving efficiency or enhancing adaptation performance. Efficiency-focused variants such as QLoRA (Dettmers et al., 2023) leverage low-bit quantization to reduce memory cost, while VeRA (Kopiczko et al., 2024) uses shared random matrices and learns small scaling vectors to improve parameter efficiency. Variants focused on adaptation performance include AdaLoRA (Zhang et al., 2023b), which dynamically allocates ranks during training based on importance scores, and DoRA (Liu et al., 2024a), which modifies the adaptation mechanism by separating weight updates into magnitude and direction components for better expressiveness.

Adaptation of Inherent Parameters

There are several methods that only fine-tune a subset of existing parameters. Cai et al. (2020) propose to freeze the model weights and updates only the bias parameters during fine-tuning. By avoiding the storage of intermediate activations, this approach achieves significant memory savings. Similarly, Zaken et al. (2022) explore a related technique for pre-trained language models, where only the biases and the final output layer are fine-tuned. Diff Pruning (Guo et al., 2021) reparameterizes the fine-tuned model parameters $\boldsymbol{\theta}$ as the sum of the pre-trained parameters $\boldsymbol{\theta}$ and a difference vector $\Delta\boldsymbol{\theta}$, such that: $\boldsymbol{\theta}' = \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$. To encourage $\Delta\boldsymbol{\theta}$ to be as sparse as possible, diff pruning applies a differentiable approximation of the L_0-norm penalty to regularize $\Delta\boldsymbol{\theta}$ and promote sparsity.

2.4 Datasets

2.4.1 Text-to-Speech Synthesis

LJ Speech

The LJ Speech dataset (Ito, 2017) is an English-language speech corpus consisting of 13,100 audio clips of a single female speaker reading passages from seven non-fiction books. Each clip is paired with a corresponding text transcription. The recordings were captured by the LibriVox project between 2016 and 2017 and provided in 16-bit PCM WAV format at a sampling rate of 22,050 Hz. Clip durations range from approximately 1 to 10 seconds, totaling about 24 hours of audio. The source texts were published between 1884 and 1964 in the public domain.

VCTK

The CSTR VCTK (voice cloning toolkit) corpus (Yamagishi et al., 2019) consists of approximately 44,000 English-language speech recordings produced by 110 speakers with a variety of accents, primarily from the United Kingdom. Each speaker reads around 400 sentences selected from newspaper texts, with recordings captured in a controlled acoustic environment using a 96 kHz sampling rate and 24-bit resolution, later downsampled to 48 kHz for distribution. The dataset provides over 44 hours of recorded speech, with audio files in WAV format accompanied by transcriptions and metadata including speaker accent, gender, and age information.

LibriVox and Its Derivatives

The LibriVox [1] project provides a large collection of public domain audiobooks, primarily in English, recorded by volunteers and used as the basis for several speech corpora. LibriSpeech (Panayotov et al., 2015) is derived from LibriVox recordings and comprises approximately 982 hours of English speech from 2,484 speakers sampled at 16 kHz with corresponding transcriptions, primarily utilized for automatic speech recognition. LibriTTS (Zen et al., 2019), specifically designed for TTS research and built upon LibriSpeech's materials, offers about 585 hours of speech from 2,456 speakers at a higher 24 kHz sampling rate, segmented at sentence boundaries and including both original and normalized texts. LibriLight (Kahn et al., 2020), based on the same source material, comprises over 60,000 hours of unlabeled English speech sampled at 16 kHz and is intended for self-supervised learning.

2.4.2 Natural Language Processing

The Pile

The Pile (Gao et al., 2021) is an 825 GiB open-source English text corpus developed by EleutherAI to train large-scale language models. It comprises 22 diverse, high-quality subsets, including sources like PubMed Central, OpenWebText2, arXiv, GitHub, Stack Exchange, and Wikipedia. The dataset features a wide range of content, from academic papers and legal documents to code repositories and social media discussions. The Pile has been utilized in training various large language models, such as OPT (Zhang et al., 2022) and GPT-NeoX (Black et al., 2022).

Natural Language Understanding Benchmarks

The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019) is a standardized evaluation suite designed to assess the performance of general-purpose natural language understanding models across a diverse set of NLP tasks. It comprises nine distinct English single-sentence and sentence-pair tasks: (1) Corpus of Linguistic Acceptability (CoLA), which evaluates whether a sentence is grammatically acceptable; (2) Stanford Sentiment Treebank (SST-2), a binary classification task for determining the sentiment of movie reviews; (3) Microsoft Research Paraphrase Corpus (MRPC), which identifies whether a pair of sentences are semantic paraphrases; (4) Quora Question Pairs (QQP), which assesses if two questions are semantically equivalent; (5) Semantic Textual Similarity Benchmark (STS-B), which requires predicting a similarity score between sentence pairs; (6) Multi-Genre Natural Language Inference (MNLI), a large-scale dataset for determining whether a premise entails, contradicts, or is neutral with respect to a hypothesis; (7) Question Answering NLI (QNLI), which determines if a Wikipedia sentence contains the answer to a given question; (8) Recognizing Textual Entailment (RTE), a collection of shorter entailment datasets; and (9) Winograd NLI (WNLI), a

[1] https://librivox.org/

small-scale dataset focusing on coreference resolution as a form of natural language inference.

In addition, Chapter 5 utilizes a set of more challenging commonsense reasoning benchmarks to evaluate the fine-tuning performance of large language models: (1) AI2 Reasoning Challenge (ARC) (Clark et al., 2018): a dataset of grade-school science questions that require nontrivial reasoning and knowledge beyond simple pattern recognition; (2) Boolean Questions (BoolQ) (Clark et al., 2019): a binary question-answering dataset where each question is paired with a supporting passage; (3) OpenBookQA (OBQA) (Mihaylov et al., 2018): a multiple-choice question-answering dataset designed to test a model's ability to combine science facts with broad common knowledge; (4) WinoGrande (Sakaguchi et al., 2020): a dataset of coreference resolution tasks by presenting sentences that require sophisticated reasoning to disambiguate pronoun references.

Audio Question Answering Benchmark

The Audio Question Answering task dataset for the DCASE 2025 Challenge[2] (Yang et al., 2025) comprises three curated multiple-choice question-answering (QA) subsets—Bioacoustics QA (BQA), Temporal Soundscapes QA (TSQA), and Complex QA (CQA)—each designed to evaluate distinct dimensions of audio-language understanding and reasoning. BQA focuses on fine-grained auditory grounding in the bioacoustic domain, requiring models to recognize species-specific vocalizations of 31 marine mammals and to reason about their acoustic characteristics and ecological context. The subset includes 700 training and 200 development QA pairs based on recordings from the Watkins Marine Mammal Sound Database, featuring a wide range of sampling rates (600 Hz to 160 kHz) and durations (0.4 seconds to over 10 minutes). TSQA is designed to assess temporal reasoning capabilities by presenting models with questions concerning the classification and temporal structure of overlapping or sequential sound events. It comprises 1k training and 600 development QA pairs, derived from 10-second mono audio clips sampled at 32–48 kHz from multiple public datasets. Each question targets specific temporal relationships such as event ordering, onset and offset detection, and duration estimation. CQA contains 6.4k training and 1.6k development QA pairs and is constructed to test higher-order reasoning over complex, real-world audio scenarios. Based on audio from AudioSet (Gemmeke et al., 2017) and the Mira dataset (Ju et al., 2024), CQA involves multi-faceted questions that require integration of temporal, acoustic, and contextual cues to interpret overlapping events, auditory sequences, and abstract relational patterns.

[2] https://huggingface.co/datasets/PeacefulData/2025_DCASE_AudioQA_Official

2.5 Evaluation Metrics

2.5.1 Text-to-Speech Synthesis

Subjective Evaluation

Mean Opinion Score The Mean Opinion Score (MOS), standardized by ITU-T P.800, is a widely used subjective metric for evaluating the quality of synthesized speech in TTS systems. A group of listeners rate speech samples from various TTS systems on a 5-point scale from 1 (Bad) to 5 (Excellent), based on attributes such as naturalness and intelligibility. The averaged score, and the 95% confidence interval are typically reported to support the interpretation of results and statistical significance. In addition, a variant known as Similarity MOS (SMOS) is used to evaluate how similar the synthesized speech is to a target reference, often in tasks where preserving the original speaker's identity or style is critical.

MUSHRA The Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) test, standardized by ITU-R BS.1534, is a subjective method for evaluating audio quality, particularly effective for systems at intermediate to high-quality levels. In this test, listeners rate a set of stimuli, including the systems under evaluation, a hidden high-quality reference, and one or more degraded anchors, on a continuous scale (0-100). This comparative approach allows for finer distinctions in quality than MOS, helping to detect subtle differences in perceived audio quality among high-fidelity speech synthesis systems.

Preference test The preference test is a subjective method to determine human preference between two or more speech synthesis systems. Listeners are presented with pairs of audio samples from different systems for the same text and are asked to indicate their preference based on criteria like naturalness or overall quality, or to judge them as equally good. Results are reported as the percentage of listeners who preferred one system, with confidence intervals provided to assess the statistical reliability and significance of the differences between systems.

Objective Evaluation

Quality Conventional objective measures of speech quality, such as Mel-Cepstral Distortion (MCD), Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001), Short-Time Objective Intelligibility (STOI) (Taal et al., 2010), and Scale-Invariant Signal-to-Distortion Ratio (SI-SDR) (Roux et al., 2019), are often insufficient for distinguishing the nuanced quality differences produced by modern TTS systems. To address this limitation, recent approaches employ neural networks to predict perceptual scores like MOS, aiming to approximate human judgments. For example, UTMOS (Saeki et al., 2022), developed for the VoiceMOS Challenge 2022 (Huang et al., 2022d), adopts an ensemble approach combining fine-tuned self-supervised models such as wav2vec 2 (Baevski et al., 2020) and WavLM (Chen et al., 2022), and traditional machine learning methods applied to extracted features. TorchAudio-Squim (Kumar et al., 2023) provides a reference-less framework capable of estimating both objective metrics,such as PESQ, STOI, and SI-SDR, and subjective MOS, utilizing deep recurrent neural networks to model sequential features. While such models offer scalable and efficient means of evaluation, they remain surrogate metrics and should be complemented by formal subjective testing for reliable performance assessment.

Intelligibility A practical and increasingly adopted approach to assessing the intelligibility of synthesized speech involves applying automatic speech recognition (ASR) systems to the synthesized audio to generate transcriptions, which are then compared against ground truth text using standard metrics such as word error rate (WER) and character error rate (CER). This method leverages the availability of high-performance open-source ASR models, including wav2vec (Baevski et al., 2020), WavLM (Chen et al., 2022), and Whisper (Radford et al., 2023), enabling objective and scalable intelligibility evaluation without requiring manual transcription. Limitations of this approach include potential bias within the ASR system, misalignment with human perception, and dependency on the ASR model's performance.

Similarity Similar to intelligibility, speaker similarity can be automatically evaluated using speaker verification models that compare synthesized speech with a reference sample and produce a similarity score. High-performing models for this task include ECAPA-TDNN (Desplanques et al., 2020) and self-supervised models such as WavLM (Chen et al., 2022). Beyond speaker identity, this approach can be extended to assess similarity in paralinguistic features such as emotion and speaking style, provided appropriate recognizers are available. For example, the accuracy of emotion recognition can serve as a proxy for evaluating emotional expressiveness in synthesized speech. However, as with ASR-based intelligibility measures, the reliability of these evaluations heavily depends on the performance of the underlying models, and formal subjective assessments remain essential for comprehensive validation.

2.5.2 Natural Language Processing

Perplexity

Perplexity is a commonly used evaluation metric for language models, quantifying how well a model predicts a sample of text. It is defined as the exponentiation of the average negative log-likelihood of a test set, providing an intuitive measure of uncertainty in the model's predictions. For a language model that assigns a probability distribution $P(w_1, w_2, \ldots, w_N)$ to a sequence of words $w_1, w_2, \ldots, w_N$, the perplexity (PPL) is given by: $\text{PPL} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i|w_1, \ldots, w_{i-1})\right)$ where N is the length of the word sequence, and $P(w_i|w_1, \ldots, w_{i-1})$ represents the probability assigned to word w_i conditioned on the preceding words. A lower perplexity indicates a better-performing model, as it reflects greater confidence in its predictions of the next word.

Calibration Measures

In addition to predictive accuracy, three commonly used metrics for assessing calibration are expected calibration error (ECE), negative log-likelihood (NLL), and the Brier score. A well-calibrated model ensures that when it predicts a class with probability p, the actual probability of the prediction being correct is also close to p. Each of these metrics offers a unique perspective on how well the model's confidence aligns with its correctness.

Expected calibration error (ECE) The ECE measures the discrepancy between a model's predicted probabilities and the true empirical probabilities. ECE computes this by partitioning predictions into confidence bins and comparing the mean predicted confidence with the empirical accuracy within each bin. For M bins, ECE is defined as:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|, \tag{2.11}$$

where B_m is the set of samples in bin m, $|B_m|$ is the size of the set, $\mathrm{acc}(B_m)$ is the accuracy within bin m, $\mathrm{conf}(B_m)$ is the average predicted confidence within bin m, and n is the number of samples. Lower ECE indicates better calibration.

Negative log likelihood (NLL) Derived from the likelihood principle, NLL measures how well the predicted class probabilities align with the true labels. For a dataset of n samples, NLL is computed as:

$$\mathrm{NLL} = -\frac{1}{n} \sum_{i=1}^{n} \log p(y_i \mid \mathbf{x}_i), \tag{2.12}$$

where $p(y_i \mid \mathbf{x}_i)$ is the predicted probability for the true class y_i, $\mathbf{x}_i$ is the test input. NLL penalizes models that assign low probabilities to the correct class, reflecting performance in both calibration and discrimination. A lower NLL indicates better correctness and sharpness.

Brier score The Brier score evaluates the mean squared error between predicted probabilities and the true labels. For a classification task with K classes, the Brier score is defined as:

$$\mathrm{Brier} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(p(y_i = k \mid \mathbf{x}_i) - \mathbb{1}(y_i = k)\right)^2, \tag{2.13}$$

where $\mathbb{1}(y_i = k)$ is an indicator function. The score ranges from 0 to 1, with lower values indicating better-calibrated predictions.

3 A Diffusion-Based Adaptive TTS Model

Encoder-decoder models such as Tacotron 2 (Wang et al., 2017) and FastSpeech 2 (Ren et al., 2021a) have significantly advanced acoustic modeling for TTS, enabling high-quality and efficient speech generation. More recently, the integration of flow (Valle et al., 2021; Kim et al., 2020), diffusion (Jeong et al., 2021; Popov et al., 2021), and flow matching models (Mehta et al., 2024) as decoders of acoustic models has further enhanced the quality and naturalness of synthesized speech. In this chapter, we aim to design an architecture that not only generates high-quality, natural-sounding speech but also supports efficient adaptation in low-resource settings, both in terms of data and model parameters.

Building on the success of diffusion in synthesizing realistic speech Jeong et al. (2021); Lee et al. (2022), we investigate how diffusion can be included in adaptive TTS systems. Inspired by the adaptable layer norm modules for transformer, we adapt a new backbone of diffusion models, Diffusion Transformer, for acoustic modeling. Specifically, the adaptive layer norm in the architecture is used to condition the diffusion network on text representations, which further enables parameter-efficient adaptation. We show the new architecture to be a faster alternative to its convolutional counterpart for general TTS, while demonstrating a clear advantage on naturalness and similarity over the transformer for few-shot and few-parameter adaptation.

To evaluate our system against state-of-the-art approaches, we submitted an entry to the Blizzard Challenge 2023 (Perrotin et al., 2023), which focused on TTS for the French language. Our submission utilized the proposed model, with an additional focus on text analysis—specifically addressing liaisons and heterophonic homographs. Formal evaluations ranked our system favorably among competing entries, demonstrating its ability to achieve state-of-the-art performance in terms of synthesis quality and naturalness.

This chapter is a consolidation of the following publications:

Chen, H. and Garner, P. N. (2023a). Diffusion transformer for adaptive text-to-speech. In *12th ISCA Speech Synthesis Workshop, SSW 2023, Grenoble, France, August 26-28, 2023*, pages

157–162. ISCA

Chen, H., He, M., de Gibson, L. C., and Garner, P. N. (2023). The Idiap speech synthesis system for the Blizzard challenge 2023. In *18th Blizzard Challenge Workshop, Grenoble, France, August 29, 2023*. ISCA

3.1 Introduction

Adaptive text-to-speech (TTS) (Wang et al., 2018; Min et al., 2021; Chen et al., 2021; Casanova et al., 2022) aims to synthesize personalized voices of target speakers or speaking styles. In the typical scenario of adaptive TTS, a source acoustic model pretrained on a large multi-speaker corpus is adapted with limited data of the target to synthesize the desired voice. In general, adaptive TTS systems should be well generalizable and adaptable to various speaker traits and acoustic conditions with as few data as possible. Meanwhile, the adapted voice should be of high quality and naturalness, in terms of which deep generative models (Kim et al., 2020, 2021; Liu et al., 2022b) have demonstrated their superiority over previous solutions. In particular, the more recent diffusion models (Liu et al., 2022b; Jeong et al., 2021; Popov et al., 2021) have dominated in terms of quality and naturalness.

While the generalizability and adaptability have been the most important properties of adaptive TTS systems and in many cases interrelated, they can be attributed to different parts of the model or algorithm design. On the one hand, the techniques that improve the ability to generalize to various features in speech signals can be categorized into 1) employing reference encoders to generate representations of the desired attribute of speech on various semantic levels (Chen et al., 2021; Casanova et al., 2022; Huang et al., 2022b), which are normally plugged in before the decoder; 2) learning algorithms that help factorize such representations into expressive components (Wang et al., 2018; Min et al., 2021; Hsu et al., 2019), which are usually combined with reference encoders; and 3) ad hoc designs of the model structure that control desired features (Min et al., 2021; Chen et al., 2021; Choi et al., 2022), which are more model-specific. On the other hand, adaptability, while partly overlapping with the former, emphasizes more the application itself, including considerations of few-data (Chen et al., 2021; Kim et al., 2022), few-parameter (Chen et al., 2021) and zero-shot (Casanova et al., 2022; Wu et al., 2022) scenarios. However, no matter in which concept, there is a clear distinction between generic techniques that fit different backbones, such as reference encoders, and ones with ad hoc architectural designs of the network. The latter are more associated with the adaptability of the model, especially in few-data and parameter-efficient settings. Furthermore, when combined with generic adaptation techniques, such architectures will enable both compute-efficient zero-shot adaptation, and high-quality adaptation when finetuning is performed.

In general, we are interested in integrating adaptable components into diffusion-based acoustic models that add extra adaptability on top of their high-quality synthesis. Despite diffusion models having been well studied for general acoustic modeling, few works have explored them for adaptive TTS systems. Guided-TTS 2 (Kim et al., 2022) utilizes diffusion with classifier guidance to adapt to diverse voices, while lacking parameter efficiency since the whole decoder needs finetuning during adaptation. In Grad-StyleSpeech (Kang et al., 2022), the diffusion mostly works as a post-net that refines the output of an adaptive transformer decoder, and the researchers only tested adapting the whole diffusion post-net in the few-shot setting. Our preliminary study (Chen and Garner, 2023b) shows a convolutional diffusion decoder can be adapted using conditional layer normalization, however, it must be used with adaptive transformer layers to achieve usable adaptation quality. Our search for solutions focuses on the architecture design of the diffusion backbone. Such a design will not only facilitate parameter-efficient adaptation during finetuning, but also has the potential to be combined with a reference encoder to improve the network's generalizability.

In this context, we propose to adapt a novel backbone of diffusion models, Diffusion Transformer (DiT) (Peebles and Xie, 2022), for adaptive TTS. Inspired by the recent innovation in image synthesis and the effectiveness of conditional layer norm (Min et al., 2021; Chen et al., 2021; Wu et al., 2022) in the transformer network, we adapt the DiT's adaptive layer norm to receive a sequence as condition instead of the class embedding to make it suitable for TTS tasks. Through a series of experiments, we demonstrate that 1) for general TTS tasks, the DiT can serve as a substitute backbone for present diffusion decoders in the acoustic model, yielding comparable performance to current designs while providing faster synthesis; 2) for few-shot adaptation, the benefits of the DiT include its capability to perform parameter-efficient adaptation, and its superiority in speech quality and similarity over previous transformer-based solutions; 3) when based on zero-shot adaptation solutions, the DiT can efficiently achieve high-quality adaptation when finetuning is necessary. Audio samples are available [1].

3.2 Diffusion Transformer for TTS

Like other deep generative model-based solutions, a typical diffusion-based acoustic model comprises a transformer text encoder, a variance adapter adopted from FastSpeech 2 (Ren et al., 2021a), and a diffusion-based decoder, as is shown in Figure 3.2a. Essentially, diffusion models generate high-quality and natural samples by denoising a sample from a prior distribution to real data through a diffusion process. In most cases, the learning problem of diffusion can be expressed as learning a denoiser network that predicts the noise in each diffusion step, while other parameterization forms of the denoiser also exist.

3.2.1 Architecture

In principle, the denoiser network takes the sample from the previous step as input to predict the noise in the reverse diffusion process while being conditioned on text representations C and the step embedding t. The network design enjoys flexibility as long as its output has the

[1] https://recherchetts.github.io/dit/

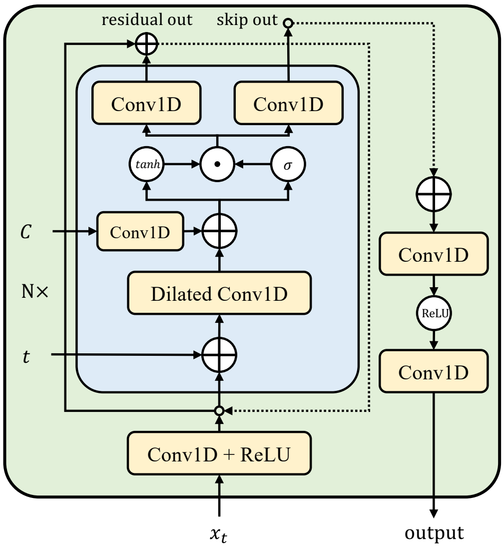
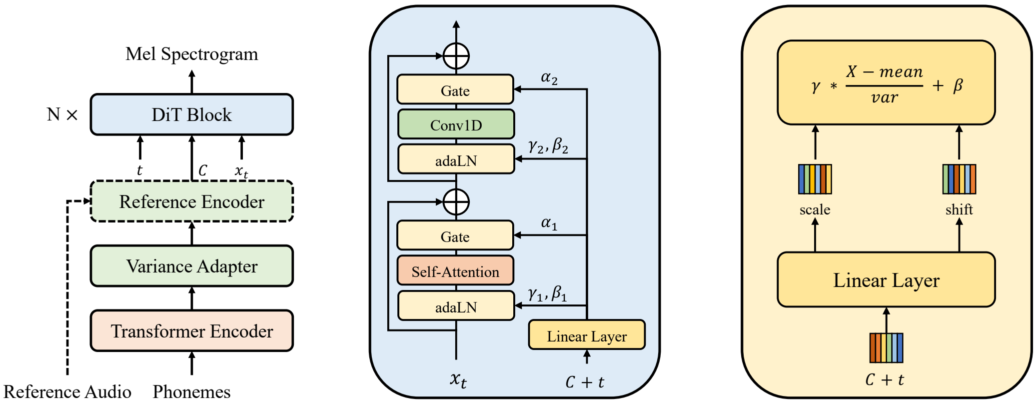


Figure 3.1: The architecture of the non-causal WaveNet-based diffusion backbone network.

same dimension as the input. The prevalent architectures of the denoiser network in acoustic models include the bidirectional dilated convolutional network (CNN) (Jeong et al., 2021; Liu et al., 2022b; Huang et al., 2022c) as is shown in Figure 3.1, also referred to as the non-causal WaveNet (van den Oord et al., 2016), and the U-Net (Popov et al., 2021; Kim et al., 2022). The former is best known for the inductive bias of audio signals and is also commonly used in variational autoencoders (Ren et al., 2021b) and flow models (Kim et al., 2020, 2021; Prenger et al., 2019), while the U-Net (Ronneberger et al., 2015) is a generic network that originates from image processing.

Recently, Peebles and Xie (2022) proposed a new class of diffusion models based on the transformer architecture, namely Diffusion Transformer (DiT), which was shown to outperform U-Net backbones and inherit the scalability, robustness and efficiency of the transformer model class. As is depicted in Figure 3.2b, the DiT blocks receive the sample from the last step as input, perform the common transformations of the transformer and generate the output. The innovation of DiT lies in the way conditions are injected into the network: the standard layer norm modules in the transformer blocks are replaced with adaptive layer norm (adaLN), so that the dimension-wise scale and shift parameters γ and β can be regressed from the sum of the class embedding c and the step embedding t through a linear layer. In addition to adaLN, the authors further propose to zero-initiate the final adaptive layer norm in each block to accelerate convergence, and also regress scaling parameters α that are placed before any residual connections within the DiT block. This is referred to as adaLN-Zero. The authors demonstrate that adaLN-Zero achieves the best performance and adds the least computation cost to the model compared to introducing conditions by in-context learning and cross-attention.

