The Effect of Mini-Batch Noise on the Implicit Bias of Adam

Anonymous Authors[1]

Abstract

With limited high-quality data and growing compute, multi-epoch training is gaining back its importance across sub-areas of deep learning. Adam(W), versions of which are go-to optimizers for many tasks such as next token prediction, has two momentum hyperparameters (β_1, β_2) controlling memory and one very important hyperparameter, batch size, controlling (in particular) the amount mini-batch noise. We introduce a theoretical framework to understand how mini-batch noise influences the implicit bias of memory in Adam (depending on β_1, β_2) towards sharper or flatter regions of the loss landscape, which is commonly observed to correlate with the generalization gap in multi-epoch training. We find that in the case of large batch sizes, higher β_2 increases the magnitude of anti-regularization by memory (hurting generalization), but as the batch size becomes smaller, the dependence of (anti-)regulariation on β_2 is reversed. A similar monotonicity shift (in the opposite direction) happens in β_1. In particular, the commonly "default" pair $(\beta_1, \beta_2) = (0.9, 0.999)$ is a good choice if batches are small; for larger batches, in many settings moving β_1 closer to β_2 is much better in terms of validation accuracy in multi-epoch training. Moreover, our theoretical derivations connect the scale of the batch size at which the shift happens to the scale of the critical batch size. We illustrate this effect in experiments with small-scale data in the about-to-overfit regime.

1. Introduction

Modifications of Adam (Kingma & Ba, 2014) such as AdamW (Loshchilov & Hutter, 2019) and AdaFactor (Shazeer & Stern, 2018) have become the standard optimizers for important deep learning tasks like training language models (Brown et al., 2020b; Anil et al., 2023; Touvron et al., 2023; Dubey et al., 2024). Apart from the learning rate, the most important hyperparameters are batch size b, and momentum hyperparameters (betas) (β_1, β_2). The choice of their values is a crucial part of preparing the training pipeline, and practitioners often employ commonly accepted heuristics when setting these hyperparameters.

Kingma & Ba (2014) recommend setting $\beta_1 = 0.9$ and $\beta_2 = 0.999$. They report a grid search of $\beta_1 \in \{0, 0.9\}$ and $\beta_2 \in \{0.99, 0.999, 0.9999\}$ on a variational autoencoder training problem, comparing training speeds after 10 and 100 epochs. Based on this recommendation, the values $(\beta_1, \beta_2) = (0.9, 0.999)$ have become the default in many libraries such as PyTorch and Optax. It is conventional wisdom that adaptive gradient methods work well with their default hyperparameters (Sivaprasad et al., 2020). This practical success appears to be the reason why practitioners rely on default values. Recently, β_2 has been adjusted to be smaller (specifically $\beta_2 = 0.95$) when training large models with AdamW (Brown et al., 2020b; Zhang et al., 2022; Zeng et al., 2022; Biderman et al., 2023; Touvron et al., 2023; Dubey et al., 2024) because it was observed that this improves training stability. In short, (β_1, β_2) are usually set by empirical tuning, whereas a foundational theoretical understanding is limited.

When comparing hyperparameter values, different performance metrics may be of importance. Large pretraining runs have often been done with only one (or less) pass over available data (Brown et al., 2020a). For such a run, some of the most important metrics are training stability, optimization speed, compute efficiency, etc. However, there is a potential shift towards multi-epoch training because of limited high-quality data (Villalobos et al., 2024; Kim et al., 2025). In addition, parts of the post-training pipeline are multi-epoch with high potential for overfitting (Xiao, 2025). For multi-epoch training, including pretraining under data constraints (Kim et al., 2025), generalization properties are important, such as the gap between validation and train losses.

Given the existing, and potentially rising (after leaving the spotlight for a while) importance of multi-epoch training, we theoretically investigate how the choice of (β_1, β_2) and

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

batch size b can affect loss landscape sharpness, a correlate of generalization (Jiang et al., 2020; Foret et al., 2021). To the best of our knowledge, this is the first such investigation. We extend the framework of the implicit bias of memory (Cattaneo & Shigida, 2025a), targeting specifically mini-batch noise effects.

1.1. Organization and Summary of Contributions

We introduce in Section 2 the framework of finding and interpreting implicit bias terms during mini-batch training with an algorithm having memory (where presence of *memory* means the next iterate depends on the whole history of previous gradients). This exposition is illustrative and uses SGD with momentum as an example.

In Section 3, we apply this approach to mini-batch Adam. We find that mini-batch noise has a large effect on which (β_1, β_2) are better for generalization. Fixing β_1 at usual values (0.9 or 0.99), we ask which β_2 is the best. For small batch sizes, higher β_2 compensates for implicit anti-penalization of sharpness by memory in Adam, and therefore leads to better predicted generalization. Hence, the default $(\beta_1, \beta_2) = (0.9, 0.999)$ are suitable for very noisy training in a regime where overfitting is a concern. However, as batches grow larger, the monotonicity direction changes: in the low-noise setting, the larger β_2 the stronger anti-penalization of sharpness, the worse predicted generalization. Hence, with large batches, it is best to take β_1 and β_2 close to each other. As we review in Section 1.2, the prescription is consistent with a substantial body of empirical work, which showcases the usefulness of this theoretical framework.

We show (in the same section) that a similar trend exists when β_1 is swept, with β_2 fixed at a common value like 0.999. For large batches and full-batch training, the larger β_1 the better (consistent with Cattaneo et al. (2024)), but this monotonicity reverts as mini-batch noise increases: for small batches, β_1 should be much smaller than $\beta_2 = 0.999$, again calling for default $(0.9, 0.999)$.

In Section 4, we confirm our theoretical findings by training a language model on a small dataset, allowing it to overfit and comparing the best validation losses achieved.

1.2. Related Work

Tuning hyperparameters of Adam Ma et al. (2022) investigate theoretically and empirically the qualitative features of full-batch Adam depending on (β_1, β_2), dividing possible training into three regimes (oscillations, spikes and divergence) and advocating for $\beta_1 = \beta_2$ for faster and smoother training. The latter prescription is consistent with our theory although we focus on different metrics (loss landscape sharpness / flatness), and we argue that increasing mini-batch noise changes the conclusions and recommendations. Relatedly, Zhao et al. (2025) include Adam's (β_1, β_2) sweeps and find that if $\beta_1 = \beta_2$, Adam behaves similarly to signed momentum (Signum), and the recently common setting for language models $(\beta_1, \beta_2) = (0.9, 0.95)$ is close to this. For small batches, however, it is empirically observed to be beneficial to increase β_2 relative to β_1. In particular, Zhang et al. (2025) recommend (based on empirical sweeps) taking smaller β_2 relative to β_1 if batch sizes are large and higher when batch sizes are small, exactly matching our theory-based prescription. There are other prior works advocating for that, and they use different principles (Porian et al., 2024; Marek et al., 2025). To the best of our knowledge, we provide the first theoretical argument based on generalization. Other works that have a substantial focus on (β_1, β_2) sweeps in Adam include Schmidt et al. (2021); Orvieto & Gower (2025); Wen et al. (2025); Pagliardini et al. (2025).

SDE approximations In the context of mini-batch noise, theoretical analysis of many optimizers often employs approximations by stochastic differential equations (SDEs). In particular, Zhou et al. (2020); Xie et al. (2022) approximate Adam and SGD with (different types of) SDEs, and use the escaping time from local minima to predict better generalization of SGD compared to Adam. The works Malladi et al. (2022); Compagnoni et al. (2025) also approximate Adam with SDEs under different assumptions and propose scaling rules for hyperparameters. Zhou et al. (2024) focus on the advantages of decoupled weight decay for generalization. These works differ substantially from the present article in purpose, methods and assumptions; in particular, typically β_1 and β_2 are assumed to converge to 1 at certain rates as step size goes to zero, whereas we consider them fixed. In addition, we do not assume that mini-batch noise in the gradients forms an i. i. d. random sequence (since we consider sampling without replacement), we are agnostic to its distribution, and we do not use distributional asymptotics.

Sharpness and generalization There has been a long history of relating flatter minima to better generalization (Hochreiter & Schmidhuber, 1994; Keskar et al., 2017; Jiang et al., 2020). There has also been some criticism based on the sensitivity of standard sharpness measures to rescaling the network's parameters even if it does not change the network's outputs (Dinh et al., 2017). In response, different scale-invariant sharpness metrics have been introduced (Yi et al., 2019; Tsuzuku et al., 2020; Rangamani et al., 2021; Kwon et al., 2021); however, empirical evidence is still mixed (Andriushchenko et al., 2023). Numerous works have explored explicit sharpness penalization to improve generalization, of which we can only name a few (Foret et al., 2021; Kwon et al., 2021; Zheng et al., 2021; Kim et al., 2022; Du et al., 2022; Liu et al., 2022; Li & Giannakis, 2023; Xie et al., 2024; Tahmasebi et al., 2024; Li et al., 2024). We study implicit, rather than explicit, penalization but otherwise our theory-based perspective is consistent with this literature.

Implicit bias A large strand of literature describes implicit biases of optimization algorithms by proving convergence to a max-margin solution (Soudry et al., 2018; Nacson et al., 2019b;c; Qian & Qian, 2019; Wang et al., 2022; Gunasekar et al., 2018a; Ji & Telgarsky, 2018b; 2019; 2018a; Gunasekar et al., 2018b; Ji & Telgarsky, 2020; Nacson et al., 2019a; Lyu & Li, 2019; Wang et al., 2021). The implicit bias of weight decay in AdamW is tackled in Zhang et al. (2019); Zhuang et al. (2022); Andriushchenko et al. (2024); Xie & Li (2024); Kobayashi et al. (2024) and others. Implicit regularization by biasing towards flatter minima at convergence is studied in Damian et al. (2021); Arora et al. (2022) besides works already listed. Most relatedly to our work, a large body of literature demonstrates implicit penalization of a gradient norm, using modified equations for SGD with or without momentum (Barrett & Dherin, 2021; Miyagawa, 2022; Smith et al., 2021; Farazmand, 2020; Kovachki & Stuart, 2021; Ghosh et al., 2023; Rosca et al., 2023), for full-batch Adam (Cattaneo et al., 2024), or using correction terms after removing memory (Cattaneo & Shigida, 2025a). Beneventano (2023) studies the difference between SGD with or without replacement with similar methods. We build on and extend this literature.

2. Proposed Theoretical Framework

Let us start by formulating the setting of multi-batch training for one (large) epoch, in which batches are sampled without replacement, as commonly done in practice.

Definition 2.1 (Losses and gradients)**.** *We will assume there are m batches in an epoch with each batch consisting of b samples, $N := mb$ samples in total, and the kth* ***mini-batch loss*** *is defined by*

$$\mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{b} \sum_{r=kb+1}^{kb+b} \ell_{\pi(r)}(\boldsymbol{\theta}), \quad k \in [0 : m-1],$$

where $\{\ell_s\}_{s=1}^N$ are ***per-sample losses*** *and $\pi \colon [1 : N] \to [1 : N]$ is a random permutation of samples chosen uniformly (that is, the probability of each permutation is $1/N!$), and $\boldsymbol{\theta}$ is the vector of all parameters of a model. The* ***full-batch loss*** *is the average of mini-batch losses:*

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{k=0}^{m-1} \mathcal{L}_k(\boldsymbol{\theta}) = \frac{1}{N} \sum_{r=1}^{N} \ell_r(\boldsymbol{\theta}).$$

To slightly shorten notations, define $n := m - 1$ (so that $\mathcal{L}_n$ is the last mini-batch loss). The ***loss gradient*** *and its sign deserve separate notations (omitting the dependence on $\boldsymbol{\theta}$ whenever this point is fixed and clear from context):*

$$g_i := \nabla_i \mathcal{L}(\boldsymbol{\theta}), \quad s_i := \operatorname{sign} g_i.$$

Definition 2.2 (Noise derivative tensors, empirical covariance matrix)**.** *We will denote*

$$d_k := (\mathcal{L}_k - \mathcal{L})(\boldsymbol{\theta}), \quad d_{k,i} := \nabla_i (\mathcal{L}_k - \mathcal{L})(\boldsymbol{\theta}),$$
$$d_{k,ij} := \nabla_{ij} (\mathcal{L}_k - \mathcal{L})(\boldsymbol{\theta}), \quad \text{etc.}$$

the ***mini-batch noise*** *and its derivatives (omitting the dependence on $\boldsymbol{\theta}$). Further, we define the* ***empirical covariance matrix $\boldsymbol{\Sigma}$*** *of per-sample gradients:*

$$\Sigma_{ij} := \frac{1}{mb} \sum_{p=1}^{mb} \nabla_i (\ell_p - \mathcal{L})(\boldsymbol{\theta}) \nabla_j (\ell_p - \mathcal{L})(\boldsymbol{\theta}).$$

Consider a general optimization algorithm that has memory (recall that this means the update depends on the whole history of previous iterates rather than one last iterate):

$$\boldsymbol{\theta}_{t+1} = \underbrace{\boldsymbol{\theta}_t - \eta \boldsymbol{F}_t(\boldsymbol{\theta}_t, \ldots, \boldsymbol{\theta}_0)}_{\text{depends on the whole history } \boldsymbol{\theta}_t, \ldots, \boldsymbol{\theta}_0}. \tag{1}$$

Cattaneo & Shigida (2025a) convert it into a memoryless iteration

$$\tilde{\boldsymbol{\theta}}_{t+1} = \underbrace{\tilde{\boldsymbol{\theta}}_t - \eta \operatorname{\mathsf{Main}}_t(\tilde{\boldsymbol{\theta}}_t) - \eta^2 \operatorname{\mathsf{Corr}}_t(\tilde{\boldsymbol{\theta}}_t)}_{\text{only depends on } \tilde{\boldsymbol{\theta}}_t \text{ (no memory)}}, \tag{2}$$

in such a way that the trajectories $\{\boldsymbol{\theta}_t\}$ and $\{\tilde{\boldsymbol{\theta}}_t\}$ stay globally $O(\eta^2)$-close for $O(\eta^{-1})$ iterations, provided that the **main iterm** and the **correction term** are carefully chosen:

$$\begin{aligned} \mathsf{Main}_t(\boldsymbol{\theta}) &:= \boldsymbol{F}_t(\boldsymbol{\theta}, \ldots, \boldsymbol{\theta}), \\ \mathsf{Corr}_{t,r}(\boldsymbol{\theta}) &:= \sum_{k=1}^{t} \frac{\partial F_{t,r}}{\partial \boldsymbol{\theta}_{t-k}}(\boldsymbol{\theta})^{\mathsf{T}} \sum_{s=t-k}^{t-1} \boldsymbol{F}_s(\boldsymbol{\theta}). \end{aligned} \tag{3}$$

We give the formal statement below.

