UCLA
UCLA Electronic Theses and Dissertations

## Title

Conditional Conformal Prediction for In-Context Learning: Architecture Scaling, Task Complexity, and Signal-to-Noise Effects

## Permalink

https://escholarship.org/uc/item/1zf327gt

## ISBN

9798247928409

## Author

Lin, Jinrui

## Publication Date

2026-05-29

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA

Los Angeles

# Conditional Conformal Prediction for In-Context Learning: Architecture Scaling, Task Complexity, and Signal-to-Noise Effects

A thesis submitted in partial satisfaction
of the requirements for the degree Master of Science
in Statistics

by

Jinrui Lin

2026

© Copyright by

Jinrui Lin

2026

ABSTRACT OF THE THESIS

Conditional Conformal Prediction for In-Context Learning:
Architecture Scaling, Task Complexity,
and Signal-to-Noise Effects

by

Jinrui Lin
Master of Science in Statistics
University of California, Los Angeles, 2026
Professor Xiaowu Dai, Chair

Transformer-based in-context learning (ICL) has demonstrated remarkable ability to perform regression tasks by conditioning on demonstration examples within a single forward pass, without parameter updates. While ICL predictions improve with more demonstrations, quantifying the uncertainty of these predictions with formal coverage guarantees remains an open challenge. In this work, we conduct a systematic empirical study of conditional conformal prediction methods—specifically CondConf and its computationally efficient variant SpeedCP—applied to ICL regression across multiple experimental axes. We investigate how *model architecture* (width, depth, and standard presets), *task complexity* (linear, quadratic, neural network, and decision tree regression), *signal-to-noise ratio*, and *input dimensionality* jointly influence both point prediction quality and the resulting conformal prediction intervals. Across 44 experimental configurations encompassing 12 architectures (S13–S24), 16 task–noise combinations (S52–S67), and 16 dimensionality configurations (S69–S84, 4 tasks $\times$ 4 dimensions), we find that SpeedCP maintains empirical coverage consistently near the nominal 95% level (93.1%–96.1%). However, the efficiency of conformal intervals—measured by average width—varies dramatically: from a 84% width reduction over the ICL trajectory for large-capacity models on low-noise linear tasks, to essentially no improvement for under-capacity models or high-noise regimes where irreducible noise dominates. At high input dimensions ($d = 100$), interval widths increase sharply despite separately training each model for its

target dimensionality. These findings provide practical guidance for deploying conformal prediction in ICL systems and highlight the intimate coupling between model capacity, task learnability, and uncertainty quantification quality.

The thesis of Jinrui Lin is approved.

Yuhua Zhu

Qing Zhou

Xiaowu Dai, Committee Chair

University of California, Los Angeles

2026

# Contents

# List of Figures

# List of Tables

# Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Xiaowu Dai, for his invaluable guidance, unwavering support, and intellectual inspiration throughout my graduate studies. His rigorous approach to statistical research and his patience in mentoring have profoundly shaped my academic growth. I am also sincerely grateful to Professors Yuhua Zhu and Qing Zhou for serving on my thesis committee and for their constructive feedback that greatly improved this work.

I owe a profound debt of gratitude to my family. To my parents, thank you for your unconditional love, your sacrifices, and your steadfast belief in me across the miles. Your encouragement has been the foundation upon which I have built everything.

I am also thankful to my friends Wenhao Kuang and Zhenyu He for their companionship and encouragement throughout this journey. I wish them all the best in their future endeavors.

Finally, I would like to acknowledge my own perseverance through the challenges of graduate school. The late nights, the moments of doubt, and the breakthroughs that followed have all been part of a journey that I am proud to have completed.

# CHAPTER 1

# Introduction

In-context learning (ICL) refers to the ability of large transformer models to adapt to new tasks at inference time by conditioning on a sequence of input–output demonstration pairs, without any gradient-based parameter updates [Garg et al., 2022, Brown et al., 2020]. Given a sequence of demonstrations $\{(x_1, y_1), \ldots, (x_L, y_L)\}$ followed by a query input $x_{L+1}$, a transformer trained on diverse regression tasks can produce a prediction $\hat{y}_{L+1}$ that improves as the number of demonstrations $L$ increases. This phenomenon has attracted significant attention in the machine learning community, both for its practical utility and its theoretical implications regarding the implicit algorithms implemented by transformer architectures [Akyurek et al., 2023, von Oswald et al., 2023].

Despite the empirical success of ICL, a critical gap remains in *uncertainty quantification*. In safety-critical applications—such as medical diagnosis, autonomous systems, and financial modeling—point predictions alone are insufficient; practitioners require prediction intervals with formal coverage guarantees. Conformal prediction [Vovk et al., 2005, Shafer and Vovk, 2008] provides a distribution-free framework for constructing such intervals: given exchangeable calibration data, conformal methods produce prediction sets that achieve marginal coverage $\Pr(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha$ with finite-sample validity.

However, marginal coverage is a *global* guarantee that may mask significant heterogeneity. A prediction interval that achieves 95% coverage on average may systematically under-cover in certain regions of the input space while over-covering in others. This limitation motivates *conditional* conformal prediction methods that aim for coverage guarantees of the form $\Pr(Y_{n+1} \in \hat{C}(X_{n+1}) \mid X_{n+1} = x) \geq 1 - \alpha$, or suitable relaxations thereof.

Gibbs et al. [2023] introduced CondConf, which formulates conditional conformal prediction as an augmented quantile regression problem over a reproducing kernel Hilbert space (RKHS). By expressing the conditional coverage target as a functional constraint over a rich function class $\mathscr{F}$,

CondConf interpolates between marginal and conditional coverage depending on the expressiveness of $\mathscr{F}$. However, the computational cost of CondConf scales poorly with the calibration set size, as each candidate conformity score requires solving a fresh optimization problem. To address this bottleneck, Jung et al. [2024] proposed SpeedCP, which exploits the piecewise-affine structure of the RKHS quantile regression solution path to trace the optimal conformal cutoff efficiently, achieving order-of-magnitude speedups while producing identical prediction intervals.

In this paper, we apply SpeedCP to the ICL regression setting and conduct a comprehensive empirical study along three axes:

1. **Architecture scaling**: How do model width, depth, and overall capacity affect both ICL prediction quality and the resulting conformal interval efficiency?
2. **Task complexity and signal-to-noise ratio**: How do different regression function classes (linear, quadratic, two-layer neural network, decision tree) and noise levels ($\sigma \in \{0.1, 0.25, 0.5, 1.0\}$) interact with ICL and conformal calibration?
3. **Input dimensionality**: How does the ambient dimension of the regression problem affect ICL prediction quality and conformal coverage? (Section 6.)

Our key findings are: (i) conformal coverage remains stable near the nominal level across all configurations, validating the theoretical guarantees of SpeedCP; (ii) interval width is primarily determined by the underlying point prediction quality (RMSE), with a near-perfect correlation ($\rho \approx 0.999$) between final RMSE and final interval width; (iii) model architecture has a dramatic impact on interval efficiency, with width reduction ranging from 84% (GPT-2 large) to $-3\%$ (GPT-2 tiny); and (iv) high noise ($\sigma = 1.0$) creates an irreducible floor that prevents interval shrinkage regardless of context length.

# CHAPTER 2

# Methodology

## 2.1 Problem Formulation

We consider the ICL regression setting where a transformer model $f_\theta$ processes a sequence of $L$ demonstration pairs followed by a query:

$$\hat{y}_{L+1} = f_\theta\big((x_1, y_1), \ldots, (x_L, y_L), x_{L+1}\big).$$

The goal is to construct a symmetric predictive interval

$$\hat{C}(x_{L+1}) = \big[\hat{y}_{L+1} - q(x_{L+1}),\ \hat{y}_{L+1} + q(x_{L+1})\big] \tag{2.1}$$

that satisfies the coverage guarantee $\Pr(Y_{L+1} \in \hat{C}(X_{L+1})) \geq 1 - \alpha$ at a prescribed level $\alpha = 0.05$. As conformity scores, we use absolute residuals $S_i = |y_i - \hat{y}_i|$.

To evaluate conformal methods, we partition each ICL evaluation episode into a calibration set of $n = 800$ points and a test set of 800 points. The calibration set is used to fit the conformal cutoff, and coverage and interval width are measured on the held-out test set.

## 2.2 CondConf: Conditional Conformal via RKHS Quantile Regression

To achieve approximate conditional coverage, Gibbs et al. [2023] proposed CondConf, which replaces the constant quantile with a covariate-dependent function $g : \mathscr{X} \to \mathbb{R}$. The conditional

coverage target is relaxed to require

$$\mathbb{E}\big[f(X_{n+1})\big(\mathbf{1}\{Y_{n+1} \in \hat{C}(X_{n+1})\} - (1-\alpha)\big)\big] = 0, \quad \forall f \in \mathscr{F}, \tag{2.2}$$

where the choice of function class $\mathscr{F}$ controls the strength of the coverage guarantee. When $\mathscr{F} = \{x \mapsto 1\}$, this reduces to marginal coverage; when $\mathscr{F}$ is an RKHS, it yields approximate conditional coverage.

The key innovation of CondConf is the *augmented quantile regression* formulation: the test point $(X_{n+1}, S)$ is included in the dataset when fitting the conditional quantile, with $S$ treated as a free parameter. Using the function class $\mathscr{F} = \{g(\cdot) = \Phi(\cdot)^\top \beta + g_K(\cdot) : g_K \in \mathscr{H}_K\}$ (defined formally in Section 2.5), the optimization problem is

$$\hat{g}_S = \arg\min_{g \in \mathscr{F}} \frac{1}{n+1} \sum_{i=1}^{n} \ell_\tau(g(X_i), S_i) + \frac{1}{n+1} \ell_\tau(g(X_{n+1}), S) + \frac{\lambda}{2} \|g_K\|^2_{\mathscr{H}_K}, \tag{2.3}$$

where $\ell_\tau(q,s) := \tau(s-q)_+ + (1-\tau)(q-s)_+$ is the pinball loss at level $\tau = 1-\alpha$.

By the representer theorem, the RKHS component admits the form $g_K(x) = \frac{1}{\lambda}\sum_{j=1}^{n} v_j K(x, X_j)$. The dual formulation yields a quadratic program:

$$\min_{v,\beta} \ \frac{1}{2\lambda} v^\top \mathbf{K} v - v^\top \mathbf{S} \quad \text{s.t.} \quad v_i \in [\tau - 1,\ \tau],\ \blacksquare^\top v = 0, \tag{2.4}$$

where $\mathbf{K}$ is the kernel matrix with entries $K_{ij} = K(X_i, X_j)$. A crucial theoretical result (Gibbs et al. [2023], Theorem 4) establishes that the map $S \mapsto \hat{g}_S(X_{n+1})$ is monotonically non-decreasing, enabling efficient computation of the prediction interval via binary search.

## 2.3 SpeedCP: Path-Tracing Acceleration

While CondConf provides strong theoretical guarantees, its computational cost is prohibitive for large calibration sets, as each evaluation of the binary search requires solving a full optimization

problem. Jung et al. [2024] introduced SpeedCP, which exploits the piecewise-affine structure of the solution path to trace the optimal conformal cutoff efficiently.

The key observation is that the KKT conditions of (2.4) partition the calibration data into three sets at any solution point:

$$\begin{aligned} \mathscr{E} &= \{i : \tau - 1 < v_i < \tau\} && \text{(Elbow: active constraints)}, \\ \mathscr{L} &= \{i : v_i = \tau - 1\} && \text{(Left: lower bound)}, \\ \mathscr{R} &= \{i : v_i = \tau\} && \text{(Right: upper bound)}. \end{aligned} \tag{2.5}$$

Between consecutive *events*—points entering or leaving the Elbow set—the dual variables $v_i$ and the intercept coefficients $\beta$ evolve as affine functions of the path parameter. SpeedCP exploits this structure in a two-stage algorithm:

**Stage 1 ($\lambda$-path).** Using only the calibration data (excluding the test point), SpeedCP traces the solution path as $\lambda$ decreases from $\infty$, performing cross-validation over a grid of kernel bandwidth parameters $\gamma$ to select the optimal hyperparameters $(\hat{\gamma}, \hat{\lambda})$.

**Stage 2 ($S$-path).** For each test point, with $(\hat{\gamma}, \hat{\lambda})$ fixed, SpeedCP traces the solution path as the imputed test score $S$ increases from an initial value. The monotonicity of $S \mapsto v_{n+1}(S)$ guarantees that the algorithm terminates when $v_{n+1}(S^*) \geq 1 - \alpha$, yielding the conformal cutoff $q(x_{n+1}) = \max(0, S^*)$.

The computational advantage stems from the fact that the Elbow set $|\mathscr{E}|$ is typically much smaller than $n$, and only $O(n)$ events occur along the entire path. In our experiments, Stage 1 tuning completes in approximately 0.2 seconds on average, while Stage 2 requires approximately 3.8 seconds per test batch.

## 2.4 Algorithm Summary

Algorithm 1 summarizes the complete SpeedCP-based conformal prediction pipeline for ICL evaluation.

## 2.5 Mathematical Preliminaries

**Notation.** Fix a context length $L$ and input dimension $d$. In each evaluation episode $i$, the prompt consists of $L$ demonstration pairs $\mathscr{D}_i := \{(x_{i,\ell}, y_{i,\ell})\}_{\ell=1}^{L}$ and a query $x_{i,L+1}$ with unobserved response $y_{i,L+1}$. A transformer with frozen parameters $\theta$ produces the point prediction $\hat{y}_{i,L+1} = f_\theta(\mathscr{D}_i, x_{i,L+1})$. We define the absolute residual conformity score $S_i := |y_{i,L+1} - \hat{y}_{i,L+1}|$ and the symmetric prediction interval

$$\hat{C}(x) := \big[\hat{y}(x) - q(x),\ \ \hat{y}(x) + q(x)\big], \tag{2.6}$$

so that $y \in \hat{C}(x) \iff |y - \hat{y}(x)| \le q(x)$. The conformal unit for episode $i$ is $Z_i := (\mathscr{D}_i, x_{i,L+1}, y_{i,L+1})$.

**Pinball loss and RKHS class.** Let $\tau := 1 - \alpha$. The pinball loss at level $\tau$ is $\ell_\tau(q,s) := \tau(s - q)_+ + (1-\tau)(q-s)_+$, minimized at the $\tau$-quantile. Following Gibbs et al. [2023] and Jung et al. [2024], we define the RKHS-plus-intercept function class

$$\mathscr{F} = \big\{ g(\cdot) = \Phi(\cdot)^\top \beta + g_K(\cdot) : \beta \in \mathbb{R}^{p_\Phi},\ g_K \in \mathscr{H}_K \big\}, \tag{2.7}$$

where $\mathscr{H}_K$ is the RKHS of the Gaussian RBF kernel $K(x,x') = \exp(-\gamma\|x - x'\|^2)$ and $\Phi(x) = 1$ (intercept-only) in all experiments. The augmented quantile regression fits a conditional cutoff by solving

$$\hat{g}_S := \arg\min_{g \in \mathscr{F}} \ \frac{1}{n+1} \sum_{i=1}^{n} \ell_\tau(g(X_i), S_i) + \frac{1}{n+1} \ell_\tau(g(X_{n+1}), S) + \tfrac{\lambda}{2} \|g_K\|^2_{\mathscr{H}_K}, \tag{2.8}$$

with the test score $S$ treated as a free parameter. The conformal prediction set is $\hat{C}(X_{n+1}) := \{y : S(X_{n+1}, y) \le \hat{g}_{S(X_{n+1},y)}(X_{n+1})\}$.

**ICL prediction error.** Let $f^\star(X)$ be the true task function, $Y = f^\star(X) + \varepsilon$ with independent noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, and $e(X) := f^\star(X) - f_\theta(X)$ be the ICL prediction bias. The expected squared error, with expectation taken over the test query $X$ and over the random prompt, is $r(L) := \mathbb{E}[e(X)^2]$. A growing body of theoretical work provides explicit bounds on $r(L)$; we defer a comprehensive survey to Section 2.7.

**CondConf/SpeedCP and the calibration gap.** Gibbs et al. [2023] showed that under RKHS-regularized calibration, the conditional coverage gap under a covariate shift $f \in \mathscr{F}$ is governed by the inner product between the fitted kernel component and the shift function. Specifically, defining

$$G_{\text{cal}}(n, \lambda, \gamma, f) := \frac{2\lambda\, \mathbb{E}[\langle \hat{g}_{S_{n+1},K}, f_K \rangle_{\mathscr{H}_K}]}{\mathbb{E}[f(X)]}, \tag{2.9}$$

the one-sided guarantee is $\mathbb{P}_f(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha - G_{\text{cal}}$ [Gibbs et al., 2023, Theorem 3]. When $f_K = 0$ (e.g., $f \equiv 1$), the gap vanishes and exact marginal coverage is recovered. SpeedCP [Jung et al., 2024] solves the same optimization (2.8) via efficient path-tracing, producing identical intervals; all coverage guarantees transfer without modification.

## 2.6 Theoretical Guarantees

We present three pillars connecting ICL prediction quality to conformal interval behavior: *exchangeability* (marginal validity), *width* (interval size controlled by ICL error), and *coverage* (conditional calibration gap). Each pillar is developed in its own subsection below.

### 2.6.1 Exchangeability

**Assumption 2.1** (Exchangeability)**.** *The conformal units* $(Z_1, \ldots, Z_n, Z_{n+1})$ *are exchangeable.*

This holds whenever episodes are drawn i.i.d. and randomly partitioned into calibration and test sets. Under Assumption 2.1, the marginal validity guarantee $\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha$ follows from the classical conformal argument [Vovk et al., 2005, Gibbs et al., 2023].

### 2.6.2 Width Pillar

**Theorem 2.1** (Width: interval width controlled by ICL error)**.** *Let $q^\star(x) := Q_\tau(S \mid X = x)$ be the oracle conditional cutoff and $q_\varepsilon := Q_\tau(|\varepsilon|)$ the noise floor ($= \sigma z_{1-\alpha/2}$ for Gaussian noise), and let $e(x) := f^\star(x) - f_\theta(x; \mathscr{P})$ be the ICL prediction bias at query $x$ under a random prompt $\mathscr{P}$. Then the expected conformal interval width satisfies*

$$\mathbb{E}\big[2q^\star(X)\big] \;\le\; 2q_\varepsilon \;+\; 2\sqrt{r(L)}, \tag{2.10}$$

*where $r(L) = \mathbb{E}[e(X)^2]$ is the ICL prediction error.*

The proof relies on a quantile stability lemma and the 1-Lipschitz property of the absolute value. For Gaussian noise with small bias $|e| \ll \sigma$, a Taylor expansion shows $q^\star(x) = q_\varepsilon + \frac{z_{1-\alpha/2}}{2\sigma} e(x)^2 + O(e^4/\sigma^3)$, so the width deviation is *second-order* in the prediction bias—explaining the empirical stability of coverage even when RMSE is non-negligible.

*Remark.* Theorem 2.1 controls the *oracle* conditional cutoff $q^\star(x) = Q_\tau(S \mid X = x)$, not the fitted SpeedCP cutoff $\hat{q}(x)$. The oracle cutoff is a population-level quantile determined entirely by the noise distribution and the ICL prediction error, so it does not involve $\lambda$. The behavior of the fitted cutoff is governed instead by Theorem 2.9 and Corollary 2.10 below, where a $\lambda$-trade-off emerges on the coverage side. A fitted-width refinement that extends this result to the SpeedCP half-width $\hat{q}(x)$ and reveals a $\sqrt{\lambda}$-vs-$1/\lambda$ trade-off is developed below in this section.

**Lemma 2.2** (Quantile stability under bounded perturbation)**.** *If random variables $U, V$ satisfy $|U - V| \le b$ almost surely, then for any $p \in (0,1)$, $|Q_p(U) - Q_p(V)| \le b$.*

*Proof.* From $|U - V| \le b$, we have $\{V \le t - b\} \subseteq \{U \le t\} \subseteq \{V \le t + b\}$, yielding $F_V(t-b) \le F_U(t) \le F_V(t+b)$. Setting $t = Q_p(V) + b$ gives $F_U(t) \ge p$, so $Q_p(U) \le Q_p(V) + b$; swapping $U, V$ gives the reverse inequality. □

*Proof of Theorem 2.1.* For fixed $x$ and prompt $\mathscr{P}$, $e(x)$ is a constant. Let $U_x := |\varepsilon + e(x)|$ and $V := |\varepsilon|$. By the 1-Lipschitz property of the absolute value, $|U_x - V| \le |e(x)|$ a.s. Lemma 2.2

immediately gives the pointwise bound

$$|q^\star(x) - q_\varepsilon| \;\le\; |e(x)| \qquad \text{for every } x. \tag{2.11}$$

Taking expectations and applying Jensen's inequality,

$$\mathbb{E}[q^\star(X)] \le q_\varepsilon + \mathbb{E}[|e(X)|] \le q_\varepsilon + \sqrt{\mathbb{E}[e(X)^2]} = q_\varepsilon + \sqrt{r(L)},$$

establishing (2.10). For $\varepsilon \sim \mathcal{N}(0,\sigma^2)$ and small bias $|e| \ll \sigma$, denoting the standard-normal CDF by $\Phi_{\mathcal{N}}$ (distinct from the basis $\Phi$ in Section 2.5), an implicit function theorem expansion of $\Phi_{\mathcal{N}}((q-e)/\sigma) + \Phi_{\mathcal{N}}((q+e)/\sigma) = 2-\alpha$ around $(q_\varepsilon, 0)$ yields the second-order expansion $q^\star(x) = q_\varepsilon + \frac{z_{1-\alpha/2}}{2\sigma}\, e(x)^2 + O(e^4/\sigma^3)$; the first-order term vanishes by symmetry of $\varepsilon$ around zero. □

**Fitted-width refinement.** Theorem 2.1 controls the oracle conditional cutoff $q^\star(x)$ and does not involve $\lambda$. We now extend the width analysis to the *fitted* SpeedCP half-width $\hat{q}(x)$, which depends on $\lambda$ through the augmented regularized quantile regression. The fitted width decomposes into four layers—oracle width, regularization bias, sample transfer, and augmentation perturbation—and the interplay between regularization bias ($\propto \sqrt{\lambda}$) and sample transfer ($\propto 1/\lambda$) produces a trade-off with the same $2/3$-power structure as the coverage-side result (Corollary 2.10).

