Square-Root Gradient Routing: GEOMETRICAL UNDERSTANDING OF ROUTER OPTIMIZATION IN SPARSE MOES

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse Mixture-of-Experts (MoE) is widely adopted to scale modern large language models. The router activates a small fraction of experts for each token, playing a critical role towards efficient and effective scaling. However, the router's dynamics during MoE training remains poorly understood. In this work, we rigorously study the routing dynamics through a tractable piecewise regression problem, where we simplify the routing with a boundary partition model. Our theory analytically proves that the loss objective is locally cubic in boundary displacement, yielding a quadratically vanishing gradient. Both gradient descent and `Adam` consequently perform polynomial convergence $O(t^{-3})$, while a *signed square-root gradient update* (`SqrtGD`) achieves geometric convergence with exponential loss decay $O(r^{3t})$. Inspired by the mechanistic understanding, we propose the *square-root gradient routing* (***SR***) on MoE-based LLM and evaluate in LLM pretraining. Our experiments on FineWeb demonstrate that the benefits from ***SR*** increase with the sparsity of routing adopted in MoE, while its effectiveness is also coupled with the decision on gating mechanism and base optimizer choices. Our findings establish a rigorous mechanistic understanding of the routing dynamics in mixture-of-experts models, which further inspires practical optimizer design for MoE-based foundation models with high sparsity.

1 INTRODUCTION

Sparse mixture-of-experts (MoE) architectures have become the standard recipe for frontier foundation models. The total parameter count grows with the number of experts, while the computation is held fixed by activating only a small subset of experts for each processed token (Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2022). Notably, a consistent trend across recent generations of MoEs demonstrates an increasingly *sparse* activation pattern: Mixtral applies a top-2 token routing among 8 experts (Jiang et al., 2024), OLMoE routes each token to 8 out of 64 experts (Muennighoff et al., 2025), while the latest DeepSeek-V4 only activate 6 out of 256 experts, with roughly 3% of the total parameters (DeepSeek-AI et al., 2026). As the activated fraction shrinks, the *router* performs an increasingly critical role. Although it is a linear map holding a negligible share of the parameters, the router determines the expert assignment for each token, which experts receive gradient signal, and hence how the model's capacity is allocated during pretraining.

Despite this central role, our understanding of how the router is optimized during MoE training remains limited. In practice, router weights are updated with the same adaptive optimizer as the experts (Kingma & Ba, 2017; Loshchilov & Hutter, 2019) combined with a series of auxiliary tricks to prevent the training instability, including load-balancing losses (Fedus et al., 2022), the router z-loss (Zoph et al., 2022), frozen or distilled routers (Dai et al., 2022), and loss-free balancing biases (Wang et al., 2024). These remedies modify the objective or the assignment rule heuristically, while none of them is designed along the router's unique training dynamics. While existing theory on MoE concerns parameter estimation (Jordan & Xu, 1995; Makkuva et al., 2019), sample complexity (Chen et al., 2022; Chowdhury et al., 2023), or feature learning under soft routing (Liao & Kyrillidis, 2025), the optimization geometry of a routing decision has not been characterized. This gap motivates our central questions:

What is the optimization geometry of a routing boundary during mixture-of-experts training?
How shall we design the optimizer for the router according to its distinct geometry?

A tractable model of routing geometry. We start from a rigorous study on a canonical piecewise linear regression problem with affine experts. The hard-routing case can be simplified with a boundary

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

optimization problem, where the trainable boundaries partition the input space and dispatch each to one corresponding affine expert (§3). Since the experts and the targets are all affine functions, alternating exact least-squares expert fitting with gradient updates on the boundary reduces the problem to an objective in the boundary displacement alone. We prove that the loss objective is *cubic* in boundary displacement, where gradient vanishes *quadratically* (Theorem 1). Consequently, gradient descent (`GD`) removes a vanishing fraction of the boundary error at each step and yields a polynomial convergence rate, with $O(t^{-3})$ loss decay (Theorem 2). `Adam` with adaptive normalization cannot repair this slowdown, retaining the $O(t^{-3})$ loss convergence (Theorem 4). In contrast, applying a *signed square root* to the boundary gradient restores a step proportional to the displacement, and the resulting update (`SqrtGD`) converges geometrically, with loss decaying in $O(r^{3t})$ (Theorem 3). This separation persists when multiple boundaries are optimized in a coupled paradigm under multi-segment target (Theorem 5, Proposition 1) and for high-dimensional input space (Remark 1).

From routing geometry to optimizer design. On the piecewise regression problem, the measured convergence exponents of `GD`, `Adam`, and `SqrtGD` match the analytical predictions (§4.1). Inspired by this mechanism understanding, we propose *square-root gradient routing* (***SR***), a one-line modification that applies the signed square root to the momentum of the router gradients, while all the other parameters keep their base optimizer (§4.2). In MoE pretraining on FineWeb with 32 experts and 1.3B total parameters, ***SR*** improves the final validation loss under both `AdamW` and `Muon` base optimizers. Remarkably, its benefit grows as routing becomes sparser. However, we note that the effect is coupled to the gating convention and the base optimizer. The benefits from ***SR*** reverses under `AdamW` with renormalized gates, which suggests that the gate mechanism and the router optimizer should be considered jointly as coupled design choices.

Contributions. We summarize our contributions as follows,

- To our knowledge, we give the first rigorous characterization of the optimization geometry of MoE routing. We prove that the routing objective is locally cubic with a quadratically vanishing gradient, where `GD` and `Adam` converge with polynomial rate $O(t^{-3})$. We further propose that applying the *signed square-root gradient update* (`SqrtGD`) accelerates the convergence with an exponential rate. The theory extends to multiple coupled boundaries and high-dimensional input space.

- Inspired by the mechanistic understanding, we propose *square-root gradient routing* (***SR***) as a router optimization algorithm compatible with any base optimizers on MoE transformers. Language-model pretraining experiments with various base optimizers (`AdamW`, `Muon`) demonstrate that the benefits from ***SR*** increase with the router sparsity. Its interaction with gate normalization exposes the need to co-design the gating mechanism and the router optimizer.

2 Related Work

Sparse MoE architectures. Conditional computation via mixtures of experts dates to Jacobs et al. (1991) and Jordan & Jacobs (1993), and was scaled to modern networks by Shazeer et al. (2017), GShard (Lepikhin et al., 2020) and Switch Transformers (Fedus et al., 2022). Contemporary LLMs adopt fine-grained variants with shared experts and top-k routing (Jiang et al., 2024; Dai et al., 2024; DeepSeek-AI et al., 2025a; Muennighoff et al., 2025). Multiple tricks are designed to improve the *training stability*. Switch Transformer introduces a load-balancing auxiliary loss and selective precision; ST-MoE (Zoph et al., 2022) adds the router z-loss to keep router logits small; expert-choice routing (Zhou et al., 2022) and BASE layers (Lewis et al., 2021) reformulate assignment as a balanced-assignment problem; DSelect-k (Hazimeh et al., 2021) and soft MoE (Puigcerver et al., 2024) smooth or remove the discreteness; auxiliary-loss-free balancing (Wang et al., 2024) replaces the balancing gradient with a bias update; and router pathologies have motivated dedicated fixes—dimension-reduced cosine routing against representation collapse (Chi et al., 2022) and frozen distilled routers against routing fluctuation (Dai et al., 2022).

Theoretical understanding of MoE. Classical analyses concern estimation: Jordan & Xu (1995) establish local convergence of joint EM for gating and expert parameters, Makkuva et al. (2019) obtain globally consistent method-of-moments algorithms, and a statistical line derives parameter-estimation rates under softmax gating (Ho et al., 2022; Nguyen et al., 2023). For gradient-based training of modern MoE, Chen et al. (2022) prove that a top-1 MoE with nonlinear experts learns cluster structure a single expert cannot; Chowdhury et al. (2023) show patch-level routing is provably sample-efficient; Dikkala et al. (2023) study when learned routing beats hashing; Kawata et al. (2025)

bound the SGD sample and runtime complexity of cluster detection; and Liao & Kyrillidis (2025) prove feature-learning guarantees for jointly trained soft-routed MoE. Complementary lines address expressivity (Wang & E, 2026), memorization versus reasoning (Jelassi et al., 2025), continual learning (Li et al., 2025), and empirical scaling laws (Clark et al., 2022; Krajewski et al., 2024), while probing studies report norm-biased expert selection and token- rather than domain-level specialization (Lo et al., 2025; Jiang et al., 2024; Muennighoff et al., 2025).

3 Boundary Geometry and Convergence Analysis

We analyze router dynamics under a piecewise linear regression setup with N target linear affine segments and k experts, defined in §3.1. We first study the minimal case with $N = k = 2$, where a single boundary determines the input partition and the corresponding expert assignments. Under the least-squares expert fitting, we show that the remaining loss is cubic in the boundary displacement (Theorem 1). On this cubic loss geometry, both gradient descent (`GD`) and `Adam` yield asymptotic $O(\frac{1}{t^3})$ convergence rate (Theorem 2, Theorem 4), while square-root gradient descent (`SqrtGD`) achieves geometric loss decay with $O(r^{3t})$ (Theorem 3) convergence rate. Section §3.3 further extends this analysis to multiple segments with $N = K > 2$ and high-dimensional input space.

3.1 Problem Setup and Preliminaries

We hereby study a hard router on the minimal piecewise linear regression problem with two affine function pieces and a single boundary. At each iteration, we alternately fit the experts on the current partition and update the routing boundary. This alternative optimization separates the error due to incomplete expert fitting and the misplacement of the boundary, i.e. the routing error.

Denote the input range as $\mathcal{X} = [x_-, x_+]$ with probability density $p(x)$ and the target boundary as $b^\star \in (x_-, x_+)$. The target consists of $N = 2$ affine pieces with separate input ranges,

$$f^\star(x) = \begin{cases} a_-^\star x + d_-^\star, & x < b^\star, \\ a_+^\star x + d_+^\star, & x \geq b^\star, \end{cases} \qquad \alpha := a_+^\star - a_-^\star. \tag{1}$$

Assumption 1 (Input distribution). *The density p is continuously differentiable and integrates to one on the input range $\mathcal{X}$, which satisfies $0 < p_{\min} \leq p(x) \leq p_{\max} < \infty$.*

Assumption 2 (Continuity). *The target pieces meet at $b^\star$, so $a_-^\star b^\star + d_-^\star = a_+^\star b^\star + d_+^\star$, and their slope gap satisfies $\alpha \neq 0$.*

The predicted boundary $b \in (x_-, x_+)$ assigns the left region $I_-(b) = [x_-, b)$ to the expert $h_-(\cdot)$ and the right region $I_+(b) = [b, x_+]$ to the other expert $h_+(\cdot)$, respectively. For $j \in \{-, +\}$, define

$$\theta_j = (d_j, a_j)^\top \in \mathbb{R}^2, \qquad \theta = (\theta_-^\top, \theta_+^\top)^\top \in \mathbb{R}^4, \qquad h_j(x; \theta_j) = d_j + a_j x,$$

where the vector $\theta = (\theta_-^\top, \theta_+^\top)^\top \in \mathbb{R}^4$ contains the expert coefficients and the boundary parameter b determines the input partitions and routing mechanism. We denote $\theta^\star$ as the target coefficients and $h_j^\star(x) = h_j(x; \theta_j^\star)$ as the corresponding affine functions. The population squared loss is

$$L(b, \theta) = \sum_{j \in \{-,+\}} \int_{I_j(b)} (f^\star(x) - h_j(x; \theta_j))^2 p(x)\, \mathrm{d}x. \tag{2}$$

Exact OLS expert fitting. Since the experts are affine functions, given a fixed boundary b, the partition gives two separate ordinary least-squares (OLS) problems with $z(x) = (1, x)^\top$,

$$\widehat{\theta}_j(b) = \arg\min_{\theta_j \in \mathbb{R}^2} \int_{I_j(b)} \left(f^\star(x) - z(x)^\top \theta_j\right)^2 p(x)\, \mathrm{d}x.$$

Define the feature and target moments on each predicted region by

$$G_j(b) = \int_{I_j(b)} z(x) z(x)^\top p(x)\, \mathrm{d}x, \qquad r_j(b) = \int_{I_j(b)} z(x) f^\star(x) p(x)\, \mathrm{d}x,$$

We can derive the OLS solution as

$$\widehat{\theta}_j(b) = G_j(b)^{-1} r_j(b), \tag{3}$$

which further gives the refitted expert function $\widehat{h}_j(x;b) = h_j(x;\widehat{\theta}_j(b))$.

With the OLS orthogonality condition, the population squared loss (Equ. 2) can be decomposed into

$$L(b,\theta) = L_E(b,\theta) + L_R(b), \tag{4}$$

$$where\ L_E(b,\theta) = \sum_j \int_{I_j(b)} \big(h_j(x;\theta_j) - \widehat{h}_j(x;b)\big)^2 p(x)\,\mathrm{d}x, \quad L_R(b) = L(b,\widehat{\theta}(b)). \tag{5}$$

L_E measures the error from incomplete expert fitting and L_R computes the loss from boundary misplacement. The complete proof is provided in Appendix A.1.

Boundary optimization. Denote the boundary displacement as $\delta = b - b^\star$. Since OLS-optimality ensures $L_E(b_t,\widehat{\theta}_t) = 0$, the piecewise regression problem can be reduced to the boundary objective

$$F(\delta) = L_R(b^\star + \delta),\ with\ F(0) = 0. \tag{6}$$

We optimize the canonical mixture-of-experts model for piecewise linear regression by alternately performing OLS expert fitting and boundary optimization. According to Algorithm 1, at step t, we iteratively fit experts following the OLS optimality $\theta_t = \widehat{\theta}(b_t)$ and update the boundary b along its gradient $g_t = L_R'(b_t) = F'(\delta_t)$. Experts are refit after each change in the boundary partition.

Algorithm 1 Alternating OLS expert fitting and boundary optimization

Require: Initial boundary b_0, learning rate η, number of steps T, router rule GD, `SqrtGD`, or Adam; population loss L.

For Adam, initialize $m_{-1} = v_{-1} = 0$ and fix $\beta_1, \beta_2, \varepsilon$.
for $t = 0, \dots, T-1$ **do**
  Form the regions $I_-(b_t) = [x_-, b_t)$ and $I_+(b_t) = [b_t, x_+]$.
  Fit $\theta_t = \widehat{\theta}(b_t)$ by Equ. (3). ▷ $L_E(b_t,\theta_t) = 0$
  Evaluate $g_t = L_R'(b_t)$ using the fitted residuals at the boundary,
  $g_t = p(b_t)\big[(f^\star(b_t) - \widehat{h}_-(b_t;b_t))^2 - (f^\star(b_t) - \widehat{h}_+(b_t;b_t))^2\big]$.
  Set $d_t = g_t$ for GD, or $d_t = \mathrm{sgn}(g_t)\sqrt{|g_t|}$ for `SqrtGD`.
  For Adam, update its moments by Equ. (14) and set $d_t = \widehat{m}_t/(\sqrt{\widehat{v}_t} + \varepsilon)$.
  Update $b_{t+1} = b_t - \eta d_t$.
end for
Fit $\theta_T = \widehat{\theta}(b_T)$ and return (b_T, θ_T).

3.2 The Cubic Boundary Loss Geometry and Convergence

We first characterize the geometry of the boundary objective $F(\delta)$ (Equ. 6), then analyze the convergence behavior of gradient descent (`GD`), `Adam`, and Square-root gradient descent (`SqrtGD`). Denote $c = \alpha^2 p(b^\star) > 0$ as a positive constant, we derive the loss, gradient and curvature of $F(\delta)$.

Theorem 1 (Cubic boundary geometry)**.** *Under Assumptions 1 and 2, as $\delta \to 0$,*

$$F(\delta) = \tfrac{c}{3}|\delta|^3 + O(|\delta|^4), \tag{7}$$

$$F'(\delta) = c\delta|\delta| + O(|\delta|^3), \tag{8}$$

$$F''(\delta) = 2c|\delta| + O(\delta^2). \tag{9}$$

The loss is locally twice continuously differentiable and strictly convex, but not strongly convex.

Theorem 1 shows that a cubic loss persists in the boundary objective, which is strictly convex but not strongly convex on any neighborhood of $\delta = 0$. Appendix A.1 provides the complete derivation on the loss and both derivatives.

Gradient descent. Furthermore, the quadratic gradient (Equ. 8) magnitude indicates that gradient descent steps update the boundary with a vanishing fraction of the remaining displacement. The resulting slowdown yields an asymptotic quadratic convergence rate of $O(t^{-3})$.

Theorem 2 (Quadratic convergence on GD). *For the objective F in Theorem 1, there exists $R > 0$ such that, for every $0 < |\delta_0| \le R$ and fixed learning rate $0 < \eta \le 1/(3cR)$, the update*

$$\delta_{t+1} = \delta_t - \eta F'(\delta_t) \tag{10}$$

in Algorithm 1 preserves sign, remains within $|\delta_t| \le R$, and satisfies

$$|\delta_t| \sim \frac{1}{\eta c t}, \qquad F(\delta_t) \sim \frac{t^{-3}}{3\eta^3 c^2}. \tag{11}$$

The boundary error decays at $O(t^{-1})$ while the loss converges at $O(t^{-3})$. The factor c also controls the speed of convergence: A smaller slope gap α or a lower input mass $p(b^\star)$ near the boundary weakens the gradients while slowing down the convergence. Complete proofs are in Appendix A.2.