Theorem 2.3 (Corollary 3.3 in Cattaneo & Shigida (2025a))**.** *Let $\mathcal{D}$ be an open convex domain in $\mathbb{R}^{\dim \boldsymbol{\theta}}$ and $\boldsymbol{F}_t \in C^2(\mathcal{D}^{t+1}; \mathbb{R}^d)$ be a family of functions, such that for any $t \in \mathbb{Z}_{\geq 0}$, $k_1, k_2 \in [0 : t]$, $r, i, j \in [1 : \dim \boldsymbol{\theta}]$,*

$$|F_{t,r}| \leq \gamma_{-1}, \quad \left| \frac{\partial F_{t,r}}{\partial \theta_{t-k_1,i}} \right| \leq \gamma_{k_1},$$
$$\left| \frac{\partial^2 F_{t,r}}{\partial \theta_{t-k_1,i} \partial \theta_{t-k_2,j}} \right| \leq \gamma_{k_1,k_2},$$

where γ_{-1}, γ_{k_1} and γ_{k_1,k_2} are families of positive reals (not depending on t) satisfying $\sum_{k_1=1}^{\infty} \gamma_{k_1} k_1^2 +$

$\sum_{k_1,k_2=1}^{\infty} \gamma_{k_1,k_2} k_1 k_2 < \infty$ *(sufficiently fast decay of memory). Then iterations* $\{\boldsymbol{\theta}_t\}_{t=0}^{\infty}$ *and* $\{\tilde{\boldsymbol{\theta}}_t\}_{t=0}^{\infty}$*, given in Equations* (1) *to* (3) *with the same initial condition* $\tilde{\boldsymbol{\theta}}_0 = \boldsymbol{\theta}_0$*, satisfy*

$$\max_{t \in [0:\lfloor T/\eta \rfloor]} \left\| \boldsymbol{\theta}_t - \tilde{\boldsymbol{\theta}}_t \right\|_\infty \leq C\eta^2$$

for some constant C *not depending on* η*.*

2.1. Warm-up: SGD with Momentum

Let us consider the mini-batch version of the simplest algorithm that has memory: SGD with momentum, given by Equation (1) with $F_t(\boldsymbol{\theta}_t, \ldots, \boldsymbol{\theta}_0) := \sum_{k=0}^{t} \beta^{t-k} \nabla \mathcal{L}_k(\boldsymbol{\theta}_k)$. We consider this simple example to introduce the main ideas and concepts underlying our general framework; see Cattaneo & Shigida (2025b) for a fine-grained analysis of this specific algorithm.

Theorem 2.3 gives an approximation (2) with

$$\begin{aligned}
\mathsf{Main}_t(\boldsymbol{\theta}) &= \sum_{k=0}^{t} \beta^{t-k} \nabla \mathcal{L}_k(\boldsymbol{\theta}), \\
\mathsf{Corr}_t(\boldsymbol{\theta}) &= \beta \sum_{b=0}^{t-1} \beta^b \sum_{l'=1}^{b+1} \sum_{b'=0}^{t-l'} \beta^{b'} \times \\
&\quad \times \nabla^2 \mathcal{L}_{t-1-b}(\boldsymbol{\theta}) \nabla \mathcal{L}_{t-l'-b'}(\boldsymbol{\theta}).
\end{aligned} \tag{4}$$

The approximating algorithm does not have memory, so Main_t and Corr_t only depend on one point, which is already a significant simplification. However, in this form these expressions are still very complex and their analysis appears impossible. The next move (due to Smith et al. (2021)), is to put $t = n$ and take the average $\mathbb{E}_\pi$ over all permutations of samples, which will inform us about the typical (average) behavior of the algorithm. After some algebra, we find

$$\begin{aligned}
&\mathbb{E}_\pi \mathsf{Main}_{n,r}(\boldsymbol{\theta}) + \eta\, \mathbb{E}_\pi \mathsf{Corr}_{n,r}(\boldsymbol{\theta}) \\
&\quad = \frac{1}{1-\beta} g_r + \eta \frac{\beta + o_n(1)}{(1-\beta)^3} \sum_i g_{ri} g_i \\
&\qquad + \eta \frac{\beta + o_n(1)}{2(1-\beta)^2(1+\beta)} \sum_i \frac{\Sigma_{ii}}{b}
\end{aligned}$$

or, in other words,

$$\begin{aligned}
&\mathbb{E}_\pi \mathsf{Main}_n(\boldsymbol{\theta}) + \eta \mathbb{E}_\pi \mathsf{Corr}_n(\boldsymbol{\theta}) \\
&\quad = \frac{1}{1-\beta} \nabla \Bigg(\mathcal{L} + \eta \frac{\beta + o_n(1)}{2(1-\beta)^2} \|\nabla \mathcal{L}\|^2 \\
&\qquad + \eta \frac{\beta + o_n(1)}{2(1-\beta)(1+\beta)} \frac{\operatorname{tr} \boldsymbol{\Sigma}}{b} \Bigg),
\end{aligned}$$

where $o_n(1)$ denotes terms that decay to zero as $n = m - 1 \to \infty$ (exponentially fast). This expression is non-random and is much easier to analyze. We see two correction terms:

- implicit regularization by memory $\eta \frac{\beta + o_n(1)}{2(1-\beta)^2} \|\nabla \mathcal{L}\|^2$ (present already in the full-batch case), and
- implicit regularization by stochasticity $\eta \frac{\beta + o_n(1)}{2(1-\beta)(1+\beta)} \frac{\operatorname{tr} \boldsymbol{\Sigma}}{b}$ (appearing as a result of mini-batch noise).

The first term implicitly penalizes the squared norm of the gradient, which is a first-order approximation of ℓ_2 sharpness (Foret et al., 2021): for small ρ,

$$\max_{\|\boldsymbol{\epsilon}\| \leq \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) - \mathcal{L}(\boldsymbol{\theta}) \approx \max_{\|\boldsymbol{\epsilon}\| \leq \rho} \nabla \mathcal{L}(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{\epsilon} = \rho \|\nabla \mathcal{L}(\boldsymbol{\theta})\|.$$

The second term implicitly penalizes gradient noise

$$\operatorname{tr} \boldsymbol{\Sigma}(\boldsymbol{\theta}) = \frac{1}{mb} \sum_{p=1}^{mb} \left\| \nabla_i (\ell_p - \mathcal{L})(\boldsymbol{\theta}) \right\|^2,$$

which is also related to flatness of the loss landscape, and is predictive of generalization (Jiang et al., 2020).

Therefore, penalizing both these terms is predictive of moving toward flatter regions of the loss landscape, and improving generalization. This is why we can classify them as "implicit regularization".

2.2. Summary

Our proposed approach is to interpret implicit biases of mini-batch versions of optimization algorithms with memory using the following three steps.

1. **Removing memory:** use Theorem 2.3 to approximate the algorithm having memory with a memoryless iteration.
2. **Calculating the average correction terms:** take expectation $\mathbb{E}_\pi \mathsf{Corr}_n(\boldsymbol{\theta})$ to remove dependence on mini-batch loss functions, and potentially make other simplifications without qualitatively changing the situation.
3. **Interpretation:** interpret the terms in the resulting expression, especially connecting to known sharpness/flatness or generalization measures (Jiang et al., 2020).

This simple, yet general framework offers a new approach to understanding how mini-batch noise influences (on average) the implicit bias of memory underlying complex optimization logarithms used for deep learning tasks.

3. Mini-Batch Noise In Adam

We now apply our proposed theoretical strategy to Adam, one of the most popular algorithms in deep learning. Its iteration can be written in terms of three variables $\{\boldsymbol{v}_t\}$, $\{\boldsymbol{m}_t\}$ and $\{\boldsymbol{\theta}_t\}$ as follows.

Definition 3.1 (Adam (Kingma & Ba, 2014)). *The* ***Adam*** *algorithm has numerical hyperparameters* $\epsilon, \beta_1, \beta_2 > 0$ *and the update rule*

$$\begin{aligned} m_{t+1,j} &= \beta_1 m_{t,j} + (1-\beta_1)\nabla_j \mathcal{L}_t(\boldsymbol{\theta}_t), \\ v_{t+1,j} &= \beta_2 v_{t,j} + (1-\beta_2)\nabla_j \mathcal{L}_t(\boldsymbol{\theta}_t)^2, \\ \theta_{t+1,j} &= \theta_{t,j} - \eta \frac{m_{t+1,j}/(1-\beta_1^{t+1})}{\sqrt{v_{t+1,j}/(1-\beta_2^{t+1}) + \epsilon}}, \end{aligned}$$

for $j \in [1 : \dim \boldsymbol{\theta}]$, *with initial conditions* $\boldsymbol{v}_0 = \mathbf{0} \in \mathbb{R}^{\dim \boldsymbol{\theta}}$, $\boldsymbol{m}_0 = \mathbf{0} \in \mathbb{R}^{\dim \boldsymbol{\theta}}$, *arbitrary* $\boldsymbol{\theta}_0 \in \mathbb{R}^{\dim \boldsymbol{\theta}}$.

This can be written in terms of just one variable $\{\boldsymbol{\theta}_t\}$ as

$$\theta_{t+1,j} = \theta_{t,j} - \eta \frac{\sum_{k=0}^{t} \mu_{t,k} \nabla_j \mathcal{L}_k(\boldsymbol{\theta}_k)}{\sqrt{\sum_{k=0}^{t} \nu_{t,k} |\nabla_j \mathcal{L}_k(\boldsymbol{\theta}_k)|^2 + \epsilon}},$$

where for $k \in [0 : t]$, $t \in \mathbb{Z}_{\geq 0}$

$$\mu_{t,k} := \frac{\beta_1^{t-k}(1-\beta_1)}{1-\beta_1^{t+1}}, \quad \nu_{t,k} := \frac{\beta_2^{t-k}(1-\beta_2)}{1-\beta_2^{t+1}}.$$

Recall from Definition 2.2 that d_k, $d_{k,i}$, $d_{k,ij}$ denote the mini-batch noise and its partial derivatives. Accordingly, we will use the notation $O(d^p)$ to mean "terms of order at least p in (derivatives of) noise". For example, all terms of the form $d_{k,ij} d_{k,i}$ are $O(d^2)$ and all terms of the form $d_{k,ijl} d_{k,ij} d_{k,l}$ are $O(d^3)$. In addition, we will use $o_\epsilon(1)$ to mean terms that go to zero as $\epsilon \to 0$ (pointwise in other quantities, that is, for all other quantities fixed), and $o_{n,\epsilon}(1)$ terms that go to zero as $n \to \infty$, $\epsilon \to 0$ (in this order).

The following simple assumption on the boundedness of derivatives is typical in the relevant literature (e. g. Kovachki & Stuart (2021); Ghosh et al. (2023); Cattaneo et al. (2024)).

Assumption 3.2 (Losses). *Assume that per-sample losses* $\{\ell_r(\boldsymbol{\theta})\}_{r=1}^{mb}$ *are three times continuously differentiable and uniformly bounded (by a constant not depending on* m, b, $\boldsymbol{\theta}$*) in the region of interest.*

3.1. Step 1: Removing Memory

Applying Theorem 2.3 under Assumption 3.2, we obtain that the iteration

$$\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_t - \eta \, \mathsf{Main}_t(\tilde{\boldsymbol{\theta}}_t) - \eta^2 \, \mathsf{Corr}_t(\tilde{\boldsymbol{\theta}}_t), \tag{5}$$

that does not have memory, is globally $O(\eta^2)$-close to $\{\boldsymbol{\theta}_t\}$ for $O(\eta^{-1})$ iterations, where the main term is defined as

$$\mathsf{Main}_{t,j}(\boldsymbol{\theta}) := \frac{\sum_{k=0}^{t} \mu_{t,k} \nabla_j \mathcal{L}_k(\boldsymbol{\theta})}{\sqrt{\sum_{k=0}^{t} \nu_{t,k} |\nabla_j \mathcal{L}_k(\boldsymbol{\theta})|^2 + \epsilon}},$$

and the correction term's definition is deferred to Appendix A.1 due to its length. These terms are still very complex and difficult to interpret. Thus, we proceed to the next step to simplify the analysis.

3.2. Step 2: Calculating the Average Correction Terms

The following proposition arises from expanding $\mathsf{Corr}_{t,j}(\boldsymbol{\theta})$ up to degree-2 monomials in noise derivatives and then calculating the average of the result with respect to permutations of samples. In the gradient-dominated regime, as opposed to noise-dominated regime, such an expansion is sufficient to see the qualitative effect of noise.

Proposition 3.3. *Suppose Assumption 3.2 holds. To avoid division by zero, assume also that no component of full-batch gradient* $\boldsymbol{g}$ *is exactly zero (at a current fixed point* $\boldsymbol{\theta}$ *which we omit). Then, the expectation* $\mathbb{E}_\pi$ *of the correction term with respect to the uniform law on all permutations* $[1 : mb] \to [1 : mb]$ *satisfies*

$$\begin{aligned} |g_j| \mathbb{E}_\pi \mathsf{Corr}_{n,j} = \mathsf{FB}(\beta_1, \beta_2) & \\ + \mathsf{MBN}_1(\beta_1, \beta_2) + \mathsf{MBN}_2(\beta_1, \beta_2) & \\ + \mathsf{MBN}_3(\beta_1, \beta_2) + \mathsf{MBN}_4(\beta_1, \beta_2) & \\ + \mathsf{MBN}_5(\beta_1, \beta_2) + O(d^3) + o_{n,\epsilon}(b^{-1}), & \end{aligned} \tag{6}$$

where the full-batch correction is given by

$$\mathsf{FB}(\beta_1, \beta_2) := \left(\frac{\beta_1}{1-\beta_1} - \frac{\beta_2}{1-\beta_2} \right) \nabla_j \|\boldsymbol{g}\|_1,$$

and five mini-batch noise corrections are given by

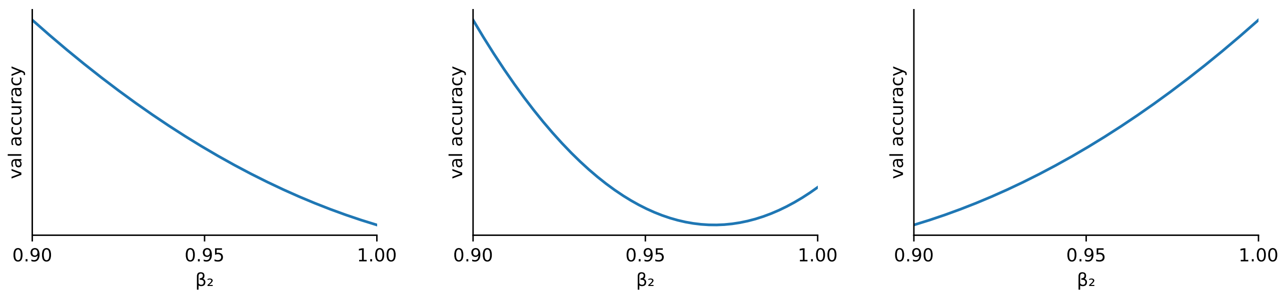
$$\begin{aligned} \mathsf{MBN}_1(\beta_1, \beta_2) &:= C_1(\beta_1, \beta_2) \nabla_j \|\boldsymbol{g}\|_1 \frac{m-1}{mb-1} \frac{\Sigma_{jj}}{g_j^2}, \\ \mathsf{MBN}_2(\beta_1, \beta_2) &:= C_2(\beta_1, \beta_2) \sum_i (\nabla_j |g_i|) \frac{m-1}{mb-1} \frac{\Sigma_{ii}}{g_i^2}, \\ \mathsf{MBN}_3(\beta_1, \beta_2) &:= C_3(\beta_1, \beta_2) \frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi d_{0,ij} d_{0,j}, \\ \mathsf{MBN}_4(\beta_1, \beta_2) &:= C_4(\beta_1, \beta_2) \sum_i \frac{1}{|g_i|} \frac{m-1}{2(mb-1)} \nabla_j \Sigma_{ii}, \\ \mathsf{MBN}_5(\beta_1, \beta_2) &:= C_5(\beta_1, \beta_2) \frac{s_j}{|g_j|} \sum_i \frac{g_{ij}}{|g_i|} \frac{m-1}{mb-1} \Sigma_{ij}, \end{aligned}$$

with the values of $\{C_i(\beta_1, \beta_2)\}_{i=1}^5$ *deferred to Appendix A.2.*

We provide the proof in Appendix A.4. For very small batches, these degree-2 monomials may not be enough to achieve an accurate approximation. However, our predictions will be directional, and we will see empirically that there is no phase shift in the high-noise regime, making the (likely infeasible) analysis of higher-order terms hardly useful.

