UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ADAPTIVE INFERENCE UNDER DISTRIBUTION SHIFT: CONDITIONAL DIFFUSION, TEST-TIME COMPUTE, AND SELECTIVE ATTENTION

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

by AHSAN BILAL

Norman, Oklahoma

2026

ADAPTIVE INFERENCE UNDER DISTRIBUTION SHIFT: CONDITIONAL DIFFUSION, TEST-TIME COMPUTE, AND SELECTIVE ATTENTION

A THESIS APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Dean F. Hougen (Chair)

Dr. Mohammed Atiquzzaman

Dr. Jie Cao

Acknowledgments

I would like to express my deepest gratitude to my advisor, Dr. Dean F. Hougen, for his guidance, support, and mentorship throughout my Master's studies. His insights and encouragement have been invaluable in shaping this research.

I am grateful to Prof. John M. Cioffi at Stanford University for the opportunity to collaborate on wireless channel estimation research and for his expert guidance on diffusion models and signal processing.

I would like to thank my thesis committee members, Dr. Mohammed Atiquzzaman and Dr. Jie Cao, for their valuable feedback and suggestions that improved this work.

Special thanks to my collaborators Muhammad Ahmed Mohsin, Ahmad Salman, and Khawar Khurshid for their contributions to the research presented in this thesis.

I acknowledge the computational resources provided by the University of Oklahoma and the support from the School of Computer Science.

Finally, I thank my family for their unwavering support and encouragement throughout this journey.

Table of Contents

List of Tables

List of Figures

Abstract

Distribution shift manifests differently across machine learning domains, shaping both the challenge and the solution in each case. In wireless channel estimation, the shift is continuous and physical: user mobility induces time-varying Doppler dynamics that evolve channel statistics on the timescale of seconds. In mathematical reasoning, shift is discrete and semantic: problem difficulty varies across instances in ways not observable from the input, so a system cannot know in advance how much computation a problem warrants. In medical imaging, shift is institutional and acquisitional: heterogeneity in scanner hardware and patient populations produces covariate shift, with pathological patterns appearing at varying scales and locations. Despite these differences, a common failure mode unites the three settings: static inference that commits to fixed computational strategies wastes resources when conditions are favorable and underperforms when they are demanding. This thesis develops adaptive inference mechanisms built on a shared principle of conditioning computational allocation on input-dependent signals, instantiated differently in each domain: for wireless channels as signal-to-noise ratio (SNR) combined with a learned temporal context; for large language model (LLM) reasoning as an intermediate quality signal from a learned Process Reward Model (PRM) evaluating trajectories in progress; and for medical imaging as a spatial attention signal concentrating capacity on diagnostically relevant regions. Empirically, a conditional diffusion framework achieves -17.3 dB mean normalized mean square error (NMSE) on 3rd Generation Partnership Project (3GPP) Urban Microcell (UMi) benchmarks; a verifier-guided adaptive test-time compute framework improves Llama-3.1-8B accuracy from 43.8% to 65.4% on MATH-500; and ZoomFormer achieves $F_1 = 0.899$ on HLung-6 with only 4.23M parameters, outperforming substantially larger baselines.

Chapter 1

Introduction

1.1 Problem Statement

The core research question addressed in this thesis is:

> *How can ML systems embed adaptive computation within the model to maintain robust performance under distribution shift while optimizing the accuracy-efficiency trade-off?*

At a high level, modern machine learning systems suffer from a common limitation: *computation is allocated statically*, i.e., models use fixed architectures and uniform computation patterns during both training and inference, regardless of input complexity, temporal dynamics, or intermediate reasoning quality. This leads to both inefficiency and degraded performance under a distribution shift. This problem manifests across three domains:

1) Wireless Channel Estimation: In urban microcell (UMi) scenarios, user mobility induces time-varying Doppler shifts that compress temporal coherence and cause channel statistics to evolve rapidly. Classical estimators, i.e., LMMSE (Arellano et al., 2024), GMM (**?**), and sequence models, i.e., LSTM (Ranjan and Sahana, 2025) trained on stationary assumptions, degrade as coherence times shrink. Existing diffusion-based denoisers ignore history-dependent structure and apply uniform computational budgets. The challenge is to design adaptive diffusion processes that condition on temporal context during training and allocate denoising computation based on signal quality during inference.

2) LLM Mathematical Reasoning: Test-time compute scaling improves reasoning, but existing methods apply fixed strategies, i.e., Best-of-N, beam search, or lookahead, uniformly across problems. This wastes compute on easy instances and under-explores hard ones. The challenge is to dynamically select reasoning strategies and allocate computation using intermediate verification signals, enabling adaptive compute allocation at inference time.

3) Medical Image Classification: HRCT-based lung disease classification requires both local and global feature understanding, yet CNNs lack global context, and transformers are computationally expensive. Distribution shift across patients and scanners further degrades performance. The challenge is to design models that adaptively allocate computation during training, focusing on diagnostically relevant regions through attention mechanisms while maintaining efficient inference.

1.2 Motivation

Modern machine learning systems face a fundamental challenge: distribution shift. When models trained on fixed datasets are deployed under evolving operating conditions, shifts in data distribution systematically degrade performance. Such shifts arise across diverse domains: in wireless communications through temporal channel dynamics caused by user mobility; in mathematical reasoning through variation in problem difficulty; and in medical imaging through pathological heterogeneity across patients and acquisition settings.

In safety-critical and resource-constrained applications, these shifts can lead to severe performance degradation. Traditional approaches rely on static models and fixed computation patterns, failing to adapt as conditions change. This motivates a shift toward *adaptive computation*, where models dynamically allocate computational resources based on input characteristics, temporal context, and intermediate signals.

This thesis addresses this challenge through a unified lens of *adaptive compute allocation*

within the model, spanning both training and inference. Rather than applying uniform computational effort across all inputs, we develop frameworks where computation is selectively concentrated on high-utility regions in the signal, reasoning trajectory, or feature space. This enables robustness under distribution shift while maintaining computational efficiency.

1.3 Adaptive Compute Perspective

Modern machine learning systems are increasingly required to operate in environments where both data distributions and task complexity vary significantly across inputs. A central limitation of existing approaches is that *computation is applied uniformly*, without considering whether an input is easy or hard, stationary or evolving, or well-structured or ambiguous.

This thesis proposes a unifying perspective: *adaptive computation embedded within the model as a core principle for robust and efficient learning and inference*. Instead of scaling models or inference cost uniformly, we argue that performance gains arise from dynamically allocating computation conditioned on input-dependent signals such as temporal context, signal quality, or intermediate reasoning correctness. Importantly, this adaptivity is realized across both the *training process* and the *inference procedure*. This principle is explored across three domains:

First, in wireless channel estimation, we address non-stationarity induced by user mobility through an adaptive diffusion denoising process. The model learns history-conditioned priors during training and allocates denoising computation based on measured SNR during inference. The overall framework is illustrated in Fig. 3.1, where temporal priors and self-conditioning guide reverse diffusion.

Second, in LLM reasoning, we treat reasoning as an iterative trajectory generation process and allocate computation adaptively at inference time using verifier signals. As shown in Fig. 4.1, computation is dynamically distributed across planning, tool selection, and strategy selection stages. Table 4.3 further demonstrates that no single fixed compute strategy

dominates across problem difficulties.

Third, in medical image classification, we design an adaptive attention-based training framework that enables the model to focus computation on diagnostically relevant regions. The ZoomFormer architecture (as shown in Fig. 5.1) combines multi-scale feature extraction with spatial-channel attention and transformer blocks, enabling adaptive feature refinement during training while maintaining efficient inference.

Across all three settings, the key insight is that *accuracy improvements stem not from uniformly increasing computation, but from embedding adaptive computation within the model itself.* This includes conditioning generative processes, dynamically selecting reasoning strategies, and learning to focus attention on informative features. Such adaptive compute allocation enables robustness under distribution shift while achieving favorable accuracy-efficiency trade-offs across both training and inference.

1.4 Contributions

This thesis makes three primary contributions, unified by the principle of adaptive inference under distribution shift:

1.4.1 Contribution 1: Conditional Prior-Based Non-Stationary Channel Estimation

Key innovations:

- A conditional diffusion framework that learns history-conditioned priors from causal observation windows using ConvLSTM-based temporal encoding with cross-time attention.
- SNR-matched initialization and geometrically spaced sampling schedules that reduce inference steps by an order of magnitude while preserving signal-to-noise trajectories.

- State-of-the-art performance on UMi benchmarks: -17.3 dB mean NMSE, i.e., 2.3 dB improvement operating within 2-4 dB of oracle performance at high SNR.

This work demonstrates adaptive inference for wireless channels by conditioning generative inference on temporal context and allocating denoising computation based on measured signal quality.

1.4.2 Contribution 2: Adaptive Test-Time Compute for Large Language Model (LLM) Reasoning

Key innovations:

- A verifier-guided adaptive inference framework treating reasoning as iterative trajectory generation and selection, with dynamic tool and compute strategy selection.
- Process Reward Model (PRM) providing unified control: step-level guidance for intra-iteration pruning and aggregated trajectory rewards for inter-iteration selection.
- Comprehensive ablations across fixed tool-strategy-parameter configurations, demonstrating no single fixed approach dominates.
- Substantial accuracy improvements: Llama-3.1-8B from 43.8% to 65.4% on MATH-500; Qwen-2.5-7B from 71.2% to 81.4%.
- Analysis of Compute-intensity Score to show gain from improved utilization rather than raw scaling.

This demonstrates adaptive test-time inference by dynamically selecting reasoning strategies conditional on problem difficulty and using verification to concentrate computation on high-utility reasoning paths.

1.4.3 Contribution 3: Efficient Medical Image Classification via Selective Attention

Key innovations:

- HLung-6 dataset: 793 annotated HRCT scans across 6 disease classes (COVID-19, pneumonia, TB, cystic bronchiectasis, Interstitial Lung Disease (ILD), normal) from two tertiary hospitals.
- ZoomFormer architecture: lightweight hybrid combining residual blocks, multi-scale zooming, spatial-channel-ECA attention, and transformer blocks.
- Strong performance with minimal parameters: ZoomFormer-small (i.e., 2.02M params, F_1=0.8760); ZoomFormer-large (i.e., 4.23M params, F_1=0.899).
- Comprehensive ablations validating attention mechanisms and architectural choices.

This work demonstrates efficient adaptive inference by selectively focusing model capacity on informative feature regions, achieving state-of-the-art accuracy with an order of magnitude fewer parameters than standard baselines.

1.5 Thesis Organization

This thesis is organized as follows:

Chapter 2: Background and Related Work reviews relevant literature across diffusion models for signal processing, test-time compute scaling for language models, and attention mechanisms for medical imaging. We identify gaps in existing approaches that motivate our adaptive inference frameworks.

Chapter 3: Conditional Prior-Based Non-Stationary Channel Estimation presents our conditional diffusion framework for wireless channels under mobility-induced non-stationarity. We detail the temporal encoding architecture, SNR-matched initialization, and experimental validation on UMi benchmarks.

Chapter 4: Adaptive Test-Time Compute for LLM Reasoning describes our verifier-guided adaptive inference framework for mathematical reasoning. We present the modular agent architecture, PRM-based trajectory selection, and comprehensive ablation studies on MATH-500 (Hendrycks et al., 2021), AIME24, and AMO-Bench (An et al., 2025).

Chapter 5: Efficient Medical Image Classification introduces ZoomFormer and the HLung-6 dataset. We describe the attention-transformer architecture, supervised contrastive training, and ablation studies validating design choices.

Chapter 6: Conclusions and Future Work summarizes contributions across domains, discusses limitations, and highlights future research directions for adaptive inference under distribution shift.

Chapter 2

Background and Related Work

This chapter reviews relevant literature across three application domains: diffusion models for signal processing in Section 2.1, test-time compute scaling for language models in Section 2.2, and attention mechanisms for medical imaging in Section 2.3. For each domain, we identify gaps in existing approaches that motivate our adaptive inference frameworks.

2.1 Diffusion Models for Signal Processing

Generative models are a class of machine learning models that learn to model the probability distribution of training data and can generate new samples from that distribution. Unlike discriminative models, which learn to map inputs directly to labels, generative models capture the underlying structure of the data itself. Examples include Generative Adversarial Networks (GANs) (Goodfellow et al., 2020), Variational Autoencoders (VAEs) (Kingma and Welling, 2019), and diffusion models.

Diffusion probabilistic models have emerged as powerful generative frameworks by gradually corrupting data with Gaussian noise through a forward diffusion process and learning to reverse this process via score matching (Ho et al., 2020; Song et al., 2021). Intuitively, the forward process is like slowly adding noise to a clean image until it becomes pure static; the model then learns to run this process in reverse, recovering a clean sample from noise. Given a data distribution $p_{\text{data}}(x_0)$, the forward process defines a sequence of latent variables $x_1, \ldots, x_T$ as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; x_{t-1}\sqrt{1-\beta_t}, \beta_t I), \tag{2.1}$$

where $\beta_t \in (0, 1)$ specifies the variance schedule that controls the amount of noise added at each step. The reverse process is parameterized as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{2.2}$$

where θ denotes the learnable parameters of the reverse process, with $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ representing the predicted mean and covariance respectively. Song et al. (2021) shows that score-matching objectives enable training without explicit likelihood computation by learning the score function $\nabla_{x_t} \log p_t(x_t)$. The score function is simply the gradient of the log-probability of the data at a given noise level; learning it tells the model which direction to move in to make a noisy sample more likely under the data distribution. In practice, the model predicts the noise term ϵ added during the forward process, where $\epsilon \sim \mathcal{N}(0, I)$ represents standard Gaussian noise is injected at each diffusion step. The denoising score matching objective:

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{t,x_0,\epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right], \tag{2.3}$$

trains the network ϵ_θ, where $\epsilon_\theta : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ denotes the denoising neural network parameterized by θ that estimates the noise added at step t, connecting diffusion models to denoising autoencoders and enabling stable and efficient training (Ho et al., 2020).

2.1.1 Diffusion Models for Wireless Channel Estimation

Recent work has applied diffusion models to channel estimation (Arvinte and Tamir, 2022; Fesl, Strasser, Joham and Utschick, 2024), where the goal is to recover the underlying wireless channel response from noisy pilot observations. Channel estimation is a fundamental task in wireless communications: the receiver must infer how the transmitted signal was distorted by the propagation environment before it can decode the original message. This is especially challenging in Multiple-Input Multiple-Output (MIMO) systems, which use multiple antennas at both ends of the link, and in Orthogonal Frequency Division Multi-

plexing (OFDM) systems, which split the signal across many parallel subcarriers. Arvinte and Tamir (2022) propose score-based channel estimation for massive MIMO, demonstrating that diffusion models can exploit learned priors to outperform classical estimators. However, their approach assumes stationary channel statistics and applies uniform denoising schedules regardless of signal quality. Fesl, Böck, Strasser, Baur, Joham and Utschick (2024); Fesl, Strasser, Joham and Utschick (2024) introduce accelerated sampling via SNR-based initialization for OFDM channel estimation, reducing computational cost while maintaining performance. However, they do not condition on temporal context or adapt to time-varying statistics.