Square-root gradients. The quadratic decay of the gradient motivates the square-root gradient descent (SqrtGD), which takes the signed square root of the gradient at each update. SqrtGDretains the step's magnitude to be proportional to the boundary displacement δ,

$$\delta_{t+1} = \delta_t - \eta \operatorname{sgn}(F'(\delta_t))\sqrt{|F'(\delta_t)|}, \qquad \operatorname{sgn}(0) = 0. \tag{12}$$

Theorem 3 (Geometric convergence on SqrtGD). *For the objective F in Theorem 1, let $0 < \eta\sqrt{c} < 1$ and $r = 1 - \eta\sqrt{c}$. For any $q \in (r, 1)$, sufficiently small nonzero initializations give sign-preserving iterates satisfying $|\delta_t| \le |\delta_0| q^t$ and, for some $A > 0$,*

$$|\delta_t| \sim A r^t, \qquad F(\delta_t) \sim \tfrac{cA^3}{3} r^{3t}. \tag{13}$$

Theorem 3 demonstrates that SqrtGD accelerates the loss convergence from polynomial to geometric. Its asymptotic contraction factor is $r = 1 - \eta\sqrt{c}$, giving exponential loss decay at rate $O(e^{-3\log r})$. The larger factor q provides a bound valid throughout the local trajectory. Detailed proofs are presented in Appendix A.3.

Adam. We next examine how the adaptive normalization mechanism of Adamimpacts the convergence exponent. Denote $g_t = F'(\delta_t)$, Adam (Kingma & Ba, 2017) initializes two momentum factors $m_{-1} = v_{-1} = 0$ and updates

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2,$$
$$\widehat{m}_t = m_t/(1-\beta_1^{t+1}), \qquad \widehat{v}_t = v_t/(1-\beta_2^{t+1}), \tag{14}$$
$$\delta_{t+1} = \delta_t - \eta \widehat{m}_t/(\sqrt{\widehat{v}_t} + \varepsilon). \tag{15}$$

Theorem 4 (Adam remains the quadratic convergence). *For the objective F in Theorem 1, fix $\eta, \varepsilon > 0$ and $\beta_1, \beta_2 \in [0, 1)$. From sufficiently small nonzero δ_0, Adam achieves the decay on boundary displacement and loss by*

$$|\delta_t| \sim \frac{\varepsilon}{\eta c t}, \qquad F(\delta_t) \sim \frac{\varepsilon^3}{3\eta^3 c^2} t^{-3}. \tag{16}$$

Adam can achieve faster convergence than GD at the same learning rate η due to the factor of ε^3 on its leading constant. However, it still retains the polynomial loss decay at $O(t^{-3})$. Therefore, the adaptivity of Adamdoes not remove the asymptotic slowdown in this regime. The complete proof is provided in Appendix A.4.

3.3 Multiple segments with coupled boundaries

We now extend our analysis from the single-boundary case to $N = K > 2$ affine segments and experts. The two-segment model isolates the behavior of one boundary, while in multiple-segment case the neighboring boundaries are allowed to interact through their shared expert. Specifically, we extend our notations with N segments and $N-1$ boundaries

$$x_- = b_0^\star < b_1^\star < \cdots < b_{N-1}^\star < b_N^\star = x_+, \qquad h_j^\star(x) = d_j^\star + a_j^\star x,$$

and let $f^\star = h_j^\star$ on $[b_{j-1}^\star, b_j^\star)$. The predicted boundaries are represented as a vector $\boldsymbol{b} = (b_1, \ldots, b_{N-1})$, with fixed endpoints $b_0 = x_-$ and $b_N = x_+$. Expert j receives the inputs from the

j^{th} partition $I_j(\boldsymbol{b}) = [b_{j-1}, b_j)$. Define the boundary displacement vector as $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_{N-1})$ where $\delta_j = b_j - b_j^\star$, with Euclidean norm $\|\boldsymbol{\delta}\|_2$.

Assumption 3 (Separated segments and exact expert fitting)**.** *In the ground-truth function, every input segment has length at least $\ell_{\min} > 0$. The target is continuous at every boundary $b_j^\star$, with non-zero slope gaps $\alpha_j = a_{j+1}^\star - a_j^\star \neq 0$. The density satisfies Assumption 1, with all true boundaries interior to the support. Assume the predicted boundaries are constraint within radius $\max_j |\delta_j| \leq R_0$, where $0 < R_0 < \ell_{\min}/4$. For each predicted partition, the N affine experts are fitted exactly by least-squares optimality.*

Define $L(\boldsymbol{b}, \theta) = \sum_j \int_{I_j(\boldsymbol{b})} (f^\star - h_j)^2 p \, \mathrm{d}x$ with $\theta = (\theta_1, \ldots, \theta_N)$. We yield similar form of expert and router losses on multiple boundaries

$$L_E(\boldsymbol{b}, \theta) = \sum_j \int_{I_j(\boldsymbol{b})} (h_j - \widehat{h}_j)^2 p \, \mathrm{d}x, \quad L_R(\boldsymbol{b}) = L(\boldsymbol{b}, \widehat{\theta}(\boldsymbol{b})), \quad F_N(\boldsymbol{\delta}) = L_R(\boldsymbol{b}^\star + \boldsymbol{\delta}).$$

Theorem 5 (Multiple-segment boundary geometry)**.** *Let $c_j = \alpha_j^2 p(b_j^\star) > 0$. Under Assumption 3, as $\boldsymbol{\delta} \to 0$, the multi-segment boundary objective preserves the cubic geometry,*

$$F_N(\boldsymbol{\delta}) = \sum_{j=1}^{N-1} \frac{c_j}{3} |\delta_j|^3 + O(\|\boldsymbol{\delta}\|_2^4). \tag{17}$$

The fitted loss is continuously differentiable, and

$$\nabla F_N(\boldsymbol{\delta}) = (c_j \delta_j |\delta_j|)_{j=1}^{N-1} + O(\|\boldsymbol{\delta}\|_2^3), \tag{18}$$

The vector remainder is bounded in Euclidean norm, uniformly over boundary-error directions.

The leading cubic terms add up across boundaries. The joint expert refitting introduces coupling, so the full objective is not simply reducible into independent single-boundary cases. Specifically, one correct boundary can obtain nonzero gradients if its neighboring boundaries are incorrect.

Proposition 1 (GD and square-root GD with multiple boundaries)**.** *Under Assumption 3, there exist fixed constants $R > 0$ and $\eta_{\mathrm{GD}} > 0$ such that the following hold for exact expert refitting at every iteration. Let $c_{\min} = \min_j c_j$ and $c_{\max} = \max_j c_j$. For $0 < \|\boldsymbol{\delta}_0\|_2 \leq R$ and $0 < \eta \leq \eta_{\mathrm{GD}}$, GD satisfies*

$$\|\boldsymbol{\delta}_t\|_2 = \Theta(t^{-1}), \qquad F_N(\boldsymbol{\delta}_t) = \Theta(t^{-3}).$$

For $\|\boldsymbol{\delta}_0\|_2 \leq R$ and $0 < \eta \leq 1/\sqrt{c_{\max}}$, component-wise signed square-root GD satisfies

$$\|\boldsymbol{\delta}_t\|_2 \leq \left(1 - \frac{\eta\sqrt{c_{\min}}}{2}\right)^t \|\boldsymbol{\delta}_0\|_2, \qquad F_N(\boldsymbol{\delta}_t) = O\left(\left(1 - \frac{\eta\sqrt{c_{\min}}}{2}\right)^{3t}\right).$$

The polynomial-versus-geometric loss comparison therefore survives expert coupling. These bounds do not require comparable errors across boundaries or symmetry of the segments. Theorem 4 remains a scalar result; extending its conclusion requires a separate analysis of Adam's coupled vector moments. We provide the proof in Appendix B.1 and B.2.

Remark 1 (High-dimensional inputs with a fixed routing normal). Denote a high-dimensional input $X \in \mathbb{R}^d$ and the routing follows $\boldsymbol{n}^\top X < b$, where $\boldsymbol{n}$ is a fixed unit vector and b is the trainable boundary. Assume that the two target affine functions $\boldsymbol{h}_\pm^\star(x) = \boldsymbol{d}_\pm^\star + (\boldsymbol{a}_\pm^\star)^\top \boldsymbol{x}$ are continuous on their intersection interface $\{\boldsymbol{n}^\top X = b^\star\}$. Also, the gap between two target functions along the normal direction is nonzero, i.e. $\alpha = \boldsymbol{n}^\top(\boldsymbol{a}_+^\star - \boldsymbol{a}_-^\star) \neq 0$. Under the compact-support and smooth-density conditions stated in Appendix B.3, exact refitting of full d-dimensional affine experts gives

$$F(\delta) = \frac{c_{\mathrm{H}}}{3} |\delta|^3 + O(|\delta|^4), \qquad c_{\mathrm{H}} = \alpha^2 p_S(b^\star) > 0, \qquad S = n^\top X.$$

Here p_S is the marginal density of the routing coordinate. The gradient and curvature expansions in Theorem 1, and the optimizer conclusions of Theorems 2–4, hold with $c = c_{\mathrm{H}}$. Appendix B.3 presents the complete proofs.

4 EXPERIMENTS

We first verify our theory on the piecewise linear regression setup (§4.1) with single-boundary case, multiple segments with coupled boundaries, and high-dimensional input space. We further assess the effectiveness of `SqrtGD` on real LLM pretraining (§4.2), where we critically evaluate its benefits on the transformer architecture beyond the strict hard-routing constraint.

4.1 EMPIRICAL VALIDATION ON PIECEWISE LINEAR REGRESSION

To simulate the stochasticity, we apply mini-batch procedure on all optimizers. Since the exact population density $p(x)$ is unknown, the exact boundary gradient $F'(b_i) = p(b_i)\,\Delta_i(b_i)$, where $\Delta_i(x) = (f^\star(x) - \widehat{h}_i(x))^2 - (f^\star(x) - \widehat{h}_{i+1}(x))^2$ is not applicable. We therefore apply the kernel-weighted sample mean to estimate the boundary gradient

$$\widehat{g}_i = \frac{1}{n}\sum_{k=1}^{n} K_h(x_k - b_i)\,\Delta_i(x_k), \qquad K_h(u) = \frac{1}{h}\,\sigma\Big(\frac{u}{h}\Big)\Big(1 - \sigma\Big(\frac{u}{h}\Big)\Big), \tag{19}$$

where the expectation $\int K_h(x - b_i)\Delta_i(x)p(x)\,dx$ approaches $p(b_i)\Delta_i(b_i)$ as $h \to 0$. We set $h = 10^{-4}$. All optimizers adopt the same learning rate $\eta = 0.1$. `Adam` adopts $(\beta_1, \beta_2) = (0.9, 0.95)$, $\varepsilon = 0.1$ and zero initial moments. Appendix C provides the complete implementation details, procedures and results.

Single-boundary experiments. Consider $f^\star(x) = 0.3 + 0.5x + \max\{x, 0\}$ on $\mathcal{X} = [-1/2, 1/2]$ with non-uniform density $p(x) = 1 + 0.8x$, s.t. $b^\star = 0$, $c = 1$, $r = 0.9$. Figure 1 reports the relative boundary error $|\delta_t|/|\delta_0|$ and the relative loss $L(b_t)/L(b_0)$. Notably, all three optimizers recover their analytical convergence rates. Both `GD` and `Adam` yield a polynomial convergence rate with component 3.11 ± 0.01 and 2.81 ± 0.05 against the theoretical value 3. `SqrtGD` also achieves an exponential convergence, with empirically measured exponent 0.303 ± 0.015 perfectly matches the predicted $-3\log 0.9 \approx 0.316$.

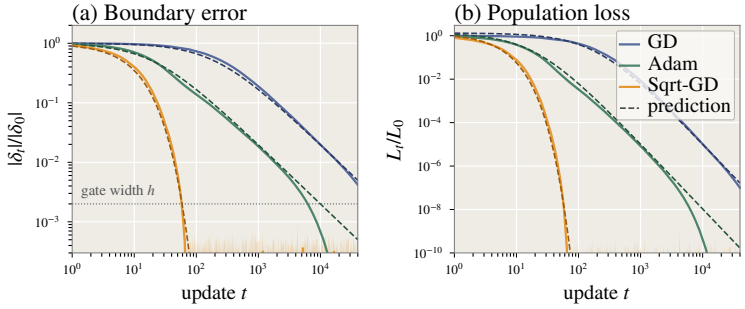

Figure 1: **Boundary dynamics on single-boundary model.** (a) Relative boundary error. (b) Relative population loss. Dashed curves denote the theoretical predicted references. $|\delta_t| = (|\delta_0|^{-1} + \eta c t)^{-1}$ for `GD`, $(|\delta_0|^{-1} + (\eta/\varepsilon)ct)^{-1}$ for `Adam`, $|\delta_0| r^t$ for `SqrtGD`; with corresponding loss references $\frac{c}{3}|\delta_t|^3$.

Multiple segments with coupled boundaries. Consider $N = K = 6$ segments on $\mathcal{X} = [0, 1]$ with $p(x) = 1 + 0.8(x - \frac{1}{2})$, the ground-truth boundaries are $\boldsymbol{b}^\star = (0.15, 0.32, 0.50, 0.66, 0.85)$. We choose the slopes to ensure $c_j = (3, 2, 1, 2.5, 5)$, where the third boundary is the slowest with $r_{\max} = 1 - \eta\sqrt{c_{\min}} = 0.9$, $r_{bound} = 1 - \frac{\eta\sqrt{c_{\min}}}{2} = 0.95$. According to Figure 4, the total loss decays at the asymptotic rate $\approx 3\log(1/r_{\max})$ set by the slowest boundary, while $3\log(1/r_{bound})$ derived from Theorem 5 serves as a loose one-sided upper bound. Figure 2 demonstrates that across 24 runs with $N \in \{2, 3, 4, 6, 8, 12\}$, the measured decay rate of the slowest boundary mostly follows $-\log r_{\max}$, which demonstrate the slowest-updating boundary largely dominates the boundary optimization under the multi-segment target setup. The complete results across various N are presented in Appendix C.2.2.

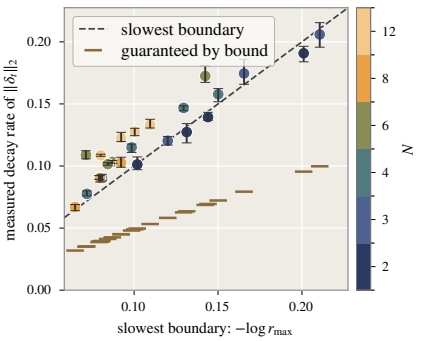

Figure 2: **Boundary error rates compared with slowest boundary predictions** across 24 multi-segment runs.

High-dimensional inputs. Consider the target $f^\star(x) = 0.2 + \boldsymbol{a}^\top \boldsymbol{x} + \max\{\boldsymbol{n}^\top \boldsymbol{x} - b^\star, 0\}$ with high-dimensional inputs $X = LY \in \mathbb{R}^d$, where Y is uniform distributed on the unit ball and $L = \Sigma^{1/2}$ with $\Sigma_{ij} = 0.5^{|i-j|}$. Along the fixed direction $\boldsymbol{n} = (1, \ldots, 1)/\sqrt{d}$, the true threshold is $b^\star = 0.1\,\sigma$ with $\sigma = \|Ln\|$. Both experts refit all $d + 1$ affine coefficients on every minibatch. Figure 6 (a) plots the empirically measured rates at dimension $d = 16$ with $c_{\mathrm{H}} = 0.91$, where the measured convergence rates matches the

theoretical prediction in Remark 1 across all three optimizers. We further conducted a sweep across $d \in \{2, 8, 32, 128, 256\}$ on `SqrtGD` and report the fitted loss decay rate in Figure 6 (b), where the measured exponent reproduces the predicted exponential rate along various c_{H}. The complete boundary error and loss curves across various d are presented in Appendix C.2.3.

4.2 Square-root Gradient Routing in LLM Pretraining

Square-root Gradient Routing on LLM. We further explore the effectiveness of `SqrtGD` on large-scale mixture-of-experts in LLM pretraining. Specifically, we apply the square-root transformation only to router weights, named square-root gradient routing (***SR***). Therefore, the `SqrtGD` only impacts $\approx 0.1\%$ of model parameters, while the other parameters are optimized by the *base optimizer*. Practically, we implement square-root gradient routing (***SR***) upon the first-order momentum,

$$u_t = 0.9u_{t-1} + 0.1g_t, \qquad W_{t+1} = W_t - \eta_t \operatorname{sgn}(u_t)\sqrt{|u_t| + 10^{-8}}, \qquad u_{-1} = 0, \tag{20}$$

where g_t denotes the router gradient after global clipping.

Training setup. We train 8-layer decoder-only transformers on FineWeb (Penedo et al., 2024). The MLP layer in each transformer block is replace by one mixture-of-experts (MoE) layer. Each expert is a $768 \to 3072 \to 768$ MLP block with GELU activation. We conducted MoE training runs with $E = 32$ experts with various *top-k* routing at 1.3B scale with 140M-360M activated parameters. All MoE models are trained with the standard load-balancing loss ($\times 0.01$) and router z-loss ($\times 0.001$) following Fedus et al. (2022) and Zoph et al. (2022). Each training run consumes 3.3B tokens over 25,000 iterations, with 2,000 warmup steps followed by cosine decay. We assess the utility of square-root gradient routing (***SR***) by combining it with two commonly used optimizers `AdamW` and `Muon`, according to the final validation log-perplexity. We provide the complete configurations in Appendix D.

Table 1: **Final validation loss** on various MoE configurations. The diverged runs are marked with (✗); the best scores are marked in **Bold**.

