PREDICTING CHARACTERISTIC LEARNING RATE BASED ON SECOND-MOMENT-SHAPED STOCHASTIC MOBILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural networks are trained using local gradients with finite step size, making the learning rate α central to balancing exploration in the loss landscape and stability in training dynamics. To effectively navigate a highly heterogeneous loss landscape, adaptive training is widely used in which training steps are further adjusted adaptively using the second moment. However, the selection of learning rate remains empirical and often costly. Here we derive a statistical physics prediction for this parameter. We find that the competition between stochastic exploration and curvature-induced confinement in training can be quantitatively defined, so that there exists an optimal characteristic α_c. We estimate α_c from a few mini-batches at a cost of 31 gradient-equivalent steps, rather than selecting it by comparing final performance across multiple training runs. Across deep neural networks and large-scale language models, the predicted α_c lies near the generalization optimum and closely matches the best learning rate found by grid search. Moreover, we provide a measurable criterion for identifying the predictor's applicability regime and show that it attains the highest development accuracy among seven inexpensive selection baselines. Therefore, learning rate selection can be transformed from a costly empirical tuning procedure into an intrinsic dynamical prediction.

1 INTRODUCTION

In a neural network, the space of model parameters defines a loss landscape. Training the neural network is like searching for low-loss regions without a global map. Specifically, it is a dynamical search in a high-dimensional parameter space with a landscape characterized by broad valleys, narrow ravines, nearly flat plateaus, saddle points, and curvatures that can vary by many orders of magnitude (Yang et al., 2023; Feng & Tu, 2021; Feng et al., 2023; Zhang & Tang, 2025; Ly & Gong, 2025; Shen et al., 2025; Liu et al., 2025; Baldassi et al., 2020; Verpoort et al., 2020; Geiger et al., 2019; HaoChen et al., 2021; Ding et al., 2026). An optimizer guides the model parameters to move through this landscape using only local gradient information, without knowing where the best solution lies. Accordingly, a good optimizer is expected to balance three competing objectives: to descend toward low loss, to avoid becoming confined by unfavorable local narrow structures, and to remain stable while moving through a highly heterogeneous terrain.

To reduce the loss, the most direct approach is deterministic gradient descent (GD), which updates the parameters along the negative local gradient $\theta_{t+1} = \theta_t - \alpha \nabla F(\theta_t)$, where $F(\theta)$ denotes the loss function, θ represents the model parameters, and α is the learning rate. GD behaves like a traveler who always steps in the steepest downhill direction. However, on a nonconvex landscape, seeing only the slope under one's feet is not enough to identify the best region of the landscape. A sequence of locally optimal steps does not necessarily lead to a globally optimal solution. The trajectory may settle into unfavorable local regions, become trapped in narrow valleys, or make little progress along stiff and poorly conditioned directions. To avoid such local traps and increase exploration capability, stochastic gradient descent (SGD) changes the optimization trajectory by introducing mini-batch-induced noise: $g_t = \nabla F(\theta_t) + \xi_t$, $\Delta\theta_t^{\mathrm{SGD}} = -\alpha g_t$, where ξ_t denotes the stochastic fluctuation arising from mini-batch sampling.

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.

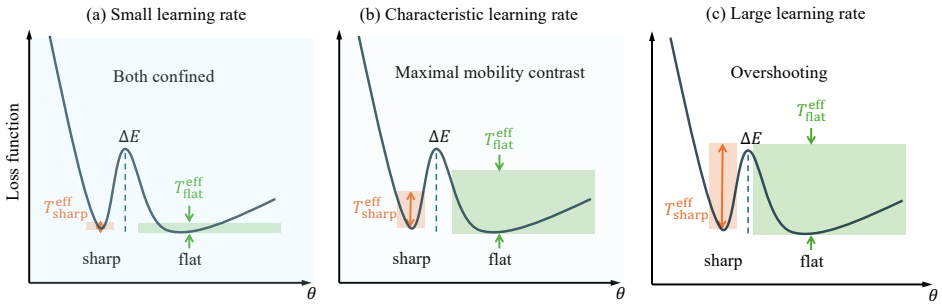


Figure 1: **Learning-rate-dependent mobility contrast on a loss landscape.** The illustrative loss landscape is shown across panels. Adam modifies the effective stochastic dynamics through second moment normalization. (**a**) For small learning rates, both high- and low-curvature regions have low effective mobility, and the dynamics remains confined. (**b**) At the characteristic learning rate α_c, adaptive optimizer maximizes the relative mobility contrast $R(\alpha) = M_{\text{flat}}/M_{\text{sharp}}$ between low- and high-curvature regions, corresponding to the maximal mobility contrast regime. The mobility expression $M \propto \exp(-\Delta E/T_{\text{eff}})$ is used as a local mobility proxy. (**c**) For large learning rates, both regions become highly mobile, the sharp-flat contrast is lost, and global overshooting occurs.

In the landscape picture, SGD can be viewed as a downhill traveler continuously perturbed by random kicks. These kicks may occasionally drive the trajectory away from purely downhill paths, providing opportunities to escape shallow local traps and narrow valleys that deterministic descent would tend to follow. For both GD and SGD, the global learning rate α controls the magnitude of deterministic descent and stochastic fluctuations, balancing stable updates and exploration. A large learning rate tends to jump out of local traps and to enhance exploration, but it also increases the risk of divergence. Conversely, a small learning rate improves local stability but suppresses stochastic exploration, causing SGD to behave increasingly like deterministic GD. In many practical settings, neural network training has been observed to approach the edge of stability, where generalization coexists with marginal instability (Cohen et al., 2021; Feng et al., 2019; Arora et al., 2022).

However, modern large-scale deep models make this picture considerably more complicated. Their highly heterogeneous loss landscapes exhibit diverse curvature patterns, leading to curvature-sensitive loss spikes and optimization instabilities (Bai et al., 2025). To address this issue, adaptive optimizers, particularly Adam, are widely used to adjust the search dynamics adaptively based on historical gradient statistics (Kingma & Ba, 2015; Loshchilov & Hutter, 2019; Devlin et al., 2019; Brown et al., 2020; Zhang et al., 2020b). Specifically, Adam updates the parameters according to

$$\Delta\theta_t^{\text{Adam}} = -\alpha \frac{m_t}{\sqrt{v_t} + \epsilon}, \tag{1}$$

where m_t and v_t are running estimates of the first and second gradient moments. Compared with GD and SGD, the factor $1/\sqrt{v_t}$ acts like a car gear which shifts automatically: in terrains with persistently large gradients, the optimizer is shifted into lower gears, whereas in terrains with smaller gradients, it shifts into higher gears. This adaptive dynamics better balances fast search with avoiding instability. However, the global learning rate α in Eq. 1 is still a free hyperparameter. This hyperparameter is like the throttle of a car which controls the overall magnitude of parameter updates and is also critical to the learning dynamics. Selecting it therefore still requires costly empirical sweeps in many applications (Shallue et al., 2019; Goyal et al., 2017; Kalra & Barkeshli, 2024).

In this work, we develop a reduced statistical physics framework to quantify this adaptive exploration mechanism. Adam is described as a stochastic process in which mini-batch fluctuations induce parameter-space diffusion after second moment normalization. We characterize this diffusion by a noise-normalized diffusion scale, denoted the effective temperature T_{eff}, which measures how efficiently stochastic gradient noise is converted into parameter mobility (Fig. 1). Unlike thermodynamic temperature, this effective temperature describes the dynamical state of optimization under adaptive normalization. Because the normalization factor $1/\sqrt{v_t}$ depends on local gradient statistics, different regions of the loss landscape experience different T_{eff} and mobility levels. The global learning rate α does not create this heterogeneity but controls its overall magnitude. When α is too small, all regions become effectively frozen and exploration is suppressed. When α is too large,

mobility increases across the entire landscape and the distinction between regions disappears. Therefore, a characteristic learning rate exists where the contrast between highly mobile and dynamically confined regions is maximized (Fig. 1b).

We validate the framework in a controlled two-basin landscape, models trained from scratch (MLP, CNN, and ResNet), and pretrained transformers (BERT, GPT, Llama, ModernBERT, and Qwen). In a two-basin toy landscape, the predicted characteristic learning rate α_c identifies the regime where the mobility contrast between low- and high-curvature regions is maximized. Across empirical tasks, the predicted characteristic learning rate α_c falls within a near-optimal generalization regime and is comparable to the best value obtained through costly grid search. We additionally compare it with seven inexpensive learning-rate selectors. Among these inexpensive selectors, our predictor obtains the highest mean development accuracy in both tested model-task blocks. The same principle also extends empirically to AdamW and Adafactor and remains effective with warmup and cosine decay. More broadly, our result suggests that learning rate selection in adaptive optimization is not merely a matter of empirical tuning, but can emerge as an intrinsic dynamical scale of the learning process.

2 Related Work

Learning rate selection. For gradient descent, curvature bounds the stable step size ($\alpha < 2/\lambda_{\max}$) (Boyd & Vandenberghe, 2004; Cohen et al., 2021), and an analogous bound on the preconditioned sharpness holds for Adam (Cohen et al., 2022). These criteria indicate when training becomes unstable, but not which learning rate below that limit is favorable. In practice, Adam's learning rate is still selected by grid search or empirical choice 3×10^{-4} (Shallue et al., 2019; Kingma & Ba, 2015). Cheaper alternatives either transfer a pre-tuned value across batch sizes (Malladi et al., 2022) or adapt the step size online (Defazio & Mishchenko, 2023; Mishchenko & Defazio, 2023; Ivgi et al., 2023). In contrast, we predict, before main training and at a fixed cost, a characteristic learning rate linked to generalization, without using validation data or modifying the optimizer.

Statistical physics perspectives. Statistical physics has been widely used to study SGD, through diffusion approximations (Mandt et al., 2017), Langevin formulations (Chaudhari & Soatto, 2018), and noise-induced preference for flat minima (Feng & Tu, 2021; Smith et al., 2021; Yang et al., 2023). These analyses assume that noise enters the update directly, which does not hold for adaptive methods. We map Adam's second-moment-normalized updates onto a curvature-dependent effective temperature, and use a Kramers-type mobility proxy (Kramers, 1940) to identify the learning rate that maximizes the mobility contrast between low- and high-curvature regions.

3 Effective temperature framework from second moments

The second moment normalization in Adam induces a curvature-dependent reshaping of gradient noise, producing distinct effective temperatures for high-curvature (i.e., sharp) and low-curvature (i.e., flat) regions. To formalize it, we decompose Adam's second moment-normalized mini-batch updates into deterministic drift and stochastic fluctuations (Yang et al., 2023; Feng & Tu, 2021; Lin et al., 2024), where gradient noise induces an effective temperature T_{eff}. We use the exponential form to define a local mobility proxy (Jaynes, 1957; Kramers, 1940; Risken, 1989)

$$M = A \exp(-\Delta E / T_{\text{eff}}), \tag{2}$$

where A is a prefactor constant. Throughout the derivation we work in natural units of the reduced model, measuring energies (ΔE and T^{eff}) in a reference loss scale E_0 and a reference parameter scale θ_0, and set $E_0 = \theta_0 = 1$, analogous to setting $k_B = 1$ in statistical physics. For analytical tractability, we assume a matched-barrier ΔE approximation, in which the basin-specific barrier height and prefactor are treated as slowly varying constants. By comparing the effective temperatures associated with sharp and flat basins, with the latter favoring better generalization due to their wide basins and lower sensitivity to parameter perturbations, our framework leads to a closed-form prediction of the characteristic learning rate α_c.

Specifically, consider a smooth loss landscape $F(\theta)$ with parameters $\theta \in \mathbb{R}^d$, where d is the number of trainable parameters. Around a local reference point $\theta^\star$, the loss admits a second-order expansion. Since our mobility analysis depends on curvature, we retain the quadratic contribution: $F(\theta) \approx F(\theta^\star) + \frac{1}{2}\sum_{i=1}^{d} \lambda_i (\theta_i - \theta_i^\star)^2$, where $\{\lambda_i\}$ represents the Hessian eigenvalues. Here, we use the

dominant Hessian eigenvalue $\lambda_{\max}$ as a proxy for local sharpness (Arora et al., 2022). Along a local coordinate or projected direction i, we model $g_{i,t} = \nabla_i F(\theta_t) + \xi_{i,t}, \mathbb{E}[\xi_{i,t}] = 0, \mathrm{Var}(\xi_{i,t}) = \sigma_i^2/b$, where b is the mini-batch size and σ^2 characterizes the gradient noise strength for the given batch-size b training setting. At iteration t, the Adam update rules are $m_{t+1} = \beta_1 m_t + (1-\beta_1) g_t, v_{t+1} = \beta_2 v_t + (1-\beta_2) g_t^2, \theta_{t+1} = \theta_t - \alpha \frac{m_{t+1}}{\sqrt{v_{t+1}+\epsilon}}$, where β_1, β_2 are exponential decay rate constants (typically close to 1), and $\epsilon > 0$ is a small constant for numerical stability.