3.3. Step 3: Interpretation

We need to analyze each term in the right-hand side of Equation (6). Since the analysis can be different for different choices of hyperparameters, we will confine ourselves to



(a) $b \ll \frac{2N\mathcal{B}_{\text{simple}}}{N+2\mathcal{B}_{\text{simple}}-1}$ *(b)* $b \sim \frac{2N\mathcal{B}_{\text{simple}}}{N+2\mathcal{B}_{\text{simple}}-1}$ *(c)* $b \gg \frac{2N\mathcal{B}_{\text{simple}}}{N+2\mathcal{B}_{\text{simple}}-1}$

Figure 1. Schematic illustration: validation accuracy vs. β_2 across regimes.

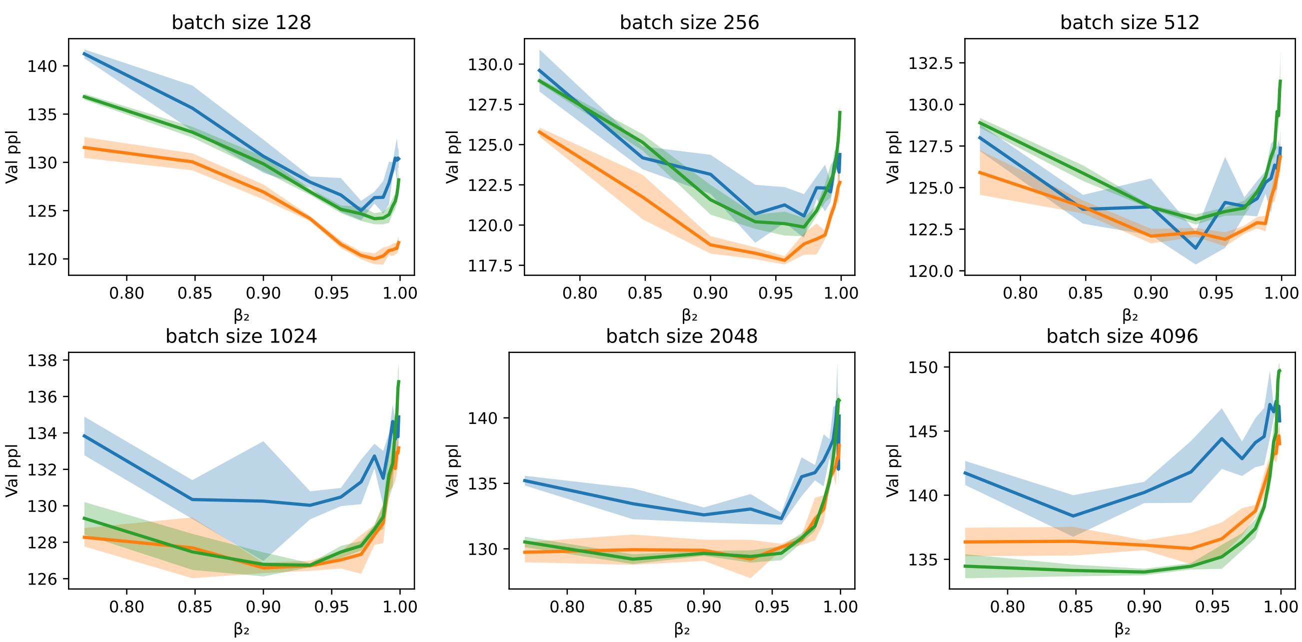


Figure 2. Minimal validation perplexity (before overfitting) of Transformer-XL trained with Adam on WikiText-2 with different batch sizes, learning rates $\{10^{-3}, 10^{-3.5}, 10^{-4}\}$, $\beta_1 = 0.9$, $\epsilon = 10^{-6}$ (averaged over three iterations).

situations where one of the betas is fixed at a reasonable value and we are seeking the best value of the other beta, based only on loss landscape flatness metrics (as discussed in the introduction). Specifically, we will focus on two settings: first, how to set β_2 if β_1 is fixed (at, say, 0.9 or 0.99); second, how to set β_1 if β_2 is fixed at the "default" value 0.999.

How to set β_2 if β_1 is fixed We will start by focusing on the setting where β_1 is set at its default value 0.9 and β_2 is in the interval $[0.9, 1)$ (aligning well with common practice):

$$\beta_1 = 0.9, \quad \text{seeking best } \beta_2 \in [0.9, 1).$$

It is the simplest to deal with the terms containing $\mathsf{MBN}_4(\beta_1, \beta_2)$ and $\mathsf{MBN}_5(\beta_1, \beta_2)$: Lemma A.1 implies that they can be neglected because they are small compared to other terms. In addition, we argue in Appendix A.3.1 that the term containing $C_3(\beta_1, \beta_2)$ is neutral for generalization.

We are left with the sum of three terms: $\mathsf{FB}(\beta_1, \beta_2)$, $\mathsf{MBN}_1(\beta_1, \beta_2)$ and $\mathsf{MBN}_2(\beta_1, \beta_2)$. The first term $\mathsf{FB}(\beta_1, \beta_2)$ anti-penalizes (if $\beta_1 < \beta_2$) the 1-norm of the gradient, which is a first-order approximation of ℓ_∞-sharpness: for small ρ, one has

$$\max_{\|\boldsymbol{\epsilon}\|_\infty \le \rho} \mathcal{L}(\boldsymbol{\theta} + \boldsymbol{\epsilon}) - \mathcal{L}(\boldsymbol{\theta}) \approx \max_{\|\boldsymbol{\epsilon}\|_\infty \le \rho} \nabla \mathcal{L}(\boldsymbol{\theta})^\mathsf{T} \boldsymbol{\epsilon} = \rho \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_1.$$

Thus, we can refer to this term as anti-regularization, same as in the setting with zero noise (Cattaneo et al., 2024). The term containing $C_1(\beta_1, \beta_2)$ (it can be checked that it

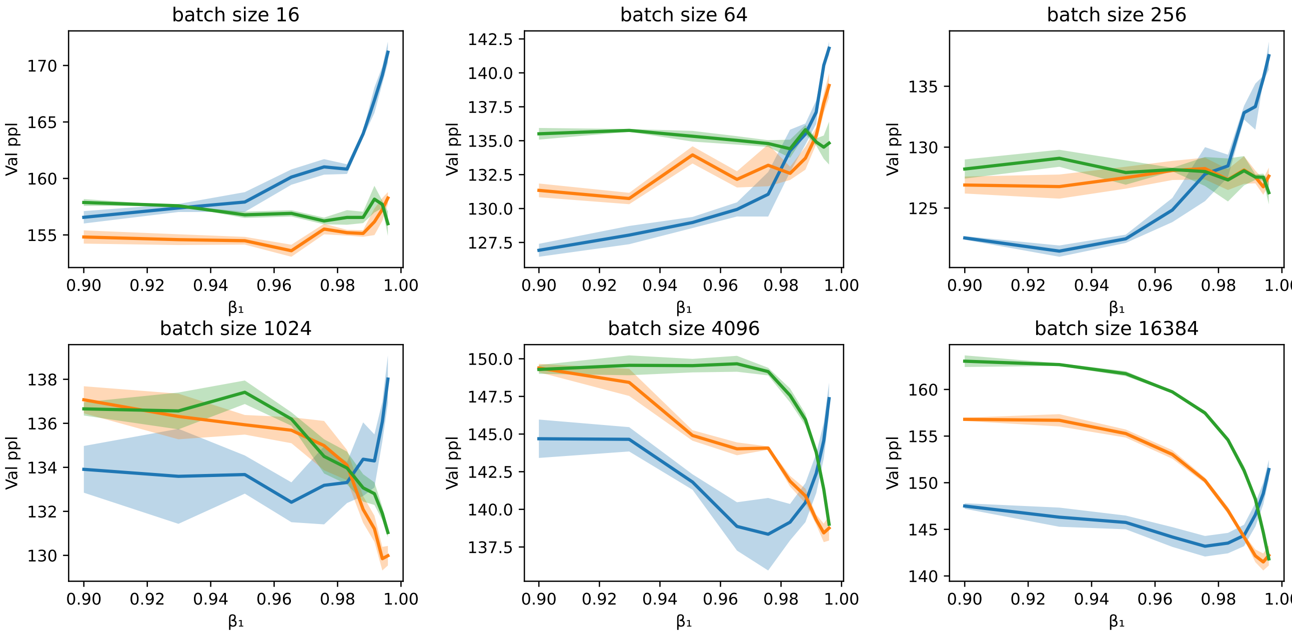


Figure 3. Minimal validation perplexity (before overfitting) of Transformer-XL trained with Adam on WikiText-2 with different batch sizes, learning rates $\{10^{-3.5}, 10^{-4}, 10^{-4.5}\}$, $\beta_2 = 0.999$, $\epsilon = 10^{-6}$ (averaged over three iterations).

is positive in our setting) provides regularization: it also penalizes the ℓ_1 gradient norm although the magnitude of this penalization in each component j depends on the per-component noise-to-signal ratio Σ_{jj}/g_j^2.

The term containing $C_2(\beta_1, \beta_2)$ is more complicated, so we resort to the following simplification. Suppose that at a current point $\boldsymbol{\theta}$ the per-component noise-to-signal scale Σ_{ii}/g_i^2 can be replaced with one constant describing its typical value, the "simple noise scale" $\mathcal{B}_{\text{simple}}$ from (McCandlish et al., 2018), given by

$$\mathcal{B}_{\text{simple}} := \frac{\operatorname{tr} \boldsymbol{\Sigma}}{\sum_j g_j^2} = \frac{\operatorname{tr} \boldsymbol{\Sigma}}{\|\boldsymbol{g}\|^2}.$$

Of course, we lose the per-component variance of Σ_{ii}/g_i^2 but this replacement should not qualitatively change conclusions about the effect of noise. After such a replacement, the terms $\mathsf{MBN}_1(\beta_1, \beta_2)$ and $\mathsf{MBN}_2(\beta_1, \beta_2)$ look the same up to coefficients:

$$C_1(\beta_1, \beta_2) \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \frac{m-1}{mb-1} \mathcal{B}_{\text{simple}},$$
$$C_2(\beta_1, \beta_2) \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \frac{m-1}{mb-1} \mathcal{B}_{\text{simple}}.$$

We can conclude that the aggregated effect of the terms in the right-hand side of (6) is providing implicit (anti-)penalization of an approximate non-adaptive sharpness measure, with magnitude $C_{\text{total}}\bigl(\beta_1, \beta_2, \frac{m-1}{mb-1}\mathcal{B}_{\text{simple}}\bigr)$, where

$$\begin{aligned} C_{\text{total}}(\beta_1, \beta_2, \lambda) &:= \frac{\beta_1}{1-\beta_1} - \frac{\beta_2}{1-\beta_2} \\ &\quad + \{C_1(\beta_1, \beta_2) + C_2(\beta_1, \beta_2)\}\lambda. \end{aligned} \tag{7}$$

If $C_{\text{total}}(\beta_1, \beta_2, \frac{m-1}{mb-1}\mathcal{B}_{\text{simple}}) > 0$, this can be interpreted as regularization, otherwise as anti-regularization.

It remains to find out what is the nature of the dependence of (7) on λ. The following fact is easy to check.

Lemma 3.4. *If $\lambda \geq 0.5082$, the function $C_{total}(0.9, \beta_2, \lambda)$ is strictly increasing in $\beta_2 \in [0.9, 1)$. If $0 < \lambda < 0.494$, it is strictly decreasing in $\beta_2 \in [0.9, 1)$.*

We have obtained the following prediction: for fixed $\beta_1 = 0.9$, on the interval $\beta_2 \in [0.9, 1)$ the quantity (recall that $N = mb$ is the number of samples)

$$\frac{N/b - 1}{N - 1} \mathcal{B}_{\text{simple}} \tag{8}$$

reverts the dependence of the approximate "regularization magnitude" (7) on β_2: if it is significantly less than 0.5, (7) is decreasing in β_2, and if it is significantly higher than 0.5, (7) is increasing in β_2. Theoretically, the transition happens very quickly around the point where the batch size is $\frac{2N\mathcal{B}_{\text{simple}}}{N + 2\mathcal{B}_{\text{simple}} - 1}$ (although simplifications that we made likely make the theoretical transition quicker than it is in practice). This is schematically illustrated in Figure 1.

If β_1 is fixed at 0.99 rather than 0.9, the qualitative picture is the same except the transition between increasing and decreasing $C_{\text{total}}(0.99, \beta_2, \lambda)$ is less sharp. The relevant lemma is below.