Experts	*top-k*	`AdamW`	+***SR***	`Muon`	+***SR***
Un-renormalized					
32	2	6.550 ✗	**3.203**	3.393	**3.367**
32	4	3.140	**3.124**	3.331	**3.318**
32	8	**3.086**	3.094	**3.292**	**3.292**
Renormalized					
32	2	**3.255**	7.192 ✗	3.356	**3.347**
32	4	**3.198**	3.610	3.303	**3.302**
32	8	**3.097**	3.098	**3.282**	3.287

Routing and gating specifics in LLM training. Our theory analyzes the routing geometry on the hard routing case, while the practical LLM pretraining commonly adopts a soft-routing mechanism, which linearly combine the outputs from $k \geq 1$ selects experts. Following Fedus et al. (2022), for router logits $\boldsymbol{s}$ and selected *top-k* expert set $\mathcal{S}$, our primary setting computes the distribution $p = \mathrm{softmax}(\boldsymbol{s})$ then combine outputs from the selected experts by $y = \sum_{i \in \mathcal{S}} p_i h_i(x)$. We denote this gating mechanism *un-renormalized*, since the selected probabilities do not sum to 1. The *renormalized* gating alternative (Jiang et al., 2023) applies a second normalization upon the k selected experts $w_i = p_i / \sum_{j \in \mathcal{S}} p_j$, which yields the output $y = \sum_{i \in \mathcal{S}} w_i h_i(x)$.

4.2.1 Results

SR* improves MoE under sparse-routing regime.** Table 1 presents the final validation losses across three MoE model configurations with various sparsity. We apply the same learning rate $\eta = 1e^{-3}$ across all configurations. With *un-renormalized* gating, introducing ***SR effectively improves the final validation loss upon both `AdamW` and `Muon` base optimizers under sparse-routing regime, while the benefit shrinks with the sparsity. With 25% experts activated at $k = 8$, adding ***SR***

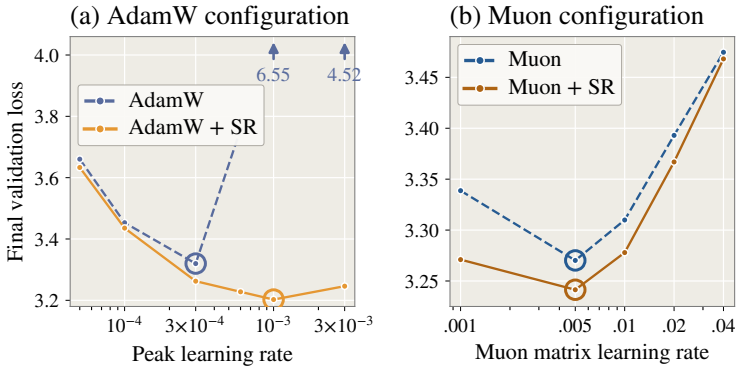

Figure 3: **Learning-rate sweep at $E = 32$ with top-2 routing.** Circles marks the optimal learning rate for each configuration.

keeps a comparable performance on both base optimizers. Under the renormalized gating mechanism, ***SR*** induces a large degradation upon `AdamW`. However, it is worth noting that `AdamW`+***SR*** with un-renormalized gating still outperforms the renormalized `AdamW` across all routing sparsity levels.

Co-design the optimizer and gate mechanism. The MoE training shows remarkably distinct behaviors between different optimizers. While sparse MoE with `AdamW`is sensitive to learning rate with a risk of divergence, the training runs with `Muon`are relatively stable across learning rates across all configurations. Also, introducing ***SR*** demonstrates opposite impacts under *re-normalized* gating mechanism. Upon `AdamW`, adding ***SR*** leads to instability and a performance degradation; while it improves upon `Muon` under sparse routing regimes and keeps a healthy training curve. Therefore, the practitioners should design the optimization and gating mechanisms in an interleaving view.

Learning rate sensitivity. We conduct a learning rate sweep on the most sparse MoE configuration with $k = 2$ and report the final validation loss in Figure 3. Across all learning rates η, adding ***SR*** improves the final validation performance upon both `AdamW` and `Muon`. On `AdamW`, the optimal learning rate without ***SR*** $\eta^\star = 3 \times 10^{-4}$ yields final validation loss at 3.320 while `AdamW`+***SR*** achieves 3.203 final loss at $\eta^\star = 1 \times 10^{-3}$, obtaining a $\Delta = 0.117$ improvement. Additionally, with ***SR*** applied on the router parameters, the sparse MoE training with `AdamW` is more stable under large learning rates. Under the `Muon` configurations, both routing optimizers obtain the lowest validation loss at $\eta^\star = 0.005$, where `Muon`+***SR*** introduces a $\Delta = 0.03$ improvement in validation loss at the optimal learning rates. We present the complete results in Appendix D.1.

5 Discussion and Limitations

Discrepancies between theoretical analysis and practical implementations. Our convergence theory is deliberately developed upon a simplified piecewise linear regression environment with linear experts, well-defined boundaries, and exact population gradients. It allows us to analyze the routing dynamics analytically. However, this ideal setup can be far from the practical large-scale MoE for LLM pretraining, where the routers process high-dimensional and evolving token representations with high stochasticity. The widely adopted soft-routing convention with gate renormalization also differs from our hard-routing partitions, which are exactly defined by the boundaries.

Theorem 3 should therefore be considered as a mechanistic motivation for practical router design on large-scale sparse MoE models, rather than as a convergence guarantee. Extending the theory to top-k soft-routing, stochastic gradients, and momentum-based power transforms is the next step toward a better understanding of the routing geometry in MoE-based foundation models.

Limited experiment scales on LLM pretraining. Our experiments on LLM pretraining train 8-layer MoE transformers with $E = 32$ experts with 1.3B total and 140M–360M activated parameters, which is at a tiny scale compared to frontier foundation models (e.g. DeepSeek V4 (DeepSeek-AI et al., 2026) with 1.6T/A49B parameters). Whether the gains of ***SR*** persist, grow, or diminish at larger scales and longer horizons remains for future work. The shrinking improvement as k increases suggests that ***SR*** is most effective under highly sparse regime, which is adopted in production by frontier foundation models. More empirical analysis on the ***SR***, including the impact on various downstream task accuracies and expert specialization, and compatibility with recent designs such as fine-grained and shared experts (Dai et al., 2024) or auxiliary-loss-free load balancing (DeepSeek-AI et al., 2025b) are left for future work.

6 Conclusion

In this work, we present the first rigorous analysis of the routing mechanism and how mixture-of-experts learns piecewise linear functions. The cubic loss geometry of routing motivates the proposed Square-root gradient Routing (***SR***), which accelerates convergence to an exponential rate for hard routing. We validate our theory on the piecewise linear regression problem, and further extend it to MoE-based LLM pretraining. Our experiments demonstrate that ***SR*** improves sparse-routing MoEs under both `AdamW` and `Muon`, and its benefit increases with the model sparsity. Our findings improve the mechanistic understanding of the routing dynamics in mixture-of-experts models, and also inspire the practical optimizer design for MoE-based foundation models with increasing sparsity.

AI DISCLOSURE

In this work, we **USED** generative AI tools for paper polishing and proof writing assistance, analyzing existing literature and coding implementation assistance. We have **NOT** used generative AI tools for formulating research or proof ideas, developing theoretical models or conceptual frameworks and synthetic data generation, synthetic evaluation are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendices A–B provide detailed mathematical derivations on the proposed theory with supported assumptions clarified. Appendices C–D provide detailed experiment protocols and setups for reproducibility including model architectures, optimizer hyperparameters, training configurations, and hyperparameter sweeps.

ETHICS STATEMENT

This work studies the optimization behavior on MoE routing mechanism. All the experiments are conducted on public available datasets. We identify no additional ethical concerns specific to this study.

REFERENCES

Zixiang Chen, Yihe Deng, Yue Wu, Quanquan Gu, and Yuanzhi Li. Towards understanding mixture of experts in deep learning, 2022. URL https://arxiv.org/abs/2208.02813.

Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. On the representation collapse of sparse mixture of experts, 2022. URL https://arxiv.org/abs/2204.09179.

Mohammed Nowaz Rabbani Chowdhury, Shuai Zhang, Meng Wang, Sijia Liu, and Pin-Yu Chen. Patch-level routing in mixture-of-experts is provably sample-efficient for convolutional neural networks, 2023. URL https://arxiv.org/abs/2306.04073.

Aidan Clark, Diego de las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, George van den Driessche, Eliza Rutherford, Tom Hennigan, Matthew Johnson, Katie Millican, Albin Cassirer, Chris Jones, Elena Buchatskaya, David Budden, Laurent Sifre, Simon Osindero, Oriol Vinyals, Jack Rae, Erich Elsen, Koray Kavukcuoglu, and Karen Simonyan. Unified scaling laws for routed language models, 2022. URL https://arxiv.org/abs/2202.01169.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. Stablemoe: Stable routing strategy for mixture of experts, 2022. URL https://arxiv.org/abs/2204.08396.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025a. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen,

Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL https://arxiv.org/abs/2412.19437.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin

Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Nishanth Dikkala, Nikhil Ghosh, Raghu Meka, Rina Panigrahy, Nikhil Vyas, and Xin Wang. On the benefits of learning to route in mixture-of-experts models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9376–9396, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.583. URL https://aclanthology.org/2023.emnlp-main.583/.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed H. Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning, 2021. URL https://arxiv.org/abs/2106.03760.

Nhat Ho, Chiao-Yu Yang, and Michael I. Jordan. Convergence rates for gaussian mixtures of experts, 2022. URL https://arxiv.org/abs/1907.04377.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi: 10.1162/neco.1991.3.1.79.

Samy Jelassi, Clara Mohri, David Brandfonbrener, Alex Gu, Nikhil Vyas, Nikhil Anand, David Alvarez-Melis, Yuanzhi Li, Sham M. Kakade, and Eran Malach. Mixture of parrots: Experts improve memorization more than reasoning, 2025. URL https://arxiv.org/abs/2410.19034.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the em algorithm. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, volume 2, pp. 1339–1344 vol.2, 1993. doi: 10.1109/IJCNN.1993.716791.

Michael I. Jordan and Lei Xu. Convergence results for the em approach to mixtures of experts architectures. *Neural Networks*, 8(9):1409–1431, 1995. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(95)00014-3. URL https://www.sciencedirect.com/science/article/pii/0893608095000143.

Ryotaro Kawata, Kohsei Matsutani, Yuri Kinoshita, Naoki Nishikawa, and Taiji Suzuki. Mixture of experts provably detect and learn the latent cluster structure in gradient-based learning, 2025. URL https://arxiv.org/abs/2506.01656.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling laws for fine-grained mixture of experts, 2024. URL https://arxiv.org/abs/2402.07871.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding, 2020. URL https://arxiv.org/abs/2006.16668.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models, 2021. URL https://arxiv.org/abs/2103.16716.

Hongbo Li, Sen Lin, Lingjie Duan, Yingbin Liang, and Ness B. Shroff. Theory on mixture-of-experts in continual learning, 2025. URL https://arxiv.org/abs/2406.16437.

Fangshuo Liao and Anastasios Kyrillidis. Guided by the experts: Provable feature learning dynamic of soft-routed mixture-of-experts, 2025. URL https://arxiv.org/abs/2510.07205.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. A closer look into mixture-of-experts in large language models, 2025. URL https://arxiv.org/abs/2406.18219.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Ashok Vardhan Makkuva, Sewoong Oh, Sreeram Kannan, and Pramod Viswanath. Breaking the gridlock in mixture-of-experts: Consistent and efficient algorithms, 2019. URL https://arxiv.org/abs/1802.07417.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. Olmoe: Open mixture-of-experts language models, 2025. URL https://arxiv.org/abs/2409.02060.

Huy Nguyen, TrungTin Nguyen, and Nhat Ho. Demystifying softmax gating function in gaussian mixture of experts, 2023. URL https://arxiv.org/abs/2305.03288.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale, 2024. URL https://arxiv.org/abs/2406.17557.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts, 2024. URL https://arxiv.org/abs/2308.00951.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, 2017. URL https://arxiv.org/abs/1701.06538.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

Mingze Wang and Weinan E. On the expressive power of mixture-of-experts for structured complex tasks, 2026. URL https://arxiv.org/abs/2505.24205.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. Mixture-of-experts with expert choice routing, 2022. URL https://arxiv.org/abs/2202.09368.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models, 2022. URL https://arxiv.org/abs/2202.08906.

CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Boundary Geometry and Convergence Analysis **3**

3.1 Problem Setup and Preliminaries 3

3.2 The Cubic Boundary Loss Geometry and Convergence 4

3.3 Multiple segments with coupled boundaries 5

4 Experiments **7**

4.1 Empirical Validation on Piecewise Linear Regression 7

4.2 Square-root Gradient Routing in LLM Pretraining 8

4.2.1 Results 8

5 Discussion and Limitations **9**

6 Conclusion **9**

A Proofs for Single-Boundary Routing Geometry on Piecewise Linear Regression **17**

A.1 Loss decomposition and cubic router geometry 17

A.2 Proof of Theorem 2 under gradient descent 22

A.3 Proof of Theorem 3 under square-root GD 23

A.4 Proof of Theorem 4 under Adam 23

B Multiple-segment loss geometry and convergence **27**

B.1 Proof of Theorem 5 27

B.2 Proof of Proposition 1 29

B.3 High-dimensional inputs and a translated hyperplane 30

C Piecewise Regression Experiments **33**

C.1 Experiment protocols. 33

C.2 Settings 34

C.2.1 Single-Boundary Experiments 34

C.2.2 Multiple Segments with Coupled Boundaries 34

C.2.3 Piecewise linear regression with High-dimensional inputs 36

D Experimental protocol and supplementary controls **39**

D.1 Complete Results on Learning Rate Ablation 39

A Proofs for Single-Boundary Routing Geometry on Piecewise Linear Regression

We present proof for each theorems in section §3.

A.1 Loss decomposition and cubic router geometry

We denote $\theta^\star$ as the ground-truth expert coefficients and the corresponding affine functions $h_j^\star(x) = h_j(x;\theta_j^\star)$. Define the loss with the ground-truth experts with wrong boundary point b and their squared distance from the refitted experts by

$$R_{\text{true}}(b) = L(b,\theta^\star), \qquad Q(b) = L_E(b,\theta^\star). \tag{21}$$

The decomposition implies

$$L_R(b) = R_{\text{true}}(b) - Q(b). \tag{22}$$

We provide the proof prove $R_{\text{true}}(b^\star+\delta) = c|\delta|^3/3 + O(|\delta|^4)$ and $Q(b^\star+\delta) = O(|\delta|^4)$, then obtain the gradient and curvature directly.

Proof of the decomposition and Theorem 1. We first justify the loss decomposition in Equ. 22 with the OLS optimal conditions. We then derive the cubic loss geometry upon this decomposition. Lastly, we differentiate the fitted loss to obtain gradient and curvature. Throughout the prove, the notations are consistent as in §3: $\delta = b - b^\star$ and $c = \alpha^2 p(b^\star)$; $x \in \mathcal{X}$ denotes the input coordinate.

Step 1. Derive the the loss decomposition in Equ. 4. Given a fixed boundary b, the OLS-optimal expert $\theta_j = (d_j, a_j)^T$ on each region $I_j(b)$ minimize the squared loss

$$\int_{I_j(b)} [f^\star(x) - d - ax]^2 p(x)\,\mathrm{d}x.$$

Differentiating with respect to d and a at the optimal solution, we have

$$\int_{I_j(b)} [f^\star(x) - \widehat{h}_j(x;b)]p(x)\,\mathrm{d}x = 0, \qquad \int_{I_j(b)} x[f^\star(x) - \widehat{h}_j(x;b)]p(x)\,\mathrm{d}x = 0. \tag{23}$$

Therefore, for any candidate affine expert h_j, expand the squared error by injecting the OLS optimal solution $\widehat{h}_j$,

$$(f^\star - h_j)^2 = (f^\star - \widehat{h}_j)^2 + (h_j - \widehat{h}_j)^2 - 2(f^\star - \widehat{h}_j)(h_j - \widehat{h}_j). \tag{24}$$

The difference $h_j - \widehat{h}_j$ is also an affine function in a form of $u + vx$. Plug in the OLS-optimal conditions in Equ. 23,

$$\int_{I_j(b)} (f^\star - \widehat{h}_j)(u+vx)p(x)\,\mathrm{d}x = u\int_{I_j(b)} (f^\star - \widehat{h}_j)p(x)\,\mathrm{d}x + v\int_{I_j(b)} x(f^\star - \widehat{h}_j)p(x)\,\mathrm{d}x = 0.$$

Integrating Equ. 24, we proved the Equ. 4

$$L(b,\theta) = \int_{I_j(b)} (f^\star - h_j)^2 = \int_{I_j(b)} [(f^\star - \widehat{h}_j)^2 + (h_j - \widehat{h}_j)^2 + 0] = L_E(b,\theta) + L_R(b). \tag{25}$$

Step 2. Calculate the loss values. The continuity of the ground-truth functions ensures

$$h_+^\star(x) - h_-^\star(x) = \alpha(x - b^\star).$$

With the experts fixed as the oracle $\theta^\star$, the model predictions are all correct except the mis-routed region between b and $b^\star$. The squared loss on the fixed oracle experts can be written as

$$R_{\text{true}}(b) = \alpha^2 \int_{\min(b,b^\star)}^{\max(b,b^\star)} (x - b^\star)^2 p(x)\,\mathrm{d}x. \tag{26}$$