**Additional notation.** Let $\mathscr{R}(g) := \mathbb{E}[\ell_\tau(g(X),S)]$ denote the population pinball risk. Define the *population regularized target*

$$g^\star_\lambda \;\in\; \arg\min_{g\in\mathscr{F}} \left\{ \mathscr{R}(g) + \tfrac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K} \right\},$$

the *non-augmented empirical fit*

$$\bar{g}_n \;\in\; \arg\min_{g\in\mathscr{F}} \left\{ \frac{1}{n}\sum_{i=1}^n \ell_\tau(g(X_i),S_i) + \tfrac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K} \right\},$$

and, for test covariate $x$ and imputed score $s$, the *augmented fit*

$$\hat{g}^{\mathrm{aug}}_{x,s} \;\in\; \arg\min_{g\in\mathscr{F}} \left\{ \frac{1}{n+1}\sum_{i=1}^{n} \ell_\tau(g(X_i),S_i) + \frac{1}{n+1}\ell_\tau(g(x),s) + \tfrac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K} \right\}.$$

The *fitted half-width* is $T_{n,x}(s) := \hat{g}^{\mathrm{aug}}_{x,s}(x)$ and $\hat{q}(x) := \sup\{s \geq 0 : s \leq T_{n,x}(s)\}$, consistent with SpeedCP's $S$-path output. We further define the *evaluation constant* $L_0 := \sup_x(|\Phi(x)|_2 + \sqrt{K(x,x)})$, which equals 2 in all experiments ($\Phi(x) = 1$, RBF kernel with $K(x,x) = 1$); the *approximation envelope*

$$A_{\mathrm{appr}}(\lambda) \;:=\; \inf_{g\in\mathscr{F}} \left\{ \mathscr{R}(g) - \mathscr{R}(q^\star) + \tfrac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K} \right\};$$

the *sample-transfer term* $s_{n,\lambda} := \mathbb{E}|\bar{g}_n(X) - g^\star_\lambda(X)|$; the *augmentation perturbation* $a^{\mathrm{aug}}_n := \mathbb{E}|\hat{q}(X) - \bar{g}_n(X)|$; and the *parameter-space distance* $d((\beta,g_K),(\beta',g'_K)) := |\beta-\beta'|_2 + \|g_K - g'_K\|_{\mathscr{H}_K}$ (following Gibbs et al., 2023, Appendix A.4.2).

**Assumption 2.2** (Quantile margin / density lower bound)**.** *There exists a constant $m_q > 0$ such that for all $g \in \mathscr{F}$ in a neighborhood of $q^\star$,*

$$\mathscr{R}(g) - \mathscr{R}(q^\star) \;\geq\; \frac{m_q}{2}\,\mathbb{E}\big[(g(X) - q^\star(X))^2\big].$$

*Remark.* This is a standard quantile-regression margin condition, satisfied whenever $m_q \geq \inf_x f_{S|X=x}(q^\star(x)) > 0$. For our Gaussian noise setting, $S = |\varepsilon + e(x)|$ follows a folded normal distribution whose density at $q^\star(x)$ is positive and bounded away from zero. This assumption is the width-side counterpart of the coverage-side's local strong convexity (Assumption 2.6).

**Assumption 2.3** (Realizability (optional strengthening))**.** $q^\star \in \mathscr{F}$ *and* $\|q^\star_K\|_{\mathscr{H}_K} \leq B_q$.

*Remark.* Used only to make $A_{\mathrm{appr}}(\lambda)$ explicit. Without it, Theorem 2.7 holds with $A_{\mathrm{appr}}(\lambda)$ abstract. With it, $A_{\mathrm{appr}}(\lambda) \leq \lambda B_q^2/2$.

**Lemma 2.3** (Regularization bias: $q^\star \to g^\star_\lambda$)**.** *Under Assumption 2.2,* $\mathbb{E}|g^\star_\lambda(X) - q^\star(X)| \leq \sqrt{2A_{\mathrm{appr}}(\lambda)/m_q}$. *Under Assumption 2.3,* $\mathbb{E}|g^\star_\lambda(X) - q^\star(X)| \leq B_q\sqrt{\lambda/m_q}$.

*Proof.* By optimality of $g_\lambda^\star$: $\mathscr{R}(g_\lambda^\star) - \mathscr{R}(q^\star) \le A_{\text{appr}}(\lambda)$. By Assumption 2.2: $\frac{m_q}{2}\mathbb{E}[(g_\lambda^\star(X) - q^\star(X))^2] \le A_{\text{appr}}(\lambda)$. Jensen's inequality gives the first bound. Under Assumption 2.3, $A_{\text{appr}}(\lambda) \le \frac{\lambda}{2}B_q^2$. □

**Lemma 2.4** (Sample transfer: $g_\lambda^\star \to \bar{g}_n$, rate $1/\lambda$)**.** *Under Assumptions 2.5–2.6,* $s_{n,\lambda} := \mathbb{E}|\bar{g}_n(X) - g_\lambda^\star(X)| \le C_{\text{tr}} L_0 \lambda^{-1}\sqrt{p_\Phi \log n / n}$.

*Proof. Step 1 (Evaluation bound).* Write $\bar{g}_n = \Phi^\top \bar{\beta}_n + \bar{g}_{n,K}$ and $g_\lambda^\star = \Phi^\top \beta^\star + g_{\lambda,K}^\star$. For any $x$,

$$|\bar{g}_n(x) - g_\lambda^\star(x)| \;\le\; |\Phi(x)|_2\,|\bar{\beta}_n - \beta^\star|_2 + \sqrt{K(x,x)}\,\|\bar{g}_{n,K} - g_{\lambda,K}^\star\|_{\mathscr{H}_K} \;\le\; L_0 \cdot d(\bar{z}_n, z_\lambda^\star),$$

where $\bar{z}_n := (\bar{\beta}_n, \bar{g}_{n,K})$, $z_\lambda^\star := (\beta^\star, g_{\lambda,K}^\star)$. Taking expectations:

$$s_{n,\lambda} \;\le\; L_0 \cdot \mathbb{E}\big[d(\bar{z}_n, z_\lambda^\star)\big]. \tag{2.12}$$

*Step 2 (Expectation bound for $d(\bar{z}_n, z_\lambda^\star)$).* This follows from Gibbs et al. [2023, Proposition 2] (Appendix A.4.2). Their argument proceeds through:

(a) *Peeling/shell decomposition*: Using dyadic shells $A_j = \{(\beta, g_K) : 2^{j-1} < \sqrt{\lambda n / p_\Phi} \cdot d(\beta, g_K) \le 2^j\}$, combined with Assumption 2.6 and Rademacher complexity bounds (Gibbs et al., 2023, Lemma 6), they establish $d(\bar{z}_n, z_\lambda^\star) = O_P(\sqrt{p_\Phi/(\lambda n)})$.

(b) *Truncated integral for the expectation*: Writing $\mathbb{E}[d] = \int_0^\infty \mathbb{P}(d > t)\,dt$ and splitting at the strong-convexity radius $\delta_M$:

- *Integral term* (0 to $\delta_M$): Using the tail bound from (a), this gives $O\big((1/\sqrt{\lambda})\sqrt{p_\Phi \log n / n}\big)$.
- *Tail term* (beyond $\delta_M$): This involves $\mathbb{E}[(\|\bar{g}_{n,K}\|_{\mathscr{H}_K} + \|g_{\lambda,K}^\star\|_{\mathscr{H}_K}) \cdot \mathbf{1}\{d > \delta_M\}]$. By Gibbs et al. [2023, Lemma 4], $\|\bar{g}_{n,K}\|_{\mathscr{H}_K} \le \sqrt{\mathbb{E}|S|/\lambda} = O(1/\sqrt{\lambda})$. Multiplied by $\mathbb{P}(d > \delta_M) = O(\sqrt{p_\Phi/(\lambda n)})$, this gives $O(1/\sqrt{\lambda} \cdot \sqrt{p_\Phi/(\lambda n)}) = O(\sqrt{p_\Phi/(\lambda^2 n)}) = O((1/\lambda)\sqrt{p_\Phi/n})$.

For small $\lambda$ the tail term dominates, and the combined rate is

$$\mathbb{E}\big[d(\bar{z}_n, z_\lambda^\star)\big] \;=\; O\left(\frac{1}{\lambda}\sqrt{\frac{p_\Phi \log n}{n}}\right). \tag{2.13}$$

Substituting (2.13) into (2.12) yields the claimed bound. □

*Critical note.* The rate is $1/\lambda$, not $1/\sqrt{\lambda}$. The $1/\sqrt{\lambda}$ is only the $O_P$ concentration rate; the expectation conversion adds an extra $1/\sqrt{\lambda}$ factor from the tail, giving $1/\lambda$ total. This distinction is crucial for the trade-off structure. On the coverage side (Corollary 2.10), the $1/\lambda$ is multiplied by $2\lambda$ from the coverage formula, causing exact cancellation and producing a $\lambda$-free middle term. On the width side, there is no such cancellation—the $1/\lambda$ appears directly.

**Lemma 2.5** (Augmentation perturbation: $\bar{g}_n \to \hat{q}$)**.** *Under the conditions of Lemma 2.4,* $a_n^{\text{aug}} := \mathbb{E}|\hat{q}(X) - \bar{g}_n(X)| \le C_a p_\Phi \log n/(\lambda(n+1))$.

*Proof. Step 1.* For *fixed* $\beta$, the augmented kernel fit $\hat{g}_\beta^{\text{aug}}$ and non-augmented kernel fit $\bar{g}_{n,\beta}$ differ by at most $\sup_x K(x,x)/(2\lambda(n+1))$ in sup-norm. This is a direct application of Gibbs et al. [2023, Lemma 1] (one-point stability of RKHS regression under bounded kernel).

*Step 2.* The change in $\beta$ between the augmented and non-augmented fits is controlled by discretizing the $\beta$-space via an $\varepsilon$-net and applying Gibbs et al. [2023, Lemma 2] (stability of the kernel fit under $\beta$-perturbation). The $\varepsilon$-net has covering number $\exp(C\, p_\Phi \log(1/(\lambda\varepsilon)))$.

*Step 3.* Combining Steps 1 and 2 via the same high-moment/covering argument used in Gibbs et al.'s Proposition 1 proof (their page 41, the calculation leading to $O(2^m(m/(\lambda n))^m)$), one obtains, for all $s \ge 0$ simultaneously,

$$|\hat{g}_{x,s}^{\text{aug}}(x) - \bar{g}_n(x)| \;\le\; \frac{C_a' p_\Phi \log n}{\lambda(n+1)}.$$

*Step 4.* Since this bound holds uniformly in $s$, the fixed-point sandwich lemma (Lemma 2.6) gives $|\hat{q}(x) - \bar{g}_n(x)| \le C_a' p_\Phi \log n/(\lambda(n+1))$, provided $\bar{g}_n(x)$ exceeds this bound (which holds for all reasonable configurations since $\bar{g}_n(x)$ estimates a positive quantile while the bound is $O(1/n)$). Taking expectations yields the claimed bound.

*Why strong convexity is not used here:* The empirical pinball loss is piecewise linear in the unpenalized $\beta$ direction, so the empirical objective $J_n$ cannot satisfy quadratic growth $J_n(g) - J_n(\bar{g}_n) \ge (\mu/2)\|g - \bar{g}_n\|^2$ in the $\beta$ direction. The covering/discretization approach of

Gibbs et al. [2023, Proposition 1] circumvents this obstacle entirely by separating the $\beta$ and $g_K$ components. □

**Lemma 2.6** (Fixed-point sandwich)**.** *If $|T(s)-c| \le a$ for all $s \ge 0$ and $c > a > 0$, then $q := \sup\{s \ge 0 : s \le T(s)\}$ satisfies $|q-c| \le a$.*

*Proof.* If $s < c-a$, then $T(s) \ge c-a > s$, so $q \ge c-a$. If $s > c+a$, then $T(s) \le c+a < s$, so $q \le c+a$. □

*Order comparison.* The augmentation term $O(p_\Phi \log n/(\lambda n))$ is strictly lower-order than the sample-transfer term $O(L_0/\lambda \cdot \sqrt{p_\Phi \log n/n})$.

**Theorem 2.7** (Fitted-width bound)**.** *Under Assumptions 2.1, 2.4, 2.5–2.6, and 2.2:*

$$\mathbb{E}\big[2\hat{q}(X)\big] \le 2q_\varepsilon + 2\sqrt{r(L)} + 2\sqrt{\frac{2A_{\mathrm{appr}}(\lambda)}{m_q}} + \frac{2C_{\mathrm{tr}}L_0}{\lambda}\sqrt{\frac{p_\Phi \log n}{n}} + \frac{2C_a p_\Phi \log n}{\lambda(n+1)}. \tag{2.14}$$

*Under Assumption 2.3:*

$$\mathbb{E}\big[2\hat{q}(X)\big] \le 2q_\varepsilon + 2\sqrt{r(L)} + 2B_q\sqrt{\frac{\lambda}{m_q}} + \frac{2C_{\mathrm{tr}}L_0}{\lambda}\sqrt{\frac{p_\Phi \log n}{n}} + \frac{2C_a p_\Phi \log n}{\lambda(n+1)}. \tag{2.15}$$

*Proof.* By the triangle inequality: $\hat{q}(X) \le q^\star(X) + |g^\star_\lambda(X) - q^\star(X)| + |\bar{g}_n(X) - g^\star_\lambda(X)| + |\hat{q}(X) - \bar{g}_n(X)|$. Taking expectations and substituting Theorem 2.1, Lemmas 2.3–2.5, then multiplying by 2. □

**Corollary 2.8** (Width-optimal $\lambda$)**.** *Under Assumption 2.3, balancing the $\sqrt{\lambda}$ and $1/\lambda$ terms:*

$$\lambda^\star_{\mathrm{width}} \asymp \left(\frac{C_{\mathrm{tr}}L_0\sqrt{m_q p_\Phi \log n/n}}{B_q}\right)^{2/3}. \tag{2.16}$$

*In particular, $\lambda^\star_{\mathrm{width}} \asymp (\log n/n)^{1/3}$ (treating all problem-dependent constants as $O(1)$). For this*

*thesis's experimental setting ($p_\Phi = 1$):*

$$\lambda^\star_{\text{width}} \asymp \left( \frac{C_{\text{tr}} L_0 \sqrt{m_q \log n/n}}{B_q} \right)^{2/3}.$$

*Proof.* Setting $a/(2\sqrt{\lambda}) = b/\lambda^2$ with $a = 2B_q/\sqrt{m_q}$ and $b = 2C_{\text{tr}} L_0 \sqrt{p_\Phi \log n/n}$ gives $\lambda^{3/2} = 2b/a$, hence $\lambda^\star = (2b/a)^{2/3}$. □

*Remark (comparison with coverage-side optimal $\lambda$).* The coverage-side Corollary 2.10 gives $\lambda^\star_{\text{cov}} \asymp (\rho_f \, p_\Phi \log n/(\kappa_f \, n\sqrt{M \, r(L)}))^{2/3}$. Both have the same $2/3$**-power structure** and the same $(\log n/n)^{1/3}$ scaling. The coefficients differ because:

- Width-side involves $B_q$ (RKHS norm of $q^\star$) and $m_q$ (density *lower* bound).
- Coverage-side involves $\kappa_f$ (shift function norm), $\rho_f$ (max-to-mean ratio), $M$ (density *upper* bound), and $r(L)$ (ICL error).

The ICL error $r(L)$ appears explicitly in $\lambda^\star_{\text{cov}}$ but only implicitly in $\lambda^\star_{\text{width}}$ (through $B_q$, which depends on the smoothness of $q^\star$, which in turn depends on $e(x) = f^\star(x) - f_\theta(x)$).

*Remark (consistency with experiments).* At fixed $\lambda$, the $\sqrt{\lambda}$ and $1/\lambda$ terms are constants and the fitted-width bound reduces to $2q_\varepsilon + 2\sqrt{r(L)} + \text{const}$, consistent with the empirical finding $\rho \approx 0.999$ between width and RMSE.

### 2.6.3 Coverage Pillar

**Assumption 2.4** (Density regularity)**.** *There exists a constant $M > 0$ such that, for almost all $x \in \mathscr{X}$,*

$$f_{S|X=x}(t) \;\le\; M, \qquad \forall t \in \big[\min(q_\varepsilon, q^\star(x)),\ \max(q_\varepsilon, q^\star(x))\big].$$

**Assumption 2.5** (RKHS regularity; Gibbs et al., 2023)**.** *The data $(X_1,Y_1),\ldots,(X_{n+1},Y_{n+1})$ are i.i.d. The reproducing kernel $K$ is uniformly bounded, $\sup_{x\in\mathscr{X}} K(x,x) < \infty$. The joint distribution of $(\Phi(X), S)$ has uniformly bounded first three moments (in the sense of Gibbs et al., 2023, Appendix Assumption 1). The conditional distribution of $S \mid X$ is continuous with a uniformly bounded density.*

**Assumption 2.6** (Local strong convexity; Gibbs et al., 2023)**.** *The population regularized pinball objective*

$$M_\infty(\beta, g_K) \;:=\; \mathbb{E}\big[\ell_\tau\big(\Phi(X)^\top\beta + g_K(X), S\big)\big] + \lambda\,\|g_K\|^2_{\mathscr{H}_K}$$

*is locally strongly convex near its minimizer (in the sense of Gibbs et al., 2023, Appendix Assumption 2).*

*Remark.* Assumptions 2.5–2.6 are satisfied in our experimental setting: Gaussian inputs with the RBF kernel yield a bounded kernel ($K(x,x) = 1$); all moment conditions hold under Gaussian noise; and local strong convexity follows from the positive conditional density of $|\varepsilon|$ under $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

**Theorem 2.9** (Coverage: conditional gap controlled by ICL error)**.** *Under Assumptions 2.1, 2.4, and 2.5, for any nonnegative $f = \Phi^\top\beta_f + f_K \in \mathscr{F}$ with $\mathbb{E}[f(X)] > 0$:*

$$\mathbb{P}_f\big(Y_{n+1} \in \hat{C}(X_{n+1})\big) \;\geq\; 1 - \alpha \;-\; \frac{2\sqrt{\lambda\, M\, r(L)}\;\|f_K\|_{\mathscr{H}_K}}{\mathbb{E}[f(X)]} \;-\; |R_n(f)|, \tag{2.17}$$

*where $|R_n(f)| \leq 2\lambda\,\psi_n(\lambda,\gamma)\,\|f_K\|_{\mathscr{H}_K}/\mathbb{E}[f(X)]$ is the empirical-to-population remainder and $\psi_n(\lambda,\gamma) := \mathbb{E}\|\hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}\|_{\mathscr{H}_K}$ is the sample-to-population transfer rate (pinned down quantitatively in Corollary 2.10).*

The proof establishes the chain $r(L) \to \Delta_\varepsilon \to \|g^\star_{\lambda,K}\| \to G^{\mathrm{pop}}_{\mathrm{cal}} \to G_{\mathrm{cal}}$ via a pinball-risk identity, RKHS norm control, and Cauchy–Schwarz.

*Proof of Theorem 2.9. Step 1 (Pinball-risk identity).* For fixed $x$, define the conditional pinball risk $R_x(q) := \mathbb{E}[\ell_\tau(q,S) \mid X = x]$ and the conditional CDF $F_x(t) := \mathbb{P}(S \leq t \mid X = x)$. The map $q \mapsto \ell_\tau(q,s)$ is convex with subgradient $F_x(q) - \tau$ at continuity points, and $F_x(q^\star(x)) = \tau$; integrating yields the pinball-risk excess identity

$$R_x(q) - R_x(q^\star(x)) \;=\; \int_{\min(q,q^\star(x))}^{\max(q,q^\star(x))} |F_x(t) - \tau|\,dt.$$

*Step 2 (ICL error controls the constant-vs-oracle regret).* Define $\Delta_\varepsilon := \mathbb{E}[\ell_\tau(q_\varepsilon, S)] - \mathbb{E}[\ell_\tau(q^\star(X), S)]$. Under Assumption 2.4, $|F_x(t) - \tau| \leq M|t - q^\star(x)|$ on the interval between $q_\varepsilon$ and

$q^\star(x)$, so Step 1 gives

$$R_x(q_\varepsilon) - R_x(q^\star(x)) \;\le\; \tfrac{M}{2}(q^\star(x) - q_\varepsilon)^2 \;\le\; \tfrac{M}{2} e(x)^2,$$

where the last inequality uses the pointwise bound (2.11) from the proof of Theorem 2.1. Taking expectations,

$$\Delta_\varepsilon \;\le\; \tfrac{M}{2}\, r(L). \tag{2.18}$$

(The distribution-free branch $\Delta_\varepsilon \le \sqrt{r(L)}$ follows from $|F_x(t) - \tau| \le 1$.)