To quantify the stochastic behavior, we focus on the stochastic component of the parameter update. For coordinate i, the stochastic increment can be written as $\Delta\theta_i^{\mathrm{stoch}} = -\alpha \frac{\delta m_i}{\sqrt{v_i+\epsilon}}$, where δm_i denotes the fluctuation of the first-moment estimate around its local mean. Over a short training window, v_i is treated as locally quasi-stationary and weakly correlated with the fast component of δm_i (Appendix Fig. 1). Thus, the variance of the update becomes $\mathrm{Var}\big[\Delta\theta_i^{\mathrm{stoch}}\big] \approx \alpha^2 \frac{\mathrm{Var}(\delta m_i)}{\langle v_i \rangle + \epsilon}$. In the regime of non-vanishing gradients, the contribution of ϵ is a lower-order correction. The leading-order momentum variance $\mathrm{Var}(\delta m_i) \approx \sigma^2/b$ inherits from the mini-batch gradient noise. Meanwhile, the expected second moment $\langle v_i \rangle = \mathbb{E}[(\nabla F_i)^2] + \sigma^2/b$ combines both the squared deterministic gradient and the stochastic contribution from mini-batch noise.

To connect the discrete stochastic updates of Adam with a diffusion description, we begin from the discrete variance-diffusion relation. For a stochastic process with increments $\Delta\theta$ measured over a time interval Δt, the diffusion coefficient D is defined through the variance-diffusion relation $\mathrm{Var}[\Delta\theta] = 2D\,\Delta t$. In standard algorithmic time, one optimizer step would correspond to $\Delta t = 1$. However, a mini-batch update is not a single-sample stochastic step: it aggregates gradient information from b samples into one parameter update. Thus, measuring diffusion per unit of processed sample information gives an effective time interval $\Delta t_{\mathrm{eff}} = 1/b$ per mini-batch step. Substituting this effective time scale into the variance-diffusion relation gives

$$D_i^{\mathrm{eff}} = \frac{\mathrm{Var}[\Delta\theta_i^{\mathrm{stoch}}]}{2\Delta t_{\mathrm{eff}}} \approx \frac{b\alpha^2}{2} \frac{\mathrm{Var}[\delta m_i]}{\langle v_i \rangle}. \tag{3}$$

Then, we define the effective temperature as a noise-normalized diffusion efficiency $T_i^{\mathrm{eff}} = D_i^{\mathrm{eff}}/\sqrt{\mathrm{Var}(\xi_i)}$, where $\sqrt{\mathrm{Var}(\xi_i)} = \sigma_i/\sqrt{b}$ is the intrinsic amplitude of the mini-batch gradient fluctuation. Increasing b reduces the raw gradient noise amplitude while simultaneously modifying the sample-normalized diffusion scale associated with each aggregated update. Substituting the expressions for D_{eff} and $\mathrm{Var}(\delta m_i)$ yields $T_i^{\mathrm{eff}} \approx \frac{\alpha^2 \sigma_i \sqrt{b}}{2\langle v_i \rangle}$, where T_i^{eff} quantifies how efficiently mini-batch stochasticity is converted, after Adam's second moment normalization, into parameter-space diffusion per unit sample information.

Moreover, the effective temperature exhibits different behavior depending on the local geometry of the loss landscape. In low-curvature regions (e.g., flat basins), where gradients are small and curvature is minimal $\langle v_i \rangle \approx \sigma^2/b$, leading to: $T_{\mathrm{flat}}^{\mathrm{eff}} = \frac{\alpha^2 b^{3/2}}{2\sigma}$. In high-curvature regions (e.g., sharp basins), where the dominant eigenvalue $\lambda_{\max}$ controls the dynamics, a self-consistent closure of sharp directions yields the cubic relationship $\lambda_{\max}^2 (T^{\mathrm{eff}})^3 = \alpha^2 \sigma b^{1/2}/2$, with the solution: $T_{\mathrm{sharp}}^{\mathrm{eff}} = \left(\frac{\alpha^2 \sigma b^{1/2}}{2\lambda_{\max}^2}\right)^{1/3}$, indicating that stochastic motion is strongly suppressed by curvature, as confirmed in Appendix Fig. 11. The effective temperatures obtained above characterize local stochastic mobility rather than equilibrium thermodynamic temperature.

We adopt a matched effective energy-barrier ΔE approximation, supported by comparable finite-radius loss increases across different-curvature directions (Appendix Fig. 2). We then define the curvature-dependent mobility contrast $R(\alpha) = \frac{M_{\mathrm{flat}}}{M_{\mathrm{sharp}}} = \exp\!\left(-\frac{C_{\mathrm{flat}}}{\alpha^2} + \frac{C_{\mathrm{sharp}}}{\alpha^{2/3}}\right)$ with $C_{\mathrm{flat}} = \frac{2\sigma\Delta E}{b^{3/2}}, C_{\mathrm{sharp}} = \frac{\Delta E (2\lambda_{\max}^2)^{1/3}}{b^{1/6}\sigma^{1/3}}$, where $R(\alpha)$ measures the relative enhancement of local mobility in lower-curvature regions compared with higher-curvature regions. It is therefore a local mobility-contrast measure, not a fixed final basin selection. Its maximum determines a characteristic learning rate that maximizes the mobility contrast: below this learning rate, both regions remain effectively confined, whereas above it, both regions become increasingly mobile and potentially unstable, and the mobility contrast progressively decreases. Optimizing this ratio by setting $\frac{dR}{d\alpha} = 0$ yields:

$$\frac{dR}{d\alpha} = R(\alpha) \left(\frac{2C_{\mathrm{flat}}}{\alpha^3} - \frac{2C_{\mathrm{sharp}}}{3\alpha^{5/3}} \right) = 0, \tag{4}$$

leading to the characteristic learning rate condition $\alpha_c^{4/3} = \frac{3C_{\text{flat}}}{C_{\text{sharp}}}$. We also found that this extremum is a maximum by demonstrating that $\frac{d^2R}{d\alpha^2} = R(\alpha)\,\phi''(\alpha) < 0$, where $R(\alpha) > 0$ and the second derivative of the exponent is $\phi''(\alpha_c) = -\frac{8C_{\text{flat}}}{3\alpha_c^4} < 0$ at α_c. Solving Eq. 4 for α_c explicitly gives the closed-form expression:

$$\alpha_c \approx 3.22\,\frac{\sigma}{b\,\lambda_{\max}^{1/2}}, \tag{5}$$

which provides the closed-form characteristic scale predicted by the reduced mobility theory. Detailed derivations are provided in the Appendix. Restoring the reference scales E_0 and θ_0 gives $\alpha_c \approx 3.22\,\theta_0\sigma/(b\sqrt{\lambda_{\max}E_0})$, which is covariant under both loss rescaling and reparameterization, consistent with Adam's dynamics (Appendix C). Despite the simplified modeling assumptions, our maximal-mobility framework remains predictive across diverse settings.

4 Estimation and computational cost of the predicted learning rate

The estimator requires only a few gradient computations and a short power iteration at a fixed checkpoint. Because σ and $\lambda_{\max}$ can be estimated directly from the model and mini-batch gradients, no optimizer warm-up is required. This checkpoint lies well before progressive sharpening sets in (Appendix Fig. 7), and the dominant curvature has been observed to remain relatively stable during early training (Ghorbani et al., 2019; Sagun et al., 2017). Unless otherwise specified, we use $K = 20$ mini-batches to estimate σ^2 and $T = 10$ power iterations to estimate $\lambda_{\max}$, which we found sufficient for stable estimates. The first stage requires 20 forward-backward passes, while the second requires one backward pass to construct the gradient graph, followed by T Hessian-vector products (HVPs). Counting each forward-backward pass or HVP as one gradient-equivalent (GE) step, a single estimate of $(\sigma^2, \lambda_{\max}, \alpha_c)$ requires about $20 + 1 + 10 = 31$ GE steps. Since the estimator requires a fixed number of gradient-equivalent operations, its relative overhead does not increase with model size. In practice, on NVIDIA A100-SXM4-80GB GPUs, prediction took 3.54 s for ModernBERT-base on QNLI and 25.09 s for Qwen3-1.7B on RTE, corresponding to 0.44% and 8.63% of the respective fine-tuning time. The higher fraction on RTE reflects its short fine-tuning run. For pretraining Llama-1.1B, prediction took 9.5 s on the same GPUs, compared with 2,420.3 s for the five complete fixed-grid runs. Thus, prediction required only 0.39% of the compute used by the grid search and selected the learning rate before training without consulting the validation loss.

Sensitivity of estimating α_c. Table 1 summarizes the sensitivity of characteristic learning-rate estimate to perturbations in $\lambda_{\max}$ and σ^2. Even a 25% error in estimating the top curvature leads to only a modest 10-15$\%$ change in α_c. Moreover, α_c depends on the noise level, so a 25% error in σ^2 results in approximately a 12% change in α_c. Even when the curvature or noise estimates may contain uncertainty, the induced error in α_c is comparably smaller, indicating that the characteristic learning-rate estimate is more stable than either of its raw inputs.

Estimate error	α_c scaling factor	Relative change (%)
$\lambda_{\max}+25\%$	$1.25^{-1/2} \approx 0.894$	-10.56%
$\lambda_{\max}-25\%$	$0.75^{-1/2} \approx 1.154$	$+15.47\%$
$\lambda_{\max}+50\%$	$1.5^{-1/2} \approx 0.816$	-18.35%
$\lambda_{\max}-50\%$	$0.5^{-1/2} \approx 1.414$	$+41.42\%$
$\sigma^2+25\%$	$\sqrt{1.25} \approx 1.118$	$+11.80\%$
$\sigma^2-25\%$	$\sqrt{0.75} \approx 0.866$	-13.40%
$\sigma^2+50\%$	$\sqrt{1.5} \approx 1.225$	$+22.47\%$
$\sigma^2-50\%$	$\sqrt{0.5} \approx 0.707$	-29.29%

Table 1: Scaling and relative changes of α_c when $\lambda_{\max}$ or σ^2 varies by $\pm 25\%$ or $\pm 50\%$.

5 A toy landscape model for curvature-dependent mobility

To validate the mechanism underlying our predicted characteristic learning rate α_c, we construct a one-dimensional toy landscape consisting of two basins: a sharp basin centered at $\theta = -0.5$ and a flatter basin at $\theta = 0.5$. Each basin is approximated as a quadratic potential with different curvatures, and an additive Gaussian noise term simulates a stochastic loss landscape (Chen et al., 2022). The dynamics of a single parameter in this landscape evolves under Adam updates with mini-batch noise and second moment normalization, reflecting the curvature-dependent stochasticity of adaptive optimization. This two-basin toy model is designed to compare how Adam converts stochastic fluc-

tuations into local parameter mobility in high- and low-curvature regions. It is therefore a controlled case of curvature-dependent mobility contrast, rather than a model of final basin selection.

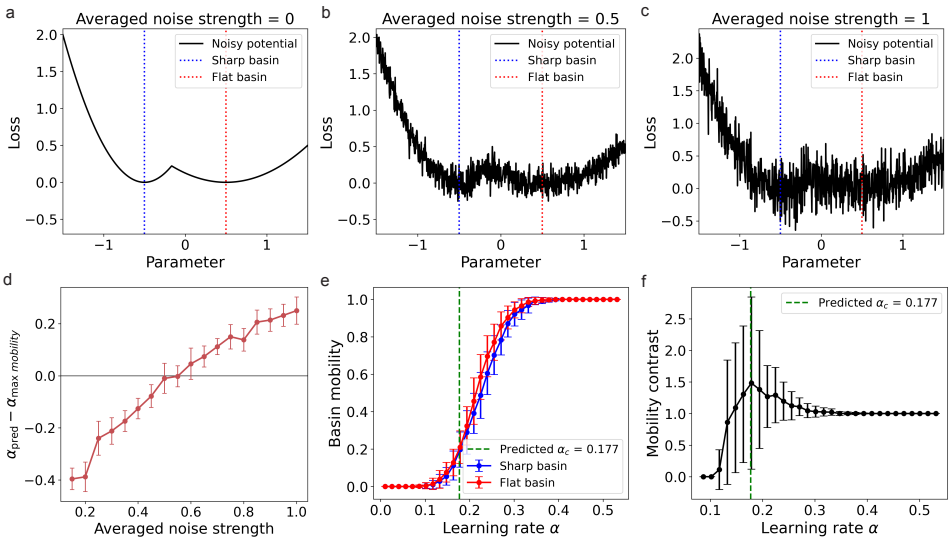


Figure 2: **Two-basin toy model illustrating curvature-dependent mobility under stochastic noise.** (**a-c**) One-dimensional landscapes with a sharp basin at $\theta = -0.5$ (blue) and a flat basin at $\theta = 0.5$ (red), shown under increasing averaged noise strength (variance normalized by batch size). No noise (**a**) yields smooth potential but negligible stochastic driving. Intermediate noise (**b**) allows mobility while preserving basin structure. Strong noise (**c**) distorts the landscape basins. (**d**) Difference between the predicted characteristic learning rate α_{pred} and the empirically observed rate $\alpha_{\max\text{mobility}}$ versus noise strength. Error bars denote standard deviation over 100 independent runs. (**e**) Absolute basin mobility as a function of α at noise strength 0.5. Red and blue points denote flat and sharp basins, with error bars over 100 independent runs. (**f**) Relative mobility contrast $R(\alpha) = M_{\text{flat}}/M_{\text{sharp}}$ versus α, computed over 1000 independent runs. Green dashed line indicates the predicted α_c, corresponding to maximal relative mobility contrast.