Gap: Existing diffusion-based channel estimators are *unconditional*; they ignore the history-dependent structure induced by mobility and apply a fixed computational budgets regardless of signal quality.

2.1.2 Sequence Models for Time-Varying Channels

Sequence models are neural networks designed to process ordered data by maintaining an internal state that summarizes past inputs. Long Short-Term Memory (LSTM) (Graves, 2012) networks are a widely used variant that use gating mechanisms to selectively retain or discard information over time, making them well-suited for modeling temporal dependencies in signals. Recurrent neural networks and LSTMs have been applied to time-varying channel estimation (Rakhmania et al., 2025). These approaches implicitly model temporal dependencies but entangle measurement noise with channel dynamics and typically assume piecewise-stationary statistics. They degrade under rapid variation and require frequent retraining as coherence times change.

We address these gaps by developing a conditional diffusion framework as discussed in Chapter 3, which learns history-conditioned score functions from causal observation windows, explicitly separates signal evolution from measurement noise, and adapts denoising computation through SNR-matched initialization.

2.2 Test-Time Compute Scaling for Language Models

Large Language Models (LLMs) are neural networks trained on large corpora of text to predict the next token in a sequence. At inference time, they generate responses autoregressively, one token at a time. The amount of computation spent at inference, often called test-time compute, has emerged as a key factor in determining output quality, independently of model size. This section reviews three lines of work relevant to test-time scaling: inference-time reasoning strategies in Section 2.2.1, search-based methods that treat reasoning as a structured search problem in Section 2.2.2, and verification models that provide quality signals during generation in Section 2.2.3.

2.2.1 Inference-Time Computation and Reasoning

Scaling inference-time computation has emerged as a primary lever for improving LLM reasoning (Brown et al., 2024; Snell et al., 2024). This paradigm shift from pretraining compute to test-time compute reframes inference as an optimization problem: *how should additional computation be allocated to maximize solution reliability?* 1) Chain-of-Thought (CoT) prompting (Wei et al., 2022) elicits step-by-step reasoning, substantially improving performance on mathematical and symbolic tasks. However, CoT benefits are domain-dependent (Sprague et al., 2024), and generated reasoning steps may be unfaithful to the model's actual decision process (Lanham et al., 2023; Turpin et al., 2023). 2) Self-consistency and majority voting (Wan et al., 2025) sample multiple reasoning paths and select via voting, improving reliability but applying uniform sampling budgets regardless of problem difficulty. 3) Best-of-N sampling generates N independent solutions and selects via verification, but commits to fixed N and uses verification only post hoc rather than to guide generation.

2.2.2 Search-Based Reasoning

Search-based reasoning methods treat the generation of a solution as a search problem over a space of possible reasoning steps, borrowing ideas from classical AI planning and tree search algorithms. Tree-search methods adapt classical AI techniques. 1) Beam search maintains multiple candidate trajectories in parallel and prunes low-probability paths (Wiseman and Rush, 2016), but it uses a fixed beam width and does not adapt to intermediate solution quality. 2) Monte Carlo Tree Search (MCTS) (Rakhsha et al., 2025; Xie et al., 2024) guides exploration using value estimates, yet it typically relies on fixed search budgets and predefined expansion policies. 3) Lookahead search evaluates short-horizon continuations before committing to a decision, but it operates with a fixed lookahead depth and does not dynamically adjust computation.

2.2.3 Verification and PRMs

A PRM is a model trained to evaluate the quality of individual intermediate reasoning steps, rather than only the final answer. This allows it to provide fine-grained feedback during generation, enabling early detection of reasoning errors before they propagate to the final output. PRMs evaluate intermediate reasoning steps and enable trajectory-level control (Lightman et al., 2023). However, existing approaches typically embed PRMs within fixed search procedures rather than using them for dynamic strategy selection. Recent extensions incorporate LLM-based critics for error detection (McAleese et al., 2024), interactive self-improvement mechanisms (Yu et al., 2024), and trajectory-aware verification frameworks (Zou et al., 2025).

Gap: Existing methods (i) apply expensive reasoning uniformly across inputs, (ii) commit to fixed inference hyperparameters, and (iii) use verification mainly post hoc rather than as an online control signal for dynamic tool and strategy selection.

We propose a verifier-guided adaptive framework, as discussed in Chapter 4, in which a

PRM provides unified control: step-level guidance for intra-iteration pruning and aggregated rewards for inter-iteration selection. The agent dynamically selects tools (e.g., CoT, self-reflection), compute strategies (e.g., Best-of-N (BoN), beam search, or lookahead), and exploration parameters on a per-problem basis.

2.3 Attention Mechanisms for Medical Imaging

Convolutional Neural Networks (CNNs) (O'Shea and Nash, 2015) are a class of neural networks that apply learned filters over local spatial regions of an input image, making them highly effective at extracting local texture and shape features. However, because each filter operates over a fixed local window, CNNs have limited ability to capture relationships between distant regions of an image. Attention mechanisms address this by allowing a model to dynamically weight different parts of the input based on their relevance to the current prediction.

2.3.1 Deep Learning for Lung Disease Classification

CNNs have demonstrated strong performance on medical image classification (Jadhav et al., 2021), including lung disease detection from chest X-rays (Abbas et al., 2021; Goyal and Singh, 2023) and High-Resolution Computed Tomography (HRCT) scans (Kim et al., 2018, 2022). HRCT is an imaging modality that produces detailed cross-sectional images of the lungs, revealing fine structural patterns such as ground-glass opacities and consolidations that are indicative of specific diseases. Transfer learning from pretrained models (e.g., VGG, ResNet, EfficientNet) have proven effective with limited medical data (Alshmrani et al., 2023; Zak and Krzyżak, 2020).

For Interstitial Lung Disease (ILD) classification, CNN-based methods effectively analyze patterns from HRCT (Anthimopoulos et al., 2016; Huang et al., 2020). Recent work achieves fine-grained classification of ILD subtypes (e.g., Usual Interstitial Pneumonia (UIP),

Non-Specific Interstitial Pneumonia (NSIP), and Organizing Pneumonia (OP)) with high accuracy (Xu et al., 2025; Zhang, He, Wei, Tong, Yang, Wu, Guo, Shi and Jin, 2025).

Limitation: Pure CNNs struggle to capture long-range dependencies and global context crucial for identifying spatially dispersed pathological features.

2.3.2 Attention Mechanisms and Vision Transformers

Attention mechanisms further enhance feature discrimination: spatial attention emphasizes diagnostically relevant regions, while channel attention mechanisms such as Squeeze-and-Excitation (Hu et al., 2018) recalibrates channel-wise feature responses, and Efficient Channel Attention (ECA) (Wang et al., 2020) reduces parameter overhead through lightweight 1D convolutions. Channel attention works by learning a weight for each feature channel that reflects its global importance, allowing the network to suppress uninformative channels and amplify useful ones. In medical imaging, transformers help overcome the spatial locality limitations of CNNs by capturing long-range dependencies (Li et al., 2023). Vision Transformers (ViTs) (Dosovitskiy et al., 2021) apply self-attention to model global relationships and have demonstrated strong performance in computer vision (Pereira and Hussain, 2024). Self-attention allows every position in the input to directly attend to every other position, making it possible to model dependencies regardless of spatial distance. Hybrid CNN–transformer architectures combine local feature extraction with global reasoning to improve overall performance (Wang et al., 2022).

2.3.3 Computational Efficiency and Clinical Deployment

Resource constraints in clinical settings necessitate lightweight architectures. MobileNets (Howard et al., 2019) and EfficientNets (Tan and Le, 2019) employ depthwise separable convolutions and compound scaling. However, these generic architectures may not optimally leverage domain structure in medical imaging.

Gap: Existing approaches face a trade-off: pure CNNs lack global context while pure

transformers incur prohibitive costs. Generic efficient architectures do not exploit the hierarchical, heterogeneous nature of lung pathology.

We propose ZoomFormer as discussed in Chapter 5, a domain-adapted hybrid combining multi-scale residual and zooming blocks with custom spatial-channel-ECA attention and transformer modules. Supervised contrastive learning promotes tight intra-class clustering, and the architecture achieves state-of-the-art accuracy with an order of magnitude fewer parameters than standard baselines.

2.4 Summary

Across domains, existing approaches lack mechanisms for adaptive inference under distribution shift:

1. Wireless channel estimators ignore temporal context and apply uniform denoising budgets
2. LLM reasoning systems commit to fixed strategies and use verification only post hoc
3. Medical image classifiers face efficiency-accuracy trade-offs without selective attention

The following chapters present adaptive frameworks addressing these gaps through conditional compute allocation, verification-guided trajectory selection, and selective feature refinement in detail.

Chapter 3

Conditional Prior-Based Non Stationary Channel Estimation

This chapter presents our conditional diffusion framework for wireless channel estimation under mobility-induced non-stationarity (Mohsin et al., 2026).

3.1 Introduction

Accurate channel state information (CSI) is fundamental to modern wireless systems. In wideband Multiple-Input Multiple-Output (MIMO) deployments, reliable channel estimation enables spatial multiplexing gains (Liang et al., 2014). However, in urban microcell (UMi) scenarios, user acceleration compresses temporal coherence through time-varying Doppler, degrading pilot-aided interpolation and yielding stale CSI (Abdolee et al., 2013; Khalili et al., 2015).

This *non-stationarity* is characterized by time-varying tap statistics and correlation length breaks the stationary assumptions underlying classical estimators, i.e., LMMSE (Arellano et al., 2024) and deep learning methods, i.e., LSTM (Ranjan and Sahana, 2025), GMM (**?**). Recent diffusion-based denoisers (Arvinte and Tamir, 2022; Fesl, Strasser, Joham and Utschick, 2024) offer principled alternatives via score matching but are typically *unconditional*, ignoring history-dependent structure. We address these limitations through:

1. Conditional prior learning: History-conditioned score functions from causal observation

windows

2. Temporal encoding: ConvLSTM with cross-time attention capturing instantaneous coherence

3. Accelerated inference: SNR-matched initialization and geometrically spaced sampling

3.2 Problem Formulation

We consider channel estimation in a time-varying (non-stationary) wireless system. Let x_k denote the clean channel snapshot at time k, and let the noisy observation be

$$y_k = x_k + n_k, \quad n_k \sim \mathcal{N}(0, \sigma_k^2 I). \tag{3.1}$$

where σ_k^2 denotes the noise variance at time step k, which may vary as signal quality changes over time. Due to user mobility, channel statistics change over time. Therefore, estimation must exploit temporal correlation from a causal window $\mathcal{T}_k = \{y_{k-T_w+1}, \ldots, y_{k-1}\}$.

3.2.1 Forward Diffusion (Noising)

To enable principled denoising, we corrupt a clean channel sample x_0 with Gaussian noise over T steps:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{3.2}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$ is the cumulative noise schedule derived from the variance schedule $\{\beta_s\}_{s=1}^{T}$, controlling the total noise level injected up to step t. As t increases, $\bar{\alpha}_t \to 0$ and the sample becomes progressively noisier.

3.2.2 Temporal Prior and Conditioning

A temporal encoder f_{enc} compresses the causal observation window $\mathcal{T}_k = \{y_{k-T_w+1}, \ldots, y_{k-1}\}$, where T_w denotes the temporal window length, into a context vector $\mathcal{C}_k = f_{\text{enc}}(\mathcal{T}_k) \in \mathbb{R}^{d_c}$, where d_c is the context dimension. This context vector represents the channel's instantaneous coherence and defines a conditional prior $p(x_0 \mid \mathcal{C}_k)$ that adapts estimation to mobility.

3.2.3 FiLM-Based Fusion

The denoiser is conditioned on three signals: (i) diffusion step t, (ii) sequence index k, and (iii) temporal context $\mathcal{C}_k$. Their embeddings are fused via a learned fusion network f_{fuse} to generate FiLM parameters:

$$(\gamma_k, \beta_k) = f_{\text{fuse}}(t, k, \mathcal{C}_k), \tag{3.3}$$

which modulate spatial features as

$$\tilde{v} = (1 + \gamma_k) \odot \phi(x_t) + \beta_k, \tag{3.4}$$

where $\phi(x_t) \in \mathbb{R}^{d_f}$ denotes the spatial features extracted from x_t by the convolutional backbone, $\tilde{v} \in \mathbb{R}^{d_f}$ is the resulting modulated feature map, and $\odot$ denotes element-wise multiplication. This mechanism steers denoising according to both the time index and temporal channel dynamics.

3.2.4 Self-Conditioning

To enforce temporal consistency, the previous clean estimate $\hat{x}_0^{(k-1)}$ is injected into the context. This introduces an implicit Gauss-Markov prior that reduces estimation variance:

$$\text{Var}(x_0^{(k)} \mid \hat{x}_0^{(k-1)}) \leq \text{Var}(x_0^{(k)}). \tag{3.5}$$

3.2.5 Training Objective

The denoiser ϵ_θ is trained to predict the injected noise:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}\left[\|\epsilon - \epsilon_\theta(x_t, t, \mathcal{C}_k)\|_2^2\right]. \tag{3.6}$$

An optional temporal smoothness term encourages consecutive estimates to remain coherent during training:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{noise}} + \lambda \|\hat{x}_0^{(k)} - \hat{x}_0^{(k-1)}\|_2^2, \tag{3.7}$$

where $\lambda > 0$ is a regularization hyperparameter controlling the strength of the temporal smoothness penalty. During inference, the model iteratively removes noise to recover the clean channel estimate. Figure 3.1 illustrates the overall framework, including forward diffusion, temporal prior encoding, FiLM conditioning, and self-conditioned reverse denoising.

3.3 Experimental Results

3.3.1 Experimental Setup

We benchmark in a simulated UMi environment at carrier frequency $f_c = 3.5$ GHz with 80 MHz bandwidth. User speeds vary over time: $v_u(0) \in [2, 4]$ mph accelerating to $v_u(K-1) \in [30, 80]$ mph over $t_{\text{accel}} \in [1.0, 2.5]$ s. This increases Doppler spread, shortens coherence time, and accelerates channel aging.

Figure 3.2a shows the mobility profile and resulting correlation decay, validating the problem setting and highlighting the need for temporally aware estimation.