According to Assumption 1, p has a continuous first derivative. Choose a fixed $r > 0$ such that $[b^\star - r, b^\star + r]$ lies inside the support. A continuous function is bounded on a closed bounded interval, so

$$C_p := \max_{u\in[b^\star - r, b^\star + r]} |p'(u)| < \infty.$$

For any x in this interval with $x \neq b^\star$, the mean value theorem gives a point ξ between x and $b^\star$ such that

$$p(x) - p(b^\star) = p'(\xi)(x - b^\star).$$

which gives

$$|p(x) - p(b^\star)| \leq C_p |x - b^\star|.$$

Split the density into its boundary value and the residue difference,

$$R_{\text{true}}(b) = \alpha^2 p(b^\star) \int_{\min(b,b^\star)}^{\max(b,b^\star)} (x - b^\star)^2 \, \mathrm{d}x + \alpha^2 \int_{\min(b,b^\star)}^{\max(b,b^\star)} (x - b^\star)^2 [p(x) - p(b^\star)] \, \mathrm{d}x.$$

The first integral is explicit. For $\delta > 0$ it is

$$\int_{b^\star}^{b^\star+\delta} (x - b^\star)^2 \, \mathrm{d}x = \left[\frac{(x - b^\star)^3}{3} \right]_{b^\star}^{b^\star+\delta} = \frac{\delta^3}{3}.$$

For $\delta < 0$, the limits are reversed and it is

$$\int_{b^\star+\delta}^{b^\star} (x - b^\star)^2 \, \mathrm{d}x = \left[\frac{(x - b^\star)^3}{3} \right]_{b^\star+\delta}^{b^\star} = -\frac{\delta^3}{3} = \frac{|\delta|^3}{3}.$$

The absolute value of the second term in the decomposition is bounded by

$$\alpha^2 C_p \int_{\min(b,b^\star)}^{\max(b,b^\star)} |x - b^\star|^3 \, \mathrm{d}x = \frac{\alpha^2 C_p}{4} |\delta|^4.$$

Recalling $c = \alpha^2 p(b^\star)$, we have

$$\boxed{R_{\text{true}}(b) = R_{\text{true}}(b^\star + \delta) = \frac{c}{3}|\delta|^3 + O(|\delta|^4).}$$

Step 3. Compute the residue loss Q after expert refitting. We first consider $\delta > 0$, so $b = b^\star + \delta > b^\star$. The right region $[b, x_+]$ matches the correct $\widehat{h}_+(x; b) = h_+^\star(x)$ with zero error. On the left region, we have

$$f^\star(x) - h_-^\star(x) = \begin{cases} 0, & x \in [x_-, b^\star), \\ \alpha(x - b^\star), & x \in [b^\star, b]. \end{cases}$$

To fit the left expert, consider the candidate line with two variables A and B

$$h_-^\star(x) + A + B(x - b^\star).$$

Denote the population loss $L(b, \theta)$ for this candidate left expert and the exact right expert by $J(A, B; \delta)$ with changed variables. Substitution gives

$$J(A, B; \delta) = \int_{x_-}^{b^\star} [A + B(x - b^\star)]^2 p(x) \, \mathrm{d}x + \int_{b^\star}^{b^\star+\delta} [\alpha(x - b^\star) - A - B(x - b^\star)]^2 p(x) \, \mathrm{d}x. \tag{27}$$

The first integral is the error on the correctly assigned interval. The second is the error on the incorrectly assigned interval. For each fixed δ, let $A(\delta), B(\delta)$ be the values that minimize J. Thus

$$\widehat{h}_-(x; b) = h_-^\star(x) + A(\delta) + B(\delta)(x - b^\star).$$

The partial derivatives of J w.r.t A and B should be zero, gives

$$\begin{aligned} \frac{1}{2} \frac{\partial J}{\partial A}(A, B; \delta) = & \int_{x_-}^{b^\star} [A + B(x - b^\star)] p(x) \, \mathrm{d}x \\ & - \int_{b^\star}^{b} [\alpha(x - b^\star) - A - B(x - b^\star)] p(x) \, \mathrm{d}x \\ = & A \int_{x_-}^{b} p(x) \, \mathrm{d}x + B \int_{x_-}^{b} (x - b^\star) p(x) \, \mathrm{d}x - \alpha \int_{b^\star}^{b} (x - b^\star) p(x) \, \mathrm{d}x \\ = & 0. \end{aligned}$$

and

$$\begin{aligned}\frac{1}{2}\frac{\partial J}{\partial B}(A,B;\delta) &= A\int_{x_-}^{b}(x-b^\star)p(x)\,\mathrm{d}x + B\int_{x_-}^{b}(x-b^\star)^2p(x)\,\mathrm{d}x \\ &\quad - \alpha\int_{b^\star}^{b}(x-b^\star)^2p(x)\,\mathrm{d}x \\ &= 0.\end{aligned}$$

Denote

$$M_k(\delta) = \int_{x_-}^{b^\star+\delta}(x-b^\star)^k p(x)\,\mathrm{d}x, \qquad k=0,1,2.$$

We simplify the partial derivatives into a linear system,

$$\begin{aligned}M_0(\delta)A(\delta) + M_1(\delta)B(\delta) &= \alpha\int_{b^\star}^{b^\star+\delta}(x-b^\star)p(x)\,\mathrm{d}x, \\ M_1(\delta)A(\delta) + M_2(\delta)B(\delta) &= \alpha\int_{b^\star}^{b^\star+\delta}(x-b^\star)^2p(x)\,\mathrm{d}x.\end{aligned} \tag{28}$$

Solve the linear system equation 28 by eliminating coefficients, we have

$$\begin{aligned}&M_0(\delta)M_2(\delta) - M_1(\delta)^2 \\ &= M_0(\delta)\left[M_2(\delta) - \frac{M_1(\delta)^2}{M_0(\delta)}\right] \\ &= M_0(\delta)\left[M_2(\delta) - 2\frac{M_1(\delta)}{M_0(\delta)}M_1(\delta) + \frac{M_1(\delta)^2}{M_0(\delta)^2}M_0(\delta)\right] \\ &= M_0(\delta)\int_{x_-}^{b^\star+\delta}\left[(x-b^\star)^2 - 2\frac{M_1(\delta)}{M_0(\delta)}(x-b^\star) + \frac{M_1(\delta)^2}{M_0(\delta)^2}\right]p(x)\,\mathrm{d}x \\ &= M_0(\delta)\int_{x_-}^{b^\star+\delta}\left[(x-b^\star) - \frac{M_1(\delta)}{M_0(\delta)}\right]^2 p(x)\,\mathrm{d}x.\end{aligned}$$

At $\delta = 0$, the interval $[x_-, b^\star]$ has positive length and $p(x) > 0$. The function $x - b^\star$ cannot equal the constant $M_1(0)/M_0(0)$ throughout that interval, so the squared integral is strictly positive. Thus $M_0(0)M_2(0) - M_1(0)^2 > 0$. Therefore, $\exists\kappa > 0$ and $\exists K < \infty$ such that, for all sufficiently small $\delta > 0$, we have,

$$M_0(\delta)M_2(\delta) - M_1(\delta)^2 \geq \kappa, \qquad |M_k(\delta)| \leq K \quad (k=0,1,2).$$

Take absolute values in the eliminated equation for $A(\delta)$, divide by the common coefficient, and use the two integral bounds after equation 28. This gives

$$\begin{aligned}|A(\delta)| &\leq \frac{|\alpha|p_{\max}}{\kappa}\left(|M_2(\delta)|\frac{\delta^2}{2} + |M_1(\delta)|\frac{\delta^3}{3}\right) \\ &\leq \frac{|\alpha|p_{\max}K}{\kappa}\left(\frac{\delta^2}{2} + \frac{\delta^3}{3}\right) \leq \frac{5|\alpha|p_{\max}K}{6\kappa}\delta^2, \qquad 0 < \delta \leq 1.\end{aligned}$$

The last inequality uses $\delta^3 \leq \delta^2$ for $0 < \delta \leq 1$. Applying the same absolute-value bounds to the eliminated equation for $B(\delta)$ gives

$$\begin{aligned}|B(\delta)| &\leq \frac{|\alpha|p_{\max}}{\kappa}\left(|M_0(\delta)|\frac{\delta^3}{3} + |M_1(\delta)|\frac{\delta^2}{2}\right) \\ &\leq \frac{|\alpha|p_{\max}K}{\kappa}\left(\frac{\delta^3}{3} + \frac{\delta^2}{2}\right) \leq \frac{5|\alpha|p_{\max}K}{6\kappa}\delta^2.\end{aligned}$$

Thus each coefficient is bounded in absolute value by a fixed constant times δ^2 for all sufficiently small positive δ. This is precisely the meaning of

$$A(\delta) = O(\delta^2), \qquad B(\delta) = O(\delta^2). \tag{29}$$

Therefore, we have,

$$Q(b) = \int_{x_-}^{b} [A(\delta) + B(\delta)(x - b^\star)]^2 p(x)\,\mathrm{d}x = O(\delta^4). \tag{30}$$

For the last equality, $|x - b^\star|$ is bounded on the support, so the quantity inside the square is at most a fixed constant times δ^2. Squaring and integrating against a probability density gives a bound of order δ^4. Combining this with previous decomposition, we derived the cubic loss

$$F(\delta) = R_{\mathrm{true}}(b) - Q(b) = \frac{c}{3}\delta^3 + O(\delta^4) \qquad (\delta > 0).$$

Step 4. Derive the Gradient. For $\delta > 0$, the right expert has zero loss, so the full fitted loss is

$$F(\delta) = J(A(\delta), B(\delta); \delta).$$

This expression depends on δ in three ways: through $A(\delta)$, through $B(\delta)$, and through the upper integration endpoint $b^\star + \delta$ in equation 27. Applying the chain rule gives

$$F'(\delta) = \underbrace{\frac{\partial J}{\partial A}(A(\delta), B(\delta); \delta)}_{=0} A'(\delta) + \underbrace{\frac{\partial J}{\partial B}(A(\delta), B(\delta); \delta)}_{=0} B'(\delta) + \frac{\partial J}{\partial \delta}(A(\delta), B(\delta); \delta).$$

The first two partial derivatives are precisely those set to zero when fitting the expert in Step 3. In the last partial derivative, the coefficients are held fixed. The fundamental theorem of calculus therefore gives

$$\frac{\partial J}{\partial \delta}(A, B; \delta) = [\alpha\delta - A - B\delta]^2 p(b^\star + \delta).$$

Consequently, defining the boundary residual by

$$d(\delta) = \alpha\delta - A(\delta) - \delta B(\delta),$$

we obtain the exact gradient formula

$$F'(\delta) = p(b^\star + \delta) d(\delta)^2. \tag{31}$$

By Step 3, $d(\delta) = \alpha\delta + O(\delta^2)$. Hence $d(\delta)^2 = \alpha^2\delta^2 + O(\delta^3)$. Multiplication by $p(b^\star + \delta) = p(b^\star) + O(\delta)$ yields

$$F'(\delta) = c\delta^2 + O(\delta^3).$$

The coefficient functions used in the chain rule are continuously differentiable: they solve the two linear equations equation 28, whose coefficients and right-hand sides are continuously differentiable and whose determinant is nonzero.

Step 5. Derive curvature. To differentiate equation 31, we need bounds on $A'(\delta)$ and $B'(\delta)$. Differentiate the first equation in equation 28, using $M_0'(\delta) = p(b)$ and $M_1'(\delta) = \delta p(b)$,

$$p(b)A(\delta) + M_0(\delta)A'(\delta) + \delta p(b)B(\delta) + M_1(\delta)B'(\delta) = \alpha\delta p(b).$$

Moving the terms without coefficient derivatives to the right gives

$$M_0(\delta)A'(\delta) + M_1(\delta)B'(\delta) = p(b)[\alpha\delta - A(\delta) - \delta B(\delta)] = p(b)d(\delta).$$

Differentiating the second equation, using $M_2'(\delta) = \delta^2 p(b)$, gives

$$\delta p(b)A(\delta) + M_1(\delta)A'(\delta) + \delta^2 p(b)B(\delta) + M_2(\delta)B'(\delta) = \alpha\delta^2 p(b),$$
$$M_1(\delta)A'(\delta) + M_2(\delta)B'(\delta) = \delta p(b)d(\delta).$$

The right-hand sides are $O(\delta)$ and $O(\delta^2)$. To isolate $A'(\delta)$, multiply the first equation by $M_2(\delta)$ and subtract $M_1(\delta)$ times the second. To isolate $B'(\delta)$, multiply the second by $M_0(\delta)$ and subtract $M_1(\delta)$ times the first. In both cases, the right-hand side is $O(\delta)$, and the coefficient of the remaining unknown is the determinant $M_0(\delta)M_2(\delta) - M_1(\delta)^2$, which is bounded away from zero. Division gives $A'(\delta) = O(\delta)$ and $B'(\delta) = O(\delta)$. It follows that

$$d'(\delta) = \alpha - A'(\delta) - B(\delta) - \delta B'(\delta) = \alpha + O(\delta).$$

The product rule applied to equation 31 now gives

$$\begin{aligned} F''(\delta) &= p'(b)d(\delta)^2 + 2p(b)d(\delta)d'(\delta) \\ &= O(\delta^2) + 2[p(b^\star) + O(\delta)][\alpha\delta + O(\delta^2)][\alpha + O(\delta)] \\ &= 2c\delta + O(\delta^2). \end{aligned}$$

We have differentiated the exact formula, so no derivative of an unspecified $O(\cdot)$ remainder is needed.

Step 6. Consider $\delta < 0$. When $\delta < 0$, the left expert is exact and the right expert receives the incorrectly assigned interval $[b, b^\star]$. On this interval $f^\star(x) - h_+^\star(x) = -\alpha(x - b^\star)$. Write the fitted right line as

$$\widehat{h}_+(x; b) = h_+^\star(x) + A_+(\delta) + B_+(\delta)(x - b^\star).$$

Repeating the coefficient differentiation in Step 3 gives

$$\left(\int_b^{x_+} p(x)\,\mathrm{d}x\right) A_+(\delta) + \left(\int_b^{x_+} (x - b^\star)p(x)\,\mathrm{d}x\right) B_+(\delta) = -\alpha \int_b^{b^\star} (x - b^\star)p(x)\,\mathrm{d}x,$$

$$\left(\int_b^{x_+} (x - b^\star)p(x)\,\mathrm{d}x\right) A_+(\delta) + \left(\int_b^{x_+} (x - b^\star)^2 p(x)\,\mathrm{d}x\right) B_+(\delta) = -\alpha \int_b^{b^\star} (x - b^\star)^2 p(x)\,\mathrm{d}x.$$

The right-hand sides are $O(|\delta|^2)$ and $O(|\delta|^3)$, and the determinant has a positive limit on the fixed interval $[b^\star, x_+]$. The elimination in Step 3 therefore proves $A_+, B_+ = O(|\delta|^2)$ and $Q(b) = O(|\delta|^4)$ on this side as well. In particular,

$$Q(b) = \int_b^{x_+} [A_+(\delta) + B_+(\delta)(x - b^\star)]^2 p(x)\,\mathrm{d}x = O(|\delta|^4).$$

Using the negative-side expansion of R_{true} established in Step 2 gives

$$F(\delta) = R_{\mathrm{true}}(b) - Q(b) = \frac{c}{3}|\delta|^3 + O(|\delta|^4), \qquad \delta < 0.$$

Thus the cubic loss expansion holds on both sides of the true boundary, without any symmetry assumption on the density or the support.

The residual at the boundary is now

$$d_+(\delta) = -\alpha\delta - A_+(\delta) - \delta B_+(\delta) = -\alpha\delta + O(\delta^2).$$

The fitted right loss has b as its lower integration endpoint. Thus the chain-rule calculation in Step 4 gives a minus sign,

$$F'(\delta) = -p(b)d_+(\delta)^2 = -c\delta^2 + O(|\delta|^3).$$

Differentiating the two coefficient equations above gives right-hand sides $-p(b)d_+(\delta)$ and $-\delta p(b)d_+(\delta)$, respectively. Their orders are $O(|\delta|)$ and $O(\delta^2)$, so $A'_+, B'_+ = O(|\delta|)$ and $d'_+ = -\alpha + O(|\delta|)$. Consequently,

$$F''(\delta) = -p'(b)d_+(\delta)^2 - 2p(b)d_+(\delta)d'_+(\delta) = -2c\delta + O(\delta^2) = 2c|\delta| + O(\delta^2).$$

Combining both sides establishes all three expansions in Theorem 1. □

A.2 Proof of Theorem 2 under gradient descent

Proof. At each iteration, the least-squared expert fitting gives $L_E(b_t, \theta_t) = 0$ and $L(b_t, \theta_t) = L_R(b_t) = F(\delta_t)$. We analyze the GD update $\delta_{t+1} = \delta_t - \eta F'(\delta_t)$.

According to Theorem 1, the gradient gives

$$\frac{F'(\delta)}{\delta|\delta|} = c + O(|\delta|) \longrightarrow c > 0 \qquad (\delta \to 0,\ \delta \neq 0).$$

there exists a fixed $R > 0$, on which $\frac{F'(\delta)}{\delta|\delta|}$ lies between $c/2$ and $3c/2$. The interval $[b^\star - R, b^\star + R]$ can also be taken inside the input support. Thus, whenever $0 < |\delta| \leq R$,

$$\frac{c}{2}|\delta|^2 \leq |F'(\delta)| \leq \frac{3c}{2}|\delta|^2. \tag{32}$$

Assume $0 < |\delta_0| \leq R$ and take a fixed learning rate satisfying

$$0 < \eta \leq \frac{1}{3cR}.$$

Here R specifies the region where the local gradient bounds hold, and the learning-rate condition ensures that GD stays in that region.