Figures 2a-c show the noisy realizations of the potential landscape. The sharp basin has high curvature, whereas the flat basin has low curvature. In Fig. 2f, we evaluate the relative mobility contrast initialized near each basin as a function of the global learning rate α. We found that this contrast rises steeply near the predicted α_c, suggesting a characteristic learning rate at which flat-region mobility becomes pronounced while sharp-region motion remains relatively confined. This does not imply that α_c maximizes absolute mobility from either basin. Rather, it marks the edge at which the curvature-dependent mobility contrast is strongest before both basins become mobile and global overshooting dominates. We further tested the effect of stochastic landscape roughness on the mobility-contrast prediction (Fig. 2d). The prediction becomes less reliable when the noise is too weak to generate sufficient mobility or too strong to preserve the original sharp-flat basin structure. It is most reliable at intermediate noise levels, where stochastic fluctuations drive local exploration without destroying the underlying curvature contrast.

The absolute mobility measures from the sharp and flat basins are shown in Fig. 2e. Across learning rates, the flat basin exhibits higher absolute mobility than the sharp basin, consistent with the weaker confinement of low-curvature regions. At small learning rates, mobility from both basins is suppressed. At large learning rates, mobility is enhanced from both sharp and flat basins, which destroys sharp-flat selectivity and causes overshooting. Thus, the characteristic learning rate does not correspond to maximizing the absolute mobility probability from either basin, but rather identifies a regime that maximizes the contrast between mobility in flat regions and confinement in sharp regions. We further analysed the role of the moment decay parameters (β_1 and β_2) (Appendix Figs. 3-4) and found that β_1 may determine whether a peak appears in the relative mobility contrast, while β_2 controls the peak magnitude. These results provide an intuitive demonstration of the mechanism behind the predicted α_c and its role in guiding parameter space exploration in adaptive optimizers.

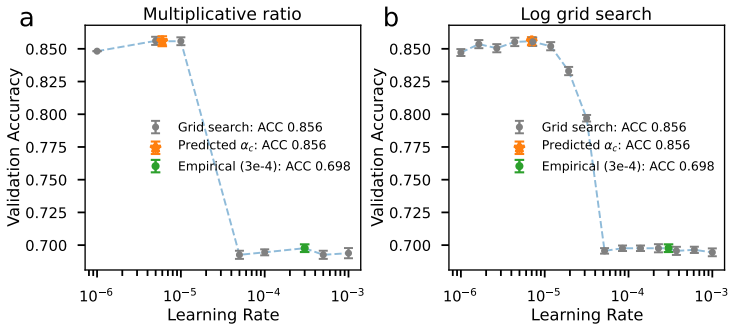


Figure 3: **Predicted characteristic learning rate versus the empirical grid-search optimum on the language dataset.** Training dynamics and validation accuracy for fine-tuning Llama-1.1B on CoLA benchmark across learning rates selected via multiplicative grid search (10^{-6}-10^{-3}), along with theoretically predicted and empirical choices. Error bars represent the standard deviation across 10 independent runs. The theoretically predicted learning rate (red) attains a generalization performance closely matching that of the best grid-searched configurations, while avoiding costly hyperparameter exploration.

QNLI-ModernBERT-base	
Method	Dev accuracy (%)
Ours (31-GE predictor)	**85.88 ± 0.25**
D-Adaptation	85.49 ± 0.45
Prodigy	85.74 ± 0.56
L-DoG	84.21 ± 0.88
Schedule-Free + 1% sweep	85.47 ± 0.16
$\sqrt{b}$ transfer	85.66 ± 0.32
$38/\rho$	82.41 ± 1.67
Adam 1% sweep	85.24 ± 0.73
Empirical choice	83.90 ± 0.14

RTE-Qwen3-1.7B	
Method	Dev accuracy (%)
Ours (31-GE predictor)	**73.72 ± 0.86**
D-Adaptation	70.25 ± 5.30
Prodigy	71.12 ± 2.30
L-DoG	68.88 ± 2.73
Schedule-Free + 1% sweep	72.49 ± 7.44
$\sqrt{b}$ transfer	66.35 ± 2.69
$38/\rho$	65.92 ± 4.89
Adam 1% sweep	69.10 ± 3.89
Empirical choice	70.97 ± 3.06

Table 2: Development accuracy (%) obtained with inexpensive learning-rate selectors, including the empirical choice 3×10^{-5} from the standard BERT fine-tuning grid (Devlin et al., 2019). Baseline references: D-Adaptation (Defazio & Mishchenko, 2023), Prodigy (Mishchenko & Defazio, 2023), L-DoG (Ivgi et al., 2023), Schedule-Free (Defazio et al., 2024), $\sqrt{b}$ transfer (Malladi et al., 2022), and $38/\rho$ (Cohen et al., 2022). All methods use batch size $b = 32$, and QNLI is trained for three epochs and RTE for five epochs. Values are mean ± s.d. over ten seeds.

6 Comparison and Performance Sensitivity

Training from scratch. In empirical tasks, we first train a five-layer MLP on MNIST and a two-layer CNN and a ResNet-28 on CIFAR-10 (Appendix Table 1). Across architectures, α_c lies within a near-optimal plateau, with smaller learning rates slowing optimization and larger values reducing stability or accuracy. In the CNN/CIFAR-10 grid comparison, performance at α_c is close to that at the best searched value, with no statistically significant difference ($t(9) = -0.35$, $p = 0.734$), and loss spikes coincide with increases in $\lambda_{\max}$ and the Hessian trace (Appendix Figs. 5-8). We further pretrain a 1B-parameter Llama model on OpenWebMath, updating all parameters. Our predictor selects $(1.6 \pm 0.2) \times 10^{-5}$, close to the grid-optimal 2×10^{-5}, with a validation loss only 0.113% above the grid optimum (95% CI: 0.049%-0.176% in Appendix Table 4).

Transformer fine-tuning. We next fine-tune BERT-base, GPT-base and Llama-base on SST-2 and CoLA tasks (Socher et al., 2013; Warstadt et al., 2018). Appendix Table 2 summarizes all language-model settings. On CoLA dataset, α_c closely matches the best grid-searched setting within run-to-run variation ($t(9) = -0.91$, $p = 0.39$; Fig. 3). On SST-2 dataset, α_c again falls in the near-optimal plateau, while substantially smaller or larger values respectively slow convergence or destabilize training (Appendix Fig. 9). These experiments extend our prediction from models trained from scratch to pretrained transformers.

Comparison with learning-rate selection baselines. We compare our predictor with seven inexpensive alternatives on QNLI and RTE from GLUE (Wang et al., 2018), using ModernBERT (Warner et al., 2024) and Qwen3 (Yang et al., 2025). D-Adaptation estimates the unknown distance-to-solution scale online (Defazio & Mishchenko, 2023), whereas Prodigy accelerates this adaptation using observed gradients (Mishchenko & Defazio, 2023). L-DoG sets layer-wise steps from parameter displacement and accumulated gradient norms, with the polynomial iterate averaging (Ivgi et al., 2023). Schedule-Free AdamW removes learning-rate decay but still requires a base learning rate (Defazio et al., 2024). The $\sqrt{b}$ rule transfers a reference learning rate across batch sizes (Malladi et al., 2022), here we anchor it at 2×10^{-4} for $b = 16$. The remaining controls are the curvature-based $38/\rho$ boundary (Cohen et al., 2022) and a three-point Adam sweep. The Schedule-Free and Adam sweeps test $\{2 \times 10^{-4}, 8 \times 10^{-4}, 2 \times 10^{-3}\}$ and $\{5 \times 10^{-5}, 2 \times 10^{-4}, 8 \times 10^{-4}\}$ on 1% of the training data, respectively. Unlike the sweep, our predictor does not optimize a downstream validation metric, instead it returns a learning rate from early gradient-noise and curvature measurements. Table 2 focuses on downstream performance. Our predictor attains the highest mean development accuracy in both tested model-task blocks. Because the performance distributions overlap with those of the closest methods on QNLI, we interpret our result as evidence that the predictor reaches a competitive near-optimal performance plateau at low fixed selection cost.

$\alpha_c\times$ scaling factor ($\pm$25% error)	Adam	AdamW_warmup	AdamW_warmup + cosine_decay	AdamW_cosine_decay	Adafactor
$\times 10 \pm 25\%$	0.860 ± 0.010	0.845 ± 0.010	0.875 ± 0.008	0.885 ± 0.008	0.820 ± 0.012
$\times 5 \pm 25\%$	0.925 ± 0.005	0.915 ± 0.006	0.935 ± 0.004	0.938 ± 0.004	0.910 ± 0.008
$\times 2 \pm 25\%$	0.935 ± 0.004	0.938 ± 0.004	0.945 ± 0.003	0.942 ± 0.003	0.935 ± 0.005
$\times 1.5 \pm 25\%$	0.938 ± 0.004	0.941 ± 0.003	0.946 ± 0.003	$\mathbf{0.945} \pm 0.003$	0.939 ± 0.004
$\times 1 \pm 25\%$ (*)	$\mathbf{0.940} \pm 0.004$	$\mathbf{0.943} \pm 0.003$	$\mathbf{0.948} \pm 0.003$	$\mathbf{0.945} \pm 0.003$	$\mathbf{0.941} \pm 0.004$
$\times 0.5 \pm 25\%$	0.938 ± 0.004	0.940 ± 0.004	0.942 ± 0.003	0.940 ± 0.004	0.938 ± 0.005
$\times 0.1 \pm 25\%$	0.925 ± 0.006	0.928 ± 0.005	0.930 ± 0.005	0.928 ± 0.006	0.920 ± 0.008

Table 3: Accuracy (mean $\pm$ s.d.) for Llama on SST-2 across second-moment-based optimizers. Learning rates include a scaling factor with $\pm 25\%$ error, evaluated over 10 independent runs.

Performance sensitivity of optimizers to learning rates. To test whether the predicted learning rate remains useful beyond vanilla Adam, we evaluated AdamW (Loshchilov & Hutter, 2019), warmup (Kosson et al., 2024), cosine decay, and Adafactor (Shazeer & Stern, 2018) under multiplicative deviations from α_c. With warmup or decay, α_c is used as the peak learning rate. As reported in Table 3, α_c consistently corresponds to a regime of improved generalization across all optimizers, with warmup+cosine_decay achieving the highest performance. We found that Adam's intrinsic robustness creates a broad near-optimal plateau around the best learning rate, as previously observed (Sivaprasad et al., 2020). That means the role of α_c is not to identify a unique point with statistically distinct accuracy but to provide a principled estimate of a learning rate within this plateau at a small fixed cost ($\sim$31 GE steps), without requiring repeated full training runs as in grid search. The low standard deviations across independent seeds show the intrinsic robustness of second moment-based optimizers. Comparable trends are observed for ResNet-28 training (Appendix Table 3).

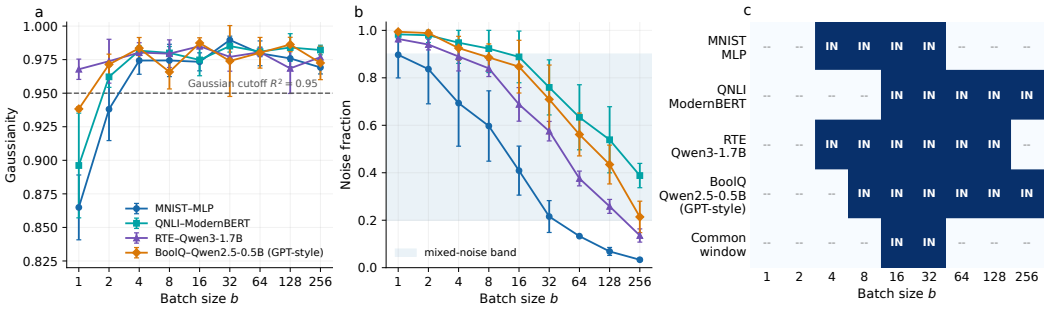


Figure 4: **Evaluation of batch-size applicability.** (**a**) Gaussianity of the measured gradient noise. The dashed line marks the 0.95 threshold. (**b**) Fraction of the gradient second moment contributed by mini-batch noise. The blue band/region marks the intermediate-noise regime. (**c**) Batch sizes passing both checks. Points show mean $\pm$ s.d. over ten sampling seeds.