Lemma 3.5. *If $\lambda \geq 2.685$, the function $C_{\text{total}}(0.99, \beta_2, \lambda)$ is strictly increasing in $\beta_2 \in [0.9, 1)$. If $0.5 < \lambda < 2.684$, it is strictly convex in β_2 with a unique minimizer inside $(0.9, 1)$. If $0 < \lambda < 0.499$, it is strictly decreasing in $\beta_2 \in [0.9, 1)$.*

In summary, for both cases ($\beta_1 \in \{0.9, 0.99\}$): if (8) is much higher than 0.5 (batch size small enough), the coefficient is increasing in β_2 at least near 1 (the higher β_2, the higher the penalization of sharpness, often leading to better generalization), and it is best to take β_2 as high as possible while keeping the training stable (e. g. 0.999); if (8) is significantly less than 0.5 (batch size large enough), the coefficient is decreasing in β_2 (the higher β_2, the lower the penalization of sharpness, often leading to worse generalization), and it is best to take β_2 as low as possible while keeping the training convergent (e. g. β_2 equal to β_1).

How to set β_1 if β_2 is fixed Next, we will consider the one-dimensional sweep of a different kind, where β_2 is fixed at some (say, default) value and β_1 varies. This sweep is less common in practice but it still useful for understanding the full picture:

$$\beta_2 = 0.999, \quad \text{seeking best } \beta_1 \in [0.9, 1).$$

The following lemma describes this situation.

Lemma 3.6. *If $\lambda \geq 1.002$, the function $C_{\text{total}}(\beta_1, 0.999, \lambda)$ is strictly decreasing in $\beta_1 \in [0.9, 1)$. If $0 < \lambda < 0.995$, it is strictly increasing in $\beta_1 \in [0.9, 1)$.*

In this case, we can conclude: if (8) is much higher than one (batch size small enough), the approximate "regularization magnitude" (7) is decreasing in β_1 (the higher β_1, the weaker the bias towards flatter regions, the worse generalization), and it is best to take β_1 is low as possible while keeping the training stable (e. g. 0.9); if (8) is much less than one (batch size large enough), the coefficient is increasing in β_1 (the higher β_1, the higher penalization, the better generalization), and it is best to take β_1 as high as possible while keeping the training convergent (e. g. $\beta_1 = \beta_2$).

Conclusion At this point, we found that in the high-noise (small-batch) regime the choice of default hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$ makes sense. In the low-noise (large-batch) regime, our theoretically motivated suggestion is to take β_1 roughly equal to β_2, but we have not yet found out at which value. The following lemma aims to answer this question.

Lemma 3.7. *The function $C_{\text{total}}(\beta, \beta, \lambda)$ is strictly increasing in $\beta \in [0.5, 1)$ for any $\lambda > 0$.*

Therefore, the value at which $\beta_1 = \beta_2$ should be taken is likely quite close to 1.

After this (partly mathematical, partly heuristic) theoretical analysis, we can arrive at the following concluding rule of thumb.

If (8) is much higher than 1 (batch size smaller than $\frac{N\mathcal{B}_{\text{simple}}}{N+\mathcal{B}_{\text{simple}}-1}$), take β_1 as small and β_2 as large as possible while keeping the training stable enough (e. g., the default values $\beta_1 = 0.9$, $\beta_2 = 0.999$ are a reasonable first choice).

If (8) is significantly less than 0.5 (batch size larger than $\frac{2N\mathcal{B}_{\text{simple}}}{N+2\mathcal{B}_{\text{simple}}-1}$), take $\beta_1 = \beta_2$ as high as possible while keeping the training convergent (e. g., $\beta_1 = \beta_2 = 0.999$ is a reasonable first choice).

The analysis above provides a novel theoretical perspective on the dependence of the best choices of β_1 and β_2 on the batch size. Of course, we made some bold simplifications, justifying only directional rather than precisely quantitative predictions. In particular, $\mathcal{B}_{\text{simple}}$ technically both depends on the hyperparameters of the training run, and varies during training. However, only the scale of this quantity matters, and it is not difficult to estimate (McCandlish et al., 2018). Thus, the concluding rule of thumb can guide practical choices of β_1 and β_2, avoiding very large grids.

4. Experiments

We train Transformer-XL (Dai et al., 2019) on WikiText-2 (Merity et al., 2017) with different batch sizes and learning rates. The implementation follows Dai et al. (2019); Zhang et al. (2020) as in Kunstner et al. (2023). We fix the default value $\beta_1 = 0.9$, and sweep β_2. The model quickly overfits as training loss continues to go to zero. Therefore, we train for sufficiently many epochs to let the model overfit, and plot the minimal validation perplexity achieved, depending on β_2.

The results are shown in Figure 2. We observe that in small-batch Adam, larger β_2 mostly helps the model generalize better (decreases minimal validation perplexity), and this behavior smoothly transitions into the opposite as the batch size increases. Note also that the improvements from tuning β_2 are quite substantial (getting up to 13.01%; see Appendix B for additional details).

Similarly, we sweep β_1 fixing $\beta_2 = 0.999$, and observe in Figure 3 that increasing the batch size largely reverts the dependence of the optimal validation perplexity on β_1 (for small batch sizes, taking $\beta_1 = 0.9$ is near-optimal, whereas for large batch sizes it can be highly suboptimal and taking β_1 much closer to β_2 is better from the perspective of generalization).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. Specifically, it sheds new light on the role of hyperparameters underlying optimization algorithms commonly used in deep learning tasks. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Andriushchenko, M., Croce, F., Müller, M., Hein, M., and Flammarion, N. A modern look at the relationship between sharpness and generalization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/andriushchenko23a.html.

Andriushchenko, M., D'Angelo, F., Varre, A., and Flammarion, N. Why do we need weight decay in modern deep learning?, 2024. URL https://openreview.net/forum?id=RKh7DI23tz.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Arora, S., Li, Z., and Panigrahi, A. Understanding gradient descent on the edge of stability in deep learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/arora22a.html.

Barrett, D. and Dherin, B. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3q5IqUrkcF.

Beneventano, P. On the trajectories of sgd without replacement. *arXiv preprint arXiv:2312.16143*, 2023.

Biderman, S., Schoelkopf, H., Anthony, Q. G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M. A., Purohit, S., Prashanth, U. S., Raff, E., Skowron, A., Sutawika, L., and Van Der Wal, O. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cattaneo, M. D. and Shigida, B. How memory in optimization algorithms implicitly modifies the loss. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=2qd4lpXz7u.

Cattaneo, M. D. and Shigida, B. Modified loss of momentum gradient descent: Fine-grained analysis. *arXiv preprint arXiv:2509.08483*, 2025b.

Cattaneo, M. D., Klusowski, J. M., and Shigida, B. On the implicit bias of Adam. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5862–5906. PMLR, 2024. URL https://proceedings.mlr.press/v235/cattaneo24a.html.

Compagnoni, E. M., Liu, T., Islamov, R., Proske, F. N., Orvieto, A., and Lucchi, A. Adaptive methods through the lens of SDEs: Theoretical insights on the role of noise. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ww3CLRhF1v.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:57759363.

Damian, A., Ma, T., and Lee, J. D. Label noise sgd provably prefers flat global minimizers. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 27449–27461. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/e6af401c28c1790eaef7d55c92ab6ab6-Paper.pdf.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1019–1028. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/dinh17b.html.

Du, J., Zhou, D., Feng, J., Tan, V., and Zhou, J. T. Sharpness-aware training for free. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 23439–23451. Curran Associates, Inc., 2022.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Farazmand, M. Multiscale analysis of accelerated gradient methods. *SIAM Journal on Optimization*, 30(3):2337–2354, 2020.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM.

Ghosh, A., Lyu, H., Zhang, X., and Wang, R. Implicit regularization in heavy-ball momentum accelerated stochastic gradient descent. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ZzdBhtEH9yB.

Gunasekar, S., Lee, J., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018b.

Hochreiter, S. and Schmidhuber, J. Simplifying neural nets by discovering flat minima. In Tesauro, G., Touretzky, D., and Leen, T. (eds.), *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL https://proceedings.neurips.cc/paper_files/paper/1994/file/01882513d5fa7c329e940dda99b12147-Paper.pdf.

Ji, Z. and Telgarsky, M. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018a.

Ji, Z. and Telgarsky, M. Risk and parameter convergence of logistic regression. *arXiv preprint arXiv:1803.07300*, 2018b.

Ji, Z. and Telgarsky, M. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pp. 1772–1798. PMLR, 2019.

Ji, Z. and Telgarsky, M. Directional convergence and alignment in deep learning. *Advances in Neural Information Processing Systems*, 33:17176–17186, 2020.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=H1oyRlYgg.

Kim, K., Kotha, S., Liang, P., and Hashimoto, T. Pre-training under infinite compute, 2025. URL https://arxiv.org/abs/2509.14786.

Kim, M., Li, D., Hu, S. X., and Hospedales, T. Fisher SAM: Information geometry and sharpness aware minimisation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11148–11161. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kim22f.html.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kobayashi, S., Akram, Y., and von Oswald, J. Weight decay induces low-rank attention layers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=oDeqjIM9Sk.

Kovachki, N. B. and Stuart, A. M. Continuous time analysis of momentum methods. *Journal of Machine Learning Research*, 22(17), 2021.

Kunstner, F., Chen, J., Lavington, J. W., and Schmidt, M. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=a65YK0cqH8g.

Kwon, J., Kim, J., Park, H., and Choi, I. K. Asam: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kwon21b.html.

Li, B. and Giannakis, G. B. Enhancing sharpness-aware optimization through variance suppression. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Sf3t6Bth4P.

Li, T., Zhou, P., He, Z., Cheng, X., and Huang, X. Friendly sharpness-aware minimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.1109/CVPR52733.2024.00538.

Liu, Y., Mai, S., Chen, X., Hsieh, C.-J., and You, Y. Towards efficient and scalable sharpness-aware minimization. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. doi: 10.1109/CVPR52688.2022.01204.

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

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Ma, C., Wu, L., and E, W. A qualitative study of the dynamic behavior for adaptive gradient algorithms. In *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, volume 145 of *Proceedings of Machine Learning Research*, pp. 671–692. PMLR, 2022. URL https://proceedings.mlr.press/v145/ma22a.html.

Malladi, S., Lyu, K., Panigrahi, A., and Arora, S. On the SDEs and scaling rules for adaptive gradient algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pp. 7697–7711. Curran Associates, Inc., 2022.

Marek, M., Lotfi, S., Somasundaram, A., Wilson, A. G., and Goldblum, M. Small batch size training for language models: When vanilla SGD works, and why gradient accumulation is wasteful. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=52Ehpe0Lu5.

McCandlish, S., Kaplan, J., Amodei, D., and Team, O. D. An empirical model of large-batch training, 2018. URL https://arxiv.org/abs/1812.06162.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Miyagawa, T. Toward equation of motion for deep neural networks: Continuous-time gradient descent and discretization error analysis. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=qq84D17BPu.

Nacson, M. S., Gunasekar, S., Lee, J., Srebro, N., and Soudry, D. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, pp. 4683–4692. PMLR, 2019a.

Nacson, M. S., Lee, J., Gunasekar, S., Savarese, P. H. P., Srebro, N., and Soudry, D. Convergence of gradient descent on separable data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3420–3428. PMLR, 2019b.

Nacson, M. S., Srebro, N., and Soudry, D. Stochastic gradient descent on separable data: Exact convergence with a fixed learning rate. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3051–3059. PMLR, 2019c.

Orvieto, A. and Gower, R. M. In search of Adam's secret sauce. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=CH72XyZs4y.

Pagliardini, M., Ablin, P., and Grangier, D. The adEMAMix optimizer: Better, faster, older. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jj7b3p5kLY.

Porian, T., Wortsman, M., Jitsev, J., Schmidt, L., and Carmon, Y. Resolving discrepancies in compute-optimal scaling of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=4fSSqpk1sM.

Qian, Q. and Qian, X. The implicit bias of adagrad on separable data. *Advances in Neural Information Processing Systems*, 32, 2019.

Rangamani, A., Nguyen, N. H., Kumar, A., Phan, D., Chin, S. P., and Tran, T. D. A scale invariant measure of flatness for deep network minima. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1680–1684, 2021. doi: 10.1109/ICASSP39728.2021.9413771.

Rosca, M., Wu, Y., Qin, C., and Dherin, B. On a continuous time model of gradient descent dynamics and instability in deep learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=EYrRzKPinA.

Schmidt, R. M., Schneider, F., and Hennig, P. Descending through a crowded valley - benchmarking deep learning optimizers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9367–9376. PMLR, 2021. URL https://proceedings.mlr.press/v139/schmidt21a.html.

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

Sivaprasad, P. T., Mai, F., Vogels, T., Jaggi, M., and Fleuret, F. Optimizer benchmarking needs to account for hyperparameter tuning. In *International conference on machine learning*, pp. 9036–9045. PMLR, 2020.

Smith, S. L., Dherin, B., Barrett, D., and De, S. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Tahmasebi, B., Soleymani, A., Bahri, D., Jegelka, S., and Jaillet, P. A universal class of sharpness-aware minimization algorithms. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9Ub6nLqdMo.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tsuzuku, Y., Sato, I., and Sugiyama, M. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using PAC-Bayesian analysis. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9636–9647. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/tsuzuku20a.html.

Villalobos, P., Ho, A., Sevilla, J., Besiroglu, T., Heim, L., and Hobbhahn, M. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Wang, B., Meng, Q., Chen, W., and Liu, T.-Y. The implicit bias for adaptive optimization algorithms on homogeneous neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 7 2021. URL https://proceedings.mlr.press/v139/wang21q.html.

Wang, B., Meng, Q., Zhang, H., Sun, R., Chen, W., Ma, Z.-M., and Liu, T.-Y. Does momentum change the implicit regularization on separable data? *Advances in Neural Information Processing Systems*, 35:26764–26776, 2022.

Wen, K., Hall, D., Ma, T., and Liang, P. Fantastic pretraining optimizers and where to find them, 2025. URL https://arxiv.org/abs/2509.02046.

Xiao, L. Rethinking conventional wisdom in machine learning: From generalization to scaling, 2025. URL https://arxiv.org/abs/2409.15156.

Xie, S. and Li, Z. Implicit bias of AdamW: ℓ_∞-norm constrained optimization. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54488–54510. PMLR, 2024. URL https://proceedings.mlr.press/v235/xie24e.html.

Xie, W., Pethick, T., and Cevher, V. SAMPa: Sharpness-aware minimization parallelized. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=IGn0ktYDwV.

Xie, Z., Wang, X., Zhang, H., Sato, I., and Sugiyama, M. Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/xie22d.html.