The original DiT was tested on image synthesis tasks, in which only the class embedding



(a) Overview (b) DiT block (c) Adaptive layer norm (adaLN)

Figure 3.2: The architecture of the DiT-based acoustic model. The reference encoder only exists in adaptive TTS systems.

controls the content to be generated. To adapt it for TTS, we make the adaLN-Zero accept a sequence of encoded text representations. In actuality, the implementation of adaLN-Zero requires no modification whatsoever. The novelty lies in the fact that the regression of all scale and shift parameters is now performed on the sum of the text representation matrix C and the step embedding t, generating the necessary scale and shift parameters for each vector in the input sequence, as is shown in Figure 3.2c. Note that the size of the text representation matrix matches that of the hidden representations in the DiT block, since they are both expanded to the length of the mel spectrogram using phoneme durations. Therefore, instead of the same scale and shift vectors applied on the entire input sequence in the affine transform of the layer norm, a sequence of such vectors with the same dimension as the input is applied. This allows the adaptive layer norm to modulate the input sequence using the text representations without adding any computation cost compared to the original adaLN.

3.2.2 Generator-Based Diffusion

Following Chapter 2.1.3, the common parameterization method of diffusion is to let the neural network be a noise predictor. It originates from the reverse diffusion process (Eq. 2.7):

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right) \tag{3.1}$$

where the reverse transition probability $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is parameterized by a neural network $\boldsymbol{\theta}$. By setting $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ to a constant and reparameterizing $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right)$ which is derived from the noise adding function of the forward process (Eq. 2.6): $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, the problem of learning $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ can be converted to estimating the Gaussian noise ϵ, resulting in the simplified loss function (Eq. 2.10):

$$\mathcal{L}_{\text{simple}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right\|^2\right] \tag{3.2}$$

With the diffusion in this form, it usually requires hundreds to thousands of denoising steps to ensure high-quality synthesis.

An alternative way to parameterize the denoiser is to make it directly predict the clean data in each denoising step. Specifically, the neural network $f_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$ that outputs $\mathbf{x}_0$ given $\mathbf{x}_t$ now models the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_0 \mid \mathbf{x}_t)$. Next, $\mathbf{x}_{t-1}$ is sampled using the posterior distribution:

$$\begin{aligned} q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) &= N\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right) \\ \tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t, \\ \tilde{\beta}_t &= \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \end{aligned} \tag{3.3}$$

The rest of the inference process remains the same. The loss is then defined in the data space:

$$\mathcal{L}_{\text{simple}}^{\text{gen}}(\boldsymbol{\theta}) = \mathbb{E}_{t,\mathbf{x}_0}\left[\left\|\mathbf{x}_0 - f_{\boldsymbol{\theta}}\left(\mathbf{x}_t, t\right)\right\|^2\right] \tag{3.4}$$

This parameterization method is sometimes referred to as the generator-based method (Salimans and Ho, 2022; Huang et al., 2022c). Some recent work (Huang et al., 2022c; Liu et al., 2022c) utilizes this method to enable fast synthesis for diffusion-based acoustic models. Huang et al. (2022c) compared the generator-based method with the conventional denoising-based method with varying diffusion steps and found that the former achieved the highest quality in all settings. To accelerate inference while maintaining high synthesis quality, we adopt the generator-based method in our model.

3.2.3 Comparison with Baseline

We first test our model on basic TTS tasks and compare the DiT architecture with the prevalent non-causal WaveNet. We would expect the DiT to perform identically to the baseline in terms of speech quality.

Implementation details Both models consist of a 4-layer transformer phoneme encoder with a hidden size of 256, a variance adapter that is the same as the one in FastSpeech 2 (Ren et al., 2021a), and a diffusion decoder. The DiT network is configured as 4-layer with a hidden size of 256 and 2 attention heads, which is the same as a commonly-used transformer decoder, while the WaveNet network is set to 20-layer with 256 hidden size. Our implementation is based on the open-source software [2] [3] of related models. The numbers of parameters of the WaveNet-based model and the DiT-based model are 30.50M and 28.83M, respectively.

[2]NATSpeech: https://github.com/NATSpeech/NATSpeech
[3]DiT: https://github.com/facebookresearch/DiT

Table 3.1: The MOS scores with 95% confidence interval, SECS and CER scores on LJSpeech, and real-time factors.

Arch.	MOS (↑)	SECS (↑)	CER (↓)	RTF (↓)
Vocoder	4.35 ± 0.10	0.983	1.83%	-
WaveNet	4.06 ± 0.10	0.790	2.41%	0.021
DiT	4.01 ± 0.10	0.784	2.38%	0.012

Data We train the models on the single speaker corpus LJSpeech (Ito, 2017). Two sets of 500 utterances are selected as the validation and test set, while the rest are used as training set. All data are preprocessed following the practice in FastSpeech 2, with a sampling rate of 22,050 Hz.

Training and inference The models are trained on one NVIDIA RTX3090 using a batch size of 40,000 speech frames, with the "rsqrt" (reciprocal of the square root) scheduler, 4,000 warm-up steps, and a learning rate factor of 2. For the diffusion process, a beta schedule of 16 steps is used for both training and inference. A HiFi-GAN (Kong et al., 2020) vocoder trained on LJSpeech is used to synthesize waveforms. The inference is performed on the same hardware.

Evaluation For objective evaluation, we utilize the SpeechBrain (Ravanelli et al., 2021) toolkit to run speaker verification and speech recognition [4] on the entire test set. The averaged speaker embedding cosine similarity (SECS) and character error rate (CER) are calculated as indicators of how well the model captures the speaker identity and the intelligibility of synthesized samples. For subjective evaluation, we recruited 20 native raters on Prolific [5] crowd-sourcing platform to rate the overall quality and naturalness of randomly selected 20 samples from the test set using the P.808 toolkit (Naderi and Cutler, 2020). We also calculate the real-time factor (RTF) of the two models that reflects the synthesis speed, which is conducted when synthesizing around 200 paragraphs.

3.2.4 Results

All test results are listed in Table 3.1. The subjective test results show the DiT architecture has a gap of only 0.05 compared to the non-causal WaveNet within the 95% confidence interval of 0.10 which, consistent with our expectation, suggests the DiT offers a similar synthesis quality to the prevalent architecture. This is also reflected on the two objective test scores, which only demonstrate minor difference between the two architectures.

The RTFs indicate that the model with a DiT backbone is overall 70% faster than the one with

[4] spkrec-ecapa-voxceleb; asr-wav2vec2-librispeech
[5] https://www.prolific.co

a WaveNet backbone, using the model configuration above. By breaking down the time cost into different components, we found that the 4-layer DiT-based decoder has around 2.4 times the speed of a 20-layer WaveNet-based decoder.

Overall, the results of the basic TTS task demonstrate that the DiT is a faster alternative of the diffusion backbone to the non-causal WaveNet, which also shows a slight advantage on the model size. This is perhaps not persuasive enough for switching the diffusion backbone, however, the merit of DiT lies in its ability to be adapted efficiently, which will be elaborated in the next section.

3.3 Adaptive Diffusion Transformer

3.3.1 Method

In the transformer architecture (Vaswani et al., 2017), the layer norm (Ba et al., 2016) helps reduce the variance of the hidden representations after the attention and feed-forward transformation to stabilize and speed up training. Previous work (Min et al., 2021; Chen et al., 2021; Wu et al., 2022) has found that the layer norm in transformer can greatly influence the hidden activation and the final prediction with the learnable scale and shift parameters. Furthermore, these parameters can be regressed from the speaker or style representation, e.g. the speaker embedding, through a small neural network, which can be finetuned during adaptation. The method significantly reduces the number of parameters to be adapted for each new speaker or style, while maintaining high-quality synthesis.

As for DiT, the architecture unification enables us to apply the same method to the adaptive layer norm. Inherently, the adaLN receives all the conditional input to the decoder, including the speaker embedding and possibly embeddings from reference encoders. This cancels the requirement for any additional input to the decoder.

In the following experiments, we compare our adaptive DiT model with AdaSpeech, a transformer-based solution with conditional layer norm. Given the diffusion's superiority in high-quality synthesis, we would expect the DiT to offer better speech quality and speaker similarity compared to the baseline.

3.3.2 Experimental Setup

Implementation details We implement necessary components to construct AdaSpeech using the same TTS framework as before, including the phoneme- and utterance-level encoders in the acoustic condition modeling module and the conditional layer norm in the transformer decoder layers. We use the same acoustic condition modeling module as AdaSpeech, thus the only difference between the DiT-based model and AdaSpeech is the decoder architecture. The model configuration of AdaSpeech follows the official settings, while the DiT follows the previous configuration.

Table 3.2: The subjective and objective test results of few-shot adaptation experiments.

Dataset		**VCTK**				**LibriTTS**	
Metric	**#Params**	**MOS (↑)**	**SMOS (↑)**	**SECS (↑)**	**CER (↓)**	**SECS (↑)**	**CER (↓)**
Vocoder	-	4.37 ± 0.08	-	0.955	3.16%	0.929	2.61%
AdaSpeech	1.184M	2.76 ± 0.08	2.86 ± 0.10	0.505	3.12%	0.508	3.77%
DiT	1.711M	3.77 ± 0.09	3.94 ± 0.10	0.570	2.50%	0.582	3.46%

Data All models are pretrained on the two clean subsets train-clean-360 and train-clean-100 of the multi-speaker LibriTTS dataset (Zen et al., 2019), with a total of 1151 speakers and 245 hours. For adaptation, we use LibriTTS and the multi-speaker corpus VCTK (Yamagishi et al., 2019) to test the in-domain and out-of-domain adaptation performances. For LibriTTS, we select 10 speakers from the `test-clean` subset, and 10 random utterances for each speaker as test set. For VCTK, 11 speakers (7 females and 4 males) with different accents are selected following (Casanova et al., 2022), while for each speaker 10 utterances with the same spoken content across all speakers are selected as test set.

Training, adaptation, and inference Following AdaSpeech, all models are trained in two stages in which the numbers of training steps are 60,000 and 40,000 respectively, on the same hardware as before. The batch size is set to 50,000 speech frames for AdaSpeech and 40,000 for the DiT-based model. Other configurations follow the official or previous settings unless otherwise stated. During adaptation, only the speaker embedding and the layer norm modules are finetuned using 10 random utterances of the target speaker for 2,000 steps using a fixed learning rate of 2×10^{-4}. A HiFi-GAN vocoder trained on VCTK is used to synthesize waveforms.

Evaluation Subjective tests are carried out for the more challenging LibriTTS to VCTK out-of-domain adaptation task. The same 20 native raters are involved in the subjective test to rate the MOS for naturalness and the SMOS (Similarity MOS) for speaker similarity of 22 speaker-balanced samples from the VCTK test set generated by each system. The reference of each utterance given in the subjective test is the vocoder synthesized sample of the utterance. The objective SECS and CER scores are calculated on the entire test sets of both VCTK and LibriTTS. We calculate the number of parameters to be finetuned for each model.

3.3.3 Results and Analyses

The subjective and objective test results are shown in Table 3.2. In the out-of-domain adaptation task, subjective test results demonstrate a clear improvement of both naturalness and speaker similarity by the DiT decoder compared to the transformer. In objective evaluation, the DiT achieves a higher speaker similarity score and a lower character error rate, which indicates the DiT is able to generate more intelligible speech with a voice more similar to the reference. In the in-domain adaptation task, the DiT results in a higher speaker similarity score, while AdaSpeech does not improve much. The DiT has approximately 50% more parameters finetuned compared to the transformer, due to the extra scaling parameters α.

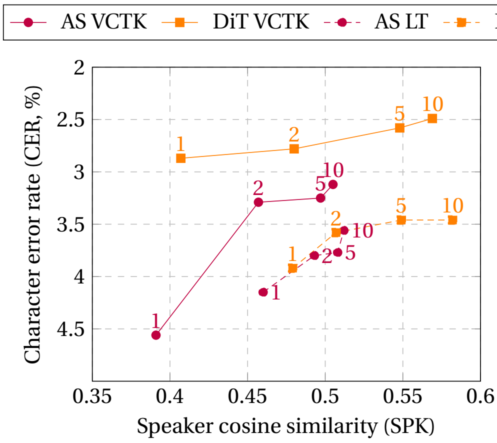


Figure 3.3: The speaker embedding cosine similarity (SECS) and character error rate (CER) of varying adaptation data. The number of utterances used for adaptation is labeled on each data point. AS: AdaSpeech, LT: LibriTTS.

We further study the naturalness and speaker similarity with varying amount of adaptation data on VCTK and LibriTTS, and conduct objective tests. As is shown in Figure 3.3, with increasing number of utterances used for adaptation, the speaker similarity and intelligibility continue to improve for all models and on both datasets. Overall, the DiT outperforms the transformer in both metrics under all settings, and the difference between the two models becomes larger when the more utterances are available.

It is worth noting that, during our test listening of the adapted samples, we found AdaSpeech is more sensitive to the noise in the training data than the DiT, resulting in the adapted samples being more noisy. This is likely due to the low-quality samples in the `train-clean-360` subset, since adapting an AdaSpeech trained on VCTK results in a cleaner voice. Nonetheless, this phenomenon suggests the DiT is more robust against noise, which can be explained with the diffusion's denoising nature.

3.3.4 Zero-Shot Adaptation

Previous experiments have demonstrated that the DiT when adapted is able to generate a more high-quality voice with better similarity to the target compared to the transformer. Although we mainly focus on few-shot adaptation tasks, we are still interested to see how the architecture performs in the zero-shot setting. We also take the chance to demystify what part of the model architecture contributes the most to the generalizability of the model.

We first test the transformer decoder, the DiT decoder, and the non-causal WaveNet-based diffusion decoder on top of the acoustic condition modeling module (the reference encoding) of AdaSpeech. All three models are trained on LibriTTS using the recipe described in Section 3.3. For inference, we randomly select one utterance from the target speaker in the VCTK test set. The objective test results are shown in Table 3.3. It can be observed that the DiT-based and the WaveNet-based diffusion decoders bring similar slight improvements to the speaker similarity compared to the transformer decoder, although all scores are significantly lower than few-shot adaptation. The WaveNet-based diffusion decoder seems to yield better intelligibility than DiT, however both diffusion decoders outperform the transformer.

We further base the two diffusion decoders on a state-of-the-art zero-shot solution, GenerSpeech (Huang et al., 2022b), and its official implementation [6]. All models share the same official training recipe. Note that in GenerSpeech, a flow-based post-net is used on top of the transformer decoder to refine the output. We found the 4-layer DiT in this setting is difficult to converge, hence we use a 6-layer one instead. This time the diffusion does not show much improvement on the speaker similarity compared to the transformer. However, the two diffusion-based models yield notably higher intelligibility which is reflected on the CER, with the WaveNet backbone slightly better than the DiT.

Table 3.3: The objective test results of zero-shot adaptation.

Arch.	**AdaSpeech**		**GenerSpeech**	
Metric	**SECS (↑)**	**CER (↓)**	**SECS (↑)**	**CER (↓)**
Vocoder	0.955	3.16%	0.955	3.16%
Transformer	0.107	2.66%	0.292	6.90%
DiT	0.132	2.34%	0.299	4.43%
WaveNet	0.134	2.20%	0.307	4.06%

Overall, the results suggest that despite the diffusion providing slightly better speaker similarity, the bulk of generalizability lies in the reference encoding part of one adaptive system. Under these certain architectures of the acoustic model, the main benefit of a diffusion decoder in a zero-shot adaptive system is the higher-quality synthesis, rather than better similarity. In comparison with few-shot adaptation, the results also demonstrate the necessity of finetuning to achieve high similarity. On the choice of backbone architecture in the zero-shot setting, the WaveNet seems to slightly outperform the DiT. However, as is discussed above, the adaptive layer norm in the DiT backbone enables the model to be adapted efficiently when finetuning is performed, while the DiT is still a decent alternative to the prevalent non-causal WaveNet in zero-shot usage.

[6] https://github.com/Rongjiehuang/GenerSpeech

3.4 Blizzard Challenge 2023

To formally evaluate the performance of our system relative to other state-of-the-art TTS systems, we submitted an entry to the Blizzard Challenge 2023, which focused on the task of French TTS. Our system follows the conventional pipeline of text analysis, acoustic modeling (AM) and vocoding. For text analysis, open-source pretrained part-of-speech (POS) taggers and lemmatizers are utilized to provide more accurate grapheme-to-phoneme (G2P) conversion on top of eSpeak. The rest of the system incorporates a fully diffusion-based approach which comprises a diffusion transformer-based acoustic model and FastDiff as the vocoder, both of which are trained only on the provided data to ensure high-quality synthesis. Our entry provides a baseline for the cascading diffusion AM-vocoder architecture since no extra design is adopted to enhance the naturalness of speech. Evaluation results have demonstrated high synthesis quality of our system and the effectiveness of the proposed phonemization pipeline.

3.4.1 Introduction

The hub task of the Blizzard Challenge 2023 is to build a voice from the provided French data, which consists of around 51 hours of audiobook recordings read by a female French speaker. The spoke task focuses on speaker adaptation and aims to build a voice from around 2 hours of audiobook recordings read by another female French speaker. The Idiap system was submitted to both the hub task and the spoke task.

The top priority of the text-to-speech (TTS) task is to generate high-quality, natural, and intelligible speech. Since neural networks were first introduced to TTS (van den Oord et al., 2016; Wang et al., 2017), the quality of the synthesized speech has been improved dramatically over the intervening years. In recent years, deep generative model (DGM) based TTS systems (Kim et al., 2020, 2021; Lee et al., 2022) have demonstrated their superiority in high-quality and fast synthesis over previous sequence-to-sequence modeling counterparts (Shen et al., 2018; Ren et al., 2021a; Zheng et al., 2020). In particular, the more recent diffusion-based acoustic models (Jeong et al., 2021; Popov et al., 2021; Lee et al., 2022) and vocoders (Kong et al., 2021; Lam et al., 2022; Huang et al., 2022a) have dominated in terms of quality and naturalness. Since 2023, emerging large-scale pretrained language models (Wang et al., 2023a; Rubenstein et al., 2023) and DGMs (Shen et al., 2023; Le et al., 2023) have revolutionized speech synthesis research in generating human-level natural speech and adapting to the target speaker, speaking style or language with very few data. However, these models are neither open to the research community nor can be trained on normal hardware.

Given the provided data are of sufficient quality and quantity, the challenges mainly lie in how to process liaisons and heterophonic homographs in the language which takes place during the text analysis. In French, liaison refers to the act of pronouncing a linking consonant between two words in a suitable phonetic and syntactic context, which usually gives information about the grammatical structure of a noun phrase. The relatively rare heterophonic homographs refer to words that are spelled the same but pronounced differently, and almost always occur between words of different grammatical categories. These special properties require extra efforts to deliver accurate grapheme-to-phoneme (G2P) conversion in a neural TTS system that uses phoneme input. Available open-source non-neural French phonemizers include the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017), Gruut [7] and eSpeak (also eSpeak-ng) [8]. Among them, the first two only perform G2P on word level and handle neither liaison nor homographs. While the eSpeak is a rule-based phonemizer and handles liaison in many cases, it is unable to distinguish heterophonic homographs at the grammatical level since it does not consider part-of-speech. There are also open-source neural G2P models (Zhu et al., 2022) available for the French language, however these models are normally trained on open-source lexicons that do not usually include liaisons and homographs; this limits their performance in real-life scenarios. For systems that support character input (Shen et al., 2018; Zheng et al., 2020; Kim et al., 2020, 2021), the problem can be solved to some extent by the neural network itself given the corpus covers a wide range of the special cases. However, the use of characters as textual input will largely induce higher computation cost and decelerate training and inference due to longer input length compared to using phonemes.

From the practical point of view, the limited computational resources available to us and the short time frame of the challenge are pertinent. Here at Idiap, the servers are mostly equipped with consumer GPU cards and are not optimized for multi-GPU training, leaving us a limited selection of model architectures. In addition, despite Idiap's being situated in a French speaking region, no dedicated toolboxes or dictionaries have been developed for French TTS in recent years. This requires us to utilize publicly available resources as much as possible to cope with the aforementioned particularities of the French language.

Based on the analyses above, we aimed to build a TTS system that 1) employs accessible model architectures that offer high-quality and natural synthesis, 2) properly handles the special properties of the French language, and 3) can be trained efficiently on our infrastructure to allow fast verification and iteration. Specifically, for text analysis, we leveraged publicly available part-of-speech (POS) taggers and lemmatizers to achieve more accurate G2P conversion on top of the eSpeak backend. For neural architectures, our system adopts a conventional cascading architecture consisting of a diffusion transformer-based acoustic model and FastDiff (Huang et al., 2022a), a diffusion-based vocoder. The acoustic model employs a standard non-autoregressive encoder-decoder design that purely relies on the generative modeling power of the diffusion, which makes our system a baseline of the diffusion-based AM-vocoder architecture. Evaluation results have shown a high quality synthesis achieved by our system and the effectiveness of the text analysis pipeline.

[7] https://github.com/rhasspy/gruut
[8] https://github.com/espeak-ng/espeak-ng

3.4.2 Text Analysis

Liaisons

Liaison in the French language refers to the phonetic linking or connection between words in spoken language. It involves the pronunciation of a consonant sound at the end of a word when the following word begins with a vowel sound. Liaison is a characteristic feature of French pronunciation and helps maintain the smooth flow of speech. In most cases, it is limited to word sequences that have a logical connection in meaning, such as an article followed by a noun, an adjective followed by a noun, a personal pronoun followed by a verb, and similar patterns.

The presence of specific liaison patterns in French makes rule-based phonemization a highly suitable technique, which is exactly the one built into eSpeak. Other types of phonemizers also exist, such as the lexicon-based Gruut. In a lexicon-based phonemizer, words are either looked up in a pre-existing lexicon or their pronunciations are predicted using a pretrained G2P model. However, the word-by-word nature of lexicon-based phonemization necessitates additional rules to handle liaisons between words, which are often unavailable in such systems. Recent advancements in G2P solutions, such as sequence-to-sequence neural networks utilized in (Rao et al., 2015; Zhu et al., 2022), directly predict phonemes from the input text. Nevertheless, the effectiveness of these models heavily relies on the coverage of the training text corpus, limiting their practicality due to the scarcity of high-quality datasets.

Heterophonic Homographs

In general, heterophonic homographs in French are words that are spelled the same but pronounced differently and have different meanings. Fortunately, their existence is relatively rare, and the phenomenon almost always occurs between words of different grammatical categories, which makes it possible to disambiguate by inferring from the grammatical context.

The first step is to understand in what grammatical categories the common homographs exist. Among publicly available resources online, Wiktionary [9] provides a comprehensive list of 813 heterophonic homographs that exist in the French language. In one blog [10] and Hajj et al. (2022), the most common scenarios are summarized and corresponding examples are given. In summary, these scenarios include 1) indicative imperfect first person plural of a verb vs. plural of a noun that end with “-tions”, 2) indicative present third person plural of a verb vs. adjective or noun that end with “-ent”, 3) infinitive of a first group verb vs. nouns that end with “-er”, and 4) miscellaneous cases.

Intuitively, for most cases where words in a pair fall in different grammatical categories, the disambiguation can be done by identifying the part-of-speech of the word. For other cases

[9] https://fr.wiktionary.org/wiki/Cat%C3%A9gorie:Homographes_non_homophones_en_fran%C3%A7ais
[10] https://a3nm.net/blog/frencblizzard_non_homophonous_homographs.html

where the two words belong to the same category, such as “convient” and “pressent”, this can be solved by inferring the original form of the word from the context, i.e., lemmatization, to determine their pronunciations.

Method

Having known the above particularities in the French language, we construct the text analysis module as follows. First, the text input is phonemized by the eSpeak G2P backend. Since eSpeak is able to process liaisons, we only need to refine its corresponding output for homographs considering the grammatical context. To achieve this, we first create a look-up table where different pronunciations of each homographs and the corresponding part-of-speech categories or original forms can be queried, mainly referring to the last two sources mentioned above. During inference, if any homograph in the look-up table exists in the text, we utilize publicly available pretrained POS taggers [11] and lemmatizers [12] to recognize the part-of-speech or the original form of the homograph. Using the inferred information, we refer to the look-up table to obtain the actual phonemes of each homograph. Finally, we compare the phonemes generated by eSpeak with the queried phonemes and rectify the incorrect output.

3.4.3 Neural Architectures

To balance synthesis quality and training efficiency, we employ a cascading diffusion-based architecture consisting of a diffusion transformer acoustic model and the FastDiff vocoder.

