How Does Local Landscape Geometry Evolve in Language Model Pre-Training?

Anonymous authors
Paper under double-blind review

Abstract

The scale and expense of pre-training large language models make efficient hyperparameter tuning essential, yet principled guidance remains limited. To address this gap, we analyze language model pre-training dynamics from a local landscape geometry perspective. Our study reveals two distinct phases. In the *early* phase, sharpness of the local landscape is initially high, leading to instability and loss plateaus under large learning rates (LRs). As training progresses, the landscape shifts from sharp to flatter regions. This dynamic explains the necessity of LR warmup and further suggests that larger peak LRs require proportionally longer warmup periods. In the *late* phase, the local landscape is governed by the gradient noise scale: high noise from smaller batches widens the loss basin, whereas reduced noise from larger batches deepens it. This insight inspires a dynamic batch-size (BS) schedule that increases the BS when the loss plateaus, achieving lower terminal loss with significant fewer tokens than constant-BS training. Together with our *theory*, we provide a unified account of loss landscape evolution, which translates into actionable tuning strategies for large-scale pre-training.

1 Introduction

Training large language models efficiently requires carefully tuned hyperparameters, yet principled guidance for tuning remains limited. While practitioners often rely on grid search or trial-and-error, these approaches are costly and unreliable at scale. Recent research (Foret et al., 2021; Cohen et al., 2021; Gilmer et al., 2022) highlighted that the geometry of the local loss landscape offers fundamental insights into optimization, revealing how factors such as sharpness[1] interact with hyperparameters to shape training dynamics. Consequently, leveraging insights from the local landscape presents a promising path toward principled hyperparameter tuning for language model pre-training.

Several pioneering works have already attempted to study language models from the local landscape perspective. Wen et al. (2024) introduced the "river-valley" landscape to explain the effectiveness of Warmup-Stable-Decay (WSD) schedules (Hu et al., 2024). Zhang et al. (2024); Wang et al. (2025) identified blockwise sharpness patterns in language models through Hessian-based analyses. Peng et al. (2024); Chen et al. (2025) further visualized the loss landscapes of finetuned language models, offering geometric insights into the safety alignment. However, few studies have investigated the *dynamics* of local landscape geometry during language model pre-training.

To this end, we pose the central research questions of this paper:

1. *How does the local landscape geometry evolve in language model pre-training?*
2. *What implications does this evolution have for principled hyperparameter tuning?*

Our contributions. In this work, we present the first systematic study of the evolution of local landscape geometry during language model pre-training. As illustrated in Figure 1, our analysis reveals two distinct phases, each with significant implications for hyperparameter tuning.

• *Early in Training: From Sharp to Flat Landscapes.* In the early phase, we observe that the model shifts from sharper regions of the loss landscape toward flatter ones, contrary to the progressive sharpening phenomenon in prior works (Cohen et al., 2021; Song & Yun, 2023; Cohen et al., 2025). Linear stability analysis in Section 4 shows that the maximum stable learning rate (LR) η is inversely

[1] In this work, we define sharpness as the largest eigenvalue of the Hessian matrix at the current iterate.

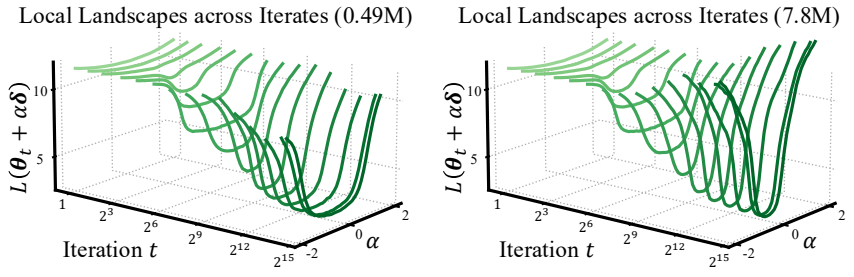

Figure 1: **The evolution of local loss landscape throughout pre-training.** We train `LLaMA-2` models with 170M parameters using different BSs (0.49M and 7.8M), and visualize the one-dimensional loss landscape at iterate $\boldsymbol{\theta}_t$ along a random direction $\boldsymbol{\delta}$, i.e., plot $L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ vs. the perturbation coefficient α. The landscapes are shown across different training iterations t. **Early phase.** The landscapes gradually widens/flattens for both training runs. **Late phase.** Training with smaller BS produces wider landscapes than training with larger BS.

proportional to sharpness. Since sharpness is extremely high early in pre-training, using large peak LRs without sufficient warmup leads to instabilities, such as loss spikes and plateaus (see Figure 2).

Implications. The sharp-to-flat transition explains the necessity of LR warmup: LR should remain small until sharpness has sufficiently decayed, preventing training instabilities. This further provides a practical tuning recipe: within a reasonable range, larger peak LRs require proportionally longer warmup, to safely navigate the sharpest stage of training.

• *Late in Training: Basin Selection Governed by Noise Scale.* In the late phase, the local landscape geometry is largely governed by the noise scale during training, with batch size (BS) B serving as its primary controller. Our analysis shows that smaller BS widens the loss basin, while larger BS deepens it. To explain this, we analyze the diffusion limit of preconditioned SGD, which uncovers a depth–flatness trade-off: reduced gradient noise tends to minimize the loss, leading to deeper minima; whereas increased noise tends to regularize the sharpness of landscape, moving toward wider ones.

Implications. The trade-off motivates a principled BS scheduling strategy: increases the BS once the loss reduction progress slows. Consider that the trade-off only emerges in the late phase, we start with a small BS to improve token efficiency. Our scheduling ensures steady loss reduction with minimal token consumption, ultimately achieving lower terminal loss than constant-BS training. Moreover, since the noise scale is proportional to η/B in our theory, we predict that BS ramping and LR decay reduce the noise scale in similar ways and thus yield comparable performance (see Figure 8).

In summary, our work provides a two-phase picture of landscape evolution in pre-training: an early sharp-to-flat transition that necessitates LR warmup, and a late noise-driven regime that motivates BS scheduling. This unified view advances our understanding of pre-training dynamics and underscores the importance of landscape geometry in offering principled guidance for hyperparameter tuning.

2 Related Works

Local Landscape Geometry (Sharpness) Evolution. Understanding how local landscape geometry, particularly sharpness, evolves during training has drawn increasing attention. Wu et al. (2018); Cohen et al. (2022); Song & Yun (2023); Cohen et al. (2025) showed that initially gradient descent (GD) tends to move from flatter to sharper regions of the landscape. In addition, Jastrzębski et al. (2019); Jastrzebski et al. (2020) argued that in SGD, sharpness also changes monotonically but either increase or decrease depending on the setting. In the later phase, however, sharpness is largely governed by the properties of the optimizer (Zhou et al., 2025). One notable example is that the stochastic noise introduced by SGD and its variants implicitly biases training toward flat minima (Wu et al., 2018; Zhu et al., 2019; Xie et al., 2021; Wu et al., 2022). Yet, these findings are largely restricted to small-scale networks; *In comparison*, our work present the *first systematic study* of how local landscape geometry evolves in large-scale language model pre-training, offering new insights into LR warmup and the design of BS schedules.

Large-Scale Pre-training: Learning Rate Warmup. Learning rate warmup, first introduced in large-batch ResNet (He et al., 2016; Goyal et al., 2017) and Transformer training (Vaswani et al., 2017), is now standard in large-scale pre-training (Shoeybi et al., 2019; Zhang et al., 2022; Hu et al.,

2024). Its mechanism, however, remains only partly understood. Gotmare et al. (2019) showed that warmup prevents excessively large early parameter updates; Gilmer et al. (2022) argued that warmup guides optimization into flatter regions where large LRs are stable; and Kosson et al. (2024) showed in language model pre-training that warmup mitigates momentum bias correction and correlated gradients that otherwise drive unstable representation shifts. Yet no unified explanation exists. *In comparison*, our work views warmup from a *unified* geometric perspective, suggesting that larger peak LRs demand proportionally longer warmup.

Large-Scale Pre-training: Batch Size Schedules. Batch size is another critical hyperparameter in large-scale pre-training, shaping the trade-off between step efficiency and data efficiency. Most prior work (McCandlish et al., 2018; Kaplan et al., 2020; Gray et al., 2023; 2024; Zhang et al., 2025) has focused on the critical batch size (CBS), the point where further increasing BS yields diminishing returns. However, CBS is typically treated as a constant, and much less attention has been given to *BS scheduling*. Early works on adaptive sampling proposed gradually increasing BS to balance efficiency and noise reduction (De et al., 2017; Lau et al., 2024b;a; 2025; Ostroukhov et al., 2024). However, these studies remain mostly theoretical. Advanced language models (Brown et al., 2020; Touvron et al., 2023; Liu et al., 2024; Li et al., 2025) employed stage-wise BS schedules, but without systematic analysis. *In contrast*, our work connects BS scheduling to the evolving local landscape geometry, providing a principled foundation for when and how to expand BS during pre-training.

3 Preliminaries

Basic Notations. We use bold lowercase letters (e.g., $\boldsymbol{x} = (x_i)$) to denote vectors and bold uppercase letters (e.g., $\mathbf{A} = (a_{ij})$) to denote matrices. For a matrix $\mathbf{A}$, let $\|\mathbf{A}\|_2$, $\|\mathbf{A}\|_F$, and $\mathrm{Tr}(\mathbf{A})$ denote its spectral norm, Frobenius norm and trace, respectively. The Hadamard product are denoted by $\odot$.