Yi, M., Meng, Q., Chen, W., Ma, Z.-m., and Liu, T.-Y. Positively scale-invariant flatness of relu neural networks. *arXiv preprint arXiv:1903.02237*, 2019. URL https://arxiv.org/abs/1903.02237.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Zhang, G., Wang, C., Xu, B., and Grosse, R. Three mechanisms of weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1lz-3Rct7.

Zhang, H., Morwani, D., Vyas, N., Wu, J., Zou, D., Ghai, U., Foster, D., and Kakade, S. M. How does critical batch size scale in pre-training? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=JCiF03qnmi.

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S., Kumar, S., and Sra, S. Why are adaptive methods good for attention models? *Advances in Neural Information Processing Systems*, 33:15383–15393, 2020.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Zhao, R., Morwani, D., Brandfonbrener, D., Vyas, N., and Kakade, S. M. Deconstructing what makes a good optimizer for autoregressive language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=zfeso8ceqr.

Zheng, Y., Zhang, R., and Mao, Y. Regularizing neural networks via adversarial model perturbation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. doi: 10.1109/CVPR46437.2021.00806.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H., and E, W. Towards theoretically understanding why sgd generalizes better than adam in deep learning. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3f27a324736617f20abbf2ffd806f6d-Paper.pdf.

Zhou, P., Xie, X., Lin, Z., and Yan, S. Towards understanding convergence and generalization of AdamW. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(9):6486–6493, 2024. doi: 10.1109/TPAMI.2024.3382294.

Zhuang, Z., Liu, M., Cutkosky, A., and Orabona, F. Understanding adamw through proximal methods and scale-freeness, 2022. URL https://openreview.net/forum?id=GU11Lbci5J.

A. Theoretical Analysis: Details

A.1. Omitted Expression

In Equation (5), the correction term $\mathsf{Corr}_{t,j}(\boldsymbol{\theta})$ is given by

$$\mathsf{Corr}_{t,j}(\boldsymbol{\theta}) := \frac{L_{t,j}(\boldsymbol{\theta})}{R_{t,j}(\boldsymbol{\theta})} - \frac{M_{t,j}(\boldsymbol{\theta})P_{t,j}(\boldsymbol{\theta})}{R_{t,j}(\boldsymbol{\theta})^3}, \tag{9}$$

with the following auxiliary notations used:

$$\begin{aligned}
M_{t,j}(\boldsymbol{\theta}) &:= \sum_{k=0}^{t} \mu_{t,k} \nabla_j \mathcal{L}_k(\boldsymbol{\theta}), \\
R_{t,j}(\boldsymbol{\theta}) &:= \sqrt{\sum_{k=0}^{t} \nu_{t,k} |\nabla_j \mathcal{L}_k(\boldsymbol{\theta})|^2 + \epsilon}, \\
L_{t,j}(\boldsymbol{\theta}) &:= \sum_{k=0}^{t-1} \mu_{t,k} \sum_{i=1}^{\dim \boldsymbol{\theta}} \nabla_{ij} \mathcal{L}_k(\boldsymbol{\theta}) \sum_{l=k}^{t-1} \frac{M_{l,i}(\boldsymbol{\theta})}{R_{l,i}(\boldsymbol{\theta})}, \\
P_{t,j}(\boldsymbol{\theta}) &:= \sum_{k=0}^{t-1} \nu_{t,k} \nabla_j \mathcal{L}_k(\boldsymbol{\theta}) \sum_{i=1}^{\dim \boldsymbol{\theta}} \nabla_{ij} \mathcal{L}_k(\boldsymbol{\theta}) \sum_{l=k}^{t-1} \frac{M_{l,i}(\boldsymbol{\theta})}{R_{l,i}(\boldsymbol{\theta})}.
\end{aligned}$$

A.2. Values of Constants

The values of $\{C_i(\beta_1, \beta_2)\}_{i=1}^5$ in are given by

$$\begin{aligned}
C_1(\beta_1, \beta_2) :=& \frac{1-\beta_1^2}{\beta_1(1-\beta_1\beta_2)} + \frac{(1-\beta_1)^2}{\beta_1(1-\beta_1\beta_2)^2} + \frac{3(1+\beta_1)}{2(1-\beta_1)(1+\beta_2)} \\
& - \frac{2}{\beta_1(1-\beta_1)} + \frac{3}{2-2\beta_2} + \frac{3}{(1+\beta_2)^2} - 2, \\
C_2(\beta_1, \beta_2) :=& \frac{(\beta_1-\beta_2)(\beta_1\beta_2^2 - \beta_1\beta_2 + \beta_1 + \beta_2^2 - 2\beta_2)}{(1-\beta_1)(1-\beta_2)(1+\beta_2)(1-\beta_1\beta_2)}, \\
C_3(\beta_1, \beta_2) :=& \left[(1-\beta_2)(1+\beta_2)^2(1-\beta_1\beta_2)^2\right]^{-1} \big\{-2\beta_1^2\beta_2^5 + (\beta_1^2 + 8\beta_1)\beta_2^4 + (-5\beta_1^2 + 2\beta_1 - 4)\beta_2^3 \\
& + (2\beta_1^2 - 2\beta_1 - 1)\beta_2 - 2\beta_1\beta_2^5 + (2\beta_1 + 1)\beta_2^2\big\}, \\
C_4(\beta_1, \beta_2) :=& -\frac{(\beta_1-\beta_2)^2}{(1+\beta_1)(1+\beta_2)(1-\beta_1\beta_2)}, \\
C_5(\beta_1, \beta_2) :=& \frac{\beta_2(\beta_2-\beta_1)(2\beta_2 - 3\beta_1 - 1)}{(1+\beta_1)(1+\beta_2)^2(1-\beta_1\beta_2)}.
\end{aligned}$$

A.3. Simplification of Terms

First, the following lemma implies that the terms containing $C_4(\beta_1, \beta_2)$ and $C_5(\beta_1, \beta_2)$ are small compared to other terms and can therefore be neglected.

Lemma A.1 ($C_4(\beta_1, \beta_2)$ and $C_5(\beta_1, \beta_2)$ are small). *The following bounds hold:*

$$\begin{aligned}
\sup_{\beta_2 \in [0.9, 1)} |C_4(0.9, \beta_2)/C_1(0.9, \beta_2)| &< 3 \times 10^{-4}, \\
\sup_{\beta_2 \in [0.9, 1)} |C_5(0.9, \beta_2)/C_1(0.9, \beta_2)| &< 4 \times 10^{-3}, \\
\sup_{\beta_2 \in [0.9, 1)} |C_4(0.99, \beta_2)/C_2(0.99, \beta_2)| &< 10^{-3}, \\
\sup_{\beta_2 \in [0.9, 1)} |C_5(0.99, \beta_2)/C_2(0.99, \beta_2)| &< 6 \times 10^{-3},
\end{aligned}$$

$$\sup_{\beta_1 \in [0.9,1)} |C_4(\beta_1, 0.999)/C_2(\beta_1, 0.999)| < 6 \times 10^{-5},$$
$$\sup_{\beta_1 \in [0.9,1)} |C_5(\beta_1, 0.999)/C_2(\beta_1, 0.999)| < 5 \times 10^{-4}.$$

Proof. Direct numerical optimization verifies this result. □

A.3.1. THE TERM WITH $C_3(\beta_1, \beta_2)$ IS NEUTRAL FOR GENERALIZATION

We assert that $C_3(\beta_1, \beta_2) \frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi d_{0,ij} d_{0,j}$ provides neither regularization nor anti-regularization, i. e. is neutral. We start by rewriting

$$\frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi d_{0,ij} d_{0,j} = \frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi (\nabla_{ij} \mathcal{L}_0 - \nabla_{ij} \mathcal{L}) d_{0,j} = \frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi \nabla_{ij} \mathcal{L}_0 d_{0,j}.$$

In the gradient-dominated (as opposed to noise-dominated) regime, the sign of a mini-batch gradient component is typically the same as the sign of the full-batch gradient component: $\operatorname{sign} \nabla_i \mathcal{L}_0 \approx \operatorname{sign} \nabla_i \mathcal{L} = s_i$. Then

$$\frac{s_j}{|g_j|} \sum_i s_i \mathbb{E}_\pi \nabla_{ij} \mathcal{L}_0 d_{0,j} \approx \frac{1}{g_j} \sum_i \mathbb{E}_\pi (\nabla_j |\nabla_i \mathcal{L}_0|) d_{0,j} = \mathbb{E}_\pi \frac{d_{0,j}}{g_j} \nabla_j \sum_i |\nabla_i \mathcal{L}_0| = \mathbb{E}_\pi \frac{d_{0,j}}{g_j} \nabla_j \|\nabla \mathcal{L}_0\|_1.$$

The factor $d_{0,j}/g_j$ can be equally likely positive or negative, so there is no preferred choice whether the 1-norm of the gradient is penalized or anti-penalized. Since our interest is the sign of (anti-)penalization, we can interpret this term as neutral for our purposes.

A.4. Proof of Proposition 3.3

The plan is to first expand $\mathsf{Corr}_{n,j}(\boldsymbol{\theta})$ up to degree-2 monomials in noise derivatives (that is, up to $O(d^2)$) and then calculate $\mathbb{E}_\pi[\cdot]$ of the result.

A.4.1. EXPANDING THE CORRECTION UP TO QUADRATIC TERMS IN NOISE

Proposition A.2 (Expansion of the correction up to quadratic terms in noise)**.** *The additive components $L_{n,j}(\boldsymbol{\theta})/R_{n,j}(\boldsymbol{\theta})$ and $M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})/R_{n,j}(\boldsymbol{\theta})^3$ of the correction defined in Equation (9) admit the following formal expansion up to $O(d^2)$ and vanishing quantities as $\epsilon \to 0$:*

$$\begin{aligned} L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} = & \left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_0 + \left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_1 + \left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_2 \\ & + O(d^3), \end{aligned}$$

where[1]

$$\begin{aligned} &\left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_0 = \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \sum_{k=0}^{n-1} \mu_{n,k} (n-k)(1 + o_\epsilon(1)), \\ &\left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_1 = \text{[skipped]}, \\ &\left[L_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \right]_2 \\ &\quad = \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \mu_{n,k} \frac{3\nu_{l,p}^2 - \nu_{l,p} - 2\mu_{l,p}\nu_{l,p}}{2} d_{p,i}^2 (1 + o_\epsilon(1)) \\ &\qquad + \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \mu_{n,k} (\mu_{l,p} - \nu_{l,p}) d_{k,ij} d_{p,i} (1 + o_\epsilon(1)) \\ &\qquad + \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \sum_{l=0}^{n-1} \sum_{k=0}^{l} \sum_{0 \le p < q \le l} \mu_{n,k} (3\nu_{l,p}\nu_{l,q} - \mu_{l,p}\nu_{l,q} - \mu_{l,q}\nu_{l,p}) d_{p,i} d_{q,i} (1 + o_\epsilon(1)) \end{aligned}$$

[1]We skip the monomials of degree exactly 1 in noise derivatives because they are mean-zero and will not influence the expectation $\mathbb{E}_\pi[\cdot]$.

$$
\begin{aligned}
&- \frac{s_j}{|g_j|^2} \sum_{r=0}^{n} \nu_{n,r} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \mu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{p,i} d_{r,j} (1 + o_\epsilon(1)) \\
&- \frac{s_j}{|g_j|^2} \sum_{r=0}^{n} \nu_{n,r} \sum_i s_i \sum_{k=0}^{n} \mu_{n,k}(n-k) d_{k,ij} d_{r,j} (1 + o_\epsilon(1)) \\
&+ \frac{\nabla_j \|\boldsymbol{g}\|_1}{2|g_j|^3} \sum_{r=0}^{n} (3\nu_{n,r}^2 - \nu_{n,r}) \sum_{k=0}^{n} \mu_{n,k}(n-k) d_{r,j}^2 (1 + o_\epsilon(1)) \\
&+ \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{k=0}^{n} \mu_{n,k}(n-k) \sum_{0 \le p < q \le n} 3\nu_{n,p}\nu_{n,q} d_{p,j} d_{q,j} (1 + o_\epsilon(1)),
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3} = & \left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_0 + \left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_1 + \left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_2 \\
& + O(d^3),
\end{aligned}
$$

where

$$
\begin{aligned}
&\left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_0 := \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \sum_{k=0}^{n-1} (n-k) \nu_{n,k} (1 + o_\epsilon(1)), \\
&\left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_1 := \text{[skipped]}, \\
&\left[\frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}\right]_2 \\
&\quad := \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{r=0}^{n-1} (n-r) \nu_{n,r} \sum_{k=0}^{n} (4\nu_{n,k}^2 - \nu_{n,k} - 2\mu_{n,k}\nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&\quad + \frac{2\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{r=0}^{n-1} (n-r) \nu_{n,r} \sum_{0 \le p < q \le n} (4\nu_{n,p}\nu_{n,q} - \mu_{n,p}\nu_{n,q} - \mu_{n,q}\nu_{n,p}) d_{p,j} d_{q,j} (1 + o_\epsilon(1)) \\
&\quad + \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \sum_{r=0}^{n} \nu_{n,k} (\mu_{l,p} - \nu_{l,p})(\mu_{n,r} - 2\nu_{n,r}) d_{p,i} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad + \frac{s_j}{|g_j|^2} \sum_i s_i \sum_{k=0}^{n-1} \sum_{r=0}^{n} (n-k) \nu_{n,k} (\mu_{n,r} - 2\nu_{n,r}) d_{k,ij} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad + \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{k=0}^{n-1} \sum_{r=0}^{n} (n-k) \nu_{n,k} (\mu_{n,r} - 2\nu_{n,r}) d_{k,j} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad - \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{p=0}^{n-1} (n-p) \nu_{n,p} \sum_{k=0}^{n} \sum_{r=0}^{n} \nu_{n,k} (\mu_{n,r} - 2\nu_{n,r}) d_{k,j} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad + \frac{\nabla_j \|\boldsymbol{g}\|_1}{2|g_j|^3} \sum_{r=0}^{n-1} (n-r) \nu_{n,r} \sum_{k=0}^{n} (3\nu_{n,k}^2 - \nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&\quad + \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{r=0}^{n-1} (n-r) \nu_{n,r} \sum_{0 \le p < q \le n} (3\nu_{n,p}\nu_{n,q}) d_{p,j} d_{q,j} (1 + o_\epsilon(1)) \\
&\quad - \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k} (\mu_{l,p} - \nu_{l,p}) \sum_{r=0}^{n} \nu_{n,r} d_{p,i} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad - \frac{s_j}{|g_j|^2} \sum_i s_i \sum_{k=0}^{n-1} (n-k) \nu_{n,k} \sum_{r=0}^{n} \nu_{n,r} d_{k,ij} d_{r,j} (1 + o_\epsilon(1)) \\
&\quad - \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \sum_{k=0}^{n-1} (n-k) \nu_{n,k} \sum_{r=0}^{n} \nu_{n,r} d_{k,j} d_{r,j} (1 + o_\epsilon(1))
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{2|g_i|^2} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k} (3\nu_{l,p}^2 - \nu_{l,p} - 2\mu_{l,p}\nu_{l,p}) d_{p,i}^2 (1 + o_\epsilon(1)) \\
&+ \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \sum_{l=0}^{n-1} \sum_{k=0}^{l} \sum_{0 \le p < q \le l} \nu_{n,k} (3\nu_{l,p}\nu_{l,q} - \mu_{l,p}\nu_{l,q} - \mu_{l,q}\nu_{l,p}) d_{p,i} d_{q,i} (1 + o_\epsilon(1)) \\
&+ \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k} (\mu_{l,p} - \nu_{l,p}) d_{k,ij} d_{p,i} (1 + o_\epsilon(1)) \\
&+ \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k} (\mu_{l,p} - \nu_{l,p}) d_{k,j} d_{p,i} (1 + o_\epsilon(1)) \\
&+ \frac{s_j}{|g_j|^2} \sum_i s_i \sum_{k=0}^{n-1} (n-k) \nu_{n,k} d_{k,j} d_{k,ij} (1 + o_\epsilon(1)).
\end{aligned}
$$

The proof is immediate from lemmas collected below.