*Step 3 (Regret controls the RKHS norm).* Let $g^\star_\lambda$ minimize the population regularized objective $J_\lambda(g) := \mathbb{E}[\ell_\tau(g(X), S)] + \frac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K}$ over $\mathscr{F}$. The constant function $x \mapsto q_\varepsilon$ lies in $\mathscr{F}$ with zero RKHS penalty, so by optimality of $g^\star_\lambda$,

$$\mathbb{E}[\ell_\tau(g^\star_\lambda(X), S)] + \tfrac{\lambda}{2}\|g^\star_{\lambda,K}\|^2 \;\le\; \mathbb{E}[\ell_\tau(q_\varepsilon, S)].$$

Since $q^\star$ minimizes the unregularized pinball risk, $\mathbb{E}[\ell_\tau(q^\star(X), S)] \le \mathbb{E}[\ell_\tau(g^\star_\lambda(X), S)]$; substituting,

$$\tfrac{\lambda}{2}\,\|g^\star_{\lambda,K}\|^2_{\mathscr{H}_K} \;\le\; \Delta_\varepsilon \;\le\; \tfrac{M}{2}\, r(L), \qquad \text{hence} \qquad \|g^\star_{\lambda,K}\|_{\mathscr{H}_K} \;\le\; \sqrt{\tfrac{M\, r(L)}{\lambda}}. \tag{2.19}$$

*Step 4 (Population GAP upper bound).* Define $G^{\mathrm{pop}}_{\mathrm{cal}}(\lambda, \gamma, f) := 2\lambda\, \langle g^\star_{\lambda,K}, f_K\rangle_{\mathscr{H}_K} / \mathbb{E}[f(X)]$. By Cauchy–Schwarz,

$$|G^{\mathrm{pop}}_{\mathrm{cal}}| \;\le\; \frac{2\lambda\, \|g^\star_{\lambda,K}\|_{\mathscr{H}_K}\, \|f_K\|_{\mathscr{H}_K}}{\mathbb{E}[f(X)]} \;\le\; \frac{2\sqrt{\lambda\, M\, r(L)}\, \|f_K\|_{\mathscr{H}_K}}{\mathbb{E}[f(X)]}.$$

*Step 5 (Sample-level GAP).* The CondConf guarantee (2.9) involves the empirical minimizer $\hat{g}_{S_{n+1}}$, not $g^\star_\lambda$. Adding and subtracting,

$$G_{\mathrm{cal}}(n, \lambda, \gamma, f) = G^{\mathrm{pop}}_{\mathrm{cal}}(\lambda, \gamma, f) + R_n(f), \qquad R_n(f) := \frac{2\lambda\, \mathbb{E}[\langle \hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}, f_K\rangle_{\mathscr{H}_K}]}{\mathbb{E}[f(X)]}.$$

By Cauchy–Schwarz and the sample-to-population transfer rate $\psi_n(\lambda,\gamma) := \mathbb{E}\|\hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}\|_{\mathscr{H}_K}$, we have $|R_n(f)| \le 2\lambda\ \psi_n(\lambda,\gamma)\,\|f_K\|/\mathbb{E}[f(X)]$. Substituting Step 4 and this remainder bound into the one-sided CondConf guarantee $\mathbb{P}_f(Y_{n+1} \in \hat{C}(X_{n+1})) \ge 1-\alpha-G_{\text{cal}}$ yields (2.17). □

**Sanity checks.** (i) If $f_K = 0$ (e.g., $f \equiv 1$), both $G^{\text{pop}}_{\text{cal}}$ and $R_n$ vanish, recovering exact marginal coverage. (ii) To obtain the $\sqrt{r(L)}$ scaling (rather than $r(L)^{1/4}$), the density upper bound $M$ is essential. (iii) For Gaussian noise, $M = O(1/\sigma)$, so the leading gap term is $O(\sqrt{\lambda\, r(L)/\sigma})$.

**Corollary 2.10** (Explicit form and $\lambda$-trade-off)**.** *Let* $\kappa_f := \|f_K\|_{\mathscr{H}_K}/\mathbb{E}[f(X)]$, $\rho_f := \mathbb{E}[\max_{1\le i\le n+1}|f(X_i)|]/\mathbb{E}[f(X)]$, *and* $p_\Phi := \dim(\Phi)$ *(so that* $p_\Phi = 1$ *in all experiments of this thesis, since* $\Phi(x) = 1$*; in particular* $p_\Phi$ *is not the input dimension d). Under Assumptions 2.1–2.6, the abstract remainder in Theorem 2.9 admits the explicit bound*

$$|R_n(f)| \ \le\ C_R\,\kappa_f\sqrt{\frac{p_\Phi \log n}{n}}, \tag{2.20}$$

*where* $C_R$ *depends only on the kernel bound, moment constants, density upper bound M, and the local strong convexity constant from Assumption 2.6. Combining with the upper-bound counterpart of Theorem 2.9 (which adds an interpolation-error term* $\varepsilon_{\text{int}}$ *from Gibbs et al. [2023, Proposition 1]), the conditional coverage gap satisfies the two-sided bound*

$$\left|\mathbb{P}_f\big(Y_{n+1}\in\hat{C}(X_{n+1})\big) - (1-\alpha)\right| \ \le\ 2\kappa_f\sqrt{\lambda\, M\, r(L)} + C_R\,\kappa_f\sqrt{\frac{p_\Phi\log n}{n}} + C_I\,\rho_f\,\frac{p_\Phi\log n}{\lambda\, n}, \tag{2.21}$$

*where* $C_I$ *is the constant from Gibbs et al. [2023, Proposition 1].*

*Remark (*$\lambda$*-trade-off).* The three terms in (2.21) isolate distinct sources of error: (i) a regularization cost ($\sqrt{\lambda}$), arising from the population RKHS norm of the conditional quantile; (ii) a statistical estimation term ($\lambda$*-independent*), arising from finite-sample fluctuation of the augmented quantile estimator; and (iii) an interpolation error ($1/\lambda$), arising from boundary effects when fitting the regularized quantile regression. The first term increases in $\lambda$ while the third decreases, producing a clear trade-off.

*Remark (optimal $\lambda$).* Since the middle term of (2.21) is $\lambda$-independent, the optimal $\lambda$ balances the first and third:

$$\lambda_f^\star \asymp \left( \frac{\rho_f \, p_\Phi \log n}{\kappa_f \, n \sqrt{M \, r(L)}} \right)^{2/3}. \tag{2.22}$$

This expression shows that $\lambda_f^\star$ depends on $r(L)$: when $r(L)$ is large (weak model or noisy task), the optimal $\lambda$ is smaller; when $r(L)$ is small (strong model or easy task), the optimal $\lambda$ is larger. This is consistent with the empirical observation that interval efficiency is dominated by base model quality, since at fixed $\lambda$ the $\sqrt{\lambda}$ and $1/\lambda$ terms are constants and only $r(L)$ varies across configurations.

*Proof of Corollary 2.10.* The proof proceeds in six steps: (1) start from the two-sided RKHS bound of Gibbs et al. [2023, eq. (3.4)]; (2) decompose into population and empirical parts; (3) reuse the population RKHS norm bound from Step 3 of the proof of Theorem 2.9; (4) bound the empirical remainder via Gibbs et al. [2023, Proposition 2], observing that the explicit factor of $\lambda$ cancels; (5) bound the interpolation error via Gibbs et al. [2023, Proposition 1]; (6) combine and minimize.

*Step 1: Two-sided bound from Gibbs et al.* The RKHS specialization of Gibbs et al. [2023, Theorem 3, eq. (3.4)] gives, under Assumption 2.5,

$$\mathbb{P}_f\big(Y_{n+1} \in \hat{C}(X_{n+1})\big) \;\geq\; 1 - \alpha - \frac{2\lambda \, \mathbb{E}[\langle \hat{g}_{S_{n+1},K}, f_K \rangle_{\mathscr{H}_K}]}{\mathbb{E}[f(X)]},$$

$$\mathbb{P}_f\big(Y_{n+1} \in \hat{C}(X_{n+1})\big) \;\leq\; 1 - \alpha - \frac{2\lambda \, \mathbb{E}[\langle \hat{g}_{S_{n+1},K}, f_K \rangle_{\mathscr{H}_K}]}{\mathbb{E}[f(X)]} + \frac{|\varepsilon_{\text{int}}|}{\mathbb{E}[f(X)]},$$

where $\varepsilon_{\text{int}}$ is the interpolation error of Gibbs et al. [2023, Proposition 1]. Taking absolute values and combining,

$$\left|\mathbb{P}_f(Y_{n+1} \in \hat{C}(X_{n+1})) - (1-\alpha)\right| \;\leq\; \frac{2\lambda \left|\mathbb{E}[\langle \hat{g}_{S_{n+1},K}, f_K \rangle_{\mathscr{H}_K}]\right|}{\mathbb{E}[f(X)]} \;+\; \frac{|\varepsilon_{\text{int}}|}{\mathbb{E}[f(X)]}. \tag{2.23}$$

*Step 2: Population/empirical decomposition.* Write $\mathbb{E}[\langle \hat{g}_{S_{n+1},K}, f_K \rangle_{\mathscr{H}_K}] = \langle g^\star_{\lambda,K}, f_K \rangle_{\mathscr{H}_K} + \mathbb{E}[\langle \hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}, f_K \rangle_{\mathscr{H}_K}]$, where $g^\star_{\lambda,K}$ is the RKHS component of the population minimizer of the regularized pinball objective. By Cauchy–Schwarz applied to each inner product, (2.23) is bounded

by

$$\left|\mathbb{P}_f - (1-\alpha)\right| \;\le\; 2\lambda\,\kappa_f\,\|g^\star_{\lambda,K}\|_{\mathscr{H}_K} \;+\; 2\lambda\,\kappa_f\,\mathbb{E}\big\|\hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}\big\|_{\mathscr{H}_K} \;+\; \frac{|\varepsilon_{\text{int}}|}{\mathbb{E}[f(X)]}. \tag{2.24}$$

*Step 3: Population RKHS norm bound (the $\sqrt{\lambda}$ term).* From (2.19), under Assumption 2.4, $\|g^\star_{\lambda,K}\|_{\mathscr{H}_K} \le \sqrt{M\,r(L)/\lambda}$, so the first term in (2.24) is bounded by $2\lambda\,\kappa_f\,\|g^\star_{\lambda,K}\|_{\mathscr{H}_K} \le 2\kappa_f\sqrt{\lambda\,M\,r(L)}$.

*Step 4: Empirical remainder ($\lambda$ cancels exactly).* Under Assumptions 2.5–2.6, Gibbs et al. [2023, Proposition 2] establishes

$$\mathbb{E}\big\|\hat{g}_{S_{n+1},K} - g^\star_{\lambda,K}\big\|_{\mathscr{H}_K} \;=\; O\!\left(\frac{1}{\lambda}\sqrt{\frac{p_\Phi \log n}{n}}\right).$$

Their proof (Appendix A.4.2 of the arXiv version) combines a truncated integral representation of the $\mathscr{H}_K$-norm with a peeling argument built on local strong convexity (Assumption 2.6), yielding the rate above with the explicit $1/\lambda$ pre-factor. The $p_\Phi \log n$ dependence arises from standard concentration for the finite-dimensional $\Phi$-component of the estimator combined with empirical-process control over the RKHS component. Multiplying the rate by $2\lambda\,\kappa_f$, the explicit factor of $\lambda$ cancels:

$$2\lambda\,\kappa_f \cdot O\!\left(\frac{1}{\lambda}\sqrt{\frac{p_\Phi \log n}{n}}\right) \;=\; C_R\,\kappa_f\sqrt{\frac{p_\Phi \log n}{n}},$$

which establishes (2.20).

*Step 5: Interpolation error (the $1/\lambda$ term).* By Gibbs et al. [2023, Proposition 1], under Assumption 2.5,

$$\frac{|\varepsilon_{\text{int}}|}{\mathbb{E}[|f(X)|]} \;\le\; O\!\left(\frac{p_\Phi \log n}{\lambda\, n}\right)\frac{\mathbb{E}[\max_{1\le i\le n+1}|f(X_i)|]}{\mathbb{E}[|f(X)|]}.$$

For nonnegative $f$, $|f| = f$ and the shape factor is precisely $\rho_f$; therefore $|\varepsilon_{\text{int}}|/\mathbb{E}[f(X)] \le C_I\,\rho_f\,p_\Phi \log n/(\lambda n)$.

*Step 6: Combine and minimize.* Substituting Steps 3–5 into (2.24) yields the two-sided bound (2.21). Minimizing $a\sqrt{\lambda} + b/\lambda$ with $a = 2\kappa_f\sqrt{M\,r(L)}$ and $b = C_I\,\rho_f\,p_\Phi \log n/n$ via the

| Paper | Task class | $r(L)$ | Width excess | Cov. gap |
|---|---|---|---|---|
| *Nonparametric regression (minimax-optimal)* | | | | |
| Ching et al. '26 | Hölder | $L^{-\frac{2\alpha}{2\alpha+d}}$ | $L^{-\frac{\alpha}{2\alpha+d}}$ | $\sqrt{\lambda M}\,L^{-\frac{\alpha}{2\alpha+d}}\,\kappa_f$ |
| Kim et al. '24 | Besov | $N^{-\frac{2\alpha}{d}}+\frac{N\log N}{L}$ | $\sqrt{\text{r.h.s.}}$ | $\sqrt{\lambda M\cdot\text{r.h.s.}}\,\kappa_f$ |
| *Linear regression (explicit risk formulas)* | | | | |
| Wu et al. '24 | Linear (Gaussian) | $\min\{1, s/L\}$ | $\sqrt{s/L}$ | $\sqrt{\lambda M s/L}\,\kappa_f$ |
| Zhang et al. '24 | Linear (trained) | $(d+1)\,\mathrm{tr}\Lambda/L$ | $\sqrt{(d+1)\,\mathrm{tr}\Lambda/L}$ | $\sqrt{\lambda M(d+1)\,\mathrm{tr}\Lambda/L}\,\kappa_f$ |
| Chen et al. '24 | Multi-task lin. | $\sigma^2 d/L$ | $\sigma\sqrt{d/L}$ | $\sqrt{\lambda M\sigma^2 d/L}\,\kappa_f$ |
| *Algorithm selection* | | | | |
| Bai et al. '23 | Mixed-noise | $(\log K/L)^{1/3}$ | $(\log K/L)^{1/6}$ | $\sqrt{\lambda M}(\log K/L)^{1/6}\kappa_f$ |
| *Empirical / mechanistic (no closed-form $r(L)$)* | | | | |
| Garg et al. '22 | Lin., 2NN, NDT | — | — | — |
| Akyürek et al. '23 | Linear (GD) | — | — | — |
| von Oswald et al. '23 | Linear (GD) | — | — | — |

Table 2.1: ICL error-rate survey with plug-in width rates from Theorem 2.1 and coverage rates from Theorem 2.9 (made explicit by Corollary 2.10). All rates suppress $O(\cdot)$ constants. $\kappa_f := \|f_K\|_{\mathscr{H}_K}/\mathbb{E}[f(X)]$. Kim et al.'s full rate includes a pretraining term $N^2\log N/T$ omitted for space. "—" = no explicit rate available. Note: the symbol $N$ has a row-specific meaning in this table (feature dim in Kim et al., training context length in Zhang et al., etc.); it is distinct from the $N$ used elsewhere in this thesis to denote the number of evaluation episodes.

first-order condition $a/(2\sqrt{\lambda}) = b/\lambda^2$ gives $\lambda^{3/2} = 2b/a$, hence $\lambda_f^\star \asymp (b/a)^{2/3}$, which is (2.22).

*Convention note.* This thesis uses the penalty $\frac{\lambda}{2}\|g_K\|^2_{\mathscr{H}_K}$ in the augmented quantile regression, while Gibbs et al. [2023] uses $\lambda\|g_K\|^2_{\mathscr{H}_K}$. This affects only the constants $C_R$ and $C_I$ by factors of order one, not the rates; the final bound absorbs these into the constants. □

## 2.7 ICL Error-Rate Survey and Plug-in Guide

Theorems 2.1 and 2.9 hold for any ICL error bound $r(L)$. In this subsection we provide a comprehensive survey of all known explicit $r(L)$ bounds from the literature, show how each plugs into our width and coverage results, and discuss coverage gaps. Table 2.1 gives a unified summary; we expand on each entry below.

**1. Ching et al. [2026]: minimax-optimal Hölder nonparametric ICL.** *Architecture*: transformer with $\Theta(\log L)$ parameters, realizable via 2 ReLU attention heads implementing linear attention. *Task class*: $\alpha$-Hölder smooth regression in $d$ dimensions. *Rate*: the minimax-optimal excess prediction risk in squared loss for $L$ in-context examples:

$$\mathbb{E}[R(\hat{f}_\Gamma)] - \sigma^2 \;\le\; C L^{-\frac{2\alpha}{2\alpha+d}},$$

provided $\Gamma \ge C L^{\frac{2\alpha}{2\alpha+d}} \log^3(eL)$ pretraining sequences. The dominant term predicts severe degradation as $d$ increases for fixed $L$, which we verify empirically in Section 6. Their construction uses linear attention realizable with two ReLU heads (see the multi-head extension in Section C). *Plug-in*: direct—this is a squared-loss excess risk, exactly $r(L)$ in our framework.

**2. Kim et al. [2024]: Besov ICL with three-term decomposition.** *Architecture*: deep neural network feature map ($N$-dimensional output) followed by one linear attention layer. *Task class*: Besov spaces, anisotropic Besov, piecewise $\gamma$-smooth sequences. *Rate*: a three-term decomposition of ICL risk:

$$r(L) \;\lesssim\; \underbrace{N^{-2\alpha/d}}_{\text{approx.}} + \underbrace{\frac{N\log N}{L}}_{\text{in-context gen.}} + \underbrace{\frac{N^2\log N}{T}}_{\text{pretraining gen.}},$$

where choosing $N \asymp L^{d/(2\alpha+d)}$ yields the minimax rate when $T$ is large. This decomposition explicitly separates the roles of model capacity $N$, context length $L$, and pretraining scale $T$, revealing a bias–variance tradeoff that formalizes the architecture scaling behavior observed in Section 4. *Plug-in*: direct, optionally keeping the two-parameter form $r(L,T)$.

**3. Wu et al. [2024]: explicit linear regression risk.** *Architecture*: single-layer linear attention (one-step GD with matrix stepsize). *Task class*: linear regression with Gaussian prior; covariance spectrum $\Lambda$ with eigenvalues $\lambda_i$. *Rate*: exact average-risk formulas:

$$\mathscr{L}(h;X) - \sigma^2 \;\asymp\; \sum_i \psi^2 \min\{\mu_i/L,\; \lambda_i\},$$

where $\mu_i \asymp \sigma^2/(\psi^2 L)$ under isotropic conditions. Concrete rates: $r(L) \asymp \min\{1, s/L\}$ for rank-$s$ uniform spectrum, and $r(L) \asymp (\log L)/L$ for exponential decay. *Plug-in*: direct for linear tasks; specialize to the assumed spectrum.

**4. Zhang et al. [2024]: trained linear self-attention.** *Architecture*: one-layer transformer with linear self-attention, trained by gradient flow. *Task class*: linear predictors with Gaussian covariates (covariance $\Lambda$, condition number $\kappa$). *Rate*:

$$\mathbb{E}(\hat{y}_{\text{query}} - \langle w, x_{\text{query}} \rangle)^2 \;\leq\; \frac{(d+1)\operatorname{tr}(\Lambda)}{L} + \frac{(1+2d+d^2\kappa)\operatorname{tr}(\Lambda)}{N^2},$$

where $L$ is the test context length and $N$ the training context length, showing $O(1/L)$ test scaling plus $O(1/N^2)$ training-length dependence. *Plug-in*: direct as squared-loss $r(L)$, with dimension and covariance constants.

**5. Chen et al. [2024]: multi-head softmax with head separation.** *Architecture*: one-layer multi-head softmax attention, trained by gradient flow. *Task class*: multi-task linear regression with structured task decomposition. *Rate*: $r(L) \approx \sigma^2 d/L$, with an explicit separation: the multi-head solution achieves $\approx \sigma^2 d/L$ while the single-head solution has a strictly worse constant. Their bounds include information-theoretic lower bounds confirming the multi-head advantage. *Plug-in*: direct; the head/task separation factor is relevant for our architecture scaling experiments.

**6. Bai et al. [2023]: algorithm selection with Bayes-gap.** *Architecture*: multi-layer transformer with normalized ReLU attention; explicit constructions for least squares, ridge, Lasso, GLMs, and 2-layer NN GD. *Task class*: mixed-noise linear regression with $K$ noise levels; algorithm selection across tasks. *Rate*: $r(L) = O((\log K/L)^{1/3})$ above the Bayes benchmark, plus an ERM pretraining generalization bound with a $d\sigma^2/N$ term. *Plug-in*: direct for mixed-noise tasks.

**7. Garg et al. [2022]: empirical ICL scaling.** *Architecture*: standard GPT-2 style transformer. *Task class*: linear regression, sparse linear, 2-layer neural networks, decision trees. *Rate*: no explicit

$r(L)$ theorem; demonstrates empirical ICL capability across function classes. *Plug-in*: not directly; serves as the canonical empirical reference.

**8. Akyurek et al. [2023]: GD/Ridge mechanisms.** *Architecture*: multi-layer transformer. *Task class*: linear models. *Rate*: no explicit $r(L)$; proves transformers can implement GD and ridge updates using $O(d)$ hidden size for one GD step and $O(d^2)$ for ridge. *Plug-in*: not directly; provides mechanistic understanding.

**9. von Oswald et al. [2023]: GD in weight space.** *Architecture*: stacked linear self-attention layers. *Task class*: linear models. *Rate*: no explicit $r(L)$; shows one linear self-attention layer = one GD step, deeper stacks = more steps. *Plug-in*: not directly; foundational mechanistic insight motivating depth-scaling experiments.