Theoretical Setup. Our main discoveries are grounded in a simple theoretical model. In particular, we study the local dynamics of preconditioned stochastic gradient descent (SGD) near an local minimum of the empirical risk. We consider a model with parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ and a training set of n examples. Let $L_i(\boldsymbol{\theta})$ be the fitting error evaluated at i-the example and $L(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^n L_i(\boldsymbol{\theta})$ be the empirical risk. Suppose $L(\boldsymbol{\theta})$ admits a strict local minimizer $\boldsymbol{\theta}^\star$, and denote the Hessian at $\boldsymbol{\theta}^\star$ by $\mathbf{H}(\boldsymbol{\theta}^\star) := \nabla^2 L(\boldsymbol{\theta}^\star) \succ 0$. In a neighborhood of $\boldsymbol{\theta}^\star$, the loss can be approximated quadratically:

$$L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}^\star) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^\star)^\top \mathbf{H}(\boldsymbol{\theta}^\star)(\boldsymbol{\theta} - \boldsymbol{\theta}^\star) \tag{1}$$

We analyze the preconditioned SGD with a fixed preconditioner $\mathbf{M} \succ 0$[2]. Let $\boldsymbol{e}_k := \boldsymbol{\theta}_k - \boldsymbol{\theta}^\star$. At iteration k, the update rule[3] gives:

$$\boldsymbol{e}_{k+1} = (\mathbf{I} - \eta \mathbf{M}\mathbf{H}(\boldsymbol{\theta}^\star))\boldsymbol{e}_k + \eta \mathbf{M}\boldsymbol{\xi}_k, \tag{2}$$

where $\eta > 0$ is the LR and $\{\boldsymbol{\xi}_k\}$ are i.i.d. random noise vectors with

$$\mathbb{E}[\boldsymbol{\xi}_k] = \mathbf{0}, \quad \mathbb{E}[\boldsymbol{\xi}_k \boldsymbol{\xi}_k^\top] = \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)/B. \tag{3}$$

Note that $\boldsymbol{\Sigma}(\boldsymbol{\theta}^\star) = \frac{1}{n}\sum_{i=1}^n \nabla L_i(\boldsymbol{\theta}^\star)\nabla L_i(\boldsymbol{\theta}^\star)^\top - \nabla L(\boldsymbol{\theta}^\star)\nabla L(\boldsymbol{\theta}^\star)^\top$ is the gradient covariance at $\boldsymbol{\theta}$, and B denotes the BS. Despite its simplicity, this model captures key aspects of pre-training dynamics, especially in the late phase (see Section 5). Similar formulations have been widely used in dynamical stability analyses (Wu et al., 2018; Cohen et al., 2021; Zhou et al., 2025) and theoretical advances on BS scaling (McCandlish et al., 2018; Zhang et al., 2019).

Experimental Setup. We train `LLaMA-2` architecture (Touvron et al., 2023) models with 93M and 170M parameters. Training is performed on the `FineWeb-Edu` dataset (Penedo et al., 2024), with sufficient training budgets ranging from 50 to 1000 tokens-per-parameter (TPP)[4] and a context length of 1024. We adopt AdamW (Kingma & Ba, 2014) with hyperparameters $\beta_1 = 0.95$, $\beta_2 = 0.95$, and weight decay 0.1, together with gradient clipping at 1.0 for stability. Evaluation is conducted on a held-out validation split of $\sim$50M tokens to monitor training stability and convergence.

Our experiments varies the LRs and BSs. In Section 4, we primarily study the role of LR and warmup length, fixing BS at 7.8M. In Section 5, we focus on the effect of BS, with LR fixed at 2^{-10}. To

[2]Most practical preconditioners are positive-definite: $\mathbf{M} = I$ for SGD, diagonal $\mathbf{M}$ for AdaGrad (Duchi et al., 2011), RMSProp (Tieleman & Hinton, 2012), Adam, etc.

[3]Derived from the Equation (1) and $\boldsymbol{\theta}_k = \boldsymbol{\theta}_k - \eta\mathbf{M}(\nabla L(\boldsymbol{\theta}_k) + \boldsymbol{\xi}_k)$.

[4]At least $10\times$ over Chinchilla-optimal tokens (Hoffmann et al., 2022).

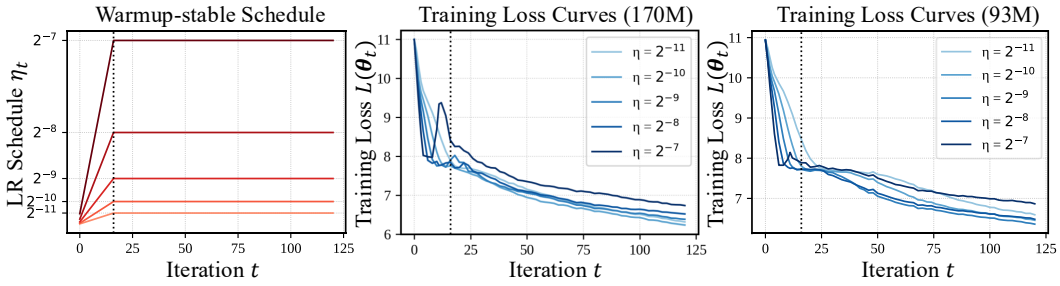

Figure 2: **Loss spikes and plateau early in training.** We train a series of `LLaMA-2` models with 93M and 170M parameters. We adopt a warmup-stable schedule, where the warmup length is shortened to 16 iterations and the peak LR is varied, $\eta \in \{2^{-11}, 2^{-10}, 2^{-9}, 2^{-8}, 2^{-7}\}$. **(Left).** LR schedule: η_t vs. training iteration t. **(Middle, Right).** Training loss curves for different model sizes: $L(\boldsymbol{\theta}_t)$ vs. training iteration t. The vertical dashed line marks the end of the warmup phase.

decouple BS ramping from LR decay, we adopt a *warmup-stable* schedule: after linear warmup to the peak value, the LR remains constant (similar to WSD Hu et al. (2024), but without decay phase).

4 Early in Pre-Training: From Sharp to Flat Landscapes

In this section, we provide evidence that, during the early phase of pre-training, the local landscape of language models evolves from sharp regions toward flatter ones. We first observe that training with large LRs and insufficient warmup often leads to instability and early loss plateaus. By linear stability analysis, we then attribute these behaviors to sharp-to-flat dynamics occurring in the initial phase of training. This finding explains why pre-training needs LR warmup and suggests that larger peak LRs require proportionally longer warmup periods.

Motivating Observations: Instability and Loss Plateaus Early in Training. The loss curves for pre-training are typically smooth initially; the model escapes from random initialization and the loss decreases rapidly. Yet, surprisingly, when the warmup length is extremely shortened, we *consistently* observe loss spikes and plateaus near the end of the warmup phase.

To demonstrate this, we train models of different sizes with a fixed warmup length of 16 iterations while varying the peak LR. As shown in Figure 2, a loss plateau reliably appears around the end of the warmup phase across all settings. Additionally, larger LRs produce higher spikes, which mark a characteristic feature of early training instability. Given these results, two natural questions arises:

Q1. *Why does shortened warmup induce training instability?*

Q2. *Why do spikes and plateaus occur only at the very beginning of training?*

To shed light on these questions, we begin with a simple quadratic model that enables us to analyze stability through the lens of linear dynamics.

Linear Stability Analysis: Sharpness Matters. Consider the quadratic model introduced in Section 3. When the noise term $\boldsymbol{\xi}$ in Equation (2) is set to zero, the dynamics reduce to:

$$\boldsymbol{e}_{k+1} = (\mathbf{I} - \eta \mathbf{M}\mathbf{H}(\boldsymbol{\theta}^\star))\boldsymbol{e}_k, \tag{4}$$

a linear system whose stability depends on the spectral properties of $\mathbf{I} - \eta\mathbf{M}\mathbf{H}(\boldsymbol{\theta}^\star)$. Intuitively, the LR η interacts directly with the curvature of the landscape: if η is too large relative to the sharpest direction, the updates will diverge. The following lemma formalizes this stability condition for preconditioned GD.

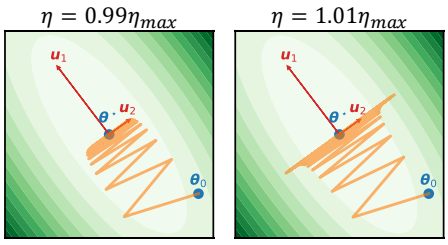

Figure 3: **Gradient descent on a simple quadratic function:** $L(\boldsymbol{\theta}) = \boldsymbol{\theta}\mathbf{H}\boldsymbol{\theta}^\top$. Here, $\boldsymbol{\theta} \in \mathbb{R}^2, \mathbf{H} \in \mathbb{R}^{2\times 2} \succ 0$ with $\lambda_1 = 8$ and $\lambda_2 = 1$. Thus, $\eta_{max} = 2/\lambda_1 = 0.25$. Training with $\eta \approx \eta_{max}$ causes slow convergence, and when $\eta > \eta_{max}$, it diverges.

Lemma 4.1 (Stability Condition for Preconditioned GD)**.** *Define the preconditioned curvature matrix* $\mathbf{S} := \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}$*, and let* $\{\lambda\}_{i=1}^p$ *be the eigenvalues of* $\mathbf{S}$*. The linear system in Equation (4) is asymptotically stable (i.e.,* $\lim_{k\to\infty} \boldsymbol{e}_k = \mathbf{0}$ *for any initial* $\boldsymbol{e}_0$*) if* η *satisfies* $0 < \eta < \frac{2}{\lambda_{\max}(\mathbf{S})}$.

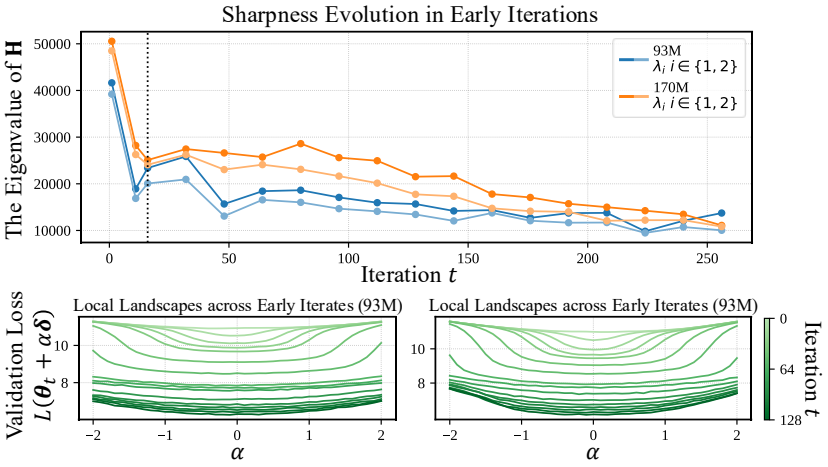

Figure 4: **Early pre-training shifts iterates from sharp to flat regions.** We visualize the local landscape geometry evolution of training runs in Figure 2. For each model size, we select the training run with LR 2^{-10}. **(Top).** Evolution of the top eigenvalues of the Hessian across iterations: $\lambda_i(\mathbf{H}(\boldsymbol{\theta}_t))$ vs. iteration t. **(Bottom).** One-dimensional loss landscape along a random perturbation direction: the perturbed loss $L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ vs. perturbation coefficient α, shown across early training iterations t.