Proof. To get the expansion for $L_{n,j}(\boldsymbol{\theta})/R_{n,j}(\boldsymbol{\theta})$, multiply the expansions for $L_{n,j}(\boldsymbol{\theta})$ (from Lemma A.4) and $R_{n,j}(\boldsymbol{\theta})^{-1}$ (from Lemma A.3).

To get the expansion for $M_{n,j}(\boldsymbol{\theta})P_{n,j}(\boldsymbol{\theta})/R_{n,j}(\boldsymbol{\theta})^3$, multiply the expansions for $P_{n,j}(\boldsymbol{\theta})R_{n,j}(\boldsymbol{\theta})^{-1}$ and $M_{n,j}(\boldsymbol{\theta})R_{n,j}(\boldsymbol{\theta})^{-2}$ from Lemma A.5. □

Now we state and prove the lemmas.

We start with a very simple expansion separated for pedagogical reasons to illustrate the approach (all following expansions are done similarly).

Lemma A.3 (Illustration of the approach: expansions for $M_{n,j}(\boldsymbol{\theta})$ and $R_{n,j}(\boldsymbol{\theta})^{-1}$)**.** *We have*

$$
M_{n,j}(\boldsymbol{\theta}) = g_j + \sum_{k=0}^{n} \mu_{n,k} d_{k,j}, \tag{10}
$$

$$
\begin{aligned}
R_{n,j}(\boldsymbol{\theta})^{-1} = |g_j|^{-1}(1 + o_\epsilon(1)) & \\
&- \frac{s_j}{|g_j|^2} \sum_{k=0}^{n} \nu_{n,k} d_{k,j} (1 + o_\epsilon(1)) \\
&+ \frac{1}{2|g_j|^3} \sum_{k=0}^{n} (3\nu_{n,k}^2 - \nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&+ \frac{1}{|g_j|^3} \sum_{0 \le p < q \le n} 3\nu_{n,p}\nu_{n,q} d_{p,j} d_{q,j} (1 + o_\epsilon(1)) \\
&+ O\big(d^3\big).
\end{aligned}
$$

Proof. Equation (10) follows directly from definitions. The expansion $R_{n,j}(\boldsymbol{\theta})^{-1}$ is obtained by the following chain of equalities:

$$
\begin{aligned}
R_{n,j}(\boldsymbol{\theta})^{-1} &= \left(g_j^2 + \epsilon + 2g_j \sum_{k=0}^{n} \nu_{n,k} d_{k,j} + \sum_{k=0}^{n} \nu_{n,k} d_{k,j}^2 \right)^{-1/2} \\
&= (g_j^2 + \epsilon)^{-1/2} - (g_j^2 + \epsilon)^{-3/2} g_j \sum_{k=0}^{n} \nu_{n,k} d_{k,j} - \frac{1}{2} (g_j^2 + \epsilon)^{-3/2} \sum_{k=0}^{n} \nu_{n,k} d_{k,j}^2 \\
&\quad + \frac{3}{2} (g_j^2 + \epsilon)^{-5/2} g_j^2 \left(\sum_{k=0}^{n} \nu_{n,k} d_{k,j} \right)^2 + O(d^3)
\end{aligned}
$$

$$\begin{aligned}
&= \frac{1}{(g_j^2+\epsilon)^{1/2}} - \frac{g_j}{(g_j^2+\epsilon)^{3/2}}\sum_{k=0}^{n}\nu_{n,k}d_{k,j} - \frac{1}{2(g_j^2+\epsilon)^{3/2}}\sum_{k=0}^{n}\nu_{n,k}d_{k,j}^2 \\
&\quad + \frac{3}{2}\frac{g_j^2}{(g_j^2+\epsilon)^{5/2}}\sum_{k=0}^{n}\nu_{\nu,k}^2 d_{k,j}^2 + 3\frac{g_j^2}{(g_j^2+\epsilon)^{5/2}}\sum_{0\le p<q\le n}\nu_{n,p}\nu_{n,q}d_{p,j}d_{q,j} + O(d^3) \\
&= |g_j|^{-1}(1+o_\epsilon(1)) - \frac{s_j}{g_j^2}\sum_{k=0}^{n}\nu_{n,k}d_{k,j}(1+o_\epsilon(1)) \\
&\quad + \frac{1}{2|g_j|^3}\sum_{k=0}^{n}(3\nu_{n,k}^2-\nu_{n,k})d_{k,j}^2(1+o_\epsilon(1)) \\
&\quad + \frac{1}{|g_j|^3}\sum_{0\le p<q\le n}3\nu_{n,p}\nu_{n,q}d_{p,j}d_{q,j}(1+o_\epsilon(1)) + O(d^3),
\end{aligned}$$

where we used $\sum_{k=0}^{n}\nu_{n,k}=1$. □

Lemma A.4 (Warm-up: expansions for $L_{n,j}(\boldsymbol{\theta})$ and $P_{n,j}(\boldsymbol{\theta})$)**.** *The following formal expansions (up to quadratic terms in noise) hold:*

$$\begin{aligned}
L_{n,j}(\boldsymbol{\theta}) &= \sum_i g_{ij}s_i\sum_{k=0}^{n}\mu_{n,k}(n-k)(1+o_\epsilon(1)) \\
&\quad + \sum_i \frac{g_{ij}}{|g_i|}\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\mu_{n,k}(\mu_{l,p}-\nu_{l,p})d_{p,i}(1+o_\epsilon(1)) \\
&\quad + \sum_i s_i\sum_{k=0}^{n}\mu_{n,k}(n-k)d_{k,ij}(1+o_\epsilon(1)) \\
&\quad + \sum_i \frac{g_{ij}s_i}{|g_i|^2}\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\mu_{n,k}\frac{3\nu_{l,p}^2-\nu_{l,p}-2\mu_{l,p}\nu_{l,p}}{2}d_{p,i}^2(1+o_\epsilon(1)) \\
&\quad + \sum_i \frac{1}{|g_i|}\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\mu_{n,k}(\mu_{l,p}-\nu_{l,p})d_{k,ij}d_{p,i}(1+o_\epsilon(1)) \\
&\quad + \sum_i \frac{g_{ij}s_i}{|g_i|^2}\sum_{l=0}^{n-1}\sum_{k=0}^{l}\sum_{0\le p<q\le l}\mu_{n,k}(3\nu_{l,p}\nu_{l,q}-\mu_{l,p}\nu_{l,q}-\mu_{l,q}\nu_{l,p})d_{p,i}d_{q,i}(1+o_\epsilon(1)) \\
&\quad + O(d^3), \\
P_{n,j}(\boldsymbol{\theta}) &= g_j\nabla_j\|\boldsymbol{g}\|_1\sum_{k=0}^{n-1}(n-k)\nu_{n,k}(1+o_\epsilon(1)) \\
&\quad + g_j\sum_i\frac{g_{ij}}{|g_i|}\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\nu_{n,k}(\mu_{l,p}-\nu_{l,p})d_{p,i}(1+o_\epsilon(1)) \\
&\quad + g_j\sum_i s_i\sum_{k=0}^{n-1}(n-k)\nu_{n,k}d_{k,ij}(1+o_\epsilon(1)) \\
&\quad + \nabla_j\|\boldsymbol{g}\|_1\sum_{k=0}^{n-1}(n-k)\nu_{n,k}d_{k,j}(1+o_\epsilon(1)) \\
&\quad + g_j\sum_i\frac{g_{ij}s_i}{2|g_i|^2}\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\nu_{n,k}(3\nu_{l,p}^2-\nu_{l,p}-2\mu_{l,p}\nu_{l,p})d_{p,i}^2(1+o_\epsilon(1))
\end{aligned}$$

$$
\begin{aligned}
&+ g_j \sum_i \frac{g_{ij} s_i}{|g_i|^2} \sum_{l=0}^{n-1} \sum_{k=0}^{l} \sum_{0 \le p < q \le l} \nu_{n,k}(3\nu_{l,p}\nu_{l,q} - \mu_{l,p}\nu_{l,q} - \mu_{l,q}\nu_{l,p}) d_{p,i} d_{q,i}(1 + o_\epsilon(1)) \\
&+ g_j \sum_i \frac{1}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{k,ij} d_{p,i}(1 + o_\epsilon(1)) \\
&+ \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{k,j} d_{p,i}(1 + o_\epsilon(1)) \\
&+ \sum_i s_i \sum_{k=0}^{n-1} (n-k)\nu_{n,k} d_{k,j} d_{k,ij}(1 + o_\epsilon(1)) \\
&+ O(d^3).
\end{aligned}
$$

Proof. Multiplying the formal expansions for $M_{n,j}(\boldsymbol{\theta})$ and $R_j^{-1}(\boldsymbol{\theta})$ from Lemma A.3 gives

$$
\begin{aligned}
&M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} \\
&\quad = s_j(1 + o_\epsilon(1)) \\
&\qquad + |g_j|^{-1} \sum_{k=0}^{n} (\mu_{n,k} - \nu_{n,k}) d_{k,j}(1 + o_\epsilon(1)) \\
&\qquad + \frac{s_j}{2|g_j|^2} \sum_{k=0}^{n} (3\nu_{n,k}^2 - \nu_{n,k} - 2\mu_{n,k}\nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&\qquad + \frac{s_j}{|g_j|^2} \sum_{0 \le p < q \le n} (3\nu_{n,p}\nu_{n,q} - \mu_{n,p}\nu_{n,q} - \mu_{n,q}\nu_{n,p}) d_{p,j} d_{q,j}(1 + o_\epsilon(1)) \\
&\qquad + O(d^3).
\end{aligned}
$$

Inserting this into the definitions of $L_{n,j}(\boldsymbol{\theta})$ and $P_{n,j}(\boldsymbol{\theta})$ gives the result. □

Lemma A.5 (Preparation: expansions for $P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}$ and $M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}$)**.** *We have*

$$
\begin{aligned}
P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1} =& \left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_0 + \left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_1 + \left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_2 \\
&+ O(d^3),
\end{aligned}
$$

where

$$
\begin{aligned}
\left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_0 :=& s_j \nabla_j \|\boldsymbol{g}\|_1 \sum_{k=0}^{n-1} (n-k)\nu_{n,k}(1 + o_\epsilon(1)) \\
\left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_1 :=& s_j \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{p,i}(1 + o_\epsilon(1)) \\
&+ s_j \sum_i s_i \sum_{k=0}^{n-1} (n-k)\nu_{n,k} d_{k,ij}(1 + o_\epsilon(1)) \\
&+ \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \sum_{k=0}^{n-1} (n-k)\nu_{n,k} d_{k,j}(1 + o_\epsilon(1)) \\
&- \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \sum_{r=0}^{n-1} (n-r)\nu_{n,r} \sum_{k=0}^{n} \nu_{n,k} d_{k,j}(1 + o_\epsilon(1)) \\
\left[P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}\right]_2 :=& \frac{s_j \nabla_j \|\boldsymbol{g}\|_1}{2|g_j|^2} \sum_{r=0}^{n-1} (n-r)\nu_{n,r} \sum_{k=0}^{n} (3\nu_{n,k}^2 - \nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1))
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{s_j \nabla_j \|\boldsymbol{g}\|_1}{|g_j|^2} \sum_{r=0}^{n-1} (n-r)\nu_{n,r} \sum_{0 \le p < q \le n} (3\nu_{n,p}\nu_{n,q}) d_{p,j} d_{q,j} (1 + o_\epsilon(1)) \\
&- \frac{1}{|g_j|} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) \sum_{r=0}^{n} \nu_{n,r} d_{p,i} d_{r,j} (1 + o_\epsilon(1)) \\
&- \frac{1}{|g_j|} \sum_i s_i \sum_{k=0}^{n-1} (n-k)\nu_{n,k} \sum_{r=0}^{n} \nu_{n,r} d_{k,ij} d_{r,j} (1 + o_\epsilon(1)) \\
&- \frac{s_j \nabla_j \|\boldsymbol{g}\|_1}{|g_j|^2} \sum_{k=0}^{n-1} (n-k)\nu_{n,k} \sum_{r=0}^{n} \nu_{n,r} d_{k,j} d_{r,j} (1 + o_\epsilon(1)) \\
&+ s_j \sum_i \frac{g_{ij} s_i}{2|g_i|^2} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(3\nu_{l,p}^2 - \nu_{l,p} - 2\mu_{l,p}\nu_{l,p}) d_{p,i}^2 (1 + o_\epsilon(1)) \\
&+ s_j \sum_i \frac{g_{ij} s_i}{|g_i|^2} \sum_{l=0}^{n-1} \sum_{k=0}^{l} \sum_{0 \le p < q \le l} \nu_{n,k}(3\nu_{l,p}\nu_{l,q} - \mu_{l,p}\nu_{l,q} - \mu_{l,q}\nu_{l,p}) d_{p,i} d_{q,i} (1 + o_\epsilon(1)) \\
&+ s_j \sum_i \frac{1}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{k,ij} d_{p,i} (1 + o_\epsilon(1)) \\
&+ \frac{1}{|g_j|} \sum_i \frac{g_{ij}}{|g_i|} \sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \nu_{n,k}(\mu_{l,p} - \nu_{l,p}) d_{k,j} d_{p,i} (1 + o_\epsilon(1)) \\
&+ \frac{1}{|g_j|} \sum_i s_i \sum_{k=0}^{n-1} (n-k)\nu_{n,k} d_{k,j} d_{k,ij} (1 + o_\epsilon(1)),
\end{aligned}
$$

and

$$
\begin{aligned}
M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2} = &\left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_0 + \left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_1 + \left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_2 \\
&+ O(d^3),
\end{aligned}
$$

where

$$
\begin{aligned}
\left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_0 :=& \frac{s_j}{|g_j|}(1 + o_\epsilon(1)), \\
\left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_1 :=& \frac{1}{|g_j|^2} \sum_{k=0}^{n} (\mu_{n,k} - 2\nu_{n,k}) d_{k,j} (1 + o_\epsilon(1)), \\
\left[M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}\right]_2 :=& \frac{s_j}{|g_j|^3} \sum_{k=0}^{n} (4\nu_{n,k}^2 - \nu_{n,k} - 2\mu_{n,k}\nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&+ \frac{2s_j}{|g_j|^3} \sum_{0 \le p < q \le n} (4\nu_{n,p}\nu_{n,q} - \mu_{n,p}\nu_{n,q} - \mu_{n,q}\nu_{n,p}) d_{p,j} d_{q,j} (1 + o_\epsilon(1)).
\end{aligned}
$$

Proof. The expansion for $P_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-1}$ follows by multiplying the expansions for $R_{n,j}(\boldsymbol{\theta})^{-1}$ (from Lemma A.3) and $P_{n,j}(\boldsymbol{\theta})$ (from Lemma A.4).