Acoustic Model

The acoustic model (Chen and Garner, 2023a) comprises 1) the transformer-based text encoder that encodes phoneme embeddings into hidden representations, 2) the variance adapter that predicts the pitch, energy, and duration of each phoneme and expands the hidden representations to the length of the mel-spectrogram, and 3) the diffusion transformer decoder which generates the mel-spectrogram through a diffusion process. The diffusion transformer is an faster alternative to the most commonly used non-causal WaveNet that offers equivalent synthesis quality.

The architecture of the acoustic model is rather standard: there are no extra components or designs that particularly enhance the naturalness or the speaking style, thus it purely relies on the generative modeling power of the diffusion to render natural speech. We take the chance to see how the standard diffusion architecture performs compared to other more advanced competitors, especially when trained on a highly expressive corpus.

[11] https://huggingface.co/qanastek/pos-french-camembert-flair
[12] https://github.com/explosion/spacy-models/releases/tag/fr_dep_news_trf-3.5.0

Vocoder

FastDiff (Huang et al., 2022a) is a conditional diffusion-based vocoder for high-quality waveform synthesis. The denoiser network employs a stack of time-aware location-variable convolutions with diverse receptive field patterns to model long-term time dependencies. Originally, a noise predictor was further adopted to derive tighter schedules to accelerate inference without distinct quality degradation. However, we found this algorithm is difficult to implement and the derived sampling schedule must be optimized for every dataset, which makes it less favorable for the adaptation task. Therefore, we use the linear schedule instead of the fast schedule. We also found that FastDiff can be trained more efficiently compared to its GAN-based counterparts, which usually require days of training and multiple GPUs.

3.4.4 Experiments

Data

For the hub task, the NEB corpus consists of 289 chapters of 5 audiobooks from Librivox read by a female French speaker Nadine Eckert-Boulet (NEB), totaling 51 hours and 12 minutes. Around two thirds of the utterances are annotated with texts, phonemes and phoneme durations, while the other third has text only. We found the phoneme annotations provided in the dataset lack the tonal and stress marks that are offered by eSpeak, and are likely to be generated by speech recognition models since minor errors can be found. Given the phonemes are unavailable during inference as part of the challenge, and the provided data are insufficient to train a dedicated G2P model, we decide to use eSpeak's phoneme set and run the phoneme-audio alignment using Montreal Forced Aligner (McAuliffe et al., 2017) to obtain the phoneme durations. Two sets of 500 utterances are selected as the validation and test set, while the rest are used as training set. All data are preprocessed following the practice in FastSpeech 2 (Ren et al., 2021a), with a sampling rate of 22,050 Hz.

For the spoke task, the AD corpus consists of 2515 utterances read by another female French speaker Aurélie Derbier (AD), totaling 2 hours and 3 minutes. We randomly select 50 utterances for the validation set and test set respectively, while the rest specifications follow the hub task.

Implementation Details

The model configurations of the acoustic model follow Chen and Garner (2023a), including a 4-layer transformer encoder with 256 hidden size, a variance adapter same as the one in Ren et al. (2021a), and a 4-layer diffusion transformer decoder with 256 hidden size and 2 heads. For the vocoder, we use the official implementation [13] without modification. The number of parameters of the acoustic model is around 29M, while the vocoder has around 13M parameters.

[13] https://github.com/Rongjiehuang/FastDiff

Training and Inference

All experiments are conducted on a single NVIDIA RTX 3090 GPU. For the hub task, the acoustic model is trained using a batch size of 40,000 speech frames for 200k iterations, with the “rsqrt” (reciprocal of the square root) scheduler, 4,000 warm-up steps, and a learning rate factor of 2. For the diffusion process, a beta schedule of 16 steps is used for both training and inference. The vocoder is trained using a batch size of 25,600 samples for 1M iterations, with a constant learning rate of 2×10^{-4}. We use a diffusion schedule of 1000 steps for training and a faster schedule of 200 steps for inference. Both of the acoustic model and the vocoder are trained from scratch, which takes around 1 day and 2 days, respectively. The real-time factor of the entire system is 0.48, in which the acoustic model counts for 0.01 while the vocoder takes up the majority of inference time.

For the spoke task, we finetune the entire acoustic model and vocoder used for the hub task to adapt to the AD voice. Specifically, the acoustic model is finetuned for 20k steps with a learning rate of 2×10^{-4}, while the vocoder is finetuned for 10k steps with a learning rate of 1×10^{-4}.

3.4.5 Results and Analyses

Our system is identified as *T*, whereas *A* represents natural speech, and *BF* and *BT* are two reference systems.

Hub Task: Quality

Our system is ranked the 7th among 18 participants with a mean MOS score of 3.8. Three systems achieved significantly higher synthesis quality compared to ours, while four together with our system yielded comparable results. In the detailed results broken down by the qualification of testers, we found that non-native listeners and non-speech experts tended to give higher scores compared to native listeners and speech experts. The results suggest that despite our system offering high signal quality, it might be at a disadvantage in terms of naturalness. This can be attributed to the lack of more advanced prosody modeling techniques in the acoustic model, since only the conventional variance adapter was used.

Hub Task: Similarity

For the similarity test, the ranking is 9/18 with a mean MOS score of 3.0. Similar patterns can be found in the results breakdown as in the quality test. We also notice that the speaking style of the generated speech can sometimes be distinct from the reference, which can be attributed to the generative modeling nature of the diffusion decoder and the highly variable voice in the audio book. Additional style modeling methods should be introduced to alleviate the issue.

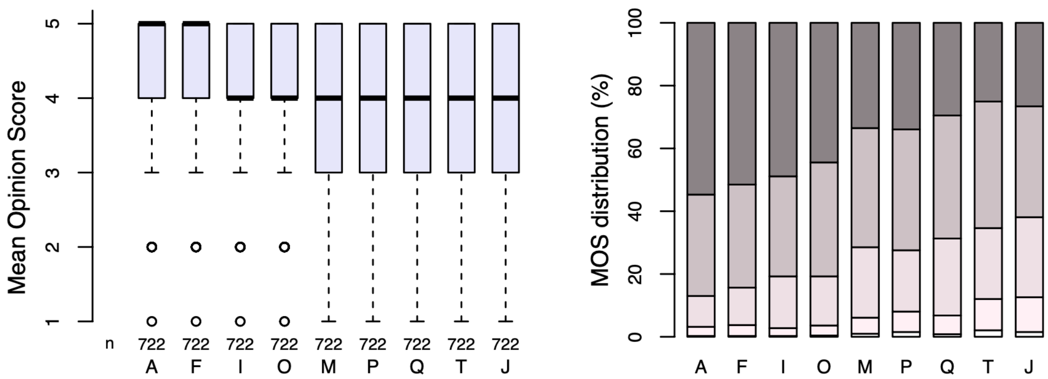


Figure 3.4: MOS results of quality, hub task.

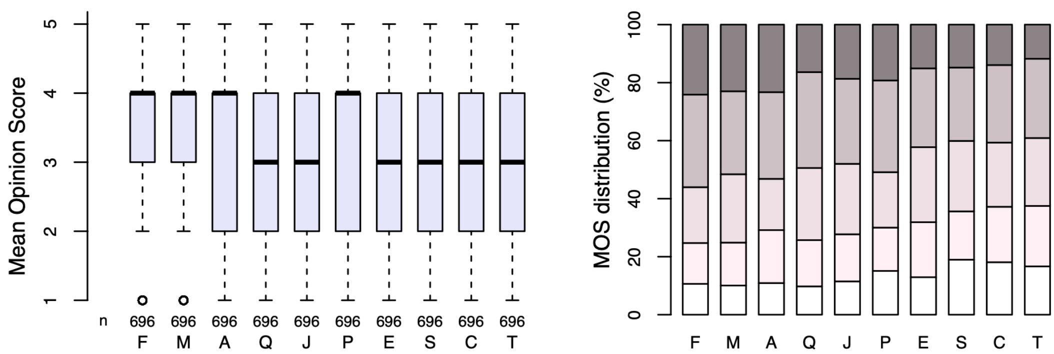
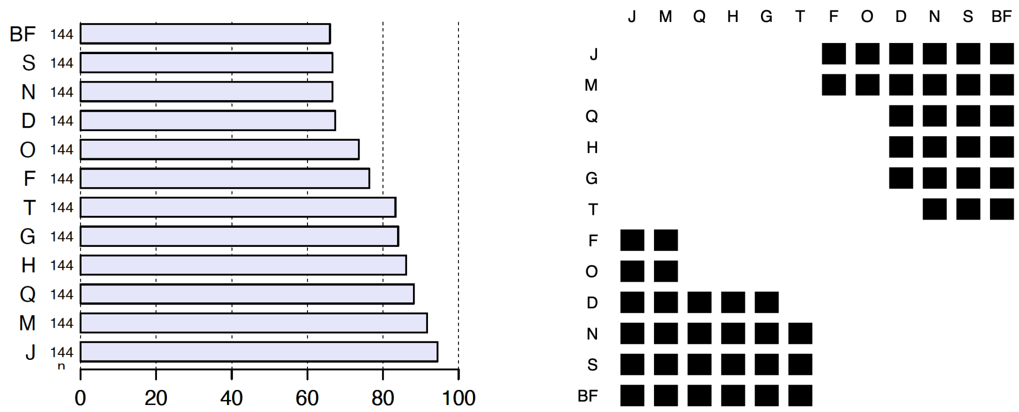


Figure 3.5: MOS results of similarity, hub task.

Hub Task: Intelligibility

In the heterophonic homograph intelligibility test, our system, ranked 6/18, achieves an accuracy of 83% (the percentage of test utterances that are pronounced correctly), which is 17% higher than the reference system *BF* that relies solely on eSpeak. The results demonstrate the effectiveness of our proposed text analysis pipeline. Since our method mainly depends on the POS tagger and lemmatizer to correct the incorrect output of eSpeak, we would expect using more accurate models can further improve the phonemization accuracy.



(a) Pronunciation accuracy (%) (b) Significant difference

Figure 3.6: Intelligibility of heterophonic homographs, hub task.

However, in the conventional intelligibility test, the word error rate of our system is surprisingly high at 19.4%. One possible explanation for this phenomenon is that the lack of speaking style and prosody modeling techniques in the acoustic model results in the fast speaking rate commonly existing in the audio book corpus, which hampers the understanding of such semantically unpredictable sentences. It could also have been caused by the inaccurate alignment between phonemes and speech frames generated by MFA, in which case using a more advanced forced alignment tool would help mitigate the issue.

Spoke Task

In the spoke task of speaker adaptation, our system, ranked in the middle, receives a quality MOS of 3.9 and a similarity MOS of 3.6. Around four systems achieved significantly higher scores than our system in both tests. The results are reasonable since we only perform finetuning on the acoustic model and the vocoder without other dedicated adaptation techniques.

3.5 Conclusions

In this chapter, we proposed to utilize a new backbone of diffusion models, Diffusion Transformer, for adaptive TTS. Specifically, the adaptive layer norm in the architecture was used to condition the diffusion network on text representations, which further enabled parameter-efficient adaptation. On basic TTS tasks, the new architecture was verified to be a faster alternative to its convolutional counterpart. For few-shot adaptation, the DiT decoder demonstrated a clear advantage on naturalness and speaker similarity over the transformer decoder while maintaining parameter efficiency. When used in a zero-shot adaptive system, while we found the DiT is a decent alternative to the non-causal WaveNet, its main merit is to provide efficient high-quality adaptation when finetuning is performed. Combined with a diffusion-based vocoder and additional efforts on text analysis for French, our system is ranked favorably in the Blizzard Challenge 2023, demonstrating its capability of high-quality and natural speech synthesis.

4 Bayesian Transfer Learning for Parameter-Efficient Fine-Tuning

This chapter is situated within the context of recent advances in TTS systems, which increasingly rely on large-scale models pre-trained on extensive data. These models—particularly those incorporating language model-based architectures—usually demonstrate strong zero-shot synthesis capabilities and adopt general-purpose architectures such as transformers. As a result, parameter-efficient fine-tuning (PEFT) techniques, originally developed for broader adaptation tasks, have emerged as a compelling approach for domain adaptation in TTS.

Despite their efficiency, PEFT methods remain vulnerable to catastrophic forgetting, where fine-tuning can degrade the pre-trained model's inherent capabilities. In the context of TTS, this issue manifests as a loss of zero-shot synthesis performance, ultimately compromising generalizability and overall synthesis quality. To address this challenge, we investigate Bayesian transfer learning theory to overcome forgetting within the PEFT framework. We demonstrate that existing Bayesian transfer learning techniques can be applied to PEFT to prevent catastrophic forgetting as long as the parameter shift of the fine-tuned layers can be calculated differentiably. In a principled series of experiments on language modeling and speech synthesis tasks, we utilize established Laplace approximations, including diagonal and Kronecker-factored approaches, to regularize PEFT with low-rank adaptation (LoRA) and compare their performance in pre-training knowledge preservation. Our results demonstrate that catastrophic forgetting can be overcome by our methods without degrading the fine-tuning performance, and using the Kronecker-factored approximation produces a better preservation of the pre-training knowledge than the diagonal ones.

The work in this chapter has been published as:

Chen, H. and Garner, P. N. (2024). Bayesian parameter-efficient fine-tuning for overcoming catastrophic forgetting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4253–4262

4.1 Introduction

In the context of TTS, it has long been of interest to adapt a generic model to a specific domain such as a given speaker identity, language, or emotion. The process is termed *adaptation*; typically the generic model would be well-trained on a large dataset, whereas the (domain-specific) adaptation dataset would be too small to train a bespoke model. Adaptation proved particularly useful in statistical parametric and neural TTS (Yamagishi et al., 2009; Arik et al., 2018), and remains a goal of the recent Blizzard Challenge (Perrotin et al., 2023). More recently, the state of the art in TTS is represented by more generic generative models that have arisen in the machine learning community, with advances made in the domains of text (Brown et al., 2020; OpenAI, 2023), vision (Rombach et al., 2022; Saharia et al., 2022), and audio (Borsos et al., 2023; Vyas et al., 2023), all feeding through to TTS.

A key paradigm that has emerged in the development and application of such generic models is the pre-training fine-tuning approach, which involves initially training a model on a large dataset (pre-training) and subsequently fine-tuning it on a task-specific dataset. The paradigm has proven to be highly effective, leading to substantially more accurate and robust outcomes. More recent large pre-trained models have increasingly been equipped with in-context or zero-shot learning capabilities (Rombach et al., 2022; Wang et al., 2023a; Vyas et al., 2023). However, when there are more data available for the target task, fine-tuning is still useful to further improve the performance considerably (Mosbach et al., 2023). Notice that, whilst the vocabulary differs slightly, the goal is the same as for TTS. It follows that current research in fine-tuning provides the means to adapt current TTS models.

The performance gains achieved by large pre-trained models are undeniably linked to their scale. Larger models, with their increased capacity, tend to deliver superior performance. However, as the size of pre-trained models increases, the costs associated with fine-tuning and storing all parameters become prohibitively high, making it practically infeasible. This has led to the study of parameter-efficient fine-tuning (PEFT) techniques (Houlsby et al., 2019; Li and Liang, 2021; Zaken et al., 2022; Hu et al., 2022), which optimize a small subset of the model parameters (either original parameters or additional ones) while leaving the rest unchanged, significantly reducing computation and storage costs. PEFT techniques have not only facilitated fine-tuning of large pre-trained models on low-resource devices but also enabled the easy sharing and deployment of customized models as far fewer parameters need to be stored and transferred.

Despite the benefits of (parameter-efficient) fine-tuning, it is not without its pitfalls. One significant risk is catastrophic forgetting (McCloskey and Cohen, 1989; French, 1999; Goodfellow et al., 2014), where the model loses much of the knowledge it gained during pre-training. This loss can adversely affect the model's ability to generalize to unseen data, a critical aspect of any machine learning model. The phenomenon is even more unfavorable on modern large pre-trained models that are usually multi-functional by training on a diverse range of tasks and data. For example, a language model may forget its general knowledge after continual instruction tuning (Luo et al., 2023), or hypothetically, the controllability of emotions of a speech synthesizer may be compromised after fine-tuning on a specific voice.

Bayesian learning theory provides a principled solution to overcoming catastrophic forgetting. Considering optimizing the neural network as performing a maximum a posteriori (MAP) estimation of the network parameters given the fine-tuning data, it tries to find the optimal trade-off between the likelihood of the fine-tuning data and the prior knowledge of the pre-trained model, of which the latter is accessible in the form of the posterior over the parameters given the pre-training data. Although the true posterior is intractable, it can be approximated by fitting a Gaussian distribution with a mean equal to the MAP solution and a precision equal to the observed Fisher information. The technique is known as the Laplace approximation (MacKay, 1992) and has been thoroughly studied (Kirkpatrick et al., 2017; Martens and Grosse, 2015; Botev et al., 2017; Ritter et al., 2018b).

In this work, we demonstrate quite generally that existing Bayesian learning techniques can be applied to PEFT to overcome catastrophic forgetting. Deriving from the Bayesian transfer learning framework, we show that it is viable to regularize PEFT to preserve the pre-training knowledge as long as the parameter shift of the fine-tuned layers can be expressed in a differentiable manner. Utilizing established Laplace approximation techniques including diagonal (Kirkpatrick et al., 2017; Li et al., 2018) and Kronecker-factored (Martens and Grosse, 2015; Ritter et al., 2018a) approximations of the Hessian, we conduct a series of experiments on language modeling and speech synthesis tasks with low-rank adaptation (LoRA) (Hu et al., 2022) to demonstrate the effectiveness and compare the performance of different methods. Specifically, we start from a study on text classification and causal language modeling tasks, the quantitative nature of which allows both rigorous comparison of techniques and comparison with existing literature. We then verify our findings on our target task of speaker adaptation of speech synthesis, where the results are typically more subjective and more onerous to generate. Our results demonstrate that catastrophic forgetting can be overcome by such methods without degrading the fine-tuning performance, and the Kronecker-factored approximations generate a better preservation of the pre-training knowledge than the diagonal ones. Audio samples and source code are available[1].

4.2 Related Work

4.2.1 Laplace Approximation

The Laplace approximation (MacKay, 1992) is an established technique in statistics and machine learning to approximate a complex posterior distribution with a Gaussian distribution. This is achieved by identifying the mode of the posterior distribution, which is the maximum a posteriori estimate, and then approximating the distribution around this mode using a second-order Taylor expansion. Two popular kinds of Laplace approximation are the diagonal

[1] https://github.com/idiap/bayesian-peft

approximation (Kirkpatrick et al., 2017; Li et al., 2018), which only considers the variance of each model parameter itself and ignores the interactions between model parameters, and the Kronecker-factored approximation (Martens and Grosse, 2015) that also takes the covariance between parameters within each layer into account. Thanks to the additional information on the off-diagonal elements of the Hessian, the Kronecker-factored approximation has been shown to be more accurate than the diagonal approximation in capturing the loss landscape (Ritter et al., 2018a).

The Laplace approximation has been widely applied in neural network optimization (natural gradient descent) (Pascanu and Bengio, 2014; Martens and Grosse, 2015; Botev et al., 2017; George et al., 2018), improving calibration of neural networks (predictive uncertainty estimation) (Ritter et al., 2018b; Kristiadi et al., 2020; Immer et al., 2021a; Daxberger et al., 2021), and overcoming catastrophic forgetting in transfer and continual learning (Kirkpatrick et al., 2017; Ritter et al., 2018a; Kao et al., 2021). In this work, we focus on its application in mitigating catastrophic forgetting in the PEFT setting.

4.2.2 Parameter-Efficient Fine-Tuning

There exists a variety of PEFT techniques taking different approaches to adding new trainable components to, or modifying existing parameters of the pre-trained model. Representative PEFT techniques include

1. inserting serial or parallel adapters with a bottleneck structure to the model (Houlsby et al., 2019; Pfeiffer et al., 2020; He et al., 2022),
2. prepending trainable tokens to the input and hidden states of the transformer block (Li and Liang, 2021; Lester et al., 2021),
3. fine-tuning the bias terms inside the model only (Zaken et al., 2022),
4. optimizing the low-rank approximation of the change of weights (Hu et al., 2022; Hyeon-Woo et al., 2022; Edalati et al., 2023; Yeh et al., 2024), and
5. the combination of the above methods (He et al., 2022; Mao et al., 2022).

4.2.3 Continual Learning

Continual learning aims to enable the model to learn from non-stationary streams of data. (van de Ven et al., 2022) categorizes continual learning into three types: task-, domain-, and class-incremental learning. In the context of the adaptation of TTS models, we are interested in the scenario where the pre-trained model is fine-tuned to solve the same task as the pre-training one using data from different domains. This is an example of the domain-incremental type. Despite close ties with continual learning, the scenario concerned aligns better with *transfer learning* and *domain adaptation*. Further constraints that should be considered include that not all pre-training data are accessible and that the pre-training process cannot be replayed. All such constraints limit the usage of techniques designed for task- and class-incremental learning, such as Learning without Forgetting (Li and Hoiem, 2016) and Synaptic Intelligence (Zenke et al., 2017).

There have been attempts to utilize PEFT techniques, mainly low-rank adaptation (LoRA), in the continual learning setting. C-LoRA (Smith et al., 2024) leverages a self-regularization mechanism with LoRA to prevent catastrophic forgetting in continual customization of text-to-image models; O-LoRA (Wang et al., 2023b) continually learns tasks in different low-rank subspaces that are kept orthogonal to each other to minimize interference. For general fine-tuning, (Xiang et al., 2023) proposes to regularize the LoRA weights with Elastic Weight Consolidation (Kirkpatrick et al., 2017) when fine-tuning language models on question-answering tasks while preserving their general inference abilities.

4.3 Bayesian Transfer Learning

4.3.1 Framework

The optimization of neural networks can be interpreted as performing a maximum a posteriori (MAP) estimation of the network parameters $\boldsymbol{\theta}$ given the training data. In the transfer learning setting, the model has been pre-trained on a task $\mathcal{A}$ using data $\mathcal{D}_\mathcal{A}$, and is then fine-tuned on a downstream task $\mathcal{B}$ using data $\mathcal{D}_\mathcal{B}$. The overall objective is to find the optimal parameters on task $\mathcal{B}$ while preserving the prior knowledge of the pre-trained model on task $\mathcal{A}$. The posterior to be maximized in the MAP estimation can be written as:

$$\begin{aligned} p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}) &= \frac{p(\mathcal{D}_\mathcal{B}|\boldsymbol{\theta}, \mathcal{D}_\mathcal{A})\, p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A})}{p(\mathcal{D}_\mathcal{B}|\mathcal{D}_\mathcal{A})} \\ &= \frac{p(\mathcal{D}_\mathcal{B}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A})}{p(\mathcal{D}_\mathcal{B})} \end{aligned} \tag{4.1}$$

where $\mathcal{D}_\mathcal{B}$ is assumed to be independent of $\mathcal{D}_\mathcal{A}$. Taking a logarithm of the posterior, the MAP objective is therefore:

$$\begin{aligned} \boldsymbol{\theta}^* &= \underset{\boldsymbol{\theta}}{\arg\max} \log p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A}, \mathcal{D}_\mathcal{B}) \\ &= \underset{\boldsymbol{\theta}}{\arg\max} [\log p(\mathcal{D}_\mathcal{B}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A}) - \log p(\mathcal{D}_\mathcal{B})] \\ &= \underset{\boldsymbol{\theta}}{\arg\max} [\log p(\mathcal{D}_\mathcal{B}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A})] \end{aligned} \tag{4.2}$$

The first term $p(\mathcal{D}_\mathcal{B}|\boldsymbol{\theta})$ is the likelihood of the data $\mathcal{D}_\mathcal{B}$ given the parameters $\boldsymbol{\theta}$, which can be expressed as the training loss function on task $\mathcal{B}$, denoted by $\mathcal{L}_\mathcal{B}(\boldsymbol{\theta})$. The second term $p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A})$ is the posterior of the parameters given the pre-training data $\mathcal{D}_\mathcal{A}$. If training the network from scratch, i.e., assuming $\mathcal{D}_\mathcal{A}$ and $\mathcal{D}_\mathcal{B}$ to be one dataset $\mathcal{D}$, this term is usually approximated by a zero-mean isotropic Gaussian distribution, i.e., $p(\boldsymbol{\theta}|\mathcal{D}) = \mathcal{N}(\boldsymbol{\theta}|0, \sigma^2\mathbf{I})$, corresponding to $\mathcal{L}_2$ regularization. However, for transfer learning, this posterior must encompass the prior knowledge of the pre-trained model to reflect which parameters are important for task $\mathcal{A}$. Despite the true posterior being intractable, $\log p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A})$ can be defined as a function $f(\boldsymbol{\theta})$ and approximated around the optimum point $f(\boldsymbol{\theta}_0)$ (MacKay, 1992), where $\boldsymbol{\theta}_0$ is the pre-trained values and $\nabla f(\boldsymbol{\theta}_0) = 0$. Performing a second-order Taylor expansion on $f(\boldsymbol{\theta})$ around $\boldsymbol{\theta}_0$ gives:

$$\begin{aligned} \log p(\boldsymbol{\theta}|\mathcal{D}_\mathcal{A}) &\approx f(\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^\top \nabla^2 f(\boldsymbol{\theta}_0)(\boldsymbol{\theta}-\boldsymbol{\theta}_0) \\ &= f(\boldsymbol{\theta}_0) + \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^\top \mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_0) \end{aligned} \tag{4.3}$$

where $\mathbf{H}$ is the Hessian matrix of $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta}_0$. The second term suggests that the posterior of the parameters on the pre-training data can be approximated by a Gaussian distribution with mean $\boldsymbol{\theta}_0$ and covariance $\mathbf{H}^{-1}$. Note that the negation of the expected value of the Hessian over the data distribution is the Fisher information matrix (FIM) $\mathbf{F}$, i.e., $\mathbf{F} = -\mathbb{E}_{\mathcal{D}_\mathcal{A}}[\mathbf{H}]$. Following Equation 4.2, the training objective becomes:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} [\mathcal{L}_\mathcal{B}(\boldsymbol{\theta}) - \frac{1}{2}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^\top \mathbf{H}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)] \tag{4.4}$$

Finally, the loss function that we minimize during fine-tuning can be written as:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_\mathcal{B}(\boldsymbol{\theta}) + \lambda(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^\top \mathbf{F}(\boldsymbol{\theta}-\boldsymbol{\theta}_0) \tag{4.5}$$

where λ is the regularization strength that determines how much prior knowledge should be preserved during fine-tuning.