Lemma 4.1 shows that stability of preconditioned GD is governed by the largest eigenvalue of $\mathbf{S}$. If the curvature along the sharpest direction is too large, only a sufficient small LR can prevent divergence. This generalizes the stability analysis in Wu et al. (2018) to the preconditioned setting.

We next characterize the one-step loss change as η approaches the stability boundary $2/\lambda_{\max}(\mathbf{S})$.

Lemma 4.2 (Exact One-step Loss Change). *Let* $\mathbf{S} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ *with* $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_p)$ *and define* $\boldsymbol{z} = \mathbf{Q}^\top \mathbf{M}^{-1/2} \boldsymbol{e}_k$, *then*

$$\frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = \sum_{j=1}^{p} w_j (1 - \eta\lambda_j)^2, \quad w_j := \frac{\lambda_j z_j^2}{\sum_{\ell=1}^{p} \lambda_\ell z_\ell^2} \in [0, 1], \quad \sum_j w_j = 1.$$

In particular, $\min_{\boldsymbol{z}} \frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = (1 - \eta\lambda_{\max})^2$, *so that* $\boxed{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}_{k+1}) \xrightarrow{\eta \uparrow 2/\lambda_{\max}} 0}$.

Lemma 4.2 states that when η is close to $2/\lambda_{\max}(\mathbf{S})$, each update yields only a marginal decrease in loss. Together with Lemma 4.1, it is clear that training near the stability boundary naturally leads to characteristic loss spikes and plateaus (see Figure 3 for an illustration). Importantly, the stability boundary is determined by the sharpness of the loss landscape. To further address **Q1-2**, we analyze how sharpness evolves during the early phase of pre-training.

The Early Dynamics: From Sharp to Flat Landscapes. We study how the local landscape geometry, particularly the sharpness, evolves for training runs in Figure 2. Specifically, we track the evolution of the top eigenvalues of the Hessian[5] $\mathbf{H}(\boldsymbol{\theta}_t)$ during early pre-training. For the early checkpoints $\boldsymbol{\theta}_t$, we also visualize the one-dimensional loss landscape along a random direction by plotting the function $\mathcal{L}(\alpha) := L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ with $\boldsymbol{\delta} \sim \mathcal{N}(0, \mathbf{I})$. Li et al. (2018) showed that such random-direction visualizations reliably capture intrinsic properties of the loss landscape properties, such as sharpness. To ensure fair comparison across iterations, we fix the same random vector $\boldsymbol{\delta}$ for all $\boldsymbol{\theta}_t$.

In Figure 4 (top), the largest eigenvalues of the Hessian $\mathbf{H}(\boldsymbol{\theta}_t)$ start at high values[6] and then decrease sharply, indicating a substantial reduction in curvature along the sharpest direction. Furthermore, in Figure 4 (bottom), the loss landscape along a random direction progressively widens as training proceeds, confirming that the model shifts from sharp to flat regions even in the most-case directions.

[5] Following Cohen et al. (2021), we use the `Lanczos` algorithm to calculate top eigenvalues of Hessian.

[6] In fact, at initialization, sharpness is extremely low but rises sharply after the first update. The sharpness curves reported in Figure 4 therefore start from the first iteration.

A Tuning Recipe: Larger Peak LR, Longer Warmup. We have seen that training stability depends on sharpness: when the landscape is steep, only a sufficiently small LR can keep updates stable; and pre-training initially traverses from sharp landscapes to flatter ones. Now let us return to **Q1** and **Q2**:

A1. *If the warmup phase is shortened, the LR rises too quickly while the model is still in sharp regions, leading to loss spikes and plateaus.*

A2. *As training progresses, the landscape becomes flatter and the same LR no longer threatens stability, which explains why instability is confined to the very beginning.*

Therefore, in practice, we need a sufficiently long warmup phase to keep the LR small until sharpness has decayed, thereby preventing loss spikes and plateaus. This rational further suggests a practical tuning recipe: *the larger the peak LR, the longer the warmup should be*, ensuring iterates safely transition into flatter landscapes before reaching full step size.

To validate this, we train models with varied peak LRs η and warmup lengths T_w (in iterations). In Figure 5, larger peak LRs require *proportionally* longer warmup to achieve the optimal validation loss $L(\boldsymbol{\theta}_{\text{best}})$. However, this proportionality does not hold universally. When $\eta = 2^{-7}$, the optimal warmup length remains 2^{10} iterations, the same as for $\eta = 2^{-8}$. Thus, the relationship applies within a reasonable range, when both the peak LR and warmup length are neither too small nor too large.

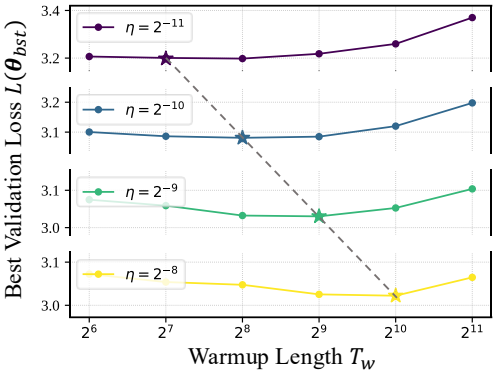

Figure 5: **Larger Peak LR, Longer Warmup.** We train a series of `LLaMa-2` models with 170M parameters and 100 TPP. We vary the peak LRs η and warmup lengths T_w. We plot the best validation loss $L(\boldsymbol{\theta}_{\text{bst}})$ vs. T_w for different η. For each η, the optimal T_w is highlighted with a star.

Comparison with Gilmer et al. (2022); Kalra & Barkeshli (2024). These works also studied warmup from a sharpness perspective but focused mainly on standard image classification tasks (e.g., ResNet on CIFAR-10). In contrast, we study language model pre-training, where the regime is fundamentally different: for instance, in Figure 6, a 16-iteration warmup accounts for only $\sim 0.078\%$ of the whole training process while the loss remains high, whereas in small-scale settings the loss has already decreased significantly after warmup. Furthermore, we introduce the warmup tuning recipe for large-scale pre-training, which to our knowledge has not yet been carefully explored before.

5 Late in Pre-Training: Local Landscape Governed by Noise Scale

In this section, we turn to the evolution of the local landscapes in the late phase of pre-training. We observe that BS plays a central role: training with a large BS tends to find a *deeper* basin of the landscape, whereas training with a small BS favors a *wider* basin. By analyzing the diffusion limit of Equation (2), we prove that this trade-off between *wide* or *deepen* is governed by the training *noise scale*. Building on this insight, we further propose a dynamic BS scheduler for the data-limited regime, which achieves lower terminal loss while consuming fewer tokens.

The Effect of BS: Local Landscapes Late in Training. We conduct experiments to systematically investigate the role of BS in shaping the local landscape geometry during the late phase of pre-training. Specifically, we train models with different BSs for $T = 20{,}480$ iterations. Figure 6 (top left) shows the validation loss curves for each run. Evidently, larger BS consistently leads to lower terminal loss and faster convergence in term of iterations[7]. We then visualize the loss landscape around the final iterate $\boldsymbol{\theta}_T$. In Figure 6 (top right), it is clear that small BS produces flatter basins, whereas large BS yields deeper ones. To further demonstrate, Figure 6 (bottom) compares the landscape evolution of runs with $B = 0.49$M and $B = 7.8$M, indicating that in the late training phase, larger BS tends to deepen the basin, while smaller BS shifts toward wider basins.

Despite these convincing results, two key question remains:

[7] In terms of processed tokens, small BS training converges faster.

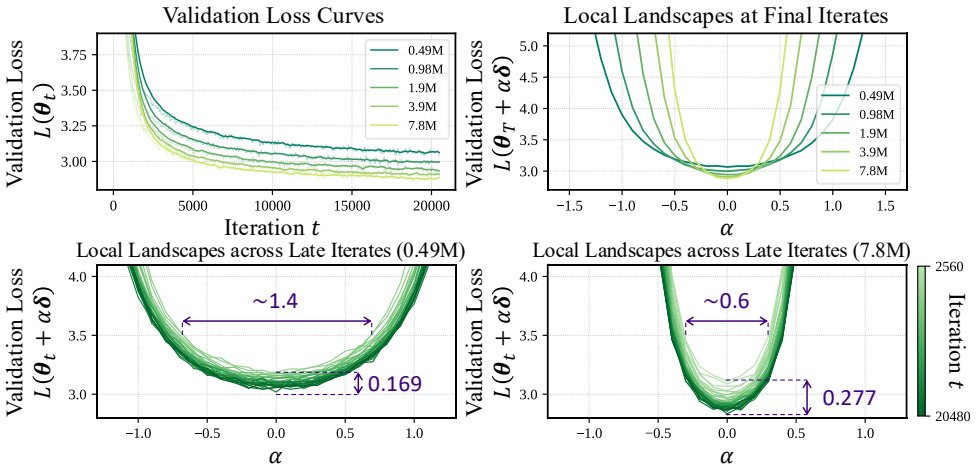

Figure 6: **Large BS deepens the basin, small BS widens the basin.** We train a series of `LLaMA-2` models (170M) for $T = 20{,}480$ iterations, using BSs $B \in \{0.49\text{M}, 0.98\text{M}, 1.9\text{M}, 3.9\text{M}, 7.8\text{M}\}$. **(Top left).** Validation loss curves for different BSs: $L(\boldsymbol{\theta}_t)$ vs. training iteration t. **(Top right).** One-dimensional loss landscapes at the final iterates $\boldsymbol{\theta}_T$ along a random perturbation direction: perturbed loss $L(\boldsymbol{\theta}_T + \alpha\boldsymbol{\delta})$ vs. perturbation coefficient α, visualized across different BSs. **(Bottom).** One-dimensional loss landscape: the perturbed loss $L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ vs. perturbation coefficient α, shown across late training iterations t for $B = 0.49\text{M}$ and $B = 7.8\text{M}$.