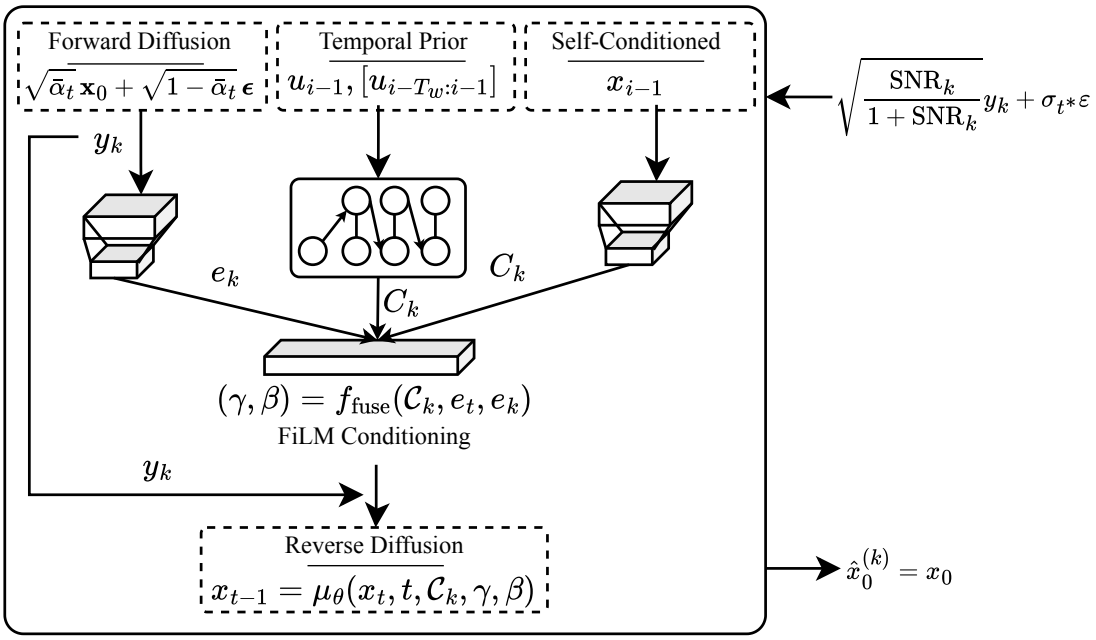
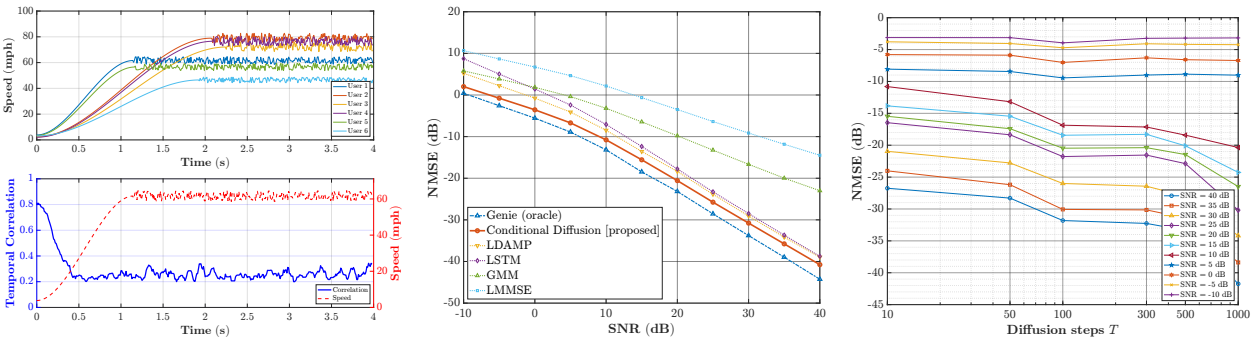


Figure 3.1: Conditional diffusion architecture combining noisy forward diffusion, ConvLSTM-attention temporal priors, and self-conditioning to guide reverse diffusion via FiLM parameters. The temporal encoder compresses historical observations into a context vector that steers denoising toward the channel's instantaneous coherence structure.



(a) User speed and correlation decay

(b) NMSE vs. SNR

(c) NMSE vs. diffusion steps

Figure 3.2: UMi benchmark results: (a) mobility profile showing acceleration from 2-4 mph to 30-80 mph and corresponding temporal correlation decay, (b) NMSE vs. SNR demonstrating consistent outperformance of baselines across all regimes, (c) convergence behavior showing SNR-dependent adaptive compute allocation.

3.3.2 Performance Comparison

The proposed framework combines temporal priors with diffusion-based denoising to adapt to non-stationary channels. As shown in Figure 3.2b, it consistently outperforms LMMSE (Arellano et al., 2024), GMM (Ali et al., 2025), LSTM (Ranjan and Sahana, 2025), and LDAMP (He et al., 2018). The method achieves a mean NMSE of -17.3 dB, representing a 2.3 dB improvement over the strongest baseline, while approaching near-oracle performance at high SNR and maintaining robustness at low SNR.

3.3.3 Computational Efficiency

Figure 3.2c reveals SNR-dependent convergence: high-SNR conditions achieve rapid convergence with few iterations, while low-SNR scenarios benefit from extended schedules. This adaptive trade-off demonstrates that the model maintains estimation quality without unnecessary overhead across diverse conditions. The SNR-matched initialization and geometric scheduling reduce inference steps by an order of magnitude compared to uniform sampling while preserving estimation accuracy.

3.4 Discussion

The conditional diffusion framework provides three main advantages. First, it enables temporal adaptation by learning history-conditioned priors from causal windows, allowing the model to track instantaneous channel coherence as user speeds vary, unlike unconditional denoisers that ignore temporal structure. Second, it improves computational efficiency through SNR-matched initialization and geometric scheduling, which allocate denoising computation based on signal quality and achieve near-oracle performance with significantly fewer sampling steps than uniform schedules. Third, it offers principled noise handling, as score matching explicitly separates signal evolution from measurement noise, avoiding the entan-

glement issues common in sequence models such as LSTMs while providing calibrated uncertainty estimates. However, the current framework assumes known noise statistics and focuses on single-user channels; future extensions should address massive MIMO systems, joint estimation-detection settings, online adaptation to unknown or time-varying noise statistics, and federated learning for distributed deployments.

3.5 Summary

This chapter introduced a conditional diffusion framework for non-stationary channel estimation. By using temporal context and SNR-matched initialization, the method adapts computation to signal quality instead of training across all SNR levels, reducing overall compute. It achieves state-of-the-art performance on UMi benchmarks with significantly lower inference cost. More generally, this shows that allocating computation based on input conditions enables efficient and robust inference under distribution shift.

Chapter 4

Adaptive Test-Time Compute for LLM Reasoning

This chapter presents our verifier-guided adaptive inference framework for mathematical reasoning (Bilal et al., 2026).

4.1 Introduction

The paradigm for improving LLM reasoning has shifted from scaling model parameters during pretraining to scaling computation at test time (Brown et al., 2024; Snell et al., 2024). However, current test-time compute scaling methods face three fundamental limitations:

1. **Uniform allocation:** Expensive reasoning procedures are applied uniformly across inputs, wasting computation on easy problems while under-allocating to hard instances
2. **Fixed strategies:** Inference procedures (i.e., Best-of-N, beam search, lookahead) commit to predetermined hyperparameters rather than adapting based on intermediate quality
3. **Post hoc verification:** Verification signals are used only for reranking completed solutions rather than guiding trajectory generation

We address these limitations with a *verifier-guided adaptive test-time inference framework* that treats reasoning as an iterative, trajectory-level decision process. For each problem, 1)

the agent may first decompose complex tasks into subgoals, 2) then dynamically select reasoning tools such as CoT, self-reflection, or verifiers, 3) choose compute strategies including Best-of-N, beam search, or lookahead along with appropriate exploration parameters, generate candidate reasoning trajectories scored by a PRM, and finally select the answer based on aggregated trajectory rewards across iterations. The whole framework is shown in Figure 4.1. The PRM provides unified control: step-level guidance for intra-iteration pruning and trajectory-level rewards for inter-iteration selection.

4.2 Methodology

4.2.1 Problem Formulation

We formalize adaptive test-time reasoning as an iterative process of trajectory generation and selection. Given problem x, the agent allocates inference computation by selecting 1) Reasoning configuration tools $\mathcal{T}$, 2) Compute strategy ($c \in \{\text{BoN, BS, LA}\}$) with exploration parameter m.

For reproducibility, we fix maximum iterations to $K = 10$ across all experiments. **Adaptivity operates within this fixed budget**: the agent dynamically selects tools, strategies, and parameters per-problem and per-iteration, using verifier signals to prioritize, prune, or terminate low-utility trajectories early. Easier problems consume less effective computation through:

- Shorter reasoning trajectories
- Fewer tool invocations
- Reduced exploration (lower Best-of-N samples, narrower beams)
- Early termination of weak trajectories

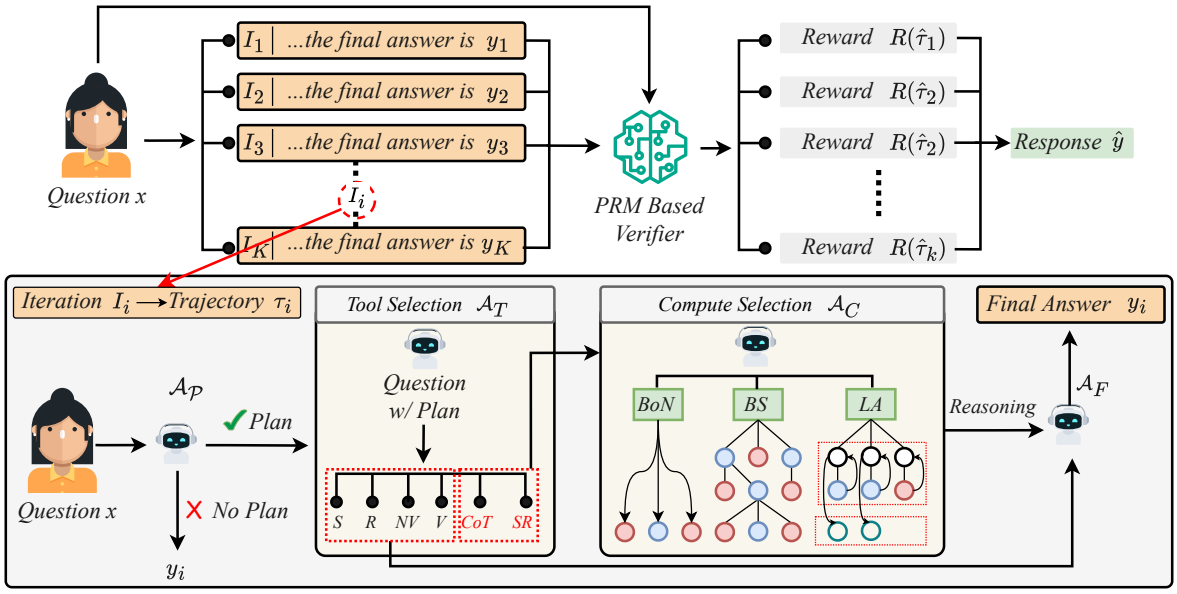


Figure 4.1: Universal reasoning agent architecture. The system generates K candidate trajectories $\{I_1, \ldots, I_K\}$ with answers $\{A_1, \ldots, A_K\}$, scored by a PRM to produce rewards $\{R_1, \ldots, R_K\}$ for selecting the best response (top). Each iteration follows a four-stage workflow (bottom): planning ($\mathcal{A}_P$), tool selection ($\mathcal{A}_T$), compute strategy selection ($\mathcal{A}_C$), and answer extraction ($\mathcal{A}_F$).

Harder problems trigger deeper reasoning and broader search within the same budget, yielding verifier-guided *compute utilization* rather than variable global budgets.

4.2.2 Agent Architecture

Figure 4.1 illustrates the modular architecture with four hierarchical components:

1. Planning Module ($\mathcal{A}_P$): Optional high-level plan $\pi = \mathcal{A}_P(x)$ decomposing complex problems into subgoals. Planning introduces overhead potentially unjustified for straightforward problems; hence we make it optional and condition use on problem characteristics.

2. Tool Selection Module ($\mathcal{A}_T$): Dynamically chooses reasoning tools $\mathcal{T} = \mathcal{A}_T(x, \pi)$ from toolkit:

$$\mathcal{T}_{\text{all}} = \{\text{CoT}, \text{SR}, \text{V}, \text{NV}, \text{R}, \text{S}\}, \tag{4.1}$$

where:

- **CoT:** Chain-of-Thought step-by-step decomposition

- **SR:** Self-Reflection with iterative critique and refinement
- **V:** General Verifier checking logical consistency
- **NV:** Numeric Verifier validating numerical correctness
- **R:** Reframer reformulating ambiguous problems
- **S:** Summarizer compressing long trajectories

By treating CoT as one tool among many rather than a default strategy, we enable adaptive reasoning strategy selection based on problem structure.

3. Compute Selection Module ($\mathcal{A}_C$): Determines inference strategy c and exploration parameter m:

$$c, m = \mathcal{A}_C(x, \pi, \mathcal{T}), \tag{4.2}$$

where m specifies:

- Best-of-N: $N = m$ samples
- Beam Search: beam width $k = m$
- Lookahead: lookahead depth $d = m$

These strategies span an exploration-exploitation spectrum with different computational trade-offs.

4. Answer Extraction Module ($\mathcal{A}_F$): Synthesizes final answer $y = \mathcal{A}_F(\tau)$ from completed trajectory $\tau = (s_1, \ldots, s_T)$ in standardized format.

This modular architecture cleanly decouples reasoning strategy selection (tools $\mathcal{T}$), exploration extent (compute strategy c with parameter m), and output formatting.

4.2.3 PRMs and Multi-Iteration Selection

We use a PRM as unified control for both *intra-iteration compute allocation* and *inter-iteration selection*. In all experiments, we employ pretrained **Qwen2.5-Math-PRM-7B**

(Zhang, Zheng, Wu, Zhang, Lin, Yu, Liu, Zhou and Lin, 2025) as a local step validator evaluating individual reasoning steps. Trajectories are generated incrementally using the selected tool configuration $\mathcal{T}$ and compute strategy c, and are segmented at explicit step boundaries. Let $\tau = (s_1, s_2, \ldots, s_{|\tau|})$ denote a reasoning trajectory, where s_t represents the t-th reasoning step (i.e., a contiguous block of tokens delimited by explicit step boundaries). For each intermediate step s_t, the PRM assigns a validity score:

$$v_t = \mathrm{PRM}(s_{t-1}, s_t) \in [0, 1], \tag{4.3}$$

reflecting whether the local transformation is mathematically valid while intentionally ignoring the global strategy. *All inference strategies use step-delimited generation.* Trajectory reward is computed as the mean step validity:

$$R(\tau) = R_{\text{mean}}(\tau) = \frac{1}{T} \sum_{t=1}^{T} v_t, \tag{4.4}$$

where $|\tau|$ denotes the number of reasoning steps in trajectory τ.