4.3.2 Diagonal Approximation of the Hessian

Modern neural networks typically have millions to billions of parameters, thus the Hessian, being at least terabytes, is intractable to compute and store. One practical approximation of the Hessian is the diagonal of the Fisher information matrix, i.e., the expected square of the gradients over the data distribution, known as Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017). The loss function of EWC is:

$$\mathcal{L}_{EWC}(\boldsymbol{\theta}) = \mathcal{L}_\mathcal{B}(\boldsymbol{\theta}) + \lambda \mathbf{F}_{EWC}(\boldsymbol{\theta}-\boldsymbol{\theta}_0)^2 \tag{4.6}$$

where $\mathbf{F}_{EWC}$ is the vectorized expected square of the gradients over the distribution of $\mathcal{D}_\mathcal{A}$.

To estimate $\mathbf{F}_{EWC}$, a small subset of the pre-training data $\mathcal{D}_\mathcal{A}$ is sampled and used to compute the gradients of the training loss function $\mathcal{L}_\mathcal{A}(\boldsymbol{\theta})$ on task $\mathcal{A}$. The final $\mathbf{F}_{EWC}$ is then the average of the square gradients over the sampled data.

A simplified version of EWC, named L2-SP (Li et al., 2018), assigns equal importance to all parameters, which is equivalent to assuming that the Fisher information matrix is an identity matrix. The loss function of L2-SP is:

$$\mathcal{L}_{L2-SP}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^2 \tag{4.7}$$

L2-SP can be regarded as an extension of the $\mathcal{L}_2$ regularization: instead of zero, it limits the parameters to be close to the pre-trained values during fine-tuning by assigning a Gaussian prior $\mathcal{N}(\boldsymbol{\theta}_0, \sigma^2\mathbf{I})$. Despite being overly simplified, L2-SP proves to be effective in preventing catastrophic forgetting in transfer learning (Li et al., 2018), and is particularly useful when the pre-training data are unavailable since no estimation of the FIM is required.

4.3.3 Kronecker-Factored Approximation of the Hessian

While first-order approximations such as EWC and L2-SP are simple and efficient, they are not accurate enough to capture the complete loss landscape since they ignore the off-diagonal elements of the Hessian, i.e., the interactions between parameters. To address this issue, recent advances in second-order optimization (Martens and Grosse, 2015; Botev et al., 2017) utilize block-diagonal approximations of the Hessian: the diagonal blocks of the Hessian, corresponding to the interactions between parameters within a single layer, can be approximated as a Kronecker product of two much smaller matrices. This approximation is known as the Kronecker-factored approximate curvature, usually abbreviated as KFAC.

Following (Martens and Grosse, 2015), we denote the input, the weight, the pre-activations, the non-linear function, and the output of the l-th layer as $\mathbf{a}_{l-1}$, $\mathbf{W}_l$, $\mathbf{s}_l$, ϕ_l and $\mathbf{a}_l$, respectively. For simplicity, we only consider linear layers with no bias term, thus $\mathbf{s}_l = \mathbf{W}_l\mathbf{a}_{l-1}$ and $\mathbf{a}_l = \phi_l(\mathbf{s}_l)$. We further define $\mathbf{g}_l = \frac{\partial\mathcal{L}}{\partial\mathbf{s}_l}$ as the gradient of the loss function $\mathcal{L}$ with respect to the pre-activations $\mathbf{s}_l$. The FIM with respect to the weights $\mathbf{W}_l$ can be written as:

$$\mathbf{F}^l_{KFAC} = \frac{\partial^2\mathcal{L}}{\partial^2\texttt{vec}(\mathbf{W}_l)} = \mathbf{A}_l \otimes \mathbf{G}_l \tag{4.8}$$

where $\texttt{vec}(\mathbf{W}_l)$ is the vectorized form of $\mathbf{W}_l$, $\mathbf{A}_l = \mathbf{a}_{l-1}\mathbf{a}_{l-1}^\top$, $\mathbf{G}_l = \mathbf{g}_l\mathbf{g}_l^\top$ and $\otimes$ is the Kronecker product operator. To calculate the expectation, the two factors are assumed to be independent, thus the expected Kronecker product is approximated as the Kronecker product of the expected factors. Thanks to a property of the Kronecker product, the quadratic penalty term for each layer can be efficiently calculated:

$$(\mathbf{A}_l \otimes \mathbf{G}_l)\texttt{vec}(\Delta\mathbf{W}_l) = \texttt{vec}(\mathbf{G}_l\Delta\mathbf{W}_l\mathbf{A}_l) \tag{4.9}$$

where $\Delta\mathbf{W}_l = \mathbf{W}_l - \mathbf{W}_l^0$ is the parameter shift from the pre-trained weight $\mathbf{W}_l^0$ of the l-th layer.

The overall loss function of KFAC is:

$$\mathcal{L}_{KFAC}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda \sum_{l=1}^{L} \mathtt{vec}(\Delta \mathbf{W}_l) * \mathtt{vec}(\mathbf{G}_l \Delta \mathbf{W}_l \mathbf{A}_l) \tag{4.10}$$

Despite KFAC's assumption of independence between layers, the most important in-layer parameter interactions are taken into account. It has been demonstrated that KFAC leads to better prior knowledge preservation in continual learning than using a diagonal approximation of the Hessian (Ritter et al., 2018a).

4.4 Bayesian PEFT

In this work, we aim to show that Bayesian transfer learning can provide a unifying framework for a variety of PEFT techniques. Such an approach not only retains the parameter efficiency of PEFT but also brings a principled approach to regularization, in turn overcoming catastrophic forgetting.

Looking back on Eq. 4.5, it is not difficult to see that, as long as the parameter shift $\Delta \mathbf{W}_l$ of the fine-tuned layers can be expressed in a differentiable way, the Bayesian transfer learning framework can be applied to any PEFT technique in the form of modification to the inherent weight of the pre-trained model. The loss function of Bayesian transfer learning with PEFT is therefore:

$$\mathcal{L}_{PEFT}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda \sum_{l=1}^{L} \mathtt{vec}(\Delta \mathbf{W}_l)^{\top} \mathbf{F}_l \mathtt{vec}(\Delta \mathbf{W}_l) \tag{4.11}$$

The most representative PEFT technique that fits this requirement is the low-rank adaptation (LoRA) family. LoRA (Hu et al., 2022) aims to optimize the low-rank approximation of the change of the original weight matrices based on the hypothesis that the change of weights during fine-tuning has a low intrinsic rank. It is formulated as adding the matrix product of two low-rank matrices to the original weight matrix, i.e., $\mathbf{W}_l = \mathbf{W}_l^0 + \gamma \mathbf{A}_l \mathbf{B}_l^{\top}$, where $\mathbf{W}_l^0 \in \mathbb{R}^{d_o \times d_i}$ is the pre-trained weight matrix, γ is a scaling factor, $\mathbf{A}_l \in \mathbb{R}^{d_o \times r}$ and $\mathbf{B}_l \in \mathbb{R}^{d_i \times r}$ are two low-rank matrices. Therefore, the weight modification (delta weight) of each layer is simply $\Delta \mathbf{W}_l = \gamma \mathbf{A}_l \mathbf{B}_l^{\top}$. Following Eq. 4.11, the loss function of Bayesian transfer learning with LoRA is:

$$\mathcal{L}_{LoRA}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda \sum_{l=1}^{L} \mathtt{vec}(\gamma \mathbf{A}_l \mathbf{B}_l^{\top})^{\top} \mathbf{F}_l \mathtt{vec}(\gamma \mathbf{A}_l \mathbf{B}_l^{\top}) \tag{4.12}$$

Apart from the original LoRA, there exist several variants of LoRA including AdaLoRA (Zhang et al., 2023b), which adaptively assigns the rank to the LoRA matrices in each layer, FedPara (LoHa) (Hyeon-Woo et al., 2022; Yeh et al., 2024), of which the delta weight is the Hadamard product of two LoRA delta weights, and KronA (LoKr) (Edalati et al., 2023; Yeh et al., 2024), which generates the delta weight by the Kronecker product of two low-rank matrices. Thanks to the explicit formulation of the delta weight, the LoRA family fits any aforementioned approximation of the Hessian in the Bayesian transfer learning framework. We also note that other PEFT methods such as $(\texttt{IA})^3$ (Liu et al., 2022a) and Orthogonal Butterfly (Liu et al., 2024b), that do not explicitly calculate the delta weight, also fit in the framework, although regularizing these methods may require extra computation and memory. Given that the original LoRA has achieved sufficiently good performance, e.g., it matches the full fine-tuning performance on the GLUE benchmark (Hu et al., 2022), and other LoRA variants only offer insubstantial improvements, we only employ the original LoRA and focus on the study of regularization methods in our experiments.

4.5 Experiments: Language Modeling

4.5.1 Tasks

We first apply our methods to fine-tuning pre-trained language models with LoRA on two sets of language modeling tasks: text classification and causal language modeling. The reason for this choice of task is twofold: The first is that language models can be evaluated quantitatively; a clear metric is associated with each task. The second is that it allows objective comparison with the wider literature.

Text Classification

We select three sentence-pair classification tasks and one single-sentence classification task from the GLUE benchmark (Wang et al., 2019). The sentence-pair tasks are: MNLI (Williams et al., 2018), a natural language inference task of predicting whether a premise entails, contradicts or is neutral to a hypothesis, QQP (Iyer et al., 2019), a paraphrase detection task of predicting whether a pair of sentences are semantically equivalent, and QNLI (Rajpurkar et al., 2016), a question answering task of predicting whether a sentence answers a question. The single-sentence task is SST-2 (Socher et al., 2013), a sentiment analysis task of predicting whether a sentence has positive or negative sentiment. For all tasks, the fine-tuning performance is reflected by the accuracy on the validation set. The number of training examples in the 4 selected datasets are MNLI: 393k, QQP: 363k, QNLI: 105k, and SST-2: 67k.

Causal Language Modeling

We experiment on the two subsets, WikiText-2 and WikiText-103, of the WikiText dataset (Merity et al., 2017), a collection of over 100 million tokens extracted from the set of verified good and featured articles on Wikipedia. The number of tokens in WikiText-2 and WikiText-103 are 2.1M and 103M, respectively. The fine-tuning performance is reflected by the perplexity on the validation set, which is shared by the two subsets.

4.5.2 Model: OPT

We select the Open Pre-trained Transformers (OPTs) (Zhang et al., 2022) with 350M and 1.3B parameters as the pre-trained models for our experiments. The OPTs are a suite of decoder-only transformers ranging from 125M to 175B parameters pre-trained on a series of large open-access corpora, including a subset of the Pile (Gao et al., 2021). Our choice of model sizes is based on those of state-of-the-art pre-trained TTS models ranging from 100M to 1B parameters (Li et al., 2023; Vyas et al., 2023; Lajszczak et al., 2024), so that the findings will hopefully provide useful guidance for our target task.

For text classification, a classification head is added on the last token the model generates and trained along with LoRA. This is purely for the simplicity of the implementation, though it could also be done by instruction tuning. For causal language modeling, the model structure remains unchanged.

4.5.3 Experimental Details

Implementation We base our code on the text classification and the causal language modeling examples of the Hugging Face Transformers library (Wolf et al., 2020). The Bayesian transfer learning techniques are implemented with the Hugging Face Parameter-Efficient Fine-Tuning (PEFT) library (Mangrulkar et al., 2022).

Hessian estimation The Hessian estimates are computed on the pre-training task, i.e., the causal language modeling task, and are shared by all fine-tuning tasks. We randomly sample 20,000 examples from the subset of the Pile used to pre-train the OPTs to compute the Hessian estimates for EWC and KFAC, and another 2,000 examples for the evaluation of the pre-training knowledge preservation.

Training and evaluation All models are trained using the Adam optimizer (Kingma and Ba, 2015) on each dataset for 3 epochs without weight decay. The learning rate is set to 5×10^{-4} for the 350M model and 2×10^{-4} for the 1.3B model, both with a linear decay schedule. For the text classification tasks, the batch size for all models is set to 32, while for the causal language modeling tasks, the batch size is set to 16 for the 350M model and 8 for the 1.3B model with a context window of 1024 tokens. LoRA is applied to the linear modules that produce the query and value in every self-attention module. The rank and the scaling factor of LoRA are set to 16 and 2 respectively for all models, resulting in the percentage of trainable parameters of the 350M and 1.3B model being 0.473% and 0.239%, respectively. To evaluate the fine-tuning performance, we calculate the accuracy or the perplexity on the validation set for the text classification tasks and the causal language modeling tasks respectively. For MNLI, the “matched” validation set is used. For the evaluation of the pre-training knowledge preservation, we calculate the perplexity on the sampled test set of the Pile. We run a coarse

Table 4.1: Main results of language modeling experiments.

Model	Method	λ	PT PPL	Classification (ACC↑ / PPL↓)				CLM (PPL↓ / PPL↓)	
				MNLI	QQP	QNLI	SST-2	WikiText-2	WikiText-103
OPT-350M	None	-	15.40	83.33% / 523.7	88.97% / 1234	89.79% / 51.11	93.81% / 19.05	13.48 / 20.35	15.21 / 31.74
	L2-SP	10^{-3}		83.35% / 33.65	88.28% / 19.91	89.84% / 23.69	93.72% / 16.66	13.62 / 18.21	15.95 / 20.61
	EWC	10^{4}		83.67% / 18.67	88.73% / 15.94	89.88% / 16.91	93.78% / 15.60	13.55 / 17.17	15.80 / 16.87
	KFAC	10^{6}		84.21% / 17.24	89.28% / 15.80	90.13% / 16.41	93.76% / 15.56	13.59 / 16.22	15.60 / 16.08
OPT-1.3B	None	-	11.18	87.70% / 23.55	90.97% / 16.28	92.59% / 13.45	95.94% / 11.87	9.81 / 13.08	10.53 / 24.32
	L2-SP	10^{-4}		87.77% / 15.66	90.32% / 15.94	92.51% / 13.33	96.10% / 11.78	9.82 / 12.72	10.71 / 15.93
	EWC	10^{4}		87.78% / 11.72	90.62% / 11.32	92.41% / 11.40	96.08% / 11.23	9.81 / 11.89	10.70 / 13.45
	KFAC	10^{5}		87.76% / 11.45	90.64% / 11.25	92.28% / 11.43	96.17% / 11.20	9.84 / 11.73	10.70 / 11.55

* ACC: accuracy, PPL: perplexity, PT PPL: perplexity of pre-trained model on the sampled test set from the Pile, CLM: causal language modeling.

hyper-parameter sweep on the regularization strength λ with a step size of 10 times for each method on each task. The optimal λ is selected balancing the fine-tuning performance and the preservation of pre-training knowledge, typically the point where fine-tuning performance is going to drop greatly if the regularization further strengthens. All experiments were conducted on machines equipped with one NVIDIA RTX3090. The results are averaged over 5 runs with different random seeds.

4.5.4 Results and Analyses

The main results are shown in Table 4.1. Note that the method "None" refers to LoRA without regularization. We elaborate our findings from several perspectives.

Catastrophic forgetting Compared to the pre-trained models, all models fine-tuned without regularization demonstrated significant forgetting of the pre-training knowledge, e.g., the perplexity on the pre-training data increased from 15.40 to 523.7 when fine-tuned on MNLI. Comparing different tasks, it is obvious that the forgetting is more severe when the model is fine-tuned on more data. In terms of model sizes, we notice that larger models tend to forget the pre-training knowledge less than smaller models, which suggests larger models have better resistance to catastrophic forgetting.

Comparison of regularization methods All regularization methods significantly reduced the loss of pre-training knowledge. Among them, L2-SP underperforms other methods by a large margin, which is reasonable given its over-simplified assumption of diagonal Hessian with equal importance on all parameters. In general, the Kronecker-based methods outperform EWC especially when there is more fine-tuning data, however, the difference is less significant for larger models. This demonstrates that knowledge preservation does benefit from more accurate Hessian estimations.

Regularization strength We provide an example of the regularization strength λ sweep for the 350M model fine-tuned on MNLI, which is shown in Table 4.2. As λ increases, the parameters are more constrained to the pre-trained values, thus the fine-tuning performance drops. We select the optimal λ as the one that achieves a fine-tuning performance better than that of using the original LoRA and has the lowest perplexity on the pre-training data. It can be seen that, compared to KFAC-based methods, the pre-training knowledge preservation of EWC is worse when achieving the same level of fine-tuning performance. We also observe that the fine-tuning benefits from the regularization when λ is small, which can be attributed to the fact that the Hessian estimation introduces a Gaussian prior that better describes the loss landscape than assuming an isotropic Gaussian prior at zero. This suggests that Bayesian transfer learning can lead to better fine-tuning performance as well as overcoming catastrophic forgetting.

Table 4.2: Comparison of performance with varying regularization strength of OPT-350M on MNLI.

Method	λ	Accuracy↑	Perplexity↓
Pre-trained	-	-	15.40
None	-	83.33%	523.74
L2-SP	10^{-4}	84.52%	52.51
	$\mathbf{10^{-3}}$	**83.35%**	**33.65**
	10^{-2}	81.51%	34.23
EWC	10^{3}	84.11%	26.84
	$\mathbf{10^{4}}$	**83.67%**	**18.67**
	10^{5}	82.03%	16.88
KFAC	10^{5}	84.32%	19.38
	$\mathbf{10^{6}}$	**84.21%**	**17.24**
	10^{7}	83.12%	17.10

Hessian estimates with varying samples We further experiment on Hessian estimates with a reduced amount of pre-training data to investigate the effect of the sample size on the accuracy of the Hessian estimation. The results are shown in Table 4.3. We observe that EWC is more robust to the sample size than KFAC, showing no degradation in pre-training knowledge preservation with Hessian estimates on fewer samples, whereas KFAC demonstrates significant degradation in perplexity on the pre-training data when the sample size is reduced to 20. This can also be corroborated by the increasing fine-tuning performance of KFAC when sample sizes decrease, which signifies less effective regularization. However, for other larger sample sizes, KFAC always outperforms EWC. Overall, the results suggest that KFAC, while being superior to EWC, requires more data to be estimated accurately than EWC, which is reasonable given its additional off-diagonal elements in the Hessian estimation.

Table 4.3: Comparison of Hessian estimates with varying samples.

Model	Samples	MNLI		WikiText-103	
		EWC	KFAC	EWC	KFAC
OPT-350M	20000	83.67% / 18.67	84.21% / 17.24	15.80 / 16.87	15.60 / 16.08
	2000	83.66% / 18.77	84.30% / 17.64	15.80 / 16.96	15.57 / 16.22
	200	83.71% / 18.50	84.51% / 17.60	15.83 / 16.84	15.47 / 16.79
	20	83.59% / 18.63	84.47% / 21.39	15.83 / 16.96	15.37 / 18.50
OPT-1.3B	20000	87.78% / 11.72	87.76% / 11.45	10.70 / 13.45	10.70 / 11.55
	2000	87.79% / 11.74	87.70% / 11.46	10.70 / 13.36	10.70 / 11.53
	200	87.74% / 11.70	87.76% / 11.54	10.71 / 13.22	10.66 / 11.68
	20	87.85% / 11.67	87.71% / 11.94	10.70 / 13.49	10.59 / 12.53

Computational cost and memory usage We compare the computational cost and memory usage of each regularization method in Table 4.4. Note that the calculation is based on a linear layer with weight $\mathbf{W}_l \in \mathbb{R}^{d_o \times d_i}$ using a single sample. The computational cost has two sources: the estimation stage, where a small subset of the pre-training data is sampled to compute the FIM, and the training stage, where the regularization loss is computed at each iteration.

Table 4.4: Comparison of computational cost and memory usage.

Method	Computation		Memory
	Estimation	Regularization	
L2-SP	0	$\mathcal{O}(d_o d_i)$	0
EWC	$\mathcal{O}(d_o d_i)$	$\mathcal{O}(d_o d_i)$	$\mathcal{O}(d_i d_o)$
KFAC	$\mathcal{O}(d_o^2 + d_i^2)$	$\mathcal{O}(d_o d_i (d_o + d_i))$	$\mathcal{O}(d_o^2 + d_i^2)$

Overall, the comparison highlights a trade-off between computational efficiency and the expressiveness of the regularization. While L2-SP incurs virtually no estimation cost and has negligible memory overhead, it provides only a coarse constraint. EWC introduces moderate additional cost by requiring the averaged square of gradient but remains relatively manageable. In contrast, KFAC offers a more accurate approximation of curvature information, but at the expense of substantially higher computational and memory requirements.

4.6 Experiments: Speech Synthesis

4.6.1 Tasks

Having verified the efficacy of our methods quantitatively and objectively on language modeling tasks, we further apply them to our target application: the fine-tuning of speech synthesis models. Such models are typically more onerous and subjective to evaluate. Our strategy is to demonstrate that the results from the objective evaluation also apply to the more specific target application.

Specifically, we fine-tune a pre-trained zero-shot speech synthesizer with LoRA to adapt it to an unseen speaker. Next, we evaluate the speaker similarity on both the target speaker and other out-of-domain (OOD) speakers, of which the former represents the fine-tuning performance and the latter indicates how well the model preserves the pre-training knowledge. To amplify the effect of catastrophic forgetting, the target speaker and other OOD speakers should be distinct from the pre-training data, thus we select speakers with particular accents for both fine-tuning and evaluation.

We appreciate that the task of evaluating the pre-training knowledge preservation is perhaps of less practical value since there is more interest in getting a similar voice to the target speaker than maintaining the zero-shot performance on other speakers in such a setting. However, this is a necessary compromise owing to several reasons. Firstly, the current publicly available state-of-the-art speech synthesis models mainly target speaker adaptation and are far from being omnipotent, meaning a good zero-shot performance on other speech characteristics is not guaranteed. Further, both the objective and subjective evaluation methods of speaker similarity are well-established, which is not the case for most of the others. Finally, the multi-speaker speech data are easy to obtain, while in other cases the data are not. Despite the limitation, we believe the results will provide practical guidance not only for speaker adaptation on this model but also for many other models and usages where catastrophic forgetting is detrimental to the model's inherent capabilities.

4.6.2 Model: StyleTTS 2

To proceed with the proposed tasks, we need an open-access pre-trained TTS model that has good synthesis quality and zero-shot performance for speaker adaptation. StyleTTS 2 (Li et al., 2023) is a recently proposed end-to-end TTS model that utilizes style diffusion and adversarial training with a large speech language model to generate human-level expressive and diverse speech. It also achieves a remarkable zero-shot performance though only trained on limited data of 245 hours from the LibriTTS dataset (Zen et al., 2019) compared to large-scale models such as VALL-E (Wang et al., 2023a), which is trained on 60k hours of data. Initial experiments on zero-shot synthesis show that despite StyleTTS 2 rendering excellent synthesis quality, the synthesized speech tends to lose the accent traits of the target speaker, which can be attributed to the limited training data. Nevertheless, this could be suitable for our experiments as it makes the improvement brought by fine-tuning or the degradation of zero-shot performance more distinguishable.

StyleTTS 2 has a variety of components, many of which are composed of modules that are not compatible with LoRA or whose Hessian estimation needs extra calculation, such as LSTMs and 1D/2D convolutions. However, we found in our initial experiments that only fine-tuning the linear modules in StyleTTS 2 already achieves reasonably good performance. Therefore, for convenience, we only fine-tune the linear modules in all components that are useful for inference of StyleTTS 2.

4.6.3 Experimental Details

Implementation Our code is based on the official implementation of StyleTTS 2 [2]. The same PEFT library for previous experiments is used for applying Bayesian methods and LoRA to the model.