Q3. *Why is there a trade-off between widening and deepening the basin?*

Q4. *Which factor underlying the hyperparameter BS governs this trade-off?*

To delve deeper into these questions, we revisit the simple quadratic model[8] introduced in Section 3.

Widen or Deepen: Noise Scale Governs Basin Selection. Recall the update rule of preconditioned SGD:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\mathbf{M}(\nabla L(\boldsymbol{\theta}_k) + \boldsymbol{\xi}_k), \quad \mathbb{E}[\boldsymbol{\xi}_k] = \mathbf{0}, \quad \mathbb{E}[\boldsymbol{\xi}_k\boldsymbol{\xi}_k^\top] = \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)/B. \tag{5}$$

For simplicity, we take the continuous-time limit of Equation (5), where dynamics are approximated by a stochastic differential equation (SDE). The following proposition formalizes this approximation.

Proposition 5.1 (Convergence to SDE)**.** *Consider the scaled discrete process $\boldsymbol{\theta}_{\lfloor t/\eta \rfloor}$ as $\eta \to 0$. Suppose the noise covariance satisfies*[9]

$$\frac{\eta}{B}\mathbf{M}\boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^\top = 2\tau\mathbf{M} + \mathcal{O}(\eta), \tag{6}$$

for some temperature $\tau > 0$. Then the process converges weakly to the Itô SDE:

$$d\boldsymbol{\theta}_t = -\mathbf{M}\nabla L(\boldsymbol{\theta}_t)dt + \sqrt{2\tau}\mathbf{M}^{1/2}dW_t \tag{7}$$

where W_t is standard Brownian motion.

Proposition 5.1 shows that preconditioned SGD converges to a noisy gradient flow in the small–η limit. The temperature τ quantifies the noise scale and is proportional to η/B.

Subsequently, we establish the trade-off between deepening and widening the loss basin using both the quadratic model (see Equation (1)) and the SDE limit of preconditioned SGD.

Theorem 5.1 (Depth-Flatness Trade-off)**.** *Let the empirical risk $L(\boldsymbol{\theta})$ admit multiple local minima $\{\boldsymbol{\theta}_i^\star\}_{i=1}^m$ with Hessians $\mathbf{H}(\boldsymbol{\theta}_i^\star) \succ 0$. Under the SDE in Equation (7) with temperature τ, the stationary probability that training resides in basin i is given by:*

$$P_\tau(\textit{basin } i) = \frac{\exp(-F_i(\tau)/\tau)}{\sum_j \exp(-F_j(\tau)/\tau)}, \quad F_i(\tau) := L(\boldsymbol{\theta}_i^\star) + \frac{\tau}{2}\log\det\mathbf{H}(\boldsymbol{\theta}_i^\star).$$

[8] We analyze the late phase where $\boldsymbol{\theta}_t$ sits near a local minimum and the quadratic model is accurate.

[9] The assumption in Equation (6) is justified in Appendix B.2.

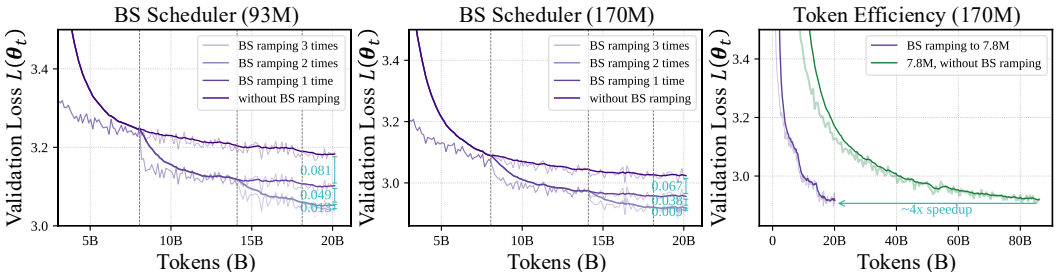

Figure 7: **BS scheduling improves data efficiency.** We train `LLaMA-2` models with 93M and 170M parameters, using a BS schedule that starts at 0.49M and increases by $4\times$ at each ramp. Models are trained with $1, 2$, or 3 ramping steps, while models without ramping serve as the baseline. Vertical gray dashed lines indicate ramping positions. **(Left, Middle).** The validation curves for each run. **(Right).** Comparison between training with BS ramping to 7.8M and training with a fixed 7.8M BS.

Theorem 5.1 states that the basin selection is controlled by the free energy function $F(\tau) = L(\boldsymbol{\theta}^\star) + \frac{\tau}{2}\log\det \mathbf{H}(\boldsymbol{\theta}^\star)$. In early training, the loss term $L(\boldsymbol{\theta}^\star)$ dominates, so the model primarily seeks regions of lower loss. In later training, $L(\boldsymbol{\theta}^\star)$ is comparable to the flatness penalty $\log\det \mathbf{H}(\boldsymbol{\theta}^\star)$, and basin selection becomes increasingly sensitive to the noise scale τ. Together with Proposition 5.1, fixing the LR η yields the following trade-off:

Deepen. Large BS $B \Rightarrow$ low $\tau \Rightarrow$ selection is L-dominated; training tends to *deepen* the basin.

Widen. Small BS $B \Rightarrow$ high $\tau \Rightarrow$ selection is $\mathbf{H}$-dominated; training tends to *widen* the basin.

Efficient Pre-Training: A BS Scheduler in Data-Limited Regime. Turning back to **Q3** and **Q4**, the trade-off arises because basin selection balances loss minimization against curvature regularization (**A3**), with the governing factor being the noise scale τ (**A4**). Since the primary objective of pre-training is to minimize the training loss, this balance naturally favors using the largest BS available. In practice, however, data availability is limited, and excessively large BS substantially increase data consumption[10]. Thus, scheduling BS in pre-training is crucial, particularly in the data-limited regime.

Inspired by our theory, we propose the design principle for a BS scheduler in the data-limited regime:

1. **Start at a small BS.** *Loss reduction dominates early in training; large BS provides little benefit.*
2. **Ramp the BS once loss reduction becomes marginal.** *The flatness penalty is suppressed and training converges to deeper minima.*

To validate this, we train models of different sizes, using a BS schedule that starts at 0.49M and ramps[11] by $4\times$ once loss minimization slows. In Figure 7 (left, middle), models with $1, 2$ or 3 BS ramping steps consistently achieve lower validation loss than those trained without BS ramping. For example, in the 170M case, the 3-step BS ramping improves the final validation loss by about 0.114. Moreover, Figure 7 (right) highlights the data-efficiency of the BS scheduling: ramping the BS up to 7.8M achieves nearly the same final validation loss as training with a fixed 7.8M BS, but requires only about $\frac{1}{4}$ of the tokens (i.e., a $\sim 4\times$ speedup). These results confirm that BS scheduling preserves the benefits of large BS while substantially reducing data consumption.

Comparison with McCandlish et al. (2018); Li et al. (2025). McCandlish et al. (2018) linked BS scaling to the gradient noise and introduced the notion of critical BS. We extend this work by showing that the noise scale governs the depth–flatness trade-off in basin selection, and by proposing a BS scheduling strategy that improves data efficiency. Li et al. (2025) also explored the BS scheduling, increasing BS once the loss dropped below a certain threshold. While similar in spirit, their approach is heuristic, whereas our work uncovers the theoretical mechanism underlying BS scheduling.

6 More Discussions: LR decay and The Blockwise Structure

Comparing BS Ramping with LR Decay. So far, we have excluded LR decay in our experiments to isolate the effect of BS ramping. Yet, recall from Proposition 5.1 that the noise scale τ is proportional

[10]For example, in Figure 6 (top right), when $B = 7.8$M, the run consumes approximately 160B tokens.

[11]The ramping is implemented as step functions, not linear schedule, so each ramp is sharp change.

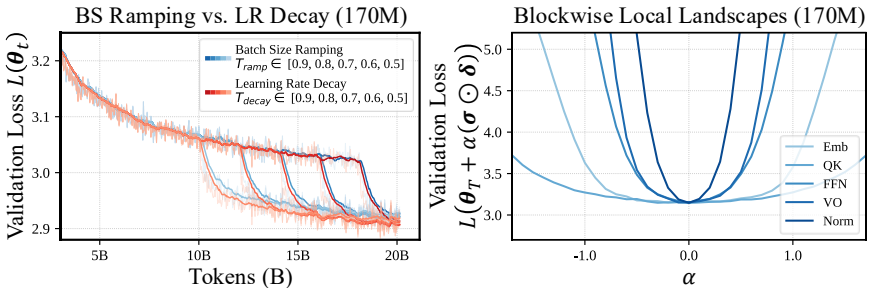

Figure 8: **(Left) BS ramping performs similarly to LR decay.** Validation loss curves for training with either BS ramping or LR decay. In all runs, BS starts at 0.49M and LR begins at 2^{-10} (after linear warmup). For BS ramping, BS increases to $8\times$ its initial value; for LR decay, the LR drops to $1/8$ of its initial value. Each method applies a single step at varying positions. **(Right) Local loss landscape exhibits blockwise structure.** One-dimensional loss landscapes at the final iterate $\boldsymbol{\theta}_T$ along a masked random perturbation direction $\boldsymbol{\sigma} \odot \boldsymbol{\delta}$. Here, $\boldsymbol{\theta} \in \mathbb{R}^p$ is the random perturbation vector, and $\boldsymbol{\sigma}$ is a blockwise mask that zeros out perturbations outside the specified block type.

to η/B. Together with Theorem 5.1, this suggests that decaying the LR and ramping the BS both reduce the noise scale, and thus may have similar effects on basin selection. A natural question then arises: *does BS ramping behave like LR decay in practice?*

To investigate this, we train models using either BS ramping or LR decay. Both methods apply a one-time step change: BS ramping multiplies the BS by 8 at T_{ramp}, while LR decay divides the LR by 8 at T_{decay}. We align T_{ramp} and T_{decay} so that the changes occur at the same positions, enabling a direct comparison of their effects. In Figure 8 (left), BS ramping and LR decay produce remarkably similar validation loss curves across all change positions. This result supports the theoretical prediction that both methods reduce the noise scale in comparable ways, thereby validating our theory.