Intra-Iteration Control: PRM scores guide compute allocation *within* each iteration. During generation, trajectories are segmented at step boundaries, and PRM scores on step transitions ($s_{t-1} \rightarrow s_t$) are used to select, expand, or prune candidates *before iteration completes.*

For **Best-of-N**, sample N independent trajectories and select:

$$\hat{\tau}_i = \arg\max_{\tau \in \mathcal{S}_i} R(\tau). \tag{4.5}$$

For **Beam Search**, maintain beam set $\mathcal{B}_t$ over partial prefixes and prune using mean step validity accumulated so far:

$$R(\tau_{1:t}) = \frac{1}{t} \sum_{u=1}^{t} v_u. \tag{4.6}$$

For **Lookahead Search**, at partial prefix $\tau_{1:t}$, evaluate candidate next-step actions by rolling out short depth-d continuations using step-delimited generation. Each rollout is segmented into steps scored by the PRM, and the rollout score is the mean over step-level scores. Select the highest-scoring continuation, commit the chosen next step, and proceed from the updated prefix.

Inter-Iteration Selection: Across K independent iterations, each produces completed trajectory $\hat{\tau}_i$. Select the final output by maximizing aggregated step validity:

$$\hat{\imath} = \arg\max_{i \in \{1,\ldots,K\}} R(\hat{\tau}_i), \quad \hat{y} = y(\hat{\tau}_i). \tag{4.7}$$

4.2.4 Compute Cost Metrics

We report two hardware-agnostic metrics covering *all* model executions, including controller and verification calls.

Theoretical FLOPs aggregate raw compute across every model $j \in \mathcal{M}$ (base LLM, controllers $\mathcal{A}_P, \mathcal{A}_T, \mathcal{A}_C$, PRM):

$$F_{\text{theo}} = \sum_{j \in \mathcal{M}} 2\, M_j \cdot \min\big(\bar{T}^{(j)}_{\text{total}},\, L^{(j)}_{\text{ctx}}\big) \cdot \bar{G}^{(j)}, \tag{4.8}$$

with M_j parameters, $\bar{G}^{(j)}$ average forward passes per problem, $\bar{T}^{(j)}_{\text{total}}$ average tokens per forward pass, and $L^{(j)}_{\text{ctx}}$ maximum context length. The factor of 2 accounts for the multiply-accumulate (MAC) operations standard in FLOPs estimation, where each parameter contributes one multiplication and one addition per token. The $\min(\cdot)$ operator caps the effective token count at the context window length to avoid overestimation for long sequences. The bar notation (e.g., $\bar{G}^{(j)}$, $\bar{T}^{(j)}_{\text{total}}$) denotes averages taken over the evaluation set.

Compute Intensity S_{CI} penalizes redundant generation and auxiliary overhead to measure *utilization*:

$$S_{\text{CI}} = \frac{\bar{G}_{\text{base}}\, \bar{T}_{\text{base}}\, (1 + \alpha \bar{C})}{\kappa}, \tag{4.9}$$

where $\bar{G}_{\text{base}}$, $\bar{T}_{\text{base}}$ are base-model generations and tokens per generation, $\bar{C}$ is auxiliary calls (controller + PRM) per problem, α=0.1, and κ=10^6. Lower S_{CI} reflects more computation concentrated on high-utility trajectories. F_{theo} measures total compute; S_{CI} measures how efficiently it is spent.

4.3 Experimental Setup

4.3.1 Datasets and Models

We evaluate our framework on three mathematical reasoning benchmarks, i.e., MATH-500 (Hendrycks et al., 2021), a 500-problem competition subset stratified across five difficulty levels and seven domains; AMO-Bench (An et al., 2025), a curated collection of advanced Olympiad-level mathematics problems designed to evaluate deep, multi-step symbolic reasoning and generalization; and AIME24, consisting of 30 problems from the 2024 American Invitational Mathematics Examination that require long-horizon multi-step reasoning. Experiments are conducted using **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024) and **Qwen-2.5-7B-Instruct** (Qwen Team, 2024). For each problem, we run $K = 10$ independent agent iterations under either a fixed configuration or a dynamic controller implemented using the same base LLM, prompted with role-specific templates to select reasoning tools ($\mathcal{A}_T$) and compute strategies ($\mathcal{A}_C$). All generated trajectories are scored using *Qwen2.5-Math-PRM-7B*, and the final prediction is chosen by maximizing the mean PRM reward R_{mean} across iterations. The base reasoning model uses stochastic decoding (temperature $T = 0.7$, top-$p = 0.9$, maximum length 1024 tokens), while the controller modules and PRM use deterministic greedy decoding ($T = 0$), with all settings held constant across experiments.

4.3.2 Experimental Configurations

We evaluate four configurations progressively introducing adaptivity:

1. **Direct:** Single-pass generation without tool selection, compute scaling, or PRM guidance (lower-bound baseline)
2. **Direct + PRM Selection:** $K = 10$ independent direct generations with PRM-based iteration selection (i.e., isolates the effect of multi-iteration evaluation)
3. **Dynamic:** LLM controller selects tools and compute strategies within a single iteration (i.e., isolates the benefit of adaptive configuration)
4. **Dynamic + PRM Selection:** Combines adaptive tool/compute selection with $K = 10$ iterations and PRM-based selection (full framework)

Additionally, we perform fixed-configuration ablations applying specific tool-compute-parameter combinations uniformly across problems with varying iteration counts, enabling controlled comparisons isolating the contribution of adaptive selection.

4.4 Results

4.4.1 Adaptive Configuration Over Fixed Strategies

Table 4.1 compares configurations progressively introducing adaptivity across all three benchmarks, showing verifier-guided adaptive inference consistently outperforms uniform compute allocation.

Multi-iteration sampling alone is insufficient: Repeating direct inference with PRM selection yields only marginal gains ($\sim$+1 point on MATH-500, no gain on AIME24 or AMO-Bench), indicating that when the underlying reasoning strategy remains fixed, repetition provides limited benefit.

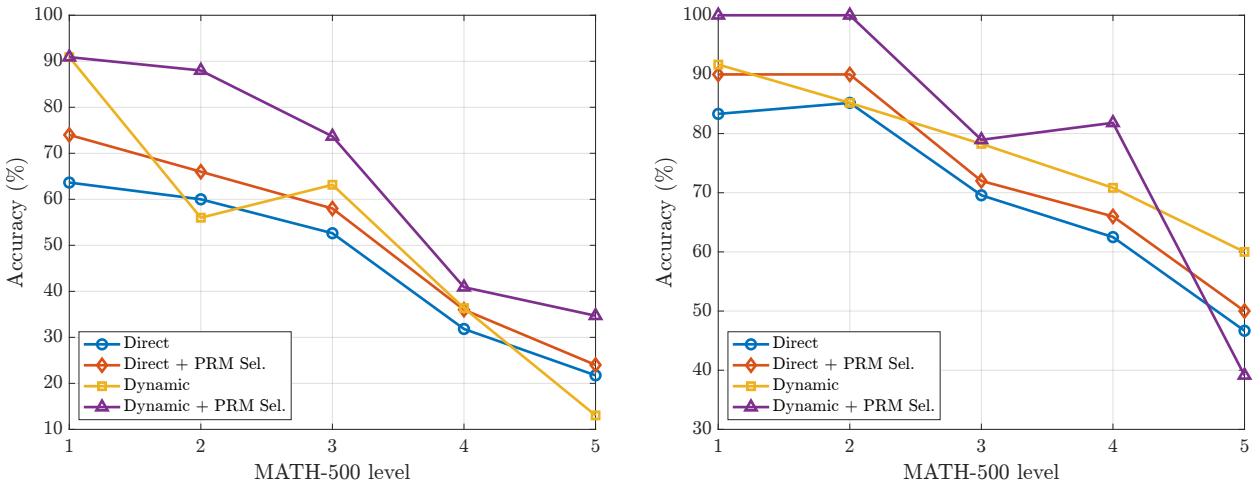
Table 4.1: Main accuracy results on MATH-500, AIME24, and AMO-Bench. Dynamic + PRM Selection represents the full adaptive framework.

Dataset	**Setting**	**Accuracy (%)**	
		Llama-3.1-8B	**Qwen-2.5-7B**
MATH-500	Direct	43.8	71.2
	+ PRM Sel.	44.6	72.0
	Dynamic	47.2	74.2
	+ PRM Sel.	**65.4**	**81.4**
AIME24	Direct	3.3	6.67
	+ PRM Sel.	6.67	6.67
	Dynamic	6.67	10.0
	+ PRM Sel.	**10.0**	**13.3**
AMO-Bench	Direct	2.0	2.0
	+ PRM Sel.	2.0	2.0
	Dynamic	**4.0**	**4.0**
	+ PRM Sel.	**4.0**	**4.0**

Problem-specific configuration matters: Enabling dynamic tool and compute selection within a single iteration improves performance ($\sim$+3 points on MATH-500, +3.3 on AIME24 with Qwen, and doubles AMO-Bench accuracy from 2.0% to 4.0%), showing adaptive allocation drives gains rather than mere repetition.

Full framework performs best: Dynamic + PRM Selection achieves largest improvements ($>$+20 points for Llama on MATH-500, $\sim$+10 points for Qwen, $>$+3$\times$ gains on AIME24) by combining adaptive allocation *within* trajectories with PRM-guided selection *across* trajectories. On AMO-Bench, the hardest benchmark, adaptive configuration alone, Dynamic already yields the maximum improvement, with PRM selection providing no additional gain consistent with the difficulty of obtaining reliable verifier signals on Olympiad-level problems.

This matches our problem statement: effective test-time scaling requires deciding *when, where, and how* to spend compute during reasoning and using verification to prioritize high-utility trajectories rather than uniformly scaling generation.



(a) Llama-3.1-8B-Instruct (b) Qwen-2.5-7B-Instruct

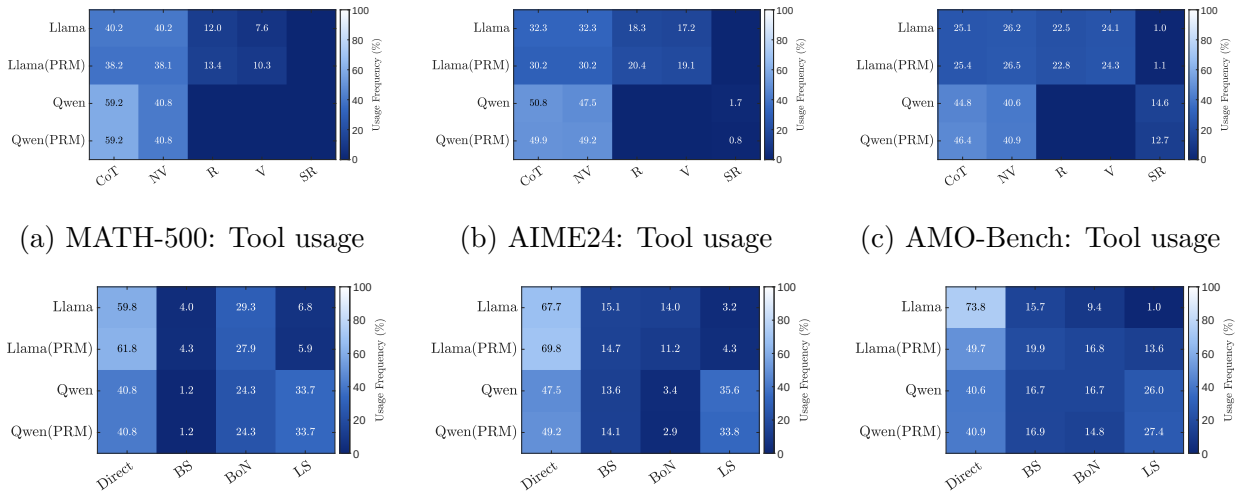
Figure 4.2: Per-level accuracy on MATH-500 across difficulty levels. Adaptive inference yields large gains on Levels 1-4 where verification signals are reliable. On Level 5, PRM-based selection can degrade performance when confident-but-wrong trajectories are scored higher than partially correct ones.

4.4.2 Performance by Difficulty Level

Figure 4.2 reports MATH-500 accuracy by difficulty level 1-5.

Llama-3.1-8B-Instruct (as shown in Figure 4.2a): Adaptive inference yields large gains on Levels 1-4 where verification signals are reliable and early pruning suppresses low-utility paths. On Level 5, single-iteration Dynamic degrades, but multi-iteration PRM selection partially recovers accuracy, indicating repeated adaptive attempts increase the chance of discovering the correct high-level strategy.

Qwen-2.5-7B-Instruct (as shown in Figure 4.2b): Dynamic + PRM Selection achieves near-saturated performance on Levels 1-2 and consistent improvements on Levels 3-4. However, performance drops on Level 5 when PRM-based selection is applied across iterations. This drop occurs because the PRM can select the wrong solutions on Level 5. These problems are very long, mistakes show up late, so answers can look good step-by-step, but still be wrong. The PRM may score confident-but-wrong trajectories higher than partially correct ones, so selection across iterations sometimes chooses the wrong final output.



(a) MATH-500: Tool usage (b) AIME24: Tool usage (c) AMO-Bench: Tool usage

(d) MATH-500: Compute strategy (e) AIME24: Compute strategy (f) AMO-Bench: Compute strategy

Figure 4.3: Dynamic agent configuration statistics showing usage frequency (%) across all three benchmarks. Tool usage shows clear model-dependent behavior: Llama spreads decisions across multiple tools while Qwen concentrates on structured reasoning plus numeric checking. Compute strategy usage is complementary: Llama favors direct generation while Qwen allocates more to exploration strategies. Dataset-specific patterns reflect differences in problem structure and reasoning difficulty.

4.4.3 Tool and Compute Strategy Selection Patterns

Figure 4.3 illustrates how the agent adapts reasoning tools and compute strategies under dynamic configuration across all three benchmarks.

Tool Usage: Clear model-dependent behavior emerges across all benchmarks. Llama spreads decisions across multiple tools (CoT and verifiers each ∼10-15%), whereas Qwen concentrates on structured reasoning plus numeric checking (CoT ∼40-50%, Numeric Verifier ∼30-40%). On AMO-Bench, tool distributions remain similar to MATH-500 and AIME24, suggesting selections are primarily driven by model-specific reasoning characteristics rather than dataset-level difficulty. Patterns remain stable from 1 to 10 iterations across all benchmarks.