Hessian estimation We use the official fine-tuning code to calculate the Hessian estimates, during which all training losses are enabled to ensure the gradients are properly back-propagated to all components. Based on the experience from language modeling experiments, we ran-

[2] https://github.com/yl4579/StyleTTS2

domly sample 1,000 utterances from the `train-clean-360` subset of the LibriTTS dataset for Hessian estimation to ensure accuracy.

Data We select p248, a female speaker with an Indian accent in the VCTK dataset (Yamagishi et al., 2019) as the target speaker and randomly split the data into the training set of 356 utterances (approximately 21 minutes) and the test set of 20 utterances. For OOD speakers, we select another 9 speakers (5 females, 4 males) with different accents from VCTK and randomly choose 20 utterances of each speaker as test sets.

Training and inference We adopt the official multi-stage fine-tuning strategy of 50 epochs described in the code repository for all models, only reducing the batch size from 8 to 2 due to hardware limits. LoRA is applied to the linear modules in all components except for the discriminators and the text aligner which are fully trained and only used during training. The rank and the scaling factor of LoRA are set to 16 and 2 respectively, resulting in an overall percentage of trainable parameters of 1.639% (2.26M of 138M). The fine-tuning is conducted 3 times with different random seeds. For inference, we synthesize test samples using the test sentences for every speaker using the fine-tuned model. All experiments were conducted on the same hardware as previous experiments.

Evaluation We conduct both objective and subjective evaluations, focusing exclusively on the speaker similarity. Essentially, we use the objective test results as the guideline for our experiments and corroborate our findings with subjective test results. More details are provided in the following sections.

Regularization Based on the fact that L2-SP is far inferior to other methods, we only experiment with EWC and KFAC in this section. The optimal regularization strength λ is selected using the same criterion as in the language modeling experiments based on the results of the hyperparameter sweep. It is 10^3 for both EWC and KFAC.

4.6.4 Objective Evaluation

For the objective evaluation, we use an ECAPA-TDNN (Desplanques et al., 2020) speaker verification model [3] to compute the averaged speaker embedding cosine similarity (SECS) score between the synthesized speech and the ground truth on the test set of each speaker. The averaged results of the three runs are shown in Table 4.5. Note that OOD All/Female/Male are the aggregated scores of all/female/male OOD speakers, "Full" and "Linear" stand for full fine-tuning and linear module-only fine-tuning, respectively. We analyze the results from the following perspectives.

[3] https://huggingface.co/speechbrain/spkrec-ecapa-voxceleb

Table 4.5: Main objective test results of speech synthesis experiments.

Speaker	Accent	Model					
		Pre-trained	**Full**	**Linear**	**LoRA**	**LoRA+EWC**	**LoRA+KFAC**
p248 (f, target)	Indian	0.216	0.695	0.652	0.654	0.633	0.648
OOD All	-	0.293	0.159	0.204	0.203	0.224	0.280
OOD Female	-	0.325	0.184	0.226	0.227	0.247	0.291
OOD Male	-	0.254	0.127	0.175	0.174	0.196	0.267
p225 (f)	English	0.318	0.167	0.241	0.252	0.296	0.352
p234 (f)	Scottish	0.385	0.221	0.257	0.240	0.274	0.297
p261 (f)	Northern Irish	0.448	0.206	0.288	0.281	0.323	0.374
p294 (f)	American	0.267	0.131	0.173	0.181	0.166	0.241
p335 (f)	New Zealand	0.205	0.195	0.171	0.179	0.176	0.188
p245 (m)	Irish	0.324	0.143	0.189	0.209	0.256	0.319
p302 (m)	Canadian	0.262	0.109	0.169	0.170	0.219	0.308
p326 (m)	Australian	0.165	0.132	0.112	0.105	0.082	0.164
p347 (m)	South African	0.262	0.123	0.232	0.210	0.228	0.276

* A suffix (m/f) is added to the speaker name to indicate the gender. Speakers in bold are selected for subjective evaluation.

Fine-tuning performance After fine-tuning, the SECS score of the target speaker p248 increases from 0.216 to above 0.6, which manifests that fine-tuning is essential for improving speaker similarity. Without a doubt, the full fine-tuning achieves the best performance. The linear module only fine-tuning ("Linear") and its LoRA-enabled counterpart ("LoRA") perform similarly, however falling behind by a less than 10% margin. This demonstrates the efficacy of the linear module-only fine-tuning scheme. Applying EWC and KFAC on top of LoRA further degrades the performance slightly, with KFAC performing slightly better than EWC.

Zero-shot performance The overall scores on all OOD speakers clearly demonstrate the catastrophic forgetting, dropping from 0.293 for the pre-trained model to 0.159 for the fully fine-tuned model. Fine-tuning the linear modules only with or without LoRA slightly mitigates the forgetting, suggesting it is necessary to apply additional regularization. Under optimal λ settings, KFAC (0.280) performs substantially better than EWC (0.224), only showing a slight degradation compared to the pre-trained model. The gender breakdown indicates that the fine-tuned model generally achieves a higher similarity on females than males, which can be attributed to the female fine-tuning data. This is confirmed by our test listening that the male speech synthesized by models without regularization severely deteriorates and resembles female speech more. In the speaker breakdown, despite the pre-trained model performing well on some speakers, the fine-tuning degrades similarities on all OOD speakers. One of the reasons for this could be the distinction between the target speaker and the OOD speakers in terms of the accent and the timbre. Moreover, the similarity drops more on speakers that previously had high similarity before fine-tuning. However, in any case, KFAC successfully preserves the zero-shot performance of the model, exceeding EWC by a large margin.

Table 4.6: Comparison of EWC and KFAC with varying regularization strength.

λ	EWC		KFAC	
	Target	**OOD**	**Target**	**OOD**
10^2	0.641	0.213	0.647	0.261
$\mathbf{10^3}$	**0.633**	**0.224**	**0.648**	**0.280**
10^4	0.575	0.270	0.593	0.283
10^5	0.379	0.271	0.491	0.271

Regularization strength We provide the λ sweep results in Table 4.6. It can be seen that under all λ settings, KFAC always achieves better fine-tuning performance and better zero-shot performance preservation than EWC. When matching a good similarity score above 0.6 on the target, EWC shows a significant degradation on OOD speakers. Furthermore, as λ increases, EWC's fine-tuning performance drops faster than KFAC and its zero-shot performance never surpasses that of KFAC. Overall, the results suggest that KFAC helps maintain the zero-shot synthesis ability of the pre-trained model while achieving good fine-tuning performance, whereas EWC suffers from a significant loss of fine-tuning performance when preserving the pre-training knowledge. This is consistent with the results of language modeling experiments on the smaller 350M model, however here the phenomenon is more pronounced.

4.6.5 Subjective Evaluation

Sample selection Having verified the efficacy with objective tests, we further conduct a subjective evaluation to corroborate our findings. One of the concerns is that the synthesized samples of OOD speakers usually result in a much lower perceptual similarity than those of the target speaker, making it difficult to distinguish the performance of low-performing models. In this regard, we select two OOD speakers that have the highest SECS scores and the most difference among models in each gender for the listening test, which are p225, p261, p245, and p302. 10 samples of the target speaker and 5 samples of each OOD speaker are randomly selected, totaling 10 female samples and 10 male samples of the OOD speakers for each model. We also add a ground truth (GT) group for comparison.

Implementation We hired 20 native English speakers from the United Kingdom on the Prolific [4] crowd-sourcing platform to rate the speaker similarity between the synthesized speech and the reference on a 5-point scale (5: completely same speaker, 4: mostly similar, 3: equally similar and dissimilar, 2: mostly dissimilar, 1: completely different speaker), using a modified Degradation Category Rating (DCR) method based on the P.808 toolkit (Naderi and Cutler, 2020). The reference is a random recording of the speaker with spoken content different from that of the test sample and is bound to each test sample. The averaged result is often referred to as the Similarity Mean Opinion Score (SMOS).

Table 4.7: Subjective test results with 95% confidence interval.

Model	**Target**	**OOD All**	**OOD Female**	**OOD Male**
GT	4.46 ± 0.11	4.59 ± 0.07	4.65 ± 0.10	4.52 ± 0.11
Pre-trained	1.90 ± 0.15	2.22 ± 0.13	2.36 ± 0.20	2.08 ± 0.17
Linear	4.06 ± 0.16	1.50 ± 0.10	1.83 ± 0.17	1.18 ± 0.07
LoRA	3.86 ± 0.16	1.48 ± 0.09	1.83 ± 0.17	1.13 ± 0.06
LoRA+EWC	3.60 ± 0.14	1.51 ± 0.10	1.77 ± 0.17	1.26 ± 0.09
LoRA+KFAC	3.81 ± 0.16	2.08 ± 0.13	2.31 ± 0.20	1.85 ± 0.16

Results and analyses The results are shown in Table 4.7. In general, the subjective test results corroborated our findings from objective tests, hence we mainly comment on the discrepancies between the two tests. For the target speaker, fine-tuning linear modules ("Linear") achieves an SMOS of 4.06, which is a significant improvement from the pre-trained model of 1.90 and is considerably good given the ground truth of 4.46. Different from the

[4] https://www.prolific.com

objective test results, the LoRA-only model shows a disadvantage of 0.20 compared to “Linear”, meaning fine-tuning a low-rank representation does degrade the fine-tuning performance for this model. The small difference between EWC and KFAC shown by SECS scores is actually perceivable, indicated by a difference of 0.21 in SMOS. In terms of zero-shot performance, EWC’s preservation effect is not reflected on SMOS considering all OOD speakers, which is in contrast with KFAC. The gender breakdown shows a slight degradation on male OOD speakers for the LoRA with KFAC model, suggesting KFAC did not perfectly preserve the zero-shot performance of the pre-trained model as the SECS scores showed.

4.7 Conclusions

In this work, we explored applying Bayesian learning techniques to parameter-efficient fine-tuning to overcome catastrophic forgetting. We started from the derivation of the Bayesian transfer learning framework and demonstrated that PEFT could be regularized to preserve the pre-training knowledge as long as the parameter shift of the fine-tuned layers could be calculated differentiably. We then conducted experiments with LoRA on both language modeling and speech synthesis tasks to verify the efficacy of the proposed methods and compared the performance of different Laplace approximations. Our results show that catastrophic forgetting can be overcome by our methods without degrading the fine-tuning performance. Furthermore, the results on both tasks suggest using the Kronecker-factored approximations of the Hessian produces more effective preservation of the pre-training knowledge and better fine-tuning performance than the diagonal approximations, even though the former requires more data to be estimated accurately.

Current limitations of this work include that it cannot be applied to PEFT techniques that add new components to the model such as bottleneck adapters; however this is not a serious concern given suitable techniques like LoRA already provide good fine-tuning performance. Further, it is only feasible when at least part of the pre-training data is accessible. Finally, the efficacy on larger (TTS) models has not been verified due to the inaccessibility to these models and hardware constraints. We would like to evaluate our methods on larger TTS models when they become publicly available in the future.

5 Variational Learning for Parameter-Efficient Fine-Tuning

In the previous chapter, we explored the application of the Laplace approximation in Bayesian transfer learning: the parameter distribution of the pre-trained model can be approximated post hoc by a Gaussian distribution through a second-order Taylor expansion around the pre-trained mode. This distribution can then be used to regularize parameter-efficient fine-tuning (PEFT) to preserve pre-training knowledge. In parallel with the Laplace approximation, variational inference generalizes this idea by reframing posterior approximation as an optimization problem. Despite sharing the ultimate goal of learning distributions of neural network parameters, variational inference techniques usually appear as online optimizers that estimate posterior distributions during training. This flexibility allows variational methods to learn more expressive posterior distributions along the training process.

In this chapter, we investigate the applications of variational learning in PEFT, utilizing the Improved Variational Online Newton (IVON), a state-of-the-art variational inference optimizer. In the first part, we demonstrate that variational learning can effectively improve predictive accuracy and calibration in PEFT, benchmarking its performance on natural language and audio understanding tasks against the Laplace approximation. In the second part, we utilize the online estimation of the posterior distribution of parameters to prune unimportant ranks for low-rank adaptation (LoRA), enabling automatic allocation of parameter budget to different layers and modules across the model.

The work in the second part is adapted from the following publication:

Chen, H. and Garner, P. N. (2025). A Bayesian interpretation of adaptive low-rank adaptation. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

5.1 Predictive Uncertainty Estimation

5.1.1 Introduction

Despite large language models (LLMs) having succeeded on a wide range of language and speech processing tasks, their probabilistic predictions are often poorly calibrated when fine-tuned on small datasets: the confidence scores they output do not always align with the true likelihood of correctness, often demonstrating overconfidence (Jiang et al., 2021; Tian et al., 2023; OpenAI, 2023). This misalignment may compromise predictive accuracy, posing challenges in high-stakes or risk-sensitive applications where reliable uncertainty estimation is crucial.

Variational inference (VI) offers a principled framework for uncertainty estimation by treating model parameters as posterior distributions rather than point estimates. By learning such posterior distributions, VI enables direct modeling of parameter uncertainty, which in turn leads to better-calibrated predictions and improved generalization by ensembling the output using multiple model samples during inference. Recent advances in efficient VI methods, such as Improved Variational Online Newton (IVON) (Shen et al., 2024), have made it feasible to fine-tune large-scale models while maintaining computational tractability. A closely related, state-of-the-art approach to improving calibration is the Linearized Laplace Approximation (LLA) (Daxberger et al., 2021), which provides a post-hoc Bayesian solution by fitting a Gaussian posterior around the pre-trained mode. When applied to LoRA-based fine-tuning, LLA can yield well-calibrated uncertainty estimates without extensive retraining (Yang et al., 2024), outperforming conventional uncertainty estimation methods such as deep ensemble (Lakshminarayanan et al., 2017), stochastic weight averaging (Maddox et al., 2019), and Monte-Carlo dropout (Gal and Ghahramani, 2016).

In this section, we examine how variational inference (VI) and the Laplace approximation can improve calibration in LLMs, with a focus on their theoretical foundations and practical trade-offs. By applying the two methods to LoRA-based fine-tuning on a series of commonsense reasoning and audio understanding tasks, we aim to assess their respective strengths and limitations in enhancing general fine-tuning performance and calibration within the framework of PEFT.

5.1.2 Variational Inference

Overview

Variational inference (VI) transforms the modeling of neural network parameter distributions into an optimization problem. It seeks a tractable surrogate distribution (often Gaussian) by minimizing the Kullback-Leibler (KL) divergence between the approximation and the true posterior. Essentially, VI leverages the evidence lower bound (ELBO) as a variational objective, framing inference as an optimization task that enables the use of stochastic gradient descent

Algorithm 1 Improved Variational Online Newton (IVON).

Require: Learning rates $\{\alpha_t\}$, weight-decay $\delta > 0$.
Require: Momentum parameters $\beta_1, \beta_2 \in [0, 1)$.
Require: Hessian init $h_0 > 0$.
Init: $\mathbf{m} \leftarrow$ (NN-weights), $\mathbf{h} \leftarrow h_0$, $\mathbf{g} \leftarrow 0$, $\lambda \leftarrow N$.
Init: $\boldsymbol{\sigma} \leftarrow 1/\sqrt{\lambda(\mathbf{h} + \delta)}$.
Optional: $\alpha_t \leftarrow (h_0 + \delta)\alpha_t$ for all t.

for $t = 1, 2, \ldots$ **do**
  $\widehat{\mathbf{g}} \leftarrow \widehat{\nabla}\bar{\ell}(\boldsymbol{\theta})$, where $\boldsymbol{\theta} \sim q$
  $\widehat{\mathbf{h}} \leftarrow \widehat{\mathbf{g}} \cdot (\boldsymbol{\theta} - \mathbf{m})/\boldsymbol{\sigma}^2$
  $\mathbf{g} \leftarrow \beta_1 \mathbf{g} + (1-\beta_1)\widehat{\mathbf{g}}$
  $\mathbf{h} \leftarrow \beta_2 \mathbf{h} + (1-\beta_2)\widehat{\mathbf{h}} + \frac{1}{2}(1-\beta_2)^2(\mathbf{h} - \widehat{\mathbf{h}})^2/(\mathbf{h} + \delta)$
  $\bar{\mathbf{g}} \leftarrow \mathbf{g}/(1 - \beta_1^t)$
  $\mathbf{m} \leftarrow \mathbf{m} - \alpha_t(\bar{\mathbf{g}} + \delta\mathbf{m})/(\mathbf{h} + \delta)$
  $\boldsymbol{\sigma} \leftarrow 1/\sqrt{\lambda(\mathbf{h} + \delta)}$
end for
return $\mathbf{m}, \boldsymbol{\sigma}$

Figure 5.1: Improved Variational Online Newton (IVON). [1]

within modern deep learning frameworks to efficiently fit probabilistic models. This flexibility has made VI a practical solution to training Bayesian neural networks (Khan et al., 2018; Osawa et al., 2019; Shen et al., 2024), where parameter uncertainty is explicitly modeled to improve generalization and robustness. Thanks to the uncertainty estimation of parameters, VI provides advantages over deep learning methods such as Adam (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) including better calibration and generalization, better predictive uncertainty estimation, and the possibility of model merging for knowledge transfer.

In contrast with traditional deep learning methods that estimate parameters by minimizing the empirical risk $\ell(\boldsymbol{\theta})$ (the loss function) with gradient descent, variational methods estimate a posterior distribution $q(\boldsymbol{\theta})$ over parameters by minimizing

$$\mathcal{L}(q) = \mathbb{E}_{q(\boldsymbol{\theta})}[\ell(\boldsymbol{\theta})] + \mathbb{D}_{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta})) \tag{5.1}$$

where $p(\boldsymbol{\theta})$ is the prior. The optimization of $\mathcal{L}(q)$ is fundamentally different from minimizing $\ell(\boldsymbol{\theta})$ using gradient descent. For example, the expectation term requires sampling of $\boldsymbol{\theta}$ before each forward pass, and the number of parameters of q is doubled for the commonly used Gaussian distribution with a diagonal covariance. Early approaches (Graves, 2011; Blundell et al., 2015) aim to optimize $q(\boldsymbol{\theta})$ ($\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ for diagonal Gaussian) using different stochastic gradient estimators. However, these methods have failed to scale up on modern architectures. Recent natural gradient-based methods (Khan et al., 2018; Osawa et al., 2019) have shown promising results using an Adam-like form; however, they still underperform Adam and have

[1] Originally in Shen et al. (2024).

significantly higher computational costs.

Improved Variational Online Newton (IVON)

The Improved Variational Online Newton (IVON) (Shen et al., 2024) is a recent VI optimizer that matches the performance of Adam at a comparable computational cost. Its key innovations include bypassing the expensive per-example gradient square computation through a reparameterization trick and incorporating several practical techniques to enhance performance. IVON stands out as the first VI optimizer proven to be both effective and efficient for training large networks, while still delivering the benefits of VI.

Figure 5.1 shows the algorithm of IVON. In comparison with Adam, the main differences include 1) the sampling of neural network parameters θ before the forward pass in line 2; 2) the reparameterization trick for estimating the current Hessian $\hat{\mathbf{h}}$ in line 3; 3) tracking the exponential moving average of Hessian $\mathbf{h}$ instead of the gradient square; 4) there is no square root over Hessian $\mathbf{h}$; and 5) the output of the mean $\mathbf{m}$ and the standard deviation σ instead of a point estimate of parameters $\boldsymbol{\theta}$. Overall, IVON offers an Adam-like framework without any significant computational overheads.

IVON introduces several additional hyperparameters that should be taken into consideration. Here, we list important ones that could greatly impact the training stability and the final performance.

1. Hessian initialization h_0: the Hessian is the inverse of the variance. Therefore, a larger h_0 corresponds to a smaller initial variance, leading to a more concentrated and deterministic initial posterior. This typically results in more stable training in the early stages. However, it also reduces the benefits of uncertainty estimation, potentially resulting in poorer performance.

2. Learning rate α_t: the learning rate for IVON is usually set to a higher value compared to Adam, typically on the order from 10^{-2} to 10^{-1}. In the case of PEFT or that the training set is small, the learning rate could be set even higher to facilitate fast convergence.

3. Effective sample size λ: λ modulates the scale of the estimated variance, thereby controlling the level of stochasticity introduced by sampling prior to each forward pass. A smaller λ increases the sampling temperature, which can lead to greater variance and potential instability during training. In practice, λ is often set equal to the size of the training dataset. However, for very small datasets, using a larger λ can help stabilize the short training process and improve overall performance.

5.1.3 Linearized Laplace Approximation

Similar to the Bayesian transfer learning framework introduced in the previous chapter, the use of the Laplace approximation for predictive uncertainty estimation also builds upon the maximum a posteriori (MAP) estimation. However, the objective here is distinct: rather than transferring knowledge from pre-training to fine-tuning data, the focus is on capturing predictive uncertainty during inference for a model fine-tuned with PEFT on a downstream task. Accordingly, the derivation begins from the MAP estimate of the PEFT parameters based solely on the fine-tuning data, independent of the pre-training data or the original pre-trained model.

MAP Estimation

For classification or next-token prediction tasks, the training objective is to estimate the posterior distribution of model parameters $\boldsymbol{\theta}$:

$$p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathbf{y} \mid \mathbf{X})} \tag{5.2}$$

where $\mathbf{X}$ represents the input matrix, and $\mathbf{y}$ represents the target vector. Here, $p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y})$ is the posterior distribution, $p(\mathbf{y} \mid \boldsymbol{\theta}, \mathbf{X})$ is the likelihood, and $p(\mathbf{y} \mid \mathbf{X})$ is the evidence (marginal likelihood). We employ an isotropic Gaussian prior with precision λ:

$$p(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{0}, \lambda^{-1}\mathbf{I}) \tag{5.3}$$

Taking a logarithm of the posterior, the MAP estimation maximizes the following function $f(\boldsymbol{\theta})$, which is the numerator on the right-hand side of Eq. 5.2:

$$\begin{gathered} f(\boldsymbol{\theta}) = \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) + \text{const} \\ \boldsymbol{\theta}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \end{gathered} \tag{5.4}$$

Performing a second-order Taylor expansion of $f(\boldsymbol{\theta})$ around $\boldsymbol{\theta}_{\text{MAP}}$ gives:

$$f(\boldsymbol{\theta}) \approx f(\boldsymbol{\theta}_{\text{MAP}}) - \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}})^\top (\nabla^2_{\boldsymbol{\theta}} f(\boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}})(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{MAP}}) \tag{5.5}$$

This quadratic term corresponds to a Gaussian posterior centered at $\boldsymbol{\theta}_{\text{MAP}}$ with covariance given by the inverse Hessian:

$$\begin{gathered} p(\boldsymbol{\theta} \mid \mathbf{X}, \mathbf{y}) \approx \mathcal{N}(\boldsymbol{\theta} | \boldsymbol{\theta}_{\text{MAP}}, \boldsymbol{\Sigma}) \\ \boldsymbol{\Sigma} = -(\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}})^{-1} = -(\nabla^2_{\boldsymbol{\theta}} \log p(\mathbf{y} \mid \mathbf{X}, \boldsymbol{\theta})|_{\boldsymbol{\theta}_{\text{MAP}}} + \lambda \mathbf{I})^{-1} \end{gathered} \tag{5.6}$$

We utilize the Fisher information matrix to approximate the covariance:

$$FIM(\boldsymbol{\theta}) = \sum_{n=1}^{N} \mathbb{E}_{p(\mathbf{y}|f_{\boldsymbol{\theta}}(\mathbf{x}_n))}\left[\nabla_{\boldsymbol{\theta}} p(\mathbf{y} \mid f_{\boldsymbol{\theta}}(\mathbf{x}_n))\big(\nabla_{\boldsymbol{\theta}} p(\mathbf{y} \mid f_{\boldsymbol{\theta}}(\mathbf{x}_n))\big)^{\top}\right] \tag{5.7}$$

where the expectation is taken with respect to the model's output distribution. Same as in Chapter 4, the structures of the FIM include diagonal and Kronecker-factored approximations.

Neural Network Linearization

Neural network linearization (Kunstner et al., 2019; Immer et al., 2021b; Antorán et al., 2022) approximates a nonlinear neural network with a linear model around a specific point in parameter space using a first-order Taylor expansion. It has been found that making predictions using the linearized model is more effective than sampling from the approximate posterior over the weights (Daxberger et al., 2021; Deng et al., 2022). The linearized model can be expressed as:

$$f_{\boldsymbol{\theta}}(\mathbf{x}_*) \approx f_{\boldsymbol{\theta}_{\mathrm{MAP}}}(\mathbf{x}_*) + \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^{\top}_{\boldsymbol{\theta}_{\mathrm{MAP}}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{MAP}}) \tag{5.8}$$

where $\mathbf{x}_*$ is a test input. Note that $\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^{\top}_{\boldsymbol{\theta}_{\mathrm{MAP}}}$ represents a matrix containing the gradient over parameters $\boldsymbol{\theta}$ per output dimension (number of classes or tokens). This formulation corresponds to the linearized Laplace approximation.