Raising the expansion for $R_{n,j}(\boldsymbol{\theta})^{-1}$ (from Lemma A.3) to the second power yields

$$
\begin{aligned}
R_{n,j}(\boldsymbol{\theta})^{-2} =& \frac{1}{|g_j|^2}(1 + o_\epsilon(1)) \\
&- \frac{2s_j}{|g_j|^3} \sum_{k=0}^{n} \nu_{n,k} d_{k,j} (1 + o_\epsilon(1))
\end{aligned}
$$

$$
\begin{aligned}
&+ \frac{1}{|g_j|^4} \sum_{k=0}^{n} (4\nu_{n,k}^2 - \nu_{n,k}) d_{k,j}^2 (1 + o_\epsilon(1)) \\
&+ \frac{8}{|g_j|^4} \sum_{0 \le p < q \le n} \nu_{n,p}\nu_{n,q} d_{p,j} d_{q,j} (1 + o_\epsilon(1)) + O(d^3).
\end{aligned}
$$

Multiplying this by the expansion for $M_{n,j}(\boldsymbol{\theta})$ (from Lemma A.3), we obtain the expansion for $M_{n,j}(\boldsymbol{\theta}) R_{n,j}(\boldsymbol{\theta})^{-2}$, concluding the proof. □

A.4.2. Calculating the Expectation of the Result

Next, we calculate $\mathbb{E}_\pi[\cdot]$ of the result.

Proposition A.6 (Calculating $\mathbb{E}_\pi[\cdot]$ of the expansions obtained)**.** *We have*

$$
\begin{aligned}
\mathbb{E}_\pi \frac{L_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})} =& \frac{\beta_1}{1-\beta_1} \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} (1 + o_{n,\epsilon}(1)) \\
&+ \frac{\beta_1(\beta_1\beta_2^2 - \beta_1\beta_2 + \beta_1 + \beta_2^2 - 2\beta_2)}{(1-\beta_1)(1+\beta_2)(1-\beta_1\beta_2)} \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \mathbb{E}_\pi d_{0,i}^2 (1 + o_{n,\epsilon}(1)) \\
&+ \frac{\beta_1(\beta_2 - \beta_1)}{(1+\beta_1)(1-\beta_1\beta_2)} \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \mathbb{E}_\pi d_{0,ij} d_{0,i} (1 + o_{n,\epsilon}(1)) \\
&- \frac{\beta_1\beta_2(\beta_2 - \beta_1)(1-\beta_2)}{(1+\beta_2)(1-\beta_1\beta_2)^2} \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \mathbb{E}_\pi d_{0,i} d_{0,j} (1 + o_{n,\epsilon}(1)) \\
&- \frac{\beta_1\beta_2(1-\beta_1)(1-\beta_2)}{(1-\beta_1\beta_2)^2} \frac{s_j}{|g_j|^2} \sum_i s_i \mathbb{E}_\pi d_{0,ij} d_{0,j} (1 + o_{n,\epsilon}(1)) \\
&+ \frac{\beta_1(1-2\beta_2)}{(1-\beta_1)(1+\beta_2)} \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \mathbb{E}_\pi d_{0,j}^2 (1 + o_{n,\epsilon}(1)) \\
&+ O(d^3) + o_{n,\epsilon}(b^{-1}),
\end{aligned}
$$

and

$$
\begin{aligned}
&\mathbb{E}_\pi \frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3} \\
&= \frac{\rho}{1-\rho} \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} (1 + o_{n,\epsilon}(1)) \\
&\quad - \left[(1-\rho)(1+\rho)^2(1-\beta\rho)^2\right]^{-1} \big\{4\beta^2\rho^5 + 2\beta\rho^5 + 3\beta^2\rho^4 - 6\beta\rho^4 - 3\rho^4 - 5\beta^2\rho^3 - 10\beta\rho^3 + 3\rho^3 \\
&\qquad + 3\beta^2\rho^2 + 6\beta\rho^2 + 9\rho^2 + \beta^2\rho - 4\beta\rho - 3\rho\big\} \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|^3} \mathbb{E}_\pi d_{0,j}^2 (1 + o_{n,\epsilon}(1)) \\
&\quad + \frac{\rho(\rho-\beta)[\beta^2(\rho^2 - 3\rho - 1) + \beta(3\rho^2 - \rho + 2) + 1 - 2\rho]}{(1+\beta)(1+\rho)^2(1-\beta\rho)^2} \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \mathbb{E}_\pi d_{0,i} d_{0,j} (1 + o_{n,\epsilon}(1)) \\
&\quad + \frac{3\beta^2\rho^5 - \beta^2\rho^4 + 3\beta^2\rho^3 - \beta^2\rho + \beta\rho^5 - 8\beta\rho^4 - 2\beta\rho^2 + \beta\rho + 4\rho^3 - \rho^2 + \rho}{(1-\rho)(1+\rho)^2(1-\beta\rho)^2} \frac{s_j}{|g_j|^2} \sum_i s_i \mathbb{E}_\pi d_{0,ij} d_{0,j} (1 + o_{n,\epsilon}(1)) \\
&\quad + \frac{\rho(\beta\rho^2 - \beta\rho + \beta + \rho^2 - 2\rho)}{(1+\rho)(1-\rho)(1-\beta\rho)} \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \mathbb{E}_\pi d_{0,i}^2 (1 + o_{n,\epsilon}(1)) \\
&\quad + \frac{\rho(\rho-\beta)}{(1+\rho)(1-\beta\rho)} \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \mathbb{E}_\pi d_{0,ij} d_{0,i} (1 + o_{n,\epsilon}(1)) + O(d^3) + o_{n,\epsilon}(b^{-1}). \qquad (11)
\end{aligned}
$$

Proof. Consider the average of the term like $d_{p,i} d_{q,j}$ where the mini-batch indices p and q are not equal:

$$
\begin{aligned}
\mathbb{E}_\pi d_{p,i} d_{q,j} &= \mathbb{E}_\pi \frac{1}{b} \sum_{r=pb+1}^{(p+1)b} \nabla_i (\ell_{\pi(r)} - \mathcal{L}) \frac{1}{b} \sum_{s=qb+1}^{(q+1)b} \nabla_j (\ell_{\pi(s)} - \mathcal{L}) \\
&= \frac{1}{b^2} \sum_{r=pb+1}^{(p+1)b} \sum_{s=qb+1}^{(q+1)b} \mathbb{E}_\pi \nabla_i (\ell_{\pi(r)} - \mathcal{L}) \nabla_j (\ell_{\pi(s)} - \mathcal{L})
\end{aligned}
$$

$$
\begin{aligned}
&= \mathbb{E}_\pi \nabla_i(\ell_{\pi(1)} - \mathcal{L}) \nabla_j(\ell_{\pi(2)} - \mathcal{L}) \\
&= \frac{1}{mb(mb-1)} \sum_{1 \le r_1 \ne r_2 \le mb} \nabla_i(\ell_{r_1} - \mathcal{L}) \nabla_j(\ell_{r_2} - \mathcal{L}) \\
&= \frac{1}{mb(mb-1)} \Bigg(\sum_{r_1=1}^{mb} \nabla_i(\ell_{r_1} - \mathcal{L}) \sum_{r_2=1}^{mb} \nabla_j(\ell_{r_2} - \mathcal{L}) - \sum_{r=1}^{mb} \nabla_i(\ell_r - \mathcal{L}) \nabla_j(\ell_r - \mathcal{L}) \Bigg) \\
&= -\frac{1}{mb-1} \underbrace{\frac{1}{mb} \sum_{r=1}^{mb} \nabla_i(\ell_r - \mathcal{L}) \nabla_j(\ell_r - \mathcal{L})}_{O(1)} \\
&= O((mb)^{-1}) = o_n(b^{-1}),
\end{aligned}
$$

so, when taking expectations, we can neglect all second-degree monomials of noise derivatives where the two derivatives correspond to different mini-batches (with indices $p \ne q$ in this example). Having made this observation and recalling the expansions obtained in Proposition A.2, it is left to use the linearity of expectation and calculate basic exponential series limits:

$$
\begin{aligned}
&\mathbb{E}_\pi \frac{L_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})} \\
&\quad = \frac{\nabla_j \|\boldsymbol{g}\|_1}{|g_j|} \underbrace{\sum_{k=0}^{n} \mu_{n,k}(n-k)}_{\xrightarrow[n\to\infty]{} \frac{\beta_1}{1-\beta_1}} (1 + o_\epsilon(1)) \\
&\qquad + \frac{1}{|g_j|} \sum_i \frac{g_{ij} s_i}{|g_i|^2} \underbrace{\sum_{l=0}^{n-1} \sum_{k,p=0}^{l} \mu_{n,k} \frac{3\nu_{l,p}^2 - \nu_{l,p} - 2\mu_{l,p}\nu_{l,p}}{2}}_{\xrightarrow[n\to\infty]{} \frac{\beta_1(\beta_1\beta_2^2 - \beta_1\beta_2 + \beta_1 + \beta_2^2 - 2\beta_2)}{(1-\beta_1)(1+\beta_2)(1-\beta_1\beta_2)}} \mathbb{E}_\pi d_{0,i}^2 (1 + o_\epsilon(1)) \\
&\qquad + \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \underbrace{\sum_{l=0}^{n-1} \sum_{k=0}^{l} \mu_{n,k}(\mu_{l,k} - \nu_{l,k})}_{\xrightarrow[n\to\infty]{} \frac{\beta_1(\beta_2-\beta_1)}{(1+\beta_1)(1-\beta_1\beta_2)}} \mathbb{E}_\pi d_{0,ij} d_{0,i} (1 + o_\epsilon(1)) \\
&\qquad - \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \underbrace{\sum_{l=0}^{n-1} \sum_{k=0}^{l} \sum_{p=0}^{l} \mu_{n,k}\nu_{n,p}(\mu_{l,p} - \nu_{l,p})}_{\xrightarrow[n\to\infty]{} \frac{\beta_1\beta_2(\beta_2-\beta_1)(1-\beta_2)}{(1+\beta_2)(1-\beta_1\beta_2)^2}} \mathbb{E}_\pi d_{0,i} d_{0,j} (1 + o_\epsilon(1)) \\
&\qquad - \frac{s_j}{|g_j|^2} \sum_i s_i \underbrace{\sum_{k=0}^{n} \nu_{n,k}\mu_{n,k}(n-k)}_{\xrightarrow[n\to\infty]{} \frac{\beta_1\beta_2(1-\beta_1)(1-\beta_2)}{(1-\beta_1\beta_2)^2}} \mathbb{E}_\pi d_{0,ij} d_{0,j} (1 + o_\epsilon(1)) \\
&\qquad + \frac{\nabla_j \|\boldsymbol{g}\|_1}{2|g_j|^3} \underbrace{\sum_{r=0}^{n} (3\nu_{n,r}^2 - \nu_{n,r}) \sum_{k=0}^{n} \mu_{n,k}(n-k)}_{\xrightarrow[n\to\infty]{} \frac{2\beta_1(1-2\beta_2)}{(1-\beta_1)(1+\beta_2)}} \mathbb{E}_\pi d_{0,j}^2 (1 + o_\epsilon(1)) \\
&\qquad + O\big(d^3\big) + o_{n,\epsilon}(b^{-1}),
\end{aligned}
$$

and similarly,

$$
\mathbb{E}_\pi \frac{M_{n,j}(\boldsymbol{\theta}) P_{n,j}(\boldsymbol{\theta})}{R_{n,j}(\boldsymbol{\theta})^3}
$$