If $0 < |\delta_t| \leq R$, the gradient has the sign of δ_t, so subtracting it moves the boundary toward zero. The learning-rate condition gives

$$\eta|F'(\delta_t)| \leq \frac{3\eta c}{2}|\delta_t|^2 \leq \frac{3\eta cR}{2}|\delta_t| \leq \frac{1}{2}|\delta_t|.$$

Thus the step cannot reach or cross zero. Sign preservation gives

$$|\delta_{t+1}| = |\delta_t| - \eta|F'(\delta_t)|, \qquad \frac{1}{2}|\delta_t| \leq |\delta_{t+1}| < |\delta_t| \leq R.$$

Take the difference of consecutive reciprocal errors,

$$\frac{1}{|\delta_{t+1}|} - \frac{1}{|\delta_t|} = \frac{|\delta_t| - |\delta_{t+1}|}{|\delta_t||\delta_{t+1}|} = \frac{\eta|F'(\delta_t)|}{|\delta_t||\delta_{t+1}|}.$$

For a lower bound, the numerator is at least $(\eta c/2)|\delta_t|^2$ and the denominator is at most $|\delta_t|^2$. For an upper bound, the numerator is at most $(3\eta c/2)|\delta_t|^2$ and the denominator is at least $|\delta_t|^2/2$. Therefore

$$\frac{\eta c}{2} \leq \frac{1}{|\delta_{t+1}|} - \frac{1}{|\delta_t|} \leq 3\eta c.$$

Adding these inequalities over iterations $0, \ldots, t-1$ cancels the intermediate reciprocal errors and yields

$$|\delta_0|^{-1} + \frac{\eta c}{2}t \leq |\delta_t|^{-1} \leq |\delta_0|^{-1} + 3\eta ct.$$

All terms are positive, so taking reciprocals reverses the inequalities,

$$\frac{1}{|\delta_0|^{-1} + 3\eta ct} \leq |\delta_t| \leq \frac{1}{|\delta_0|^{-1} + (\eta c/2)t}.$$

Therefore, we prove $|\delta_t| \to 0$ and $|\delta_t| = \Theta(t^{-1})$. Combined with the cubic loss from Theorem 1 derives $F(\delta_t) = \Theta(t^{-3})$.

Plug in the gradient formula from Theorem 1, we have

$$|F'(\delta_t)| = c|\delta_t|^2 + O(|\delta_t|^3).$$

Since $|\delta_t| \to 0$, the exact update implies

$$\frac{|\delta_{t+1}|}{|\delta_t|} = 1 - \eta\frac{|F'(\delta_t)|}{|\delta_t|} = 1 - \eta c|\delta_t| + O(|\delta_t|^2) \longrightarrow 1.$$

Returning to the reciprocal-error identity, we obtain

$$\frac{1}{|\delta_{t+1}|} - \frac{1}{|\delta_t|} = \eta \frac{|F'(\delta_t)|/|\delta_t|^2}{|\delta_{t+1}|/|\delta_t|} \longrightarrow \eta c. \tag{33}$$

Summing this identity and dividing by t gives

$$\frac{|\delta_t|^{-1}}{t} = \frac{|\delta_0|^{-1}}{t} + \frac{1}{t}\sum_{s=0}^{t-1}\left(\frac{1}{|\delta_{s+1}|} - \frac{1}{|\delta_s|}\right) \longrightarrow \eta c.$$

To justify the last limit, each summand approaches ηc. For any prescribed tolerance, all but finitely many summands lie within that tolerance of ηc; the finitely many earlier terms contribute zero in the limit after division by t. Hence their average also approaches ηc. It follows that

$$|\delta_t| \sim \frac{1}{\eta c t}, \qquad F(\delta_t) = \frac{c}{3}|\delta_t|^3(1 + O(|\delta_t|)) \sim \frac{1}{3\eta^3 c^2} t^{-3},$$

which proves equation 11. If $\delta_0 = 0$, the gradient is zero and every iterate remains at zero. □

A.3 Proof of Theorem 3 under square-root GD

Proof. With $L_E(b_t, \theta_t) = 0$ and $L(b_t, \theta_t) = F(\delta_t)$. The gradient expansion and its local sign imply

$$\begin{aligned}\operatorname{sgn}(F'(\delta))\sqrt{|F'(\delta)|} &= \operatorname{sgn}(\delta)\sqrt{c|\delta|^2\big(1 + O(|\delta|)\big)} \\ &= \sqrt{c}\,\delta\sqrt{1 + O(|\delta|)} \\ &= \sqrt{c}\,\delta\big(1 + O(|\delta|)\big).\end{aligned}$$

Thus the update can be written as

$$\delta_{t+1} = \delta_t \Phi(\delta_t), \qquad \Phi(\delta) = r + O(|\delta|), \qquad r = 1 - \eta\sqrt{c}. \tag{34}$$

Fix $q \in (r, 1)$ and choose the radius so that $|\Phi(\delta) - r| \le \min\{r/2, q - r\}$. Then

$$0 < r/2 \le \Phi(\delta) \le q < 1.$$

The update preserves sign and nonzero errors, the radius is invariant, and induction gives $|\delta_t| \le |\delta_0| q^t$. In particular $|\delta_t| \to 0$ and $|\delta_{t+1}|/|\delta_t| = \Phi(\delta_t) \to r$.

To obtain the exact asymptotic, put $u_t = \Phi(\delta_t)/r - 1 = O(|\delta_t|)$. The geometric bound implies $\sum_t |u_t| < \infty$, and each $1 + u_t$ is positive and bounded away from zero. The series $\sum_t \log(1 + u_t)$ therefore converges to a finite real number, so

$$\frac{|\delta_t|}{r^t} = |\delta_0| \prod_{s=0}^{t-1}(1 + u_s) \longrightarrow A \in (0, \infty).$$

This proves the stated boundary asymptotic. Applying the cubic loss from Theorem 1 gives $F(\delta_t) = (cA^3/3)r^{3t}(1 + o(1))$ and the loss-ratio limit r^3. Therefore, we yield $r^{3t} = \exp(-\lambda t)$ with $\lambda = -3\log r$. □

A.4 Proof of Theorem 4 under Adam

Proof. After every exact expert fit, $L_E = 0$ and the gradient is $g_t = F'(\delta_t)$. Adam uses

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t, & \widehat{m}_t &= \frac{m_t}{1 - \beta_1^{t+1}}, \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, & \widehat{v}_t &= \frac{v_t}{1 - \beta_2^{t+1}}, \\ \delta_{t+1} &= \delta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \varepsilon}, & m_{-1} &= v_{-1} = 0. \end{aligned}$$

Here $\eta, \varepsilon > 0$ and $0 \le \beta_1, \beta_2 < 1$ are fixed. We first ensure that every update moves toward zero without crossing it. We then prove that $\widehat{m}_t/g_t \to 1$ and $\widehat{v}_t \to 0$, and use these limits to calculate the rate.

Step 1. State the local gradient bounds and the learning-rate condition. Fix a neighborhood $|\delta| \le R$ where Theorem 1 gives

$$\operatorname{sgn}(F'(\delta)) = \operatorname{sgn}(\delta), \qquad \frac{c}{2}|\delta|^2 \le |F'(\delta)| \le \frac{3c}{2}|\delta|^2 \qquad (0 < |\delta| \le R). \tag{35}$$

Such an $R > 0$ exists because $F'(\delta)/(\delta|\delta|) \to c > 0$, as explained in Appendix A.2. The sharper expansion is $|F'(\delta)| = c|\delta|^2(1 + O(|\delta|))$.

Define the fixed number

$$q = \frac{1 + \sqrt{\beta_1}}{2}, \qquad \sqrt{\beta_1} < q < 1. \tag{36}$$

The inequality $q > \sqrt{\beta_1}$ ensures that the geometric series with ratio β_1/q^2 converges; this series bounds the contribution of past gradients to the first moment.

Assume $0 < |\delta_0| \le R$ and impose the learning-rate condition

$$0 < \eta \le \frac{2(1-q)(1-\beta_1/q^2)\varepsilon}{3c|\delta_0|}. \tag{37}$$

For a fixed learning rate, this is equivalent to

$$|\delta_0| \le \frac{2(1-q)(1-\beta_1/q^2)\varepsilon}{3\eta c}.$$

Thus it proves the theorem's claim for every fixed $\eta > 0$ from sufficiently small nonzero initialization. If a bound independent of the particular initialization is preferred, replacing $|\delta_0|$ by R in equation 37 gives a sufficient learning-rate condition for every $0 < |\delta_0| \le R$.

Step 2. Bound the first moment and prevent crossing zero. Repeated substitution in the first-moment recurrence, starting with $m_{-1} = 0$, gives

$$\widehat{m}_t = \frac{1-\beta_1}{1-\beta_1^{t+1}} \sum_{k=0}^{t} \beta_1^k g_{t-k}. \tag{38}$$

These weights are nonnegative and sum to one, since

$$\frac{1-\beta_1}{1-\beta_1^{t+1}} \sum_{k=0}^{t} \beta_1^k = 1.$$

If $\beta_1 = 0$, this formula means $\widehat{m}_t = g_t$; only the current-gradient term is present.

We prove by induction that the boundary keeps its initial sign and

$$q|\delta_t| \le |\delta_{t+1}| < |\delta_t|. \tag{39}$$

Suppose this has been established for all updates before time t. Then $|\delta_i| \le |\delta_0| \le R$ for $i \le t$, and all past gradients have the same sign as δ_t. Their weighted average $\widehat{m}_t$ has that sign too. Moreover, applying the lower ratio bound repeatedly gives

$$|\delta_{t-k}| \le q^{-k}|\delta_t| \qquad (0 \le k \le t).$$

There is no cancellation among the same-sign gradients, so taking absolute values in equation 38 and using equation 35 yields

$$\begin{aligned} |\widehat{m}_t| &= \frac{1-\beta_1}{1-\beta_1^{t+1}} \sum_{k=0}^{t} \beta_1^k |g_{t-k}| \\ &\le \frac{3c}{2} \sum_{k=0}^{t} \beta_1^k |\delta_{t-k}|^2 \le \frac{3c}{2}|\delta_t|^2 \sum_{k=0}^{t} (\beta_1/q^2)^k. \end{aligned}$$

The first inequality uses $(1-\beta_1)/(1-\beta_1^{t+1}) \le 1$. Bounding the finite sum by its convergent infinite series gives

$$0 < |\widehat{m}_t| \le \frac{3c}{2(1-\beta_1/q^2)}|\delta_t|^2. \tag{40}$$

At $t = 0$, the same bound holds directly because $\widehat{m}_0 = g_0$ and $3c/[2(1 - \beta_1/q^2)] \geq 3c/2$.

The Adam denominator is at least ε. Therefore its step has magnitude at most

$$\eta \frac{|\widehat{m}_t|}{\sqrt{\widehat{v}_t} + \varepsilon} \leq \frac{3\eta c}{2\varepsilon(1 - \beta_1/q^2)} |\delta_t|^2 \leq \frac{3\eta c|\delta_0|}{2\varepsilon(1 - \beta_1/q^2)} |\delta_t| \leq (1 - q)|\delta_t|.$$

The step is strictly positive in magnitude and points toward zero. Since $1 - q < 1$, it cannot reach or cross zero. Thus

$$|\delta_{t+1}| = |\delta_t| - \eta \frac{|\widehat{m}_t|}{\sqrt{\widehat{v}_t} + \varepsilon} \in [q|\delta_t|, |\delta_t|).$$

This establishes the induction step, including its base at $t = 0$. It proves sign preservation and equation 39 for every iteration, so all iterates remain in the fixed local neighborhood.

Step 3. Prove convergence and compare consecutive errors. The weight of the current gradient in equation 38 is at least $1 - \beta_1$. Because all terms have the same sign,

$$|\widehat{m}_t| \geq (1 - \beta_1)|g_t| \geq \frac{(1 - \beta_1)c}{2} |\delta_t|^2.$$

Unrolling the second-moment recurrence gives

$$\widehat{v}_t = \frac{1 - \beta_2}{1 - \beta_2^{t+1}} \sum_{i=0}^{t} \beta_2^{t-i} g_i^2. \tag{41}$$

Its weights also sum to one. Since $|\delta_i| \leq |\delta_0|$, every g_i^2 is at most $(3c/2)^2|\delta_0|^4$. Consequently

$$0 \leq \widehat{v}_t \leq (3c/2)^2|\delta_0|^4, \qquad \sqrt{\widehat{v}_t} \leq (3c/2)|\delta_0|^2.$$

Combining these bounds with the upper bound from Step 2 yields

$$\frac{\eta(1 - \beta_1)c}{2[\varepsilon + (3c/2)|\delta_0|^2]} |\delta_t|^2 \leq |\delta_t| - |\delta_{t+1}| \leq \frac{3\eta c}{2\varepsilon(1 - \beta_1/q^2)} |\delta_t|^2. \tag{42}$$

Divide the lower inequality by $|\delta_t||\delta_{t+1}| \leq |\delta_t|^2$. The resulting reciprocal-error increment is bounded below by a fixed positive number. Summing over the first t iterations gives

$$\frac{1}{|\delta_t|} \geq \frac{1}{|\delta_0|} + t \frac{\eta(1 - \beta_1)c}{2[\varepsilon + (3c/2)|\delta_0|^2]}.$$

The right-hand side tends to infinity, proving $|\delta_t| \to 0$. Dividing the upper inequality in equation 42 by $|\delta_t|$ now gives

$$0 \leq 1 - \frac{|\delta_{t+1}|}{|\delta_t|} \leq \frac{3\eta c}{2\varepsilon(1 - \beta_1/q^2)} |\delta_t| \longrightarrow 0. \tag{43}$$

Thus $|\delta_{t+1}|/|\delta_t| \to 1$. For every fixed integer $k \geq 0$, multiplying finitely many consecutive ratios gives $|\delta_{t-k}|/|\delta_t| \to 1$. The gradient expansion then implies

$$\frac{|g_{t-k}|}{|g_t|} = \left(\frac{|\delta_{t-k}|}{|\delta_t|} \right)^2 \frac{1 + O(|\delta_{t-k}|)}{1 + O(|\delta_t|)} \longrightarrow 1.$$

This says that each fixed number of recent gradients becomes nearly equal to the current gradient.

Step 4. Prove that the first moment tracks the current gradient. Recent-gradient agreement alone does not control the entire moving average, which also contains older gradients. We bound their contribution explicitly. From equation 35 and equation 39,

$$\frac{|g_{t-k}|}{|g_t|} \leq \frac{(3c/2)|\delta_{t-k}|^2}{(c/2)|\delta_t|^2} \leq 3q^{-2k}.$$

Since the weights in equation 38 sum to one,

$$\left| \frac{|\widehat{m}_t|}{|g_t|} - 1 \right| \leq \frac{1 - \beta_1}{1 - \beta_1^{t+1}} \sum_{k=0}^{t} \beta_1^k \left| \frac{|g_{t-k}|}{|g_t|} - 1 \right|.$$

Fix an integer $K \geq 0$ and split this sum at $k = K$. For $k = 0, \ldots, K$, every ratio tends to one by Step 3, so this finite part tends to zero as $t \to \infty$. For the remaining terms, use the prefactor bound by one and the ratio bound above,

$$\frac{1-\beta_1}{1-\beta_1^{t+1}} \sum_{k=K+1}^{t} \beta_1^k \left| \frac{|g_{t-k}|}{|g_t|} - 1 \right| \leq \sum_{k=K+1}^{\infty} \left[3(\beta_1/q^2)^k + \beta_1^k \right] = \frac{3(\beta_1/q^2)^{K+1}}{1-\beta_1/q^2} + \frac{\beta_1^{K+1}}{1-\beta_1}.$$

Both ratios are less than one, so this bound can be made arbitrarily small by taking K large, independently of t. For that fixed K, the finite part can then be made arbitrarily small by taking t large. This proves $|\widehat{m}_t|/|g_t| \to 1$. Their signs coincide, so also $\widehat{m}_t/g_t \to 1$. When $\beta_1 = 0$, the conclusion holds exactly at every iteration because $\widehat{m}_t = g_t$.

Step 5. Prove that the second moment tends to zero. We have $g_t \to 0$ because $\delta_t \to 0$ and $|g_t| \leq (3c/2)|\delta_t|^2$. Fix any $\tau > 0$. There is an index N such that $g_i^2 \leq \tau$ for every $i \geq N$. In equation 41, the total weight of those terms is at most one. Hence, for $t \geq N$,

$$0 \leq \widehat{v}_t \leq \frac{1-\beta_2}{1-\beta_2^{t+1}} \sum_{i=0}^{N-1} \beta_2^{t-i} g_i^2 + \tau.$$

For $0 < \beta_2 < 1$, each of the finitely many terms in the displayed sum tends to zero as $t \to \infty$. Thus $\limsup_t \widehat{v}_t \leq \tau$. Since τ was arbitrary and $\widehat{v}_t \geq 0$, this proves $\widehat{v}_t \to 0$. If $\beta_2 = 0$, then $\widehat{v}_t = g_t^2 \to 0$ directly. In either case,

$$\sqrt{\widehat{v}_t} + \varepsilon \longrightarrow \varepsilon > 0.$$

No estimate of $\widehat{v}_t/g_t^2$ is needed.

