Speculative Fine-Tuning: Accelerated Supervised Fine-Tuning with Speculative Heads

Anonymous ACL submission

Abstract

Supervised fine-tuning is a standard method for adapting large language models to downstream tasks, but it can still require substantial compute, especially when repeated across models, datasets, and deployment settings. In this work, we show that speculative decoding, originally introduced for inference acceleration, can also be used to reduce the number of optimization steps required for supervised fine-tuning. Our key observation is that gradients from speculative heads are strongly correlated with gradients from the main language-model head. We use this correlation to construct control-variate gradient estimators that reduce stochastic gradient variance without modifying the original supervised fine-tuning objective. We further show that the standard multi-token prediction (MTP) objective admits a natural control-variate interpretation, connecting MTP training to variance reduction. Moreover, we find that the correlation varies across the layers of the model, allowing us to develop a layer-aware variant of the method. Across supervised fine-tuning experiments, our proposed optimizers require up to a 3.4× reduction in optimization steps or a 1.7× reduction in wall-clock time to reach the same loss. To our knowledge, this is the first work to use speculative decoding for optimizer updates, separating the optimization role of speculative heads from their conventional use as auxiliary prediction heads.

An anonymized code for reproducing our experiments is available at `https://anonymous.4open.science/r/SpeculativeFineTuning-BBE0/`.

1 Introduction

Fine-tuning large language models (LLMs) is a standard approach for adapting pretrained models to downstream tasks, including instruction following, domain adaptation, and task-specific supervised learning (Raffel et al., 2020; Wei et al., 2021; Ouyang et al., 2022). In supervised fine-tuning (SFT), either all model parameters or a smaller set of adapter parameters are optimized on task-specific datasets, which are typically much smaller than the pretraining corpus (Hu et al., 2022; Dettmers et al., 2023; Han et al., 2024). In recent years, fine-tuning has become a widely used method for adapting models to specific target scenarios due to its affordability. Although fine-tuning is substantially less expensive than pretraining, it is still repeatedly performed across model sizes, datasets, and deployment scenarios, making its optimization cost a bottleneck. In this work, we propose a theoretical framework and a practical method for reducing the number of optimization steps required to perform SFT.

The cost of fine-tuning has been extensively studied in previous work. One widely used solution is parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019), which trains only a small number of weights while maintaining acceptable performance. Another line of work aims to reduce memory requirements. Such techniques include quantization and methods for reducing gradient memory (Dettmers et al., 2023; Lv et al., 2024; Zhao et al., 2024). These methods have significantly reduced the cost of fine-tuning, but their main targets are parameter count, memory usage, or per-step cost. In contrast, our paper studies an underexplored area of fine-tuning: reducing the number of optimizer steps. Moreover, our method is orthogonal to the existing methods discussed above and can be combined with them.

Our method is motivated by *speculative decoding*, a method originally developed to accelerate LLM inference. Early versions of speculative decoding used an additional, smaller draft model to predict future tokens, which were then verified by the main model (Leviathan et al., 2023). However, more recent methods use additional speculative heads that reuse the internal states of the main

model to make predictions (Cai et al., 2024; Li et al., 2024). Additionally, recent work on multi-token prediction (MTP) (Gloeckle et al., 2024) has shown that training the main model and speculative heads simultaneously can improve the generation quality of the main model. This training strategy has been adopted by several recent models (Liu et al., 2024; Qwen Team, 2026). Together with the findings of Samragh et al. (2025), who show that the internal states of LLMs contain information about future tokens, this suggests that speculative heads may provide useful information beyond token prediction.

The central observation of this paper is that speculative heads provide informative optimization signals during fine-tuning. Specifically, we find that stochastic gradients induced by speculative-head losses are strongly correlated with the gradients induced by the main language-model loss, with respect to the same trainable parameters. This correlation allows us to use speculative gradients as *control variates* (Lavenberg and Welch, 1981; Wang et al., 2013) for the main SFT gradient, reducing stochastic gradient variance while leaving the original supervised fine-tuning objective unchanged. We further observe that the strength of this correlation varies across model layers, which motivates a layerwise control-variate estimator with different coefficients for different parts of the network. Building on this view, we derive speculative variants of SGD and Adam, analyze their connection to MTP-style training, and show empirically that they converge faster than standard fine-tuning baselines.

Our control-variate perspective provides a new interpretation of multi-token prediction training. Standard MTP augments the main loss with an auxiliary speculative loss, thereby changing the optimization objective. We show that, when the speculative gradient is used as a control variate, the MTP gradient decomposes into two components: a variance-reduced estimator of the main-gradient direction and a deterministic drift given by the mean speculative gradient. Equivalently, the sample-level stochasticity of the auxiliary speculative gradient can be canceled while retaining its mean optimization effect. This reformulation suggests that part of the benefit of MTP-style training comes from variance reduction rather than only from additional token-prediction supervision.

We provide experimental results on downstream supervised fine-tuning using the GSM8K dataset (Cobbe et al., 2021), considering both full fine-tuning and LoRA. Our method achieves a consistent 3–3.4× reduction in the number of optimization steps while achieving a better final loss.

This work broadens the role of speculative decoding beyond inference acceleration. While prior work has used speculative mechanisms to reduce decoding latency, we show that speculative heads can also serve as control variates for stochastic gradients during optimization, thereby reducing gradient variance and accelerating supervised fine-tuning.

Our contributions are as follows:

- We empirically identify strong correlations between stochastic gradients from the main language-model head and speculative heads.
- We formulate speculative-head gradients as control variates for the main gradient, yielding variance-reduced optimizer updates.
- We reinterpret multi-token prediction training through a control-variate decomposition, separating its mean speculative-gradient contribution from its stochastic variance-reduction effect.
- We show that gradient correlations are layer-dependent and introduce a layer-aware control-variate estimator that exploits this structure.
- We demonstrate that the proposed estimators reduce the number of optimization steps needed to reach comparable loss by up to 3.4×, or reduce wall-clock time by up to 1.7×, across both full fine-tuning and LoRA.
- We position speculative decoding as a tool for optimization, rather than only for inference acceleration, opening a complementary direction for efficient fine-tuning.

2 Preliminaries

In this section, we briefly introduce supervised fine-tuning and speculative decoding.

2.1 Supervised Fine-Tuning

A standard pipeline for training large language models consists of pre-training followed by post-training (alignment). While pre-training is crucial for enabling models to acquire broad language understanding and contextual representations, it is computationally expensive and therefore mainly

accessible to large technology companies. In SFT, the model is trained on instruction-response data to adapt its output distribution to target tasks, domains, or response styles.

We next formalize SFT as an optimization problem. Let

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$$

be a supervised fine-tuning dataset, where x_i is an input prompt and $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ is the target response. An autoregressive language model with parameters θ defines

$$p_\theta(y_i \mid x_i) = \prod_{t=1}^{T_i} p_\theta(y_{i,t} \mid x_i, y_{i,<t}).$$

SFT minimizes the negative log-likelihood

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \log p_\theta(y_{i,t} \mid x_i, y_{i,<t}).$$

At each step, a mini-batch $B \subset \mathcal{D}$ is sampled, and the stochastic gradient $g_t = \nabla_\theta \mathcal{L}_B(\theta_t)$ of the empirical loss over this mini-batch is computed. The parameters are then updated according to the optimizer's step rule. For stochastic gradient descent, this rule is typically

$$\theta_{t+1} = \theta_t - \eta g_t,$$

where η is the learning rate.

We consider both full backbone fine-tuning and LoRA adapter tuning (Hu et al., 2022), where only low-rank adapter parameters are updated while the pretrained weights remain frozen.

2.2 Speculative Decoding

Speculative decoding accelerates autoregressive generation by using a draft model or auxiliary prediction heads to propose future tokens, which are then verified by the main model (Leviathan et al., 2023). Recent integrated methods attach speculative heads to the main model and reuse its hidden states to predict multiple future tokens (Cai et al., 2024; Li et al., 2024).

Motivated by speculative decoding, Gloeckle et al. (2024) proposed multi-token prediction (MTP), a pretraining objective in which the main language model is trained together with additional speculative heads that predict multiple future tokens, improving final model quality and convergence speed. MTP augments the standard next-token pretraining loss with auxiliary speculative losses:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{main}}(\theta) + \lambda \mathcal{L}_{\text{spec}}(\theta).$$

This objective is therefore modified and does not directly optimize only the main model's outputs. However, empirical results show that optimizing this objective improves the quality of the main model. Although this method has been widely incorporated into recent models, its theoretical basis remains underexplored. We further show that the standard MTP update admits a natural control-variate interpretation: its stochastic auxiliary-gradient component can be canceled while its mean optimization effect is retained. This connection provides a theoretical explanation for part of the optimization benefit of MTP-style training.

3 Speculative Fine-Tuning

In this section, we first develop the theoretical foundation of our method and then derive two optimizer variants, SpecSGD and SpecAdam, which are the main algorithmic contributions of this work.

3.1 Speculative Gradient as Control Variate

Let us denote the gradient computed from the outputs of the main head on a mini-batch $\mathcal{B} \subset \mathcal{D}$ with respect to the trainable parameters θ by $\nabla_\theta M(\mathcal{B})$. Similarly, let $\nabla_\theta S(\mathcal{B})$ denote the gradient computed from the speculative-head loss on the same mini-batch.

Since $\nabla_\theta M(\mathcal{B})$ and $\nabla_\theta S(\mathcal{B})$ are vector-valued random variables, we define their variance, covariance, and correlation using the Euclidean inner product:

$$\mathrm{Var}(\nabla_\theta M(\mathcal{B})) := \mathbb{E}_{\mathcal{B}} \left[\left\| \nabla_\theta M(\mathcal{B}) - \bar{\nabla}_\theta M \right\|_2^2 \right],$$

$$\mathrm{Var}(\nabla_\theta S(\mathcal{B})) := \mathbb{E}_{\mathcal{B}} \left[\left\| \nabla_\theta S(\mathcal{B}) - \bar{\nabla}_\theta S \right\|_2^2 \right],$$

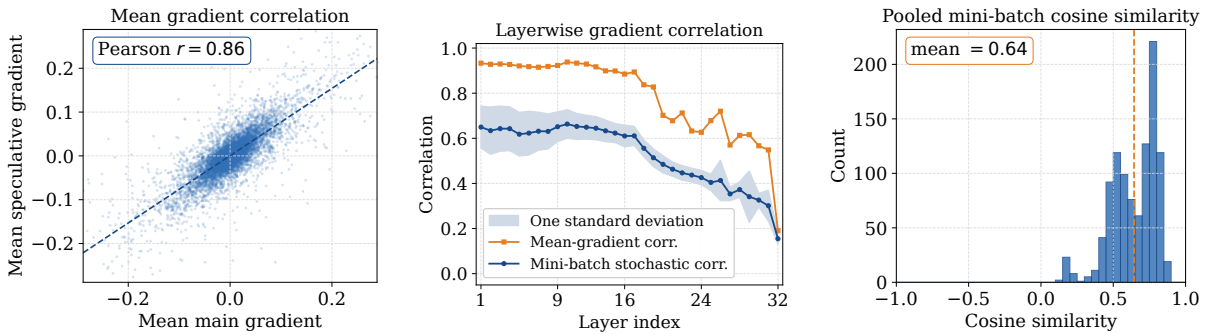
and

$$\mathrm{Cov}(\nabla_\theta M(\mathcal{B}), \nabla_\theta S(\mathcal{B})) := \mathbb{E}_{\mathcal{B}} \left[\left\langle \nabla_\theta M(\mathcal{B}) - \bar{\nabla}_\theta M, \nabla_\theta S(\mathcal{B}) - \bar{\nabla}_\theta S \right\rangle \right].$$

The corresponding correlation is

$$\mathrm{Corr}(\nabla_\theta M(\mathcal{B}), \nabla_\theta S(\mathcal{B})) := \frac{\mathrm{Cov}(\nabla_\theta M(\mathcal{B}), \nabla_\theta S(\mathcal{B}))}{\sqrt{\mathrm{Var}(\nabla_\theta M(\mathcal{B}))\mathrm{Var}(\nabla_\theta S(\mathcal{B}))}}.$$

We use the following assumption.