$$
\begin{aligned}
&= \frac{\beta_2}{1-\beta_2}\frac{\nabla_j\|\boldsymbol{g}\|_1}{|g_j|}(1+o_\epsilon(1)) \\
&\quad + \frac{\nabla_j\|\boldsymbol{g}\|_1}{|g_j|^3}\underbrace{\frac{\beta_2}{1-\beta_2}\sum_{k=0}^{n}(4\nu_{n,k}^2-\nu_{n,k}-2\mu_{n,k}\nu_{n,k})}_{\xrightarrow[n\to\infty]{}\frac{\beta_2(3-5\beta_2)}{1-\beta_2^2}-\frac{2\beta_2(1-\beta_1)}{1-\beta_1\beta_2}}\mathbb{E}_\pi d_{0,j}^2(1+o_\epsilon(1)) \\
&\quad + \frac{s_j}{|g_j|^2}\sum_i\frac{g_{ij}}{|g_i|}\underbrace{\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\nu_{n,k}(\mu_{l,p}-\nu_{l,p})(\mu_{n,p}-2\nu_{n,p})}_{\xrightarrow[n\to\infty]{}\frac{\beta_2(\beta_2-\beta_1)(\beta_1^2\beta_2^2-2\beta_1^2\beta_2-\beta_1^2+3\beta_1\beta_2^2+\beta_1-2\beta_2)}{(1+\beta_1)(1+\beta_2)^2(1-\beta_1\beta_2)^2}}\mathbb{E}_\pi d_{0,i}d_{0,j}(1+o_\epsilon(1)) \\
&\quad + \frac{s_j}{|g_j|^2}\sum_i s_i\underbrace{\sum_{k=0}^{n-1}(n-k)\nu_{n,k}(\mu_{n,k}-2\nu_{n,k})}_{\xrightarrow[n\to\infty]{}\frac{\beta_1\beta_2(1-\beta_1)(1-\beta_2)(1+\beta_2)^2-2\beta_2^2(1-\beta_1\beta_2)^2}{(1-\beta_1\beta_2)^2(1+\beta_2)^2}}\mathbb{E}_\pi d_{0,ij}d_{0,j}(1+o_\epsilon(1)) \\
&\quad + \frac{\nabla_j\|\boldsymbol{g}\|_1}{|g_j|^3}\underbrace{\sum_{k=0}^{n-1}(n-k)\nu_{n,k}(\mu_{n,k}-2\nu_{n,k})}_{\xrightarrow[n\to\infty]{}\frac{\beta_1\beta_2(1-\beta_1)(1-\beta_2)(1+\beta_2)^2-2\beta_2^2(1-\beta_1\beta_2)^2}{(1-\beta_1\beta_2)^2(1+\beta_2)^2}}\mathbb{E}_\pi d_{0,j}^2(1+o_\epsilon(1)) \\
&\quad - \frac{\nabla_j\|\boldsymbol{g}\|_1}{|g_j|^3}\underbrace{\frac{\beta_2}{1-\beta_2}\sum_{k=0}^{n}\nu_{n,k}(\mu_{n,k}-2\nu_{n,k})}_{\xrightarrow[n\to\infty]{}-\frac{\beta_2(1-\beta_2)(1+\beta_1)}{(1-\beta_1\beta_2)(1+\beta_2)}}\mathbb{E}_\pi d_{0,j}^2(1+o_\epsilon(1)) \\
&\quad + \frac{\nabla_j\|\boldsymbol{g}\|_1}{2|g_j|^3}\underbrace{\frac{\beta_2}{1-\beta_2}\sum_{k=0}^{n}(3\nu_{n,k}^2-\nu_{n,k})}_{\xrightarrow[n\to\infty]{}\frac{2\beta_2(1-2\beta_2)}{(1-\beta_2)(1+\beta_2)}}\mathbb{E}_\pi d_{0,j}^2(1+o_\epsilon(1)) \\
&\quad - \frac{s_j}{|g_j|^2}\sum_i\frac{g_{ij}}{|g_i|}\underbrace{\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\nu_{n,k}(\mu_{l,p}-\nu_{l,p})\nu_{n,p}}_{\xrightarrow[n\to\infty]{}\frac{\beta_2^2(\beta_2-\beta_1)}{(1+\beta_2)^2(1-\beta_1\beta_2)}}\mathbb{E}_\pi d_{0,i}d_{0,j}(1+o_\epsilon(1)) \\
&\quad - \frac{s_j}{|g_j|^2}\sum_i s_i\underbrace{\sum_{k=0}^{n-1}(n-k)\nu_{n,k}^2}_{\xrightarrow[n\to\infty]{}\frac{\beta_2^2}{(1+\beta_2)^2}}\mathbb{E}_\pi d_{0,ij}d_{0,j}(1+o_\epsilon(1)) \\
&\quad - \frac{\nabla_j\|\boldsymbol{g}\|_1}{|g_j|^3}\underbrace{\sum_{k=0}^{n-1}(n-k)\nu_{n,k}^2}_{\xrightarrow[n\to\infty]{}\frac{\beta_2^2}{(1+\beta_2)^2}}\mathbb{E}_\pi d_{0,j}^2(1+o_\epsilon(1)) \\
&\quad + \frac{1}{|g_j|}\sum_i\frac{g_{ij}s_i}{2|g_i|^2}\underbrace{\sum_{l=0}^{n-1}\sum_{k,p=0}^{l}\nu_{n,k}(3\nu_{l,p}^2-\nu_{l,p}-2\mu_{l,p}\nu_{l,p})}_{\xrightarrow[n\to\infty]{}\frac{2\beta_2(\beta_1\beta_2^2-\beta_1\beta_2+\beta_1+\beta_2^2-2\beta_2)}{(1+\beta_2)(1-\beta_2)(1-\beta_1\beta_2)}}\mathbb{E}_\pi d_{0,i}^2(1+o_\epsilon(1))
\end{aligned}
$$

$$+ \frac{1}{|g_j|} \sum_i \frac{1}{|g_i|} \underbrace{\sum_{l=0}^{n-1} \sum_{k=0}^{l} \nu_{n,k} (\mu_{l,k} - \nu_{l,k})}_{\xrightarrow[n \to \infty]{} \frac{\beta_2(\beta_2 - \beta_1)}{(1+\beta_2)(1-\beta_1\beta_2)}} \mathbb{E}_\pi d_{0,ij} d_{0,i} (1 + o_\epsilon(1))$$

$$+ \frac{s_j}{|g_j|^2} \sum_i \frac{g_{ij}}{|g_i|} \underbrace{\sum_{l=0}^{n-1} \sum_{k=0}^{l} \nu_{n,k} (\mu_{l,k} - \nu_{l,k})}_{\xrightarrow[n \to \infty]{} \frac{\beta_2(\beta_2 - \beta_1)}{(1+\beta_2)(1-\beta_1\beta_2)}} \mathbb{E}_\pi d_{0,j} d_{0,i} (1 + o_\epsilon(1))$$

$$+ \frac{s_j}{|g_j|^2} \sum_i s_i \underbrace{\sum_{k=0}^{n-1} (n-k) \nu_{n,k}}_{\xrightarrow[n \to \infty]{} \frac{\beta_2}{1-\beta_2}} \mathbb{E}_\pi d_{0,ij} d_{0,j} (1 + o_\epsilon(1)) + O(d^3) + o_{n,\epsilon}(b^{-1}),$$

concluding the proof. □

Lemma A.7. *We have for all $k \in [1 : mb]$, $i, j \in [1 : \dim \boldsymbol{\theta}]$*

$$\mathbb{E}_\pi d_{k,i} d_{k,j} = \frac{m-1}{mb-1} \Sigma_{ij},$$
$$\mathbb{E}_\pi d_{k,ij} d_{k,j} = \frac{m-1}{2(mb-1)} \nabla_i \Sigma_{jj},$$

where $\boldsymbol{\Sigma}$ is the empirical covariance matrix of per-sample gradients:

$$\Sigma_{ij} := \frac{1}{mb} \sum_{p=1}^{mb} \nabla_i (\ell_p - \mathcal{L}) \nabla_j (\ell_p - \mathcal{L}).$$

Proof. For any $r \in [1 : mb]$ we have

$$\mathbb{E}_\pi [\nabla_{ij} (\ell_{\pi(r)} - \mathcal{L}) \nabla_j (\ell_{\pi(r)} - \mathcal{L})] = \frac{1}{mb} \sum_{p=1}^{mb} \nabla_{ij} (\ell_p - \mathcal{L}) \nabla_j (\ell_p - \mathcal{L}) = \frac{1}{2} \nabla_i \Sigma_{jj},$$

and for $r \neq \tilde{r}$,

$$\begin{aligned}
\mathbb{E}_\pi [\nabla_{ij} (\ell_{\pi(r)} - \mathcal{L}) \nabla_j (\ell_{\pi(\tilde{r})} - \mathcal{L})] &= \frac{1}{mb(mb-1)} \sum_{\substack{p,q=1 \\ p \neq q}}^{mb} \nabla_{ij} (\ell_p - \mathcal{L}) \nabla_j (\ell_q - \mathcal{L}) \\
&= -\frac{1}{mb(mb-1)} \sum_{p=1}^{mb} \nabla_{ij} (\ell_p - \mathcal{L}) \nabla_j (\ell_p - \mathcal{L}) \\
&= -\frac{1}{2(mb-1)} \nabla_i \Sigma_{jj}.
\end{aligned}$$

Next,

$$\begin{aligned}
\mathbb{E}_\pi d_{k,i} d_{k,j} &= \mathbb{E}_\pi \left(\frac{1}{b} \sum_{r=kb+1}^{kb+b} (\nabla_i \ell_{(\pi(r))} - \nabla_i \mathcal{L}) \right) \left(\frac{1}{b} \sum_{r=kb+1}^{kb+b} (\nabla_j \ell_{\pi(r)} - \nabla_j \mathcal{L}) \right) \\
&= \frac{1}{b^2} \sum_{r=kb+1}^{kb+b} \mathbb{E}_\pi (\nabla_i \ell_{\pi(r)} - \nabla_i \mathcal{L}) (\nabla_j \ell_{\pi(r)} - \nabla_j \mathcal{L}) \\
&\quad + \frac{1}{b^2} \sum_{kb+1 \leq r \neq \tilde{r} \leq kb+b} \mathbb{E}_\pi (\nabla_i \ell_{\pi(r)} - \nabla_i \mathcal{L}) (\nabla_j \ell_{\pi(\tilde{r})} - \nabla_j \mathcal{L})
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{b}\mathbb{E}_\pi(\nabla_i \ell_{\pi(1)} - \nabla_i \mathcal{L})(\nabla_j \ell_{\pi(1)} - \nabla_j \mathcal{L}) \\
&\quad + \frac{b-1}{b}\mathbb{E}_\pi(\nabla_i \ell_{\pi(1)} - \nabla_i \mathcal{L})(\nabla_j \ell_{\pi(2)} - \nabla_j \mathcal{L}) \\
&= \frac{1}{mb^2}\sum_{p=1}^{mb}(\nabla_i \ell_p - \nabla_i \mathcal{L})(\nabla_j \ell_p - \nabla_j \mathcal{L}) \\
&\quad + \frac{b-1}{mb^2(mb-1)}\sum_{\substack{p,q=1 \\ p\neq q}}^{mb}(\nabla_i \ell_p - \nabla_i \mathcal{L})(\nabla_j \ell_q - \nabla_j \mathcal{L}) \\
&= \frac{1}{mb^2}\sum_{p=1}^{mb}(\nabla_i \ell_p - \nabla_i \mathcal{L})(\nabla_j \ell_p - \nabla_j \mathcal{L}) \\
&\quad - \frac{b-1}{mb^2(mb-1)}\sum_{p=1}^{mb}(\nabla_i \ell_p - \nabla_i \mathcal{L})(\nabla_j \ell_p - \nabla_j \mathcal{L}) \\
&= \frac{m-1}{mb-1}\Sigma_{ij}.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
\mathbb{E}_\pi d_{k,ij} d_{k,j} &= \frac{m-1}{mb-1}\frac{1}{mb}\sum_{p=1}^{mb}(\nabla_{ij} \ell_p - \nabla_{ij} \mathcal{L})(\nabla_j \ell_p - \nabla_j \mathcal{L}) \\
&= \frac{m-1}{2(mb-1)}\nabla_i \Sigma_{jj}.
\end{aligned}
$$

□

Combining Proposition A.6 and Lemma A.7 concludes the proof of Proposition 3.3.

B. Further Evidence and Experiment Details

The benefit of tuning the hyperparameters As pointed out above, the improvements in validation perplexity before overfitting can be substantial if one tunes β_2 (Table 1). In this sense, multi-epoch training can be qualitatively different from large online runs, where Adam is quite stable with respect to (β_1, β_2) (Zhao et al., 2025).

Note also that for some experiments with moderate to large batch sizes the best β_2 is less than $\beta_1 = 0.9$ and is at the left boundary of the sweep (which means the optimal β_1 for validation accuracy is even less). This is consistent with the fact that in the large batch regime, taking β_2 much larger than β_1 is not the best from the perspective of generalization.

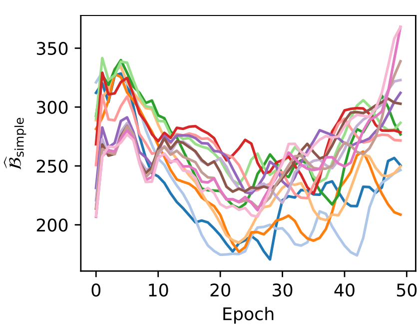
Estimating $\mathcal{B}_{\text{simple}}$ In Figure 4, we plot how the $\mathcal{B}_{\text{simple}}$ quantity changes during training for a few runs from Figure 2 with different learning rates and batch sizes. We see that the scale of $\mathcal{B}_{\text{simple}}$ at relevant epochs is a few hundreds.

Table 1. Transformer-XL trained from scratch on WikiText-2: "optimal" hyperparameter values $\beta_2(\eta)$ we found, and relative improvements $\Delta(\eta)$ in validation perplexity for different learning rates η and batch sizes (after averaging over three iterations). Note that all "optimal" β_2's are smaller than even 0.99, let alone the default 0.999.

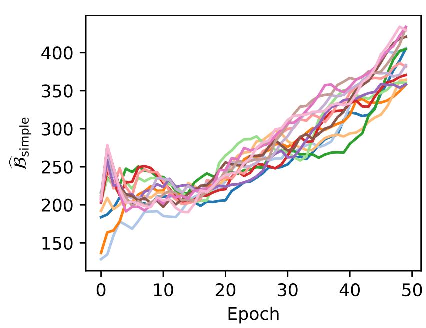
Batch Size	$\beta_2(10^{-3})$	$\Delta(10^{-3})$
128	0.972	4.11%
256	0.972	3.07%
512	0.934	4.72%
1024	0.934	3.60%
2048	0.957	5.59%
4096	0.848	5.09%
8192	0.848	5.87%
16384	0.848	5.29%

Batch Size	$\beta_2(10^{-3.5})$	$\Delta(10^{-3.5})$
128	0.981	1.39%
256	0.957	3.95%
512	0.957	3.91%
1024	0.900	4.95%
2048	0.934	6.31%
4096	0.934	5.68%
8192	0.934	6.11%
16384	0.769	7.43%

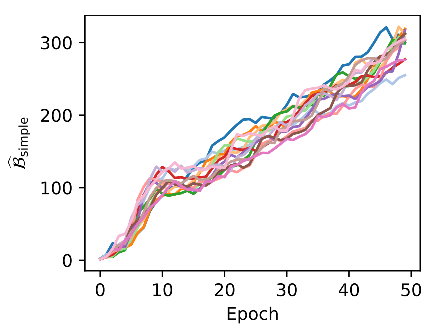
Batch Size	$\beta_2(10^{-4})$	$\Delta(10^{-4})$
128	0.981	3.14%
256	0.972	5.61%
512	0.934	6.33%
1024	0.934	7.36%
2048	0.848	8.58%
4096	0.900	10.49%
8192	0.848	9.11%
16384	0.769	13.01%



(a) batch size 128, $\eta = 10^{-4}$



(b) batch size 512, $\eta = 3.16 \times 10^{-4}$



(c) batch size 4096, $\eta = 0.001$

Figure 4. The estimated simple noise scale $\widehat{\mathcal{B}}_{\text{simple}}$ for different training runs of Transformer-XL on WikiText-2.