Batch-size evaluation. Our derivation assumes that mini-batch gradient fluctuations are approximately Gaussian and that both stochastic noise and the deterministic gradient contribute appreciably to Adam's second moment estimate. This intermediate-noise condition is not guaranteed at every batch size (Appendix Fig. 10).

At very small batch sizes, gradient noise can become heavy-tailed and distort the loss landscape (Simsekli et al., 2019), such that stable optimization requires the initial learning rate to remain below the predicted α_c. At very large batch sizes, mini-batch noise is strongly reduced and the dynamics may become nearly deterministic (Keskar et al., 2017; Cohen et al., 2021; Zhang & Tang, 2025), requiring learning rates larger than α_c to sustain sufficient exploration. This near-deterministic regime can occur at smaller batch sizes for Adam than for SGD because second moment normalization suppresses stochasticity more aggressively (Zhang et al., 2020a). Consistent with this interpretation, the noisy two-basin toy model in Fig. 2d shows that the predicted mobility-contrast learning rate is most accurate at intermediate noise.

To identify batch sizes within the regime supported by our learning-rate predictor, we use two empirical criteria (Fig. 4). Panel a requires a Gaussianity score of at least 0.95 for the gradient noise, while Panel b requires the mini-batch noise to contribute between 20% and 90% of the gradient second moment. A batch size is labeled "IN" in Panel c only when both criteria are consistently satisfied across repeated measurements. This yields suitable batch-size ranges of 4-32 for MNIST-MLP, 16-256 for QNLI-ModernBERT, 4-128 for RTE-Qwen3-1.7B, and 8-256 for BoolQ-Qwen2.5-0.5B (GPT-style) (Yang et al., 2024). Similar ranges are observed across the tested vision and language tasks, suggesting that structured correlations in both data modalities may induce comparable effective gradient-noise regimes during mini-batch training.

7 Discussion

A central implication of this result is that adaptive second moment estimator should not be viewed only as a numerical device for stabilizing gradients. The resulting mobility contrast is closely connected to generalization (Yang et al., 2023). High-curvature regions are more sensitive to parameter and sampling perturbations, so reducing their relative absorbing strength increases the chance that training remains in or reaches perturbation-stable low-curvature regions. From this perspective, the selection of the learning rate can be formulated as a measurable dynamical problem rather than a purely empirical hyperparameter search. The predictable α_c provides a principled initialization scale that places training near this favorable regime without repeated full training runs.

These conclusions are obtained under several simplifying assumptions and suggest natural directions for extension. First, our analysis treats the moment decay parameters β_1 and β_2 as fixed algorithmic constants, typically set to their default values, although the toy model shows that they contribute to shaping the mobility contrast. These approximations define a reduced mobility theory rather than an exact description of Adam. Our observations suggest that β_1 influences the emergence of the relative mobility peak, whereas β_2 primarily modulates its magnitude. The default values lie in a regime where the predicted α_c remains consistent with the observed mobility optimum (Appendix Fig. 4). Second, although the sharp-flat temperature asymmetry induced by second moment normalization remains robust in AdamW (Table 3 and Appendix Table 3), our derivation does not explicitly include the deterministic drift introduced by decoupled weight decay. Third, the exponential form captures only the leading learning-rate dependence of local mobility. Thus, our work is to identify the learning rate scale at which second moment normalization maximally separates low-curvature mobility from high-curvature confinement, rather than absolute transition probabilities. Moreover, extending the theory beyond the dominant mode of the Adam-preconditioned local operator to fully high-dimensional landscapes remains an open challenge.

Taken together, our results support a view of adaptive optimization as a stochastic search process shaped by feedback through the second moment estimator v_t. In this physical picture, the learning rate controls the balance between mini-batch-driven diffusion and curvature-induced confinement, and its characteristic value can be estimated from early training statistics by maximizing a sharp-flat mobility contrast. The direct baseline comparison further shows that this estimate reaches the highest mean development accuracy in the tested ModernBERT/QNLI and Qwen3/RTE settings at a small fixed selection cost. Therefore, our work provides a physical description of adaptive optimization in which learning proceeds as a curvature-modulated nonequilibrium diffusion process.

AI USE STATEMENT

All research ideas, theoretical derivations, experimental design, and analyses were conducted solely by the authors. The use of Large Language Models (LLMs) was limited to minor language refinement, including grammar correction and improvements to sentence clarity, and all such edits were checked by the authors.

ETHICS STATEMENT

This work is purely theoretical and computational. It does not involve human subjects, personally identifiable information, or sensitive data.

REPRODUCIBILITY STATEMENT

A complete description of the experimental settings, including model architectures, optimization hyperparameters, and training procedures, is provided in the main text and in the Appendix.

REFERENCES

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. *International Conference on Machine Learning*, pp. 948–1024, 2022.

Zhiwei Bai, Zhangchen Zhou, Jiajie Zhao, Xiaolong Li, Zhiyu Li, Feiyu Xiong, Hongkang Yang, Yaoyu Zhang, and Zhi-Qin John Xu. Adaptive preconditioners trigger loss spikes in adam. *arXiv preprint*, pp. arXiv:2506.04805, 2025.

Carlo Baldassi, Fabrizio Pittorino, and Riccardo Zecchina. Shaping the learning landscape in neural networks around wide flat minima. *Proceedings of the National Academy of Sciences*, 117(1): 161–170, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *International Conference on Learning Representations*, pp. 1–10, 2018.

Hao Chen, Lili Zheng, Raed Al Kontar, and Garvesh Raskutti. Gaussian process parameter estimation using mini-batch stochastic gradient descent: convergence guarantees and empirical benefits. *Journal of Machine Learning Research*, 23(227):1–59, 2022.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *International Conference on Learning Representations*, 2021.

Jeremy M Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, David Cardoze, Zachary Nado, George E Dahl, et al. Adaptive gradient methods at the edge of stability. *arXiv preprint arXiv:2207.14484*, 2022.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In *International Conference on Machine Learning*, pp. 7449–7479. PMLR, 2023.

Aaron Defazio, Xingyu Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. *Advances in Neural Information Processing Systems*, 37: 9974–10007, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1:4171–4186, 2019.

Xuehao Ding, HT Quan, and Yuhai Tu. Stochastic thermodynamics of score matching in diffusion models. *arXiv preprint*, pp. arXiv:2606.17252, 2026.

Ling Feng, Lin Zhang, and Choy Heng Lai. Optimal machine intelligence at the edge of chaos. *arXiv preprint*, pp. arXiv:1909.05176, 2019.

Yu Feng and Yuhai Tu. The inverse variance–flatness relation in stochastic gradient descent is critical for finding flat minima. *Proceedings of the National Academy of Sciences*, 118(9):e2015617118, 2021.

Yu Feng, Wei Zhang, and Yuhai Tu. Activity–weight duality in feed-forward neural networks reveals two co-determinants for generalization. *Nature Machine Intelligence*, 5(8):908–918, 2023.

Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115, 2019.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. *International Conference on Machine Learning*, pp. 2232–2241, 2019.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training imagenet in 1 hour. *arXiv preprint*, pp. arXiv:1706.02677, 2017.

Jeff Z HaoChen, Colin Wei, Jason Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. *Conference on Learning Theory*, pp. 2315–2357, 2021.

Maor Ivgi, Oliver Hinder, and Yair Carmon. Dog is SGD's best friend: A parameter-free dynamic step size schedule. In *International Conference on Machine Learning*, pp. 14465–14499. PMLR, 2023.

Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. *Advances in Neural Information Processing Systems*, 37:111760–111801, 2024.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *International Conference on Learning Representations*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Analyzing & reducing the need for learning rate warmup in GPT training. *Advances in Neural Information Processing Systems*, 37:2914–2942, 2024.

Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

Zequn Lin, Zhaofan Lu, Zengru Di, and Ying Tang. Learning noise-induced transitions by multi-scaling reservoir computing. *Nature Communications*, 15(1):6584, 2024.

Jing Liu, Ying Tang, and Pan Zhang. Efficient optimization of variational autoregressive networks with natural gradient. *Physical Review E*, 111(2):025304, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations*, 2019.

Andrew Ly and Pulin Gong. Optimization on multifractal loss landscapes explains a diverse range of geometrical and dynamical properties of deep learning. *Nature Communications*, 16(1):3252, 2025.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the sdes and scaling rules for adaptive gradient algorithms. *Advances in Neural Information Processing Systems*, 35: 7697–7711, 2022.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. *arXiv preprint arXiv:2306.06101*, 2023.

Hannes Risken. *Fokker-planck equation*. Springer, 1989.

Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *International Conference on Learning Representations*, 2017.

Christopher J Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, and George E Dahl. Measuring the effects of data parallelism on neural network training. *Journal of Machine Learning Research*, 20(112):1–49, 2019.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. *International Conference on Machine Learning*, pp. 4596–4604, 2018.

Zi-Song Shen, Feng Pan, Yao Wang, Yi-Ding Men, Wen-Biao Xu, Man-Hong Yung, and Pan Zhang. Free-energy machine for combinatorial optimization. *Nature Computational Science*, 5(4):322–332, 2025.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. *International Conference on Machine Learning*, pp. 5827–5837, 2019.

Prabhu Teja Sivaprasad, Florian Mai, Thijs Vogels, Martin Jaggi, and François Fleuret. Optimizer benchmarking needs to account for hyperparameter tuning. *International Conference on Machine Learning*, pp. 9036–9045, 2020.

Samuel L Smith, Benoit Dherin, David GT Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. *International Conference on Learning Representations*, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Philipp C Verpoort, Alpha A Lee, and David J Wales. Archetypal landscapes for deep neural networks. *Proceedings of the National Academy of Sciences*, 117(36):21857–21864, 2020.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, et al. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv.2412.13663*, 2024.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint*, pp. arXiv:1805.12471, 2018.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ning Yang, Chao Tang, and Yuhai Tu. Stochastic gradient descent introduces an effective landscape-dependent regularization favoring flat solutions. *Physical Review Letters*, 130(23):237101, 2023.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *International Conference on Learning Representations*, 2020a.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020b.

Xin-Ya Zhang and Chao Tang. Heavy-tailed update distributions arise from information-driven self-organization in nonequilibrium learning. *Proceedings of the National Academy of Sciences*, 122 (51):e2523012122, 2025.

APPENDIX

A THEORETICAL FRAMEWORK FOR ADAM OPTIMIZER

We first formulate a local stochastic surrogate model for the dynamics of Adam under mini-batch training. Let the empirical loss over the full training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ be

$$F(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; x_i, y_i), \tag{1}$$

where $\ell(\theta; x_i, y_i)$ is the per-sample loss. At training step t, a mini-batch $\mathcal{B}_t \subset \mathcal{D}$ of size b gives the stochastic gradient

$$g_t = \nabla f_{\mathcal{B}_t}(\theta_t) = \frac{1}{b} \sum_{(x,y)\in\mathcal{B}_t} \nabla \ell(\theta_t; x, y). \tag{2}$$

We decompose this mini-batch gradient into the full gradient and a sampling fluctuation,

$$g_t = \nabla F(\theta_t) + \xi_t, \tag{3}$$

where ξ_t denotes the mini-batch gradient noise. For unbiased mini-batch sampling,

$$\mathbb{E}[\xi_t \mid \theta_t] = 0. \tag{4}$$

In realistic neural-network training, the covariance of ξ_t is generally anisotropic, state-dependent, and temporally correlated (Yang et al., 2023; Zhang et al., 2021; HaoChen et al., 2021; Simsekli et al., 2019; Zhu et al., 2019). Our goal here is not to construct a complete stochastic model of this full covariance structure. Instead, we use a local diagonal surrogate that isolates the leading scaling relevant for Adam's second-moment-shaped mobility. In a local coordinate system, we approximate

$$\mathbb{E}[\xi_{i,t}] = 0, \tag{5}$$

$$\mathrm{Var}[\xi_{i,t}] = \frac{\sigma_i^2}{b}, \tag{6}$$

$$\mathbb{E}[\xi_{i,t}\xi_{j,s}] \approx \delta_{ij}\delta_{ts}\frac{\sigma_i^2}{b}, \tag{7}$$

where σ_i^2 is the per-sample gradient-noise variance along coordinate i,

$$\sigma_i^2 = \mathbb{E}_Z\left[(\nabla \ell_i(\theta; Z) - \nabla F_i(\theta))^2\right]. \tag{8}$$

We next introduce the local curvature structure of the loss landscape. Near a reference point $\theta^\star$ associated with a local basin, the empirical loss can be expanded as

$$F(\theta) = F(\theta^\star) + \nabla F(\theta^\star)^\top(\theta - \theta^\star) + \frac{1}{2}(\theta - \theta^\star)^\top H(\theta - \theta^\star) + O(\|\theta - \theta^\star\|^3), \tag{9}$$

where $H = \nabla^2 F(\theta^\star)$ is the Hessian. At a local minimum, $\nabla F(\theta^\star) = 0$, so the leading local geometry is

$$F(\theta) \approx F(\theta^\star) + \frac{1}{2}(\theta - \theta^\star)^\top H(\theta - \theta^\star). \tag{10}$$

For the leading-curvature analysis, we approximate the local geometry by a symmetric Hessian surrogate H and diagonalize it as $H = Q\Lambda Q^\top$. In this local eigenbasis, the quadratic form becomes

$$F(\theta) \approx F(\theta^\star) + \frac{1}{2}\sum_{i=1}^{d} \lambda_i(\theta_i - \theta_i^\star)^2, \tag{11}$$

where λ_i are Hessian eigenvalues. Throughout the derivation, the distinction between sharp and flat regions is relative: regions with larger dominant curvature $\lambda_{\max}$ are treated as sharper, whereas regions with smaller dominant curvature are treated as flatter. This local quadratic approximation provides the curvature scale entering the sharp-region effective temperature derived below.