**Task coverage gaps.** For the nonlinear task families studied in this paper—quadratic regression, two-layer neural networks, and decision trees—explicit theoretical $r(L)$ bounds remain largely unavailable. Obtaining minimax-optimal ICL rates for these broader function classes is an important open problem.

The table reveals the key experimental predictions:

- **Dimension $d$:** The Hölder exponent $\alpha/(2\alpha+d)$ becomes very small at high $d$, predicting the sharp width increase at $d = 100$ (Section 6).
- **Noise $\sigma$:** The noise floor $q_\varepsilon = \sigma z_{1-\alpha/2}$ linearly raises the irreducible width baseline; for Gaussian noise $M = O(1/\sigma)$, so the coverage gap scales as $O(\sqrt{\lambda\, r(L)/\sigma})$.
- **Model capacity $N$:** Under the Besov rate, increasing $N$ reduces the approximation term $N^{-2\alpha/d}$ but inflates $N \log N/L$, formalizing the capacity tradeoff in Section 4.

**Remark on architectural assumptions.** Both the Hölder rate [Ching et al., 2026] and the Besov rate [Kim et al., 2024] are derived under simplified transformer architectures: Ching et al. use single-head linear attention, while Kim et al. use a deep neural network followed by one linear

attention layer. Our experiments employ standard GPT-2 style transformers with multi-head softmax attention and ReLU feed-forward layers. In Appendix C, we show that Ching's minimax-optimal rate can be rigorously extended to the multi-head ReLU-attention transformer class of Song et al. [2024] via a block-wise exact embedding argument (Theorem C.1). A fully rigorous extension to standard softmax attention and to the broader range of task families (quadratic, neural network, decision tree) studied in this paper remains an open problem and is beyond the scope of the present work.

**Algorithm 1** SpeedCP Conformal Prediction for ICL Regression

**Require:** Trained ICL model $f_\theta$; evaluation episodes $\{(x_i^{(1:L)}, y_i^{(1:L)}, x_i^{(L+1)}, y_i^{(L+1)})\}_{i=1}^N$; coverage level $1-\alpha$; kernel $K$ with bandwidth $\gamma$; regularization $\lambda$; basis $\Phi$

**Ensure:** Prediction intervals $\hat{C}(x_i^{(L+1)})$ for each test point

**// Data generation and prediction**
**for** each episode $i = 1, \ldots, N$ **do**
  $\hat{y}_i \leftarrow f_\theta(x_i^{(1:L)}, y_i^{(1:L)}, x_i^{(L+1)})$ ▷ ICL forward pass
  $S_i \leftarrow |y_i^{(L+1)} - \hat{y}_i|$ ▷ Absolute residual score
**end for**
Partition $\{1, \ldots, N\}$ into calibration set $\mathscr{I}_{\text{cal}}$ ($|\mathscr{I}_{\text{cal}}| = n$) and test set $\mathscr{I}_{\text{test}}$

**// Stage 1: $\lambda$-path (calibration only, run once)**
Compute kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, $\mathbf{K}_{ij} = K(x_i^{(L+1)}, x_j^{(L+1)})$ for $i, j \in \mathscr{I}_{\text{cal}}$
Compute basis matrix $\blacksquare \in \mathbb{R}^{n \times p_\Phi}$, $\blacksquare_i = \Phi(x_i^{(L+1)})$ for $i \in \mathscr{I}_{\text{cal}}$
Trace the regularization path from $\lambda = \infty$ down to target $\hat{\lambda}$:
  $\{v^*(\lambda), \beta^*(\lambda)\} \leftarrow \textsc{LambdaPath}(\{S_i\}_{\mathscr{I}_{\text{cal}}}, \blacksquare, \mathbf{K}, \alpha)$ ▷ Piecewise-affine path
Extract dual solution $(v^*, \beta^*)$ at $\lambda = \hat{\lambda}$ and active sets $\mathscr{E}, \mathscr{L}, \mathscr{R}$

**// Stage 2: $S$-path (per test point)**
**for** each test point $j \in \mathscr{I}_{\text{test}}$ **do**
  Compute cross-kernel vector $\mathbf{k}_j \in \mathbb{R}^n$, $(\mathbf{k}_j)_i = K(x_j^{(L+1)}, x_i^{(L+1)})$ for $i \in \mathscr{I}_{\text{cal}}$
  Augment: $\tilde{\mathbf{K}} \leftarrow \begin{pmatrix} \mathbf{K} & \mathbf{k}_j \\ \mathbf{k}_j^\top & K(x_j, x_j) \end{pmatrix}$, $\tilde{\blacksquare} \leftarrow \begin{pmatrix} \blacksquare \\ \Phi(x_j)^\top \end{pmatrix}$
  Trace the $S$-path with $(v^*, \beta^*)$ as warm start:
    $S^* \leftarrow \textsc{SPath}(\{S_i\}_{\mathscr{I}_{\text{cal}}}, \tilde{\blacksquare}, \tilde{\mathbf{K}}, \hat{\lambda}, \alpha, v^*, \beta^*)$
    ▷ Terminate when $v_{n+1}(S^*) \geq 1 - \alpha$
  $q_j \leftarrow \max(0, S^*)$ ▷ Conformal half-width
  $\hat{C}(x_j^{(L+1)}) \leftarrow [\hat{y}_j - q_j,\ \hat{y}_j + q_j]$
**end for**

# CHAPTER 3

# Experimental Setup

## 3.1 Transformer Architectures

We train GPT-2-style decoder-only transformers on diverse regression task distributions. All models use RMSNorm, a feed-forward hidden dimension of $4\times$ the model width, and are trained for 500,000 steps with a batch size of 64 and a learning rate of $10^{-4}$. To study architecture scaling, we define three families of models (Table 3.1).

## 3.2 Regression Tasks and Noise Levels

To study the interaction between task complexity and conformal prediction, we fix the architecture to a moderately large model (width 256, depth 12) and vary both the regression function family and the noise standard deviation $\sigma \in \{0.1, 0.25, 0.5, 1.0\}$. The four task families are:

1. **Noisy Linear Regression (NLR)**: $y = w^\top x + \varepsilon$, where $w$ is sampled uniformly and normalized (S52–S55; context length up to $L = 80$).
2. **Noisy Quadratic Regression (NQR)**: $y = (w^\top x)^2 + \varepsilon$ (S56–S59; context length up to $L = 200$).
3. **Two-layer Neural Network (2NN)**: $y = v^\top \mathrm{ReLU}(Wx) + \varepsilon$ with hidden layer size 4 (S60–S63; context length up to $L = 200$).
4. **Decision Tree (NDT)**: $y = T(x) + \varepsilon$ where $T$ is a random decision tree of depth 4 (S64–S67; context length up to $L = 200$).

In all cases, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and the input dimension grows during training via a curriculum from $d = 10$ to $d = 40$.

| Family | Models | Width | Depth | Description |
|---|---|---|---|---|
| Width scaling | S13 | 32 | 6 | Narrowest |
| | S14 | 64 | 6 | Baseline width |
| | S15 | 128 | 6 | 2× width |
| | S16 | 256 | 6 | 4× width |
| Depth scaling | S17 | 64 | 2 | Shallowest |
| | S18 | 64 | 4 | Moderate depth |
| | S19 | 64 | 8 | 2× depth |
| | S20 | 64 | 12 | Deepest |
| Standard presets | S21 | 32 | 4 | GPT-2 tiny |
| | S22 | 64 | 8 | GPT-2 small |
| | S23 | 128 | 12 | GPT-2 medium |
| | S24 | 256 | 12 | GPT-2 large |

Table 3.1: Model architectures used in the architecture scaling study (S13–S24). All models are evaluated on the noisy linear regression task with $\sigma = 0.1$.

## 3.3 Conformal Evaluation Protocol

For each model and ICL context length $L$, we generate $N = 1{,}600$ evaluation episodes and split them evenly into $n = 800$ calibration points and 800 test points. SpeedCP is applied with an RBF kernel using deterministic thresholding (non-randomized). The SpeedCP hyperparameters—the kernel bandwidth $\gamma$ and the regularization strength $\lambda$—are selected via grid search on a held-out validation split to balance conditional calibration error and prediction-set size, consistent with the $\lambda$-trade-off in Corollary 2.10, and then fixed across all models within each study. Two hyperparameter regimes result from this procedure: $(\gamma = 0.0012, \lambda = 4.5)$ for the architecture study (S13–S24), and $(\gamma = 0.05, \lambda = 5.0)$ for the task–noise study (S52–S67). In both cases, the basis function is intercept-only ($\Phi(x) = 1$) and the kernel features are derived from the query input $x_{L+1}$. We report empirical coverage, average interval width, and RMSE at each context length.

# CHAPTER 4

# Results I: Architecture Scaling Effects

In this section, we fix the regression task to noisy linear regression with $\sigma = 0.1$ and examine how transformer architecture affects both point prediction quality and conformal interval efficiency across 12 model variants (S13–S24).

## 4.1 Point Prediction Quality

Figure 4.1 reveals a striking diversity in ICL performance across architectures. At $L = 1$, all models produce RMSE values near 1.0, consistent with the prior predictive error when the model has essentially no information about the task-specific parameters. As the context length increases, the models diverge into three qualitative regimes:

**Strong learners** (S16, S23, S24): These models achieve RMSE below 0.22 by $L = 80$, approaching the Bayes-optimal error floor set by the noise level $\sigma = 0.1$. S24 (GPT-2 large) achieves the lowest RMSE of 0.147, followed closely by S23 (GPT-2 medium) at 0.166 and S16 (width 256, depth 6) at 0.217. The near-convergence of S23 and S24 suggests diminishing returns from further capacity increases in this task regime.

**Moderate learners** (S14, S19, S20, S22): These models reach RMSE in the range 0.29–0.44 at $L = 80$. The fact that S19 (width 64, depth 8) outperforms S20 (width 64, depth 12) is notable—it suggests that depth alone is not sufficient, and that the width–depth ratio plays a critical role. Similarly, S14 (width 64, depth 6) outperforms both S15 (width 128, depth 6) and S20, indicating a non-trivial interaction between architectural dimensions.

**Weak or non-learners** (S13, S15, S17, S18, S21): These models exhibit RMSE above 0.65 at $L = 80$, with S21 (GPT-2 tiny) being the most extreme outlier at 0.987—essentially no improvement from the prior. The flat RMSE trajectory of S21 indicates that this architecture lacks

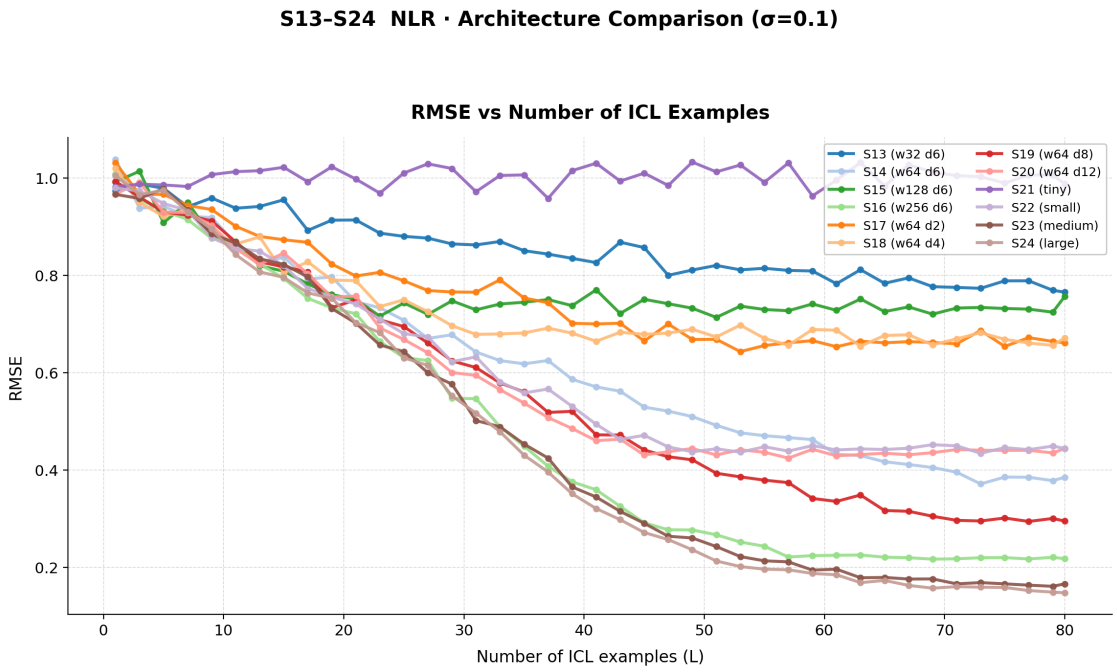



Figure 4.1: Root mean squared error (RMSE) as a function of the number of in-context examples $L$ for all 12 architectures (S13–S24) on the noisy linear regression task ($\sigma = 0.1$). All models start near RMSE $\approx 1.0$ at $L = 1$ (close to the prior predictive error), but diverge dramatically as more demonstrations are provided. Larger and deeper models achieve substantially lower RMSE at $L = 80$.

the capacity to implement the implicit algorithm required for linear regression ICL, consistent with prior findings that a minimum model capacity threshold exists for ICL emergence [Garg et al., 2022].

## 4.2 Conformal Interval Behavior

Figure 4.2 presents the SpeedCP interval width and empirical coverage as functions of context length. The coverage panel confirms that all 12 models maintain empirical coverage within the range 94.4%–95.2% on average, closely tracking the nominal 95% target. This stability validates the theoretical guarantees of the CondConf/SpeedCP framework: even as the underlying model quality varies dramatically across architectures, the conformal calibration mechanism successfully adapts the interval width to maintain the prescribed coverage level.

The interval width panel reveals the practical consequence: while coverage is uniformly

| Model | Architecture | RMSE ($L=1$) | RMSE ($L=80$) | Width drop (%) |
|---|---|---|---|---|
| S13 | w32 d6 | 0.973 | 0.765 | 19.5 |
| S14 | w64 d6 | 1.038 | 0.385 | 60.4 |
| S15 | w128 d6 | 0.992 | 0.756 | 25.1 |
| S16 | w256 d6 | 1.008 | 0.217 | 76.4 |
| S17 | w64 d2 | 1.031 | 0.661 | 28.7 |
| S18 | w64 d4 | 1.020 | 0.671 | 31.9 |
| S19 | w64 d8 | 0.993 | 0.295 | 67.9 |
| S20 | w64 d12 | 0.967 | 0.444 | 53.8 |
| S21 | GPT-2 tiny | 0.981 | 0.987 | $-3.2$ |
| S22 | GPT-2 small | 0.979 | 0.444 | 51.2 |
| S23 | GPT-2 medium | 0.966 | 0.166 | 82.4 |
| S24 | GPT-2 large | 1.005 | 0.147 | 84.0 |

Table 4.1: Summary of RMSE and SpeedCP interval width reduction for the architecture study. Width drop measures the percentage reduction in average SpeedCP interval width from $L = 1$ to $L = 80$.

maintained, the *price* of coverage differs enormously across architectures. At $L = 1$, all models produce similarly wide intervals (3.6–3.8), reflecting the uniformly poor predictions at minimal context. As $L$ increases, the width trajectories diverge in direct correspondence with RMSE trajectories. We quantify this through the *width drop* metric—the percentage reduction in average interval width from $L = 1$ to $L = 80$ (Table 4.1).

**Width scaling.** Among the width-scaling variants (S13–S16), increasing model width from 32 to 256 at fixed depth 6 improves the width drop from 19.5% to 76.4%. However, the relationship is non-monotone: S15 (width 128) achieves only 25.1% width drop, substantially worse than S14 (width 64) at 60.4%. This anomaly, also reflected in the RMSE curves, suggests that width 128 with depth 6 may represent an unfavorable capacity allocation for this task. The striking success of S16 (width 256) with a 76.4% width drop—reducing the average interval from 3.72 to 0.88—demonstrates that sufficient model width is the single most impactful architectural choice for ICL interval efficiency.

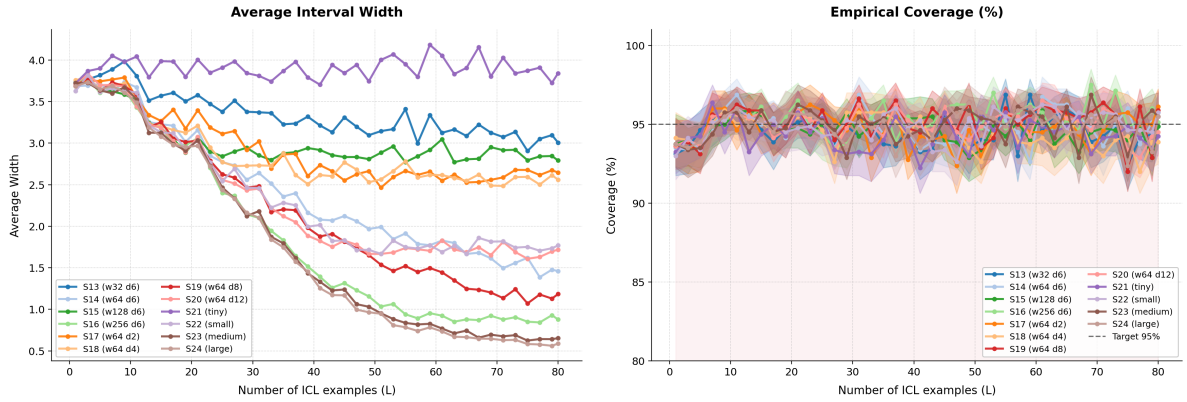



Figure 4.2: SpeedCP average interval width (left) and empirical coverage (right) as functions of the number of in-context examples $L$ for all 12 architectures. The dashed line in the coverage panel indicates the nominal 95% target. All models maintain coverage near the target, but interval width varies by a factor of 5× across architectures at $L = 80$.

**Depth scaling.** Among the depth-scaling variants (S17–S20), increasing depth from 2 to 8 at fixed width 64 improves the width drop from 28.7% to 67.9%. However, further increasing depth to 12 (S20) produces a drop back to 53.8%, suggesting that depth beyond a certain point may introduce optimization difficulties or capacity misallocation at this width. The optimal depth for width 64 appears to be around 8 layers, where S19 achieves the best balance between representation power and trainability.

**Standard presets.** The preset comparison (S21–S24) reveals the most dramatic range. S21 (GPT-2 tiny) is the only model with a *negative* width drop (−3.2%): its intervals actually widen slightly with more context, as the model fails to learn from demonstrations and the conformal mechanism can only respond to the unchanging residual distribution. At the other extreme, S24 (GPT-2 large) achieves 84.0% width drop, reducing intervals from 3.69 to 0.59. S23 (GPT-2 medium) performs nearly as well at 82.4%, suggesting diminishing returns beyond medium-scale models for this task.

A key insight from these results is the near-perfect correlation ($\rho \approx 0.999$) between RMSE at $L = 80$ and interval width at $L = 80$ across all 12 models. This indicates that in this experimental regime, conformal interval width is almost entirely determined by the underlying point prediction

quality, with the conditional calibration mechanism contributing only marginal adjustments. In other words, investing in better base models is far more effective than sophisticated conformal methods for achieving tighter intervals.

This correspondence extends beyond the endpoints: comparing Figures 4.1 and 4.2 reveals that the *average* SpeedCP interval width curves and the RMSE curves share broadly similar shapes throughout the ICL trajectory. In both panels, the 12 models start from nearly identical initial values at $L = 1$ and then fan out into approximately the same rank ordering as $L$ increases. The strong learners (S16, S23, S24) exhibit steep, monotonically decreasing trajectories in both RMSE and average width, with the sharpest improvement occurring between $L = 10$ and $L = 50$. The moderate learners (S14, S19, S20, S22) show a more gradual decline, and the weak learners (S13, S15, S17, S18) plateau early. Most tellingly, S21 (GPT-2 tiny) produces a flat or slightly increasing trajectory in *both* panels. It is important to note, however, that this similarity holds at the *population-average* level; at the level of individual test points, SpeedCP assigns different interval widths depending on each point's covariate $x_{L+1}$ through the RKHS kernel, so the width for a specific test point may deviate noticeably from the average trend. The curves in Figure 4.2 represent means over 800 test points, which smooth out these covariate-dependent fluctuations.

The macro-level coupling between RMSE and average width can be understood as follows. Under the symmetric absolute-residual score $S_i = |y_i - \hat{y}_i|$, the marginal conformal quantile is the $(1-\alpha)$-th order statistic of the calibration residuals (classical conformal guarantee; see Section 2.5). SpeedCP refines this by fitting a covariate-dependent quantile surface $q(x)$ via RKHS quantile regression, so each test point receives an individualized cutoff $q(x_{L+1})$ that can differ from the global quantile. Nevertheless, the *average* of these individualized cutoffs is strongly driven by the overall scale of the residual distribution, which in turn is governed by the RMSE. When the RMSE decreases across the ICL trajectory—indicating that residuals are systematically shrinking—both the global quantile and the average of the conditional quantiles shift downward, producing narrower intervals on average. The conditional modulation—widening intervals in high-residual regions and narrowing them in low-residual regions—is typically small relative to the overall scale shift. The

practical implication is that the ICL model quality remains the *primary* determinant of average interval efficiency, while the conditional calibration provides a secondary, covariate-level refinement.