Visualizing the Blockwise Local Loss Landscape. Transformer architecture is composed of different block types, such as query–key (QK) and value–output (VO) projections, feedforward networks (FFN), normalization layers (Norm), and embedding layers (Embed). Prior studies (Zhang et al., 2024; Wang et al., 2025) found that these block types exhibit heterogeneous levels of sharpness, suggesting that different block types contribute differently to the local loss landscape. To better understand this heterogeneity, we visualize the local loss landscape separately for each block type.

Similar to Figure 6, we perturb the final iterate $\boldsymbol{\theta}_T$ along a random direction $\boldsymbol{\delta}$, but restrict the perturbations to a selected block type using a blockwise mask $\boldsymbol{\sigma}$. In Figure 8 (right), local landscapes differs substantially across block types, and the curvature order we observe is QK $<$ Embed $<$ FFN $<$ VO $<$ Norm. This ordering is slightly different from the results reported by Wang et al. (2025), which found Embed $<$ QK $<$ FFN $<$ VO $<$ Norm. We suggest two possible reasons for this discrepancy. First, our analysis probes the most-case direction landscape, whereas Wang et al. (2024) directly estimate sharpness from the fisher information matrix. Second, embeddings may not appear as the flattest block in terms of loss landscape, but as they are least activated during gradient propagation (many embedding entries receive no gradient), they are effectively flatter in training dynamics.

7 Conclusion and Limitations

In conclusion, we present a unified theoretical and empirical view of how local landscape geometry evolves during language model pre-training. Our analysis reveals two phases: an early sharp-to-flat transition and a late noise-governed regime. The early dynamics explain the necessity of LR warmup, suggesting that larger peak LRs require proportionally longer warmup lengths. The late regime shows that BS controls a trade-off between widening and deepening loss basin, motivating a dynamic BS schedule that achieves lower terminal loss with significantly fewer tokens.

Limitations. We note that the current analysis primarily focuses on the `LLaMA-2` architecture and the AdamW optimizer. A natural future direction is to extend our empirical findings to other architectures and optimizers. We also note that the effects of BS ramping and LR decay have been studied separately. Understanding their combined impact remains an open question for future work.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

Huanran Chen, Yinpeng Dong, Zeming Wei, Yao Huang, Yichi Zhang, Hang Su, and Jun Zhu. Understanding pre-training and fine-tuning from loss landscape perspectives. *arXiv preprint arXiv:2505.17646*, 2025. 1

Jeremy Cohen, Alex Damian, Ameet Talwalkar, J Zico Kolter, and Jason D. Lee. Understanding optimization in deep learning with central flows. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2

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. 1, 3, 5

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. 2

Soham De, Abhay Yadav, David Jacobs, and Tom Goldstein. Automated inference with adaptive batches. In *Artificial Intelligence and Statistics*, pp. 1504–1513. PMLR, 2017. 3

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. 3

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. 1

Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Edward Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instabilities of deep learning models. In *International Conference on Learning Representations*, 2022. 1, 3, 6

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *International Conference on Learning Representations*, 2019. 3

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 arXiv:1706.02677*, 2017. 2

Gavia Gray, Anshul Samar, and Joel Hestness. Efficient and approximate per-example gradient norms for gradient noise scale. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*, 2023. 3

Gavia Gray, Shane Bergsma, Joel Hestness, et al. Normalization layer per-example gradients are sufficient to predict gradient noise scale in transformers. *Advances in Neural Information Processing Systems*, 37:93510–93539, 2024. 3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 2

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 3

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 1, 2, 4

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho*, and Krzysztof Geras*. The break-even point on optimization trajectories of deep neural networks. In *International Conference on Learning Representations*, 2020. 2

Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amost Storkey. On the relation between the sharpest directions of DNN loss and the SGD step length. In *International Conference on Learning Representations*, 2019. 2

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? underlying mechanisms and improvements. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 6

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 3

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

Atli Kosson, Bettina Messmer, and Martin Jaggi. Analyzing & reducing the need for learning rate warmup in GPT training. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

Tim Tsz-Kit Lau, Weijian Li, Chenwei Xu, Han Liu, and Mladen Kolar. Communication-efficient adaptive batch size strategies for distributed local gradient methods. *arXiv preprint arXiv:2406.13936*, 2024a. 3

Tim Tsz-Kit Lau, Han Liu, and Mladen Kolar. Adadagrad: Adaptive batch size schemes for adaptive gradient methods. *arXiv preprint arXiv:2402.11215*, 2024b. 3

Tim Tsz-Kit Lau, Weijian Li, Chenwei Xu, Han Liu, and Mladen Kolar. Adaptive batch size schedules for distributed training of language models with data and model parallelism. In *Proceedings of Conference on Parsimony and Learning*, 2025. 3

Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, et al. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025. 3, 8

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 5

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. 3

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018. 3, 8

Petr Ostroukhov, Aigerim Zhumabayeva, Chulu Xiang, Alexander Gasnikov, Martin Takáč, and Dmitry Kamzolov. Adabatchgrad: Combining adaptive batch size and adaptive step size. *arXiv preprint arXiv:2402.05264*, 2024. 3

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. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. 3

ShengYun Peng, Pin-Yu Chen, Matthew Daniel Hull, and Duen Horng Chau. Navigating the safety landscape: Measuring risks in finetuning large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019. 2

Minhak Song and Chulhee Yun. Trajectory alignment: understanding the edge of stability phenomenon via bifurcation theory. *arXiv preprint arXiv:2307.04204*, 2023. 1, 2

Tijmen Tieleman and Geoffrey Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *COURSERA Neural Networks Mach. Learn*, 17:6, 2012. 3

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 3

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017. 2

Jinbo Wang, Mingze Wang, Zhanpeng Zhou, Junchi Yan, Weinan E, and Lei Wu. The sharpness disparity principle in transformers for accelerating language model pre-training. In *Forty-second International Conference on Machine Learning*, 2025. 1, 9

Mingze Wang, Jinbo Wang, Haotian He, Zilin Wang, Guanhua Huang, Feiyu Xiong, Zhiyu li, Weinan E, and Lei Wu. Improving generalization and convergence by enhancing implicit regularization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 9

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *arXiv preprint arXiv:2410.05192*, 2024. 1

Lei Wu, Chao Ma, and Weinan E. How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 31:8279–8288, 2018. 2, 3, 5

Lei Wu, Mingze Wang, and Weijie J Su. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. 2

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021. 2

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George Dahl, Chris Shallue, and Roger B Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *Advances in neural information processing systems*, 32, 2019. 3

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How does critical batch size scale in pre-training? *International Conference on Learning Representations*, 2025. 3

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 2

Yushun Zhang, Congliang Chen, Tian Ding, Ziniu Li, Ruoyu Sun, and Zhi-Quan Luo. Why transformers need adam: A hessian perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 9

Zhanpeng Zhou, Mingze Wang, Yuchen Mao, Bingrui Li, and Junchi Yan. Sharpness-aware minimization efficiently selects flatter minima late in training. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3

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. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7654–7663. PMLR, 09–15 Jun 2019. 2

A STATEMENTS

A.1 LLM USAGE STATEMENT

We used the advanced language models only as writing assistants during manuscript preparation. Its role was limited to correcting grammar, improving clarity, and refining the flow of sentences, while keeping the intended meaning unchanged. All research ideas, methods, and results are entirely the work of the authors.

A.2 ETHICS STATEMENT

We confirm that this research fully complies with the ICLR Code of Ethics. All experiments were carried out with integrity, fairness, and transparency. The work involves no harm to humans, animals, or the environment, and we have ensured responsible handling of data, models, and computational resources.

A.3 REPRODUCIBILITY STATEMENT

We are confident that all experimental results in this work can be reproduced. Detailed descriptions of training and evaluation procedures, including hyperparameters, optimizer settings, and other relevant configurations, are provided in Section 3. In addition, we share open-source code in the supplemental material, and all datasets are publicly available.

B MISSING PROOF

B.1 LINEAR STABILITY ANALYSIS: THE NOISE-FREE CASE

Lemma B.1 (Spectral Properties of Preconditioned Curvature)**.** *Let* $\mathbf{H}(\boldsymbol{\theta}^\star) \succ 0$ *and* $\mathbf{M} \succ 0$*. Define* $\mathbf{S} = \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}$*. Then:*

1. $\mathbf{S}$ *is symmetric and positive definite.*
2. *The matrices* $\mathbf{MH}(\boldsymbol{\theta}^\star)$ *and* $\mathbf{S}$ *have identical eigenvalues.*

Proof. The properties are easily proved in the following:

1. Since $\mathbf{M}^{1/2} \succ 0$ and $\mathbf{H}(\boldsymbol{\theta}^\star) \succ 0$, we have $\mathbf{S} = \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2} \succ 0$. Symmetry follows from $\mathbf{S}^\top = (\mathbf{M}^{1/2})^\top\mathbf{H}(\boldsymbol{\theta}^\star)^\top(\mathbf{M}^{1/2})^\top = \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2} = \mathbf{S}$.
2. The matrices are similar: $\mathbf{MH}(\boldsymbol{\theta}^\star) = \mathbf{M}^{1/2}\mathbf{S}\mathbf{M}^{-1/2}$.