(a) Pearson correlation between full main-head and speculative-head gradients at a fixed parameter value, computed over the full dataset.

(b) Layerwise gradient correlation at a fixed parameter value, computed both over the full dataset and over mini-batches.

(c) Distribution of mini-batch cosine similarities between main-head and speculative-head gradients at a fixed parameter value.

Figure 1: Correlation between main-head and speculative-head gradients, computed using the Qwen3.5-9B model.

Assumption 1. *For every parameter value θ considered during training, the main-head and speculative-head gradients are positively correlated:*

$$\mathrm{Corr}(\nabla_\theta M(\mathcal{B}), \nabla_\theta S(\mathcal{B})) = \rho > 0.$$

This assumption arises from the nature of speculative heads. Regardless of their structure, they are trained to predict sequential tokens ahead. Thus, the correlation between the gradient from the main head and speculative heads comes from the intuition that small change (gradient in our case) in the model parameters similarly affect both main head and speculative heads. This assumption is confirmed empirically in Figures 1a and 1c.

Using this assumption, we define an unbiased *control-variate estimator* for the main gradient:

$$g_\theta(\mathcal{B}) := \nabla_\theta M(\mathcal{B}) - \alpha(\nabla_\theta S(\mathcal{B}) - \bar{\nabla}_\theta S), \quad (1)$$

where $\bar{\nabla}_\theta S = \mathbb{E}\left[\nabla_\theta S(\mathcal{B})\right]$.

Lemma 1. *$g_\theta(\mathcal{B})$ is an unbiased estimator for $\bar{\nabla}_\theta M = \mathbb{E}\left[\nabla_\theta M(\mathcal{B})\right]$.*

Lemma 2. *The variance of $g_\theta(x)$ is minimized at*

$$\alpha^* = \frac{\mathrm{Cov}(\nabla_\theta M(\mathcal{B}), \nabla_\theta S(\mathcal{B}))}{\mathrm{Var}(\nabla_\theta S(\mathcal{B}))}.$$

Moreover, with this choice,

$$\mathrm{Var}(g_\theta(\mathcal{B})) = (1 - \rho^2)\mathrm{Var}(\nabla_\theta M(\mathcal{B})).$$

These lemmas are well-known in statistics, but for completeness, we provide proofs in the sections B.1 and B.2. They show that the variance of the stochastic gradient can be reduced by a factor of $1 - \rho^2$.

An open question is how to obtain the value of $\bar{\nabla}_\theta S$, since computing it exactly would require evaluating the speculative gradient over the full dataset $\mathcal{D}$ at the current value of θ. We propose to calculate it approximately using an exponential moving average (EMA), and we empirically see that this approximation is sufficient to obtain variance reduction.

The optimal coefficient α^* can also be estimated online using moving averages of the covariance and variance terms. However, this requires storing additional optimizer state and performing extra computations during training. Instead, we estimate α^* in an offline calibration phase. Empirically, for the models and datasets considered in this work, this coefficient changes slowly during fine-tuning. Regarding the stability of the offline choice and the fact that the correlation does not change drastically during fine-tuning, we refer the reader to Sections E and F.

3.2 Layerwise Correlation

While the correlation presented in the previous subsection already gives us a way to construct the control variate, we have observed an important pattern in the behaviour of interlayer gradient correlation. To measure this, we compute both gradients with respect to each layer's parameters θ_l. In the case of LoRA fine-tuning, θ_l denotes the parameters of the LoRA adapters in the l-th layer. We denote the corresponding layerwise gradients by $\nabla^l_\theta M(\mathcal{B})$ and $\nabla^l_\theta S(\mathcal{B})$.

We observe that the layerwise correlation is highest in the first half of the model and decreases in the later layers. The empirical results are shown in Figure 1b.

This pattern can be used to choose the control-variate coefficient more precisely. Instead of using a single coefficient for the whole model, we compute a separate coefficient for each layer:

$$\alpha_l^* = \frac{\mathrm{Cov}(\nabla_\theta^l M(\mathcal{B}), \nabla_\theta^l S(\mathcal{B}))}{\mathrm{Var}(\nabla_\theta^l S(\mathcal{B}))}.$$

As in the previous section, these coefficients can be estimated in advance. Then Equation (1) becomes

$$g_\theta^l(\mathcal{B}) := \nabla_\theta^l M(\mathcal{B}) - \alpha_l^*(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S). \quad (2)$$

3.3 Combination with MTP Loss

In this subsection, we show that the previously proposed MTP loss (Gloeckle et al., 2024) admits a natural interpretation through the control-variate view. In particular, when the control-variate coefficient is chosen optimally, the stochastic component of the auxiliary MTP gradient is canceled, while its mean speculative gradient contribution is preserved.

Empirically, we find that adding an auxiliary speculative loss, as in MTP, accelerates convergence. Intuitively, this may be explained by forcing the model to solve a more difficult prediction problem: modeling several future tokens in a single forward pass. However, in this section we provide another interpretation of this observation. Formally, the objective can be written as

$$\min_\theta \left[M(\mathcal{D}) + \lambda S(\mathcal{D})\right],$$

where λ controls the strength of the auxiliary speculative loss.

The stochastic gradient of this objective is

$$\nabla_\theta M(\mathcal{B}) + \lambda \nabla_\theta S(\mathcal{B}).$$

Therefore, the target gradient is no longer the main-head gradient alone. However, the positive-correlation assumption still implies positive correlation between the combined gradient and the speculative gradient. Indeed, let us expand the correlation:

$$\mathrm{Corr}\left(\nabla_\theta M(\mathcal{B}) + \lambda \nabla_\theta S(\mathcal{B}), \nabla_\theta S(\mathcal{B})\right) = \frac{\rho\sigma_M + \lambda\sigma_S}{\sqrt{\sigma_M^2 + 2\lambda\rho\sigma_M\sigma_S + \lambda^2\sigma_S^2}} > 0,$$

where $\sigma_M^2 = \mathrm{Var}(\nabla_\theta M(\mathcal{B}))$, $\sigma_S^2 = \mathrm{Var}(\nabla_\theta S(\mathcal{B}))$ and ρ comes from Assumption 1. The derivation of this equation is deferred to Section B.3.

Thus, we can replace the control variate for $\nabla_\theta M(\mathcal{B})$ with a control variate for

$$\nabla_\theta M(\mathcal{B}) + \lambda \nabla_\theta S(\mathcal{B}).$$

Similarly to Equation (2), we define the layerwise MTP gradient with control variate as

$$g_\theta^l(\mathcal{B}; \lambda) := \left(\nabla_\theta^l M(\mathcal{B}) + \lambda \nabla_\theta^l S(\mathcal{B})\right) - \alpha_l^*(\lambda)\left(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S\right). \quad (3)$$

The layerwise estimator in Equation (2) is recovered by setting $\lambda = 0$. Hence, in the remainder of the paper, we work with Equation (3), which includes the previous estimator as a special case.

Note that α_l^* here and in Equation (2) are different. α_l^* can be calculated with an offline method as before. However, there is a simple link between this coefficient for $\lambda = 0$ and $\lambda \neq 0$.

Lemma 3. *Let $\alpha_l^*(\lambda)$ denote the optimal scalar control-variate coefficient for Equation (3) with fixed λ. Then*

$$\alpha_l^*(\lambda) = \alpha_l^*(0) + \lambda.$$

The proof can be found in Section B.4. This lemma shows that once the optimal value is known for one value of λ, it can be immediately converted to the optimal value for any other λ'. In particular,

$$\alpha_l^*(\lambda') = \alpha_l^*(\lambda) + \lambda' - \lambda.$$

Now, we show that MTP loss can be formulated in terms of a control variate for the main objective with $\alpha_l^*(0)$.

Proposition 1. *Assume that the exact mean $\bar{\nabla}_\theta^l S$ is available and that $\alpha_l^*(\lambda) = \alpha_l^*(0) + \lambda$. Then Equation (3) can be written as*

$$g_\theta^l(\mathcal{B}; \lambda) = \nabla_\theta^l M(\mathcal{B}) + \lambda \bar{\nabla}_\theta^l S - \alpha_l^*(0)\left(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S\right).$$

This proposition shows that the control variate cancels the stochastic component of the auxiliary MTP gradient while preserving its population-gradient contribution. The proof can be found in Section B.5. *Thus, speculative fine-tuning with MTP can be interpreted as pure control-variate fine-tuning of the main objective plus a deterministic drift in the direction of the mean speculative gradient.*

Corollary 1. *Assume that the exact mean $\bar{\nabla}^l_\theta S$ is available. Then the variance of Equation (3) is independent of λ:*

$$\mathrm{Var}\left(g^l_\theta(\mathcal{B};\lambda)\right) = \mathrm{Var}\left(\nabla^l_\theta M(\mathcal{B}) - \alpha^*_l(0)\left(\nabla^l_\theta S(\mathcal{B}) - \bar{\nabla}^l_\theta S\right)\right).$$

The proof can be found in Section B.6. Thus, the proposed estimator preserves the mean contribution of MTP while removing the additional sample-level variance introduced by $\lambda\nabla^l_\theta S(\mathcal{B})$.

3.4 Speculative Optimizers

Finally, we define stochastic optimizers that use the preceding theory to reduce the variance of stochastic gradients. We provide two formulations based on two popular choices for model fine-tuning: stochastic gradient descent (SGD) and Adam (Kingma and Ba, 2014). The main changes compared to the original optimizers are the use of a control-variate-corrected gradient and an EMA estimate of the mean speculative gradient $\bar{\nabla}^l_{\theta_t} S$.

For a mini-batch $\mathcal{B}_t \subset \mathcal{D}$, denote the layerwise mini-batch gradients by

$$\nabla^l_{\theta_t} M(\mathcal{B}_t) := \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} \nabla^l_{\theta_t} M(x),$$

$$\nabla^l_{\theta_t} S(\mathcal{B}_t) := \frac{1}{|\mathcal{B}_t|}\sum_{x\in\mathcal{B}_t} \nabla^l_{\theta_t} S(x).$$

We maintain an EMA estimate $\bar{s}^l_t$ of

$$\bar{\nabla}^l_{\theta_t} S := \mathbb{E}_{\mathcal{B}}\left[\nabla^l_{\theta_t} S(\mathcal{B})\right]$$

for each layer l. In the algorithms below, we use the base control-variate coefficient

$$\alpha^*_l := \alpha^*_l(0),$$

which is the optimal coefficient for the case $\lambda = 0$. For $\lambda > 0$, we use the equivalent form from Proposition 1, which absorbs the shift $\alpha^*_l(\lambda) = \alpha^*_l(0)+\lambda$ into the estimator. Thus, the implemented gradient is

$$g^l_t = \nabla^l_{\theta_t} M(\mathcal{B}_t) - \alpha^*_l\left(\nabla^l_{\theta_t} S(\mathcal{B}_t) - \bar{s}^l_t\right) + \lambda\bar{s}^l_t. \tag{4}$$

When $\bar{s}^l_t = \bar{\nabla}^l_{\theta_t} S$, this is equivalent to applying the optimal MTP-adjusted coefficient $\alpha^*_l(\lambda)$ in Equation (3). In practice, $\bar{s}^l_t$ is estimated online using EMA.