Compute Strategy Usage: Complementary patterns appear across datasets. Llama favors direct generation for the majority of instances (∼60%), while Qwen allocates more

Table 4.2: Ablation study: Accuracy (%) on MATH-500 with Qwen-2.5-7B-Instruct across fixed tool-strategy-parameter-iteration configurations.

Param	Iter	CoT			Self-Reflection		
		Best-of-N	Beam Search	Lookahead	Best-of-N	Beam Search	Lookahead
1	1	71.2	70.4	70.2	75.6	75.2	71.8
	5	81.4	81.4	81.4	80.2	80.6	79.4
	10	81.2	81.2	81.2	80.8	80.4	79.2
5	1	75.8	76.2	75.2	74.2	76.8	75.6
	5	81.4	79.4	80.8	79.4	79.2	79.8
	10	81.4	79.8	80.6	79.8	79.6	79.4
10	1	72.2	73.6	72.4	72.6	74.4	74.2
	5	81.4	80.2	80.2	79.2	79.8	79.6
	10	81.4	80.6	80.8	79.6	79.4	79.8

mass to exploration strategies like Best-of-N and Lookahead (each often ∼15-35%). On AIME24 and AMO-Bench, both models shift slightly toward broader exploration strategies relative to MATH-500, consistent with increased problem difficulty requiring more diverse trajectory search. Overall, dynamic configuration adapts to both problem characteristics and model-specific strengths rather than enforcing a single fixed inference recipe, and this adaptation is consistent across benchmarks of varying difficulty.

4.4.4 Comprehensive Ablation: Fixed Configurations

Table 4.2 presents comprehensive ablation results on MATH-500 with Qwen-2.5-7B-Instruct across all combinations of tools, compute strategies, exploration parameters ($p \in \{1, 5, 10\}$), and iteration counts ($I \in \{1, 5, 10\}$). This 36-configuration grid isolates the contribution of each component in fixed settings. The key findings from this comprehensive analysis are below:

1. **Increasing iterations improves accuracy:** For most configurations, accuracy increases substantially from $I = 1$ to $I = 5$ (e.g., CoT Best-of-N with $p = 1$: 71.2% → 81.4%), with diminishing returns at $I = 10$ (81.2%)

Table 4.3: Ablation study: Accuracy (%) on AIME24 with Qwen-2.5-7B-Instruct across fixed configurations.

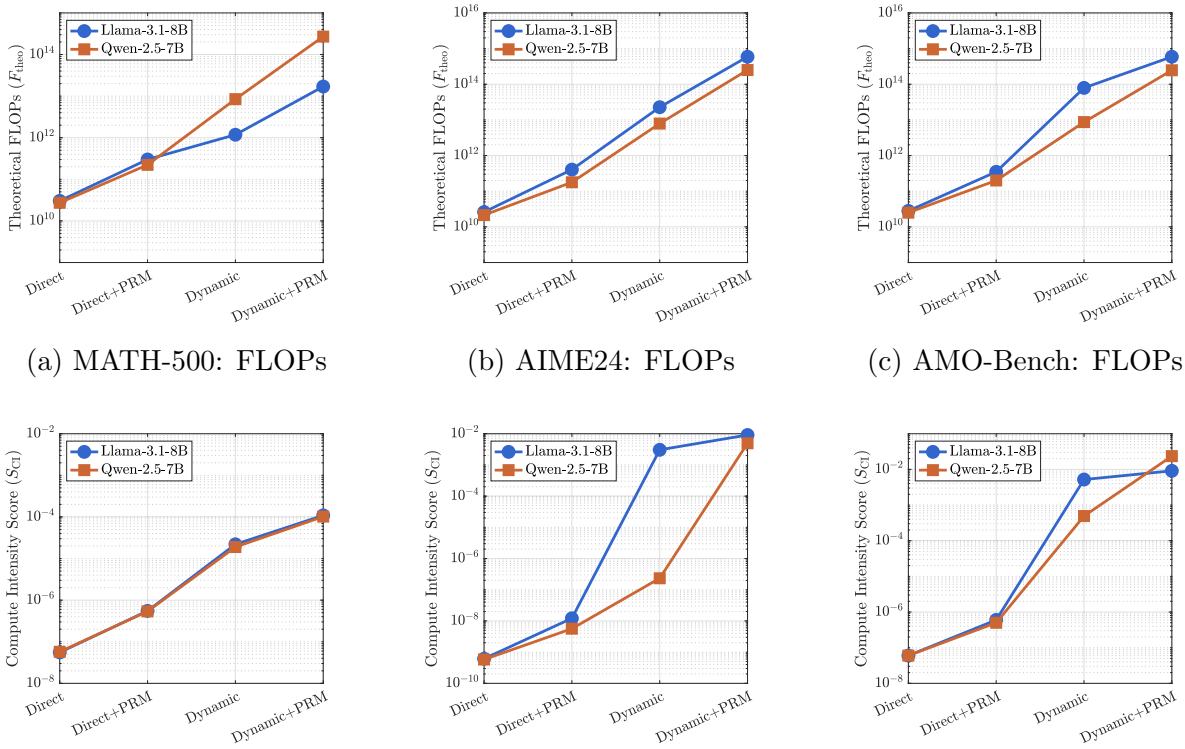
Param	Iter	CoT			Self-Reflection		
		Best-of-N	Beam Search	Lookahead	Best-of-N	Beam Search	Lookahead
1	1	3.3	3.3	6.67	3.3	3.3	3.3
	5	3.3	3.3	6.67	3.3	3.3	6.67
	10	6.67	6.67	6.67	3.3	6.67	6.67
5	1	6.67	6.67	10.0	3.3	3.3	6.67
	5	6.67	6.67	10.0	3.3	6.67	6.67
	10	10.0	6.67	10.0	6.67	6.67	6.67
10	1	6.67	6.67	3.3	3.3	6.67	3.3
	5	6.67	6.67	3.3	6.67	6.67	3.3
	10	6.67	6.67	3.3	6.67	6.67	6.67

2. **Tool choice matters:** Self-Reflection (75.6% at $p = 1$, $I = 1$) outperforms CoT (71.2%) initially, but both converge at higher iterations ($\sim$80-81% at $I = 5, 10$)

3. **Strategy-parameter interactions are complex:** Best-of-N with $p = 5$ (75.8%) outperforms $p = 10$ (72.2%) at a single iteration, suggesting excessive exploration introduces noise without multi-iteration refinement

4. **No single configuration dominates:** Peak fixed-configuration accuracy (81.4%, multiple configurations) matches our Dynamic + PRM Selection result (81.4%), but crucially, fixed approaches require *a priori* knowledge of optimal settings whereas our framework adapts automatically.

Table 4.3 reports corresponding ablations on AIME24. Unlike MATH-500, no fixed configuration exceeds 10.0% accuracy (best: CoT Lookahead, p=5, $I \in \{1, 5, 10\}$), compared to 13.3% for Dynamic + PRM Selection. Increasing p degrades performance (e.g., CoT Lookahead: 6.67% $\rightarrow$ 10.0% $\rightarrow$ 3.3% as p increases), and iteration gains are marginal, both indicating that trajectory diversity from adaptive multi-tool selection, rather than uniform repetition, is essential for highly challenging problems.



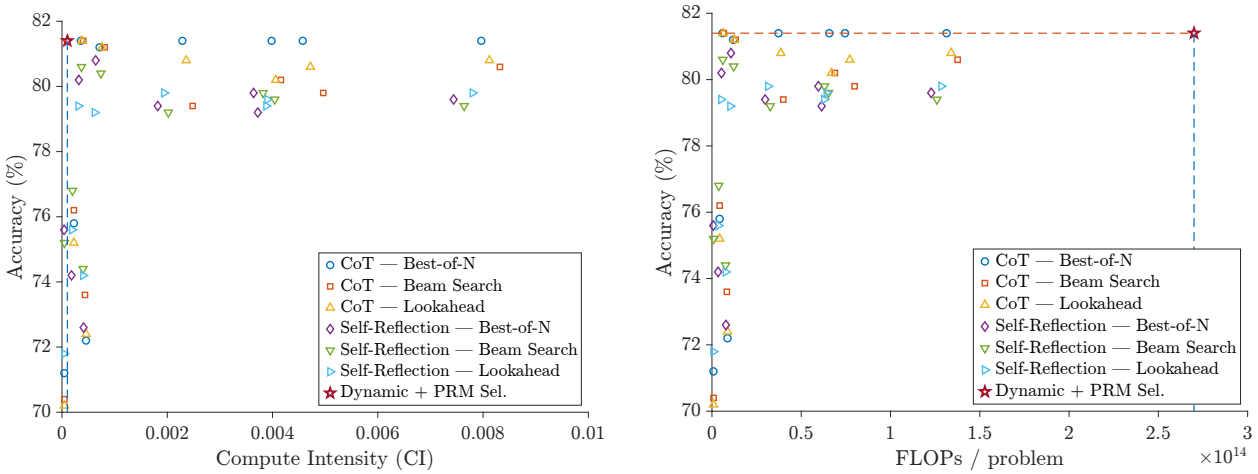
(a) MATH-500: FLOPs (b) AIME24: FLOPs (c) AMO-Bench: FLOPs

(d) MATH-500: Compute Intensity (e) AIME24: Compute Intensity (f) AMO-Bench: Compute Intensity

Figure 4.4: Compute cost scaling across all three benchmarks. Theoretical FLOPs and compute intensity (S_{CI}) are shown for main experimental configurations on MATH-500, AIME24, and AMO-Bench. As adaptivity increases from Direct to Dynamic + PRM Selection, both FLOPs and S_{CI} increase monotonically, reflecting additional forward passes for controller decisions, verification, and multi-iteration exploration.

4.4.5 Compute Cost Scaling and Efficiency

Figure 4.4 shows how inference-time compute scales across increasingly adaptive configurations measured in theoretical FLOPs and S_{CI} (compute intensity) across all three benchmarks. As adaptivity increases from Direct to Dynamic + PRM Selection, both FLOPs and S_{CI} increase monotonically across all three benchmarks. This reflects additional forward passes for controller decisions, verification, and multi-iteration exploration. The pattern is consistent across MATH-500, AIME24, and AMO-Bench, with harder benchmarks exhibiting higher absolute compute due to longer reasoning trajectories and more verification steps.



(a) Accuracy vs. Compute Intensity

(b) Accuracy vs. FLOPs

Figure 4.5: Accuracy-efficiency trade-offs on MATH-500 (Qwen-2.5-7B). Fixed strategies reach ∼81% accuracy only by sharply increasing S_{CI} through uniform search scaling, whereas Dynamic + PRM achieves the same accuracy with lower S_{CI}, indicating additional compute is preferentially allocated to high-utility reasoning trajectories.

Key insight:

1) This figure characterizes *raw compute scaling* rather than efficiency at matched accuracy. Lower FLOPs/lower S_{CI} for Direct inference indicate lower cost but also substantially lower accuracy.

2) The efficiency claim is not that Dynamic + PRM minimizes S_{CI} in absolute terms, but that it achieves *higher accuracy for given compute level*, and similar accuracy to strong fixed strategies with better utilization.

3) Figure 4.5 compares Dynamic + PRM Selection against all fixed-configuration baselines at matched accuracy levels on MATH-500.

4) Fixed strategies reach ∼81% accuracy only by sharply increasing S_{CI} through uniform search scaling, whereas Dynamic + PRM achieves the same accuracy with lower S_{CI}, indicating additional compute is preferentially allocated to high-utility reasoning trajectories rather than wasted on redundant generations.

4.5 Discussion

4.5.1 Key Takeaways

Our results support three main conclusions:

1. Adaptive inference outperforms uniform test-time scaling: Dynamic tool and compute choices plus PRM-guided selection consistently outperform fixed strategies across all three benchmarks. Multi-iteration sampling alone provides minimal benefit (i.e., Direct + PRM Selection: $\sim$+1 point on MATH-500, no gain on AIME24 or AMO-Bench), but combining adaptive configuration with PRM selection yields large gains (Dynamic + PRM Selection: $>$+20 points for Llama, $\sim$+10 points for Qwen on MATH-500; $>3\times$ improvements on AIME24; doubled accuracy on AMO-Bench).

2. Gains are difficulty-dependent: The method achieves strong improvements on Levels 1–4 of MATH-500, where verification signals are reliable and early pruning effectively removes low-utility reasoning paths. Performance weakens on Level 5 and on AMO-Bench, where the Dynamic configuration already saturates at 4.0%, and additional PRM-based selection does not improve results. On the hardest problems, the PRM sometimes assigns higher scores to confident but incorrect trajectories than to partially correct ones, leading the system to select the wrong final answer. On AMO-Bench, adaptive configuration still improves performance over the direct baseline, but multi-iteration PRM selection provides no further gains, likely due to unreliable verifier signals at Olympiad-level difficulty.

3. Adaptivity improves compute utilization: Verification signals concentrate inference on high-utility trajectories rather than uniformly increasing search. Achieving 81.4% accuracy via fixed configurations requires 1.31×10^{14} FLOPs with compute intensity 7.97×10^{-3}, indicating substantial wasted computation on problems not requiring extensive search. Dynamic + PRM achieves identical accuracy with a more favorable efficiency profile through adaptive allocation: slightly higher F_{theo} but significantly lower S_{CI}.

4.5.2 Limitations

1. **Reliance on verifier quality:** On the hardest problems (Level 5 of MATH-500, AIME24, and especially AMO-Bench), current PRMs can misrank plausible-but-wrong trajectories, which reduces the benefit of inter-iteration selection. The AMO-Bench results, where PRM selection provides no gain over single-iteration Dynamic, clearly illustrate this limitation. Future work should develop difficulty-aware PRMs, for example, through hard-subset supervision and curriculum learning across difficulty levels.

2. **Heuristic controller:** The controller is prompt-based rather than learned, motivating future work on learned, budget-aware policies that optimize tool selection, compute allocation, and stopping decisions under explicit constraints (Liu et al., 2025; Paglieri et al., 2025).

3. **Systems overhead:** Multi-branch search and verification introduce additional latency and systems overhead, suggesting the need for co-design with batching strategies and latency-aware scheduling mechanisms.

4. **Domain generalization:** Although results are strong for mathematical reasoning across benchmarks of varying difficulty, generalization to other domains (e.g., code generation, long-context question answering, safety-critical tasks) remains an open problem and will likely require domain-adaptive verification signals and broader evaluation.