A.1 Optimizer dynamics

At each training step t, Adam maintains exponential moving averages of the stochastic gradient and its element-wise square:

$$\begin{aligned} m_{t+1} &= \beta_1 m_t + (1-\beta_1) g_t, \\ v_{t+1} &= \beta_2 v_t + (1-\beta_2) g_t^2, \\ \theta_{t+1} &= \theta_t - \alpha \frac{m_{t+1}}{\sqrt{v_{t+1}+\epsilon}}, \end{aligned} \tag{12}$$

where $g_t = \nabla f_{\mathcal{B}_t}(\theta_t)$, $\beta_1, \beta_2 \in [0,1)$ are decay coefficients, α is the global learning rate, and $\epsilon > 0$ is used for numerical stability. In common PyTorch implementations, the default Adam hyperparameters are $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ (Paszke et al., 2019). The key difference from stochastic gradient descent (SGD) is the coordinate-wise normalization by $\sqrt{v_t}$, which reshapes both the deterministic drift and the stochastic fluctuations.

A.2 Variance of the first-moment fluctuation

For the i-th coordinate of the parameter vector, the first-moment update is given by

$$m_{i,t+1} = \beta_1 m_{i,t} + (1-\beta_1) g_{i,t}. \tag{13}$$

We define the deviation from the conditional expectation as

$$\delta m_{i,t} = m_{i,t} - \mathbb{E}[m_{i,t} \mid \{\theta_s\}_{s \le t}]. \tag{14}$$

To obtain a tractable leading-order approximation, we approximate the variance of the first-moment fluctuation by the mini-batch gradient-noise variance,

$$\mathrm{Var}[\delta m_i] \approx \frac{\sigma_i^2}{b}. \tag{15}$$

This approximation captures the leading dependence of the stochastic update scale on the noise amplitude σ_i and batch size b, which is sufficient for deriving the closed-form scaling of α_c. The first-moment coefficient β_1 also controls the temporal memory and inertia of Adam's updates, and therefore can affect finite-window mobility and the existence of the mobility-contrast peak. These effects are not included in the closed-form scaling argument and are examined numerically in Appendix Figs. 3-4.

A.3 Second-moment estimate

The second-moment accumulator satisfies

$$v_{i,t+1} = \beta_2 v_{i,t} + (1-\beta_2) \left[\nabla F_i(\theta_t) + \xi_{i,t}\right]^2. \tag{16}$$

Expanding the squared term gives

$$\left[\nabla F_i + \xi_i\right]^2 = (\nabla F_i)^2 + 2 \nabla F_i \xi_i + \xi_i^2. \tag{17}$$

Taking the local expectation and using $\mathbb{E}[\xi_i] = 0$ eliminates the cross term. Under a local quasi-stationary approximation for v_i, we obtain

$$\langle v_i \rangle \approx \mathbb{E}[(\nabla F_i)^2] + \frac{\sigma_i^2}{b}. \tag{18}$$

Thus, Adam's second moment contains both a deterministic contribution from the squared local gradient and a stochastic contribution from mini-batch noise. This dependence is the source of curvature-dependent mobility in the effective-temperature framework.

A.4 Effective temperature

We isolate the stochastic component of Adam's update along coordinate i:

$$\Delta\theta_i^{\mathrm{stoch}} = -\alpha \frac{\delta m_i}{\sqrt{v_i + \epsilon}}. \tag{19}$$

Over a short local window, we treat v_i as quasi-stationary and weakly correlated with the fast component of δm_i. This gives the local variance approximation

$$\mathrm{Var}\left[\Delta\theta_i^{\mathrm{stoch}}\right] \approx \alpha^2 \frac{\mathrm{Var}[\delta m_i]}{\langle v_i \rangle + \epsilon}. \tag{20}$$

In the regimes considered below, $\langle v_i \rangle \gg \epsilon$, so ϵ contributes only a lower-order correction and will be omitted. To relate the discrete update variance to a diffusion scale, we use the variance-diffusion relation

$$\mathrm{Var}[\Delta\theta] = 2D\,\Delta t. \tag{21}$$

We measure diffusion per unit of processed data rather than per optimizer step. Since one mini-batch update aggregates information from b samples, we use the sample-normalized effective time interval

$$\Delta t_{\mathrm{eff}} = \frac{1}{b}. \tag{22}$$

This sample-normalized time scale allows us to compare parameter mobility per unit of processed data rather than per optimizer step, which is important because a single mini-batch update aggregates information from b samples. With this convention,

$$D_i^{\mathrm{eff}} = \frac{\mathrm{Var}[\Delta\theta_i^{\mathrm{stoch}}]}{2\Delta t_{\mathrm{eff}}} \approx \frac{b\alpha^2}{2} \frac{\mathrm{Var}[\delta m_i]}{\langle v_i \rangle}. \tag{23}$$

We then define a noise-normalized diffusion scale, denoted T_i^{eff}:

$$T_i^{\mathrm{eff}} = \frac{D_i^{\mathrm{eff}}}{\sqrt{\mathrm{Var}(\xi_i)}}, \tag{24}$$

which quantifies how efficiently mini-batch stochasticity is converted into parameter-space motion after Adam's second-moment normalization. Accordingly, the theory below uses T^{eff} to compare the leading learning-rate dependence of local mobility, rather than to construct an equilibrium Boltzmann distribution. Since $\sqrt{\mathrm{Var}(\xi_i)} = \sigma_i/\sqrt{b}$, this gives

$$T_i^{\mathrm{eff}} \approx \frac{b^{3/2}\alpha^2}{2\sigma_i} \frac{\mathrm{Var}[\delta m_i]}{\langle v_i \rangle}. \tag{25}$$

Finally, using Eq. 15, we obtain

$$T_i^{\mathrm{eff}} \approx \frac{\alpha^2 \sigma_i \sqrt{b}}{2\langle v_i \rangle}. \tag{26}$$

This effective temperature is not an equilibrium thermodynamic temperature. It is a local diffusion-efficiency measure: it quantifies how efficiently mini-batch stochasticity is converted into parameter-space motion after Adam's second-moment normalization. Equation 26 is the starting point for comparing low- and high-curvature regions.

B Analysis of landscape regions

We now evaluate Eq. 26 in low- and high-curvature regions. The distinction enters through $\langle v_i \rangle$, which contains both the squared deterministic gradient and the mini-batch noise variance.

B.1 Low-curvature regions

In a low-curvature region, the deterministic gradient contribution is small compared with the noise contribution, so

$$\langle v_i \rangle \approx \frac{\sigma_i^2}{b}. \tag{27}$$

Substituting this into Eq. 26 gives

$$T_{i,\mathrm{flat}}^{\mathrm{eff}} = \frac{\alpha^2 \sigma_i \sqrt{b}}{2(\sigma_i^2/b)} = \frac{\alpha^2 b^{3/2}}{2\sigma_i}. \tag{28}$$

Thus, in low-curvature regions, Adam's second-moment normalization allows relatively large noise-normalized parameter mobility.

B.2 HIGH-CURVATURE REGIONS

In a high-curvature region, the deterministic gradient contribution to the second moment cannot be neglected. Along the dominant Hessian direction, the local quadratic approximation gives

$$F(\theta) \approx F(\theta^\star) + \frac{1}{2}\lambda_{\max}(\theta - \theta^\star)^2, \tag{29}$$

and hence

$$\nabla F(\theta) \approx \lambda_{\max}(\theta - \theta^\star). \tag{30}$$

The squared gradient contribution is therefore controlled by the local displacement from the basin center:

$$\mathbb{E}[(\nabla F)^2] \approx \lambda_{\max}^2 \mathbb{E}[(\theta - \theta^\star)^2]. \tag{31}$$

We next introduce a local displacement-scaling closure to estimate the curvature-dependent contribution to $\langle v_i \rangle$. For the quadratic surrogate, we assume that the characteristic fluctuation amplitude is inversely suppressed by the restoring curvature,

$$\mathbb{E}\big[(\theta - \theta^\star)^2\big] \sim \frac{T^{\text{eff}}}{\lambda_{\max}}, \tag{32}$$

which is used as a local scaling closure for the reduced stochastic dynamics, rather than as an equilibrium equipartition identity. Since $\nabla F(\theta) \approx \lambda_{\max}(\theta - \theta^\star)$ along the dominant curvature direction,

$$\mathbb{E}\big[(\nabla F)^2\big] \sim \lambda_{\max} T^{\text{eff}}. \tag{33}$$

In sharp directions, v_t produces a twofold self-limiting feedback: it suppresses the stochastic update variance through Adam's normalized step and simultaneously restricts the accessible displacement within the quadratic well.

The equation above describes the curvature-dependent gradient contribution at a given displacement scale, whereas in Adam the accumulated second moment also feeds back on that displacement through the normalized update. We therefore model the compound action of these two coupled feedback channels by a curvature-renormalized squared closure,

$$T_{\text{sharp}}^{\text{eff}} \approx \frac{C_{\text{neq}}}{\left(\lambda_{\max} T_{\text{sharp}}^{\text{eff}} + s\right)^2}, \tag{34}$$

where

$$C_{\text{neq}} = \frac{\alpha^2 \sigma_i b^{1/2}}{2}, \qquad s = \frac{\sigma_i^2}{b}. \tag{35}$$

Appendix Fig. 11 directly tests the exponent in Eq. 34 using the actual bias-corrected Adam state $\langle \hat{v} \rangle$. At fixed α, σ_i, b, β_1, and β_2, we construct the operational temperature $T^{\text{eff}} = C_{\text{neq}}/\langle \hat{v} \rangle$ and regress $C_{\text{neq}}/T^{\text{eff}}$ against $\lambda T^{\text{eff}} + s$. The squared closure predicts a log–log slope of 2, whereas a one-power closure predicts 1. In the independently selected curvature-dominated window, the free slope is 2.076 (95% CI $[1.838, 2.315]$; $R^2 = 0.9961$) and the fixed slope-2 fit gives $R^2 = 0.9948$, compared with 0.7284 for slope 1. Thus, we use Eq. 34 as the empirically supported sharp-direction closure throughout the reduced theory. In the curvature-dominated regime $\lambda_{\max} T_{\text{sharp}}^{\text{eff}} \gg s$, Eq. 34 reduces to

$$T_{\text{sharp}}^{\text{eff}} \approx \frac{C_{\text{neq}}}{(\lambda_{\max} T_{\text{sharp}}^{\text{eff}})^2}, \tag{36}$$

or

$$\lambda_{\max}^2 \left(T_{\text{sharp}}^{\text{eff}}\right)^3 = \frac{\alpha^2 \sigma_i b^{1/2}}{2}. \tag{37}$$

Solving gives

$$T_{i,\text{sharp}}^{\text{eff}} = \left(\frac{\alpha^2 \sigma_i b^{1/2}}{2\lambda_{\max}^2}\right)^{1/3}. \tag{38}$$

Compared with the low-curvature expression in Eq. 28, the high-curvature effective temperature grows as a cube root of α^2 and is suppressed by $\lambda_{\max}^{-2/3}$. This curvature-dependent suppression is the key mechanism behind the mobility contrast.

C LOCAL MOBILITY PROXY

The effective temperatures above provide local diffusion scales for Adam's stochastic dynamics. To compare the leading learning-rate dependence of mobility in low- and high-curvature regions, we therefore introduce a Kramers-inspired exponential proxy rather than an absolute transition-rate formula,

$$M = A \exp\left(-\frac{\Delta E}{T^{\text{eff}}}\right), \tag{39}$$

where ΔE is a matched effective barrier scale and A is an effective prefactor. This expression is not intended to predict the final basin selection. Instead, it is used to compare the leading-order dependence of local mobility on the learning rate under a fixed barrier. Throughout the derivation, we measure the barrier ΔE and the effective temperature T^{eff} in units of a reference loss scale E_0, and parameter displacements in units of a reference parameter scale θ_0, working in natural units with $E_0 = \theta_0 = 1$, analogous to setting $k_B = 1$ in statistical physics. This choice corresponds to the per-sample mean loss computed by standard deep learning libraries and to the standard parameterization without weight multipliers, which is the convention used in all our experiments.