Now, we present our SPECSGD and SPECADAM optimizers in Algorithms 1 and 2, respectively. The blue lines mark the modifications relative to the original optimizers: the layerwise control-variate-corrected gradient and the EMA estimate of the speculative-gradient mean. In the algorithms, we use the equivalent expanded form from Proposition 1. Setting $\lambda = 0$ recovers the pure control-variate optimizer without MTP.

Algorithm 1 SPECSGD

Require: θ_0, learning rates $\{\eta_t\}$, MTP weight λ, CV EMA rate β_{cv}, coefficients $\{\alpha^*_l(0)\}_{l=1}^L$

$\bar{s}^l_0 \leftarrow 0$ for all layers l
for $t = 0,\dots,T-1$ **do**
  Sample mini-batch $\mathcal{B}_t \subset \mathcal{D}$
  for $l = 1,\dots,L$ **do**
    $m^l_t \leftarrow \nabla^l_{\theta_t} M(\mathcal{B}_t)$
    $s^l_t \leftarrow \nabla^l_{\theta_t} S(\mathcal{B}_t)$
    $g^l_t \leftarrow m^l_t - \alpha^*_l(0)\left(s^l_t - \bar{s}^l_t\right) + \lambda\bar{s}^l_t$
    $\bar{s}^l_{t+1} \leftarrow \beta_{\mathrm{cv}}\bar{s}^l_t + (1-\beta_{\mathrm{cv}})s^l_t$
    $\theta^l_{t+1} \leftarrow \theta^l_t - \eta_t g^l_t$
  end for
end for
return θ_T

Algorithm 2 SPECADAM

Require: θ_0, learning rates $\{\eta_t\}$, Adam rates $\beta_1, \beta_2, \epsilon$, MTP weight λ, CV EMA rate β_{cv}, coefficients $\{\alpha^*_l(0)\}_{l=1}^L$

$a^l_0 \leftarrow 0,\ q^l_0 \leftarrow 0,\ \bar{s}^l_0 \leftarrow 0$ for all layers l
for $t = 1,\dots,T$ **do**
  Sample mini-batch $\mathcal{B}_t \subset \mathcal{D}$
  for $l = 1,\dots,L$ **do**
    $m^l_t \leftarrow \nabla^l_{\theta_{t-1}} M(\mathcal{B}_t)$
    $s^l_t \leftarrow \nabla^l_{\theta_{t-1}} S(\mathcal{B}_t)$
    $g^l_t \leftarrow m^l_t - \alpha^*_l(0)\left(s^l_t - \bar{s}^l_t\right) + \lambda\bar{s}^l_t$
    $\bar{s}^l_{t+1} \leftarrow \beta_{\mathrm{cv}}\bar{s}^l_t + (1-\beta_{\mathrm{cv}})s^l_t$
    $a^l_t \leftarrow \beta_1 a^l_{t-1} + (1-\beta_1)g^l_t$
    $q^l_t \leftarrow \beta_2 q^l_{t-1} + (1-\beta_2)(g^l_t)^2$
    $\hat{a}^l_t \leftarrow a^l_t/(1-\beta_1^t)$
    $\hat{q}^l_t \leftarrow q^l_t/(1-\beta_2^t)$
    $\theta^l_t \leftarrow \theta^l_{t-1} - \eta_t\hat{a}^l_t/(\sqrt{\hat{q}^l_t} + \epsilon)$
  end for
end for
return θ_T

3.5 Convergence Guarantees

We now state convergence guarantees for the proposed SPECSGD method. The result below is stated for the idealized case where the exact speculative-gradient mean is available. In practice, this mean is replaced by the EMA estimate $\bar{s}_t^l$.

This introduces a small bias in the stochastic gradient estimator. We refer the reader to Section C for the theoretical guarantees in the EMA-based updates regime. SpecAdam with EMA is Adam applied to a biased control-variate gradient. The bias is exactly the same EMA tracking-error term as in SpecSGD, and Adam's momentum only smooths and rescales this error.

For fixed $\lambda \geq 0$, define the objective optimized by SPECSGD as

$$F_\lambda(\theta) := M(\mathcal{D};\theta) + \lambda S(\mathcal{D};\theta).$$

When $\lambda = 0$, this is the original supervised fine-tuning objective.

Theorem 1 (Nonconvex convergence of SPECSGD)**.** *Assume that F_λ is L-smooth and bounded below by $F_\lambda^\star$. Assume also that, conditioned on θ_t, the* SPECSGD *gradient g_t defined layerwise in Equation (4) satisfies*

$$\mathbb{E}[g_t \mid \theta_t] = \nabla F_\lambda(\theta_t),$$

and

$$\mathbb{E}\left[\|g_t - \nabla F_\lambda(\theta_t)\|_2^2 \mid \theta_t\right] \leq \sigma_{\text{Spec}}^2.$$

If SPECSGD *is run with a constant stepsize $\eta \leq 1/L$, then*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] \leq \frac{2\left(F_\lambda(\theta_0) - F_\lambda^\star\right)}{\eta T} + L\eta\sigma_{\text{Spec}}^2.$$

Consequently,

$$\min_{0\leq t<T} \mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] = \mathcal{O}\left(\frac{\sigma_{\text{Spec}}}{\sqrt{T}}\right).$$

In particular, choosing

$$\eta = \sqrt{\frac{2\left(F_\lambda(\theta_0) - F_\lambda^\star\right)}{L\sigma_{\text{Spec}}^2 T}} \leq \frac{1}{L}$$

gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] \leq 2\sqrt{\frac{2L\left(F_\lambda(\theta_0) - F_\lambda^\star\right)\sigma_{\text{Spec}}^2}{T}}.$$

The theorem shows that SPECSGD has the same $\mathcal{O}(1/\sqrt{T})$ nonconvex convergence rate as standard SGD, but with the stochastic-gradient variance σ_{Spec}^2 of the control-variate estimator. Thus, the improvement appears in the leading constant through variance reduction.

Proof of Theorem 1 is deferred to Section B.7.

4 Experiments

We evaluate four training settings:

1. (BASE) standard supervised fine-tuning baseline with the original gradient update;
2. (BASE+CV) control-variate gradient estimator without the MTP loss ($\lambda = 0, \alpha_l^* \neq 0$);
3. (MTP) MTP objective without control variates ($\lambda \neq 0, \alpha_l^* = 0$);
4. (MTP+CV) MTP objective with the estimator defined in Equation (4) ($\lambda \neq 0, \alpha_l^* \neq 0$).

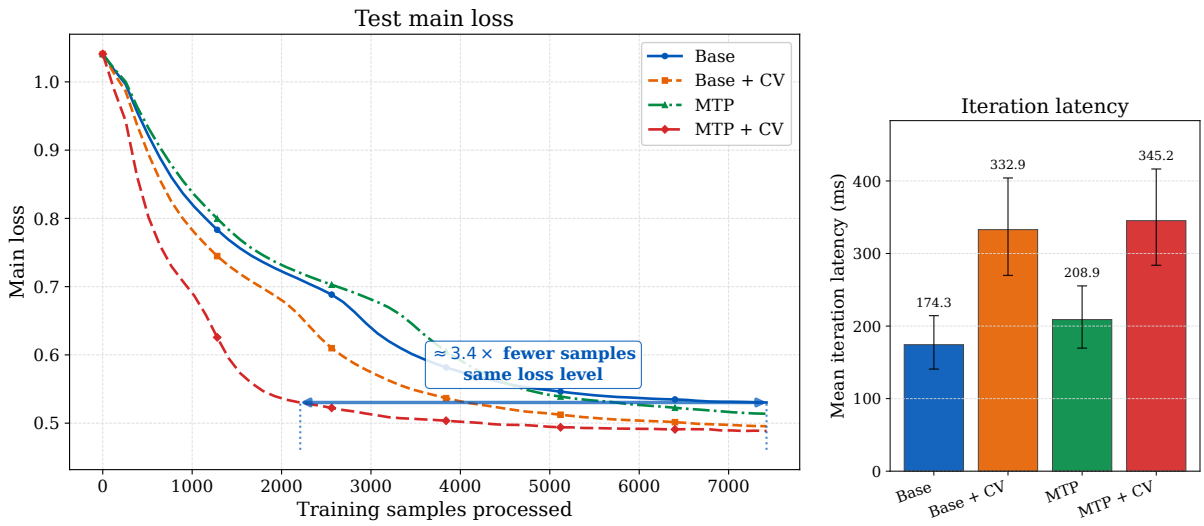
Since our method changes the stochastic-gradient variance, directly comparing all methods with the same learning rate may mix variance reduction with changes in effective step size. For SPECSGD, we therefore rescale the updates to match the variance of the standard baseline. Details are given in Section D.1. We report the main loss component, since it is the target SFT objective.

Alpha values were chosen using 128 prompts from the train dataset. For the stability of this choice refer to Section F.

We evaluate Qwen3.5-0.8B, Qwen3.5-4B, and Qwen3.5-9B using both full backbone fine-tuning and LoRA adapter tuning. These models include pre-trained MTP layers in their open-source repositories which we use as speculative head. The results for full fine-tuning of the 4B model are shown in Figure 2. We observe that, to reach the same loss value as the baseline, our method requires $3.4\times$ fewer iterations. Each iteration of our method is more expensive in wall-clock time, with a relative cost of $345.2/174.3 \approx 1.98$. Combining these two quantities, our method reaches the target loss $3.4/1.98 \approx$ **1.72**$\times$ faster than the baseline in wall-clock time. Importantly, our method not only reaches the baseline loss faster, but also converges to a **significantly lower final loss when trained on the same number of samples**. More wall-clock speedup results are provided in Table 1.

Additional results for other models and SPECADAM, are provided in the Section G.

All experiments were run on one RTX PRO 6000 GPU. We use batch size 4 for 0.8B and 4B models, and batch size 1 for the 9B model. We train on the full GSM8K training split. Learning curves use the cumulative number of training examples as the x-axis. For LoRA, we use rank 8 and scaling parameter $\alpha = 16$ for 4B model. For β_{cv} we used



(a) Main loss value for the test subset.

(b) Latency results for each regime.

Figure 2: Test results for Qwen3.5-4B full fine-tuning using SpecSGD algorithm on GSM8k.

Model	Tuning	Iter. reduction	Wall-clock speedup
0.8B	Full FT	3.3×	1.65×
0.8B	LoRA	1.8×	1.13×
4B	Full FT	3.4×	1.72×
4B	LoRA	2.4×	1.41×
9B	Full FT	3.2×	1.41×

Table 1: Wall-clock and iteration reduction results for SpecSGD

Model	Tuning	Batch size	Learning rate
0.8B	Full FT	4	1×10^{-5}
0.8B	LoRA	4	3×10^{-4}
4B	Full FT	4	2×10^{-5}
4B	LoRA	4	4×10^{-4}
9B	Full FT	1	1×10^{-5}

Table 2: Hyperparameters for SpecSGD

value 0.8, $\lambda = 0.9$. Further hyperparameter details are provided in Table 2.