4.6 Summary

This chapter presented a verifier-guided adaptive test-time inference framework for mathematical reasoning. By dynamically selecting reasoning tools and compute strategies on a per-problem basis and using a PRM to provide step-level control for trajectory pruning and multi-iteration selection, the method achieves substantial accuracy improvements across all three benchmarks: Llama-3.1-8B from 43.8% to 65.4% on MATH-500, from 3.3% to 10.0%

on AIME24, and from 2.0% to 4.0% on AMO-Bench; Qwen-2.5-7B from 71.2% to 81.4% on MATH-500, from 6.67% to 13.3% on AIME24, and from 2.0% to 4.0% on AMO-Bench.

Comprehensive ablations demonstrate no single fixed configuration dominates, and compute intensity analysis shows gains arise from improved utilization rather than raw scaling. Notably, on AMO-Bench (i.e., the most challenging benchmark), adaptive configuration yields the full accuracy gain while PRM-based selection provides no additional benefit, highlighting that verifier reliability is a key constraint at Olympiad-level difficulty. This exemplifies the thesis principle: *adaptive allocation of test-time computation conditional on problem difficulty and intermediate quality signals enables robust, efficient reasoning under distribution shift.*

Chapter 5

Efficient Medical Image Classification

This chapter presents ZoomFormer, a lightweight attention-transformer architecture for lung disease classification.

5.1 Introduction

Lung diseases impose a substantial burden on global health, and they are among the leading causes of mortality worldwide. Timely and accurate diagnosis plays a crucial role in improving survival rates and enhancing patients' quality of life. High-Resolution Computed Tomography (HRCT) provides detailed visualization of the lung parenchyma and enables early identification of disease-specific patterns, making it an essential imaging modality for pulmonary assessment. However, radiologists often find manual interpretation of HRCT scans time-consuming and susceptible to inter-observer variability, as subtle texture differences and complex patterns can be difficult to distinguish—especially in resource-constrained settings with limited access to expert interpretation. Moreover, the growing volume of imaging data generated in modern hospitals further increases the need for automated systems that support clinicians in making accurate and fast diagnostic decisions.

Deep neural networks, especially CNNs, have achieved remarkable performance in medical image classification tasks, including lung disease detection. However, conventional CNNs struggle to capture long-range dependencies and global context that are important for identifying subtle pathological features that are distributed across different lung regions. These limitations become particularly evident in HRCT images, where minimal structural changes

such as ground-glass opacities and consolidations carry significant diagnostic value but often appear spatially dispersed across the lungs.

Recent advances introduced transformer architectures from natural language processing (Vaswani et al., 2017) into computer vision as ViTs (Dosovitskiy et al., 2021). ViT and its various variants show impressive capabilities in modeling global relationships by employing self-attention mechanisms (Pereira and Hussain, 2024). These architectures can potentially overcome spatial locality limitations of CNNs when applied to medical imaging (Li et al., 2023).

We propose *ZoomFormer*, a deep learning framework for the classification of lung diseases using HRCT images that employs transformer models and enhanced attention mechanisms. Our contributions are:

1. **HLung-6 dataset:** 793 annotated HRCT scans containing 6 disease classes (i.e., COVID-19, pneumonia, TB, cystic bronchiectasis, ILD, normal) from two tertiary hospitals

2. **ZoomFormer architecture:** Lightweight hybrid architecture that combines residual blocks, multi-scale zooming, spatial-channel-ECA attention, and transformer blocks

3. **Strong performance with minimal parameters:** ZoomFormer-small (i.e., 2.02M parameters having score F_1=0.8760); ZoomFormer-large (i.e., 4.23M parameters having score F_1=0.899)

5.2 Related Work

CNNs have significantly improved lung disease detection from chest X-rays (Abbas et al., 2021; Goyal and Singh, 2023) and CT scans (Kim et al., 2018, 2022). Researchers have successfully applied transfer learning using pretrained models such as VGG, ResNet, and EfficientNet, especially when medical datasets are limited (Alshmrani et al., 2023; Zak and Krzyżak, 2020). For ILD classification, CNN-based approaches effectively analyze HRCT

patterns (Anthimopoulos et al., 2016; Huang et al., 2020), and recent studies achieve high accuracy in fine-grained ILD subtype classification (Xu et al., 2025; Zhang, He, Wei, Tong, Yang, Wu, Guo, Shi and Jin, 2025). Attention mechanisms further improve performance by highlighting diagnostically important regions through spatial attention and by recalibrating channel-wise feature responses using channel attention mechanisms such as Squeeze-and-Excitation (Hu et al., 2018). Hybrid CNN-transformer architectures also enhance performance by combining local feature extraction with global contextual reasoning (Wang et al., 2022).

Resource constraints require the design of lightweight architectures, i.e., MobileNets (Howard et al., 2019) and EfficientNets (Tan and Le, 2019) use depthwise separable convolutions and compound scaling to improve efficiency. However, these generic efficient architectures do not always fully exploit the domain-specific structure present in medical imaging data.

Gap: Existing approaches face trade-offs: pure CNNs lack global context while pure transformers incur prohibitive costs. Generic efficient architectures do not exploit the hierarchical, heterogeneous nature of lung pathology.

Our approach: Domain-adapted hybrid combining multi-scale residual and zooming blocks with custom spatial-channel-ECA attention and transformer modules, achieving state-of-the-art accuracy with an order of magnitude fewer parameters.

5.3 ZoomFormer Architecture

5.3.1 Overall Design

Figure 5.1 illustrates the complete ZoomFormer architecture. The model processes input images through residual, zooming, proposed attention, and feature-extraction blocks before passing representations to the transformer block. Final embeddings are optimized using supervised contrastive loss.

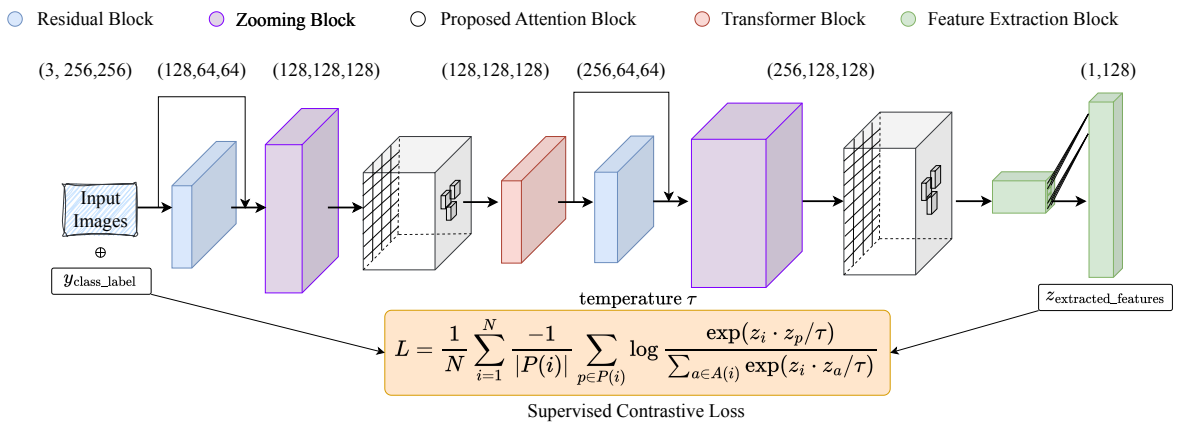


Figure 5.1: Overall ZoomFormer architecture illustrating the sequence of modules for dynamic feature refinement. The model processes input images through residual, zooming, proposed attention, and feature-extraction blocks before passing representations to the transformer block. Final embeddings are optimized using supervised contrastive loss.

5.3.2 Residual Blocks

Residual blocks stabilize training and enable learning features with greater complexity. Each residual block comprises convolutional layers with batch normalization and ReLU activation, followed by a skip connection adding input to output:

$$C_n(x, y) = \sum_{i,j} C_{n-1}(i, j)W_n(x - i, y - j), \tag{5.1}$$

where $C_n(x, y)$ denotes the feature map at spatial location (x, y) of layer n, $W_n(x, y)$ is the learned convolutional filter at layer n, and C_0 is the input image to the first residual block.

Skip connections preserve and reuse features from earlier layers, leading to more effective extraction and better convergence. Max-pooling reduces spatial dimensions:

$$Y_k(x, y) = \max_{(p,q)\in R(i,j)} C_n(p, q), \tag{5.2}$$

where $Y_k(x, y)$ denotes the pooled feature map output at spatial location (x, y), and $R(x, y)$ denotes the rectangular pooling receptive field centered at (x, y).

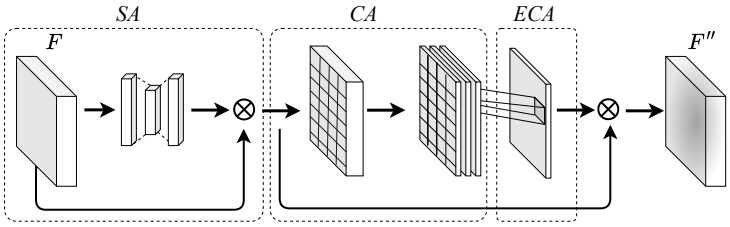


Figure 5.2: Proposed attention mechanism integrating Spatial Attention (SA) to emphasize spatially significant features, Channel Attention (CA) to enhance informative channels, and Efficient Channel Attention (ECA) for lightweight channel dependency modeling.

5.3.3 Zooming Blocks

Zooming blocks enable capturing features from multiple spatial scales simultaneously. Different kernel sizes allow learning from fine-grained details to broader contextual patterns, while different stride values result in hierarchical representation. This lets the model adaptively "zoom" into different regions of interest while maintaining computational efficiency, particularly crucial for medical image analysis, as features of interest may appear at different scales and locations.

5.3.4 Proposed Attention Block

The custom-designed attention block, as shown in Figure 5.2, integrates multiple attention mechanisms: Spatial Attention (SA), Channel Attention (CA), and Efficient Channel Attention (ECA).

Spatial Attention: Computes spatial attention map:

$$A_{sp} = \sigma\left(\sum_{c=1}^{C} W_{sp}^{c} \cdot Y + b_{sp}\right). \tag{5.3}$$

Spatial attention is applied to the feature map as:

$$X_{sp} = X \odot A_{sp}. \tag{5.4}$$

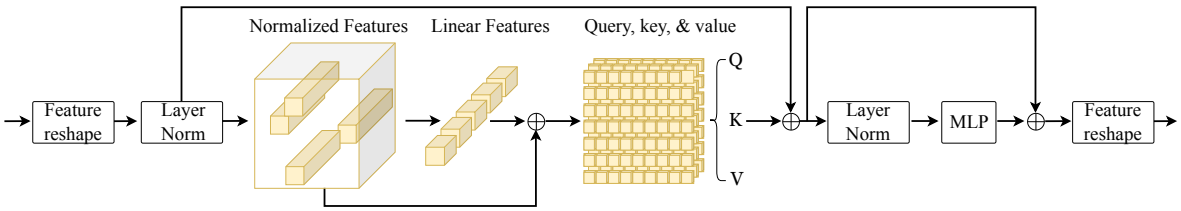


Figure 5.3: Internal transformer block used in ZoomFormer. The block applies layer normalization, linear projections, and multi-head self-attention using learned query, key, and value representations, followed by feed-forward refinement and reshaping operations to enhance informative feature regions while maintaining global context.

Channel Attention: Starts with global average pooling:

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} C_{i,j,c}, \tag{5.5}$$

where $Z_c \in \mathbb{R}$ is the global average pooled descriptor for channel c, and $X_{i,j,c}$ denotes the feature map value at spatial position (i, j) and channel c. Channel attention weights are computed as:

$$A_{ch} = \sigma(W_{ch} \cdot \mathrm{ReLU}(W_{ch1} \cdot Z + b_{ch1}) + b_{ch}), \tag{5.6}$$

where $\mathbf{Z} = [Z_1, \ldots, Z_{N_c}]^\top \in \mathbb{R}^{N_c}$ is the stacked channel descriptor vector, $W_{ch1} \in \mathbb{R}^{r \times N_c}$ and $W_{ch} \in \mathbb{R}^{N_c \times r}$ are learned weight matrices with reduction ratio r, and $b_{ch1} \in \mathbb{R}^r$, $b_{ch} \in \mathbb{R}^{N_c}$ are bias vectors.

Efficient Channel Attention: Dynamically captures channel dependencies using lightweight 1D convolutions.

By combining these mechanisms, the block assigns higher weights to critical areas, enabling prioritization of features indicative of key characteristics associated with lung disease.

5.3.5 Transformer Block

As shown in Figure 5.3, the transformer block utilizes self-attention mechanisms to capture long-range dependencies within the input CT image. Features undergo layer normalization:

$$X_{\text{norm}} = \text{LayerNorm}(X). \tag{5.7}$$

Query, key, and value representations are computed as:

$$Q = W_Q X_{\text{norm}}, \quad K = W_K X_{\text{norm}}, \quad V = W_V X_{\text{norm}}, \tag{5.8}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_k \times d_{\text{in}}}$ are learned linear projection matrices mapping the normalized input to query, key, and value representations respectively, and d_{in} is the input feature dimension. Self-attention is applied through:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right), \tag{5.9}$$

where d_k denotes the key/query projection dimension used for scaling.

Output is refined as:

$$X_{\text{att}} = AV. \tag{5.10}$$

Further transformed through MLP:

$$X_{\text{out}} = \text{MLP}(\text{LayerNorm}(X_{\text{att}})). \tag{5.11}$$

This captures both local and global contextual information, improving generalization capability for lung disease recognition.

Table 5.1: Technical specifications of HRCT scanners used in the study.

Specification	**Aquiline 64**	**Aquiline 16**
Slice count	64	16
Gantry aperture (cm)	72	70
Detector coverage (mm)	40	40
Rotation time (s)	0.35	0.50
Spatial resolution (mm)	0.35	0.35
Max scan range (cm)	160	160

5.3.6 Supervised Contrastive Loss

The supervised contrastive loss optimizes feature representation by pulling together samples from the same class and pushing apart samples from different classes:

$$\mathcal{L}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\text{sim}(z_i, z_p)/\tau)}{\sum_{a \in A(i)} \exp(\text{sim}(z_i, z_a)/\tau)}, \tag{5.12}$$

where z_i denotes the feature embedding of i-th anchor sample, z_p is the embedding of the positive sample sharing the same class label, $P(i)$ contains all positive samples for anchor i, $A(i)$ denotes the set of all other samples in the batch excluding i, $\text{sim}(z_i, z_j)$ represents cosine similarity, and $\tau = 0.01$ is the temperature parameter. This loss encourages features from images of the same class (e.g., healthy lungs) to cluster together while those from different classes (e.g., diseased lungs) are pushed apart.