□

Lemma B.2 (Stability Condition for Preconditioned GD)**.** *Define the preconditioned curvature matrix* $\mathbf{S} := \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}$*, and let* $\{\lambda\}_{i=1}^p$ *be the eigenvalues of* $\mathbf{S}$*. The linear system in Equation* (4) *is asymptotically stable (i.e.,* $\lim_{k\to\infty} \boldsymbol{e}_k = \mathbf{0}$ *for any initial* $\boldsymbol{e}_0$*) if* η *satisfies*

$$0 < \eta < \frac{2}{\lambda_{\max}(\mathbf{S})}. \tag{8}$$

Proof. **Step 1: Orthogonal Diagonalization.** Since $\mathbf{S} \succ 0$ is symmetric, it admits an orthogonal diagonalization:

$$\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top,$$

where $\mathbf{Q}^\top\mathbf{Q} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d)$.

Define a new coordinate system $\boldsymbol{z}_k := \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_k$. Substituting into Equation (4):

$$\begin{aligned}
\boldsymbol{z}_{k+1} &= \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_{k+1} \\
&= \mathbf{Q}^\top\mathbf{M}^{-1/2}(\mathbf{I} - \eta\mathbf{MH}(\boldsymbol{\theta}^\star))\boldsymbol{e}_k \\
&= \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_k - \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\boldsymbol{e}_k
\end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{z}_k - \eta \mathbf{Q}^\top \mathbf{M}^{1/2} \mathbf{H}(\boldsymbol{\theta}^\star) \mathbf{M}^{1/2} \mathbf{Q} \boldsymbol{z}_k \\
&= \boldsymbol{z}_k - \eta \mathbf{Q}^\top \mathbf{S} \mathbf{Q} \boldsymbol{z}_k \\
&= (\mathbf{I} - \eta \boldsymbol{\Lambda}) \boldsymbol{z}_k.
\end{aligned}
$$

Thus, the dynamics in the $\boldsymbol{z}$-coordinates are decoupled:

$$
z_{k+1}^{(j)} = (1 - \eta \lambda_j)\, z_k^{(j)}, \quad \text{for } j = 1, \ldots, d. \tag{9}
$$

Step 2: Spectral Stability Criterion. The system in Equation (9) is asymptotically stable if and only if the magnitude of every scalar multiplier is less than 1:

$$
\max_j |1 - \eta \lambda_j| < 1.
$$

Since $\lambda_j > 0$ and $\eta > 0$ for all j, this inequality is equivalent to:

$$
\eta \lambda_j < 2 \quad \forall j \quad \Longleftrightarrow \quad \eta < \frac{2}{\max_j \lambda_j} = \frac{2}{\lambda_{\max}(\mathbf{S})}.
$$

This proves the sufficiency and necessity of the condition in Equation (8).

□

Lemma B.3 (Exact One-step Loss Change). *Let* $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$ *with* $\boldsymbol{\Lambda} = \mathit{diag}(\lambda_1, \ldots, \lambda_p)$ *and define* $\boldsymbol{z} = \mathbf{Q}^\top \mathbf{M}^{-1/2} \boldsymbol{e}_k$, *then*

$$
\frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = \sum_{j=1}^{p} w_j (1 - \eta \lambda_j)^2, \quad w_j := \frac{\lambda_j z_j^2}{\sum_{\ell=1}^{p} \lambda_\ell z_\ell^2} \in [0, 1], \quad \sum_j w_j = 1. \tag{10}
$$

In particular, $\min_{\boldsymbol{z}} \frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = (1 - \eta \lambda_{\max})^2$, *so that* $L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}_{k+1}) \xrightarrow{\eta \uparrow 2/\lambda_{\max}} 0$.

Proof. **Step 1. Using established coordinate transformation.** By Lemma B.2, we use the coordinate transformation $\boldsymbol{z}_k = \mathbf{Q}^\top \mathbf{M}^{-1/2} \boldsymbol{e}_k$. From the proof of Lemma B.2, we have the decoupled dynamics:

$$
z_{k+1}^{(j)} = (1 - \eta \lambda_j) z_k^{(j)}, \quad j = 1, \ldots, p.
$$

Step 2. Loss expression in transformed coordinates. The loss function relative to the optimal value can be expressed as:

$$
\begin{aligned}
L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star) &= \frac{1}{2} \boldsymbol{e}_k^\top \mathbf{H} \boldsymbol{e}_k \\
&= \frac{1}{2} (\mathbf{M}^{1/2} \mathbf{Q} \boldsymbol{z}_k)^\top \mathbf{H} (\mathbf{M}^{1/2} \mathbf{Q} \boldsymbol{z}_k) \\
&= \frac{1}{2} \boldsymbol{z}_k^\top \mathbf{Q}^\top \mathbf{M}^{1/2} \mathbf{H} \mathbf{M}^{1/2} \mathbf{Q} \boldsymbol{z}_k \\
&= \frac{1}{2} \boldsymbol{z}_k^\top \boldsymbol{\Lambda} \boldsymbol{z}_k = \frac{1}{2} \sum_{j=1}^{p} \lambda_j z_{k,j}^2.
\end{aligned}
$$

Step 3. Computing the loss ratio. Using the decoupled dynamics from Step 1:

$$
\begin{aligned}
L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star) &= \frac{1}{2} \sum_{j=1}^{p} \lambda_j z_{k+1,j}^2 \\
&= \frac{1}{2} \sum_{j=1}^{p} \lambda_j (1 - \eta \lambda_j)^2 z_{k,j}^2.
\end{aligned}
$$

Therefore, the loss ratio is:

$$
\frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = \frac{\sum_{j=1}^{p} \lambda_j (1 - \eta \lambda_j)^2 z_{k,j}^2}{\sum_{j=1}^{p} \lambda_j z_{k,j}^2}
$$

$$= \sum_{j=1}^{p} w_j (1 - \eta\lambda_j)^2,$$

where $w_j = \frac{\lambda_j z_{k,j}^2}{\sum_{\ell=1}^{p} \lambda_\ell z_{k,\ell}^2}$. Clearly, $w_j \in [0, 1]$ and $\sum_{j=1}^{p} w_j = 1$.

Step 4. Worst-case analysis. Since $\frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)}$ is a convex combination of $(1 - \eta\lambda_j)^2$, its minimum is achieved when all weight is concentrated on the largest eigenvalue:

$$\min_{\boldsymbol{z}} \frac{L(\boldsymbol{\theta}_{k+1}) - L(\boldsymbol{\theta}^\star)}{L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}^\star)} = (1 - \eta\lambda_{\max})^2.$$

As $\eta \uparrow 2/\lambda_{\max}$, we have $(1 - \eta\lambda_{\max})^2 \to 1$, and thus $L(\boldsymbol{\theta}_k) - L(\boldsymbol{\theta}_{k+1}) \to 0$. □

B.2 Stationary Distribution with Noise: The Stochastic Case

Having established the stability conditions for the deterministic case, we now analyze the full stochastic dynamics by including the noise term $\boldsymbol{\xi}_k$ in Equation (2). This analysis reveals how preconditioned SGD's stationary distribution depends on an effective temperature τ, leading to the emergence of basin selection through free energy minimization.

B.2.1 Discrete-Time Solution

Lemma B.4 (Eigenbasis Decomposition)**.** *Let* $\mathbf{S} := \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}$ *with eigendecomposition* $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$, $\boldsymbol{\Lambda} = \mathit{diag}(\lambda_1, \ldots, \lambda_d)$. *Define* $\mathbf{G} := \mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}\mathbf{Q}/B$. *In coordinates* $\boldsymbol{w}_k := \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_k$, *the recursion gives:*

$$\boldsymbol{w}_{k+1} = (\mathbf{I} - \eta\boldsymbol{\Lambda})\boldsymbol{w}_k + \eta\boldsymbol{\zeta}_k, \quad \mathbb{E}[\boldsymbol{\zeta}_k\boldsymbol{\zeta}_k^\top] = \mathbf{G}$$

The stationary covariance $\boldsymbol{\Sigma}_w$ *has diagonal elements:*

$$(\boldsymbol{\Sigma}_w)_{jj} = \frac{\eta^2\mathbf{G}_{jj}}{1 - (1 - \eta\lambda_j)^2} = \frac{\eta\mathbf{G}_{jj}}{2\lambda_j - \eta\lambda_j^2} \tag{11}$$

For small $\eta\lambda_j$, *this simplifies to:*

$$(\boldsymbol{\Sigma}_w)_{jj} \approx \frac{\eta\mathbf{G}_{jj}}{2\lambda_j} \tag{12}$$

Proof. First, we verify that $\mathbf{S} = \mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}$ can be eigendecomposed. Since both $\mathbf{M}$ and $\mathbf{H}(\boldsymbol{\theta}^\star)$ are positive definite matrices, $\mathbf{S}$ is also positive definite matrix. By the spectral theorem, $\mathbf{S}$ admits the eigendecomposition $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q}$ is orthogonal and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ with $\lambda_i > 0$.