# CHAPTER 5

# Results II: Task Complexity and Signal-to-Noise Effects

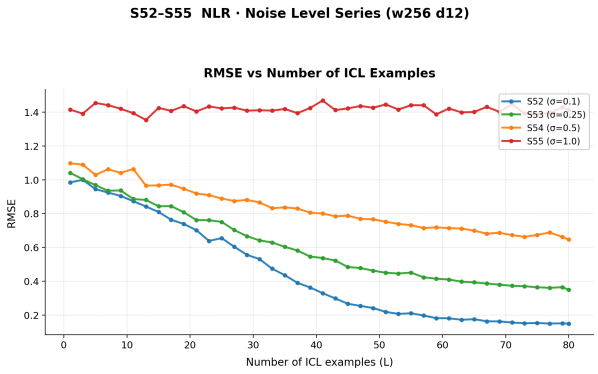
Having established that architecture plays a dominant role in determining conformal interval efficiency under a fixed task, we now turn to the complementary question: how do the regression task itself and the noise level affect ICL prediction quality and conformal calibration? We fix the architecture to a moderately large model (width 256, depth 12) and vary both the regression function class and the noise standard deviation across 16 configurations (S52–S67).

## 5.1 RMSE Under Varying Task Complexity and Noise

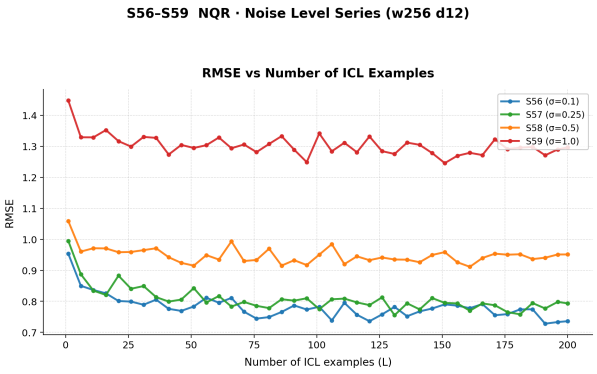
Figure 5.1 presents the RMSE curves across all 16 task–noise configurations. Several important patterns emerge from this comparison.

**Noise creates an irreducible RMSE floor.** For all four task families, the RMSE at $\sigma = 1.0$ remains elevated even at the maximum context length, while low noise ($\sigma = 0.1$) permits substantial RMSE reduction. This is theoretically expected: even an oracle estimator cannot achieve RMSE below $\sigma$ for homoscedastic Gaussian noise, and our transformer models approach this floor only for the simplest task (NLR). Quantitatively, the RMSE at the final context length grows from $\sigma = 0.1$ to $\sigma = 1.0$ by a factor of $9.5\times$ for NLR, $1.8\times$ for NQR, $3.5\times$ for 2NN, and $3.5\times$ for NDT.

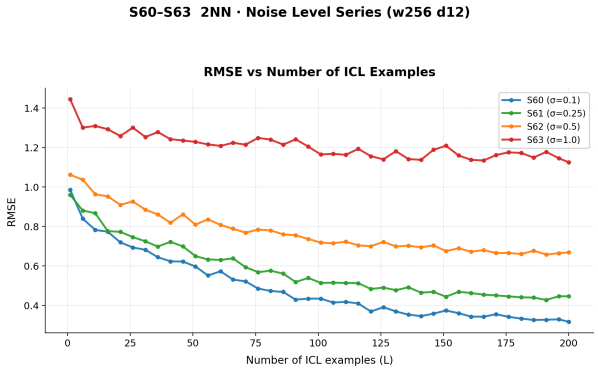
**Task learnability varies dramatically.** At the lowest noise level ($\sigma = 0.1$), the transformer achieves sharply different final RMSE values across tasks: 0.149 for NLR (near-optimal), 0.317 for 2NN, 0.415 for NDT, and 0.736 for NQR. This ordering reflects the inherent difficulty of in-context learning for each function class. Linear regression is the canonical ICL task for which transformers are known to implement approximate gradient descent [Akyurek et al., 2023, von Oswald et al., 2023], explaining the near-optimal performance. The two-layer neural network function, despite



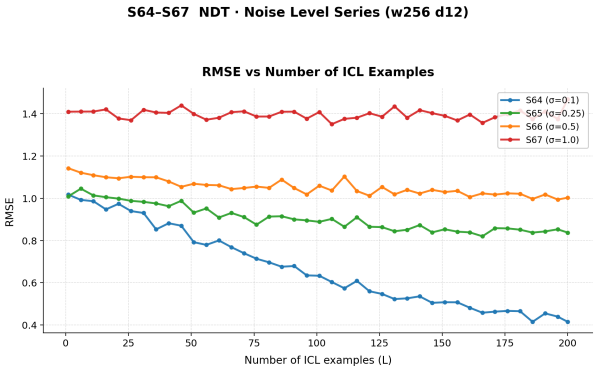

(a) Noisy linear regression (NLR)




(b) Noisy quadratic regression (NQR)




(c) Two-layer neural network (2NN)

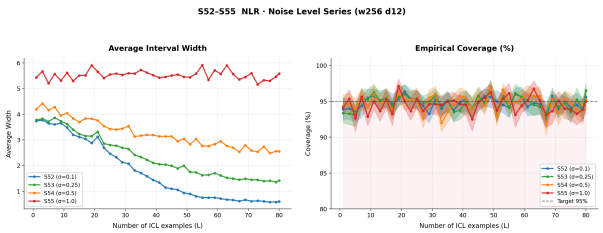



(d) Decision tree (NDT)

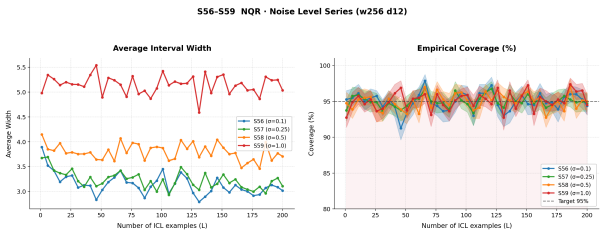
Figure 5.1: RMSE as a function of the number of in-context examples $L$ for four regression task families at four noise levels ($\sigma \in \{0.1, 0.25, 0.5, 1.0\}$). All experiments use the same architecture (width 256, depth 12). The noise level creates a floor below which RMSE cannot decrease, and the rate of RMSE improvement with $L$ depends strongly on the task family.

being nonlinear, has a relatively smooth structure that the transformer can partially learn. The decision tree function, while discontinuous, has a hierarchical structure that may be partially amenable to ICL. Surprisingly, quadratic regression proves the most resistant to ICL, with RMSE declining only from 0.96 to 0.74 over 200 demonstrations at $\sigma = 0.1$.

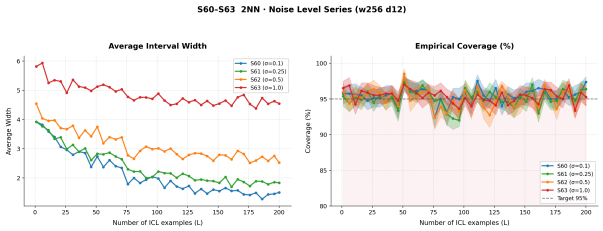
**RMSE convergence rate differs across tasks.** NLR exhibits a smooth, monotonically decreasing RMSE curve that continues to benefit from additional demonstrations up to the maximum $L = 80$. The 2NN task shows a similar but slower convergence pattern over $L = 200$. In contrast, NQR and NDT exhibit early saturation: the RMSE curves flatten substantially by $L \approx 50$–$75$, suggesting that additional demonstrations beyond this point provide diminishing marginal information for these



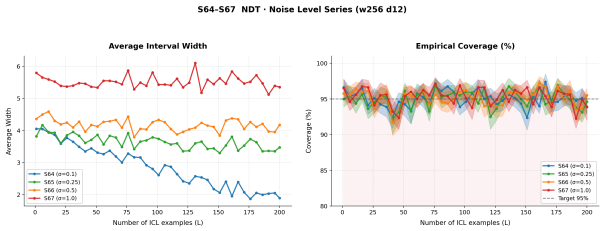

(a) NLR: S52–S55




(b) NQR: S56–S59




(c) 2NN: S60–S63




(d) NDT: S64–S67

Figure 5.2: SpeedCP interval width (left subpanels) and empirical coverage (right subpanels) across four task families at four noise levels. Coverage remains stable near the 95% target across all 16 configurations, while interval width behavior mirrors the RMSE patterns from Figure 5.1.

task families. This has important implications for practical ICL deployments, where the marginal cost of additional demonstrations must be weighed against the expected improvement.

## 5.2 Conformal Interval Width and Coverage

Figure 5.2 and Table 5.1 present the conformal interval behavior across all 16 task–noise configurations. The results reveal several key phenomena:

**Coverage robustness.** Across all 16 configurations, the curve-averaged empirical coverage ranges from 94.54% to 95.48%, consistently close to the nominal 95% target. This is a remarkable validation of the SpeedCP framework: despite substantial variation in the underlying prediction difficulty—from near-optimal NLR at $\sigma = 0.1$ to severely noise-limited NQR at $\sigma = 1.0$—the conformal calibration mechanism reliably produces valid intervals. The coverage stability holds at individual context lengths as well, with per-model standard deviations of approximately 1% over the ICL trajectory.

| Task | $\sigma$ | Cov. (%) | Width ($L=1$) | Width (final) | Width drop (%) | RMSE (final) |
|---|---|---|---|---|---|---|
| NLR | 0.1 | 94.75 | 3.73 | 0.60 | 83.9 | 0.149 |
| NLR | 0.25 | 94.83 | 3.78 | 1.42 | 62.5 | 0.349 |
| NLR | 0.5 | 94.77 | 4.19 | 2.55 | 39.1 | 0.647 |
| NLR | 1.0 | 94.54 | 5.43 | 5.58 | $-2.8$ | 1.414 |
| NQR | 0.1 | 94.98 | 3.89 | 3.02 | 22.5 | 0.736 |
| NQR | 0.25 | 95.00 | 3.67 | 3.11 | 15.4 | 0.793 |
| NQR | 0.5 | 94.97 | 4.15 | 3.71 | 10.6 | 0.951 |
| NQR | 1.0 | 95.12 | 4.98 | 5.04 | $-1.1$ | 1.296 |
| 2NN | 0.1 | 95.48 | 3.92 | 1.51 | 61.6 | 0.317 |
| 2NN | 0.25 | 95.11 | 3.92 | 1.83 | 53.3 | 0.446 |
| 2NN | 0.5 | 95.06 | 4.54 | 2.52 | 44.4 | 0.668 |
| 2NN | 1.0 | 95.37 | 5.81 | 4.54 | 21.9 | 1.124 |
| NDT | 0.1 | 95.01 | 4.05 | 1.89 | 53.4 | 0.415 |
| NDT | 0.25 | 94.97 | 3.82 | 3.47 | 9.0 | 0.838 |
| NDT | 0.5 | 95.14 | 4.36 | 4.17 | 4.3 | 1.003 |
| NDT | 1.0 | 95.30 | 5.79 | 5.35 | 7.7 | 1.469 |

Table 5.1: Summary of SpeedCP performance across all 16 task–noise configurations. Coverage is the curve-averaged empirical coverage. Width drop measures the percentage reduction from initial ($L = 1$) to final context length ($L = 80$ for NLR, $L = 200$ for others). Negative width drop indicates that intervals widened over the ICL trajectory.

**Noise-driven width inflation.** The initial interval width at $L = 1$ already reflects the noise level: for NLR, it increases from 3.73 at $\sigma = 0.1$ to 5.43 at $\sigma = 1.0$. This initial width is determined primarily by the prior predictive error, which scales with the noise standard deviation. More importantly, the *width trajectory* depends critically on the signal-to-noise ratio (SNR). At high SNR ($\sigma = 0.1$), the model rapidly improves its predictions with more context, and the conformal interval shrinks accordingly. At low SNR ($\sigma = 1.0$), the irreducible noise floor prevents meaningful RMSE improvement, and the interval width remains essentially constant or even increases slightly.

**Task-dependent ICL benefit for intervals.** The width drop metric reveals a clear ordering of tasks by their amenability to ICL-driven interval shrinkage. At $\sigma = 0.1$: NLR achieves 83.9% width drop (the strongest), followed by 2NN at 61.6%, NDT at 53.4%, and NQR at only 22.5%. This ordering closely mirrors the RMSE improvement and reflects the fundamental learnability of each function class under ICL. The practical implication is significant: conformal prediction for

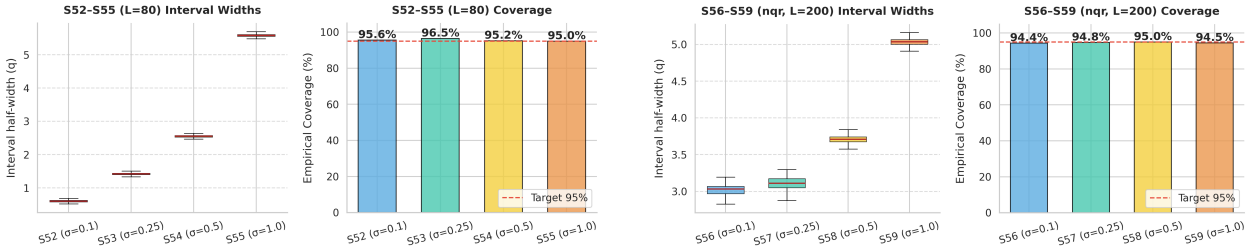
ICL is most valuable precisely when the model can effectively learn from demonstrations, as wider intervals provide no false sense of precision when the model fails to improve.

**Width saturation at high noise.** For NLR at $\sigma = 1.0$ (S55) and NQR at $\sigma = 1.0$ (S59), the width drop is slightly negative ($-2.8\%$ and $-1.1\%$, respectively), meaning that intervals actually widen marginally over the ICL trajectory. This counterintuitive result arises because the model's RMSE does not improve with more context at these noise levels, and the conformal mechanism responds to the unchanged or slightly increased residual variability. This phenomenon underscores a fundamental limitation: conditional conformal prediction cannot "create" predictive power where the base model has none.

## 5.3 Cross-Task Analysis

Comparing across task families at fixed noise levels reveals how task structure interacts with ICL and conformal prediction. At $\sigma = 0.1$, the final interval widths span a $5\times$ range: from 0.60 (NLR) to 3.02 (NQR). This range collapses at $\sigma = 1.0$, where all tasks produce final widths between 4.54 and 5.58—the noise floor dominates and task differences become secondary.

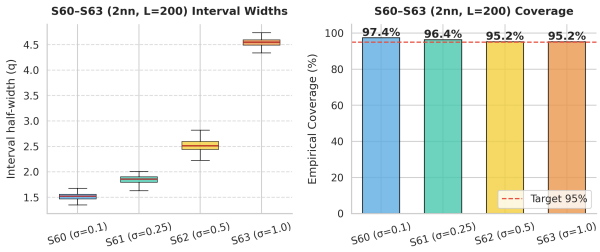
The 2NN task occupies an interesting middle ground. Despite being nonlinear, the transformer achieves 61.6% width drop at $\sigma = 0.1$, suggesting that the two-layer ReLU network structure is sufficiently "smooth" for ICL to capture meaningful patterns. Even at $\sigma = 1.0$, the 2NN task retains a positive 21.9% width drop, the highest among all tasks at this noise level. This resilience may stem from the relatively low effective complexity of the 2NN function class with hidden dimension 4. The decision tree task (NDT) shows a sharp sensitivity to noise: while it achieves a respectable 53.4% width drop at $\sigma = 0.1$, this collapses to 9.0% at $\sigma = 0.25$ and 4.3% at $\sigma = 0.5$. The discontinuous nature of decision tree functions may make them particularly vulnerable to noise perturbations that blur the decision boundaries.



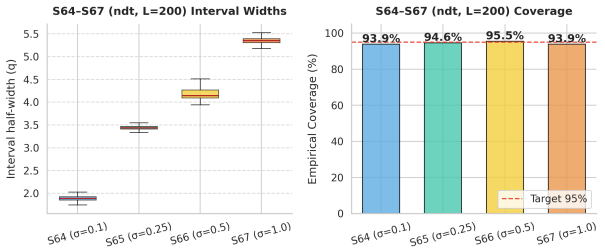

(a) NLR (S52–S55), $L = 80$




(b) NQR (S56–S59), $L = 200$




(c) 2NN (S60–S63), $L = 200$




(d) NDT (S64–S67), $L = 200$

Figure 5.3: Box plots of SpeedCP interval half-width (left subpanels) and empirical coverage at the final context length (right subpanels) for all four task families across noise levels. The box plots show the median, interquartile range (IQR), and whiskers extending to 1.5× IQR. Coverage bar charts display the empirical coverage with the 95% target indicated by the dashed red line.

## 5.4 Interval Width Distributions at Final Context Length

While the preceding analysis focuses on average interval width, the *distribution* of interval widths across test points reveals important information about the heterogeneity of the conformal calibration. Figure 5.3 presents box plots of the interval half-width and bar charts of the empirical coverage at the final context length for each task–noise configuration.

Several observations emerge from the distributional analysis. First, the interval width variability (IQR) increases with noise level across all task families, indicating that higher noise not only raises the median width but also amplifies the heterogeneity of the calibrated intervals. This is most pronounced for the NLR task, where the IQR at $\sigma = 1.0$ is substantially larger than at $\sigma = 0.1$, reflecting the wider range of residual magnitudes when noise dominates the prediction error.

Second, the 2NN task exhibits a distinctive pattern: at low noise ($\sigma = 0.1$, S60), the coverage reaches 97.4%—notably above the 95% target. This over-coverage suggests that the

SpeedCP conformal cutoff is conservative for this task at low noise, possibly because the residual distribution has heavier tails than assumed by the kernel-based calibration. As noise increases, the coverage converges toward the nominal level (95.2% at $\sigma = 0.5$ and $\sigma = 1.0$), consistent with the residual distribution becoming more Gaussian-like when additive noise dominates.

Third, the NDT task shows edge undercoverage at both extremes: 93.9% at $\sigma = 0.1$ (S64) and 93.9% at $\sigma = 1.0$ (S67), both approximately 1 percentage point below the target. At low noise, this may reflect the difficulty of calibrating intervals for a discontinuous function class, where residuals can exhibit sharp, localized spikes near decision boundaries. At high noise, the undercoverage may arise from the interaction between the noise-dominated residuals and the fixed kernel hyperparameters, which were not optimized specifically for the NDT task.

Finally, the NQR task displays the tightest IQR relative to the median across all noise levels, with box plots showing compact distributions. This reflects the fact that the quadratic regression residuals, while large in absolute terms (due to the difficulty of ICL for this task), are relatively homogeneous across test points—the model fails uniformly rather than exhibiting localized prediction quality variation.

# CHAPTER 6

# Results III: Input Dimensionality Effects

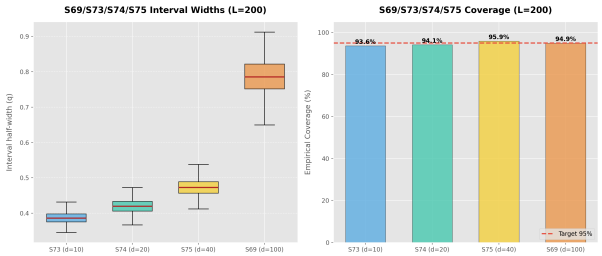
Having examined the effects of model architecture (Section 4) and task complexity (Section 5), we now study a third axis: how the input dimensionality $d$ of the regression problem affects ICL prediction quality and conformal interval behavior. We train separate models for each dimensionality $d \in \{10, 20, 40, 100\}$ across all four task families (NLR, NQR, 2NN, NDT), using the same base architecture (width 256, depth 12). SpeedCP is evaluated at context length $L = 200$ (for NLR) or $L = 500$ (for nonlinear tasks) with $N = 1{,}600$ episodes. Crucially, each of these 16 models (S69–S84) is trained from scratch on its respective dimensionality and task, so differences across $d$ reflect the inherent difficulty of learning higher-dimensional functions in-context.

## 6.1 Coverage and Interval Width at Varying Dimensions

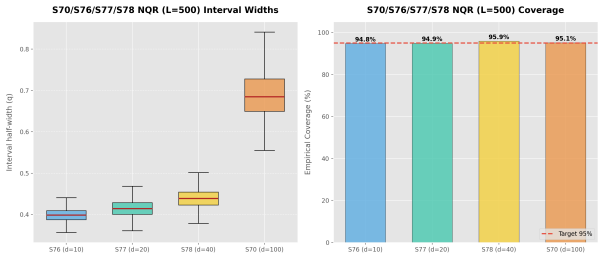
Figure 6.1 and Table 6.1 present the SpeedCP interval widths and empirical coverage across the 16 configurations.

**Coverage remains stable across most dimensions and tasks.** The empirical coverage stays within approximately 1 percentage point of the nominal 95% target for nearly all configurations, ranging from 93.1% (2NN at $d = 100$) to 96.1% (2NN at $d = 10$). This confirms that the coverage guarantee is robust to dimensionality changes across diverse function classes, even when using a single set of SpeedCP hyperparameters.

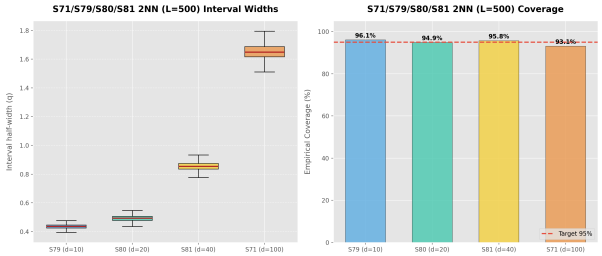
**Polynomial and smooth functions degrade gracefully with dimension.** For the linear (NLR) and quadratic (NQR) tasks, increasing the dimension from $d = 10$ to $d = 100$ leads to a moderate increase in interval width (e.g., $0.388 \to 0.786$ for NLR, $0.399 \to 0.690$ for NQR). The box plots in Figure 6.1(a, b) show that the width distributions simply shift upward with heavier tails at $d = 100$.