For low-curvature regions, substituting Eq. 28 into Eq. 39 gives

$$M_{\text{flat}} \propto \exp\left(-\frac{\Delta E}{T^{\text{eff}}_{\text{flat}}}\right) = \exp\left(-\frac{2\sigma\Delta E}{\alpha^2 b^{3/2}}\right). \tag{40}$$

We define

$$C_{\text{flat}} = \frac{2\sigma\Delta E}{b^{3/2}}, \tag{41}$$

so that

$$M_{\text{flat}} \propto \exp\left(-\frac{C_{\text{flat}}}{\alpha^2}\right). \tag{42}$$

For high-curvature regions, substituting Eq. 38 gives

$$M_{\text{sharp}} \propto \exp\left(-\frac{\Delta E}{T^{\text{eff}}_{\text{sharp}}}\right) = \exp\left(-\frac{\Delta E (2\lambda^2_{\max})^{1/3}}{b^{1/6}\alpha^{2/3}\sigma^{1/3}}\right). \tag{43}$$

Defining

$$C_{\text{sharp}} = \Delta E \frac{(2\lambda^2_{\max})^{1/3}}{b^{1/6}\sigma^{1/3}}, \tag{44}$$

we obtain

$$M_{\text{sharp}} \propto \exp\left(-\frac{C_{\text{sharp}}}{\alpha^{2/3}}\right). \tag{45}$$

D CHARACTERISTIC LEARNING RATE

We define the relative mobility contrast between low- and high-curvature regions as

$$R(\alpha) = \frac{M_{\text{flat}}}{M_{\text{sharp}}}. \tag{46}$$

Using Eqs. 42 and 45,

$$R(\alpha) = \exp\left(-\frac{C_{\text{flat}}}{\alpha^2} + \frac{C_{\text{sharp}}}{\alpha^{2/3}}\right) \tag{47}$$

is a finite-window local mobility-contrast measure. The characteristic learning rate is defined as the value of α that maximizes $R(\alpha)$ before global overshooting dominates. Differentiating gives

$$\frac{dR}{d\alpha} = R(\alpha)\left(\frac{2C_{\text{flat}}}{\alpha^3} - \frac{2C_{\text{sharp}}}{3\alpha^{5/3}}\right). \tag{48}$$

Setting $dR/d\alpha = 0$ yields

$$\frac{2C_{\text{flat}}}{\alpha_c^3} = \frac{2C_{\text{sharp}}}{3\alpha_c^{5/3}}, \tag{49}$$

and therefore

$$\alpha_c^{4/3} = \frac{3C_{\text{flat}}}{C_{\text{sharp}}}. \tag{50}$$

Using

$$C_{\text{flat}} = \frac{2\sigma\Delta E}{b^{3/2}}, \qquad C_{\text{sharp}} = \Delta E \frac{(2\lambda_{\max}^2)^{1/3}}{b^{1/6}\sigma^{1/3}}, \tag{51}$$

we obtain

$$\frac{C_{\text{flat}}}{C_{\text{sharp}}} = \frac{2\sigma\Delta E/b^{3/2}}{\Delta E(2\lambda_{\max}^2)^{1/3}/(b^{1/6}\sigma^{1/3})} \tag{52}$$

$$= \frac{2^{2/3}\sigma^{4/3}}{b^{4/3}\lambda_{\max}^{2/3}}. \tag{53}$$

Substituting this into Eq. 50 gives

$$\alpha_c^{4/3} = \frac{3 \cdot 2^{2/3}\sigma^{4/3}}{b^{4/3}\lambda_{\max}^{2/3}}. \tag{54}$$

Taking the three-fourths power yields

$$\alpha_c = \left(3 \cdot 2^{2/3}\right)^{3/4} \frac{\sigma}{b\lambda_{\max}^{1/2}}. \tag{55}$$

Since

$$\left(3 \cdot 2^{2/3}\right)^{3/4} \approx 3.22, \tag{56}$$

the characteristic learning rate is

$$\alpha_c \approx 3.22 \frac{\sigma}{b\lambda_{\max}^{1/2}}. \tag{57}$$

Restoring the reference scales, dimensional consistency fixes $\alpha_c \approx 3.22\,\theta_0\sigma/(b\sqrt{\lambda_{\max}E_0})$. Under a loss rescaling $F \to cF$ (for $\epsilon \ll \sqrt{v}$), σ, $\lambda_{\max}$, and E_0 all scale by c, leaving α_c unchanged, whereas under a reparameterization $\theta \to k\theta$, $\sigma \to \sigma/k$, $\lambda_{\max} \to \lambda_{\max}/k^2$, and $\theta_0 \to k\theta_0$, so that α_c scales by k, and both match the corresponding transformations of Adam's dynamics. These reference scales are conventions of the reduced theory rather than derived quantities, and Eq. 57 applies as written under the convention stated above.

To verify this extremum is a maximum, we consider the second derivative of $R(\alpha)$:

$$\frac{d^2R}{d\alpha^2} = R(\alpha)\left(\phi''(\alpha) + (\phi'(\alpha))^2\right), \tag{58}$$

where $\phi(\alpha) = -\frac{C_{\text{flat}}}{\alpha^2} + \frac{C_{\text{sharp}}}{\alpha^{2/3}}$ is the exponent. At the extremum, $\phi'(\alpha) = 0$, so the second derivative simplifies to

$$\frac{d^2R}{d\alpha^2} = R(\alpha)\phi''(\alpha). \tag{59}$$

Since $R(\alpha) > 0$ for all learning rate $\alpha > 0$, the sign of $\frac{d^2R}{d\alpha^2}$ is determined entirely by $\phi''(\alpha)$. The second derivative of the exponent is

$$\phi''(\alpha) = \frac{d^2}{d\alpha^2}\Big(-\frac{C_{\text{flat}}}{\alpha^2} + \frac{C_{\text{sharp}}}{\alpha^{2/3}}\Big) = -\frac{6C_{\text{flat}}}{\alpha^4} + \frac{10C_{\text{sharp}}}{9\alpha^{8/3}}. \tag{60}$$

Substituting $\alpha_c^{4/3} = 3C_{\text{flat}}/C_{\text{sharp}}$ yields

$$\phi''(\alpha_c) = -\frac{6C_{\text{flat}}}{\alpha_c^4} + \frac{10C_{\text{flat}}}{3\alpha_c^4} = -\frac{8C_{\text{flat}}}{3\alpha_c^4} < 0. \tag{61}$$

Since C_{flat} and α_c are positive, this confirms that $R(\alpha)$ reaches a maximum at α_c. Thus, α_c is the learning rate at which the local mobility contrast between low- and high-curvature regions is maximized.

E Operational evaluation of batch-size applicability

We evaluate whether a candidate batch size satisfies the Gaussian-like, intermediate-noise assumptions underlying our predictor at the checkpoint where the learning-rate statistics are measured. We freeze both the model checkpoint and the dominant raw-Hessian direction u, disable other sources of stochasticity such as dropout, and sample 48 randomly shuffled mini-batches for each candidate batch size. We repeat this procedure using ten independent sampling seeds. For each seed, Gaussianity is quantified by the Q-Q coefficient R^2 of the batch-gradient fluctuations projected onto u. The Gaussianity check is passed when $R^2 \geq 0.95$ for at least two of the ten seeds.

We additionally measure the fraction of the projected gradient second moment attributable to mini-batch noise,

$$q = \frac{\mathrm{Var}(u^\top g_B)}{\mathbb{E}[(u^\top g_B)^2]}. \tag{62}$$

Values near zero correspond to nearly deterministic gradients, whereas values near one indicate noise-dominated dynamics. We therefore require the mean noise fraction across the ten sampling seeds to satisfy $0.2 \leq q \leq 0.9$. A batch size is suitable when it passes both the Gaussianity and noise-balance checks.

F APPENDIX TABLES

In this section, we provide additional details of the experimental setups and quantitative analyses used to validate our theoretical predictions.

Model	Dataset	Parameters	Input / architecture	Batch size(s)	Evaluation
MLP-5layer	MNIST	∼47.4K	784; five FC layers, width 50	1-256 (diagnostic); 16-256 (sweep)	Dev./test accuracy
CNN-2conv	CIFAR-10	∼2.12M	$32\times32\times3$; two conv. + two FC	8-64 (diagnostic); 64 (training)	Test accuracy
ResNet-28	CIFAR-10	∼36.48M	28 layers ($6n+4$, $n=4$), width 10	8-64 (diagnostic); 4, 16 (training)	Test accuracy
Llama	OpenWebMath	∼1.1B	Full-parameter continued pretraining	8 (training)	Validation loss

Appendix Table 1: Complete model-dataset inventory for experiments trained from scratch and for the frozen-checkpoint batch diagnostics. A range 1-256 denotes powers of two $\{1, 2, 4, \ldots, 256\}$; the MNIST performance sweep uses $\{16, 32, 64, 128, 256\}$.

Model	Dataset(s)	Parameters	Fine-tuning scope	Batch size	Evaluation
BERT-base-uncased	SST-2	∼110M	Pretrained-model fine-tuning	32 (fine-tuning)	Dev. accuracy
GPT	SST-2	∼124M	Pretrained-model fine-tuning	32 (fine-tuning)	Dev. accuracy
Llama	SST-2, CoLA	∼1.1B	Pretrained-model fine-tuning	32 (fine-tuning)	Dev. accuracy
ModernBERT-base	QNLI	∼149M	Final block, norm, and head	1-256 (diagnostic); 32 (fine-tuning)	Dev. accuracy
Qwen3	RTE	∼1.7B	Final block, norm, and head	1-256 (diagnostic); 32 (fine-tuning)	Dev. accuracy
Qwen2.5 (GPT-style)	BoolQ	∼0.5B	Final block, norm, and head	1-256 (diagnostic)	Batch-noise diagnostics

Appendix Table 2: Complete model-dataset inventory for transformer fine-tuning experiments and frozen-checkpoint diagnostics. The QNLI and RTE baseline comparisons train the final Transformer block, final normalization layer, and classification head for three epochs at batch size 32. The QNLI, RTE, and BoolQ diagnostics use $b \in \{1, 2, 4, \ldots, 256\}$. Qwen2.5 is included as an additional GPT-style decoder-only architecture.

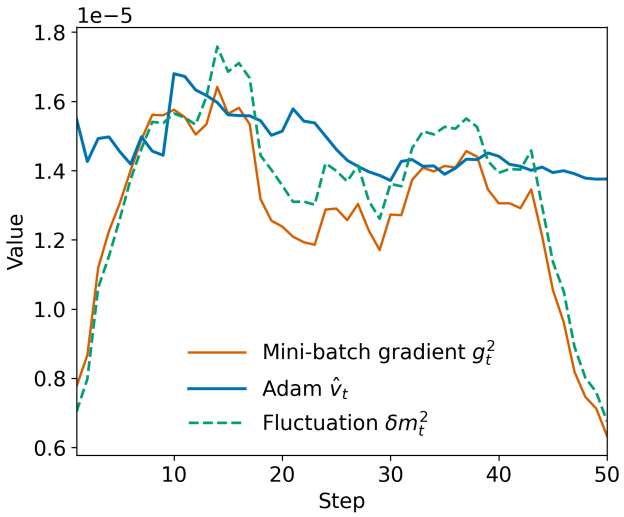
$\alpha_c\times$ scaling factor ($\pm25\%$ error)	Adam	AdamW_warmup	AdamW_warmup + cosine_decay	AdamW_cosine_decay	Adafactor
$\times10 \pm 25\%$	0.795 ± 0.008	0.776 ± 0.009	0.797 ± 0.005	0.810 ± 0.006	0.790 ± 0.007
$\times5 \pm 25\%$	0.827 ± 0.006	0.809 ± 0.005	0.828 ± 0.006	0.840 ± 0.004	0.822 ± 0.008
$\times2 \pm 25\%$	0.839 ± 0.006	0.823 ± 0.007	0.845 ± 0.005	0.851 ± 0.005	0.832 ± 0.006
$\times1.5 \pm 25\%$	0.867 ± 0.009	0.865 ± 0.004	0.891 ± 0.005	0.881 ± 0.004	0.867 ± 0.007
$\times1 \pm 25\%$ (*)	$\mathbf{0.869} \pm 0.006$	$\mathbf{0.873} \pm 0.006$	$\mathbf{0.899} \pm 0.004$	$\mathbf{0.884} \pm 0.004$	$\mathbf{0.870} \pm 0.004$
$\times0.5 \pm 25\%$	0.751 ± 0.004	0.729 ± 0.009	0.741 ± 0.006	0.753 ± 0.003	0.736 ± 0.008
$\times0.1 \pm 25\%$	0.664 ± 0.008	0.658 ± 0.006	0.661 ± 0.005	0.661 ± 0.008	0.654 ± 0.004