Starting from $\boldsymbol{e}_{k+1} = \mathbf{A}\boldsymbol{e}_k + \eta\mathbf{M}\boldsymbol{\xi}_k$ with $\mathbf{A} = \mathbf{I} - \eta\mathbf{M}\mathbf{H}(\boldsymbol{\theta}^\star)$, we change variables to $\boldsymbol{w}_k = \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_k$.

$$\begin{aligned}
\boldsymbol{w}_{k+1} &= \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_{k+1} = \mathbf{Q}^\top\mathbf{M}^{-1/2}(\mathbf{A}\boldsymbol{e}_k + \eta\mathbf{M}\boldsymbol{\xi}_k) \\
&= \mathbf{Q}^\top\mathbf{M}^{-1/2}(\mathbf{I} - \eta\mathbf{M}\mathbf{H}(\boldsymbol{\theta}^\star))\boldsymbol{e}_k + \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k \\
&= \mathbf{Q}^\top\mathbf{M}^{-1/2}\boldsymbol{e}_k - \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\boldsymbol{e}_k + \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k \\
&= \boldsymbol{w}_k - \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\mathbf{H}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}\mathbf{Q}\boldsymbol{w}_k + \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k \\
&= \boldsymbol{w}_k - \eta\mathbf{Q}^\top\mathbf{S}\mathbf{Q}\boldsymbol{w}_k + \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k \\
&= (\mathbf{I} - \eta\boldsymbol{\Lambda})\boldsymbol{w}_k + \eta\mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k
\end{aligned}$$

Defining $\boldsymbol{\zeta}_k := \mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\xi}_k$, we get:

$$\mathbb{E}[\boldsymbol{\zeta}_k\boldsymbol{\zeta}_k^\top] = \mathbf{Q}^\top\mathbf{M}^{1/2}\mathbb{E}[\boldsymbol{\xi}_k\boldsymbol{\xi}_k^\top]\mathbf{M}^{1/2}\mathbf{Q} = \mathbf{Q}^\top\mathbf{M}^{1/2}\boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}\mathbf{Q}/B =: \mathbf{G}$$

As matrix $(\mathbf{I} - \eta\mathbf{\Lambda})$ is diagonal, the recursion now decouples into independent scalar equations for each component j:

$$(\boldsymbol{w}_{k+1})_j = (1 - \eta\lambda_j)(\boldsymbol{w}_k)_j + \eta(\boldsymbol{\zeta}_k)_j.$$

For each component j, the stationary variance satisfies:

$$(\mathbf{\Sigma}_w)_{jj} = (1 - \eta\lambda_j)^2(\mathbf{\Sigma}_w)_{jj} + \eta^2\mathbf{G}_{jj} \tag{13}$$

Solving for $(\mathbf{\Sigma}_w)_{jj}$:

$$(\mathbf{\Sigma}_w)_{jj} = \frac{\eta^2\mathbf{G}_{jj}}{1 - (1 - \eta\lambda_j)^2} = \frac{\eta^2\mathbf{G}_{jj}}{1 - (1 - 2\eta\lambda_j + \eta^2\lambda_j^2)}$$
$$= \frac{\eta^2\mathbf{G}_{jj}}{2\eta\lambda_j - \eta^2\lambda_j^2} = \frac{\eta\mathbf{G}_{jj}}{2\lambda_j - \eta\lambda_j^2}$$

For small $\eta\lambda_j \ll 1$, the denominator simplifies to $2\lambda_j$, giving Equation (12). □

B.2.2 Continuous-Time Limit and Gibbs Distribution

We now take the continuous-time limit ($\eta \to 0$) to derive a simpler universal theory.

The exact solution for the variance in the eigenbasis from Lemma B.4, i.e., $(\mathbf{\Sigma}_w)_{jj} = \eta\mathbf{G}_{jj}/(2\lambda_j - \eta\lambda_j^2)$, guides the necessary scaling for the continuous-time limit. Because $(\mathbf{\Sigma}_w)_{jj}$ converges to a finite non-zero value as $\eta \to 0$, the numerator $\eta\mathbf{G}_{jj}$ must remain finite. This suggests defining a quantity τ such that for each mode j:

$$\eta\,\mathbf{G}_{jj} \to 2\tau \quad \text{as} \quad \eta \to 0.$$

We strengthen this to :

$$\eta\,\mathbf{G} \to 2\tau\mathbf{I} \quad \text{as} \quad \eta \to 0.$$

Recalling that $\mathbf{G} = \mathbf{Q}^\top\mathbf{M}^{1/2}\mathbf{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^{1/2}\mathbf{Q}/B$, this condition in the original coordinate system translates to the required scaling for the noise covariance:

$$\frac{\eta}{B}\mathbf{M}\mathbf{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^\top \to 2\tau\mathbf{M}.$$

Proposition B.1 (Convergence to SDE)**.** *Consider the scaled discrete process $\boldsymbol{\theta}_{\lfloor t/\eta \rfloor}$ as $\eta \to 0$. Suppose the noise covariance satisfies*

$$\frac{\eta}{B}\mathbf{M}\mathbf{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^\top = 2\tau\mathbf{M} + O(\eta), \tag{14}$$

for some temperature $\tau > 0$. Then the process converges weakly to the Itô SDE:

$$d\boldsymbol{\theta}_t = -\mathbf{M}\nabla L(\boldsymbol{\theta}_t)dt + \sqrt{2\tau}\mathbf{M}^{1/2}dW_t \tag{15}$$

where W_t is standard Brownian motion.

Proof. Consider the discrete preconditioned SGD update:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\mathbf{M}(\nabla L(\boldsymbol{\theta}_k) + \boldsymbol{\xi}_k),$$

Define the scaled process $\boldsymbol{\theta}^{(\eta)}(t) = \boldsymbol{\theta}_{\lfloor t/\eta \rfloor}$. The increment $\Delta\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k$ satisfies:

$$\mathbb{E}[\Delta\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_k = \boldsymbol{\theta}] = -\eta\mathbf{M}\nabla L(\boldsymbol{\theta}),$$

$$\mathrm{Cov}(\Delta\boldsymbol{\theta}_k \mid \boldsymbol{\theta}_k = \boldsymbol{\theta}) = \frac{\eta^2}{B}\mathbf{M}\mathbf{\Sigma}(\boldsymbol{\theta}^\star)\mathbf{M}^\top.$$

Given the scaling condition Equation (14), the covariance is $O(\eta)$.

The generator $\mathcal{L}^{(\eta)}$ of the discrete process for a smooth function $\boldsymbol{f}$ is:

$$\mathcal{L}^{(\eta)}\boldsymbol{f}(\boldsymbol{\theta}) = \frac{1}{\eta}\mathbb{E}[\boldsymbol{f}(\boldsymbol{\theta}_{k+1}) - \boldsymbol{f}(\boldsymbol{\theta}_k) \mid \boldsymbol{\theta}_k = \boldsymbol{\theta}].$$

Using a Taylor expansion and taking conditional expectation:

$$\mathbb{E}[\boldsymbol{f}(\boldsymbol{\theta}_{k+1}) - \boldsymbol{f}(\boldsymbol{\theta}_k) \mid \boldsymbol{\theta}] = -\eta \nabla \boldsymbol{f}(\boldsymbol{\theta})^\top \mathbf{M} \nabla L(\boldsymbol{\theta}) + \frac{1}{2}\mathbb{E}[(\Delta\boldsymbol{\theta})^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}] + O(\eta^{3/2}).$$

For the second term, with $\Delta\boldsymbol{\theta} = -\eta M(\nabla L(\boldsymbol{\theta}) + \boldsymbol{\xi}_k)$:

$$\begin{aligned}
\mathbb{E}[(\Delta\boldsymbol{\theta})^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}] &= \eta^2 \mathbb{E}[(\nabla L(\boldsymbol{\theta}) + \boldsymbol{\xi}_k)^\top \mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \mathbf{M} (\nabla L(\boldsymbol{\theta}) + \boldsymbol{\xi}_k)] \\
&= \eta^2 \mathbb{E}[\boldsymbol{\xi}_k^\top \mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \mathbf{M} \boldsymbol{\xi}_k] + O(\eta^2) \\
&= \eta^2 \mathrm{Tr}(\mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \mathbf{M} \mathbb{E}[\boldsymbol{\xi}_k \boldsymbol{\xi}_k^\top]) + O(\eta^2) \\
&= \frac{\eta^2}{B} \mathrm{Tr}(\mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \mathbf{M} \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star)) + O(\eta^2) \\
&= \frac{\eta^2}{B} \mathrm{Tr}(\mathbf{M} \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star) \mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})) + O(\eta^2)
\end{aligned}$$

where we used $\mathbb{E}[\boldsymbol{\xi}^\top \mathbf{A} \boldsymbol{\xi}] = \mathrm{Tr}(A\mathbb{E}[\boldsymbol{\xi}\boldsymbol{\xi}^\top])$ and trace cyclicity $\mathrm{Tr}(\mathbf{ABC}) = \mathrm{Tr}(\mathbf{CAB})$.

Therefore:

$$\frac{1}{2}\mathbb{E}[(\Delta\boldsymbol{\theta})^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta}) \Delta\boldsymbol{\theta}] = \frac{\eta^2}{2B} \mathrm{Tr}(\mathbf{M} \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star) \mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})) + O(\eta^2).$$

Using the scaling condition Equation (14), we have :

$$\frac{\eta^2}{2B} \mathrm{Tr}(\mathbf{M} \boldsymbol{\Sigma}(\boldsymbol{\theta}^\star) \mathbf{M}^\top \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})) = \frac{\eta}{2} \mathrm{Tr}\left(2\tau \mathbf{M} \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})\right) + O(\eta^2) = \eta\tau \mathrm{Tr}(\mathbf{M} \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})) + O(\eta^2).$$

Thus,

$$\mathcal{L}^{(\eta)} \boldsymbol{f}(\boldsymbol{\theta}) = -\nabla \boldsymbol{f}(\boldsymbol{\theta})^\top \mathbf{M} \nabla L(\boldsymbol{\theta}) + \tau \mathrm{Tr}(\mathbf{M} \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})) + O(\eta).$$

As $\eta \to 0$, $\mathcal{L}^{(\eta)} \boldsymbol{f}(\boldsymbol{\theta})$ converges to:

$$\mathcal{L} \boldsymbol{f}(\boldsymbol{\theta}) = -\nabla \boldsymbol{f}(\boldsymbol{\theta})^\top \mathbf{M} \nabla L(\boldsymbol{\theta}) + \tau \mathrm{Tr}(\mathbf{M} \nabla^2 \boldsymbol{f}(\boldsymbol{\theta})),$$

which is the generator of the Itô SDE:

$$d\boldsymbol{\theta}_t = -\mathbf{M} \nabla L(\boldsymbol{\theta}_t) dt + \sqrt{2\tau} \mathbf{M}^{1/2} dW_t.$$

By the weak convergence theory (e.g., via the martingale problem or generator convergence), the process $\boldsymbol{\theta}^{(\eta)}(t)$ converges weakly to the solution of this SDE. □