We provide further discussion on practical implementation in Section A.

5 Conclusion and Future Work

In this work we presented speculative fine-tuning, a novel framework that uses speculative heads as a source of optimization information during supervised fine-tuning. The foundation of our method lies in the fact that the gradients induced by the speculative heads are strongly correlated with the gradients induced by the main head. We used this correlation to construct a control-variate estimator for the main stochastic gradient, which remains unbiased while reducing stochastic gradient variance. We further showed that this correlation is layer-dependent, motivating a layerwise version of the control-variate coefficient. We also provided a control-variate interpretation of multi-token prediction training, showing that the stochastic component of the auxiliary speculative gradient can be canceled while its expected gradient contribution is preserved. Finally, we combined the estimator with multi-token prediction training and introduced SpecSGD and SpecAdam, two optimizer variants that incorporate speculative gradients through a simple modification of the gradient update. Experiments showed that the number of iterations required for the loss to stabilize was reduced by up to $\approx 3.4\times$, while the wall-clock time reduced by up to $\approx 1.7\times$.

Our method opens a new point of view on speculative decoding, showing its applicability to the optimization process. This opens a scope of future work. Firstly, in our paper we did not explore how to reduce the noise introduced by the EMA estimator of the expected speculative gradient. Further research in this direction can reduce the overall variance even more. Secondly, while our method uses scalar layerwise control-variate coefficients, richer estimators could exploit blockwise, parameterwise, or low-rank covariance structure between $\nabla_\theta^l M$ and $\nabla_\theta^l S$. Lastly, in this paper we focused on supervised fine-tuning optimization. We believe that a similar variance-reduction approach can be applied to the pre-training stage and may make this stage cheaper.

Limitations

Our method relies on a positive correlation between main-head and speculative-head gradients. Although this correlation is observed in our experiments, its strength may vary across model architectures, speculative-head designs, datasets, and stages of fine-tuning. When the correlation is weak or unstable, the control-variate estimator may provide smaller gains.

In practice, the expected speculative-head gradient is approximated using an exponential moving average, and the layerwise control-variate coefficients are estimated during calibration rather than updated continuously. These approximations keep the method simple and efficient, but may introduce tracking error or suboptimal coefficients. Finally, the realized wall-clock speedup depends on the overhead of computing speculative-head gradients and may vary with hardware and implementation.

References

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D. Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple LLM inference acceleration framework with multiple decoding heads. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 5209–5235. PMLR.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Roziere, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. In *International Conference on Machine Learning*, pages 15706–15734. PMLR.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

S. S. Lavenberg and P. D. Welch. 1981. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27(3):322–335.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. 2024. Eagle: Speculative sampling requires rethinking feature uncertainty. In *International Conference on Machine Learning*, pages 28935–28948. PMLR.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Kai Lv, Yuqing Yang, Tengxiao Liu, Qipeng Guo, and Xipeng Qiu. 2024. Full parameter fine-tuning for large language models with limited resources. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8187–8198.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Mohammad Samragh, Arnav Kundu, David Harrison, Kumari Nishu, Devang Naik, Minsik Cho, and Mehrdad Farajtabar. 2025. Your llm knows the future: Uncovering its multi-token prediction potential. *arXiv preprint arXiv:2507.11851*.

Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. 2013. Variance reduction for stochastic gradient optimization. *Advances in neural information processing systems*, 26.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*.

A Discussion

Control-variate calculation latency The main overhead in our implementation comes from the need to perform two backward passes. This introduces both memory overhead and additional computational cost. One possible way to mitigate this effect is to update the EMA not at every step, but every K steps. This would reduce the accuracy of the control variate, but it would also reduce the average latency of each iteration.

Multiple speculative heads In Gloeckle et al. (2024) and in many speculative decoding methods, the authors use multiple speculation steps rather than a single one. Since the goal of our paper is to introduce a new research direction for speculative methods, we limit the number of speculation steps to one, showing that even in this setting the acceleration is significant. However, prior work has shown that increasing the number of speculation steps can improve either inference efficiency or pre-training quality. We believe that a similar argument applies to our setting by replacing $\nabla_\theta S(\mathcal{B})$ with $\nabla_\theta(S_1(\mathcal{B}) + \ldots + S_n(\mathcal{B}))$. Note that this modification does not require additional backward passes, since the speculative gradient can be computed in a single backward pass.

B Proofs

Throughout the appendix, expectations, variances, and covariances are taken with respect to the random mini-batch $\mathcal{B}$, conditional on the current parameter value θ whenever θ is fixed. For readability, we write

$$M_\theta(\mathcal{B}) := \nabla_\theta M(\mathcal{B}), \qquad S_\theta(\mathcal{B}) := \nabla_\theta S(\mathcal{B}),$$

and denote their means by

$$\bar{M}_\theta := \mathbb{E}[M_\theta(\mathcal{B})], \qquad \bar{S}_\theta := \mathbb{E}[S_\theta(\mathcal{B})].$$

We also use the centered variables

$$\widetilde{M}_\theta(\mathcal{B}) := M_\theta(\mathcal{B}) - \bar{M}_\theta, \qquad \widetilde{S}_\theta(\mathcal{B}) := S_\theta(\mathcal{B}) - \bar{S}_\theta.$$

B.1 Proof of Lemma 1

By definition,

$$g_\theta(\mathcal{B}) = M_\theta(\mathcal{B}) - \alpha\left(S_\theta(\mathcal{B}) - \bar{S}_\theta\right).$$

Taking expectation over the mini-batch gives

$$\begin{aligned}
\mathbb{E}[g_\theta(\mathcal{B})] &= \mathbb{E}[M_\theta(\mathcal{B})] - \alpha\left(\mathbb{E}[S_\theta(\mathcal{B})] - \bar{S}_\theta\right) \\
&= \bar{M}_\theta - \alpha\left(\bar{S}_\theta - \bar{S}_\theta\right) \\
&= \bar{M}_\theta.
\end{aligned}$$

Thus $g_\theta(\mathcal{B})$ is an unbiased estimator of the main-head population gradient $\bar{M}_\theta$.

B.2 Proof of Lemma 2

Using the centered notation above,

$$g_\theta(\mathcal{B}) - \bar{M}_\theta = \widetilde{M}_\theta(\mathcal{B}) - \alpha\widetilde{S}_\theta(\mathcal{B}).$$

Therefore,

$$\begin{aligned}
\mathrm{Var}(g_\theta(\mathcal{B})) &= \mathbb{E}\left[\left\|\widetilde{M}_\theta(\mathcal{B}) - \alpha\widetilde{S}_\theta(\mathcal{B})\right\|_2^2\right] \\
&= \mathbb{E}\left[\left\|\widetilde{M}_\theta(\mathcal{B})\right\|_2^2\right] - 2\alpha\mathbb{E}\left[\left\langle\widetilde{M}_\theta(\mathcal{B}), \widetilde{S}_\theta(\mathcal{B})\right\rangle\right] + \alpha^2\mathbb{E}\left[\left\|\widetilde{S}_\theta(\mathcal{B})\right\|_2^2\right] \\
&= \mathrm{Var}(M_\theta(\mathcal{B})) - 2\alpha\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B})) + \alpha^2\mathrm{Var}(S_\theta(\mathcal{B})).
\end{aligned}$$

This is a quadratic function of α with positive leading coefficient $\mathrm{Var}(S_\theta(\mathcal{B})) > 0$. Hence its unique minimizer is obtained by setting the derivative to zero:

$$-2\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B})) + 2\alpha\mathrm{Var}(S_\theta(\mathcal{B})) = 0.$$

Thus,

$$\alpha^* = \frac{\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B}))}{\mathrm{Var}(S_\theta(\mathcal{B}))}.$$

Using the definition of correlation,

$$\rho = \frac{\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B}))}{\sqrt{\mathrm{Var}(M_\theta(\mathcal{B}))\mathrm{Var}(S_\theta(\mathcal{B}))}},$$

we equivalently obtain

$$\alpha^* = \rho\sqrt{\frac{\mathrm{Var}(M_\theta(\mathcal{B}))}{\mathrm{Var}(S_\theta(\mathcal{B}))}}.$$

Substituting α^* into the quadratic variance expression gives

$$\begin{aligned}\mathrm{Var}(g_\theta(\mathcal{B})) &= \mathrm{Var}(M_\theta(\mathcal{B})) - \frac{\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B}))^2}{\mathrm{Var}(S_\theta(\mathcal{B}))} \\ &= \mathrm{Var}(M_\theta(\mathcal{B})) - \rho^2 \mathrm{Var}(M_\theta(\mathcal{B})) \\ &= (1-\rho^2)\mathrm{Var}(M_\theta(\mathcal{B})).\end{aligned}$$

This proves the claim.

B.3 Derivation of the speculative gradient correlation

Let

$$M_\theta(\mathcal{B}) := \nabla_\theta M(\mathcal{B}), \qquad S_\theta(\mathcal{B}) := \nabla_\theta S(\mathcal{B}),$$

and define the combined MTP gradient

$$A_\theta(\mathcal{B}) := M_\theta(\mathcal{B}) + \lambda S_\theta(\mathcal{B}).$$

Let

$$\sigma_M^2 := \mathrm{Var}(M_\theta(\mathcal{B})), \qquad \sigma_S^2 := \mathrm{Var}(S_\theta(\mathcal{B})),$$

and assume $\sigma_M, \sigma_S > 0$. By the definition of ρ,

$$\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B})) = \rho\sigma_M\sigma_S.$$

Using bilinearity of covariance, we obtain

$$\begin{aligned}\mathrm{Cov}(A_\theta(\mathcal{B}), S_\theta(\mathcal{B})) &= \mathrm{Cov}(M_\theta(\mathcal{B}) + \lambda S_\theta(\mathcal{B}), S_\theta(\mathcal{B})) \\ &= \mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B})) + \lambda \mathrm{Var}(S_\theta(\mathcal{B})) \\ &= \rho\sigma_M\sigma_S + \lambda\sigma_S^2 \\ &= \sigma_S\left(\rho\sigma_M + \lambda\sigma_S\right).\end{aligned}$$

Similarly,

$$\begin{aligned}\mathrm{Var}(A_\theta(\mathcal{B})) &= \mathrm{Var}(M_\theta(\mathcal{B}) + \lambda S_\theta(\mathcal{B})) \\ &= \mathrm{Var}(M_\theta(\mathcal{B})) + 2\lambda\mathrm{Cov}(M_\theta(\mathcal{B}), S_\theta(\mathcal{B})) + \lambda^2\mathrm{Var}(S_\theta(\mathcal{B})) \\ &= \sigma_M^2 + 2\lambda\rho\sigma_M\sigma_S + \lambda^2\sigma_S^2.\end{aligned}$$