5.4 HLung-6 Dataset

5.4.1 Data Acquisition

We acquired images using multi-slice CT scanners at two hospitals: Hospital H_1 operated a 64-slice model, while Hospital H_2 used a 16-slice system. Table 5.1 summarizes the key hardware parameters. Technologists captured all scans at 120 kVp with automatic tube current modulation and reconstructed them into 0.5 mm slice thickness with 0.35 mm in-

Table 5.2: Class distribution of HRCT scans collected at Hospital H_1 with 64-slice scanner.

Disease	**Male**	**Female**	**Total**
COVID-19	89	61	150
Pneumonia	62	48	110
Tuberculosis (TB)	67	38	105
Cystic bronchiectasis	77	21	98
ILDs	35	23	58
Normal	0	0	0
Subtotal	330	191	521

Table 5.3: Class distribution of HRCT scans collected at Hospital H_2 with 16-slice scanner.

Disease	**Male**	**Female**	**Total**
COVID-19	0	0	0
Pneumonia	2	2	4
Tuberculosis (TB)	15	13	28
Cystic bronchiectasis	24	12	36
ILDs	48	60	108
Normal	50	46	96
Subtotal	139	133	272

plane resolution. Radiologists then selected and exported the diagnostically most informative axial slice per case in .jpg format.

5.4.2 Expert Annotation

After automatic de-identification, collaborating radiologists visually screened each volume, selected the *most representative* slice, and confirmed diagnosis per Fleischner Society's CT imaging criteria. Images were assigned to class-specific folders and double-checked by the opposite clinician to minimize label noise.

5.4.3 Dataset Statistics

Final **HLung-6** dataset comprises **793** annotated HRCT slices (**469** male, **324** female), summarized in Tables 5.2 and 5.3.

Sample distribution is intentionally imbalanced to reflect real-world disease prevalence;

Table 5.4: Model comparison (baseline models ordered by increasing parameter count; ZoomFormer models listed at end).

Model Name	Accuracy	Precision	Recall	Specificity	F_1
CNN-LSTM	0.8974	0.6920	0.6925	0.9377	0.6892
EfficientNet B0	0.8135	0.4266	0.4220	0.8872	0.4158
MobileNet V3	0.8042	0.4233	0.3963	0.8803	0.3896
DenseNet121	0.8601	0.5952	0.6121	0.9151	0.5986
ResNet18	0.8904	0.7306	0.6903	0.9342	0.6895
XceptionNet	0.9068	0.7828	0.7423	0.9434	0.7335
ResNet50	0.8531	0.6021	0.5689	0.9120	0.5591
VGG19	0.7366	0.2871	0.2085	0.8412	0.1945
ZoomFormer-small	*0.9580*	*0.8839*	*0.8741*	*0.9745*	*0.8760*
ZoomFormer-large	**0.9780**	**0.9039**	**0.8941**	**0.9895**	**0.8989**

class weighting is applied during training. All exported slices were rescaled to 512 × 512 pixels and intensity-standardized to zero mean and unit variance before training.

5.5 Experimental Results

5.5.1 Performance Comparison

Table 5.4 highlights the strong performance of **ZoomFormer** across all evaluation metrics. With approximately 2.02M parameters, **ZoomFormer-small** achieves accuracy 0.9580, precision 0.8839, recall 0.8741, specificity 0.9745, and F_1 score 0.8760. This substantially outperforms widely used lightweight and mid-sized baselines.

The larger variant **ZoomFormer-large** with approximately 4.23M parameters further improves performance to accuracy 0.9780, precision 0.9039, recall 0.8941, specificity 0.9895, and F_1 score 0.8989. These gains achieved with order-of-magnitude fewer parameters than many standard backbones underscore the effectiveness of combining residual and zooming blocks with proposed attention and transformer modules under a supervised contrastive training regime.

Table 5.5: Model Parameters and Training Time (baseline models ordered by increasing parameter count; ZoomFormer models shown separately at the end).

Model Name	**Parameters**	**Training Time (hrs)**
CNN-LSTM	1,796,879	1.7
EfficientNet B0	5,416,676	2.2
MobileNet V3	5,611,160	2.5
DenseNet121	8,106,984	3.1
ResNet18	11,817,640	3.4
XceptionNet	22,984,080	4.1
ResNet50	25,685,160	2.1
EfficientNet B7	66,476,088	4.2
VGG19	143,795,368	4.5
ZoomFormer-small	2,016,022	1.8
ZoomFormer-large	4,229,014	2.1

5.5.2 Computational Efficiency

Table 5.5 summarizes computational efficiency. Among all baselines, VGG19 has the highest parameter count, i.e., 143.80M, and requires 4.5 hours of training, followed by EfficientNet B7, i.e., 66.48M parameters with a training time of 4.2 hours.

In contrast, ZoomFormer-small uses only 2.02M parameters and converges in approximately 1.8 hours, while ZoomFormer-large uses 4.23M parameters with a training time of about 2.1 hours. Despite having the second-lowest parameter count after CNN-LSTM (1.80M parameters, 1.7 hours), ZoomFormer-small achieves the highest accuracy and F_1 score among all models except ZoomFormer-large, indicating a favorable trade-off between model size, training time, and recognition performance. ZoomFormer-large, while using twice the parameters of ZoomFormer-small, remains substantially more compact than all other baselines and achieves the highest overall accuracy and F_1 score across all evaluated models, further demonstrating that the proposed architecture efficiently scales performance without incurring the parameter overhead typical of standard backbones. Figure 5.4 provides a complementary Pareto-style visualization of the trade-off between model complexity and recognition performance.

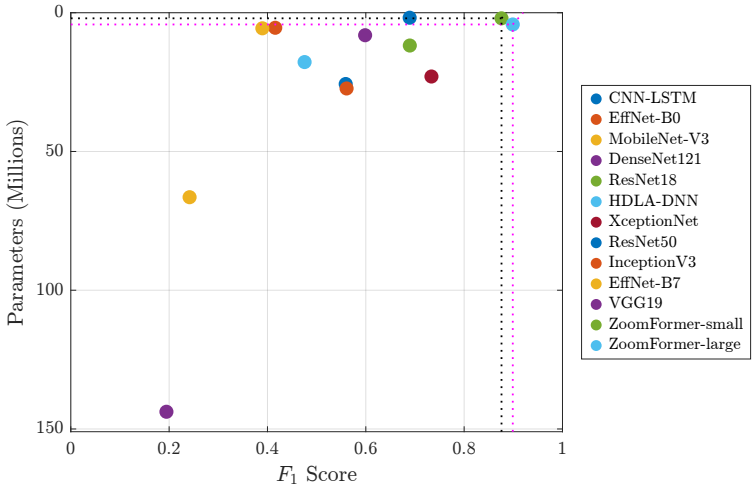


Figure 5.4: Pareto-style comparison of model complexity (parameters in millions) versus macro F_1 score on HLung-6 dataset, with ZoomFormer-small and ZoomFormer-large highlighted by dotted guide lines. Both variants achieve substantially higher F_1 scores than all baseline models while using far fewer parameters.

The plot highlights how ZoomFormer-small and ZoomFormer-large lie on the optimal accuracy-efficiency frontier: both variants achieve substantially higher macro F_1 scores than all baseline models while using far fewer parameters. Dotted guidelines further emphasize the sharp advantage of ZoomFormer in terms of both compactness and predictive power.

5.5.3 Ablation Studies

Effect of Proposed Attention Mechanism

The attention mechanism integrates spatial attention, channel attention, and efficient channel attention to selectively enhance diagnostically relevant regions. Figure 5.5 provides a qualitative comparison using Grad-CAM visualizations.

Without the attention module, activation maps are diffuse and distributed across irrelevant regions. With proposed attention, activation maps become sharply localized around pathological structures.

Quantitatively, attention yields substantial gains across all zoom factors. Table 5.6 shows that when using a zoom factor of 2, accuracy increases from **0.8880** (i.e., without attention)

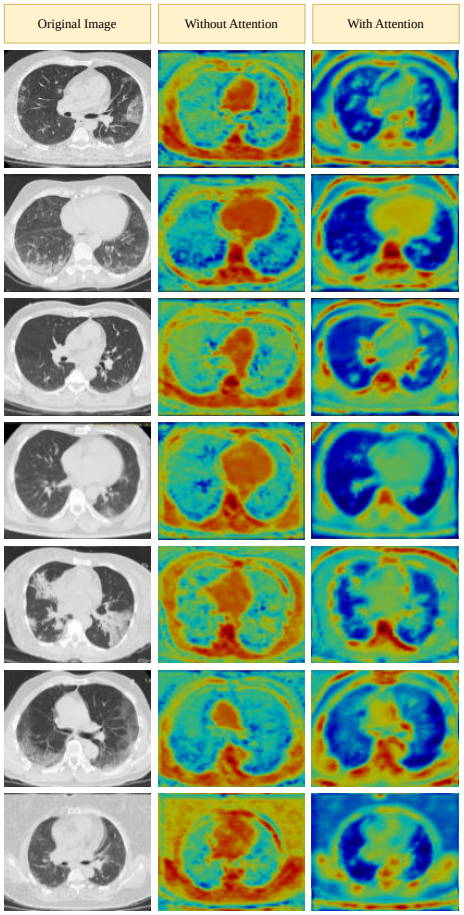


Figure 5.5: Grad-CAM comparison with and without the attention block. Without the attention module, activation maps are diffuse and distributed across irrelevant regions. With proposed attention, activation maps become sharply localized around pathological structures.

to **0.9780** (i.e., with attention).

For zoom factors 3 and 4, accuracy improves from **0.8810** to **0.9370** and from **0.8671** to **0.9020**, respectively. These results demonstrate that attention block meaningfully enhances feature localization and classification robustness.

Effect of Transformer Block

Table 5.7 quantifies the contribution of the transformer block within ZoomFormer by comparing both small and large variants with and without this module.

For *ZoomFormer-small*, incorporating the transformer block yields consistent gains across

Table 5.6: Ablation study showing the effect of attention for different zooming factors. The performance metric represents the accuracy achieved with and without the attention mechanism.

Zoom Factor	**With Attention**	**Without Attention**
2	97.80	88.80
3	93.70	88.10
4	90.20	86.71

Table 5.7: Performance of ZoomFormer variants with and without transformer block.

Metric	**With Transformer**	**Without Transformer**
	ZoomFormer-small	
Accuracy	0.9580	0.9464
Precision	0.8839	0.8297
Recall	0.8741	0.8277
Specificity	0.9745	0.9678
F_1 Score	0.8760	0.8259
	ZoomFormer-large	
Accuracy	0.9780	0.9674
Precision	0.9039	0.9192
Recall	0.8941	0.8947
Specificity	0.9895	0.9800
F_1 Score	0.8989	0.9013

all metrics. Accuracy increases from 0.9464 to 0.9580, while precision, recall, and F_1 improve from 0.8297, 0.8277, and 0.8259 to 0.8839, 0.8741, and 0.8760, respectively. These improvements highlight the transformer's ability to strengthen long-range feature interactions beyond what the residual-zooming pathway alone can capture.

For *ZoomFormer-large*, the effect is more nuanced. While accuracy and specificity benefit from the transformer block (increasing from 0.9674 to 0.9780 and from 0.9800 to 0.9895), precision and F_1 show slight decreases. This behavior suggests larger backbone already captures substantial global context, making the transformer block beneficial for overall robustness but not strictly necessary for all metrics.

Overall, ablation demonstrates the transformer block contributes most strongly to the *smaller* variant, where its ability to model global spatial relationships compensates for re-

duced feature capacity, while providing complementary, architecture-dependent benefits to the larger variant.

5.6 Discussion

5.6.1 Key Insights

ZoomFormer improves lung disease classification on HRCT images by capturing both fine-grained textures and broader contextual patterns through its multi-scale zooming blocks. Combined spatial, channel, and ECA attention modules in this focus the model on diagnostically relevant regions, and Grad-CAM visualizations show more precise lesion localization compared to models without attention. The lightweight transformer block models long-range dependencies and strengthens feature representation, especially in the smaller variant. Despite using only 4.23M parameters, ZoomFormer achieves state-of-the-art performance, significantly outperforming large baselines such as VGG19, making it practical for resource-constrained clinical settings. Supervised contrastive learning further improves class separation by encouraging tight intra-class clustering and clear inter-class boundaries in the embedding space.

5.6.2 Relation to Thesis Theme

Although ZoomFormer targets medical imaging, it embodies the thesis principle of adaptive inference under distribution shift. The HLung-6 dataset spans multiple hospitals and scanner types, introducing real-world heterogeneity, and the model responds by selectively allocating attention to diagnostically relevant regions rather than applying uniform model inference. By concentrating capacity where needed and using lightweight zooming blocks, ZoomFormer achieves strong accuracy–efficiency trade-offs, paralleling conditional compute allocation strategies in diffusion-based channel modeling and PRM-guided LLM reasoning.

5.6.3 Limitations and Future Work

Although HLung-6 provides valuable insights, evaluating ZoomFormer on the larger multi-institutional datasets would better assess generalization under stronger distribution shifts and broader pathologies. Extending the framework from 2D slices to full 3D volumetric inputs or multi-slice sequences could capture richer spatial context, while more advanced explainability methods, such as counterfactual or concept-based explanations, would strengthen clinical trust beyond Grad-CAM visualizations. Integrating multi-modal clinical metadata, including patient demographics, laboratory values, and radiology reports, could further enhance diagnostic performance and robustness.

5.7 Summary

This chapter introduced ZoomFormer, a lightweight attention-transformer architecture for lung disease classification from HRCT images. By integrating multi-scale residual and zooming blocks with spatial, channel, and ECA attention modules, transformer-based global context modeling, and supervised contrastive learning on the HLung-6 dataset, the model achieves strong performance with high efficiency. Extensive ablation studies validate the effectiveness of the attention mechanisms and architectural design. Overall, ZoomFormer demonstrates that selectively allocating model capacity to informative regions enables robust and efficient performance under distribution shift in medical imaging.

Chapter 6

Conclusions and Future Work

This chapter summarizes the key contributions of the thesis and presents a unified perspective on adaptive computation as a core principle for robust and efficient machine learning.

6.1 Summary of Contributions

This thesis addressed a central challenge in modern machine learning: how to perform *robust and efficient inference under distribution shift*. Across wireless communications (Mohsin et al., 2026), LLM reasoning (Bilal et al., 2026), and medical imaging, we showed that static inference procedures degrade when operating conditions change. Instead of applying uniform computation to all inputs, we developed methods that *adaptively allocate computational resources based on input characteristics and intermediate quality signals*.