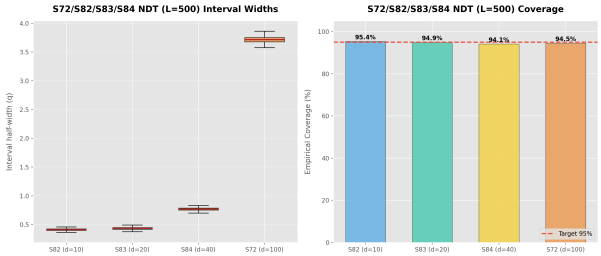

(a) NLR (S69, S73–S75), $L = 200$




(b) NQR (S70, S76–S78), $L = 500$




(c) 2NN (S71, S79–S81), $L = 500$




(d) NDT (S72, S82–S84), $L = 500$

Figure 6.1: SpeedCP interval half-width box plots (left subpanels) and empirical coverage (right subpanels) at the final context length for four input dimensionalities across four task families. Each model is separately trained on its respective dimension $d$. SpeedCP hyperparameters are fixed across dimensions.

This indicates that while higher-dimensional polynomial functions are fundamentally harder to learn in-context with limited demonstrations (e.g., $L = 200$ for NLR at $d = 100$), the models still capture significant structure, preventing complete failure.

**Non-smooth and complex functions degrade sharply at high dimensions.** The behavior is drastically different for the two-layer neural network (2NN) and decision tree (NDT) tasks. While their interval widths are comparable to NLR and NQR at $d = 10$, they explode as dimension increases. For NDT, the average width skyrockets from 0.412 at $d = 10$ to 3.716 at $d = 100$—a more than $9\times$ increase. The 2NN task similarly jumps from 0.436 to 1.652. Figure 6.1(c, d) reveals massive variance in the $d = 100$ width distributions for these tasks. This sharp degradation suggests that the sample complexity required for ICL to capture the localized, non-smooth decision boundaries of NDT or the interactions in high-dimensional 2NN overwhelmingly exceed the provided context length ($L = 500$), pushing the models toward essentially random guessing at $d = 100$.

| Task | $d$ | Model | Coverage (%) | Avg. Width |
|---|---|---|---|---|
| NLR | 10 | S73 | 93.6 | 0.388 |
| NLR | 20 | S74 | 94.1 | 0.420 |
| NLR | 40 | S75 | 95.9 | 0.474 |
| NLR | 100 | S69 | 94.9 | 0.786 |
| NQR | 10 | S76 | 94.8 | 0.399 |
| NQR | 20 | S77 | 94.9 | 0.415 |
| NQR | 40 | S78 | 95.9 | 0.439 |
| NQR | 100 | S70 | 95.1 | 0.690 |
| 2NN | 10 | S79 | 96.1 | 0.436 |
| 2NN | 20 | S80 | 94.9 | 0.493 |
| 2NN | 40 | S81 | 95.8 | 0.855 |
| 2NN | 100 | S71 | 93.1 | 1.652 |
| NDT | 10 | S82 | 95.4 | 0.412 |
| NDT | 20 | S83 | 94.9 | 0.433 |
| NDT | 40 | S84 | 94.1 | 0.769 |
| NDT | 100 | S72 | 94.5 | 3.716 |

Table 6.1: SpeedCP performance across input dimensionalities $d \in \{10, 20, 40, 100\}$ for four task families. Coverage generally remains near the 95% target, but interval widths increase substantially at higher dimensions, particularly for the highly non-linear NDT and 2NN tasks.

**Implications.** These results highlight a profound interaction between task complexity and input dimensionality in ICL. Conformal prediction efficiently tracks these transitioning failure modes: as the underlying ICL algorithm breaks down given the ratio of context length to problem dimension and task complexity, the conformal mechanism seamlessly inflates the intervals to maintain valid coverage. Practitioners deploying ICL in high-dimensional settings must carefully consider whether the model has sufficient context capacity to learn the target function class, as conformal intervals will inevitably warn of poor performance through massive width inflation.

# CHAPTER 7

# Discussion

**Coverage stability as a strong empirical finding.** Across all 44 experimental configurations—spanning 12 architectures, 4 task families $\times$ 4 noise levels, and 4 tasks $\times$ 4 input dimensionalities—the empirical coverage of SpeedCP remains within approximately 2 percentage points of the nominal 95% target (range 93.1%–96.1%). This stability is not merely a statistical artifact of large test sets; it persists at each individual context length $L$ and across the full range of model qualities, from the completely ineffective S21 (GPT-2 tiny) to the near-optimal S24 (GPT-2 large), and from $d = 10$ to $d = 100$. The per-model coverage standard deviation over the ICL trajectory is approximately 1%, indicating that coverage fluctuations are small and symmetric around the target. This finding strongly validates the CondConf/SpeedCP theoretical framework in the ICL setting: the conformal calibration mechanism successfully provides distribution-free coverage guarantees despite the complex, non-standard nature of ICL predictions.

**Interval efficiency is dominated by model quality.** The most consistent finding across all experimental axes is that conformal interval width is almost entirely determined by the quality of the underlying point predictions. The near-perfect correlation between RMSE and interval width—observed both in the architecture study (Section 4) and the task–noise study (Section 5)—implies that sophisticated conformal methods contribute only marginal improvements over a global (marginal) conformal baseline when the base model is strong or weak. The practical recommendation is clear: to achieve tight conformal intervals in ICL, the primary investment should be in improving the base model (through larger architecture, better training, or more appropriate task design), rather than in more complex conformal calibration procedures.

**Practical guidelines for ICL uncertainty quantification.** Based on our findings, we offer the following practical recommendations:

1. *Use the largest feasible model.* The most dramatic improvements in interval efficiency come from increasing model capacity, not from choosing more sophisticated conformal methods.
2. *Monitor the noise regime.* At high noise ($\sigma \geq 1.0$ relative to the signal), conformal intervals provide valid coverage but cannot be tightened through additional context. In such regimes, interval width is not informative about model quality.
3. *Consider task-specific calibration.* The optimal SpeedCP hyperparameters ($\gamma$, $\lambda$) may differ across task families, and using task-specific tuning could yield additional efficiency gains beyond the fixed-hyperparameter results reported here.
4. *Be cautious with nonlinear tasks.* ICL for quadratic and decision tree functions shows limited improvement with context length, and conformal intervals in these settings may remain wide even with large $L$.

**Limitations.** Several limitations of this study should be noted. First, all experiments use a single random seed, so inter-seed variability is not quantified. Second, the SpeedCP hyperparameters are fixed per experimental suite rather than individually tuned per configuration, which may understate the potential benefits of conditional calibration. Third, the input features used for the SpeedCP kernel ($x_{L+1}$) are relatively low-dimensional; richer feature representations (e.g., learned embeddings) could yield stronger conditional adaptation. Finally, direct empirical comparison between CondConf and SpeedCP using identical evaluation protocols was not conducted in this study, as the CondConf evaluation results for the full S13–S67 suite were not available at the time of writing.

# CHAPTER 8

# Conclusion

We have presented a systematic empirical study of conditional conformal prediction for in-context learning regression, examining the interplay between model architecture, task complexity, signal-to-noise ratio, and input dimensionality. Our experiments demonstrate that SpeedCP provides reliable coverage guarantees across a wide range of settings, maintaining empirical coverage within approximately 2 percentage points of the nominal 95% target across all 44 experimental configurations.

The central insight is that conformal interval efficiency in ICL is fundamentally limited by the quality of the base model's point predictions. Architecture scaling—particularly model width—is the dominant factor determining interval width, with the best models achieving 84.0% width reduction across the ICL trajectory. Task complexity and noise level interact to create an "ICL learnability" spectrum: from linear regression (where transformers approach Bayes-optimal performance and intervals shrink dramatically) to quadratic regression (where ICL provides limited improvement and intervals remain wide). At high noise levels, the irreducible error floor prevents interval shrinkage regardless of model quality or context length, a fundamental constraint that no conformal method can overcome.

These findings provide actionable guidance for practitioners deploying uncertainty-quantified ICL systems: invest in model capacity, understand the noise regime, and ensure that the context length comfortably exceeds the input dimensionality.

# APPENDIX A

# Detailed Experimental Parameters

This appendix provides the complete hyperparameter specifications for all experiments reported in this paper, extracted from the training configuration files. All models use the GPT-2 decoder-only transformer architecture with RMSNorm (with $\varepsilon = 10^{-6}$) replacing LayerNorm, and the MLP hidden dimension is set to $4\times$ the embedding dimension in all cases. No Mixture-of-Experts routing is used in any experiment. The input data distribution is isotropic Gaussian ($x \sim \mathcal{N}(0, I_d)$).

**Note on experiment numbering.** Experiments reported in this paper are numbered starting from S13. Configurations S1–S12 correspond to preliminary pilot experiments conducted during the early stages of the project to validate the training pipeline and evaluation infrastructure. These pilots used a simpler setup (noiseless linear regression, $d = 20$, $L = 20$, default training steps) and are not reported here, as they were superseded by the more comprehensive configurations (noisy tasks, $d = 40$, $L = 80$, 500K training steps) used in the final study.

## A.1 Shared Training Hyperparameters

Table A.1 lists the training hyperparameters shared across all experiments. These are inherited from a common base configuration and overridden only where noted in subsequent tables.

## A.2 Architecture Scaling Study (S13–S24)

All models in the architecture study are trained on noisy linear regression ($y = w^\top x + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, 0.01)$) with normalized weight vectors ($\|w\| = 1$). The dimension curriculum increases $d$ from 10 to 40 in increments of 1 every 2,000 steps, and the context length curriculum increases $L$ from 21 to 81 (maximum 80 demonstration pairs) in increments of 2 every 2,000 steps. Table A.2

| Parameter | Value |
|---|---|
| Optimizer | AdamW |
| Learning rate | $1 \times 10^{-4}$ |
| Batch size | 64 |
| Training steps | 500,000 |
| Checkpoint interval | every 10,000 steps |
| Long-term checkpoint interval | every 100,000 steps |
| Normalization | RMSNorm ($\varepsilon = 10^{-6}$) |
| MLP hidden multiplier | 4 |
| Input data distribution | $x \sim \mathcal{N}(0, I_d)$ |

Table A.1: Shared training hyperparameters for all experiments.

lists the per-model architectural parameters.

| Model | $d_{\text{model}}$ | Layers | Heads | $n_{\text{positions}}$ | $d_{\max}$ | Group |
|---|---|---|---|---|---|---|
| S13 | 32 | 6 | 4 | 81 | 40 | Width scaling |
| S14 | 64 | 6 | 8 | 81 | 40 | Width scaling |
| S15 | 128 | 6 | 8 | 81 | 40 | Width scaling |
| S16 | 256 | 6 | 8 | 81 | 40 | Width scaling |
| S17 | 64 | 2 | 8 | 81 | 40 | Depth scaling |
| S18 | 64 | 4 | 8 | 81 | 40 | Depth scaling |
| S19 | 64 | 8 | 8 | 81 | 40 | Depth scaling |
| S20 | 64 | 12 | 8 | 81 | 40 | Depth scaling |
| S21 | 32 | 4 | 4 | 81 | 40 | GPT-2 tiny |
| S22 | 64 | 8 | 8 | 81 | 40 | GPT-2 small |
| S23 | 128 | 12 | 8 | 81 | 40 | GPT-2 medium |
| S24 | 256 | 12 | 8 | 81 | 40 | GPT-2 large |

Table A.2: Per-model architectural parameters for the architecture scaling study (S13–S24). All models are trained on noisy linear regression ($\sigma = 0.1$) with $n_{\text{positions}} = 81$ (up to $L = 80$ in-context examples). $d_{\text{model}}$ denotes the embedding dimension (width), and $d_{\max} = 40$ is the maximum input dimension reached during training.

## A.3 Task–Noise Study (S52–S67)

All models in the task–noise study use a fixed architecture ($d_{\text{model}} = 256$, 12 layers, 8 heads) and vary the task family and noise level. Table A.3 lists the per-experiment configurations.

| Model | Task | $\sigma$ | $n_{\text{positions}}$ | $L_{\max}$ | Dim. curriculum | Task-specific |
|---|---|---|---|---|---|---|
| S52 | NLR | 0.1 | 81 | 80 | $10 \to 40$ | $\|w\| = 1$ |
| S53 | NLR | 0.25 | 81 | 80 | $10 \to 40$ | $\|w\| = 1$ |
| S54 | NLR | 0.5 | 81 | 80 | $10 \to 40$ | $\|w\| = 1$ |
| S55 | NLR | 1.0 | 81 | 80 | $10 \to 40$ | $\|w\| = 1$ |
| S56 | NQR | 0.1 | 201 | 200 | $10 \to 40$ | $\|w\| = 1$ |
| S57 | NQR | 0.25 | 201 | 200 | $10 \to 40$ | $\|w\| = 1$ |
| S58 | NQR | 0.5 | 201 | 200 | $10 \to 40$ | $\|w\| = 1$ |
| S59 | NQR | 1.0 | 201 | 200 | $10 \to 40$ | $\|w\| = 1$ |
| S60 | 2NN | 0.1 | 201 | 200 | $10 \to 40$ | hidden dim $= 4$ |
| S61 | 2NN | 0.25 | 201 | 200 | $10 \to 40$ | hidden dim $= 4$ |
| S62 | 2NN | 0.5 | 201 | 200 | $10 \to 40$ | hidden dim $= 4$ |
| S63 | 2NN | 1.0 | 201 | 200 | $10 \to 40$ | hidden dim $= 4$ |
| S64 | NDT | 0.1 | 201 | 200 | $10 \to 40$ | tree depth $= 4$ |
| S65 | NDT | 0.25 | 201 | 200 | $10 \to 40$ | tree depth $= 4$ |
| S66 | NDT | 0.5 | 201 | 200 | $10 \to 40$ | tree depth $= 4$ |
| S67 | NDT | 1.0 | 201 | 200 | $10 \to 40$ | tree depth $= 4$ |

Table A.3: Per-experiment configurations for the task–noise study (S52–S67). All experiments use the same architecture ($d_{\text{model}} = 256$, 12 layers, 8 heads). NLR = noisy linear regression; NQR = noisy quadratic regression; 2NN = two-layer ReLU network regression; NDT = noisy decision tree. The context length curriculum for NLR runs from $L = 21$ to 81 in increments of 2; for NQR, 2NN, and NDT from $L = 51$ to 201 in increments of 5; both with 2,000-step intervals. The "Task-specific" column notes additional parameters unique to each task family.

# APPENDIX B

# Pretrained LLM In-Context Learning on Synthetic Regression

The preceding sections studied ICL behavior in *trained-from-scratch* transformers. A natural question is whether similar phenomena arise in *pretrained large language models* (LLMs) that receive regression tasks via text prompts, without any task-specific fine-tuning. We conduct a systematic evaluation of 7 open-weight LLMs spanning four architecture families and model scales from 0.6B to $\sim$123B parameters, on the same four regression tasks (NLR, NQR, 2NN, NDT) used throughout this thesis.

## B.1 Experimental Setup

**Models.** We evaluate the following pretrained LLMs:

- **Qwen3 family**: Qwen3-0.6B, Qwen3-8B, Qwen3-14B, Qwen3-32B — providing a controlled scale comparison within a single architecture family;
- **Nemotron-120B**: an NVIDIA Mixture-of-Experts model (120B total, 12B active);
- **Mistral-Large-Instruct-v0.2**: a $\sim$123B instruction-tuned model;
- **Phi-4-mini-instruct**: a compact ($\sim$4B) model from Microsoft.

All models are served locally via vLLM or HuggingFace Transformers with greedy decoding (temperature $= 0$).

**Prompt format.** Each evaluation episode constructs a text prompt in the "words2numbers" format: each in-context example lists feature values on separate lines (e.g., "Feature 0: 0.35") followed by "Output: $y$", and the query ends with "Output:" for the model to complete. No chain-of-thought is elicited.

**Evaluation grid.** For each model and task, we sweep over 10 input dimensions $d \in \{1,2,3,4,5,8,10,20,40,100\}$ and 6–7 context sizes $L$ per dimension, chosen as approximate multiples of $d$. Each condition is evaluated on $n = 200$ independent episodes. Analytical baselines (1-Nearest Neighbor, Mean-$y$) are computed from the same context data for reference.

## B.2 ICL Curves across Tasks and Dimensions

Figures B.1 and B.2 present the full ICL curves (RMSE vs. context size $L$) for three representative models—Qwen3-32B (best overall), Nemotron-120B, and Mistral-Large—across all four tasks and ten dimensions. Several patterns emerge:

- **Low dimensions ($d \leq 5$)**: All models exhibit clear ICL learning curves, with RMSE decreasing monotonically as $L$ increases. At $d = 1$, the best models approach RMSE $\approx 0.1$, close to the noise floor ($\sigma = 0.1$). Performance is competitive with or better than the 1-NN baseline.
- **Mid dimensions ($d = 8$–10)**: ICL improvement with $L$ slows significantly. RMSE plateaus around 0.9–1.1 for NLR, well above the noise floor, suggesting that the models struggle to extract the full linear signal from text-formatted high-dimensional inputs.
- **High dimensions ($d \geq 20$)**: Increasing $L$ provides minimal benefit. For NLR and NQR, models converge to RMSE $\approx 1.0$ (comparable to the marginal standard deviation of $y$), indicating near-complete failure of ICL. The 2NN task degrades most severely, with RMSE exceeding 5.0 at $d = 40$.
- **Task difficulty ordering**: Across all models and dimensions, task difficulty follows a consistent ordering: NLR $\approx$ NDT $<$ NQR $<$ 2NN. This ordering differs from the trained-transformer setting, likely because pretrained LLMs have stronger inductive biases toward locally smooth functions.
- **Context overflow**: At $d \geq 20$ with large $L$, some conditions exceed the model's context window. These missing points are marked with $\times$ on each model's curve.

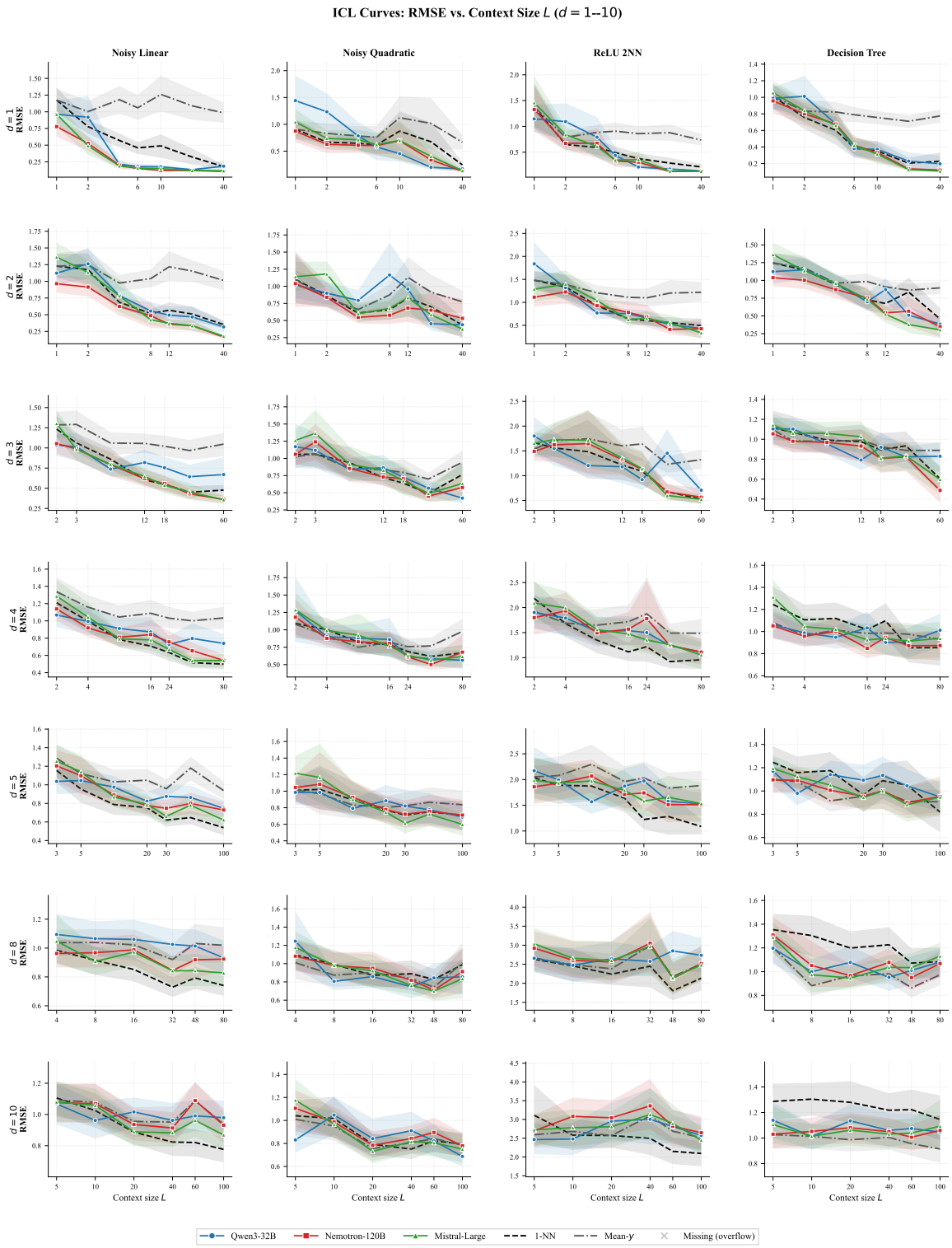



Figure B.1: ICL curves for three pretrained LLMs at low-to-mid dimensions ($d = 1$–$10$). Each row corresponds to a dimension; columns correspond to tasks. Solid lines with markers show LLM RMSE; dashed lines show the 1-NN baseline (light blue) and dotted lines show the Mean-$y$ baseline (gray). Each point averages 200 episodes.