Therefore,

$$\begin{aligned}&\mathrm{Corr}\left(M_\theta(\mathcal{B}) + \lambda S_\theta(\mathcal{B}), S_\theta(\mathcal{B})\right) \\ &= \frac{\mathrm{Cov}(A_\theta(\mathcal{B}), S_\theta(\mathcal{B}))}{\sqrt{\mathrm{Var}(A_\theta(\mathcal{B}))\mathrm{Var}(S_\theta(\mathcal{B}))}} \\ &= \frac{\sigma_S\left(\rho\sigma_M + \lambda\sigma_S\right)}{\sigma_S\sqrt{\sigma_M^2 + 2\lambda\rho\sigma_M\sigma_S + \lambda^2\sigma_S^2}} \\ &= \frac{\rho\sigma_M + \lambda\sigma_S}{\sqrt{\sigma_M^2 + 2\lambda\rho\sigma_M\sigma_S + \lambda^2\sigma_S^2}}.\end{aligned}$$

Since $\rho > 0$, $\sigma_M > 0$, $\sigma_S > 0$, and $\lambda \geq 0$, the numerator is positive. The denominator is also positive whenever the combined gradient has nonzero variance. Hence,

$$\mathrm{Corr}\left(M_\theta(\mathcal{B}) + \lambda S_\theta(\mathcal{B}), S_\theta(\mathcal{B})\right) > 0.$$

B.4 Proof of Lemma 3

Fix a layer l and a parameter value θ. For readability, write

$$M^l := \nabla_\theta^l M(\mathcal{B}), \qquad S^l := \nabla_\theta^l S(\mathcal{B}),$$

and let

$$\bar{S}^l := \mathbb{E}[S^l].$$

For a fixed value of λ, the stochastic gradient of the MTP objective in layer l is

$$M^l + \lambda S^l.$$

Using S^l as a control variate, the corresponding estimator with coefficient α is

$$g_\alpha^l(\mathcal{B};\lambda) = M^l + \lambda S^l - \alpha\left(S^l - \bar{S}^l\right).$$

Since subtracting the mean does not affect covariance with the centered control variate, the variance-minimizing scalar coefficient is

$$\alpha_l^*(\lambda) = \frac{\mathrm{Cov}(M^l + \lambda S^l, S^l)}{\mathrm{Var}(S^l)}.$$

By bilinearity of covariance,

$$\mathrm{Cov}(M^l + \lambda S^l, S^l) = \mathrm{Cov}(M^l, S^l) + \lambda \mathrm{Var}(S^l).$$

Therefore,

$$\begin{aligned}\alpha_l^*(\lambda) &= \frac{\mathrm{Cov}(M^l, S^l) + \lambda \mathrm{Var}(S^l)}{\mathrm{Var}(S^l)} \\ &= \frac{\mathrm{Cov}(M^l, S^l)}{\mathrm{Var}(S^l)} + \lambda.\end{aligned}$$

The first term is exactly the optimal control-variate coefficient for $\lambda = 0$, namely

$$\alpha_l^*(0) = \frac{\mathrm{Cov}(M^l, S^l)}{\mathrm{Var}(S^l)}.$$

Hence

$$\alpha_l^*(\lambda) = \alpha_l^*(0) + \lambda,$$

as claimed.

B.5 Proof of Proposition 1

Proof. Using Lemma 3, we have

$$\begin{aligned}g_\theta^l(\mathcal{B};\lambda) &= \left(\nabla_\theta^l M(\mathcal{B}) + \lambda \nabla_\theta^l S(\mathcal{B})\right) - (\alpha_l^*(0) + \lambda)\left(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S\right) \\ &= \nabla_\theta^l M(\mathcal{B}) + \lambda \bar{\nabla}_\theta^l S - \alpha_l^*(0)\left(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S\right).\end{aligned}$$

□

B.6 Proof of Corollary 1

Proof. By the previous proposition,

$$g_\theta^l(\mathcal{B};\lambda) = \nabla_\theta^l M(\mathcal{B}) - \alpha_l^*(0)\left(\nabla_\theta^l S(\mathcal{B}) - \bar{\nabla}_\theta^l S\right) + \lambda \bar{\nabla}_\theta^l S.$$

The last term is deterministic with respect to the random mini-batch $\mathcal{B}$, so it does not affect the variance.

□

B.7 Proof of Theorem 1

Let

$$\Delta_0 := F_\lambda(\theta_0) - F_\lambda^\star.$$

Since F_λ is L-smooth, for any update

$$\theta_{t+1} = \theta_t - \eta g_t$$

we have

$$\begin{aligned} F_\lambda(\theta_{t+1}) &\le F_\lambda(\theta_t) + \langle \nabla F_\lambda(\theta_t), \theta_{t+1} - \theta_t \rangle + \frac{L}{2} \|\theta_{t+1} - \theta_t\|_2^2 \\ &= F_\lambda(\theta_t) - \eta \langle \nabla F_\lambda(\theta_t), g_t \rangle + \frac{L\eta^2}{2} \|g_t\|_2^2. \end{aligned}$$

Taking conditional expectation with respect to the mini-batch randomness, conditioned on θ_t, gives

$$\mathbb{E}[F_\lambda(\theta_{t+1}) \mid \theta_t] \le F_\lambda(\theta_t) - \eta \langle \nabla F_\lambda(\theta_t), \mathbb{E}[g_t \mid \theta_t] \rangle + \frac{L\eta^2}{2} \mathbb{E}[\|g_t\|_2^2 \mid \theta_t].$$

By the unbiasedness assumption,

$$\mathbb{E}[g_t \mid \theta_t] = \nabla F_\lambda(\theta_t).$$

Moreover,

$$g_t = \nabla F_\lambda(\theta_t) + (g_t - \nabla F_\lambda(\theta_t)),$$

and the cross term vanishes conditionally because

$$\mathbb{E}[g_t - \nabla F_\lambda(\theta_t) \mid \theta_t] = 0.$$

Therefore,

$$\begin{aligned} \mathbb{E}[\|g_t\|_2^2 \mid \theta_t] &= \|\nabla F_\lambda(\theta_t)\|_2^2 + \mathbb{E}\left[\|g_t - \nabla F_\lambda(\theta_t)\|_2^2 \mid \theta_t\right] \\ &\le \|\nabla F_\lambda(\theta_t)\|_2^2 + \sigma_{\text{Spec}}^2. \end{aligned}$$

Substituting this into the smoothness bound yields

$$\begin{aligned} \mathbb{E}[F_\lambda(\theta_{t+1}) \mid \theta_t] &\le F_\lambda(\theta_t) - \eta \|\nabla F_\lambda(\theta_t)\|_2^2 + \frac{L\eta^2}{2} \|\nabla F_\lambda(\theta_t)\|_2^2 + \frac{L\eta^2}{2} \sigma_{\text{Spec}}^2 \\ &= F_\lambda(\theta_t) - \eta \left(1 - \frac{L\eta}{2}\right) \|\nabla F_\lambda(\theta_t)\|_2^2 + \frac{L\eta^2}{2} \sigma_{\text{Spec}}^2. \end{aligned}$$

Since $\eta \le 1/L$, we have

$$1 - \frac{L\eta}{2} \ge \frac{1}{2}.$$

Hence

$$\mathbb{E}[F_\lambda(\theta_{t+1}) \mid \theta_t] \le F_\lambda(\theta_t) - \frac{\eta}{2} \|\nabla F_\lambda(\theta_t)\|_2^2 + \frac{L\eta^2}{2} \sigma_{\text{Spec}}^2.$$

Taking total expectation gives

$$\mathbb{E}[F_\lambda(\theta_{t+1})] \le \mathbb{E}[F_\lambda(\theta_t)] - \frac{\eta}{2} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] + \frac{L\eta^2}{2} \sigma_{\text{Spec}}^2.$$

Rearranging,

$$\frac{\eta}{2} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] \le \mathbb{E}[F_\lambda(\theta_t)] - \mathbb{E}[F_\lambda(\theta_{t+1})] + \frac{L\eta^2}{2} \sigma_{\text{Spec}}^2.$$

Summing from $t = 0$ to $T - 1$, we obtain

$$\frac{\eta}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] \leq F_\lambda(\theta_0) - \mathbb{E}[F_\lambda(\theta_T)] + \frac{L\eta^2 T}{2} \sigma_{\text{Spec}}^2.$$

Since F_λ is bounded below by $F_\lambda^\star$,

$$F_\lambda(\theta_0) - \mathbb{E}[F_\lambda(\theta_T)] \leq F_\lambda(\theta_0) - F_\lambda^\star = \Delta_0.$$

Therefore,

$$\frac{\eta}{2} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] \leq \Delta_0 + \frac{L\eta^2 T}{2} \sigma_{\text{Spec}}^2.$$

Dividing by $\eta T/2$ gives

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] \leq \frac{2\Delta_0}{\eta T} + L\eta\sigma_{\text{Spec}}^2.$$

This proves the first claimed bound.

Since the minimum is bounded above by the average,

$$\min_{0 \leq t < T} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2] \leq \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla F_\lambda(\theta_t)\|_2^2].$$

Thus the same upper bound applies to the minimum.

Finally, choose

$$\eta = \sqrt{\frac{2\Delta_0}{L\sigma_{\text{Spec}}^2 T}},$$

assuming this choice satisfies $\eta \leq 1/L$. Substituting this value into the previous bound gives

$$\begin{aligned} \frac{2\Delta_0}{\eta T} + L\eta\sigma_{\text{Spec}}^2 &= \sqrt{\frac{2L\Delta_0\sigma_{\text{Spec}}^2}{T}} + \sqrt{\frac{2L\Delta_0\sigma_{\text{Spec}}^2}{T}} \\ &= 2\sqrt{\frac{2L\Delta_0\sigma_{\text{Spec}}^2}{T}}. \end{aligned}$$

Hence

$$\frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] \leq 2\sqrt{\frac{2L\left(F_\lambda(\theta_0) - F_\lambda^\star\right)\sigma_{\text{Spec}}^2}{T}}.$$

Consequently,

$$\min_{0 \leq t < T} \mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] = \mathcal{O}\left(\frac{\sigma_{\text{Spec}}}{\sqrt{T}}\right).$$

This completes the proof.

C Convergence with EMA-Estimated Speculative Means

Here we show how the result for SPECSGD changes when this mean is replaced by the EMA estimate used in the implementation.

For each layer l, let

$$s_t^l := \nabla_{\theta_t}^l S(\mathcal{B}_t), \qquad m_t^l := \nabla_{\theta_t}^l M(\mathcal{B}_t),$$

and define the speculative-gradient mean

$$\mu_t^l := \mathbb{E}_{\mathcal{B}}\left[\nabla_{\theta_t}^l S(\mathcal{B}) \mid \theta_t\right].$$

The ideal exact-mean estimator is

$$g_t^l = m_t^l - \alpha_l\left(s_t^l - \mu_t^l\right) + \lambda\mu_t^l,$$

where $\alpha_l = \alpha_l^*(0)$. The practical EMA-based estimator used by SPECSGD is

$$\widetilde{g}_t^l = m_t^l - \alpha_l\left(s_t^l - \bar{s}_t^l\right) + \lambda\bar{s}_t^l,$$

where

$$\bar{s}_{t+1}^l = \beta_{\mathrm{cv}}\bar{s}_t^l + (1-\beta_{\mathrm{cv}})s_t^l.$$

Define the *EMA tracking error*

$$e_t^l := \bar{s}_t^l - \mu_t^l.$$

Lemma 4 (Bias induced by EMA tracking error)**.** *For every layer l,*

$$\widetilde{g}_t^l = g_t^l + (\alpha_l + \lambda)e_t^l.$$

Consequently, if the exact-mean estimator satisfies

$$\mathbb{E}[g_t \mid \theta_t] = \nabla F_\lambda(\theta_t),$$

then the EMA-based estimator satisfies

$$\mathbb{E}[\widetilde{g}_t \mid \theta_t] = \nabla F_\lambda(\theta_t) + b_t,$$

where the layerwise bias is

$$b_t^l = (\alpha_l + \lambda)e_t^l.$$

In particular, if

$$c_{\max} := \max_l |\alpha_l + \lambda|,$$

then

$$\|b_t\|_2^2 \le c_{\max}^2 \sum_{l=1}^{L} \|e_t^l\|_2^2.$$

Proof. Subtracting the exact-mean estimator from the EMA-based estimator gives

$$\begin{aligned}\widetilde{g}_t^l - g_t^l &= \alpha_l(\bar{s}_t^l - \mu_t^l) + \lambda(\bar{s}_t^l - \mu_t^l)\\ &= (\alpha_l + \lambda)e_t^l.\end{aligned}$$

The remaining statements follow immediately by taking conditional expectations and summing over layers. □

We next give a simple bound on the EMA tracking error. Write

$$s_t^l = \mu_t^l + \xi_t^l,$$

where ξ_t^l is the mini-batch noise in the speculative gradient.