Appendix Table 3: Test accuracy (mean $\pm$ s.d.) for ResNet-28 trained on the CIFAR-10 dataset, evaluated across scaling factors with a $\pm25\%$ estimation error over 10 independent runs

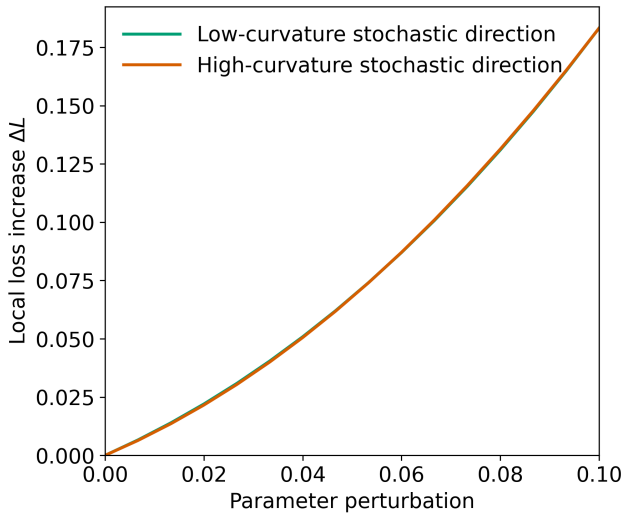
Selection rule	Learning rate	Validation loss
Fixed grid	5×10^{-6}	2.3592 ± 0.0565
Fixed grid	1×10^{-5}	2.3331 ± 0.0576
Fixed grid	2×10^{-5}	$\mathbf{2.3176 \pm 0.0590}$
Fixed grid	3×10^{-5}	2.3298 ± 0.0597
Fixed grid	5×10^{-5}	2.4064 ± 0.0610
Ours (31-GE predictor)	$(1.6 \pm 0.2) \times 10^{-5}$	$\underline{2.3202 \pm 0.0583}$
Empirical choice	3×10^{-4}	3.9323 ± 0.1047

Appendix Table 4: Full-parameter continued pretraining of Llama-1.1B on OpenWebMath using Adam. Our predictor selects $(1.6 \pm 0.2) \times 10^{-5}$ and achieves 2.3202 ± 0.0583 validation loss, compared with 2.3176 ± 0.0590 at the grid-optimal 2×10^{-5}. The empirical Adam choice is 3×10^{-4}. Values are mean $\pm$ s.d. over ten independent seeds. Within each seed, candidates share the initialization, sampled-example stream, and validation batches. All 1,100,048,384 parameters are updated. Training uses batch size 8, context length 256, 1,200 updates, and 2.46M tokens. Prediction takes 9.5 s on NVIDIA A100-SXM4-80GB GPUs, compared with 2,420.3 s for the five complete fixed-grid runs. Bold and underlining denote the lowest and second-lowest mean losses.

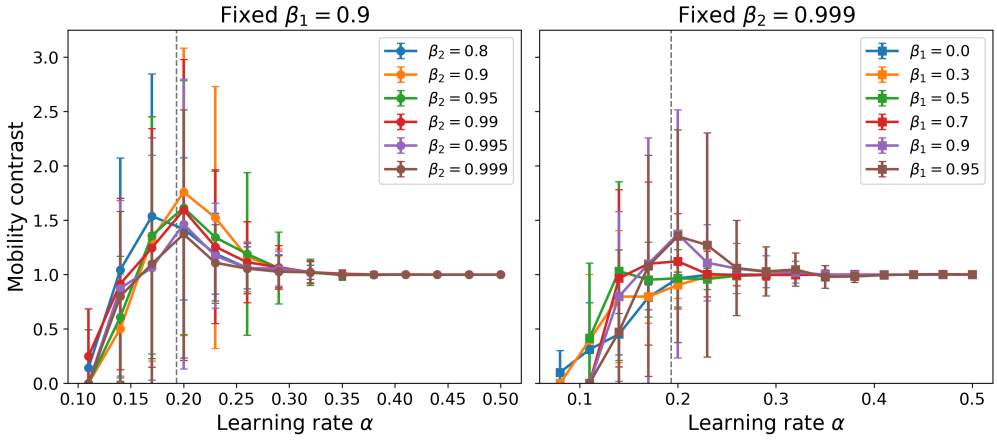
G Appendix Figures



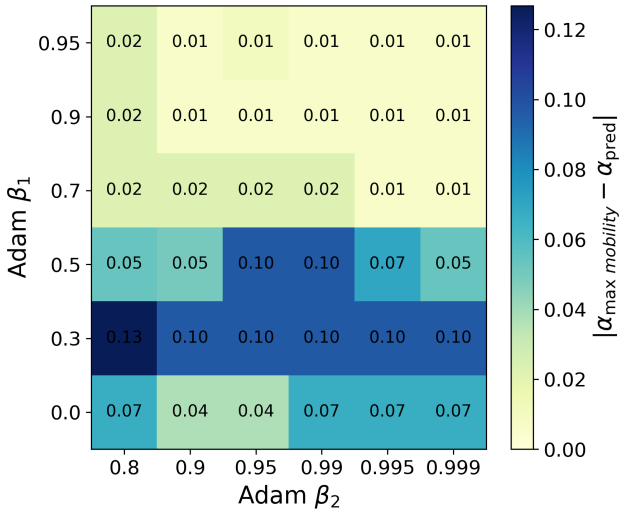
Appendix Figure 1: **Short-horizon timescale separation of the adaptive preconditioner.** To examine the timescale separation, we trained a MLP on MNIST with Adam for 50 optimization steps, corresponding to the early stage of training. At each step, we recorded the mini-batch squared-gradient signal g_t^2, the gradient fluctuation $\delta m_t^2 = \langle (g_t - \hat{m}_{t-1})^2 \rangle$, and Adam bias-corrected second-moment accumulator $\hat{v}_t$. Relative temporal variability was quantified using the coefficient of variation, $\mathrm{CV}(x) = \mathrm{std}(x)/\mathrm{mean}(x)$. The gradient-square signal fluctuates more strongly, with $\mathrm{CV}(g_t^2) = 0.460$, whereas $\hat{v}_t$ evolves more slowly, with $\mathrm{CV}(\hat{v}_t) = 0.055$. These observations are consistent with a timescale separation in which the adaptive second-moment estimator acts as a slow low-pass filter of fast stochastic gradient fluctuations.



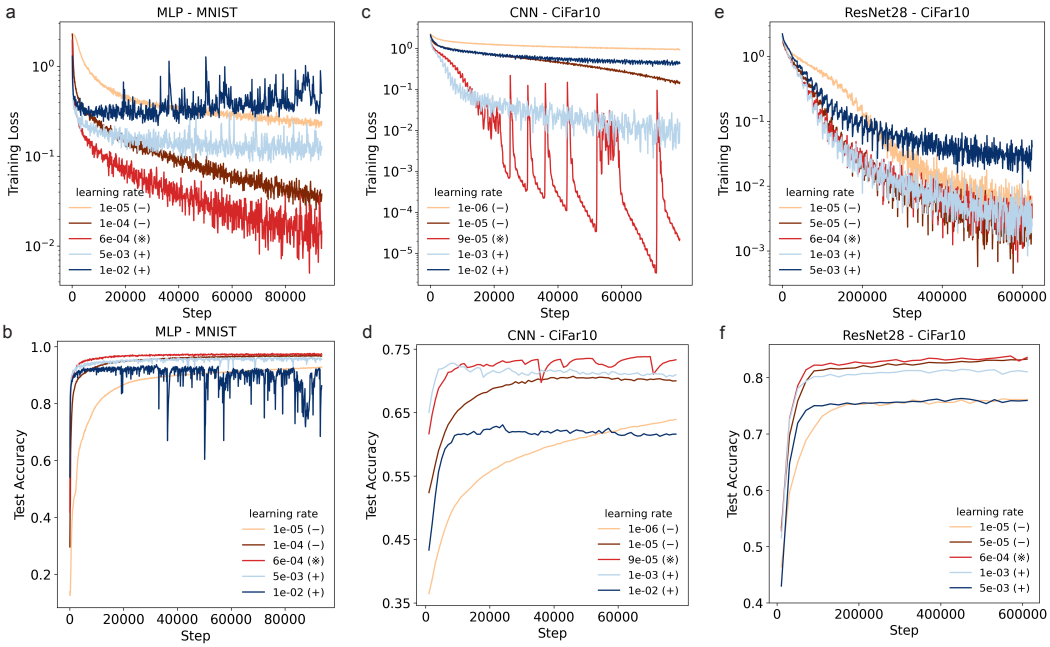
Appendix Figure 2: **Local matched-barrier check.** To assess the matched effective energy-scale approximation used in the local mobility analysis, we perturb the parameters along directions with different local curvatures and measure the resulting loss increase ΔL. Despite their different local curvature, the two directions exhibit nearly identical loss increases over the finite perturbation range considered. At a perturbation radius of $\|\delta\theta\|_2 = 0.1$, we obtain $\Delta L_{\text{low}} = 0.1833$ and $\Delta L_{\text{high}} = 0.1833$, with a max-to-min ratio of 1. This result supports the use of a matched effective energy scale when comparing local mobility within the early-training finite-displacement regime.



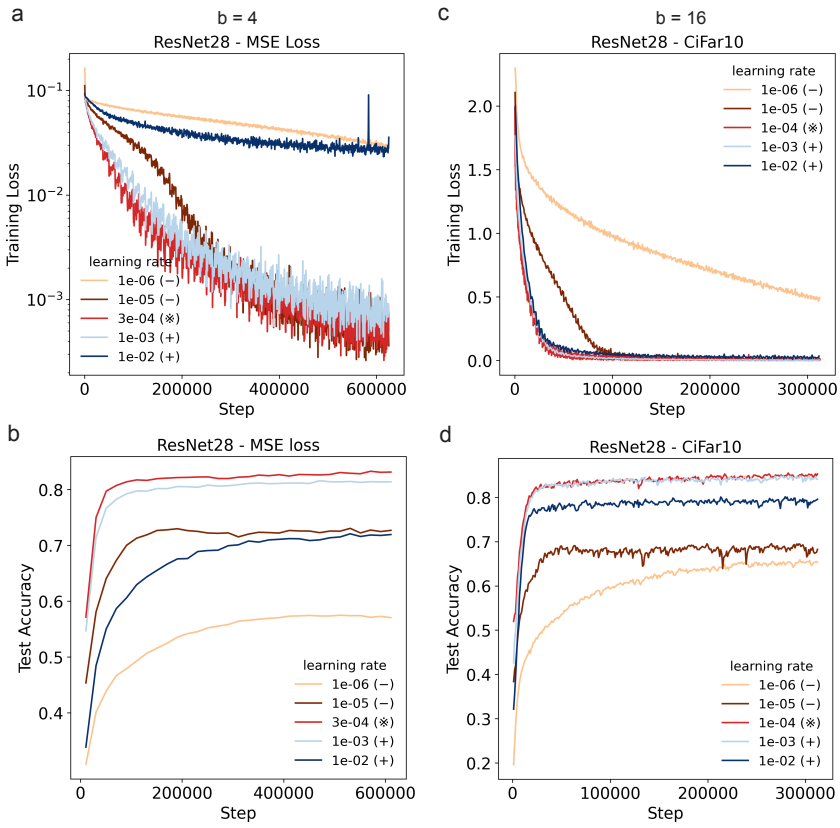
Appendix Figure 3: **The effect of Adam hyperparameters on the maximum mobility contrast.** Sweep of learning rate α for Adam with different β_1 and β_2 values, showing the mobility contrast $R(\alpha) = M_{\text{flat}}/M_{\text{sharp}}$. **Left:** β_1 fixed at 0.9, varying β_2. **Right:** β_2 fixed at 0.999, varying β_1. Error bars denote standard deviation over 100 independent seeds. Vertical dashed lines indicate the predicted characteristic learning rate α_c, where the mobility contrast is expected to be maximal. This result suggests that β_1 modulates the prominence of the peak, while β_2 primarily determines its magnitude.