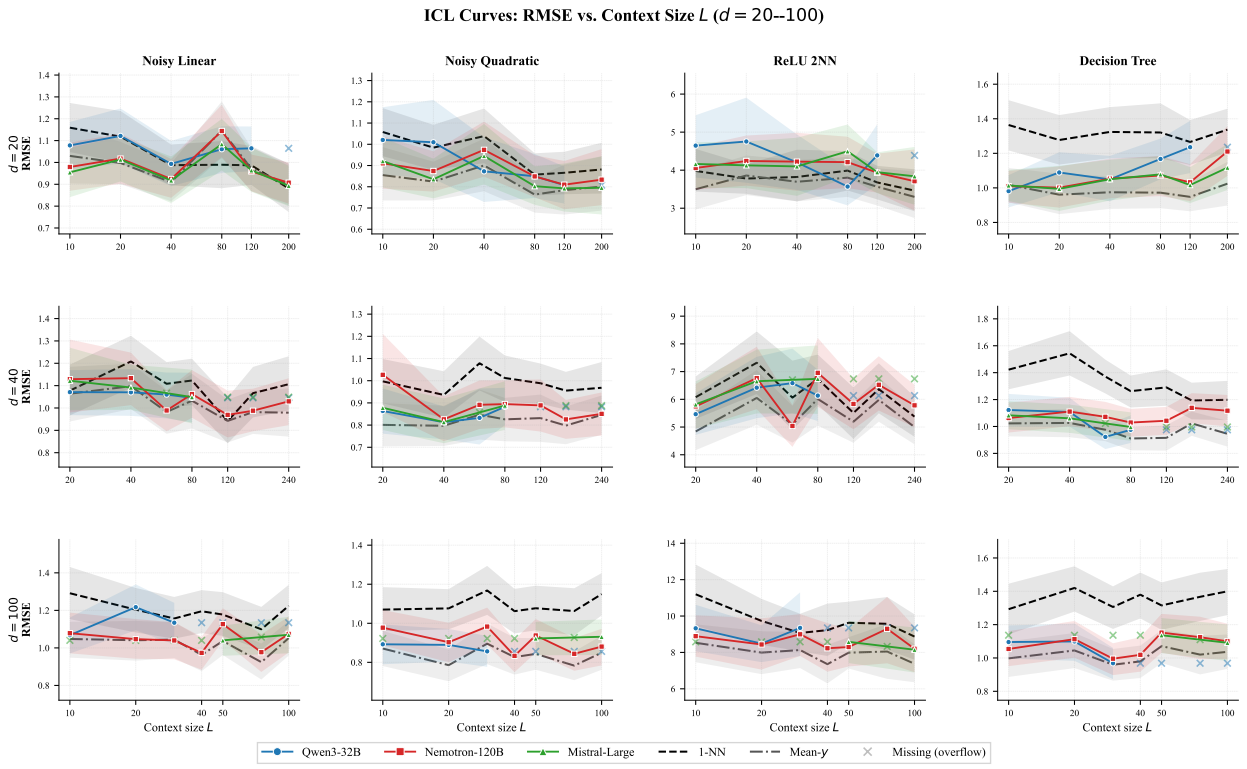



Figure B.2: ICL curves at high dimensions ($d = 20, 40, 100$). The $\times$ markers indicate conditions where the prompt exceeded the model's context window. At $d \geq 20$, increasing context size $L$ yields diminishing returns, and all models converge toward RMSE $\approx 1$ for NLR/NQR/NDT.

## B.3 Dimension Effect

Figure B.3 summarizes the dimension dependence by plotting RMSE against $d$ at a fixed context ratio $L \approx 4d$. This controls for the relative amount of information per dimension. Key observations:

- RMSE increases monotonically with $d$ for all models and tasks, confirming a strong *curse of dimensionality* in prompt-based ICL.
- For NLR, RMSE grows from $\sim$0.2 at $d = 1$ to $\sim$1.0 at $d = 20$ and plateaus thereafter, suggesting that beyond $d \approx 20$ the models effectively ignore the input features and default to a constant prediction.
- The 2NN task shows the steepest degradation, reaching RMSE $> 6$ at $d = 40$.
- All three models (Qwen3-32B, Nemotron-120B, Mistral-Large) exhibit qualitatively similar degradation curves, despite differing in architecture and scale by $\sim 17\times$.

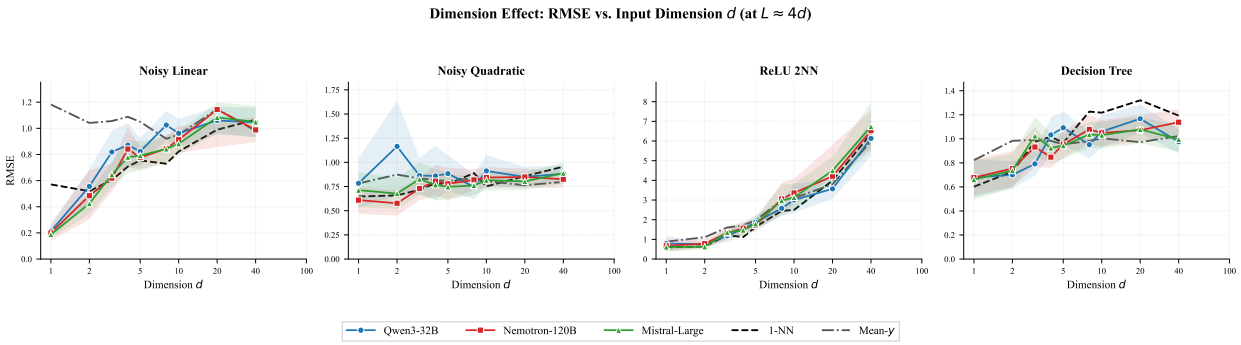



Figure B.3: Dimension effect: RMSE vs. input dimension $d$ at fixed context ratio $L \approx 4d$. All three models degrade rapidly with dimension, converging to near-constant predictions (RMSE $\approx 1$) at $d \geq 20$ for linear and quadratic tasks.

## B.4 Model Scale Effect

Figures B.4 and B.5 compare the four Qwen3 models (0.6B, 8B, 14B, 32B) to isolate the effect of model scale within a single architecture family. Findings:

- **Scale consistently helps**: Qwen3-32B achieves the lowest RMSE across nearly all conditions, followed by 14B, 8B, and 0.6B. The gap is most pronounced at low-to-mid dimensions ($d = 1$–$10$) where ICL is most effective.
- **Diminishing returns at high dimensions**: At $d \geq 20$, all four models converge to similar RMSE levels ($\approx 1.0$–$1.2$ for NLR), and the benefit of scale largely vanishes. This suggests that the bottleneck shifts from model capacity to the fundamental difficulty of extracting structure from long, high-dimensional text prompts.
- **Qwen3-0.6B as a lower bound**: The smallest model consistently underperforms, often by a large margin at low $d$ (e.g., RMSE $\approx 0.5$ vs. $\approx 0.15$ for Qwen3-32B at $d = 1$, NLR). However, even this 0.6B model exhibits meaningful ICL at $d \leq 3$, demonstrating that prompt-based ICL does not require very large models for simple tasks.

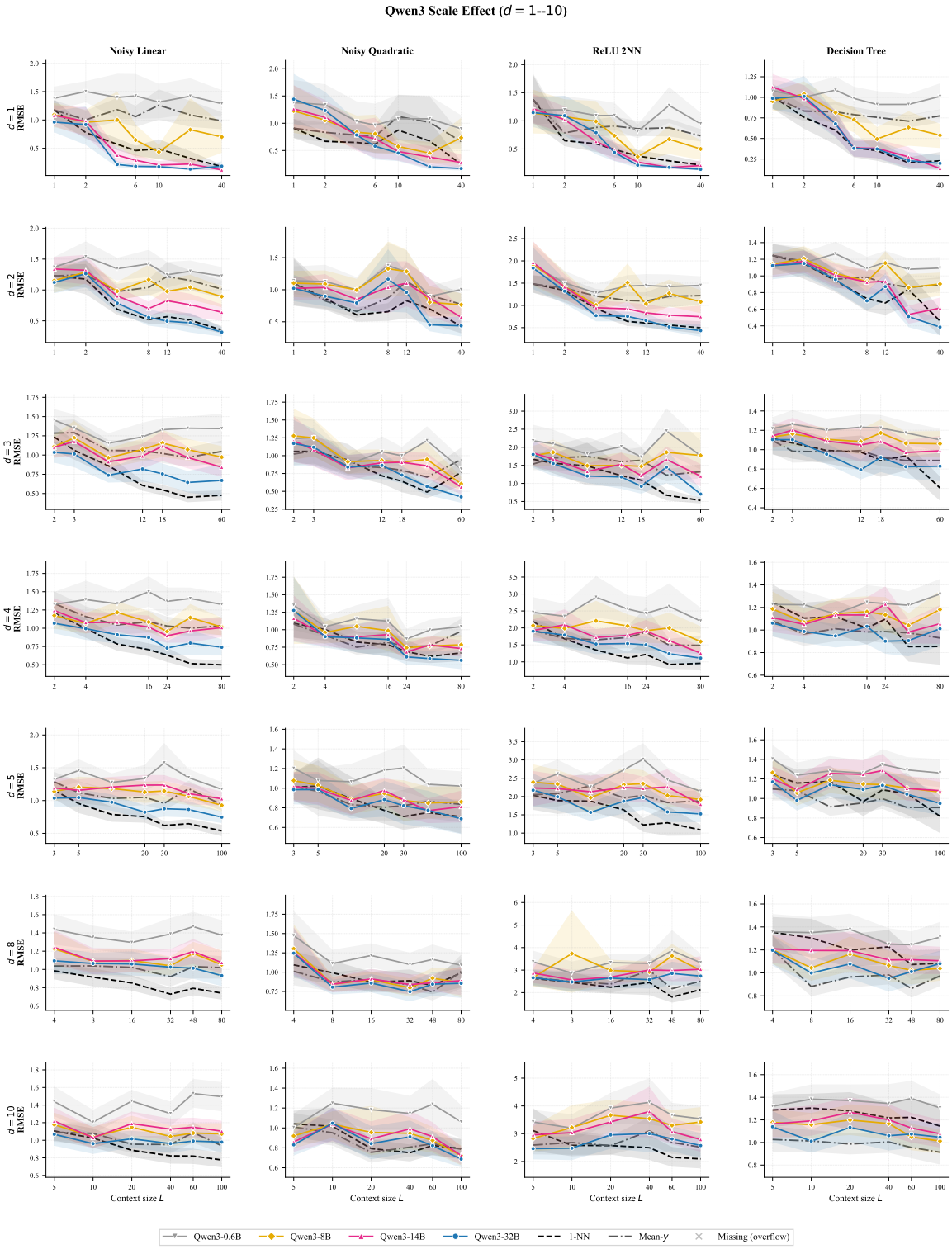



Figure B.4: Model scale effect within the Qwen3 family at low-to-mid dimensions ($d = 1$–$10$). Larger models consistently achieve lower RMSE, with the gap most pronounced at $d \leq 5$.

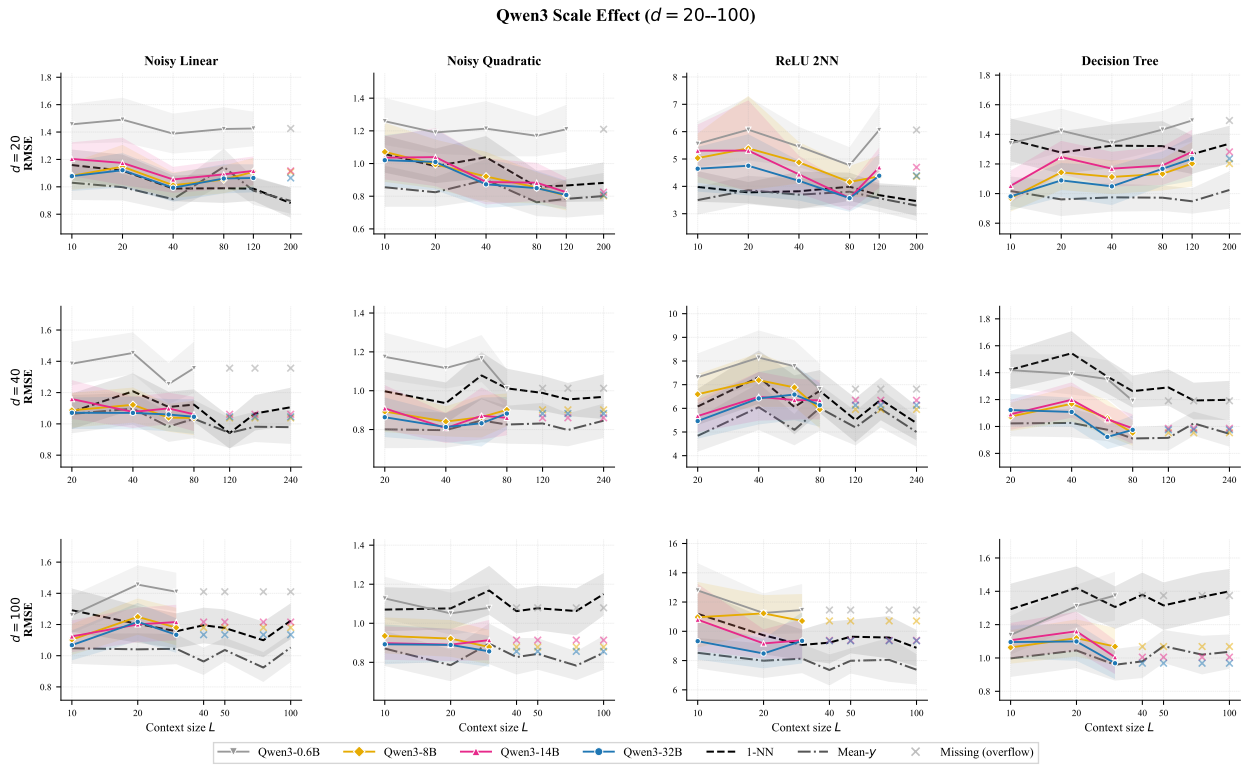



Figure B.5: Model scale effect at high dimensions ($d = 20, 40, 100$). The advantage of scale largely vanishes: all four Qwen3 models converge to similar RMSE, suggesting the bottleneck shifts from model capacity to the inherent difficulty of text-formatted high-dimensional regression.

## B.5 Noise Sensitivity: Effect of Signal-to-Noise Ratio

To complement the noise sensitivity analysis of trained transformers in Section 3.2, we evaluate how a pretrained LLM (Nemotron-120B) responds to varying noise levels. We fix the input dimension to $d = 5$ and sweep $\sigma \in \{0.1, 0.25, 0.5, 1.0\}$ across all four task families, with $L \in \{3, 5, 10, 20, 30, 50\}$ context examples and $n = 200$ evaluation episodes per condition.

Figure B.6 reveals several patterns:

- **Clear noise stratification on linear/quadratic tasks.** For NLR and NQR, the four noise levels produce well-separated RMSE curves, with $\sigma = 0.1$ consistently achieving the lowest error and $\sigma = 1.0$ the highest. This parallels the noise floor effect observed for trained transformers (Section 3.2), confirming that pretrained LLMs are similarly subject to an irreducible error proportional to $\sigma$.

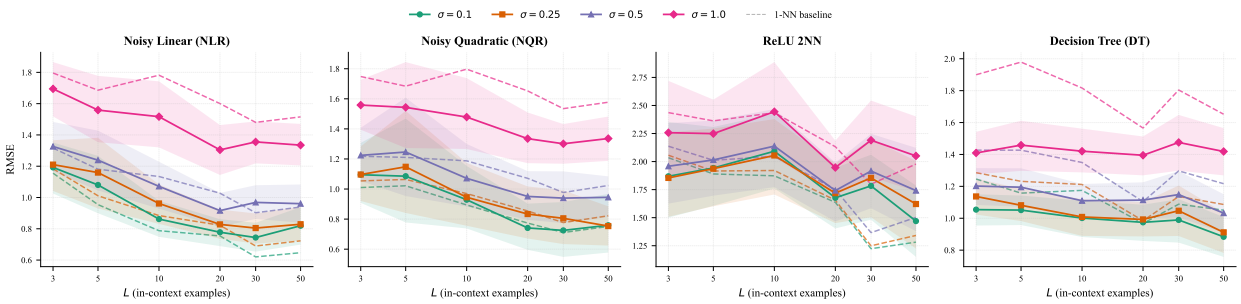



Figure B.6: Noise sensitivity of Nemotron-120B at $d = 5$. Solid lines show LLM RMSE as a function of context length $L$ at four noise levels $\sigma \in \{0.1, 0.25, 0.5, 1.0\}$. Dashed lines show the corresponding 1-NN baseline at each noise level. Higher noise degrades both the LLM and the baseline, but the LLM's relative advantage over 1-NN shrinks at high noise.

- **Diminished noise effect on nonlinear tasks.** For ReLU 2NN and Decision Tree, the gap between noise levels narrows substantially. This is because the LLM already struggles with these task families at $d = 5$ (RMSE $> 1.0$ even at $\sigma = 0.1$), so the additional noise has a smaller marginal impact—the model's approximation error dominates the observation noise.
- **Noise degrades context utilization.** At $\sigma = 0.1$, increasing $L$ from 3 to 50 reduces NLR RMSE by 31% (from 1.19 to 0.82). At $\sigma = 1.0$, the same increase yields only a 21% reduction (from 1.69 to 1.33). High noise thus weakens the LLM's ability to extract useful signal from additional context, analogous to the "noise floor blocking" phenomenon in trained transformers.
- **LLM vs. 1-NN baseline.** The LLM consistently outperforms 1-NN at $\sigma = 0.1$ for NLR and NQR when $L \geq 10$, indicating that it performs some form of regression beyond nearest-neighbor retrieval. However, at $\sigma = 1.0$, the LLM barely matches 1-NN, suggesting that the regression component of the LLM's mechanism is more sensitive to noise than pure retrieval.

Compared to the trained transformers in Section 3.2, the pretrained LLM exhibits qualitatively similar noise sensitivity patterns but at a much higher absolute RMSE level. For instance, on NLR at $\sigma = 0.1$ and $L = 50$, the trained transformer achieves RMSE $\approx 0.15$ (near Bayes-optimal), while Nemotron-120B achieves 0.82—a $5\times$ gap. This further underscores that prompt-based ICL implements a substantially weaker regression mechanism than purpose-trained models.

## B.6 Mechanism Analysis: What Algorithm Does the LLM Implement?

To probe the internal mechanism of prompt-based ICL, we conduct three discriminative experiments on **Nemotron-120B** using the *noisy linear regression* task (NLR, $\sigma = 0.1$) at three dimension–context configurations: ($d$=2,$L$=6), ($d$=5,$L$=12), and ($d$=10,$L$=20). We compare the LLM's behavior against three candidate algorithms: ordinary least squares (OLS), 1-nearest neighbor (1-NN), and 5-nearest neighbor (KNN-5).

**Experiment A: Perturbation sensitivity.** For each context example $i$, we perturb $y_i$ by a fixed amount and measure the resulting change in the LLM's prediction, yielding an *influence profile* over the context. We then compute the Pearson correlation between this profile and the theoretical influence profiles of OLS and 1-NN. Results are shown in Table B.1.

At $d$=2, the LLM's influence profile correlates more strongly with OLS ($r = 0.39$) than with 1-NN ($r = 0.32$), suggesting a degree of global regression. However, as dimension increases, all correlations decay—and the correlation with 1-NN degrades more slowly than with OLS. By $d$=10, the LLM's behavior is only weakly correlated with any simple baseline ($r \lesssim 0.15$), indicating that the mechanism becomes increasingly opaque in higher dimensions.

Table B.1: Experiment A: Pearson correlation between the LLM's perturbation-sensitivity profile and the theoretical influence profiles of OLS, 1-NN, and KNN-5.

| Condition | corr(LLM, OLS) | corr(LLM, 1-NN) | corr(LLM, KNN-5) |
|---|---|---|---|
| $d$=2, $L$=6 | **0.391** | 0.317 | 0.207 |
| $d$=5, $L$=12 | 0.239 | **0.302** | 0.209 |
| $d$=10, $L$=20 | 0.084 | **0.148** | 0.067 |

**Experiment B: Extrapolation test.** We construct queries whose true $y$ lies outside the range $[\min(y_1,\ldots,y_L), \max(y_1,\ldots,y_L)]$ of the context labels, and check whether the LLM's prediction

also falls outside this range. A pure nearest-neighbor retriever can never extrapolate, so any extrapolation implies some form of regression.

Table B.2: Experiment B: Extrapolation rates and mean absolute errors (MAE). "Extrap. rate" is the fraction of episodes where the LLM predicts outside the context $y$-range.

| Condition | Extrap. rate | MAE (LLM) | MAE (OLS) | MAE (1-NN) |
|---|---|---|---|---|
| $d$=2, $L$=6 | 76.7% | 0.732 | 0.414 | 3.426 |
| $d$=5, $L$=12 | 80.0% | 1.747 | 0.539 | 3.811 |
| $d$=10, $L$=20 | 86.7% | 1.093 | 0.634 | 3.319 |

The LLM extrapolates in 77–87% of episodes—far more than a pure retriever—but with substantially higher error than OLS. This confirms that the LLM is not simply copying a retrieved label; it performs some form of regression, albeit imprecise.