Lemma 5 (EMA tracking error)**.** *Assume that, for every layer l,*

$$\mathbb{E}[\xi_t^l \mid \theta_t] = 0, \qquad \mathbb{E}\|\xi_t^l\|_2^2 \le \sigma_{S,l}^2.$$

Assume also that the population speculative-gradient mean changes slowly:

$$\left(\mathbb{E}\left\|\mu_{t+1}^l - \mu_t^l\right\|_2^2\right)^{1/2} \le \delta_l.$$

Then, for any $\beta_{\mathrm{cv}} \in [0,1)$,

$$\left(\mathbb{E}\|e_t^l\|_2^2\right)^{1/2} \le \beta_{\mathrm{cv}}^t \|e_0^l\|_2 + \sqrt{\frac{1-\beta_{\mathrm{cv}}}{1+\beta_{\mathrm{cv}}}}\sigma_{S,l} + \frac{\delta_l}{1-\beta_{\mathrm{cv}}}.$$

Proof. Using the EMA update and the definition of e_t^l, we have

$$\begin{aligned} e_{t+1}^l &= \bar{s}_{t+1}^l - \mu_{t+1}^l \\ &= \beta_{\mathrm{cv}}\bar{s}_t^l + (1-\beta_{\mathrm{cv}})s_t^l - \mu_{t+1}^l \\ &= \beta_{\mathrm{cv}} e_t^l + (1-\beta_{\mathrm{cv}})\xi_t^l + (\mu_t^l - \mu_{t+1}^l). \end{aligned}$$

Unrolling this recursion gives

$$e_t^l = \beta_{\mathrm{cv}}^t e_0^l + (1-\beta_{\mathrm{cv}})\sum_{i=0}^{t-1}\beta_{\mathrm{cv}}^{t-1-i}\xi_i^l + \sum_{i=0}^{t-1}\beta_{\mathrm{cv}}^{t-1-i}(\mu_i^l - \mu_{i+1}^l).$$

The stochastic-noise term has second moment bounded by

$$(1-\beta_{\mathrm{cv}})^2\sum_{i=0}^{t-1}\beta_{\mathrm{cv}}^{2(t-1-i)}\sigma_{S,l}^2 \le \frac{1-\beta_{\mathrm{cv}}}{1+\beta_{\mathrm{cv}}}\sigma_{S,l}^2.$$

The drift term has root second moment at most

$$\sum_{i=0}^{t-1}\beta_{\mathrm{cv}}^{t-1-i}\delta_l \le \frac{\delta_l}{1-\beta_{\mathrm{cv}}}.$$

Combining these bounds with the triangle inequality in root mean square gives the result. □

We now state the convergence guarantee for the practical EMA-based update.

Theorem 2 (Nonconvex convergence with EMA-estimated control variate)**.** *Assume that F_λ is L_F-smooth and bounded below by $F_\lambda^\star$. Assume that the exact-mean estimator g_t satisfies*

$$\mathbb{E}[g_t \mid \theta_t] = \nabla F_\lambda(\theta_t)$$

and

$$\mathbb{E}\left[\|g_t - \nabla F_\lambda(\theta_t)\|_2^2 \mid \theta_t\right] \le \sigma_{\mathrm{CV}}^2.$$

Let SPECSGD *use the EMA-based estimator $\widetilde{g}_t$ and update*

$$\theta_{t+1} = \theta_t - \eta\widetilde{g}_t.$$

If $\eta \le 1/(4L_F)$, then

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F_\lambda(\theta_t)\|_2^2\right] \le \frac{4\left(F_\lambda(\theta_0) - F_\lambda^\star\right)}{\eta T} + 2L_F\eta\sigma_{\mathrm{CV}}^2 + 3c_{\max}^2\frac{1}{T}\sum_{t=0}^{T-1}\sum_{l=1}^{L}\mathbb{E}\|e_t^l\|_2^2.$$

Proof. Let

$$G_t := \nabla F_\lambda(\theta_t).$$

By Lemma 4, we can write

$$\widetilde{g}_t = g_t + b_t,$$

where b_t is the EMA-induced bias. By smoothness,

$$F_\lambda(\theta_{t+1}) \le F_\lambda(\theta_t) - \eta\langle G_t, \widetilde{g}_t\rangle + \frac{L_F\eta^2}{2}\|\widetilde{g}_t\|_2^2.$$

Taking conditional expectation and using $\mathbb{E}[g_t \mid \theta_t] = G_t$, we obtain

$$\mathbb{E}[F_\lambda(\theta_{t+1}) \mid \theta_t] \le F_\lambda(\theta_t) - \eta\|G_t\|_2^2 - \eta\langle G_t, b_t\rangle + \frac{L_F\eta^2}{2}\mathbb{E}[\|g_t + b_t\|_2^2 \mid \theta_t].$$

Using

$$-\langle G_t, b_t\rangle \le \frac{1}{2}\|G_t\|_2^2 + \frac{1}{2}\|b_t\|_2^2$$

and

$$\mathbb{E}[\|g_t + b_t\|_2^2 \mid \theta_t] \le 2\|G_t\|_2^2 + 2\|b_t\|_2^2 + \sigma_{\mathrm{CV}}^2,$$

we get

$$\mathbb{E}[F_\lambda(\theta_{t+1}) \mid \theta_t] \le F_\lambda(\theta_t) - \eta\left(\frac{1}{2} - L_F\eta\right)\|G_t\|_2^2 + \left(\frac{\eta}{2} + L_F\eta^2\right)\|b_t\|_2^2 + \frac{L_F\eta^2}{2}\sigma_{\mathrm{CV}}^2.$$

Since $\eta \le 1/(4L_F)$,

$$\frac{1}{2} - L_F\eta \ge \frac{1}{4} \qquad \text{and} \qquad \frac{\eta}{2} + L_F\eta^2 \le \frac{3\eta}{4}.$$

Therefore,

$$\frac{\eta}{4}\mathbb{E}\|G_t\|_2^2 \le \mathbb{E}[F_\lambda(\theta_t) - F_\lambda(\theta_{t+1})] + \frac{3\eta}{4}\mathbb{E}\|b_t\|_2^2 + \frac{L_F\eta^2}{2}\sigma_{\mathrm{CV}}^2.$$

Summing over $t = 0, \dots, T-1$, using the lower bound $F_\lambda^\star$, and applying

$$\|b_t\|_2^2 \le c_{\max}^2 \sum_{l=1}^{L} \|e_t^l\|_2^2$$

from Lemma 4 gives the stated result. □

Combining Theorem 2 with Lemma 5 shows that the practical EMA implementation converges to a neighborhood, size of which is controlled by the EMA tracking error. The error decreases when the speculative-gradient noise is small and when the population speculative-gradient mean changes slowly during fine-tuning.

D Comparison Details

D.1 Fair SGD Comparison

The update rules considered in our work modify the stochastic gradient in two different ways. First, control variates reduce the mini-batch gradient variance. Second, MTP changes the mean optimization direction by adding an auxiliary speculative-gradient component. Therefore, directly comparing raw SGD, MTP, and control-variate updates under the same learning rate can be misleading: one method may appear better simply because its stochastic update has a different variance scale.

To isolate the effect of the update direction from trivial changes in stochastic-gradient variance, we use a variance-matched SGD protocol. All methods are trained from the same initialization, use the same mini-batch order, the same base learning rate η, and the same optimizer schedule. The only difference between methods is the stochastic gradient estimator passed to SGD.

For a mini-batch $\mathcal{B}_t$, let

$$m_t^l := \nabla_{\theta^l} M(\mathcal{B}_t), \qquad s_t^l := \nabla_{\theta^l} S(\mathcal{B}_t)$$

denote the main-head and speculative-head gradients for layer l. Let

$$\mu_s^l := \mathbb{E}[s_t^l]$$

denote the population mean speculative gradient, approximated in practice by an EMA estimate. The control-variate coefficient is denoted by α_l.

We compare four SGD updates:

$$g_{\text{Base}}^l = m_t^l,$$

$$g_{\text{Base+CV}}^l = m_t^l - \alpha_l \left(s_t^l - \mu_s^l \right),$$

$$g_{\text{MTP}}^l = m_t^l + \lambda s_t^l,$$

and

$$g_{\text{MTP+CV}}^l = m_t^l - \alpha_l \left(s_t^l - \mu_s^l \right) + \lambda \mu_s^l.$$

The last expression is the control-variate form of the MTP update. It preserves the mean MTP contribution $\lambda \mu_s^l$ while canceling the stochastic part of the auxiliary speculative gradient.

For each layer, define

$$V_m^l := \text{Var}(m_t^l), \qquad V_s^l := \text{Var}(s_t^l), \qquad C_{ms}^l := \text{Cov}(m_t^l, s_t^l),$$

where variance and covariance are computed using the Euclidean inner product. The raw variances of the four estimators are

$$\text{Var}(g_{\text{Base}}^l) = V_m^l,$$

$$\text{Var}(g_{\text{Base+CV}}^l) = V_m^l - 2\alpha_l C_{ms}^l + \alpha_l^2 V_s^l,$$

$$\text{Var}(g_{\text{MTP}}^l) = V_m^l + 2\lambda C_{ms}^l + \lambda^2 V_s^l,$$

and

$$\text{Var}(g_{\text{MTP+CV}}^l) = V_m^l - 2\alpha_l C_{ms}^l + \alpha_l^2 V_s^l.$$

The final equality holds because $\lambda \mu_s^l$ is deterministic with respect to the sampled mini-batch and therefore does not affect the mini-batch variance.

We then define fixed layerwise variance-matching coefficients

$$c_{\text{Base}}^l := 1,$$

$$c_{\text{Base+CV}}^l := \sqrt{\frac{V_m^l}{V_m^l - 2\alpha_l C_{ms}^l + \alpha_l^2 V_s^l}},$$

$$c^l_{\mathrm{MTP}} := \sqrt{\frac{V^l_m}{V^l_m + 2\lambda C^l_{ms} + \lambda^2 V^l_s}},$$

and

$$c^l_{\mathrm{MTP+CV}} := c^l_{\mathrm{Base+CV}}.$$

These coefficients are estimated once on a fixed calibration subset before training and then held fixed throughout the run. They are not tuned on validation performance and do not constitute a learning-rate search.