Step 6. Calculate the exact boundary and loss rates. The exact update and the preserved sign give

$$\frac{1}{|\delta_{t+1}|} - \frac{1}{|\delta_t|} = \frac{\eta|\widehat{m}_t|}{(\sqrt{\widehat{v}_t} + \varepsilon)|\delta_t||\delta_{t+1}|} = \eta \frac{|\widehat{m}_t|}{|g_t|} \frac{|g_t|}{|\delta_t|^2} \frac{1}{\sqrt{\widehat{v}_t} + \varepsilon} \frac{|\delta_t|}{|\delta_{t+1}|} \longrightarrow \frac{\eta c}{\varepsilon}.$$

The four factors tend to 1, c, $1/\varepsilon$, and 1, respectively, by Steps 3–5 and the gradient expansion. Summing consecutive reciprocal differences and dividing by t therefore yields

$$\frac{1}{t|\delta_t|} = \frac{1}{t|\delta_0|} + \frac{1}{t} \sum_{i=0}^{t-1} \left(\frac{1}{|\delta_{i+1}|} - \frac{1}{|\delta_i|} \right) \longrightarrow \frac{\eta c}{\varepsilon}.$$

Here the average has the same limit as its summands: the finitely many early terms vanish after division by t, and every sufficiently late term is arbitrarily close to $\eta c/\varepsilon$. Taking reciprocals gives

$$|\delta_t| \sim \frac{\varepsilon}{\eta c t}.$$

Finally, Theorem 1 gives the full loss after each expert fit,

$$L(b_t, \theta_t) = F(\delta_t) = \frac{c}{3}|\delta_t|^3(1 + O(|\delta_t|)) \sim \frac{\varepsilon^3}{3\eta^3 c^2} t^{-3}.$$

□

B MULTIPLE-SEGMENT LOSS GEOMETRY AND CONVERGENCE

B.1 PROOF OF THEOREM 5

As in Appendix A.1, define $R_{\text{true}}(\boldsymbol{b}) = L(\boldsymbol{b}, \theta^\star)$ and $Q(\boldsymbol{b}) = L_E(\boldsymbol{b}, \theta^\star)$. We establish $L_R(\boldsymbol{b}) = R_{\text{true}}(\boldsymbol{b}) - Q(\boldsymbol{b})$ and the expansions

$$R_{\text{true}}(\boldsymbol{b}) = \sum_{j=1}^{N-1} \frac{c_j}{3} |\delta_j|^3 + O(\|\boldsymbol{\delta}\|_2^4), \qquad Q(\boldsymbol{b}) = O(\|\boldsymbol{\delta}\|_2^4). \tag{44}$$

Proof. Given $N = K \geq 2$ segments and experts. Let $z(x) = (1, x)^\top$ and $\theta_j = (d_j, a_j)^\top$.

Step 1. Show that each predicted expert region has a stable least-squares fit. According to Assumption 3, every boundary moves by at most R_0, the predicted region $I_j(\boldsymbol{b})$ contains the fixed interval

$$[b_{j-1}^\star + R_0, b_j^\star - R_0],$$

whose length is at least $\ell_{\min} - 2R_0 > 0$. For the first and last experts this is still an interior subinterval of their predicted regions. The coefficient matrix of the fit is

$$G_j(\boldsymbol{b}) = \int_{I_j(\boldsymbol{b})} z(x) z(x)^\top p(x)\, \mathrm{d}x.$$

For a unit vector v, its quadratic form is the integral of $(v^\top z(x))^2 p(x)$. Restricting that integral to the fixed subinterval above gives a strictly positive value, since a nonzero affine function cannot vanish on an interval. This restricted quadratic form is continuous in v and attains a positive minimum over the compact unit circle. There are finitely many experts, so a common $\gamma > 0$ satisfies

$$v^\top G_j(\boldsymbol{b}) v \geq \gamma \|v\|_2^2 \quad \text{for every } j, \boldsymbol{b}, v.$$

Thus every G_j is invertible, $\|G_j^{-1}\|_2 \leq 1/\gamma$, and the exact fit is unique. The matrices G_j also have uniformly bounded norms because the support, z, and p are bounded.

Step 2. Establish the exact decomposition. The derivatives of the expert loss with respect to its two coefficients give

$$\int_{I_j(\boldsymbol{b})} z(x)[f^\star(x) - \widehat{h}_j(x)] p(x)\, \mathrm{d}x = 0.$$

Follow the same derivation in A.1, we have

$$\int_{I_j} (f^\star - h_j)^2 p\, \mathrm{d}x = \int_{I_j} (f^\star - \widehat{h}_j)^2 p\, \mathrm{d}x + \int_{I_j} (h_j - \widehat{h}_j)^2 p\, \mathrm{d}x.$$

Summing over experts proves $L = L_R + L_E$. Substituting $h_j = h_j^\star$ proves $R_{\text{true}} = L_R + Q$. This argument applies to any nondegenerate partition; the local boundary restriction is used for the quantitative bounds that follow.

Step 3. Add the true-expert losses from the disjoint boundary intervals. The intervals between b_j and $b_j^\star$ do not overlap because $R_0 < \ell_{\min}/4$. On each such interval the predicted expert is an adjacent true expert. Continuity gives

$$h_{j+1}^\star(x) - h_j^\star(x) = \alpha_j (x - b_j^\star).$$

There is no true-expert prediction error outside these intervals, so

$$R_{\text{true}}(\boldsymbol{b}) = \sum_{j=1}^{N-1} \alpha_j^2 \int_{\min(b_j, b_j^\star)}^{\max(b_j, b_j^\star)} (x - b_j^\star)^2 p(x)\, \mathrm{d}x.$$

For each term, the density bound proved in Step 2 of Appendix A.1 gives a leading term $c_j |\delta_j|^3/3$ and an error bounded by a constant times $|\delta_j|^4$. There are finitely many boundaries, and $\sum_j |\delta_j|^4 \leq \|\boldsymbol{\delta}\|_2^4$. Hence the first expansion in equation 44 follows.

Step 4. Bound the change in every fitted expert and the refitting gain. Write $u_j = \widehat{\theta}_j(\boldsymbol{b}) - \theta_j^\star$, the two coefficient changes of expert j. Subtracting $G_j\theta_j^\star$ from its fitting equation gives the exact linear system

$$G_j(\boldsymbol{b})u_j = \int_{I_j(\boldsymbol{b})} z(x)[f^\star(x) - h_j^\star(x)]p(x)\,\mathrm{d}x.$$

The right-hand side is zero on the part of I_j inside the true segment j. It can receive contributions only near its left or right true boundary. At either endpoint, continuity bounds the target discrepancy by a fixed constant times the distance from that endpoint. Its integral over a wrongly assigned interval of length $|\delta_i|$ is therefore at most a constant times δ_i^2, since $\int_0^{|\delta_i|} s\,\mathrm{d}s = \delta_i^2/2$. Boundedness of z and p then gives

$$\left\| \int_{I_j(\boldsymbol{b})} z(x)[f^\star(x) - h_j^\star(x)]p(x)\,\mathrm{d}x \right\|_2 \le C(\delta_{j-1}^2 + \delta_j^2).$$

Here $\delta_0 = \delta_N = 0$ denote the fixed support endpoints, and C is a constant independent of the boundary errors. Multiplication by G_j^{-1} yields

$$\|u_j\|_2 \le \frac{C}{\gamma}(\delta_{j-1}^2 + \delta_j^2) \le \frac{C}{\gamma}\|\boldsymbol{\delta}\|_2^2.$$

The squared change in each fitted line is $u_j^\top G_j u_j$. Thus

$$0 \le Q(\boldsymbol{b}) = \sum_{j=1}^{N} u_j^\top G_j u_j \le C' \sum_{j=1}^{N} (\delta_{j-1}^2 + \delta_j^2)^2 \le 4C'\|\boldsymbol{\delta}\|_2^4.$$

The last inequality uses $(a+b)^2 \le 2a^2 + 2b^2$ and counts each internal boundary twice. This proves the second expansion in equation 44. Subtracting Q from the true-expert loss proves equation 17.

Step 5. Differentiate the exact fitted loss. The entries of G_j and of the fitting right-hand side $\int_{I_j} z f^\star p\,\mathrm{d}x$ are continuously differentiable in the endpoints: their integrands are continuous, including at the true boundaries. Their inverses remain nonsingular, so each fitted coefficient vector is continuously differentiable as well.

When differentiating $L(\boldsymbol{b}, \widehat{\theta}(\boldsymbol{b}))$ with respect to b_i, the chain-rule terms involving fitted coefficients vanish because every coefficient derivative of L is zero at its fit. Only the two integration endpoints adjoining b_i remain. Therefore

$$\frac{\partial F_N}{\partial \delta_i} = p(b_i)\left([f^\star(b_i) - \widehat{h}_i(b_i)]^2 - [f^\star(b_i) - \widehat{h}_{i+1}(b_i)]^2\right).$$

The displayed expression is continuous, so F_N is continuously differentiable.

First evaluate this expression with the true lines in place of the fitted lines. If $\delta_i > 0$, the left residual is $\alpha_i\delta_i$ and the right residual is zero. If $\delta_i < 0$, the left residual is zero and the right residual is $-\alpha_i\delta_i$. At $\delta_i = 0$, both are zero. Thus in all cases the true-line expression is

$$\alpha_i^2 p(b_i)\delta_i|\delta_i| = c_i\delta_i|\delta_i| + O(|\delta_i|^3).$$

Next account for the fitted-coefficient changes. Step 4 and boundedness of z give

$$|\widehat{h}_j(x) - h_j^\star(x)| = |z(x)^\top u_j| \le C''\|\boldsymbol{\delta}\|_2^2$$

for every expert and every input in the support. For either boundary residual, replacing a true line by its fitted line changes a square according to $(a-v)^2 - a^2 = -2av + v^2$. At b_i, the true residual satisfies $|a| \le |\alpha_i\delta_i|$, and the line change satisfies $|v| \le C''\|\boldsymbol{\delta}\|_2^2$. Consequently

$$\left|\frac{\partial F_N}{\partial \delta_i} - c_i\delta_i|\delta_i|\right| \le C_{\mathrm{grad}}\left(|\delta_i|^3 + |\delta_i|\|\boldsymbol{\delta}\|_2^2 + \|\boldsymbol{\delta}\|_2^4\right).$$

Each term is $O(\|\boldsymbol{\delta}\|_2^3)$ near zero. Taking the Euclidean norm over the fixed number of boundaries proves equation 18. This calculation does not differentiate the loss expansion's remainder. □

B.2 Proof of Proposition 1

Proof. Let $d = N - 1$ be the number of trainable boundaries. We use the Euclidean norm throughout. All constants below concern the fixed objective; the neighborhood will be fixed before imposing learning-rate bounds.

Step 1. Translate the expansions into uniform inequalities. For a vector $\boldsymbol{\delta} \in \mathbb{R}^d$,

$$d^{-1/2}\|\boldsymbol{\delta}\|_2^3 \le \sum_{i=1}^{d} |\delta_i|^3 \le \|\boldsymbol{\delta}\|_2^3.$$

For the lower bound, Hölder's inequality gives

$$\sum_{i=1}^{d} |\delta_i|^2 \le \left(\sum_{i=1}^{d} |\delta_i|^3\right)^{2/3} d^{1/3}.$$

Raising both sides to the power $3/2$ and rearranging gives the stated lower bound. For the upper bound, $|\delta_i|^3 \le \|\boldsymbol{\delta}\|_2 |\delta_i|^2$, and summing gives $\sum_i |\delta_i|^3 \le \|\boldsymbol{\delta}\|_2^3$. Theorem 5 therefore supplies positive constants a, b, G, C_F, C_E and a fixed neighborhood such that

$$\begin{aligned} a\|\boldsymbol{\delta}\|_2^3 \le \boldsymbol{\delta}^\top \nabla F_N(\boldsymbol{\delta}) \le b\|\boldsymbol{\delta}\|_2^3, \\ \|\nabla F_N(\boldsymbol{\delta})\|_2 \le G\|\boldsymbol{\delta}\|_2^2, \qquad F_N(\boldsymbol{\delta}) = \Theta(\|\boldsymbol{\delta}\|_2^3), \\ 0 \le F_N(\boldsymbol{\delta}) \le C_F\|\boldsymbol{\delta}\|_2^3, \\ \left\|\nabla F_N(\boldsymbol{\delta}) - (c_i\delta_i|\delta_i|)_{i=1}^d\right\|_2 \le C_E\|\boldsymbol{\delta}\|_2^3. \end{aligned}$$

For the first line, the inner product of the leading gradient with $\boldsymbol{\delta}$ is $\sum_i c_i|\delta_i|^3$, while the remainder contributes at most $C_E\|\boldsymbol{\delta}\|_2^4$. On a sufficiently small fixed neighborhood this remainder is at most half the positive leading lower bound. The loss comparison follows by the same estimate applied to its fourth-order remainder. All constants can be chosen positive even if a remainder vanishes identically.

Step 2. Prove the GD rate using changes in the norm. Fix $R > 0$ within the neighborhood above and within the boundary-separation region. Require

$$0 < \eta \le \eta_{\mathrm{GD}} := min\left\{\frac{1}{2GR}, \frac{a}{G^2R}\right\}.$$

At a current nonzero vector $\boldsymbol{\delta}$ of norm at most R, write $\boldsymbol{\delta}^+ = \boldsymbol{\delta} - \eta\nabla F_N(\boldsymbol{\delta})$ for one update. The triangle inequality gives

$$\|\boldsymbol{\delta}^+\|_2 \ge \|\boldsymbol{\delta}\|_2 - \eta G\|\boldsymbol{\delta}\|_2^2 \ge \tfrac{1}{2}\|\boldsymbol{\delta}\|_2.$$

Expanding the squared norm gives the exact identity

$$\|\boldsymbol{\delta}\|_2^2 - \|\boldsymbol{\delta}^+\|_2^2 = 2\eta\boldsymbol{\delta}^\top\nabla F_N(\boldsymbol{\delta}) - \eta^2\|\nabla F_N(\boldsymbol{\delta})\|_2^2.$$

The lower bound from Step 1 and the learning-rate condition imply that this difference is at least $\eta a\|\boldsymbol{\delta}\|_2^3 > 0$. Its upper bound is $2\eta b\|\boldsymbol{\delta}\|_2^3$. Divide by $\|\boldsymbol{\delta}\|_2 + \|\boldsymbol{\delta}^+\|_2$, which lies between $\|\boldsymbol{\delta}\|_2$ and $2\|\boldsymbol{\delta}\|_2$, to obtain

$$\frac{\eta a}{2}\|\boldsymbol{\delta}\|_2^2 \le \|\boldsymbol{\delta}\|_2 - \|\boldsymbol{\delta}^+\|_2 \le 2\eta b\|\boldsymbol{\delta}\|_2^2.$$

In particular, the norm decreases, so induction keeps every iterate within R. Dividing the last display by the product of consecutive norms gives

$$\frac{\eta a}{2} \le \frac{1}{\|\boldsymbol{\delta}_{t+1}\|_2} - \frac{1}{\|\boldsymbol{\delta}_t\|_2} \le 4\eta b.$$

Summing and taking reciprocals proves $\|\boldsymbol{\delta}_t\|_2 = \Theta(t^{-1})$. The loss comparison in Step 1 gives $F_N(\boldsymbol{\delta}_t) = \Theta(t^{-3})$.

Step 3. Bound the effect of coupling on the square-root update. For real numbers u, v, the signed square root satisfies

$$|\operatorname{sgn}(u)\sqrt{|u|} - \operatorname{sgn}(v)\sqrt{|v|}|^2 \le 2|u - v|.$$

If u, v have the same sign, use $(\sqrt{|u|}-\sqrt{|v|})^2 \le ||u|-|v||$. If their signs differ, use $(\sqrt{|u|}+\sqrt{|v|})^2 \le 2(|u| + |v|) = 2|u - v|$. These two cases prove the displayed inequality, including zero arguments.

Let $S(\boldsymbol{g})$ apply this signed square root to each coordinate of $\boldsymbol{g}$. For the leading gradient, it gives

$$S((c_i\delta_i|\delta_i|)_i) = (\sqrt{c_i}\delta_i)_i.$$

Applying the scalar inequality to each coordinate and summing, then using $\sum_i |v_i| \le \sqrt{d}\|v\|_2$, yields

$$\begin{aligned}\|S(\nabla F_N(\boldsymbol{\delta})) - (\sqrt{c_i}\delta_i)_i\|_2^2 &\le 2\sum_i \left|\frac{\partial F_N}{\partial \delta_i} - c_i\delta_i|\delta_i|\right| \\ &\le 2\sqrt{d}\, C_E\|\boldsymbol{\delta}\|_2^3.\end{aligned}$$

Thus, for a fixed constant $C_S > 0$,

$$\|S(\nabla F_N(\boldsymbol{\delta})) - (\sqrt{c_i}\delta_i)_i\|_2 \le C_S\|\boldsymbol{\delta}\|_2^{3/2}.$$

This bound remains valid when individual δ_i vanish, so it imposes no comparability condition on coordinate errors.

Step 4. Prove geometric contraction of the full boundary vector. Fix the common neighborhood R small enough that $C_S\sqrt{R} \le \sqrt{c_{\min}}/2$; this choice depends on the objective, not the learning rate. The GD argument holds on this smaller neighborhood as well. For $0 < \eta \le 1/\sqrt{c_{\max}}$, every number $1 - \eta\sqrt{c_i}$ lies in $[0, 1 - \eta\sqrt{c_{\min}}]$. The componentwise square-root update therefore satisfies

$$\begin{aligned}\|\boldsymbol{\delta}^+\|_2 &\le \|((1 - \eta\sqrt{c_i})\delta_i)_i\|_2 + \eta C_S\|\boldsymbol{\delta}\|_2^{3/2} \\ &\le \left(1 - \eta\sqrt{c_{\min}} + \eta C_S\sqrt{R}\right)\|\boldsymbol{\delta}\|_2 \\ &\le \left(1 - \frac{\eta\sqrt{c_{\min}}}{2}\right)\|\boldsymbol{\delta}\|_2.\end{aligned}$$

The factor is strictly between zero and one. Iterating proves the norm bound and keeps all iterates within R. Finally,

$$F_N(\boldsymbol{\delta}_t) \le C_F\|\boldsymbol{\delta}_0\|_2^3\left(1 - \frac{\eta\sqrt{c_{\min}}}{2}\right)^{3t}.$$

At the zero vector, the exact gradient and the signed square-root update are both zero. This completes the proof. □

B.3 High-dimensional inputs and a translated hyperplane

We justify Remark 1 for a concrete high-dimensional setting. Let $d \ge 2$. Fix a unit vector $n \in \mathbb{R}^d$ and use orthonormal coordinates $X \leftrightarrow (S, Z)$, where $S = n^\top X \in \mathbb{R}$ and $Z \in \mathbb{R}^{d-1}$ contains the tangential coordinates. This change of coordinates preserves volume. The router sends $S < b$ to the left expert and $S \ge b$ to the right expert; only the threshold b is trained.