1) First, for non-stationary wireless channel estimation, we developed a conditional diffusion framework that adapts its denoising process based on temporal context and signal quality, enabling robust estimation under mobility-induced distribution shift while reducing unnecessary computation.

2) Second, for LLM reasoning, we proposed a verifier-guided adaptive test-time framework that dynamically allocates reasoning strategies and computes per problem, using verification signals to focus effort where it matters most and improving accuracy through smarter utilization rather than brute-force scaling.

3) Third, for medical imaging, we introduced ZoomFormer, a lightweight attention-based architecture that selectively concentrates model capacity on diagnostically informative re-

gions, achieving strong performance with significantly fewer parameters by aligning computation with spatial relevance.

6.2 Unified Perspective

Although these contributions target different domains, they share a single unifying principle: *adaptive allocation of computational resources enables robust and efficient inference under distribution shift.* 1) In wireless systems, the model adapts denoising computation based on SNR and temporal context. 2) In LLM reasoning, the system allocates reasoning trajectories and tools based on verification signals and problem difficulty. 3) In medical imaging, the architecture focuses feature extraction on diagnostically relevant spatial regions using learned attention weights.

Across all domains, uniform computation proves inefficient and suboptimal when conditions vary. By conditioning inference on input-dependent signals, adaptive systems can adjust to new environments without retraining, concentrate effort on high-utility operations, and achieve superior accuracy-efficiency trade-offs.

6.3 Limitations and Future Work

While the results are strong, several limitations remain. 1) In wireless channel estimation, future work should address unknown or time-varying noise statistics, extend to massive MIMO settings, and evaluate more diverse propagation environments. 2) In LLM reasoning, improving verifier quality on the hardest problems, learning budget-aware control policies instead of using heuristic selection rules, and reducing system latency are important next steps. 3) In medical imaging, larger multi-institutional datasets, 3D volumetric modeling, stronger explainability techniques, and integration of clinical metadata would further strengthen robustness and clinical applicability.

More broadly, future research should develop learned adaptive policies using meta-learning

or reinforcement learning, establish theoretical foundations for adaptive inference under distribution shift, explore multi-modal and multi-task extensions, and address deployment challenges such as latency, edge constraints, fairness, and human-in-the-loop interaction.

6.4 Broader Impact and Final Remarks

Adaptive inference allows AI systems to regulate their reasoning depth, memory use, and search effort based on task difficulty, uncertainty, and feedback signals. Instead of applying fixed computation to every input, a self-evolving agent can expand exploration when uncertain, terminate early when confident, and iteratively refine decisions. This dynamic allocation makes large-scale intelligence more efficient and sustainable, particularly as models grow in capability. It also enables deployment on small and edge devices, where strict energy, memory, and latency constraints require intelligent resource management rather than brute-force scaling. By concentrating computation where it is most needed, adaptive inference forms a practical foundation for scalable and self-regulating AI systems.

Bibliography

Abbas, A., Abdelsamea, M. M. and Gaber, M. M. (2021), 'Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network', *Applied Intelligence* **51**(2), 854–864.
URL: *https://doi.org/10.1007/s10489-020-01829-7*

Abdolee, R., Champagne, B. and Sayed, A. H. (2013), Diffusion LMS strategies for parameter estimation over fading wireless channels, *in* '2013 IEEE International Conference on Communications (ICC)', IEEE, pp. 1926–1930.

Ali, K. S., Philip, S. P. and Perarasi, T. (2025), 'Gaussian mixture–expectation maximization-based learned AMP network with CNN deep residual network for millimeter wave communication', *International Journal of Communication Systems* **38**(4).

Alshmrani, G. M. M., Ni, Q., Jiang, R., Pervaiz, H. and Elshennawy, N. M. (2023), 'A deep learning architecture for multi-class lung diseases classification using chest x-ray (cxr) images', *Alexandria Engineering Journal* **64**, 923–935.
URL: *https://www.sciencedirect.com/science/article/pii/S1110016822007104*

An, S., Cai, X., Cao, X., Li, X., Lin, Y., Liu, J., Lv, X., Ma, D., Wang, X., Wang, Z. et al. (2025), 'AMO-bench: Large language models still struggle in high school math competitions'.

Anthimopoulos, M., Christodoulidis, S., Ebner, L., Christe, A. and Mougiakakou, S. (2016), 'Lung pattern classification for interstitial lung diseases using a deep convolutional neural network', *IEEE Transactions on Medical Imaging* **35**(5), 1207–1216.

Arellano, J. F., Altamirano, C. D., Carvajal Mora, H. R., Orozco Garzón, N. V. and Almeida García, F. D. (2024), 'On the performance of MMSE channel estimation in massive MIMO systems over spatially correlated Rician fading channels', *Wireless Communications and Mobile Computing* **2024**(1), 5445725.

Arvinte, M. and Tamir, J. I. (2022), 'MIMO channel estimation using score-based generative models', *IEEE Transactions on Wireless Communications* **22**(6), 3698–3713.

Bilal, A., Mohsin, M. A., Umer, M., Subhan, A., Rizwan, H., Mohsin, A. and Hougen, D. F. (2026), 'What if we allocate test-time compute adaptively?'.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C. and Mirhoseini, A. (2024), 'Large language monkeys: Scaling inference compute with repeated sampling'.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2021), An image is worth 16x16 words: Transformers for image recognition at scale, *in* 'International Conference on Learning Representations (ICLR)'.

Fesl, B., Böck, B., Strasser, F., Baur, M., Joham, M. and Utschick, W. (2024), 'On the asymptotic mean square error optimality of diffusion models'.

Fesl, B., Strasser, M., Joham, M. and Utschick, W. (2024), 'Diffusion-based generative prior for low-complexity MIMO channel estimation', *IEEE Wireless Communications Letters* .

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2020), 'Generative adversarial networks', *Communications of the ACM* **63**(11), 139–144.

Goyal, S. and Singh, R. (2023), 'Detection and classification of lung diseases for pneumonia and COVID-19 using machine and deep learning techniques', *Journal of Ambient Intelligence and Humanized Computing* **14**(4), 3239–3259.
URL: *https://doi.org/10.1007/s12652-021-03464-7*

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A. et al. (2024), 'The Llama 3 herd of models'.

Graves, A. (2012), Long short-term memory, *in* 'Supervised Sequence Labelling with Recurrent Neural Networks', Springer, Berlin, Heidelberg, pp. 37–45.

He, H., Wen, C.-K., Jin, S. and Li, G. Y. (2018), 'Deep learning-based channel estimation for beamspace mmWave massive MIMO systems', *IEEE Wireless Communications Letters* **7**(5), 852–855.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D. and Steinhardt, J. (2021), 'Measuring mathematical problem solving with the MATH dataset'.

Ho, J., Jain, A. and Abbeel, P. (2020), Denoising diffusion probabilistic models, *in* 'Advances in Neural Information Processing Systems', Vol. 33, pp. 6840–6851.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. et al. (2019), Searching for mobilenetv3, *in* 'Proceedings of the IEEE/CVF international conference on computer vision', pp. 1314–1324.

Hu, J., Shen, L. and Sun, G. (2018), Squeeze-and-excitation networks, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 7132–7141.

Huang, S., Lee, F., Miao, R., Si, Q., Lu, C. and Chen, Q. (2020), 'A deep convolutional neural network architecture for interstitial lung disease pattern classification', *Medical & Biological Engineering & Computing* **58**(4), 725–737.
URL: *https://doi.org/10.1007/s11517-019-02111-w*

Jadhav, S. P., Singh, H., Hussain, S., Gilhotra, R., Mishra, A., Prasher, P., Krishnan, A. and Gupta, G. (2021), *Introduction to Lung Diseases*, Springer.
URL: *https://doi.org/10.1007/978-981-33-6827-9_1*

Khalili, A., Rastegarnia, A. and Bazzi, W. M. (2015), Effect of fading channels on the performance of distributed networks with cyclic cooperation, *in* '2015 International Conference on Information and Communication Technology Research (ICTRC)', IEEE, pp. 32–35.

Kim, G. B., Jung, K.-H., Lee, Y., Kim, H.-J., Kim, N., Jun, S., Seo, J. B. and Lynch, D. A. (2018), 'Comparison of shallow and deep learning methods on classifying the regional pattern of diffuse lung disease', *Journal of Digital Imaging* **31**(4), 415–424.
URL: *https://doi.org/10.1007/s10278-017-0028-9*

Kim, S., Rim, B., Choi, S., Lee, A., Min, S. and Hong, M. (2022), 'Deep learning in multi-class lung diseases' classification on chest x-ray images', *Diagnostics* **12**(4).
URL: *https://www.mdpi.com/2075-4418/12/4/915*

Kingma, D. P. and Welling, M. (2019), 'An introduction to variational autoencoders', *Foundations and Trends® in Machine Learning* **12**(4), 307–392.

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J. et al. (2023), 'Measuring faithfulness in chain-of-thought reasoning'.

Li, J., Chen, J., Tang, Y., Wang, C., Landman, B. A. and Zhou, S. K. (2023), 'Transforming medical imaging with transformers? A comparative review of key properties, current progresses, and future perspectives', *Medical image analysis* **85**, 102762.

Liang, L., Xu, W. and Dong, X. (2014), 'Low-complexity hybrid precoding in massive multiuser MIMO systems', *IEEE Wireless Communications Letters* **3**(6), 653–656.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I. and Cobbe, K. (2023), Let's verify step by step, *in* 'The Twelfth International Conference on Learning Representations'.

Liu, T., Wang, Z., Miao, J., Hsu, I., Yan, J., Chen, J., Han, R., Xu, F., Chen, Y., Jiang, K. et al. (2025), 'Budget-aware tool-use enables effective agent scaling'.

McAleese, N., Pokorny, R. M., Uribe, J. F. C., Nitishinskaya, E., Trebacz, M. and Leike, J. (2024), 'LLM critics help catch LLM bugs'.

Mohsin, M. A., Bilal, A., Umer, M., Aali, A., Jamshed, M. A., Hougen, D. F. and Cioffi, J. M. (2026), Conditional prior-based non-stationary channel estimation using accelerated diffusion models, *in* '2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 1–5.

O'Shea, K. and Nash, R. (2015), 'An introduction to convolutional neural networks'.

Paglieri, D., Cupiał, B., Cook, J., Piterbarg, U., Tuyls, J., Grefenstette, E., Foerster, J. N., Parker-Holder, J. and Rocktäschel, T. (2025), 'Learning when to plan: Efficiently allocating test-time compute for LLM agents'.

Pereira, G. A. and Hussain, M. (2024), 'A review of transformer-based models for computer vision tasks: Capturing global context and spatial relationships'.

Qwen Team (2024), 'Qwen2.5: A party of foundation models', https://qwenlm.github.io/blog/qwen2.5/.

Rakhmania, A. E., Hudiono, H., Ro'isatin, U. A. and Hidayati, N. (2025), 'Channel estimation methods in massive MIMO: A comparative review of machine learning and traditional techniques', *Infocommunications Journal* **17**(1), 19–31.

Rakhsha, A., Madan, K., Zhang, T., Farahmand, A.-m. and Khasahmadi, A. (2025), 'Majority of the bests: Improving best-of-n via bootstrapping'.

Ranjan, A. and Sahana, B. C. (2025), 'Deep learning empowered channel estimation in massive MIMO: Unveiling the efficiency of hybrid deep learning architecture', *Journal of Ambient Intelligence and Humanized Computing* **16**(2), 375–390.

Snell, C., Lee, J., Xu, K. and Kumar, A. (2024), 'Scaling LLM test-time compute optimally can be more effective than scaling model parameters'.

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* 'International Conference on Learning Representations (ICLR)'.

Sprague, Z., Yin, F., Rodriguez, J. D., Jiang, D., Wadhwa, M., Singhal, P., Zhao, X., Ye, X., Mahowald, K. and Durrett, G. (2024), 'To CoT or not to CoT? chain-of-thought helps mainly on math and symbolic reasoning'.

Tan, M. and Le, Q. (2019), Efficientnet: Rethinking model scaling for convolutional neural networks, *in* 'International conference on machine learning', PMLR, pp. 6105–6114.

Turpin, M., Michael, J., Perez, E. and Bowman, S. (2023), Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting, *in* 'Advances in Neural Information Processing Systems', Vol. 36, pp. 74952–74965.

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', Vol. 30.

Wan, G., Wu, Y., Chen, J. and Li, S. (2025), Reasoning aware self-consistency: Leveraging reasoning paths for efficient LLM sampling, *in* 'Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)', pp. 3613–3635.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W. and Hu, Q. (2020), Eca-net: Efficient channel attention for deep convolutional neural networks, *in* 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 11534–11542.

Wang, Y., Qiu, Y., Cheng, P. and Zhang, J. (2022), 'Hybrid CNN-transformer features for visual place recognition', *IEEE Transactions on Circuits and Systems for Video Technology* **33**(3), 1109–1122.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D. et al. (2022), Chain-of-thought prompting elicits reasoning in large language models, *in* 'Advances in Neural Information Processing Systems', Vol. 35, pp. 24824–24837.

Wiseman, S. and Rush, A. M. (2016), Sequence-to-sequence learning as beam-search optimization, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', pp. 1296–1306.

Xie, Y., Goyal, A., Zheng, W., Kan, M.-Y., Lillicrap, T. P., Kawaguchi, K. and Shieh, M. (2024), 'Monte Carlo tree search boosts reasoning via iterative preference learning'.

Xu, M.-w., Zhang, Z.-h., Wang, X., Li, C.-t., Yang, H.-y., Liao, Z.-h. and Zhang, J.-q. (2025), 'PMFF-Net: A deep learning-based image classification model for UIP, NSIP, and OP', *Computers in Biology and Medicine* **195**, 110618.

Yu, X., Peng, B., Galley, M., Gao, J. and Yu, Z. (2024), Teaching language models to self-improve through interactive demonstrations, *in* 'Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)', pp. 5127–5149.

Zak, M. and Krzyżak, A. (2020), Classification of lung diseases using deep learning models, *in* V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos and J. Teixeira, eds, 'Computational Science – ICCS 2020', Springer International Publishing, Cham, pp. 621–634.

Zhang, J., He, L., Wei, Y., Tong, J., Yang, K., Wu, J., Guo, Y., Shi, F. and Jin, C. (2025), 'Deep learning for classifying imaging patterns of interstitial lung disease associated with idiopathic inflammatory myopathies', *Scientific Reports* **15**(1), 31655.

Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J. and Lin, J. (2025), 'The lessons of developing process reward models in mathematical reasoning'.

Zou, J., Yang, L., Gu, J., Qiu, J., Shen, K., He, J. and Wang, M. (2025), 'ReasonFlux-PRM: Trajectory-aware PRMs for long chain-of-thought reasoning in LLMs'.