The actual SGD updates used in the comparison are therefore

$$\theta^l_{t+1} = \theta^l_t - \eta c^l_{\mathrm{Base}} g^l_{\mathrm{Base}},$$

$$\theta^l_{t+1} = \theta^l_t - \eta c^l_{\mathrm{Base+CV}} g^l_{\mathrm{Base+CV}},$$

$$\theta^l_{t+1} = \theta^l_t - \eta c^l_{\mathrm{MTP}} g^l_{\mathrm{MTP}},$$

and

$$\theta^l_{t+1} = \theta^l_t - \eta c^l_{\mathrm{MTP+CV}} g^l_{\mathrm{MTP+CV}}.$$

By construction, all four methods have the same calibrated layerwise stochastic update variance:

$$\mathrm{Var}\left(-\eta c^l_k g^l_k\right) = \mathrm{Var}\left(-\eta m^l_t\right), \qquad k \in \{\mathrm{Base}, \mathrm{Base+CV}, \mathrm{MTP}, \mathrm{MTP+CV}\}.$$

Thus, each method is compared under the same stochastic-noise budget per layer. This prevents the comparison from being explained solely by one method using a larger or smaller stochastic update variance.

D.2 Fair Adam Comparison

For Adam-based experiments, we use a different fairness protocol from the variance-matched SGD comparison. In SGD, rescaling the stochastic gradient directly rescales the parameter update, so matching the variance of the gradient estimator has a direct interpretation in terms of update variance. This is not the case for Adam, because the update is normalized by an adaptive second-moment estimate. Therefore, matching the raw variance of the gradient estimator does not necessarily match the variance or magnitude of the actual Adam update.

For a layer l, Adam applies the update

$$\Delta^l_t = -\eta_t \frac{\hat{a}^l_t}{\sqrt{\hat{q}^l_t} + \epsilon},$$

where $\hat{a}^l_t$ and $\hat{q}^l_t$ are the bias-corrected first- and second-moment estimates computed from the gradient estimator g^l_t. If the gradient sequence in a layer is multiplied by a fixed positive constant c_l, then, ignoring the ϵ term, the first moment scales approximately as c_l and the second moment scales approximately as c_l^2:

$$\hat{a}^l_t(c_l g) \approx c_l \hat{a}^l_t(g), \qquad \hat{q}^l_t(c_l g) \approx c_l^2 \hat{q}^l_t(g).$$

Thus,

$$\frac{\hat{a}^l_t(c_l g)}{\sqrt{\hat{q}^l_t(c_l g)} + \epsilon} \approx \frac{c_l \hat{a}^l_t(g)}{c_l \sqrt{\hat{q}^l_t(g)} + \epsilon} \approx \frac{\hat{a}^l_t(g)}{\sqrt{\hat{q}^l_t(g)}}.$$

This scale-cancellation property means that the SGD variance-matching coefficient would mostly be absorbed by Adam's adaptive normalization. As a result, applying the same variance-matching correction used for SGD would not produce an equal-update-variance comparison and could introduce artifacts through the ϵ term, gradient clipping, or AdamW weight decay.

We therefore compare Adam-based methods under a fixed optimizer protocol. All methods are trained from the same initialization, use the same mini-batch order, the same number of training examples or tokens, and the same Adam or AdamW hyperparameters, including learning rate schedule, warmup, β_1, β_2, ϵ, weight decay, and gradient clipping. The only difference between methods is the gradient estimator passed to Adam.

E Correlation Preservation

Our objective is to train the main model head to predict the target samples. Since fine-tuning updates the model's internal representations, the correlation between the main-head and speculative-head gradients could, in principle, decrease over the training. However, we empirically observe that this correlation decreases only modestly and remains positive until the loss stabilizes near the end of fine-tuning. The empirical results are shown in Figure 3. This supports the use of fixed layerwise coefficients α_l throughout fine-tuning, rather than recomputing them online at every training step.

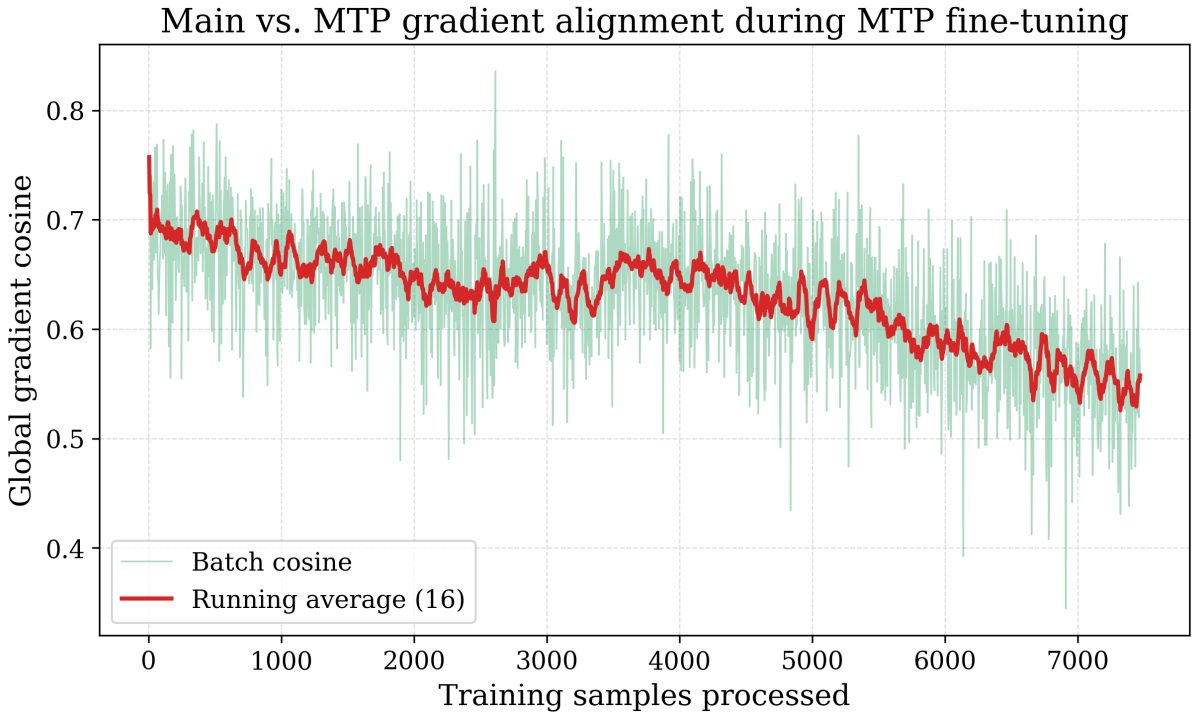


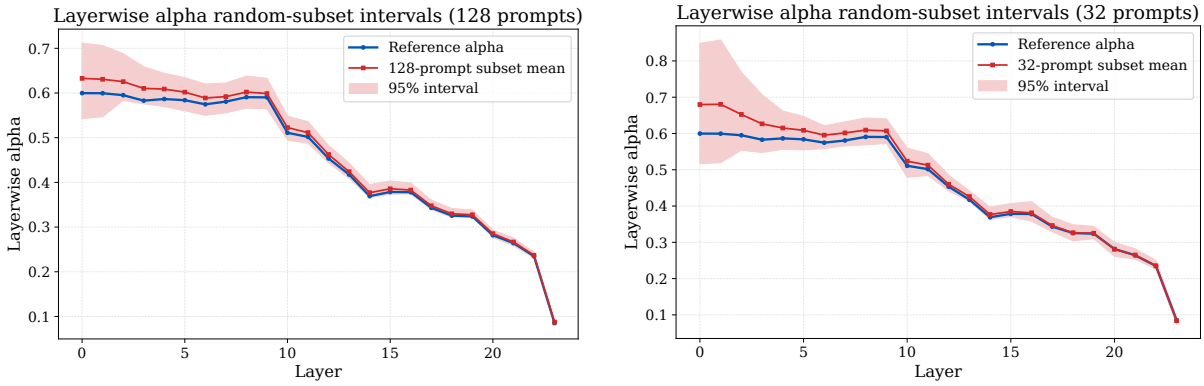
Figure 3: Correlation between the main-head and speculative-head gradients during full fine-tuning of the Qwen3.5-0.8B model.

F Alphas Estimation

In this section we show that the layerwise alpha schedule can be estimated using only a relatively small number of calibration prompts. Figures 4a and 4b compare the actual alpha values (computed over the whole dataset) with estimates obtained from random subsets of prompts, using either 32 or 128 prompts. In both cases, the subset mean follows the same overall layerwise structure as the reference curve.

The 32-prompt setting already provides a useful approximation of the reference alpha profile. Although the confidence intervals are wider, especially in the first several layers, the estimated mean remains close to the reference values across most of the network. Using 128 prompts further improves the stability of the estimate. The confidence intervals are narrower, and the subset mean aligns more closely with the reference alpha curve. However, the difference between the 32-prompt and 128-prompt estimates is relatively small compared with the overall layerwise variation.

In practice, a small subset such as 32 prompts can already provide a reasonable alpha schedule, while a larger subset such as 128 prompts offers a more stable estimate when additional calibration cost is acceptable.



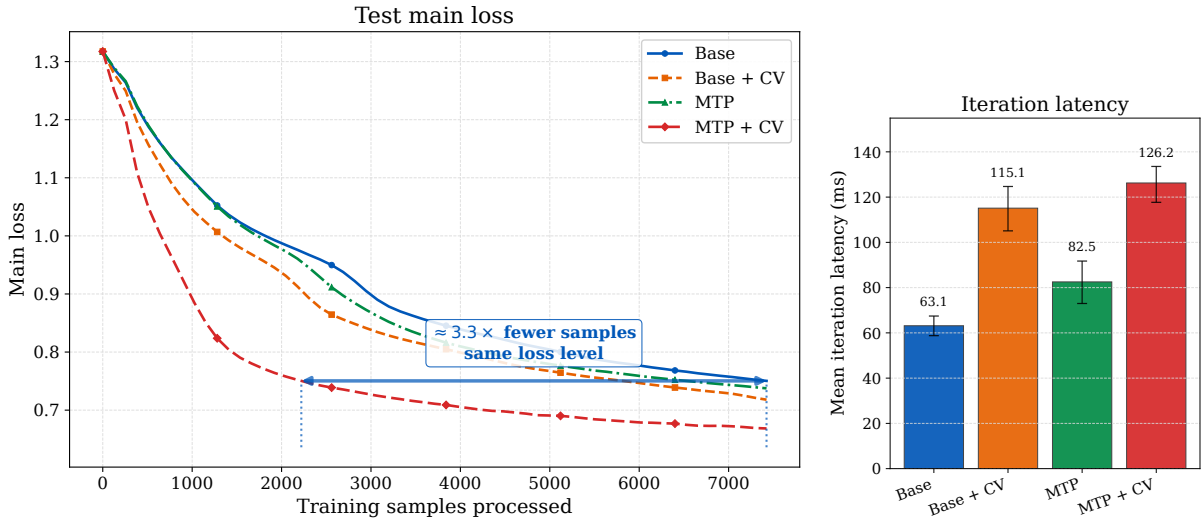
(a) Random-subset estimate using 128 prompts.

(b) Random-subset estimate using 32 prompts.