Proposition B.2 (Gibbs Stationary Distribution)**.** *The SDE in Equation* (15) *has stationary distribution:*

$$p_\infty(\boldsymbol{\theta}) \propto \exp(-L(\boldsymbol{\theta})/\tau) \tag{16}$$

Proof. The generator of the SDE (5) is $\mathcal{L}f = -\mathbf{M}\nabla L \cdot \nabla f + \tau \mathrm{tr}(\mathbf{M}\nabla^2 f)$. The Fokker-Planck equation for the probability density $p(t, \boldsymbol{\theta})$ is:

$$\partial_t p = \mathcal{L}^* p = \nabla \cdot (\mathbf{M} \nabla L \, p) + \tau \nabla \cdot (\mathbf{M} \nabla p)$$

where $\mathcal{L}^*$ is the adjoint operator. Setting $\partial_t p = 0$ for stationarity:

$$\begin{aligned}
0 &= \nabla \cdot (\mathbf{M} \nabla L \, p_\infty) + \tau \nabla \cdot (\mathbf{M} \nabla p_\infty) \\
&= \nabla \cdot (\mathbf{M} \nabla L \, p_\infty + \tau \mathbf{M} \nabla p_\infty)
\end{aligned}$$

This implies the current $J = \mathbf{M} \nabla L \, p_\infty + \tau \mathbf{M} \nabla p_\infty$ has zero divergence. For a potential-driven system, we require $J = \mathbf{0}$:

$$\begin{aligned}
\mathbf{M} \nabla L \, p_\infty + \tau \mathbf{M} \nabla p_\infty &= \mathbf{0} \\
\nabla L \, p_\infty + \tau \nabla p_\infty &= \mathbf{0} \quad \text{(since } \mathbf{M} \succ 0\text{)} \\
\frac{\nabla p_\infty}{p_\infty} &= -\frac{\nabla L}{\tau}
\end{aligned}$$

Integrating: $\log p_\infty = -L/\tau + \text{const}$, which gives Equation (16). □

B.2.3 Basin Selection via Free Energy

Theorem B.1 (Free Energy Minimization)**.** *Let the empirical risk $L(\boldsymbol{\theta})$ admit multiple local minima $\{\boldsymbol{\theta}_i^\star\}_{i=1}^m$ with Hessians $\mathbf{H}(\boldsymbol{\theta}_i^\star) \succ 0$. Under the SDE in Equation (15) with temperature τ, the stationary probability that training resides in basin i is given by:*

$$P_\tau(\textit{basin } i) = \frac{\exp(-F_i(\tau)/\tau)}{\sum_j \exp(-F_j(\tau)/\tau)}, \quad F_i(\tau) := L(\boldsymbol{\theta}_i^\star) + \frac{\tau}{2}\log\det \mathbf{H}(\boldsymbol{\theta}_i^\star). \tag{17}$$

Proof. From the Gibbs distribution Equation (16), the probability mass in basin i is:

$$P_\tau(\text{basin } i) = \frac{\int_{B_i} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta}}{\int_{\mathbb{R}^d} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta}}$$

where B_i is the basin of attraction around minimum $\boldsymbol{\theta}_i^\star$.

For the numerator, using the quadratic approximation $L(\boldsymbol{\theta}) = L(\boldsymbol{\theta}_i^\star) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)^\top \mathbf{H}(\boldsymbol{\theta}_i^\star)(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)$ in basin i we get:

$$\begin{aligned}\int_{B_i} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta} &= \int_{\mathbb{R}^d} \exp\left(-\frac{L(\boldsymbol{\theta}_i^\star)}{\tau} - \frac{1}{2\tau}(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)^\top \mathbf{H}(\boldsymbol{\theta}_i^\star)(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)\right) d\boldsymbol{\theta} \\ &= e^{-L(\boldsymbol{\theta}_i^\star)/\tau} \int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\tau}(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)^\top \mathbf{H}(\boldsymbol{\theta}_i^\star)(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)\right) d\boldsymbol{\theta}\end{aligned}$$

The integral is a multivariate Gaussian with covariance $\tau \mathbf{H}(\boldsymbol{\theta}_i^\star)^{-1}$. Using the standard formula for Gaussian integrals:

$$\int_{\mathbb{R}^d} \exp\left(-\frac{1}{2} y^\top \boldsymbol{\Sigma}^{-1} y\right) dy = (2\pi)^{d/2} (\det \boldsymbol{\Sigma})^{1/2}$$

With $\boldsymbol{\Sigma} = \tau \mathbf{H}(\boldsymbol{\theta}_i^\star)^{-1}$, we have $\det \boldsymbol{\Sigma} = \tau^d (\det \mathbf{H}(\boldsymbol{\theta}_i^\star))^{-1}$ and $\boldsymbol{\Sigma}^{-1} = \tau^{-1}\mathbf{H}(\boldsymbol{\theta}_i^\star)$:

$$\begin{aligned}\int_{\mathbb{R}^d} \exp\left(-\frac{1}{2\tau}(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)^\top \mathbf{H}(\boldsymbol{\theta}_i^\star)(\boldsymbol{\theta} - \boldsymbol{\theta}_i^\star)\right) d\boldsymbol{\theta} &= (2\pi)^{d/2} (\tau^d (\det \mathbf{H}(\boldsymbol{\theta}_i^\star))^{-1})^{1/2} \\ &= (2\pi\tau)^{d/2} (\det \mathbf{H}(\boldsymbol{\theta}_i^\star))^{-1/2}\end{aligned}$$

Therefore:

$$\begin{aligned}\int_{B_i} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta} &= e^{-L(\boldsymbol{\theta}_i^\star)/\tau} (2\pi\tau)^{d/2} (\det \mathbf{H}(\boldsymbol{\theta}_i^\star))^{-1/2} \\ &= (2\pi\tau)^{d/2} \exp\left(-L(\boldsymbol{\theta}_i^\star)/\tau - \frac{1}{2}\log\det \mathbf{H}(\boldsymbol{\theta}_i^\star)\right) \\ &= (2\pi\tau)^{d/2} \exp\left(-\frac{1}{\tau}\left(L(\boldsymbol{\theta}_i^\star) + \frac{\tau}{2}\log\det \mathbf{H}(\boldsymbol{\theta}_i^\star)\right)\right) \\ &= (2\pi\tau)^{d/2} \exp(-F_i(\tau)/\tau)\end{aligned}$$

Similarly, the total partition function is:

$$\begin{aligned}Z(\tau) = \int_{\mathbb{R}^d} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta} &= d \sum_{j=1}^m \int_{B_j} e^{-L(\boldsymbol{\theta})/\tau} d\boldsymbol{\theta} \\ &= (2\pi\tau)^{d/2} \sum_{j=1}^m \exp(-F_j(\tau)/\tau)\end{aligned}$$

Therefore:

$$P_\tau(\text{basin } i) = \frac{(2\pi\tau)^{d/2} \exp(-F_i(\tau)/\tau)}{(2\pi\tau)^{d/2} \sum_j \exp(-F_j(\tau)/\tau)} = \frac{\exp(-F_i(\tau)/\tau)}{\sum_j \exp(-F_j(\tau)/\tau)}$$

This completes the proof of the free energy formula Equation (17). □

C More Experimental Results

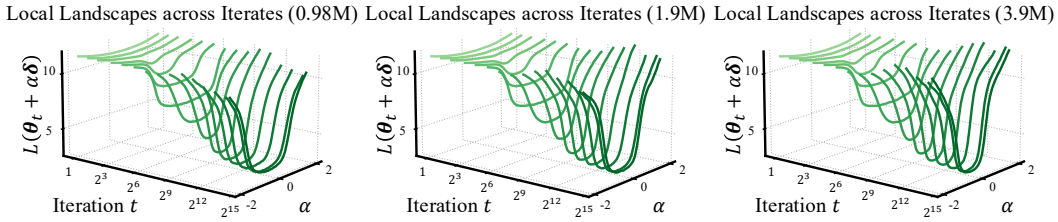

Figure 9: **Appendix: the evolution of local loss landscape throughout pre-training.** We train `LLaMA-2` models with 170M parameters using different BSs (0.98M, 1.9M and 3.9M), and visualize the one-dimensional loss landscape at iterate $\boldsymbol{\theta}_t$ along a random direction $\boldsymbol{\delta}$, i.e., plot $L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ vs. the perturbation coefficient α. The landscapes are shown across different training iterations t. **Early phase.** The landscapes gradually widens/flattens for both training runs. **Late phase.** Training with smaller BS produces wider landscapes than training with larger BS.

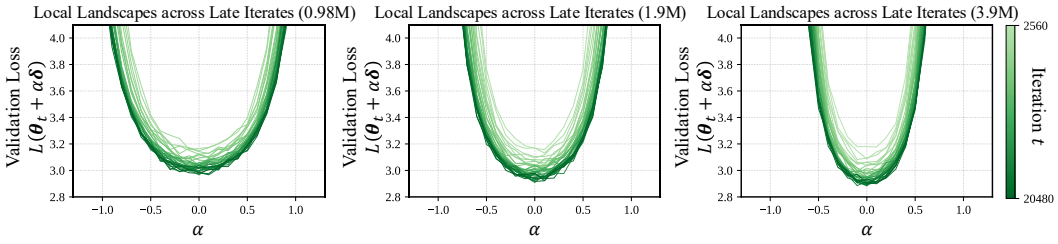

Figure 10: **Appendix: large BS deepens the basin, small BS widens the basin.** We train a series of `LLaMA-2` models (170M) for $T = 20{,}480$ iterations, using BSs $B \in \{0.49\text{M}, 0.98\text{M}, 1.9\text{M}, 3.9\text{M}, 7.8\text{M}\}$. One-dimensional loss landscape: the perturbed loss $L(\boldsymbol{\theta}_t + \alpha\boldsymbol{\delta})$ vs. perturbation coefficient α, shown across late training iterations t for $B = 0.98\text{M}$, $B = 1.9\text{M}$ and $B = 3.9\text{M}$.