**Synthesis.** Experiments A and B reveal a nuanced picture for the linear regression task. Experiment B shows that the LLM *can* extrapolate beyond the observed label range (77–87% of episodes), which is impossible for a pure retrieval mechanism such as 1-NN. Experiment A shows that the LLM's sensitivity to individual context labels is distance-dependent: at low dimensions, perturbation influence correlates with both OLS and nearest-neighbor predictions, but all correlations decay with increasing $d$.

These experiments were conducted on the linear regression task for interpretability.

## B.7 Baseline Comparison: What Algorithm Does the LLM Resemble?

To systematically identify which classical algorithm the LLM's ICL mechanism most closely resembles, we compare the per-episode predictions of Nemotron-120B against a suite of baselines: $k$-NN ($k = 1,2,3,4,5$), OLS, Ridge regression ($\lambda \in \{0.1, 1.0, 10.0\}$), and Mean-$y$. For each baseline, we compute the Pearson correlation and $R^2$ between the LLM's predictions and the baseline's predictions across 200 episodes.

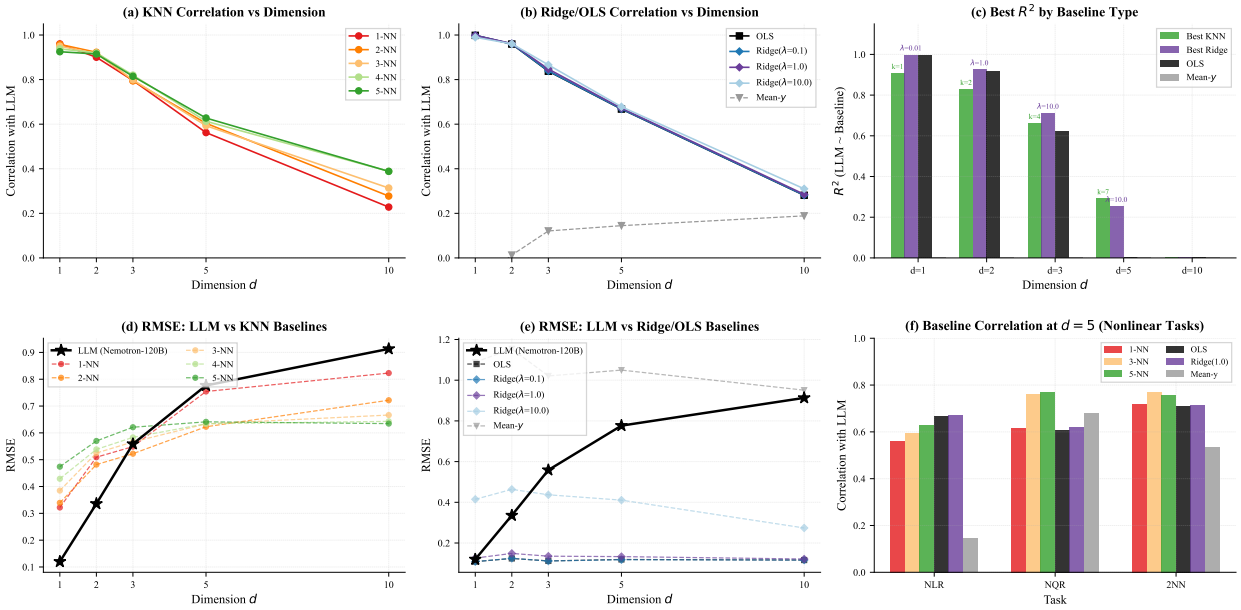



Figure B.7: Mechanism comparison for Nemotron-120B on NLR. (a,b) Correlation between LLM and baseline predictions vs. dimension. (c) Best $R^2$ for each baseline family. (d,e) RMSE comparison. (f) Correlation at $d = 5$ across tasks. At low $d$, the LLM closely matches OLS/Ridge ($r > 0.95$); at $d \geq 5$, no baseline explains the LLM well ($R^2 < 0.35$). On nonlinear tasks, $k$-NN with $k = 3$–5 provides the highest correlation.

Figure B.7 reveals a clear dimension-dependent transition:

- At $d = 1$–2, OLS and Ridge achieve correlation $> 0.95$ and $R^2 > 0.9$ with the LLM, indicating near-identical predictions. The LLM's RMSE is slightly higher than OLS, consistent with an approximate—but not exact—implementation.
- At $d = 3$–5, Ridge with larger $\lambda$ and $k$-NN with $k = 3$–5 become the best-matching baselines, but $R^2$ drops below 0.35. No single classical algorithm explains the LLM's behavior well in this regime.
- At $d = 10$, all correlations fall below 0.4 and $R^2 \approx 0$, indicating that the LLM's predictions are essentially unexplained by any simple baseline.
- For nonlinear tasks (NQR, 2NN) at $d = 5$, $k$-NN with $k = 3$–5 achieves the highest correlation ($\sim 0.75$), while OLS/Ridge correlations are lower, consistent with the LLM relying on local rather than global estimation.

## B.8 Irrelevant Feature Filtering and Effective Dimensionality

To further probe the LLM's mechanism, we design experiments that disentangle the *number of relevant features* from the *number of presented features*. We construct linear regression tasks $y = \mathbf{w}^\top \mathbf{x}_{[1:d_{\text{rel}}]} + \varepsilon$ with $d_{\text{rel}}$ relevant dimensions embedded in a $d_{\text{pres}}$-dimensional input, where the remaining $d_{\text{pres}} - d_{\text{rel}}$ features are pure noise. All experiments use $L = 20$ context examples, $n = 100$ episodes, and 5 random seeds. Each relevant coefficient is set to $w_i = 1$ to ensure equal signal strength per feature.

**Finding 1: The LLM can filter irrelevant features when $d_{\text{rel}} = 1$.** With a single relevant feature ($d_{\text{rel}} = 1$, $w = [1]$), the LLM maintains strong performance even as the number of irrelevant noise dimensions grows:

| $d_{\text{pres}}$ | 1 | 5 | 10 | 20 |
|---|---|---|---|---|
| RMSE | $0.108 \pm 0.003$ | $0.121 \pm 0.009$ | $0.156 \pm 0.026$ | $0.271 \pm 0.111$ |

Performance degrades only mildly from $d_{\text{pres}} = 1$ to 20 (RMSE: $0.108 \to 0.271$), demonstrating that the LLM can effectively identify and focus on the single informative feature amid up to 19 noise dimensions. This rules out a purely distance-based mechanism, since kernel smoothing or $k$-NN methods would be severely degraded by the noise dimensions polluting the distance metric.

**Finding 2: Feature filtering breaks down at $d_{\text{rel}} = 2$.** When two features are relevant ($d_{\text{rel}} = 2$, $w = [1, 1]$), the LLM becomes highly sensitive to irrelevant dimensions:

| $d_{\text{pres}}$ | 2 | 5 | 10 | 20 |
|---|---|---|---|---|
| RMSE | $0.110 \pm 0.004$ | $0.702 \pm 0.068$ | $1.009 \pm 0.084$ | $1.124 \pm 0.026$ |

Adding just 3 irrelevant features ($d_{\text{pres}} = 5$) causes RMSE to jump from 0.110 to 0.702—a $6\times$ degradation. The same pattern holds for $d_{\text{rel}} = 3$: RMSE rises from 0.131 ($d_{\text{pres}} = 3$) to 0.933 ($d_{\text{pres}} = 5$). Crucially, the LLM can perform 2-dimensional regression perfectly when $d_{\text{pres}} = d_{\text{rel}} = 2$

(RMSE $= 0.110$), but it cannot simultaneously perform feature selection and multi-dimensional regression.

**Finding 3: The LLM's degradation pattern reveals a mechanism transition.** To determine whether the LLM's sensitivity to irrelevant features at $d_{\text{rel}} \geq 2$ reflects a distance-based mechanism, we compare its degradation pattern against classical baselines under identical conditions ($L = 20$, $n = 200$, averaged over 3 seeds):

Table B.3: RMSE degradation when adding irrelevant features. Each row adds 3 noise dimensions. OLS and Ridge are robust; $k$-NN degrades due to distance pollution. The LLM matches Ridge at $d_{\text{rel}} = 1$ but degrades *worse than k-NN* at $d_{\text{rel}} \geq 2$.

| $d_{\text{rel}}$ | $d_{\text{pres}}$ | LLM | 1-NN | 5-NN | OLS | Ridge(1) | Degr. ratio |
|---|---|---|---|---|---|---|---|
| 1 | 1 (no noise) | 0.108 | 0.293 | 0.403 | 0.107 | 0.124 | — |
| | 4 (+3 noise) | 0.121 | 0.666 | 0.658 | 0.119 | 0.135 | LLM: 1.1×, 5-NN: 1.6× |
| 2 | 2 (no noise) | 0.110 | 0.623 | 0.729 | 0.107 | 0.138 | — |
| | 5 (+3 noise) | 0.702 | 1.033 | 0.976 | 0.124 | 0.164 | LLM: 6.4×, 5-NN: 1.3× |
| 3 | 3 (no noise) | 0.131 | 0.976 | 1.003 | 0.111 | 0.164 | — |
| | 6 (+3 noise) | 0.933 | 1.398 | 1.250 | 0.123 | 0.201 | LLM: 7.1×, 5-NN: 1.2× |

This comparison reveals a striking asymmetry:

- At $d_{\text{rel}} = 1$: the LLM degrades by only $1.1\times$ with 3 noise dimensions, matching Ridge ($1.1\times$) and far outperforming $k$-NN ($1.6$–$2.3\times$). The LLM successfully filters irrelevant features, consistent with a parametric (Ridge-like) mechanism.
- At $d_{\text{rel}} = 2$: the LLM degrades by $6.4\times$—*far worse than k-NN* ($1.3\times$) and dramatically worse than Ridge ($1.2\times$). The LLM not only fails to filter irrelevant features, but its attempted feature selection actively hurts performance compared to naive distance-based methods.
- At $d_{\text{rel}} = 3$: the pattern intensifies, with the LLM degrading $7.1\times$ while 5-NN degrades only $1.2\times$.

**Interpretation: A rank-1 projection mechanism.** The complete set of mechanism experiments—baseline comparison (Section B.7), perturbation sensitivity analysis, irrelevant feature filtering, and

the degradation analysis above—converges on a coherent picture of the LLM's ICL regression mechanism:

1. **The LLM projects data onto a single informative direction** (approximately rank-1), along which it performs effective regression. This explains why it matches OLS/Ridge at $d_{\text{rel}} = 1$ and can filter irrelevant features in that regime.
2. **When multiple directions are needed ($d_{\text{rel}} \geq 2$), the rank-1 projection fails.** The LLM cannot simultaneously identify and regress along two or more informative directions. Its attempted feature selection becomes counterproductive, leading to degradation worse than distance-based methods that make no such attempt.
3. **The mechanism is task-agnostic.** Perturbation sensitivity experiments across NLR, NQR, 2NN, and NDT tasks show nearly identical distance-dependent weight profiles (Section B.6), indicating that the LLM applies the same rank-1 projection regardless of the underlying function class.
4. **The mechanism transitions with effective dimensionality:** at $d_{\text{eff}} = 1$ it resembles Ridge regression (parametric, robust to noise features); at $d_{\text{eff}} \geq 2$ it degrades below $k$-NN (the attempted projection introduces errors worse than naive distance-based averaging); at $d_{\text{eff}} \gg 1$ it collapses toward Mean-$y$.

This rank-1 projection interpretation is consistent with recent theoretical work on ICL task vectors [Hendel et al., 2023], which shows that in-context demonstrations are compressed into low-rank representations that inherently limit the effective dimensionality of the learned mapping.

## B.9 Comparison with Trained Transformers

The mechanism analysis above provides a concrete explanation for the performance gap between pretrained LLMs and purpose-trained transformers:

1. **Dimensionality**: Trained transformers learn high-rank algorithms (e.g., Ridge regression with full-rank weight estimation), enabling strong performance up to $d = 40$. Pretrained LLMs are

constrained to rank-1 projections, limiting effective performance to $d_{\text{eff}} \approx 1$–2. This is not a limitation of the transformer architecture itself, but of the ICL mechanism that emerges from language model pretraining.

2. **Task generality**: Trained transformers achieve near-Bayes-optimal performance on their training task family. Pretrained LLMs, while never trained on regression tasks, acquire a general-purpose rank-1 regression capability during pretraining—effective for low-dimensional problems across all task families, but fundamentally limited in dimensionality.
3. **Feature selection vs. regression**: The LLM's rank-1 mechanism bundles feature selection and regression into a single operation. When these align ($d_{\text{rel}} = 1$), performance is excellent. When they conflict ($d_{\text{rel}} \geq 2$ with noise), the mechanism fails catastrophically—worse than baselines that attempt no feature selection at all.

# APPENDIX C

# Extension to Multi-Head ReLU-Attention Transformers

The ICL rates of Ching et al. [2026] are derived for single-head linear-attention transformers. Here we show that these rates extend rigorously to the multi-head ReLU-attention architecture of Song et al. [2024] via an exact block-wise embedding.

**Notation mapping.** In Ching's original paper, the number of in-context examples is denoted $n$ and the transformer depth is denoted $L$. In this thesis, $L$ denotes the context length throughout. To avoid confusion, in Theorem C.1 and its proof we use $L$ for context length (consistent with the rest of this thesis) and $D$ for the transformer depth.

**Theorem C.1** (Song–Ching multi-head rate)**.** *Let $\beta_\alpha := 2\alpha/(2\alpha+d) \in (0,1)$. Under the data model of Ching et al. [2026] with $\alpha$-Hölder regression functions on $[0,1]^d$, there exists a multi-head ReLU-attention transformer subclass $\mathscr{F}_L^{\mathrm{MH}}$ in the sense of Song et al. [2024], with at most 2 attention heads per layer, depth $D = \lceil C\log(eL)\rceil$ layers, and $\Theta(\log L)$ total parameters, such that:*

1. *(Existence) There exists $f_L^{\mathrm{MH}} \in \mathscr{F}_L^{\mathrm{MH}}$ with $R(f_L^{\mathrm{MH}}) - \sigma^2 \le C L^{-\beta_\alpha}$.*
2. *(ERM) If $\hat{f}_\Gamma^{\mathrm{MH}}$ is the empirical risk minimizer over $\mathscr{F}_L^{\mathrm{MH}}$ with $\Gamma \ge C L^{\beta_\alpha}\log^3(eL)$ pretraining sequences, then $\mathbb{E}[R(\hat{f}_\Gamma^{\mathrm{MH}})] - \sigma^2 \le C L^{-\beta_\alpha}$.*

*Proof.* The proof proceeds by showing that every Ching-type single-head linear-attention block can be exactly realized by a Song-type two-head ReLU-attention block, so the function classes are identical.

**Step 1: Block-level models.** In Ching's architecture, each transformer block consists of a single-head linear attention layer $[\mathrm{Attn}^{\mathrm{Ch}}_{Q,K,V}(Z)]_i = z_i + \sum_j \langle z_i Q, z_j K\rangle z_j V$ followed by a biased ReLU FFN: $[\mathrm{FFN}^{\mathrm{Ch}}(Z)]_i = z_i + W_2\,\mathrm{ReLU}(W_1 z_i + b_1) + b_2$. Song's architecture uses multi-head ReLU attention (with $1/N$ normalization) and unbiased token-wise MLPs.

**Step 2: Attention lift.** Augment each token $z_i \in \mathbb{R}^{d_e}$ to $\bar{z}_i := (z_i, 1) \in \mathbb{R}^{d_e+1}$. Define two heads with value matrices chosen so that $\bar{V}_1$ is the block-diagonal extension of $NV$ (absorbing Song's $1/N$ attention normalization) and $\bar{V}_2 = -\bar{V}_1$, together with $\bar{Q}_1, \bar{K}_1$ block-diagonal extensions of $Q, K$ and $\bar{Q}_2 = -\bar{Q}_1$, $\bar{K}_2 = \bar{K}_1$. Setting $u_{ij} := \langle z_i Q, z_j K \rangle$, the two-head output is

$$\frac{1}{N} \sum_j \big[\mathrm{ReLU}(u_{ij}) - \mathrm{ReLU}(-u_{ij})\big](z_j \cdot NV,\ 0) \ = \ \sum_j u_{ij}\,(z_j V,\ 0),$$

using the identity $\mathrm{ReLU}(u) - \mathrm{ReLU}(-u) = u$ together with $N \cdot \frac{1}{N} = 1$. Thus $[\mathrm{Attn}^{\mathrm{Song}}_{\bar{\vartheta}}(\bar{Z})]_i = ([\mathrm{Attn}^{\mathrm{Ch}}_{Q,K,V}(Z)]_i,\ 1)$.

**Step 3: FFN lift.** Define $\bar{W}_1 := \left(\begin{smallmatrix} W_1 & b_1 \\ 0 & 1 \end{smallmatrix}\right)$ and $\bar{W}_2 := \left(\begin{smallmatrix} W_2 & b_2 \\ 0 & 0 \end{smallmatrix}\right)$. Then $\bar{z}_i + \bar{W}_2 \mathrm{ReLU}(\bar{W}_1 \bar{z}_i) = ([\mathrm{FFN}^{\mathrm{Ch}}(Z)]_i,\ 1)$. The constant coordinate is preserved across layers.

**Step 4: Function class inclusion.** By induction over $D$ blocks, the lifted Song-type transformer computes exactly the same prediction as Ching's transformer on every prompt. Denoting by $\iota : \mathscr{F}_L^{\mathrm{Ch}} \to \mathscr{F}_L^{\mathrm{MH}}$ the block-wise lift of Steps 2–3, we have the inclusion $\iota(\mathscr{F}_L^{\mathrm{Ch}}) \subseteq \mathscr{F}_L^{\mathrm{MH}}$ as sets of functions from prompts to predictions. The reverse inclusion does not hold in general: $\mathscr{F}_L^{\mathrm{MH}}$ contains multi-head configurations that are not $\pm$-symmetric. What Step 4 establishes is that every function computable in Ching's class has an exact realization in Song's class.

**Step 5: Rate transfer.** Part (1): Ching's Theorems 2.5 and 3.1 give $f_{\mathrm{TF}} \in \mathscr{F}_L^{\mathrm{Ch}}$ with $R(f_{\mathrm{TF}}) - \sigma^2 \le CL^{-1} + CL^{-\beta\alpha} \le C'L^{-\beta\alpha}$ (the last inequality uses $\beta\alpha \le 1$). Its image $f_L^{\mathrm{MH}} := \iota(f_{\mathrm{TF}}) \in \mathscr{F}_L^{\mathrm{MH}}$ has identical risk, establishing the existence claim.

Part (2): Let $\tilde{\mathscr{F}}_L^{\mathrm{MH}} := \iota(\mathscr{F}_L^{\mathrm{Ch}}) \subseteq \mathscr{F}_L^{\mathrm{MH}}$ be the lifted subclass. The ERM over $\tilde{\mathscr{F}}_L^{\mathrm{MH}}$ is identical to the ERM over $\mathscr{F}_L^{\mathrm{Ch}}$ via the bijection $\iota$, so Ching's Theorem 3.2 transfers verbatim to give the claimed rate on $\tilde{\mathscr{F}}_L^{\mathrm{MH}}$. The ERM statement of the theorem therefore holds for $\hat{f}_\Gamma^{\mathrm{MH}}$ taken over this specific subclass; the statement does *not* make a claim about ERM over the full multi-head class $\mathscr{F}_L^{\mathrm{MH}}$, which may have strictly larger Rademacher complexity. $\square$

**Discussion.** The above lift is exact, not approximate: every Ching block becomes one Song block with 2 heads, at the cost of one extra embedding dimension and polynomial parameter scaling in $N$. The $\Theta(\log L)$ parameter count and $\Theta(L^{\beta\alpha}\log^3 L)$ pretraining requirement are preserved. We note that Song et al. independently show (Theorem 4 and Appendix D) that multi-head ReLU-attention transformers can implement convex in-context gradient descent, which is precisely the optimization subroutine used in Ching's proof for the local polynomial least-squares problem. This structural alignment confirms that the block-wise lift is not coincidental but reflects a deeper algorithmic compatibility between the two architectures.

# Bibliography


S. Garg, D. Tsipras, P. Liang, and G. Valiant. What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems*, 35, 2022.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 2020.

E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? Investigations with linear models. *International Conference on Learning Representations*, 2023.

J. von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. *International Conference on Machine Learning*, 2023.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2008.

I. Gibbs, J. J. Cherian, and E. J. Candès. Conformal prediction with conditional guarantees. *arXiv preprint arXiv:2305.12616*, 2023.

Y. J. Jung, Y. Liu, S. W. Jeong, Z. Wu, and C. Donnat. SpeedCP: Fast kernel-based conditional conformal prediction. *arXiv preprint*, 2024.

M. Ching, I. Popescu, N. Smith, T. Ma, W. G. Underwood, and R. J. Samworth. Efficient and minimax-optimal in-context nonparametric regression with transformers. *arXiv preprint arXiv:2601.15014*, 2026.

J. Kim, T. Nakamaki, and T. Suzuki. Transformers are minimax optimal nonparametric in-context learners. *Advances in Neural Information Processing Systems*, 37, 2024.

J. Wu, D. Zou, Z. Chen, V. Braverman, Q. Gu, and P. L. Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *International Conference on Learning Representations (ICLR)*, 2024.

R. Zhang, S. Frei, and P. L. Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(23):1–42, 2024.

S. Chen, H. Sheen, T. Wang, and Z. Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *Conference on Learning Theory (COLT)*, 2024.

Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in Neural Information Processing Systems*, 36, 2023.

R. Hendel, M. Geva, and A. Globerson. In-context learning creates task vectors. *Findings of EMNLP*, 2023.

Z. Song, Y. Xu, and R. Zhang. Multi-head attention is all you need: A theoretical perspective on in-context learning with multi-head transformers. *arXiv preprint*, 2024.