Figure 4: Layerwise alpha estimates obtained from random subsets of calibration prompts.

G Additional Experiments

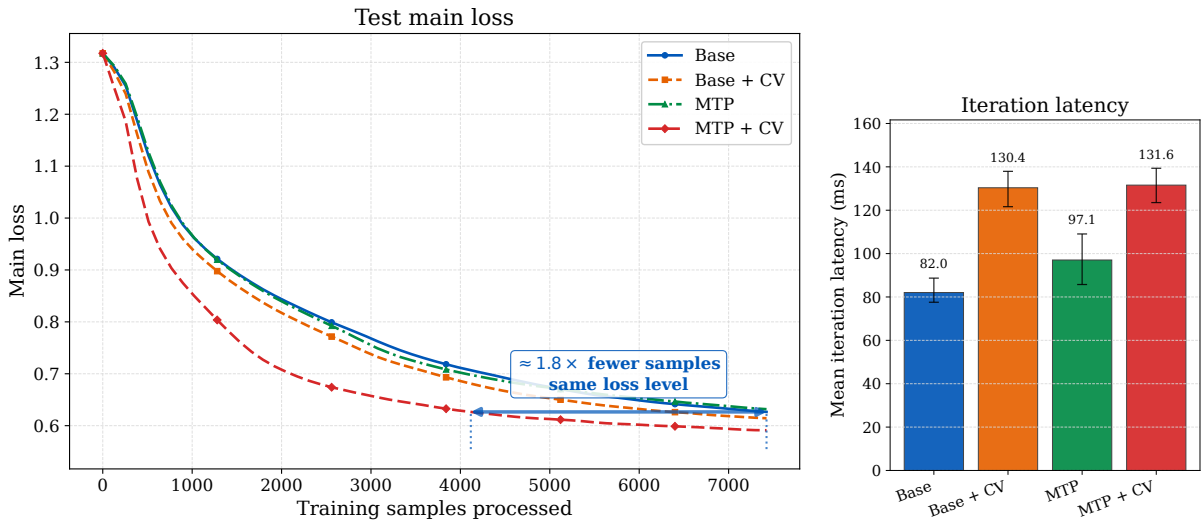
G.1 SpecSGD Qwen3.5-0.8B



(a) Main loss value for the test subset.

(b) Latency results for each regime.

Figure 5: Test results for Qwen3.5-0.8B **full fine-tuning** using SpecSGD algorithm on GSM8k.

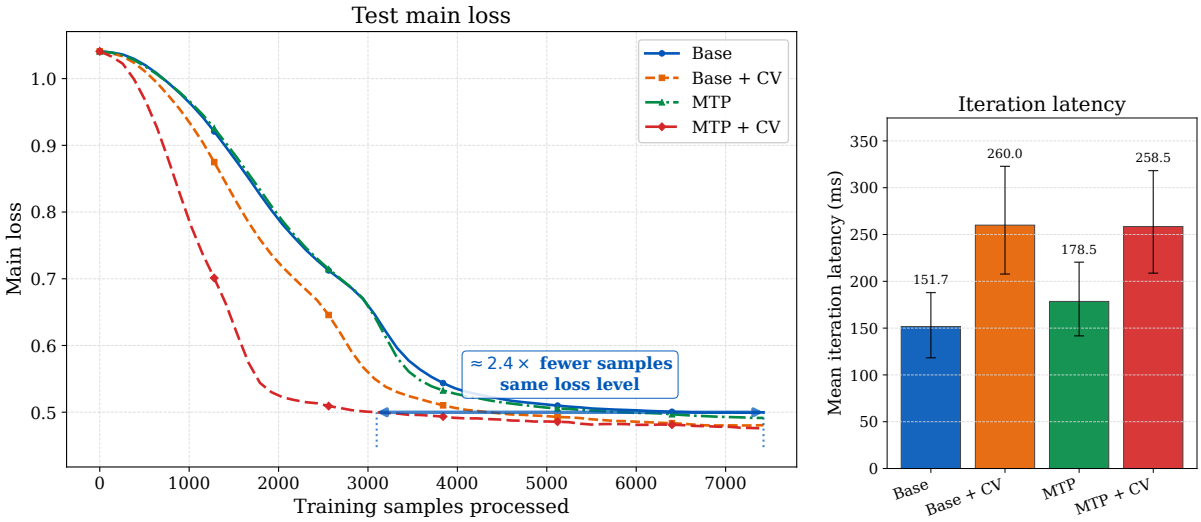


(a) Main loss value for the test subset.

(b) Latency results for each regime.

Figure 6: Test results for Qwen3.5-0.8B **LoRA fine-tuning** using SpecSGD algorithm on GSM8k.

G.2 SpecSGD Qwen3.5-4B

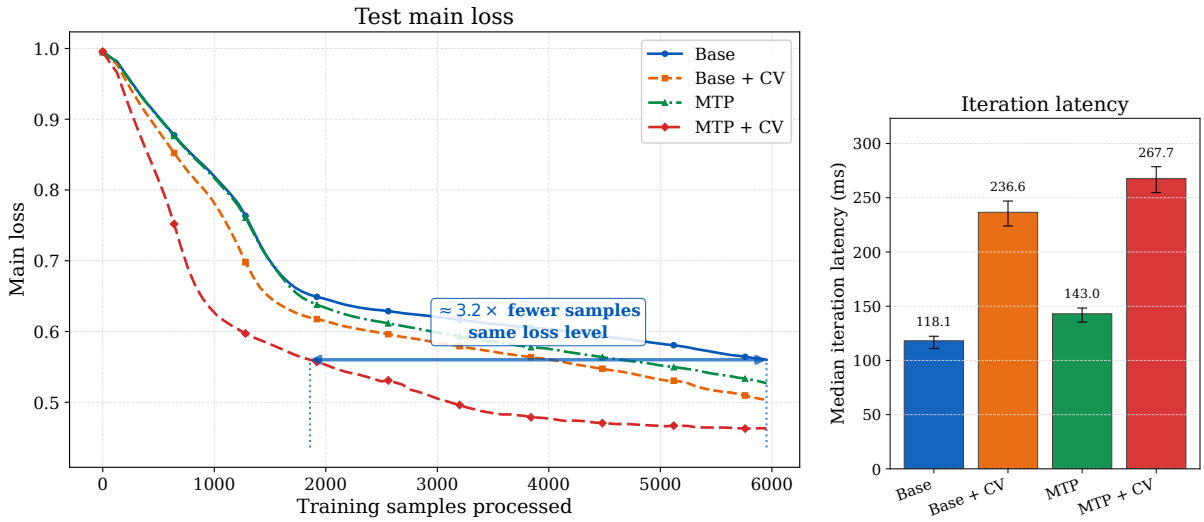


(a) Main loss value for the test subset.

(b) Latency results for each regime.

Figure 7: Test results for Qwen3.5-4B **LoRA fine-tuning** using SpecSGD algorithm on GSM8k.

G.3 SpecSGD Qwen3.5-9B



(a) Main loss value for the test subset.

(b) Latency results for each regime.

Figure 8: Test results for Qwen3.5-9B **full fine-tuning** using SpecSGD algorithm on GSM8k.

G.4 SpecAdam Qwen3.5-0.8B

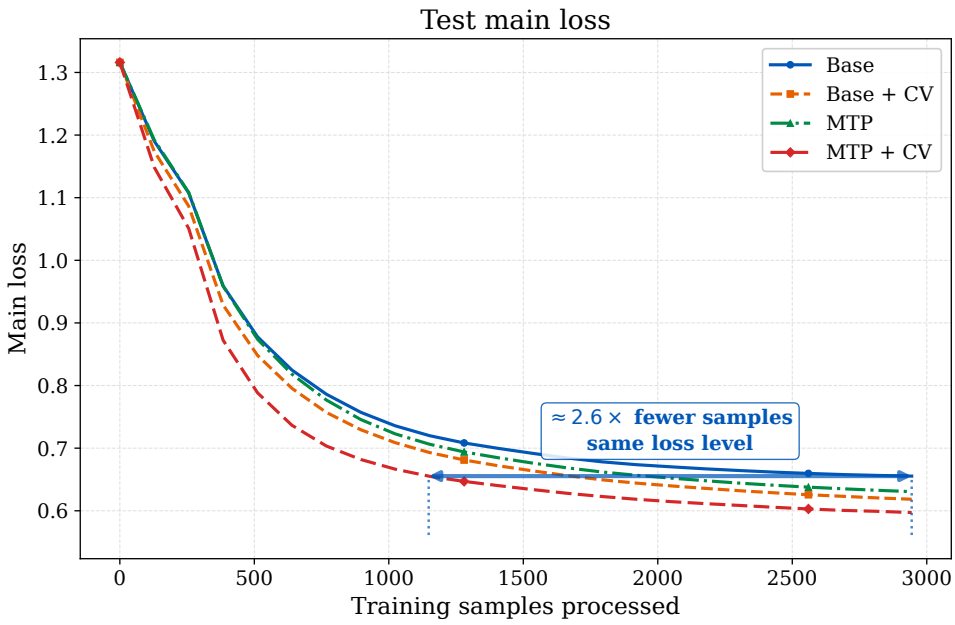


Figure 9: Test results for Qwen3.5-0.8B **full fine-tuning** using SpecAdam algorithm on GSM8k.

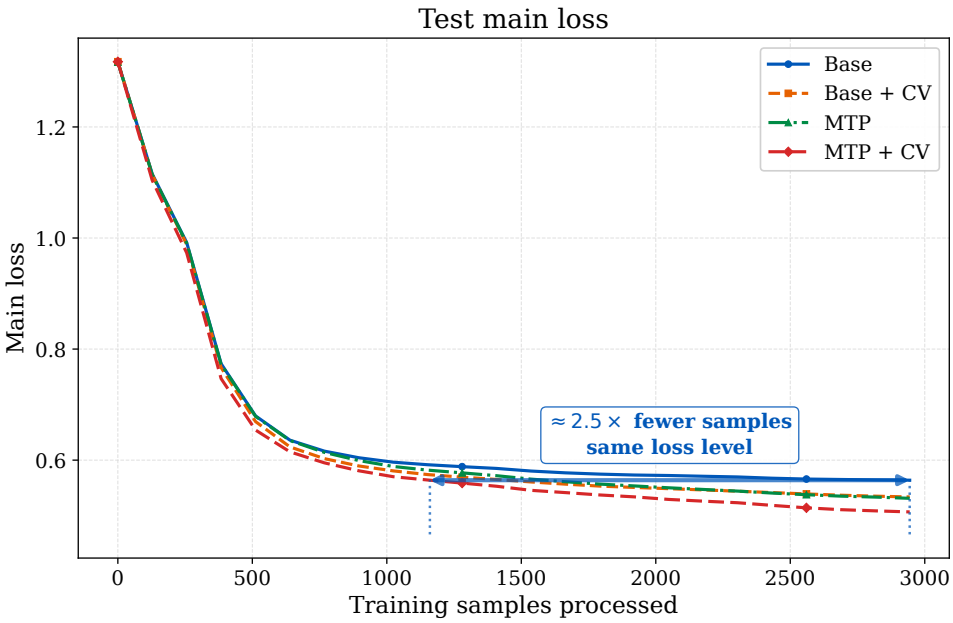


Figure 10: Test results for Qwen3.5-0.8B **LoRA fine-tuning** using SPECADAM algorithm on GSM8k.