Given the approximated posterior in Eq. 5.6 and the linearized model in Eq. 5.8, we can marginalize over the posterior of the weights to obtain a Gaussian posterior distribution on the output logits:

$$f_{\boldsymbol{\theta}}(\mathbf{x}_*) \sim \mathcal{N}(f_{\boldsymbol{\theta}_{\mathrm{MAP}}}(\mathbf{x}_*), \boldsymbol{\Lambda}) \tag{5.9}$$

where

$$\boldsymbol{\Lambda} = (\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|^{\top}_{\boldsymbol{\theta}_{\mathrm{MAP}}})\boldsymbol{\Sigma}(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\mathbf{x}_*)|_{\boldsymbol{\theta}_{\mathrm{MAP}}}) \tag{5.10}$$

To sample from $f_{\boldsymbol{\theta}}(\mathbf{x}_*)$, we utilize the Cholesky factorization of the covariance matrix ($\boldsymbol{\Lambda} = \mathbf{L}\mathbf{L}^{\top}$):

$$\tilde{f}_{\boldsymbol{\theta}}(\mathbf{x}_*) = f_{\boldsymbol{\theta}_{\mathrm{MAP}}}(\mathbf{x}_*) + \mathbf{L}\boldsymbol{\xi} \tag{5.11}$$

where $\boldsymbol{\xi}$ is a i.i.d. standard Gaussian noise vector. The model output is computed by averaging probabilities (obtained via softmax on sampled logits) by Monte-Carlo sampling on the Gaussian noise.

5.1.4 Method

We apply IVON and LLA to PEFT with LoRA and evaluate their performance in terms of both predictive accuracy and calibration. Below, we detail the particularities of each technique and introduce metrics for evaluating calibration.

IVON

When applied to LoRA, IVON serves as a drop-in replacement for Adam, specifically optimizing the two additional low-rank matrices in each module. Like Adam, IVON estimates a diagonal covariance matrix assuming independence between parameters, and thus maintains a similar computational cost related to tracking second-order gradients. During training, the primary overhead arises from the sampling step before each forward pass; however, this cost remains marginal when using a single Monte Carlo sample per iteration. At test time, there are two options: 1) use the prediction at the mean of posterior parameter distribution, which is the most computationally efficient approach; and 2) average predictions over n samples drawn from the posterior, which requires n forward passes and increases computation by a factor of n when $n > 1$. The application of IVON to LoRA also appeared in a concurrent work (Cong et al., 2024); here, we focus on the comparison between IVON and LLA under different configurations.

Linearized Laplace Approximation

For LLA, post-hoc posterior estimation is performed after standard fine-tuning with Adam by fitting the Laplace approximation on the fine-tuning data. The low-rank matrices in each LoRA adapter are treated as two separate linear layers. Two key considerations are 1) the type of the Laplace approximation: either diagonal or Kronecker-factored, with the latter offering better posterior estimates while having higher computation and memory cost as shown in the previous chapter; and 2) the layers to apply: either across all LoRA adapters (denoted by LA) or limited to the final classification head (last-layer LA, or LLLA). The choice of LA and LLLA is mainly a tradeoff between computation and memory cost and posterior estimation accuracy: while applying LA to all LoRA adapters can further improve uncertainty estimation and robustness particularly in tasks where uncertainty propagates through multiple layers, it also incurs substantially higher computational and memory cost, especially with Kronecker-factored approximations. The application of LLA to PEFT with LoRA was introduced in Yang et al. (2024) as Laplace-LoRA.

5.1.5 Experiments: Commonsense Reasoning

Models and Datasets

We fine-tune the Llama 2 7B model (Touvron et al., 2023) with LoRA on six commonsense reasoning datasets (see Section 2.4.2, with numbers of training samples in the parentheses): ARC-Challenge (ARC-C, 1.12k), ARC-Easy (ARC-E, 2.25k), BoolQ (9.43k), OpenBookQA (OBQA, 4.96k), WinoGrande-Medium (WG-M, 2.56k), and WinoGrande-Small (WG-S, 640). These are the same datasets used in Yang et al. (2024) for direct comparison with baselines. ARC-Challenge, ARC-Easy, and OpenBookQA are multiple-choice tasks, while the rest are binary-choice tasks. The input to the model is the context followed by a question and the options (A. ..., B. ..., etc.), the task is to predict the correct label (such as A) as a next token prediction task. LoRA is applied to the query and value linear modules of attention, with rank set to 8 and alpha set to 16 (corresponding to an amplification factor of 2).

Implementation Details

We use the official implementation[2] of the IVON optimizer. For LLA, we rely on the official implementation of Laplace-LoRA[3], which is built on the Laplace [4] and ASDL[5] libraries. LoRA is implemented using the Hugging Face Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022) packages.

Training and Evaluation

All models are trained for 10,000 steps with a batch size of 4 on all tasks. For IVON, we use an effective sample size λ of 10^7, a Hessian initialization h_0 of 1×10^{-3}, a weight decay δ of 10^{-8}, and a learning rate of 0.03 with linear learning rate decay to 0. Setting an effective sample size much higher than the actual number of training samples reduces the sampling temperature, which in turn ensures a more stable training process. In contrary, the Hessian initialization has been set to a relatively small value (which corresponds to a large variance) for the optimizer to learn a more expressive posterior. For LLA, we train the model using the AdamW optimizer (Loshchilov and Hutter, 2019) without weight decay (which is identical to Adam without weight decay) while adopting the same hyperparameters as in the original Laplace-LoRA implementation. The Hessian estimation is performed using all training samples, followed by the optimization of prior precision λ maximizing marginal likelihood as described in (Daxberger et al., 2021).

Evaluation is performed on the validation sets of corresponding datasets using accuracy, ECE, NLL, and Brier score. For IVON, we report results under three settings: predictions at the

[2]https://github.com/team-approx-bayes/ivon
[3]https://github.com/adamxyang/laplace-lora
[4]https://github.com/aleximmer/Laplace
[5]https://github.com/kazukiosawa/asdl

posterior mean, single-sample Monte Carlo (MC-1), and the average over 8 Monte Carlo samples (MC-8). All results are averaged over 5 runs with different random seeds. For LLA, we compare 4 different settings considering the type of the Laplace approximation (diagonal or KFAC) and the layers to apply (all LoRA adapted layers or last layer).

Results and Analyses

All results are shown in Table 5.1. We elaborate our findings from the following perspectives. Note that we use LLA to denote linearized Laplace approximation, LA to denote LLA applied to all LoRA enabled layers, and LLLA to denote LLA applied to the last layer.

Accuracy Among all methods, IVON evaluated at the posterior mean achieves the highest accuracy on most tasks, outperforming Adam both with and without LLA. Applying LLA can yield a slight improvement over the MAP solution. In general, KFAC outperforms the diagonal approximation, especially when applied to all layers. However, applying LLA with diagonal approximation to all layers (LA_{diag}) underperforms that applied to the last layer ($\text{LLLA}_{\text{diag}}$), suggesting the diagonal approximation is not accurate enough to describe the posterior distribution across the entire model. For IVON, performance drops noticeably when using a single Monte Carlo sample, compared to predictions at the mean. Increasing the number of samples to 8 improves accuracy, but a small gap to the mean prediction still remains. These results suggest that both IVON and LLA enhance downstream performance, with IVON at the posterior mean offering the best trade-off between accuracy and computational efficiency.

Calibration Overall, LLA outperforms IVON in calibration. Among all LLA configurations, KFAC applied to all layers (LA_{KFAC}) achieves the best calibration, with the lowest ECE and Brier score, while LA_{diag} significantly underperforms LA_{KFAC} in terms of ECE. This aligns with expectations as KFAC offers a more expressive posterior approximation compared to the diagonal covariance. However, the benefit of LLA diminishes when restricted to the last layer, with $\text{LLLA}_{\text{KFAC}}$ and $\text{LLLA}_{\text{diag}}$ performing similarly however better than MAP. This suggests that much of the model's uncertainty originates from intermediate layers, and both KFAC and diagonal covariance can well model the uncertainty in the last layer. For IVON, predictions at the posterior mean reduce ECE and Brier score compared to MAP, while using a single Monte Carlo sample offers no improvement but a slight degradation. Increasing to 8 Monte Carlo samples improves calibration but still underperforms LA_{KFAC}.

Computation and Memory Cost Both methods incur similar computational costs during training (fine-tuning), with IVON being approximately 1–2% slower than Adam. The key differences arise in the post-training phase. LLA requires fitting a Laplace approximation on (a subset of) the training data to estimate the posterior covariance, which entails a computational cost comparable to an additional pass over the data with full forward and backward computa-

Table 5.1: Comparison of LLA and IVON applied to fine-tuning Llama-2 7B with LoRA on commonsense reasoning tasks. The **best** and the second best results are marked.

Metric	Optimizer	Method	ARC-C	ARC-E	BoolQ	OBQA	WG-M	WG-S	Avg.
ACC ↑	Adam	MAP	64.6	84.8	85.5	78.9	74.2	66.9	75.8
		$\text{LLLA}_{\text{diag}}$	65.3	85.0	85.6	79.2	74.6	66.8	76.1
		LA_{diag}	65.1	84.3	85.7	78.5	74.6	66.8	75.8
		$\text{LLLA}_{\text{KFAC}}$	65.2	85.0	85.6	79.2	74.6	66.8	76.1
		LA_{KFAC}	66.0	85.1	85.6	79.1	74.6	66.9	76.2
	IVON	Mean	**70.3**	**87.6**	**86.7**	**81.4**	76.6	**71.8**	**79.1**
		MC-1	62.1	83.0	85.5	76.6	**76.7**	70.6	75.7
		MC-8	66.7	85.7	86.4	79.9	76.4	71.6	77.8
ECE ↓ (100×)	Adam	MAP	33.2	14.2	7.9	19.0	24.3	32.5	21.8
		$\text{LLLA}_{\text{diag}}$	20.5	10.3	7.8	17.7	23.5	17.3	16.2
		LA_{diag}	14.2	14.5	18.7	10.6	**7.3**	7.8	12.2
		$\text{LLLA}_{\text{KFAC}}$	22.4	11.2	7.9	17.9	23.6	19.1	17.0
		LA_{KFAC}	**5.1**	**3.3**	4.5	6.7	12.2	**7.1**	**6.5**
	IVON	Mean	25.5	10.4	5.6	10.3	23.0	27.8	17.1
		MC-1	29.9	12.0	5.3	9.0	23.0	29.1	18.0
		MC-8	12.3	3.5	**2.5**	**3.2**	21.5	22.9	11.0
NLL ↓	Adam	MAP	3.54	1.46	0.45	1.56	1.81	3.65	2.08
		$\text{LLLA}_{\text{diag}}$	1.29	0.69	0.45	1.32	1.57	0.78	1.02
		LA_{diag}	0.97	0.54	0.46	0.64	**0.58**	**0.63**	**0.64**
		$\text{LLLA}_{\text{KFAC}}$	1.36	0.73	0.45	1.38	1.65	0.80	1.06
		LA_{KFAC}	**0.93**	0.49	0.37	0.74	0.81	0.64	0.66
	IVON	Mean	1.97	0.69	0.35	0.64	2.30	3.34	1.55
		MC-1	2.13	0.79	0.38	0.70	2.30	3.34	1.61
		MC-8	1.00	**0.40**	**0.32**	**0.53**	2.05	2.30	1.10
Brier ↓ (100×)	Adam	MAP	67.1	28.9	22.8	39.4	49.5	65.1	45.5
		$\text{LLLA}_{\text{diag}}$	55.3	25.9	22.8	37.9	48.1	50.7	40.1
		LA_{diag}	51.4	26.1	28.6	32.7	**37.1**	44.1	36.7
		$\text{LLLA}_{\text{KFAC}}$	56.3	26.3	22.9	38.1	48.3	51.7	40.6
		LA_{KFAC}	47.2	22.2	21.5	31.3	39.5	**43.9**	**34.3**
	IVON	Mean	53.5	22.0	20.1	29.3	46.1	55.7	37.8
		MC-1	66.0	28.8	22.0	34.2	46.1	58.2	42.5
		MC-8	**46.9**	**20.2**	**19.7**	**27.6**	44.3	50.2	34.8

tions. Computation and memory costs depend on the type of the Laplace approximation; as discussed in Chapter 4, KFAC is significantly more costly than diagonal approximations, which could be problematic on low-resource devices. At inference time, LLA requires running a forward and multiple backward passes (number of classes or tokens) to obtain the gradient for the sampling on the output logits, as shown in Equation 5.10, which makes the inference speed lower than that of training. In contrast, IVON requires multiple forward passes to compute averaged predictions without backward passes, with an additional memory cost of loading optimizer state (two times the number of optimized parameters). Overall, IVON offers a flexible trade-off between calibration and computational and memory cost at inference, while LLA incurs a post-training posterior estimation overhead as well as additional computation and memory cost for the backward passes during inference.

5.1.6 Experiments: Audio Question Answering

Having demonstrated the efficacy of IVON in enhancing predictive accuracy and calibration on natural language understanding tasks, we further evaluate its effectiveness in fine-tuning a multimodal LLM for audio understanding and reasoning tasks, thereby assessing its applicability to multimodal data.

Models and Datasets

We fine-tune the Qwen2.5-Omni 3B model (Xu et al., 2025) with LoRA on the DCASE 2025 Audio Question Answering dataset (Yang et al., 2025), which consists of Bioacoustics QA (BQA, 0.7k), Temporal Soundscapes QA (TSQA, 1k), and Complex QA (CQA, 6.4k) (see Chapter 2.4.2). Qwen2.5-Omni is an end-to-end multimodal LLM designed to perceive diverse modalities, including text, images, audio, and video, while simultaneously generating text and natural speech responses in a streaming manner. Similar to previous experiments, the model is provided with the audio sequence followed by a question and several options, and the task is to predict the correct option. The experiments are conducted using the LLaMAFactory framework[6]. LoRA is applied to all linear layers, with rank set to 8 and alpha set to 16.

Training and Evaluation

All models are trained for 3 epochs with a batch size of 4. For IVON, we use an effective sample size λ of 10^7, a Hessian initialization h_0 of 1×10^{-3}, a learning rate of 0.03 with cosine learning rate decay to 0, and a weight decay δ of 0. For Adam, we train the model using the AdamW optimizer without weight decay and a learning rate of 5×10^{-5} with cosine learning rate decay to 0. Evaluation is performed on the development set using previous metrics with results reported on three subsets respectively. In addition to the averaged results across all samples (Avg.), the averaged scores across three subsets (Domain Avg.) are also calculated.

[6]https://github.com/hiyouga/LLaMA-Factory

For IVON, results are reported under two settings: predictions at the posterior mean, and the average over 8 Monte Carlo samples (MC-8). All results are averaged over 10 runs with different random seeds.

Table 5.2: Comparison of IVON and Adam applied to fine-tuning Qwen2.5-Omni 3B with LoRA on audio question answering tasks. The **best results** are marked.

Metric	Method	BQA	TSQA	CQA	Domain Avg.	Avg.
ACC ↑	Adam	88.57	**67.39**	84.21	80.06	80.45
	IVON Mean	**89.02**	67.16	**85.02**	**80.40**	**80.97**
	IVON MC-8	88.93	67.16	**85.02**	80.37	**80.97**
ECE ↓ (100×)	Adam	9.7	26.2	12.7	16.2	15.7
	IVON Mean	7.4	18.6	9.1	11.7	11.2
	IVON MC-8	**6.6**	**15.6**	**7.9**	**10.0**	**9.5**
NLL ↓	Adam	0.52	1.42	0.71	0.88	0.87
	IVON Mean	0.39	1.09	0.55	0.68	0.67
	IVON MC-8	**0.36**	**0.99**	**0.51**	**0.62**	**0.61**
Brier ↓ (100×)	Adam	20.5	57.0	28.0	35.1	34.5
	IVON Mean	17.4	50.0	24.3	30.6	30.1
	IVON MC-8	**16.9**	**47.7**	**23.5**	**29.4**	**28.9**

Results and Analyses

Overall, the results support our previous findings on commonsense reasoning tasks. In terms of accuracy, IVON evaluated at mean outperforms Adam by 0.52% across all samples and by 0.34% in domain-averaged scores. For reference, directly prompting the base model yields a domain averaged accuracy of 53.8%. For calibration metrics, IVON consistently surpasses Adam across all subsets, with particularly notable gains on BQA and TSQA with limited fine-tuning data. Leveraging 8 MC samples further enhances calibration while maintaining the accuracy of IVON at mean. These results confirm that IVON can serve as a drop-in replacement for Adam, offering improved predictive accuracy and calibration, even in challenging multimodal reasoning scenarios. Moreover, it enables further calibration improvements at test time through multiple inference passes.

5.1.7 Conclusions

In this section, we studied two uncertainty-aware fine-tuning techniques, IVON and LLA, in the context of parameter-efficient fine-tuning with LoRA. Our empirical results demonstrate that IVON evaluated at the posterior mean generally delivers the highest predictive accuracy while offering better-calibrated predictions compared to Adam. In addition, calibration can be further improved by ensembling with multiple Monte Carlo samples. In contrast, LLA, particularly when employing the Kronecker-factored approximation across all LoRA-enabled layers, achieves the best performance in calibration, albeit at the cost of substantially more computation and memory, and more complex implementation.

The strengths of the two methods are complementary: IVON is well-suited for scenarios where high predictive accuracy and inference-time flexibility are priorities, while LLA excels in settings that require well calibrated predictions and are less sensitive to computation and memory costs. These findings suggest that the two techniques should be chosen based on the specific needs of the application: whether it prioritizes predictive accuracy, calibration, or inference efficiency.

5.2 Parameter Importance Estimation

Motivated by the sensitivity-based importance score of the adaptive low-rank adaptation (AdaLoRA), we utilize uncertainty-aware metrics, including the signal-to-noise ratio (SNR), along with the IVON optimizer, for adaptive parameter budget allocation. The resulting Bayesian counterpart not only has matched or surpassed the performance of using the sensitivity-based importance metric but is also a faster alternative to AdaLoRA with Adam. Our theoretical analysis reveals a significant connection between the two metrics, providing a Bayesian perspective on the efficacy of sensitivity as an importance score. Furthermore, our findings suggest that the magnitude, rather than the variance, is the primary indicator of the importance of parameters.

5.2.1 Introduction

In the context of the adaptation of large-scale pre-trained models, it has long been of interest to fine-tune the model in a parameter-efficient manner. Parameter-efficient fine-tuning (PEFT) techniques (Ding et al., 2023a) typically optimize a small subset of the model parameters that are either original or additional ones while leaving the rest unchanged. The low-rank adaptation (LoRA) (Hu et al., 2022) is one of the most efficient and flexible PEFT techniques. Based on the assumption that the change of weights during fine-tuning has a low intrinsic rank, LoRA performs adaptation by optimizing the low-rank approximation of the change of the original weight matrices. Nevertheless, LoRA has limitations as it pre-defines an identical rank for all target weight matrices and therefore ignores the varying importance of weights across modules and layers. This is problematic as adding more trainable parameters to important weights contributes to better performance, however by contrast, doing so to less important weights yields marginal improvements or even inferior outcomes (Zhang et al., 2023b).

In light of the limitations, there arises a natural question of how to allocate trainable parameters to different modules according to their importance to maximize the fine-tuning performance. To this end, a variety of techniques for LoRA has been proposed to address the problem, the most representative one of which is AdaLoRA (Zhang et al., 2023b). AdaLoRA parameterizes the delta weight mimicking the singular value decomposition (SVD) to enable dynamic adjustment of the rank: it identifies the importance of each SVD triplet in the entire model by a sensitivity-based metric and gradually prunes less important triplets during fine-tuning to reach the parameter budget. It has been demonstrated that AdaLoRA can effectively improve the model performance and parameter efficiency compared to LoRA.

Motivated by AdaLoRA, we are primarily interested in the importance scoring mechanism as it can be generically applied to PEFT for parameter selection. The sensitivity-based importance metric is originally based on the heuristic that the importance of parameters can be quantified by the error induced by removing them, which in turn can be approximated by the square of the gradient-weight product (Theis et al., 2018; Molchanov et al., 2019). Meanwhile, there are importance metrics with strong theoretical support, many of which originate from Bayesian neural networks (BNNs). A widely recognized metric is the signal-to-noise ratio (SNR) (Graves, 2011; Blundell et al., 2015; Neklyudov et al., 2017), commonly used in BNN pruning and compression. The interpretation is straightforward: a low SNR makes the neuron's output too noisy to be useful, while a high SNR indicates valuable, low-noise output. The SNR could be a drop-in replacement for the sensitivity-based importance score in AdaLoRA, allowing the pruning of SVD triplets with low SNRs during fine-tuning for dynamic rank adjustment.

The calculation of SNR requires knowledge of the variance of the parameters, typically assuming they follow a Gaussian distribution; this is closely related to VI. VI tackles the optimization task of neural networks by approximating complex posterior distributions of the parameters; this involves selecting a simpler, parameterized distribution and minimizing the Kullback-Leibler (KL) divergence between this distribution and the true posterior. Recent advances in VI (Shen et al., 2024) have shown not only superior performance in calibration and predictive uncertainty estimation compared to traditional optimizers like Adam (Kingma and Ba, 2015), but also high efficiency and effectiveness in large-scale networks.

In this study, we leverage Bayesian importance metrics alongside the IVON optimizer to develop a Bayesian counterpart to AdaLoRA, utilizing SNR as the importance score. By comparing its performance with the sensitivity-based importance metric on the GLUE benchmark (Wang et al., 2019), we demonstrate that the Bayesian approach not only achieves comparable or superior performance but also offers a 10% speed-up over the original AdaLoRA with Adam. A closer examination of the underlying theory reveals a strong connection between these two metrics, providing a Bayesian interpretation of the sensitivity as an importance score. Additionally, our findings indicate that the magnitude, rather than the variance, is the primary indicator of the importance of parameters. The source code is available.[7]

[7] https://github.com/idiap/vilora

5.2.2 Adaptive Budget Allocation

Overview

The techniques that enable adaptively allocating trainable parameters across different modules and layers generally fall into two categories: importance scoring-based methods and regularization-based methods. For importance scoring-based methods, the key is to find a proper importance metric and prune less important components accordingly. Whilst some work (Zhang et al., 2023a; Wang et al., 2024) adopts AdaLoRA's sensitivity-based approach, other heuristic metrics, such as the magnitude of the weight (Mao et al., 2024) and the accumulated gradient (Nikdan et al., 2024), have also been explored. Among regularization-based approaches, diff pruning (Guo et al., 2021) is representative: it applies L_0 regularization to the delta weight (which shares the same dimensions as the pre-trained weights) and prunes it element-wise according to the magnitude. Similarly, but based on LoRA, SoRA (Ding et al., 2023b) introduces a gating unit in-between the two LoRA matrices and applies L_1 regularization to the gate to zero out unimportant ranks. However, regularization-based approaches cannot guarantee to achieve target parameter budgets since they depend on unpredictable sparsity regularizations controlled by sparsity-promoting priors and threshold values, and therefore often require onerous hyperparameter tuning.

Revisiting AdaLoRA

AdaLoRA has the following main components.

SVD-based adaptation AdaLoRA parameterizes the delta weight in the form of singular value decomposition: $\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W} = \mathbf{W}_0 + \mathbf{P}\boldsymbol{\Lambda}\mathbf{Q}$, where $\mathbf{P}$ and $\mathbf{Q}$ are singular vectors and the diagonal matrix $\boldsymbol{\Lambda}$ contains singular values. To avoid the intensive computational cost of SVD, a penalty $R(\mathbf{P},\mathbf{Q}) = ||\mathbf{P}^\top\mathbf{P}-\mathbf{I}||_\mathsf{F}^2 + ||\mathbf{Q}^\top\mathbf{Q}-\mathbf{I}||_\mathsf{F}^2$ is added to the loss to enforce the orthogonality of $\mathbf{P}$ and $\mathbf{Q}$ so that every rank is independent of each other. During adaptation, only the singular values are masked out while the singular vectors are maintained so that dropped triplets can be reactivated later.

Sensitivity-based importance scoring The sensitivity is defined as the magnitude of the gradient-weight product: $I(\theta) = |\theta\nabla_\theta\ell|$, where θ is a trainable parameter. The authors of AdaLoRA argue that the sensitivity itself is too variable and uncertain to be estimated due to the stochasticity of training and therefore propose to use sensitivity smoothing and uncertainty quantification:

$$\begin{aligned} \bar{I}(\theta) &= \beta_1 \bar{I}^{t-1}(\theta) + (1-\beta_1) I^t(\theta) \\ \bar{U}(\theta) &= \beta_2 \bar{U}^{t-1}(\theta) + (1-\beta_2) |I^t(\theta) - \bar{I}^t(\theta)| \end{aligned} \tag{5.12}$$

where $\bar{I}^t$ is the smoothed sensitivity by exponential moving average and $\bar{U}^t$ is the uncertainty quantification of I. The final importance score is $s^t(\theta) = \bar{I}^t(\theta) \cdot \bar{U}^t(\theta)$. The authors compared its performance with the magnitude of singular values and the sensitivity without smoothing and found the proposed metric performed the best.