Assumption 4 (Density and experts near a planar interface)**.** *The support in these coordinates is $[s_-, s_+] \times \mathcal{Z}$, where $\mathcal{Z} \subset \mathbb{R}^{d-1}$ is compact and has nonempty interior. The joint density $\varrho(s, z)$ is continuous, has a continuous partial derivative with respect to s, and satisfies $0 < \varrho_{\min} \le \varrho(s, z) \le \varrho_{\max} < \infty$ on the support. No smoothness across the support boundary is required. The true threshold $b^\star$ is interior to $[s_-, s_+]$. The target equals $h_-^\star$ for $s < b^\star$ and $h_+^\star$ for $s \ge b^\star$, where the two affine functions agree at $(b^\star, z)$ for every $z \in \mathcal{Z}$. Their normal slope difference α is nonzero. For every threshold, both experts are independently fitted by population least squares over all affine functions of (s, z).*

Proposition 2 (One threshold with high-dimensional affine experts)**.** *Under Assumption 4, the exact expert/router decomposition holds. Define*

$$p_S(s) = \int_{\mathcal{Z}} \varrho(s, z)\, \mathrm{d}z, \qquad c_{\mathrm{H}} = \alpha^2 p_S(b^\star) > 0, \qquad \delta = b - b^\star.$$

For the refitted population loss $F(\delta)$,

$$R_{\mathrm{true}}(b) = \frac{c_{\mathrm{H}}}{3}|\delta|^3 + O(|\delta|^4), \qquad Q(b) = O(|\delta|^4),$$
$$F(\delta) = R_{\mathrm{true}}(b) - Q(b) = \frac{c_{\mathrm{H}}}{3}|\delta|^3 + O(|\delta|^4),$$
$$F'(\delta) = c_{\mathrm{H}}\delta|\delta| + O(|\delta|^3), \qquad F''(\delta) = 2c_{\mathrm{H}}|\delta| + O(\delta^2).$$

The loss is locally C^2, strictly convex, and not strongly convex. Theorems 2–4 apply to its scalar threshold updates with $c = c_{\mathrm{H}}$ and their respective learning-rate, initialization, and moment conditions.

Proof. We first identify the difference between the true expert functions, then bound refitting and differentiate the population loss with respect to the threshold.

Step 1. Express continuity across the whole interface. Write the true affine experts as

$$h^\star_\pm(s, z) = d^\star_\pm + a^\star_\pm s + (v^\star_\pm)^\top z.$$

Their difference vanishes at $(b^\star, z)$ for every z in a set with nonempty interior. An affine function of z that vanishes on an open set has zero coefficients. Thus $v^\star_+ = v^\star_-$ and $d^\star_+ - d^\star_- = -(a^\star_+ - a^\star_-)b^\star$. Consequently

$$h^\star_+(s, z) - h^\star_-(s, z) = \alpha(s - b^\star), \qquad \alpha = a^\star_+ - a^\star_-.$$

Agreement at just one point of the interface would not imply this identity; agreement along the interface is the relevant continuity assumption.

Step 2. Reduce the true-expert loss to the normal coordinate. A displaced threshold changes assignments only in the slab between $s = b$ and $s = b^\star$. On that slab, Step 1 gives squared error $\alpha^2(s - b^\star)^2$, independent of z. Integrating the joint density over z yields

$$R_{\mathrm{true}}(b) = \alpha^2 \int_{\min(b,b^\star)}^{\max(b,b^\star)} (s - b^\star)^2 p_S(s)\, \mathrm{d}s.$$

Compactness and continuity make $\partial_s \varrho$ bounded. Differentiation under the integral gives $p'_S(s) = \int_{\mathcal{Z}} \partial_s \varrho(s, z)\, \mathrm{d}z$, which is continuous and bounded. Also $p_S(b^\star) > 0$ because the density is positive and $\mathcal{Z}$ has positive volume. The mean-value bound and integration in Step 2 of Appendix A.1 now give the asserted cubic expansion of R_{true}.

Step 3. Bound the changes in the full affine experts. Use the $(d + 1)$-dimensional feature vector $\phi(s, z) = (1, s, z^\top)^\top$, so each candidate expert is $h_j = \phi^\top \theta_j$ with coefficient vector $\theta_j \in \mathbb{R}^{d+1}$. The predicted regions are

$$\Omega_-(b) = [s_-, b) \times \mathcal{Z}, \qquad \Omega_+(b) = [b, s_+] \times \mathcal{Z}.$$

The fitting matrix for region $\Omega_j(b)$ is

$$G_j(b) = \int_{\Omega_j(b)} \phi\phi^\top \varrho\, \mathrm{d}s\, \mathrm{d}z.$$

For all b sufficiently close to $b^\star$, each region contains a fixed open set. A nonzero affine function cannot vanish on that set. Positivity of the density therefore makes G_j positive definite. Minimizing its quadratic form over unit coefficient vectors on a fixed subregion gives a common positive lower bound, as in Step 1 of Appendix B.1. Hence the matrix inverses stay bounded near $b^\star$ and all fits are unique.

The fitted residual is orthogonal to every coordinate of ϕ. Expanding squared error and using this orthogonality proves $L = L_E + L_R$ and $R_{\text{true}} = L_R + Q$, now with integrals over the high-dimensional regions.

For $\delta > 0$, the right expert remains exactly $h_+^\star$. Write the left expert as $\widehat{h}_- = h_-^\star + \phi^\top u(\delta)$, where $u(\delta)$ is its vector of coefficient changes. Its fitting equations give

$$G_-(b)u(\delta) = \alpha \int_{b^\star}^{b^\star+\delta} \int_{\mathcal{Z}} \phi(s,z)(s-b^\star)\varrho(s,z)\,\mathrm{d}z\,\mathrm{d}s.$$

The feature vector and density are bounded, and $\mathcal{Z}$ has finite volume. The right-hand side has norm $O(\delta^2)$, because the remaining integral of $s-b^\star$ is $\delta^2/2$. The bounded inverse gives $\|u(\delta)\|_2 = O(\delta^2)$. Consequently

$$Q(b) = u(\delta)^\top G_-(b)u(\delta) = O(\delta^4), \qquad F(\delta) = \frac{c_{\mathrm{H}}}{3}\delta^3 + O(\delta^4).$$

For $\delta < 0$, the left expert is exact, and the fitting equation for the right expert has right-hand side $-\alpha \int_b^{b^\star} \int_{\mathcal{Z}} \phi(s,z)(s-b^\star)\varrho(s,z)\,\mathrm{d}z\,\mathrm{d}s$. Its norm is $O(|\delta|^2)$, proving the same coefficient and refitting bounds on that side.

Step 4. Compute the gradient and curvature for a rightward displacement. The fitting coefficients are continuously differentiable in b, because their defining integrals have continuous integrands and their fitting matrices remain invertible. When differentiating the fitted loss, the coefficient derivatives vanish by the fitting equations. Only the moving integration boundary contributes. For $\delta > 0$, define the residual on that boundary by

$$r(\delta,z) = \alpha\delta - \phi(b^\star+\delta,z)^\top u(\delta).$$

Then the exact gradient is

$$F'(\delta) = \int_{\mathcal{Z}} r(\delta,z)^2 \varrho(b^\star+\delta,z)\,\mathrm{d}z.$$

The coefficient bound in Step 3 gives $r(\delta,z) = \alpha\delta + O(\delta^2)$ uniformly over z. The density changes by $O(\delta)$ uniformly over z, so

$$F'(\delta) = \alpha^2 p_S(b^\star)\delta^2 + O(\delta^3) = c_{\mathrm{H}}\delta^2 + O(\delta^3).$$

To justify differentiation once more, differentiate the fitted coefficient equation in Step 3. The derivative of $G_-(b)$ is $\int_{\mathcal{Z}} \phi(b,z)\phi(b,z)^\top \varrho(b,z)\,\mathrm{d}z$. Moving its product with u to the right gives

$$G_-(b)u'(\delta) = \int_{\mathcal{Z}} \phi(b,z) r(\delta,z)\varrho(b,z)\,\mathrm{d}z.$$

The right-hand side is $O(\delta)$ in norm, hence $\|u'(\delta)\|_2 = O(\delta)$. Since $\partial_s\phi$ is a fixed bounded vector,

$$\partial_\delta r(\delta,z) = \alpha - (\partial_s\phi(b,z))^\top u(\delta) - \phi(b,z)^\top u'(\delta) = \alpha + O(\delta)$$

uniformly over z. Differentiating the exact gradient integral therefore yields

$$\begin{aligned} F''(\delta) &= \int_{\mathcal{Z}} \left[2r(\delta,z)\partial_\delta r(\delta,z)\varrho(b,z) + r(\delta,z)^2\partial_s\varrho(b,z)\right]\,\mathrm{d}z \\ &= 2\alpha^2 p_S(b^\star)\delta + O(\delta^2) = 2c_{\mathrm{H}}\delta + O(\delta^2). \end{aligned}$$

Step 5. Treat a leftward displacement and transfer the scalar optimizer results. For $\delta < 0$, write $\widehat{h}_+ = h_+^\star + \phi^\top u_+(\delta)$. The boundary residual is

$$r_+(\delta,z) = -\alpha\delta - \phi(b,z)^\top u_+(\delta) = -\alpha\delta + O(\delta^2).$$

The right region has b as its lower integration endpoint, so

$$F'(\delta) = -\int_{\mathcal{Z}} r_+(\delta,z)^2\varrho(b,z)\,\mathrm{d}z = -c_{\mathrm{H}}\delta^2 + O(|\delta|^3).$$

Differentiating the right-expert fitting equation gives $G_+(b)u_+'(\delta) = -\int_{\mathcal{Z}} \phi(b,z) r_+(\delta,z)\varrho(b,z)\,\mathrm{d}z$. Thus $u_+' = O(|\delta|)$ and $\partial_\delta r_+ = -\alpha + O(|\delta|)$. The derivative of the last gradient formula is consequently $-2c_{\mathrm{H}}\delta + O(\delta^2) = 2c_{\mathrm{H}}|\delta| + O(\delta^2)$.

The proofs in Appendices A.2–A.4 derive a scalar threshold and use exactly these loss and gradient expansions. Substituting c_{H} proves convergence rates on corresponding optimizers in high-dimensional case. □

C PIECEWISE REGRESSION EXPERIMENTS

We provide the complete configuration and results on the experiments on piecewise linear regression in § 4.1.

C.1 EXPERIMENT PROTOCOLS.

The theory from Section 3 with Algorithm 1 computes the boundary gradients using exact values of density distribution $p(x)$, which remains unknown in realistic environment. To simulate the stochasticity, we apply a mini-batch optimization, without prior knowledge on $p(x)$. We specify our implementation protocol as follows.

Minibatch Sampling. At each step, we draw $n = 20{,}000$ inputs datapoint independently from input range, following the population density $p(x)$. We do not introduce extra label noise, so the only randomness comes from mini-batch sampling.

Boundary derivative estimation. Since the exact boundary gradient is not applicable, we estimate it using kernel-weighted estimator:

$$\widehat{g}_i = \frac{1}{nh}\sum_{k=1}^{n} K\Big(\frac{x_k - b_i}{h}\Big)\Big[\big(y_k - \widehat{h}_i(x_k)\big)^2 - \big(y_k - \widehat{h}_{i+1}(x_k)\big)^2\Big], \qquad K(v) = \sigma(v)\big(1-\sigma(v)\big). \tag{45}$$

It estimates a kernel average of the residual jump $\Delta_i(x) = (y - \widehat{h}_i(x))^2 - (y - \widehat{h}_{i+1}(x))^2$ over the points near b_i. The expectation of equation 45 $E[\widehat{g}_i] = \int h^{-1}K((x-b_i)/h)\,\Delta_i(x)\,p(x)\,dx$, approaching $F'(b_i)$ as $h \to 0$. We set $h = 10^{-4}$ by default.

Optimizer hyperparameters. All three optimizers adopt the same learning rate $\eta = 0.1$. `Adam` applies with bias correction, $(\beta_1, \beta_2) = (0.9, 0.95)$, zero initial moments and $\varepsilon = 0.1$ added outside the square root. `SqrtGD` takes the signed square-root on the estimated gradient, without momentum or offset.

We report the relative boundary error $|\delta_t|/|\delta_0|$ and the population loss $L_T/L_0$for each training run. To compare with the analytical convergence rates, we fit the measured boundary error and loss curves from the known parameters $(c, \eta, \varepsilon, \delta_0)$: $|\delta_t| = (|\delta_0|^{-1} + \eta c t)^{-1}$ for GD, $|\delta_t| = (|\delta_0|^{-1} + (\eta/\varepsilon) c t)^{-1}$ for Adam, $|\delta_t| = |\delta_0| r^t$ with $r = 1 - \eta\sqrt{c}$ for `SqrtGD`, and the losses $\frac{c}{3}|\delta_t|^3$. Each optimizer is run on 10 random seeds for single-boundary and multi-segment experiments; and 5 seeds on high-dimensional runs. Each run sample an independent input streams from the same input range $\mathcal{X}$.

C.2 SETTINGS

C.2.1 SINGLE-BOUNDARY EXPERIMENTS

The input density is $p(x) = 1 + 0.8x$ on $[-\frac{1}{2}, \frac{1}{2}]$ and the target is $f^\star(x) = 0.3 + 0.5x + \max\{x, 0\}$, a continuous function with one boundary point at $b^\star = 0$ and slope gap $\alpha = 1$. The results are presented in Figure 1.

C.2.2 MULTIPLE SEGMENTS WITH COUPLED BOUNDARIES

Setup. The input density is $p(x) = 1 + 0.8(x - \frac{1}{2})$ on $[0, 1]$. The target has $N = 6$ affine segments with boundaries $b^\star = (0.15, 0.32, 0.50, 0.66, 0.85)$, first segment $0.2 + 0.4x$, and slope gaps $\alpha_j = \pm\sqrt{c_j / p(b_j^\star)}$ with signs $(+, -, +, +, -)$, chosen so that the boundary constants are $c_j = (3, 2, 1, 2.5, 5)$. The corresponding contraction factors are $r_j = 1 - \eta\sqrt{c_j} = (0.827, 0.859, 0.900, 0.842, 0.776)$, where the middle boundary is the slowest updating one, so $r_{\max} = 0.9$. The one-sided bound of Proposition 1 guarantees the upper-bound of the factor $1 - \eta\sqrt{c_{\min}}/2 = 0.95$.

Results. According to Figure 4, the total loss decays at the asymptotic rate $\approx 3\log(1/r_{\max})$ set by the slowest boundary, while $3\log(1/r_{bound})$ derived from Theorem 5 serves as a loose one-sided upper bound. Figure 5 (c) demonstrates that across 24 runs with $N \in \{2, 3, 4, 6, 8, 12\}$, the measured decay rate of the slowest boundary mostly follows $-\log r_{\max}$, which demonstrate the slowest-updating boundary largely dominates the boundary optimization under the multi-segment target setup. The complete boundary error and loss curves across various d are presented in Figure 5.