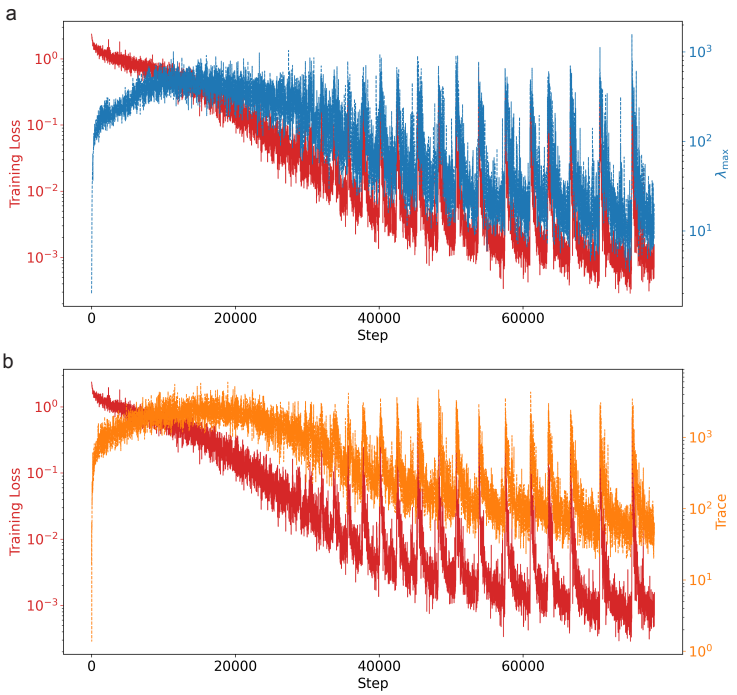
Appendix Figure 4: **The predicted characteristic learning rate across Adam hyperparameters.** Heatmap of the absolute difference between the learning rate that maximizes the mobility contrast ($\alpha_{\text{max mobility}}$) and the predicted characteristic learning rate (α_{pred}) across Adam hyperparameters β_1 and β_2. Each cell shows the mean absolute gap $|\alpha_{\text{max mobility}} - \alpha_{\text{pred}}|$, computed from 100 independent runs. The cell color indicates the magnitude, while the overlaid number shows the corresponding mean absolute value. Lower values indicate better agreement between predicted and observed maximal mobility contrast learning rates, highlighting the robustness of the theoretical estimate over the hyperparameter sweep. The typical default Adam settings correspond to $\beta_1 = 0.9$ and $\beta_2 = 0.999$.



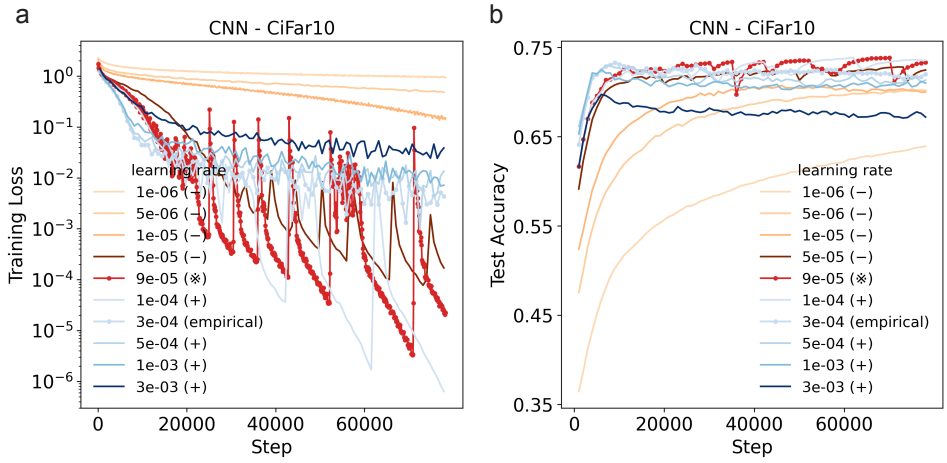
Appendix Figure 5: **Training and test performance under varying learning rates across architectures.** (**a**, **b**) MLP on MNIST (batch size 32), (**c**, **d**) CNN on CIFAR-10 (batch size 64), and (**e**, **f**) ResNet-28 on CIFAR-10 (batch size 4). Top row: training loss; bottom row: test accuracy. In each case, the learning rate marked with ※ corresponds to the theoretically predicted characteristic value α_c. The curves labeled (–) denote values smaller than α_c, while (+) represents values larger than α_c. Across datasets and architectures, α_c lies within a near-optimal performance plateau, whereas learning rates that are at least half an order of magnitude smaller or larger either slow optimization or reduce generalization performance.



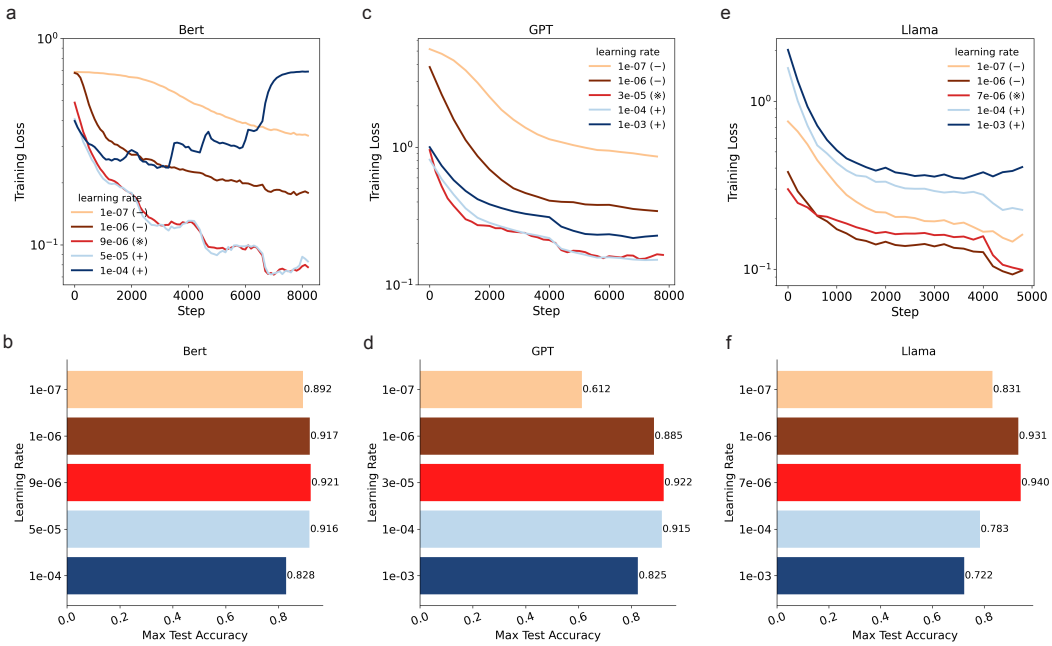
Appendix Figure 6: **Effect of batch size and loss function on the predictive near-optimal learning rate.** Training loss (top) and test accuracy (bottom) for ResNet-28 on CIFAR-10 with different batch sizes. (**a**, **b**) Small batch size ($b = 4$) with Mean Squared Error (MSE) function. (**c**, **d**) Intermediate batch size ($b = 16$) with cross-entropy function. Curves correspond to learning rates below (–), above (+), and equal to (※) the predicted value α_c. In both settings, training with the theoretically predicted learning rate achieves better stability and test accuracy compared to substantially smaller or larger values.



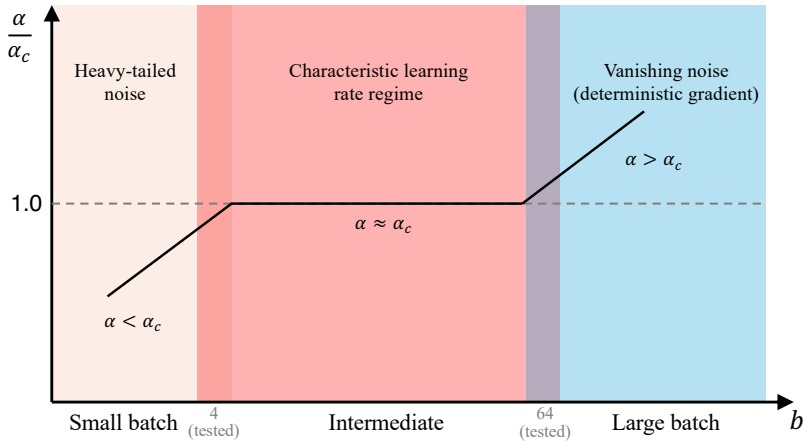
Appendix Figure 7: **Training loss and Hessian sharpness during training.** (a) Training loss (red, left axis) and the largest Hessian eigenvalue $\lambda_{\max}$ (blue, right axis) as functions of training step for a CNN trained on CIFAR-10 with batch size 64. (b) Training loss (red, left axis) and Hessian trace (orange, right axis) as functions of training step for the same run. Loss spikes consistently coincide with increases in both $\lambda_{\max}$ and the Hessian trace, suggesting that these fluctuations are associated with sharper regions of the loss landscape.



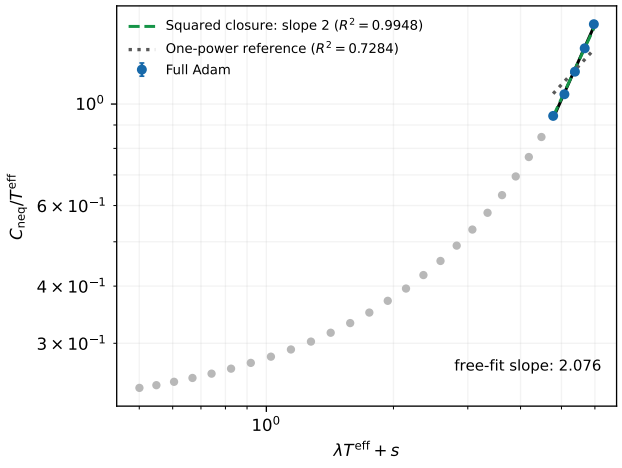
Appendix Figure 8: **Predicted characteristic learning rate versus the empirical grid-search optimum on the image dataset.** Training dynamics and test accuracy for a CNN on CIFAR-10 across learning rates selected via multiplicative grid search (10^{-6}-10^{-3}), along with theoretically predicted and empirical choices (highlighted with markers). The predicted learning rate (red) achieves near-optimal generalization performance comparable to the best grid-searched learning rates, with no statistically significant difference under a two-sided t-test ($t(9) = -0.35,\ p = 7.34 \times 10^{-1}$), while eliminating the need for expensive hyperparameter sweeps.



Appendix Figure 9: **Fine-tuning pretrained large-scale language models on SST-2 under varying learning rates.** (**a**, **b**) BERT, (**c**, **d**) GPT, and (**e**, **f**) Llama-1.1B. Top row: training loss; Bottom row: maximum test accuracy achieved in the training process. In each model, the learning rate marked with (※) denotes the theoretically predicted characteristic value α_c, while (–) and (+) represent values smaller and larger than α_c. All experiments used a batch size of 32. Performance consistently lies in a near-optimal plateau around α_c, whereas smaller learning rates slow convergence and reduce accuracy, and larger ones destabilize training.



Appendix Figure 10: **Batch-size dependence of the near-optimal learning rate.** Schematic relationship between the normalized learning rate α/α_c and batch size b, where α denotes the optimal generalization learning rate. For very small batches, gradient noise can become heavy-tailed and distort loss landscape, so near-optimal performance may favor $\alpha < \alpha_c$ to suppress excessive stochasticity. In the intermediate batch-size regime, mini-batch fluctuations are well described by variance-based noise statistics, and the predicted learning rate maximizes the curvature-dependent mobility contrast. For very large batches, stochastic noise is strongly reduced and the dynamics approaches deterministic gradient descent, so near-optimal performance may favor $\alpha > \alpha_c$ to maintain sufficient parameter-space mobility. Gray markers indicate representative small and large batch sizes evaluated in our experiments. Similar behavior was observed across the tested vision and language tasks, possibly because the structured correlations present in both natural image and natural language data produce comparable effective gradient-noise regimes during mini-batch training.



Appendix Figure 11: **Direct exponent test of the squared sharp-region closure.** Full Adam dynamics provides the stationary $\langle \hat{v} \rangle$, from which we construct $T^{\mathrm{eff}} = C_{\mathrm{neq}}/\langle \hat{v} \rangle$. Gray points are outside the prespecified curvature-dominated window and are not fitted. For the blue points, the free log–log slope of $C_{\mathrm{neq}}/T^{\mathrm{eff}}$ versus $\lambda T^{\mathrm{eff}} + s$ is 2.076 (95% CI $[1.838, 2.315]$). A fixed slope of 2, predicted by the squared closure, gives $R^2 = 0.9948$, compared with $R^2 = 0.7284$ for the one-power reference.

REFERENCES

Jeff Z HaoChen, Colin Wei, Jason Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. *Conference on Learning Theory*, pp. 2315–2357, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Umut Simsekli, Levent Sagun, and Mert Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. *International Conference on Machine Learning*, pp. 5827–5837, 2019.

Ning Yang, Chao Tang, and Yuhai Tu. Stochastic gradient descent introduces an effective landscape-dependent regularization favoring flat solutions. *Physical Review Letters*, 130(23):237101, 2023.

Wei Zhang, Mingrui Liu, Yu Feng, Xiaodong Cui, Brian Kingsbury, and Yuhai Tu. Loss landscape dependent self-adjusting learning rates in decentralized stochastic gradient descent. *arXiv preprint arXiv:2112.01433*, 2021.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. *International Conference on Machine Learning*, 2019.