Global budget scheduler The global budget is defined as the total rank of all delta weights in the model. AdaLoRA starts from an initial budget b^0 that is slightly higher (usually 1.5 times) than the target budget b^T, warms up the training for t_i steps, and gradually decreases the budget b^t to reach b^T following a cubic schedule. After this, the budget distribution is fixed until training finishes after t_f steps.

Bayesian Importance Scores

In this work, we focus on theoretically supported importance metrics that originate from Bayesian neural networks (BNN). BNNs model weights as probability distributions, enabling the network to quantify uncertainties in its predictions. The most commonly used distribution is the Gaussian distribution, therefore the model is parameterized by two sets of parameters: the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ (or the variance $\boldsymbol{\sigma}^2$, we also refer to $\boldsymbol{\sigma}$ as variance for the sake of simplicity).

SNR($\boldsymbol{\theta}$) = $|\boldsymbol{\mu}|/\boldsymbol{\sigma}$ The signal-to-noise ratio (SNR) (Graves, 2011; Blundell et al., 2015; Neklyudov et al., 2017) is a commonly used importance metric in BNN that considers both the magnitude and the variance (also the uncertainty) of the weights. It has a simple interpretation: a low SNR results in a neuron's output being too noisy to be useful, while a high SNR signifies meaningful output with minimal noise. It has been utilized in both in-training and post-training pruning of BNNs (Li et al., 2024; Graves, 2011).

SNR($|\boldsymbol{\theta}|$) Li et al. (Li et al., 2024) argue that the random sampling of weights before each forward pass of BNN needs to be considered. Instead of using $|\mu|$ which is equal to $|\mathbb{E}_q\theta|$ (where q is the posterior distribution of parameters), it is more appropriate to use $\mathbb{E}_q|\theta|$ in the SNR. The resulting metric is:

$$\mathrm{SNR}_q(|\theta|) = \frac{\mu\left(2\Phi\left(\frac{\mu}{\sigma}\right)-1\right)+\frac{2\sigma}{\sqrt{2\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right)}{\sqrt{\sigma^2+\mu^2-\left[\mu\left(2\Phi\left(\frac{\mu}{\sigma}\right)-1\right)+\frac{2\sigma}{\sqrt{2\pi}}\exp\left(-\frac{\mu^2}{2\sigma^2}\right)\right]^2}} \tag{5.13}$$

where $\Phi(x) := \int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right)dy$ is the cumulative distribution function. It has been shown the new metric outperforms the standard SNR in training sparse BNNs (Li et al., 2024).

$|\boldsymbol{\mu}|$ and $1/\boldsymbol{\sigma}$ We want to identify the key component in the SNR that reflects the importance of parameters. The absolute value of the mean, or the magnitude, is a straightforward metric that directly impacts the neuron's output. This metric is widely used in neural network pruning, commonly known as magnitude pruning (Han et al., 2015). Another choice is to use the variance alone as an importance metric. The intuition is that parameters with a low variance have less uncertainty, and therefore are more important.

Method

The calculation of SNR requires approximating a Gaussian distribution over parameters, which is exactly the objective of variational inference. In our experiments, we utilize IVON to estimate the variance of parameters, enabling the use of SNR as an importance metric following AdaLoRA's framework.

5.2.3 Experiments

Models and Datasets

We compare the fine-tuning performance of AdaLoRA using different importance scores on DeBERTaV3-base (He et al., 2023). The experiments are conducted on the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2019), which includes four natural language inference tasks, three similarity and paraphrase tasks, and two single-sentence classification tasks.

Implementation Details

We base our code on the text classification examples of the Hugging Face Transformers library (Wolf et al., 2020) and the Parameter-Efficient Fine-Tuning (PEFT) library (Mangrulkar et al., 2022). For IVON, we use the official implementation[8]. We compare the methods under two budget configurations where the target rank is set to 2 and 4 respectively, resulting in the total trainable parameters being 0.3M and 0.6M (of 86M). Full fine-tuning and LoRA applied to all modules are also added as baselines.

Training and Evaluation

Our experiments are based on the official hyperparameters of AdaLoRA[9] which are optimal when training with Adam. For IVON, the learning rate is set to 0.5 for MRPC and RTE and 0.4 for the rest. Same as Adam, a warm-up stage and the linear decay learning rate schedule are adopted. We found that IVON generally converges slower than Adam at the beginning of training, therefore requiring a much higher learning rate during warm-up for good results especially on small datasets. As a result, for COLA, STS-B, MRPC, and RTE, we use a higher learning rate of 2.0 in the warm-up stage and return to the normal learning rate afterwards. For

[8] https://github.com/team-approx-bayes/ivon
[9] https://github.com/QingruZhang/AdaLoRA

evaluation, we use the best-performing model on the validation set. The results are averaged across 5 runs with different random seeds.

5.2.4 Results and Analyses

The main results are shown in Table 5.3. For MNLI, the "matched" validation set was used for evaluation. Note that we sort the tasks according to dataset sizes and divide them into two groups since we notice that IVON needs extra tricks to ensure good results on small datasets. In general, all PEFT methods outperform full fine-tuning, and AdaLoRA outperforms LoRA. Switching the optimizer from Adam to IVON results in comparable performance, demonstrating that IVON is capable of state-of-the-art performance in PEFT. We further elaborate our findings from the following perspectives.

Comparison of Importance Scores

Both SNR(θ) and SNR($|\theta|$) outperform sensitivity when using IVON, and at least one of the SNR metrics outperforms or ties with the original AdaLoRA with Adam. However, there is no clear winner between the two SNR metrics. This could be explained by the fact that the sparsity level in the AdaLoRA case is not high (only 1/3 of the initial ranks are pruned), and that it is the SVD triplet that is pruned as a parameter group, thus the performance difference between the two metrics is not properly reflected in such a setting. Interestingly, magnitude outperforms sensitivity and one of the SNR metrics especially on small datasets. Magnitude was not experimented in Zhang et al. (2023b). On the one hand, this demonstrates the effectiveness of magnitude pruning; on the other hand, this is probably because the sensitivity or the variance needs more iterations to be estimated accurately given their smoothing nature. Using the variance alone performs the worst among all metrics, however, it still outperforms LoRA with a fixed rank, indicating that the uncertainty of parameters does correlate with the importance.

Visualizing Final Rank Distributions

Figure 5.2 shows the final rank distributions of different methods after fine-tuning the model on MNLI. An obvious difference between Adam and IVON using the sensitivity can be observed comparing (a) and (b), indicating a distinction between the training dynamics of the two optimizers. The distributions of the two SNR metrics (c, d) and the magnitude (e) resemble that of the sensitivity with IVON, which corroborates with quantified results. Unlike the magnitude (e), the variance (f) shows an evenly-distributed pattern. This confirms that the magnitude plays a determining role in reflecting the importance of parameters.

Table 5.3: Main results. The number in model names refers to the target rank. The **best** and the <u>second best</u> results are marked.

Model	Optimizer	Criterion	Group 1 MNLI 393k Acc(m)	Group 1 QQP 364k Acc	Group 1 QNLI 108k Acc	Group 1 SST-2 67k Acc	Group 2 COLA 8.5k Mcc	Group 2 STS-B 7k Corr	Group 2 MRPC 3.7k Acc	Group 2 RTE 2.5k Acc	Group 1	Group 2	All
Full FT	Adam	None	89.89	92.50	94.03	95.73	69.77	91.06	89.75	84.84	93.04	83.86	88.45
LoRA_2	Adam	$r = 2$	89.92	**91.70**	93.97	95.30	69.07	90.89	90.15	87.29	92.72	84.35	88.54
AdaLoRA_2	Adam	Sensitivity	90.40	91.66	**94.49**	<u>95.67</u>	<u>70.78</u>	91.47	90.39	87.00	<u>93.05</u>	84.91	88.98
AdaLoRA_2	IVON	Sensitivity	<u>90.44</u>	<u>91.69</u>	94.36	95.55	69.65	91.89	90.25	87.94	93.01	84.93	88.97
AdaLoRA_2	IVON	$\text{SNR}(\lvert\theta\rvert)$	<u>90.44</u>	91.68	<u>94.40</u>	**95.80**	70.28	**92.04**	90.10	<u>88.16</u>	**93.08**	85.15	89.11
AdaLoRA_2	IVON	$\lvert\mu\rvert/\sigma$	**90.46**	**91.70**	94.33	95.62	70.63	91.90	**90.83**	**88.38**	93.03	**85.43**	**89.23**
AdaLoRA_2	IVON	$\lvert\mu\rvert$	90.42	**91.70**	94.33	95.57	**70.91**	<u>91.99</u>	<u>90.64</u>	87.87	93.01	<u>85.35</u>	<u>89.18</u>
AdaLoRA_2	IVON	$1/\sigma$	90.37	91.32	94.30	95.62	69.35	91.91	90.59	<u>88.16</u>	92.90	85.00	88.95
LoRA_4	Adam	$r = 4$	89.68	**92.03**	94.13	95.32	69.58	90.69	**90.25**	87.08	92.79	84.40	88.59
AdaLoRA_4	Adam	Sensitivity	90.52	<u>91.91</u>	**94.56**	<u>95.78</u>	**69.85**	91.68	**90.25**	88.16	**93.19**	84.98	**89.09**
AdaLoRA_4	IVON	Sensitivity	90.54	91.74	94.45	**95.80**	69.41	**92.03**	89.90	88.09	93.13	84.86	89.00
AdaLoRA_4	IVON	$\text{SNR}(\lvert\theta\rvert)$	<u>90.59</u>	91.78	94.43	95.67	<u>69.83</u>	91.97	89.95	**88.52**	93.11	**85.07**	**89.09**
AdaLoRA_4	IVON	$\lvert\mu\rvert/\sigma$	**90.60**	91.77	<u>94.49</u>	95.69	69.32	<u>92.00</u>	89.95	<u>88.30</u>	<u>93.14</u>	84.89	89.01
AdaLoRA_4	IVON	$\lvert\mu\rvert$	90.56	91.77	94.40	95.55	69.72	91.98	**90.25**	88.23	93.07	<u>85.05</u>	<u>89.06</u>
AdaLoRA_4	IVON	$1/\sigma$	90.48	91.22	94.32	95.62	69.58	91.93	<u>90.20</u>	87.51	92.91	84.80	88.86

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	3	2.4	2.8	2	4.8	5	4.4	5.4	5.6	6	6	4.4
W_k	2.4	2	3	2.4	4.6	4.4	5.2	5.6	6	6	5.8	6
W_v	2.6	3.2	3.8	4.2	4.4	5	5.8	5.8	5.4	5.8	5.8	5.4
W_o	1.6	2	3.2	4.2	5.8	5.8	5.6	5.6	5.8	5.6	6	5
W_{f_1}	4.2	3.4	4.4	5.2	5.8	6	5.6	6	6	6	6	2
W_{f_2}	0.2	0.2	0.4	0.4	1	0.6	0.8	0.8	2	2.6	2.4	0.2

(a) Sensitivity, Adam

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	0.4	0	0.8	1	1.2	2.8	2.4	4.4	4.2	4	5	1.6
W_k	0.2	1	0.6	0.6	3.6	3.6	2.8	4.8	4.8	5	5.6	3.8
W_v	3	3.4	4	5.2	4.8	5	5.6	5.4	5.4	6	6	5.4
W_o	2.8	1.8	3.4	3.8	4.4	5	5.4	5.8	5.8	5.6	5.2	5.2
W_{f_1}	4	4.8	4.4	5.2	5.8	5.6	5.2	6	6	5.8	5.8	3.4
W_{f_2}	3.2	3.8	3.8	4.2	3.8	3.4	5	4.8	5.2	4.8	4.6	2.6

(b) Sensitivity, IVON

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	0.8	1.4	2.2	2.4	3.6	2.8	3.4	4.6	4.4	4.6	5.4	2.6
W_k	1.2	2.8	1.6	2	4.4	4.8	3.4	4.6	4.4	5.2	4.8	3.8
W_v	4.2	2.4	3.6	4	4.4	4.2	4.6	4.8	6	6	6	5.2
W_o	3.4	1.8	3.8	3.8	4.8	5	5	5.2	5.4	4.8	5.4	5.8
W_{f_1}	3.6	3.6	4.2	4.4	5	5.2	5.6	5.6	5.8	5.6	5.8	1.4
W_{f_2}	2.8	2.6	4	3.2	3.6	3.6	3.8	4.4	4.8	5	4.4	3.2

(c) SNR($|\theta|$), IVON

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	1	1	1.8	2.8	3.4	3	3.4	4.8	4.2	4.6	5.4	2.6
W_k	1.4	2.6	2.2	2.2	4	4.8	3.8	4.8	4.4	4.8	5	4.2
W_v	4	2.6	3.8	4.2	4.2	4.4	4.8	5.2	5.8	6	5.8	5.6
W_o	3.2	1.6	3	3.6	4.8	4.8	5.2	5.8	5.4	5	5.4	5.8
W_{f_1}	3.6	3.8	3.8	4.6	5.2	5.4	5.2	5.6	5.8	5.6	5.8	1.4
W_{f_2}	2.8	2.8	4.2	3.2	3.2	3	4	4	4.6	4.8	4.4	2.8

(d) $|\mu|/\sigma$, IVON

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	1.8	0.8	2.4	2.6	3.8	3.2	4	4.2	4.4	4.6	5.4	2.2
W_k	1.4	3.6	0.8	1.8	4.2	4.6	3.8	4.8	4.6	5	5.4	3.8
W_v	4	2.8	4	4.6	4.2	3.8	4.6	4.8	5.8	6	6	5.4
W_o	2.8	2	3.8	3.6	4.4	4.8	5.4	5.2	5.6	4.8	5.4	5.8
W_{f_1}	3.6	3.6	4.4	4.4	5.2	5	5.4	5.6	5.8	5.4	5.8	1.8
W_{f_2}	3	2.2	3.4	3.4	3.6	3.4	3.6	4	4.4	4.8	4.4	3

(e) $|\mu|$, IVON

Layer	1	2	3	4	5	6	7	8	9	10	11	12
W_q	5.2	5.6	5.4	6	4.4	4	5.2	3.8	3.8	5	3.4	4.8
W_k	5.6	5	6	6	3.8	4	4.6	3.8	4.2	3.6	2.8	3.8
W_v	4.4	4.4	4.8	4	3	4.2	3.4	4.2	2.6	2.8	4	2.6
W_o	4.2	4.2	3.6	4.2	4	3	4	3.4	4.4	3.2	3.2	2.2
W_{f_1}	3.4	4	3.6	4	4	3.2	4.6	3	4.2	3.4	3.4	3.4
W_{f_2}	4.6	4.2	4.6	4.2	3.6	3.8	4	3.8	3.6	3.6	3.6	3

(f) $1/\sigma$, IVON

Figure 5.2: Comparison of rank distributions after fine-tuning DeBERTaV3-base on MNLI, with deeper colors indicating higher ranks. Results are averaged across five runs with different random seeds. W_q, W_k, W_v, W_o: weights of the query, key, value, output layers of attention; W_{f_1}, W_{f_2}: weights of the feed-forward layers.

Speed

The variance of the parameter is inferred inherently in IVON, thus the SNR does not require the extra computation of the weight-gradient product of the sensitivity during fine-tuning. On an NVIDIA H100, using the SNR with IVON brings a 10% speed up compared to using the sensitivity with Adam, despite the IVON itself being 1-2% slower than Adam with other conditions kept the same.

A Bayesian Interpretation of Sensitivity

The similarity in performance and the rank distribution between the sensitivity and the SNR suggests a close relationship between them. A closer examination of the underlying theory reveals that sensitivity is, in fact, aligned with the principles of SNR. Specifically, in IVON, the standard deviation $\boldsymbol{\sigma}$ is calculated as $\boldsymbol{\sigma} = 1/\sqrt{\lambda(\mathbf{h}+\delta)}$, where $\mathbf{h}$ is the diagonal Hessian, λ is the effective sample size, and δ is a weight decay term. Notably, $\mathbf{h}$ can be approximated by the expected squared gradient on the training data (Kirkpatrick et al., 2017), $\mathbf{h} \approx \mathbb{E}_{\mathcal{D}}[(\nabla_{\boldsymbol{\theta}}\ell)^2]$, also known as the diagonal of the expected Fisher information matrix (FIM). Consequently, the inverse of the standard deviation, $1/\boldsymbol{\sigma}$, in the context of SNR, is akin to the root mean square of the gradient $\sqrt{\mathbb{E}_{\mathcal{D}}[(\nabla_{\boldsymbol{\theta}}\ell)^2]}$, and therefore analogous to the magnitude of the gradient $|\nabla_{\boldsymbol{\theta}}\ell|$. This implies that the sensitivity $|\theta\nabla_{\theta}\ell|$ has the component $|\nabla_{\theta}\ell|$ acting as an uncertainty measure analogous to $1/\sigma$ in SNR, thereby providing a Bayesian interpretation of the sensitivity as an importance metric. These findings resonate with the comment in Molchanov et al. (2019) that the sensitivity has connections with the FIM. Note that both methods adopt exponential moving average smoothing to compute the global value of the corresponding metric during training. The main difference is that the smoothing is applied to the magnitude of the gradient-weight product in AdaLoRA, while the SNR is computed using the global Hessian tracked by IVON.

5.2.5 Conclusions

In this study, we developed a Bayesian alternative to AdaLoRA, leveraging the signal-to-noise ratio as the importance score with the IVON optimizer. By comparing the performance of different importance metrics, we demonstrated that this Bayesian approach not only matched or surpassed the performance of using the sensitivity-based importance metric on the GLUE benchmark, but was also a faster alternative to the original AdaLoRA with Adam. The theoretical analysis uncovered a significant link between these two metrics, offering a Bayesian perspective on the efficacy of the heuristic sensitivity-based metric as an importance score. Furthermore, our results suggested that the magnitude, rather than the variance, served as the key indicator of the importance of parameters.

6 Conclusions and Future Work

6.1 Conclusions

This thesis presented three broad contributions across two primary phases, moving from ad hoc, model-specific adaptation towards more generalized PEFT frameworks.

In the first phase, we investigated the integration of diffusion models into adaptive TTS systems based on encoder-decoder architectures. Central to this effort was the use of adaptive layer normalization to condition the diffusion process on text representations, enabling parameter-efficient adaptation. On standard TTS tasks, the proposed architecture was shown to be a faster alternative to its convolutional counterpart. In few-shot adaptation scenarios, the new decoder demonstrated clear improvements in naturalness and speaker similarity over a transformer-based decoder, while maintaining parameter efficiency. The effectiveness of the approach was further validated through participation in the Blizzard Challenge 2023, where our system achieved competitive rankings in synthesis quality and naturalness.

The second phase of the thesis transitioned to exploring more general PEFT frameworks. A first contribution in this phase addressed the critical issue of catastrophic forgetting during fine-tuning, which can degrade a pre-trained model's inherent capabilities and overall generalizability. We demonstrated that Bayesian transfer learning techniques, through estimating a posterior distribution over pre-trained model parameters using Laplace approximation, can serve as an effective regularizer within the PEFT paradigm that guides parameter updates to preserve pre-training knowledge. Through a series of experiments on language modeling and TTS tasks, we showed that applying established Laplace approximations to regularize LoRA-based PEFT could overcome catastrophic forgetting without compromising fine-tuning performance, and the Kronecker-factored approximation provided superior preservation of pre-training knowledge compared to the diagonal ones.

Finally, we extended our exploration of Bayesian learning by investigating variational inference as a more flexible and expressive alternative to Laplace-based methods. Using the IVON optimizer, we first demonstrated improved predictive accuracy and calibration in PEFT and compared it with Laplace approximation. Furthermore, we leveraged IVON's online posterior estimates to develop a Bayesian approach for identifying and pruning redundant LoRA components. This enabled automatic, layer-wise allocation of the parameter budget, leading to enhanced performance and efficiency while providing a principled Bayesian interpretation of common importance scoring strategies used in PEFT parameter selection.

6.2 Future Work

The architectural unification across domains has facilitated the widespread adoption of generic adaptation techniques such as PEFT, leading to a diminished distinction among different adaptation targets. As a result, adaptation is increasingly framed as a general transfer learning problem. In light of this trend, we discuss several potential directions for future research to advance current transfer learning and PEFT frameworks, particularly from a Bayesian perspective.

This thesis has demonstrated the potential of Bayesian learning approaches that estimate posterior distributions over network parameters for a variety of applications. Specifically, the Laplace approximation enables efficient estimation of the posterior around a mode using limited data, providing both a mechanism for preserving pre-trained knowledge during adaptation and a means for uncertainty quantification to improve calibration. With regard to the former, transfer learning is in fact closely related to continual learning, as discussed in Chapter 4. In this context, Laplace-based methods can be applied to support continual learning of LoRA adapters, enabling modular integration of task-specific knowledge without mutual interference. For example in TTS, LoRA adapters could be designed as plug-in modules with disentangled functionalities, such as one encoding speaker identity and another encoding emotional tone, allowing for compositional control over the same base model. On the other hand, despite the effectiveness of Laplace-based methods, particularly those employing Kronecker-factored approximations, they can still incur significant computational and memory costs even when applied to a subset of parameters and using few data. Thus, identifying more computationally and memory-efficient Hessian estimations while maintaining precise posterior estimation, is a valuable direction for further investigation. Furthermore, beyond predictive uncertainty estimation, the Laplace approximation can also be valuable in generative settings: for example, to detect flawed outputs, filter low-quality samples, or identify out-of-domain inputs that may lead to unreliable generations.

Variational inference offers a more flexible and expressive framework for posterior estimation during training. However, its practical adoption is often hindered by the computational overhead incurred during inference, particularly when multiple samples must be drawn to form an ensemble prediction, thereby reducing inference speed in proportion to the number of samples. To address this limitation, future work could explore more efficient sampling strategies that minimize or eliminate the need for repeated forward passes. Additionally, given the demonstrated effectiveness of last-layer Laplace approximations in mitigating overconfidence in predictions, it would be worthwhile to investigate the application of variational inference to specific model components that perform prediction tasks prone to such issues to improve the model's overall performance. Another promising avenue lies in the combination of Laplace approximation and variational inference techniques for adaptation. One potential approach is to use the Laplace approximation to estimate the loss landscape around a pre-trained mode and initializing the variational posterior with the corresponding Hessian approximation. This would allow prior information from the pre-trained model to be incorporated directly into the variational fine-tuning process, potentially enhancing performance and generalizability.

Bibliography

Antorán, J., Janz, D., Allingham, J. U., Daxberger, E. A., Barbano, R., Nalisnick, E. T., and Hernández-Lobato, J. M. (2022). Adapting the linearised Laplace model evidence for modern deep learning. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 796–821. PMLR.

Arik, S. Ö., Chen, J., Peng, K., Ping, W., and Zhou, Y. (2018). Neural voice cloning with a few samples. In *Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10040–10050.

Ba, L. J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *arXiv CoRR*, abs/1607.06450.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Black, S. et al. (2022). GPT-NeoX-20B: An open-source autoregressive language model. *arXiv CoRR*, abs/2204.06745.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37, pages 1613–1622. JMLR.org.

Borsos, Z. et al. (2023). AudioLM: A language modeling approach to audio generation. *IEEE ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Botev, A., Ritter, H., and Barber, D. (2017). Practical Gauss-Newton optimisation for deep learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 557–565. PMLR.

Brown, T. B. et al. (2020). Language models are few-shot learners. In *Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cai, H., Gan, C., Zhu, L., and Han, S. (2020). TinyTL: Reduce memory, not parameters for efficient on-device learning. In *Advances in Neural Information Processing Systems 33:*

Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Casanova, E. et al. (2022). YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In *International Conference on Machine Learning, ICML 2022*, volume 162, pages 2709–2720.

Chen, H. and Garner, P. N. (2023a). Diffusion transformer for adaptive text-to-speech. In *12th ISCA Speech Synthesis Workshop, SSW 2023, Grenoble, France, August 26-28, 2023*, pages 157–162. ISCA.

Chen, H. and Garner, P. N. (2023b). An investigation into the adaptability of a diffusion-based TTS model. *arXiv CoRR*, abs/2303.01849.

Chen, H. and Garner, P. N. (2024). Bayesian parameter-efficient fine-tuning for overcoming catastrophic forgetting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4253–4262.

Chen, H. and Garner, P. N. (2025). A Bayesian interpretation of adaptive low-rank adaptation. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Chen, H., He, M., de Gibson, L. C., and Garner, P. N. (2023). The Idiap speech synthesis system for the Blizzard challenge 2023. In *18th Blizzard Challenge Workshop, Grenoble, France, August 29, 2023*. ISCA.