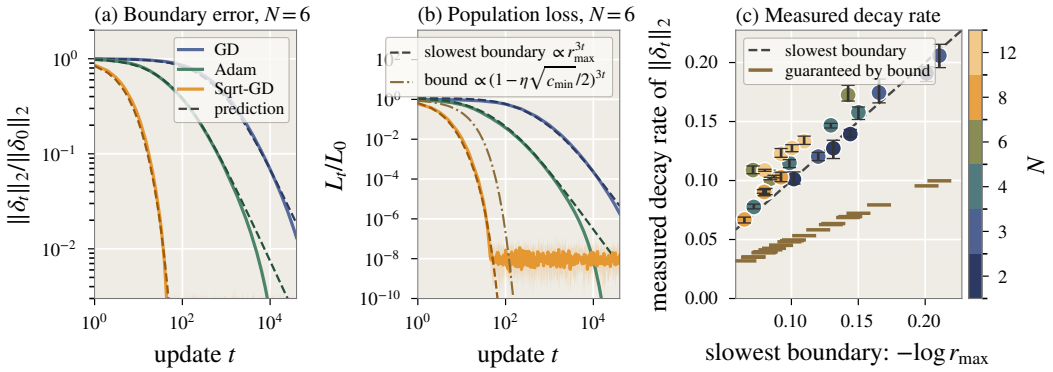

Figure 4: **Six segments with coupled boundaries, minibatch alternation**

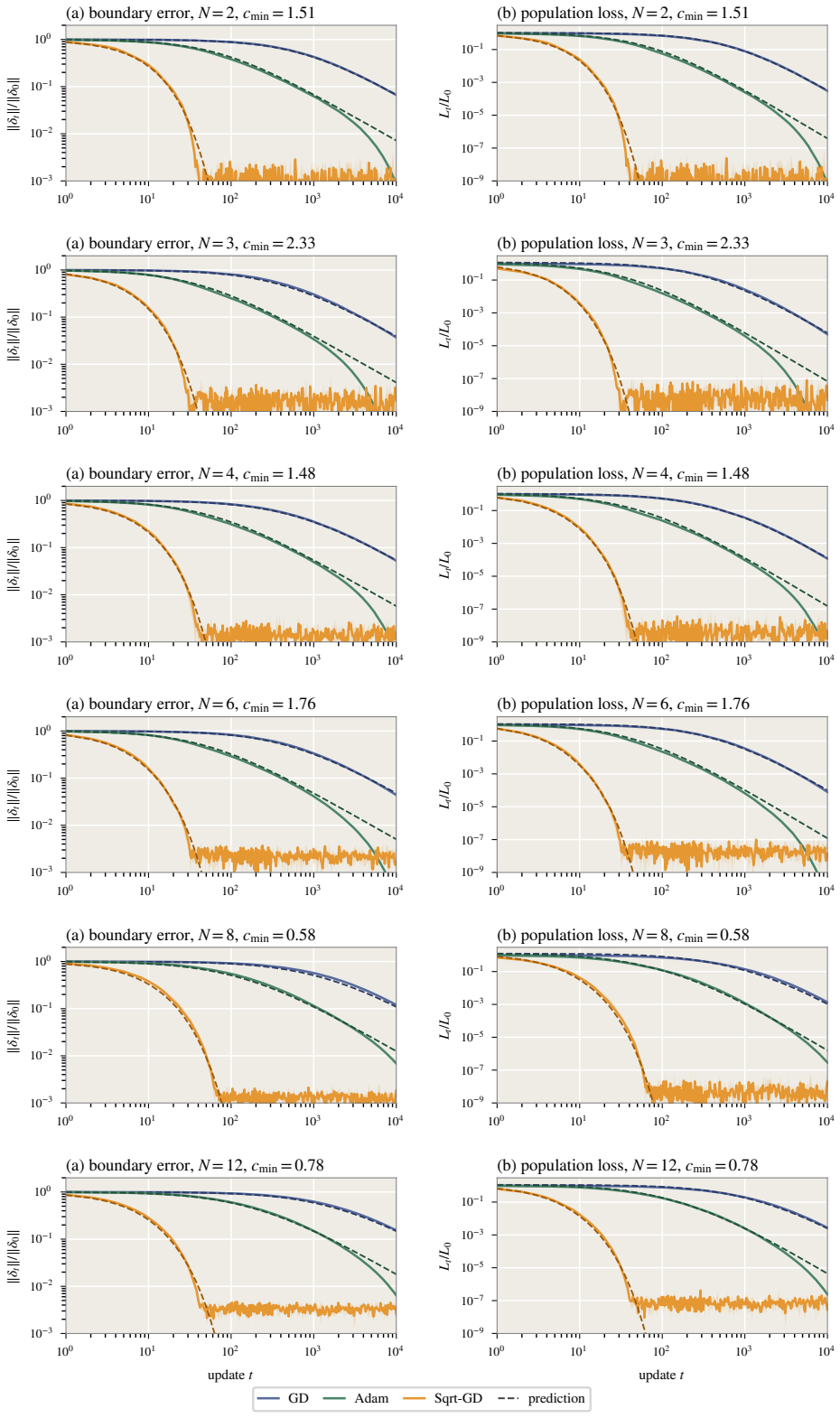

Figure 5: **Boundary Error and Population Loss Curves across various N.** Euclidean boundary error (left) and population loss (right) under GD, Adam and `SqrtGD`; solid, median over three sample streams; shaded, min–max; dashed, the analytic leading-order predictions summed over the boundaries.

C.2.3 PIECEWISE LINEAR REGRESSION WITH HIGH-DIMENSIONAL INPUTS

Setup. The input is $X = LY$, where Y is uniform on the unit ball of $\mathbb{R}^d$ and $L = \Sigma^{1/2}$ with $\Sigma_{ij} = 0.5^{|i-j|}$, so that the coordinates of X are correlated with correlation 0.5 between neighbours. The target is $f^\star(x) = 0.2 + a^\top x + \max\{n^\top x - b^\star, 0\}$ with a fixed vector $a \sim \mathcal{N}(0, \frac{1}{4} I_d/d)$ drawn once, so that the two experts agree on the interface $n^\top x = b^\star$ and their slopes differ by $\alpha = 1$ along n. The router keeps the direction $n = (1, \dots, 1)/\sqrt{d}$ fixed and learns the threshold b of the rule $n^\top x < b$. The true threshold is $b^\star = 0.1\,\sigma$ with $\sigma = \|Ln\|$ the standard scale of the routing coordinate. The constant $c_{\mathrm{H}} = \alpha^2 p_S(b^\star)$ that enters the predictions is computed from the routing marginal derived following.

The routing marginal p_S and the constant c_{H}. The theory of We show the computation steps to get the density of the routing coordinate $S = \boldsymbol{n}^\top X$ and $c_{\mathrm{H}} = \alpha^2 p_S(b^\star)$ which is required in analytical prediction in Remark 1. On the ellipsoid, $S = n^\top LY = (Ln)^\top Y = \sigma\, m^\top Y$ with $\sigma = \|Ln\|$ and $m = Ln/\sigma$ a unit vector, so S is σ times the projection $T = m^\top Y$ of a uniform point of the unit ball onto a fixed direction. The density of T follows by slicing the ball: the set $\{T = \tau\}$ is a $(d-1)$-dimensional ball of radius $\sqrt{1-\tau^2}$, whose volume is proportional to $(1-\tau^2)^{(d-1)/2}$, and normalising over $[-1, 1]$ gives

$$q_d(\tau) = C_d\,(1-\tau^2)^{(d-1)/2}, \qquad C_d = \frac{\Gamma(d/2+1)}{\sqrt{\pi}\,\Gamma\big((d+1)/2\big)}, \qquad p_S(s) = \frac{1}{\sigma}\, q_d\Big(\frac{s}{\sigma}\Big), \tag{46}$$

with the checks $q_1 \equiv \frac{1}{2}$, $q_3(\tau) = \frac{3}{4}(1-\tau^2)$ and $\operatorname{Var} T = 1/(d+2)$. With $\alpha = 1$ and $\tau^\star = b^\star/\sigma = 0.1$,

$$c_{\mathrm{H}}(d) = \frac{C_d\,(1-\tau^{\star 2})^{(d-1)/2}}{\sigma}, \qquad \sigma^2 = n^\top \Sigma n = \frac{1}{d}\sum_{i,j} 0.5^{|i-j|} \;\; (1.5 \text{ at } d=2,\ \to 3 \text{ as } d \to \infty).$$

Results. Figure 6 (a) plots the empirically measured rates at dimension $d = 16$ with $c_{\mathrm{H}} = 0.91$, where the measured convergence rates matches the theoretical prediction in Remark 1 across all three optimizers. We further conducted a sweep across $d \in \{2, 8, 32, 128, 256\}$ on `SqrtGD` and report the fitted loss decay rate in Figure 6 (b), where the measured exponent reproduces the predicted exponential rate along various c_{H}. The complete boundary error and loss curves across various d are presented in Figure 7. Table 2 lists the fitted rates (mean $\pm$ std) with their analytical values from theory.

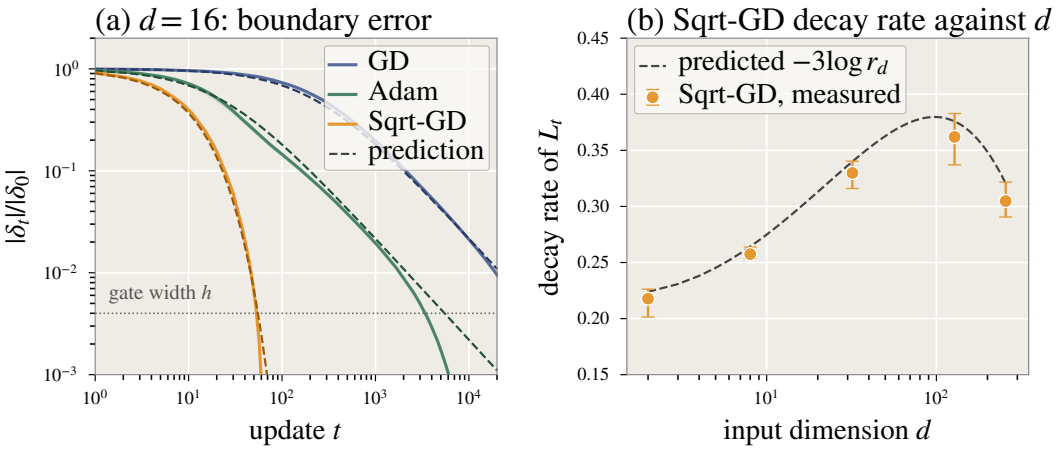

Figure 6: **High-dimensional inputs with a fixed routing direction, minibatch alternation.**

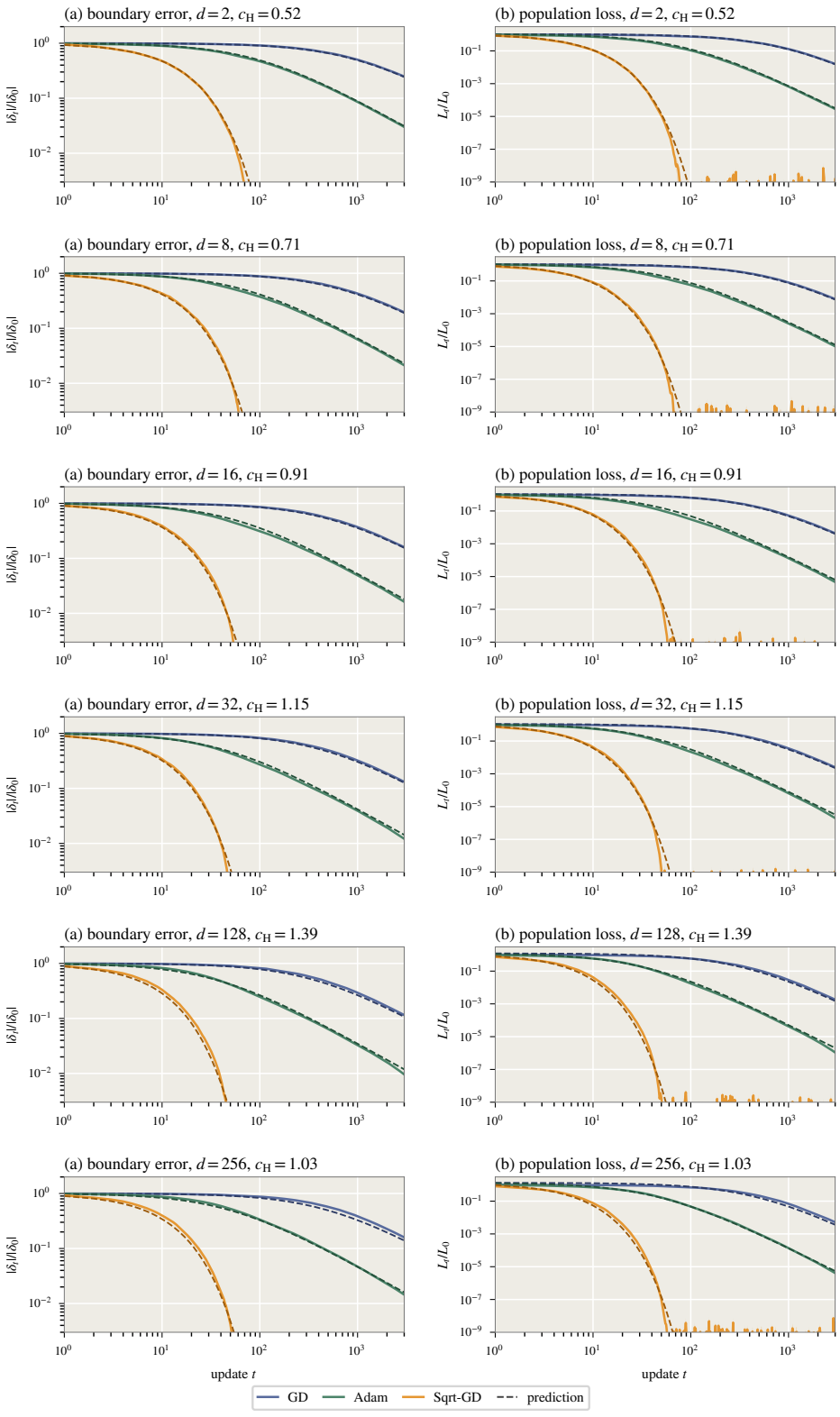

Figure 7: **Boundary Error and Population Loss Curves across various d.** Boundary error (left) and population loss (right) under GD, Adam and `SqrtGD`; solid, median over two sample streams; shaded, min–max; dashed, the scalar predictions evaluated at $c = c_H(d)$.

Table 2: **Empirically fitted convergence rates of the piecewise linear regression experiments.** Loss exponents for GD and Adam (prediction 3) and `SqrtGD` loss decay rates (prediction $-3 \log r$), mean $\pm$ standard deviation over replicates. "*Fitted steps*" denotes the range of update steps t over which the straight-line fit is taken.

Experiment	Method	Fitted steps	Measured	Predicted
Single boundary	GD	400–9714	3.11 ± 0.01	3
	Adam	40–974	2.81 ± 0.05	3
	`SqrtGD`	11–37	0.303 ± 0.015	0.316
Multiple boundaries ($N = 6$)	GD	2532–8736	3.27 ± 0.01	3
	Adam	251–860	3.17 ± 0.03	3
	`SqrtGD`	11–19	0.358 ± 0.016	0.316
high-dimensional ($d = 16$)	GD	442–5290	3.08 ± 0.01	3
	Adam	45–529	2.76 ± 0.05	3
	`SqrtGD`	11–32	0.284 ± 0.003	0.300
`SqrtGD` sweep on high-dimensional	$d = 2$	15–43	0.218 ± 0.010	0.224
	$d = 8$	13–36	0.257 ± 0.004	0.264
	$d = 32$	10–28	0.330 ± 0.009	0.340
	$d = 128$	9–25	0.362 ± 0.019	0.376
	$d = 256$	11–30	0.305 ± 0.015	0.321

D EXPERIMENTAL PROTOCOL AND SUPPLEMENTARY CONTROLS

Architecture and training. We train decoder-only transformers with 8 layers with $1.3B$ total parameters, 12 attention heads and embedding dimension 768. All the models are trained on a 10BT subset from FineWeb corpus with sequence length 1024 and vocabulary size 50,304. Active counts are approximately 96M, 134M, 209M, and 360M with various routing sparsity $k = 1, 2, 4, 8$.

Each training run consumes 3.3B tokens, with float32 parameters and bfloat16 autocasting. We adopt 2000 steps warmup and cosine-decay learning rate scheduler ends at 0.1 times the peak LR. `AdamW` uses $(\beta_1, \beta_2) = (0.9, 0.95)$ and $\varepsilon = 10^{-8}$. Global gradient clipping at 1.0 precedes optimizer updates. For `Muon`, all the $2D$ parameter matrices are optimized by `Muon`, while embeddings and output layer and remaining groups like layer norms are optimized by `AdamW`. Router parameters in `Muon` either applies zero-decay `AdamW` or ***SR***. Table 4 provides the complete model and training configurations.

Table 3: Parameter budgets for every (experts, top-k) configuration, in {total}-A{active} notation. Active parameters scale with k; total scales with E.

		top-k			
Experts	total	k=1	k=2	k=4	k=8
E=8	360M	360M-A96M	360M-A134M	—	—
E=16	662M	662M-A96M	662M-A134M	662M-A209M	662M-A360M
E=32	1.3B	1.3B-A96M	1.3B-A134M	1.3B-A209M	1.3B-A360M

Table 4: Complete architecture and training configuration for LLM experiments.

Backbone		**Training**	
Architecture	GPT-2 style decoder	Steps	25,000
Layers	8	Tokens/step	131,072
d_{model}	768	Total tokens	$\approx$ 3.3B
Heads / head dim	12 / 64	Sequence length	1024
Norm	pre-LayerNorm, no bias	Micro-batch / GPU	32
Activation	GELU	Peak LR	10^{-3} (Muon matrices 0.02)
Positional	learned absolute	Schedule	cosine, 2,000-step warmup
Vocabulary	50,304 (GPT-2 BPE)	Min LR	$0.1\times$ peak
Head	weight-tied to embedding	Weight decay	0.1 (2-D weights only)
Init	$\mathcal{N}(0, 0.02)$	Grad clip	1.0 (global norm)
Residual init	scaled $1/\sqrt{2n_{\text{layer}}}$	Precision	bf16 autocast, fp32 master
MoE layer (replaces every FFN)		**Data & systems**	
Experts E	32	Corpus	FineWeb (Penedo et al., 2024)
Top-k	1, 2, 4, 8	Tokenizer	GPT-2 (tiktoken)
Expert MLP	$768 \to 3072 \to 768$	Train / val tokens	$\approx$ 10B / 10M
Expansion	$4\times$	Hardware	$4\times$ GH200 (DDP)
Shared experts	0	Throughput	78–147K tok/s
Aux losses	load-balance 0.01, z-loss 0.001	Router share	$< 0.1\%$

D.1 COMPLETE RESULTS ON LEARNING RATE ABLATION

We present the complete statistics of learning rate ablation with top-2 routing ($E = 32, k = 2$) in table 5.

Table 5: Learning-rate sweeps at $E{=}32, k{=}2$ Diverged runs are marked with ✗. The optimum learning rates in each configuration are marked in **Bold**.

$\eta_{\texttt{AdamW}}$	AdamW	$+$***SR***	$\eta_{\texttt{Muon}}$	Muon	$+$***SR***
5×10^{-5}	3.660	3.633	0.001	3.339	3.271
1×10^{-4}	3.453	3.436	0.005	**3.270**	**3.241**
3×10^{-4}	**3.320**	3.262	0.01	3.310	3.278
6×10^{-4}	3.758	3.227	0.02	3.393	3.367
1×10^{-3}	6.550 ✗	**3.203**	0.04	3.475	3.468
3×10^{-3}	4.519 ✗	3.246			