Chen, M. et al. (2021). AdaSpeech: Adaptive text to speech for custom voice. In *9th International Conference on Learning Representations, ICLR*.

Chen, S. et al. (2022). WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.*, 16(6):1505–1518.

Chen, S. et al. (2024). VALL-E 2: Neural codec language models are human parity zero-shot text to speech synthesizers. *arXiv CoRR*, abs/2406.05370.

Choi, B. J., Jeong, M., Lee, J. Y., and Kim, N. S. (2022). SNAC: speaker-normalized affine coupling layer in flow-based architecture for zero-shot multi-speaker text-to-speech. *IEEE Signal Process. Lett.*, 29:2502–2506.

Clark, C., Lee, K., Chang, M., Kwiatkowski, T., Collins, M., and Toutanova, K. (2019). BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. ACL.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. (2018). Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv CoRR*, abs/1803.05457.

Cong, B., Daheim, N., Shen, Y., Cremers, D., Yokota, R., Khan, M. E., and Möllenhoff, T. (2024). Variational low-rank adaptation using IVON. *arXiv CoRR*, abs/2411.04421.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y., and Défossez, A. (2023). Simple and controllable music generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., and Hennig, P. (2021). Laplace redux - effortless Bayesian deep learning. In *Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20089–20103.

Défossez, A., Copet, J., Synnaeve, G., and Adi, Y. (2023). High fidelity neural audio compression. *Trans. Mach. Learn. Res.*, 2023.

Défossez, A., Mazaré, L., Orsini, M., Royer, A., Pérez, P., Jégou, H., Grave, E., and Zeghidour, N. (2024). Moshi: a speech-text foundation model for real-time dialogue. *arXiv CoRR*, abs/2410.00037.

Deng, Z., Zhou, F., and Zhu, J. (2022). Accelerated linearized laplace approximation for Bayesian deep learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 3830–3834. ISCA.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Ding, N. et al. (2023a). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nat. Mac. Intell.*, 5(3):220–235.

Ding, N. et al. (2023b). Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 4133–4145.

Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Edalati, A., Tahaei, M. S., Kobyzev, I., Nia, V. P., Clark, J. J., and Rezagholizadeh, M. (2023). KronA: Parameter efficient tuning with kronecker adapter. In *ENLSP-III NeurIPS Workshop*.

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48, pages 1050–1059. JMLR.org.

Gao, L. et al. (2021). The Pile: An 800GB dataset of diverse text for language modeling. *arXiv CoRR*, abs/2101.00027.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio Set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pages 776–780. IEEE.

George, T., Laurent, C., Bouthillier, X., Ballas, N., and Vincent, P. (2018). Fast approximate natural gradient descent in a Kronecker factored eigenbasis. In *Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 9573–9583.

Goodfellow, I. J., Mirza, M., Da, X., Courville, A. C., and Bengio, Y. (2014). An empirical investigation of catastrophic forgeting in gradient-based neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*.

Graves, A. (2011). Practical variational inference for neural networks. In *Annual Conference on Neural Information Processing Systems 2011. 12-14 December 2011, Granada, Spain*, pages 2348–2356.

Guo, D., Rush, A. M., and Kim, Y. (2021). Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4884–4896.

Hajj, M., Lenglet, M., Perrotin, O., and Bailly, G. (2022). Comparing NLP solutions for the disambiguation of French heterophonic homographs for end-to-end TTS systems. In *Speech and Computer - 24th International Conference, SPECOM 2022, Gurugram, India, November 14-16, 2022, Proceedings*, volume 13721, pages 265–278. Springer.

Han, S., Pool, J., Tran, J., and Dally, W. J. (2015). Learning both weights and connections for efficient neural network. In *Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1135–1143.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., and Neubig, G. (2022). Towards a unified view of parameter-efficient transfer learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

He, P., Gao, J., and Chen, W. (2023). DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*.

Houlsby, N. et al. (2019). Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, pages 2790–2799.

Hsu, W., Bolte, B., Tsai, Y. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3451–3460.

Hsu, W., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., Nguyen, P., and Pang, R. (2019). Hierarchical generative modeling for controllable speech synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Hu, E. J. et al. (2022). LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Huang, R. et al. (2024). InstructSpeech: Following speech editing instructions via large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Huang, R., Lam, M. W. Y., Wang, J., Su, D., Yu, D., Ren, Y., and Zhao, Z. (2022a). FastDiff: A fast conditional diffusion model for high-quality speech synthesis. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 4157–4163.

Huang, R., Ren, Y., Liu, J., Cui, C., and Zhao, Z. (2022b). GenerSpeech: Towards style transfer for generalizable out-of-domain text-to-speech synthesis. *arXiv CoRR*, abs/2205.07211.

Huang, R., Zhao, Z., Liu, H., Liu, J., Cui, C., and Ren, Y. (2022c). ProDiff: Progressive fast diffusion model for high-quality text-to-speech. In *MM '22: The 30th ACM International Conference on Multimedia*, pages 2595–2605.

Huang, W., Cooper, E., Tsao, Y., Wang, H., Toda, T., and Yamagishi, J. (2022d). The VoiceMOS challenge 2022. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, pages 4536–4540. ISCA.

Hyeon-Woo, N., Ye-Bin, M., and Oh, T. (2022). FedPara: Low-rank hadamard product for communication-efficient federated learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Immer, A., Korzepa, M., and Bauer, M. (2021a). Improving predictions of Bayesian neural nets via local linearization. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130, pages 703–711. PMLR.

Immer, A., Korzepa, M., and Bauer, M. (2021b). Improving predictions of Bayesian neural nets via local linearization. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 703–711. PMLR.

Ito, K. (2017). The LJ Speech dataset. https://keithito.com/LJ-Speech-Dataset/.

Iyer, S., Dandekar, N., and Csernai, K. (2019). Quora question pairs dataset.

Jeong, M., Kim, H., Cheon, S. J., Choi, B. J., and Kim, N. S. (2021). Diff-TTS: A denoising diffusion model for text-to-speech. In *Interspeech 2021*, pages 3605–3609.

Jiang, Z., Araki, J., Ding, H., and Neubig, G. (2021). How can we know *When* language models know? on the calibration of language models for question answering. *Trans. Assoc. Comput. Linguistics*, 9:962–977.

Ju, X., Gao, Y., Zhang, Z., Yuan, Z., Wang, X., Zeng, A., Xiong, Y., Xu, Q., and Shan, Y. (2024). MiraData: A large-scale video dataset with long durations and structured captions. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Kahn, J. et al. (2020). Libri-Light: A benchmark for ASR with limited or no supervision. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 7669–7673. IEEE.

Kalchbrenner, N. et al. (2018). Efficient neural audio synthesis. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2415–2424. PMLR.

Kang, M., Min, D., and Hwang, S. J. (2022). Any-speaker adaptive text-to-speech synthesis with diffusion models. *arXiv CoRR*, abs/2211.09383.

Kao, T., Jensen, K. T., van de Ven, G., Bernacchia, A., and Hennequin, G. (2021). Natural continual learning: success is a journey, not (just) a destination. In *Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 28067–28079.

Khan, M. E., Nielsen, D., Tangkaratt, V., Lin, W., Gal, Y., and Srivastava, A. (2018). Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2616–2625. PMLR.

Kim, J., Kim, S., Kong, J., and Yoon, S. (2020). Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. In *Advances in Neural Information Processing Systems 2020*.

Kim, J., Kong, J., and Son, J. (2021). Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *38th International Conference on Machine Learning, ICML*, volume 139, pages 5530–5540.

Kim, S., Kim, H., and Yoon, S. (2022). Guided-TTS 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data. *arXiv CoRR*, abs/2205.15370.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*.

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10236–10245.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*.

Kirkpatrick, J. et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Kong, J., Kim, J., and Bae, J. (2020). HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems 2020*.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2021). Diffwave: A versatile diffusion model for audio synthesis. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Kopiczko, D. J., Blankevoort, T., and Asano, Y. M. (2024). VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Kristiadi, A., Hein, M., and Hennig, P. (2020). Being Bayesian, even just a bit, fixes overconfidence in ReLU networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119, pages 5436–5446. PMLR.

Kumar, A., Tan, K., Ni, Z., Manocha, P., Zhang, X., Henderson, E., and Xu, B. (2023). Torchaudio-Squim: Reference-less speech quality and intelligibility measures in torchaudio. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Kunstner, F., Hennig, P., and Balles, L. (2019). Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4158–4169.

Lajszczak, M. et al. (2024). BASE TTS: lessons from building a billion-parameter text-to-speech model on 100k hours of data. *arXiv CoRR*, abs/2402.08093.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Lam, M. W. Y., Wang, J., Su, D., and Yu, D. (2022). BDDM: bilateral denoising diffusion models for fast and high-quality speech synthesis. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Le, M., Vyas, A., Shi, B., Karrer, B., Sari, L., Moritz, R., Williamson, M., Manohar, V., Adi, Y., Mahadeokar, J., et al. (2023). Voicebox: Text-guided multilingual universal speech generation at scale. *arXiv preprint arXiv:2306.15687*.

Lee, S., Kim, H., Shin, C., Tan, X., Liu, C., Meng, Q., Qin, T., Chen, W., Yoon, S., and Liu, T. (2022). PriorGrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3045–3059. ACL.

Li, J., Miao, Z., Qiu, Q., and Zhang, R. (2024). Training Bayesian neural networks with sparse subspace variational inference. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Li, X., Grandvalet, Y., and Davoine, F. (2018). Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine*

Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80, pages 2830–2839. PMLR.

Li, X. L. and Liang, P. (2021). Prefix-Tuning: Optimizing continuous prompts for generation. In *Proceedings of the ACL/IJCNLP 2021, Virtual Event, August 1-6, 2021*, pages 4582–4597.

Li, Y. A., Han, C., Raghavan, V. S., Mischler, G., and Mesgarani, N. (2023). StyleTTS 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Li, Z. and Hoiem, D. (2016). Learning without forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016*, volume 9908, pages 614–629. Springer.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Liu, H. et al. (2022a). Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Liu, J., Li, C., Ren, Y., Chen, F., and Zhao, Z. (2022b). DiffSinger: Singing voice synthesis via shallow diffusion mechanism. In *36th AAAI Conference on Artificial Intelligence*, pages 11020–11028.

Liu, S., Su, D., and Yu, D. (2022c). DiffGAN-TTS: High-fidelity and efficient text-to-speech with denoising diffusion gans. *arXiv CoRR*, abs/2201.11972.

Liu, S., Wang, C., Yin, H., Molchanov, P., Wang, Y. F., Cheng, K., and Chen, M. (2024a). DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Liu, W. et al. (2024b). Parameter-efficient orthogonal finetuning via butterfly factorization. In *The Twelfth International Conference on Learning Representations*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Luo, Y., Yang, Z., Meng, F., Li, Y., Zhou, J., and Zhang, Y. (2023). An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv CoRR*, abs/2308.08747.

MacKay, D. J. C. (1992). A Practical Bayesian Framework for Backpropagation Networks. *Neural Computation*, 4(3):448–472.

Maddox, W. J., Izmailov, P., Garipov, T., Vetrov, D. P., and Wilson, A. G. (2019). A simple baseline for Bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13132–13143.

Mahabadi, R. K., Henderson, J., and Ruder, S. (2021a). Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 1022–1035.

Mahabadi, R. K., Ruder, S., Dehghani, M., and Henderson, J. (2021b). Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 565–576. Association for Computational Linguistics.

Mangrulkar, S. et al. (2022). PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Mao, Y. et al. (2022). UniPELT: A unified framework for parameter-efficient language model tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 6253–6264. ACL.

Mao, Y., Huang, K., Guan, C., Bao, G., Mo, F., and Xu, J. (2024). DoRA: Enhancing parameter-efficient fine-tuning with dynamic rank distribution. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11662–11675.

Martens, J. and Grosse, R. B. (2015). Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37, pages 2408–2417. JMLR.org.

McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., and Sonderegger, M. (2017). Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech 2017*, pages 498–502. ISCA.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press.

Mehta, S., Tu, R., Beskow, J., Székely, É., and Henter, G. E. (2024). Matcha-TTS: A fast TTS architecture with conditional flow matching. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*, pages 11341–11345. IEEE.

Meng, L. et al. (2024). Autoregressive speech synthesis without vector quantization. *arXiv CoRR*, abs/2407.08551.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2017). Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017*. OpenReview.net.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. (2018). Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2381–2391. Association for Computational Linguistics.

Min, D., Lee, D. B., Yang, E., and Hwang, S. J. (2021). Meta-StyleSpeech : Multi-speaker adaptive text-to-speech generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139, pages 7748–7759.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., and Kautz, J. (2019). Importance estimation for neural network pruning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11264–11272. IEEE.

Mosbach, M., Pimentel, T., Ravfogel, S., Klakow, D., and Elazar, Y. (2023). Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the ACL: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12284–12314. ACL.

Naderi, B. and Cutler, R. (2020). An open source implementation of ITU-T recommendation P.808 with validation. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2862–2866. ISCA.

Neklyudov, K., Molchanov, D., Ashukha, A., and Vetrov, D. P. (2017). Structured Bayesian pruning via log-normal multiplicative noise. In *Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6775–6784.

Nikdan, M., Tabesh, S., and Alistarh, D. (2024). RoSA: Accurate parameter-efficient fine-tuning via robust adaptation. *arXiv CoRR*, abs/2401.04679.

OpenAI (2023). GPT-4 technical report. *arXiv CoRR*, abs/2303.08774.

Osawa, K. et al. (2019). Practical deep learning with Bayesian principles. In *Annual Conference on Neural Information Processing Systems 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4289–4301.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Pascanu, R. and Bengio, Y. (2014). Revisiting natural gradient for deep networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*.

Peebles, W. and Xie, S. (2022). Scalable diffusion models with transformers. *arXiv CoRR*, abs/2212.09748.

Peng, P., Huang, P., Li, S., Mohamed, A., and Harwath, D. (2024). VoiceCraft: Zero-shot speech editing and text-to-speech in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 12442–12462. Association for Computational Linguistics.

Perrotin, O., Stephenson, B., Gerber, S., and Bailly, G. (2023). The Blizzard Challenge 2023. In *Proc. 18th Blizzard Challenge Workshop*, pages 1–27.

Pfeiffer, J., Vulic, I., Gurevych, I., and Ruder, S. (2020). MAD-X: an adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7654–7673. ACL.

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. A. (2021). Grad-TTS: A diffusion probabilistic model for text-to-speech. In *38th International Conference on Machine Learning, ICML*, volume 139, pages 8599–8608.

Prenger, R., Valle, R., and Catanzaro, B. (2019). WaveGlow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 3617–3621. IEEE.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 28492–28518. PMLR.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. ACL.

Rao, K., Peng, F., Sak, H., and Beaufays, F. (2015). Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 4225–4229. IEEE.

Ravanelli, M. et al. (2021). SpeechBrain: A general-purpose speech toolkit. arXiv:2106.04624.

Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T. (2021a). FastSpeech 2: Fast and high-quality end-to-end text to speech. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Ren, Y., Liu, J., and Zhao, Z. (2021b). PortaSpeech: Portable and high-quality generative text-to-speech. In *Advances in Neural Information Processing Systems 2021*, pages 13963–13974.

Ritter, H., Botev, A., and Barber, D. (2018a). Online structured laplace approximations for overcoming catastrophic forgetting. In *Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 3742–3752.

Ritter, H., Botev, A., and Barber, D. (2018b). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rix, A. W., Beerends, J. G., Hollier, M. P., and Hekstra, A. P. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings*, pages 749–752. IEEE.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.

Roux, J. L., Wisdom, S., Erdogan, H., and Hershey, J. R. (2019). SDR - half-baked or well done? In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pages 626–630. IEEE.

Rubenstein, P. K. et al. (2023). AudioPaLM: A large language model that can speak and listen. *arXiv CoRR*, abs/2306.12925.

Saeki, T., Xin, D., Nakata, W., Koriyama, T., Takamichi, S., and Saruwatari, H. (2022). UTMOS: UTokyo-SaruLab system for VoiceMOS challenge 2022. In *23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022*, pages 4521–4525. ISCA.

Saharia, C. et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 36479–36494.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. (2020). WinoGrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8732–8740. AAAI Press.

Salimans, T. and Ho, J. (2022). Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Ryan, R., Saurous, R. A., Agiomyrgiannakis, Y., and Wu, Y. (2018). Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4779–4783. IEEE.

Shen, K., Ju, Z., Tan, X., Liu, Y., Leng, Y., He, L., Qin, T., Zhao, S., and Bian, J. (2023). Natural-Speech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv CoRR*, abs/2304.09116.

Shen, Y. et al. (2024). Variational learning is effective for large deep networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 44665–44686. PMLR.

Smith, J. S. et al. (2024). Continual diffusion: Continual customization of text-to-image diffusion with c-loRA. *Transactions on Machine Learning Research*.

Socher, R. et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Seattle, Washington, USA*, pages 1631–1642. ACL.

Song, Y., Chen, Z., Wang, X., Ma, Z., and Chen, X. (2025). ELLA-V: stable neural codec language modeling with alignment-guided sequence reordering. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 25174–25182. AAAI Press.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. (2010). A short-time objective intelligibility measure for time-frequency weighted noisy speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19 March 2010, Sheraton Dallas Hotel, Dallas, Texas, USA*, pages 4214–4217. IEEE.

Theis, L., Korshunova, I., Tejani, A., and Huszár, F. (2018). Faster gaze prediction with dense networks and Fisher pruning. *arXiv CoRR*, abs/1801.05787.

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. D. (2023). Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5433–5442. Association for Computational Linguistics.

Touvron, H. et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv CoRR*, abs/2307.09288.

Valle, R., Shih, K. J., Prenger, R., and Catanzaro, B. (2021). Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. In *9th International Conference on Learning Representations, ICLR*.

van de Ven, G. M., Tuytelaars, T., and Tolias, A. S. (2022). Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv CoRR*, abs/1609.03499.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Vyas, A. et al. (2023). Audiobox: Unified audio generation with natural language prompts. *arXiv CoRR*, abs/2312.15821.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Wang, C. et al. (2023a). Neural codec language models are zero-shot text to speech synthesizers. *arXiv CoRR*, abs/2301.02111.

Wang, H., Liu, T., Zhao, T., and Gao, J. (2024). RoseLoRA: Row and column-wise sparse low-rank adaptation of pre-trained language model for knowledge editing and fine-tuning. *arXiv CoRR*, abs/2406.10777.

Wang, X. et al. (2023b). Orthogonal subspace learning for language model continual learning. In *Findings of the ACL: EMNLP 2023, Singapore, December 6-10, 2023*, pages 10658–10671. ACL.

Wang, X. et al. (2025). Spark-TTS: An efficient llm-based text-to-speech model with single-stream decoupled speech tokens. *arXiv CoRR*, abs/2503.01710.

Wang, Y. et al. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *35th International Conference on Machine Learning, ICML*, volume 80, pages 5167–5176.

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. A. (2017). Tacotron: Towards end-to-end speech synthesis. In *Interspeech 2017*, pages 4006–4010. ISCA.

Williams, A., Nangia, N., and Bowman, S. R. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122. ACL.

Wolf, T. et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Wu, Y., Tan, X., Li, B., He, L., Zhao, S., Song, R., Qin, T., and Liu, T. (2022). AdaSpeech 4: Adaptive text to speech in zero-shot scenarios. *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pages 2568–2572.

Xiang, J. et al. (2023). Language models meet world models: Embodied experiences enhance language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xin, D. et al. (2024). RALL-E: robust codec language modeling with chain-of-thought prompting for text-to-speech synthesis. *arXiv CoRR*, abs/2404.03204.

Xu, J. et al. (2025). Qwen2.5-Omni technical report. *arXiv CoRR*, abs/2503.20215.

Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J. (2009). Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):66–83.

Yamagishi, J., Veaux, C., and MacDonald, K. (2019). CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92).

Yang, A. X., Robeyns, M., Wang, X., and Aitchison, L. (2024). Bayesian low-rank adaptation for large language models. In *The Twelfth International Conference on Learning Representations*.

Yang, C. H. et al. (2025). Multi-domain audio question answering toward acoustic content reasoning in the DCASE 2025 challenge. *arXiv CoRR*, abs/2505.07365.

Yeh, S., Hsieh, Y., Gao, Z., Yang, B. B. W., Oh, G., and Gong, Y. (2024). Navigating text-to-image customization: From LyCORIS fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*.

Zaken, E. B., Goldberg, Y., and Ravfogel, S. (2022). BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. ACL.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. (2022). SoundStream: An end-to-end neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:495–507.

Zen, H. et al. (2019). LibriTTS: A corpus derived from LibriSpeech for text-to-speech. In *Interspeech 2019*, pages 1526–1530.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 3987–3995. PMLR.

Zhang, F., Li, L., Chen, J., Jiang, Z., Wang, B., and Qian, Y. (2023a). IncreLoRA: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv CoRR*, abs/2308.12043.

Zhang, Q. et al. (2023b). Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Zhang, S. et al. (2022). OPT: open pre-trained transformer language models. *arXiv CoRR*, abs/2205.01068.

Zhang, Z. et al. (2023c). Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *arXiv CoRR*, abs/2303.03926.

Zheng, Y., Li, X., Xie, F., and Lu, L. (2020). Improving end-to-end speech synthesis with local recurrent neural network enhanced transformer. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*, pages 6734–6738.

Zhu, J., Zhang, C., and Jurgens, D. (2022). ByT5 model for massively multilingual grapheme-to-phoneme conversion. In *Interspeech 2022*, pages 446–450. ISCA.

Haolin Chen

haolin-chen@outlook.com

EXPERIENCE

Idiap Research Institute, Martigny, Switzerland

- Research Assistant / Ph.D. Student — Aug 2021 – Sep 2025
- Thesis: Efficient Adaptation for Speech Technology
  - Deep generative models and their adaptation for text-to-speech synthesis (TTS)
  - Bayesian transfer learning for generalizable parameter-efficient fine-tuning (PEFT)
  - Variational learning for importance scoring and uncertainty estimation in PEFT
- Advisor: Dr. Philip N. Garner
- Keywords: TTS, deep generative model, parameter-efficient fine-tuning, Bayesian learning

Amazon, Cambridge, United Kingdom

- Applied Scientist Intern — Sep 2024 – Feb 2025
- Project: Natural language controlled text-to-speech synthesis
  - Automatic speech captioning pipeline applied to large-scale speech data
  - Chain-of-thought (CoT) prompting for enhanced robustness and controllability
  - Supervised fine-tuning (SFT) of speech-to-speech multimodal foundation model
- Team: Amazon AGI TTS
- Keywords: controllable TTS, large language model, speech foundation model, multimodality

Center for Speech and Language Technologies (CSLT), Tsinghua University, Beijing, China

- Research Assistant — Mar 2019 – Aug 2020
- Research on various topics in speech processing:
  - Deep generative models for speech enhancement
  - Continual learning for speech recognition
  - CN-Celeb speaker recognition dataset audiovisual automatic collection pipeline
- Advisor: Associate Professor Dong Wang
- Keywords: deep generative model, speech enhancement, speech recognition, speaker recognition

EDUCATION

EPFL (École polytechnique fédérale de Lausanne), Lausanne, Switzerland

- Ph.D. in Electrical Engineering — Aug 2021 – Oct 2025

Tsinghua University, Beijing, China

- B.Eng. in Electrical Engineering and Automation — Aug 2017 – Jun 2021

PUBLICATIONS

JOURNALS

[1] H. Chen and P. N. Garner, "Bayesian parameter-efficient fine-tuning for overcoming catastrophic forgetting" *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 2024.

CONFERENCES

[1] H. Chen and P. N. Garner, "A Bayesian interpretation of adaptive low-rank adaptation" In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 2025 (ICASSP 2025)*.

[2] P. Korshunov, H. Chen, P. N. Garner, and S. Marcel, "Vulnerability of automatic identity recognition to audio-visual deepfakes" In *Proc. IEEE International Joint Conference on Biometrics 2023 (IJCB 2023)*.

[3] H. Chen, M. He, L. Coppieters, and P. N. Garner, "The Idiap speech synthesis system for the Blizzard Challenge 2023" In *Proc. 18th Blizzard Challenge Workshop (Blizzard 2023)*.

[4] H. Chen and P. N. Garner, "Diffusion transformer for adaptive text-to-speech" In *Proc. 12th ISCA Speech Synthesis Workshop (SSW 2023)*.

[5] F. Mai, A. Pannatier, F. Fehr, H. Chen, F. Marelli, F. Fleuret, and J. Henderson, "HyperMixer: an MLP-based low cost alternative to transformers" In *Proc. 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

[6] Y. Shi, H. Chen, L. Li, Z. Tang, D. Wang, and J. Han, "Can we trust deep speech prior?" In *Proc. IEEE Spoken Language Technology Workshop 2021 (SLT 2021)*.

[7] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "CN-Celeb: a challenging Chinese speaker recognition dataset" In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing 2020 (ICASSP 2020)*.

[*Updated on 2025-09-11*]