ENABLING DOWNSTREAM ADAPTATION OF SPIKING LLMS: FROM ZEROTH-ORDER LEARNING TO CROSS-RESOLUTION TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Spiking large language models (LLMs) communicate through spike trains and support event-driven computation, offering the potential for energy-efficient language processing on neuromorphic hardware. However, adapting these models to downstream tasks remains challenging. Backpropagation through time incurs memory costs that grow with time steps and compound the substantial memory demands of LLMs, making forward-only ZO-SGD an attractive alternative. To examine its effectiveness, we derive a convergence bound for spiking LLMs that accounts for conversion error, suggesting that its accumulation may limit fine-tuning gains under limited-resolution neuron settings. Our empirical results are consistent with this analysis, showing that ZO-SGD fine-tuning remains effective at higher neuron resolutions but yields substantially smaller gains under limited-resolution settings. Motivated by these findings, we propose a two-stage framework that combines task learning with knowledge transfer across neuron resolutions to obtain task-adapted spiking LLMs with limited-resolution neurons. The task learning stage applies ZO-SGD at a high-resolution neuron setting to acquire task-specific knowledge. The knowledge transfer stage uses the adapted high-resolution model as a frozen teacher for progressive block-wise representation alignment, matching the limited-resolution model's intermediate representations to the corresponding teacher outputs. This teacher-guided alignment is designed to compensate for representation mismatch across neuron resolutions while preserving the task knowledge acquired during high-resolution fine-tuning. Experimental results show that the proposed framework improves downstream task performance over direct ZO-SGD fine-tuning under limited-resolution neuron settings.

1 INTRODUCTION

Spiking neural networks (SNNs) encode and transmit information through temporally sparse spike trains, supporting event-driven and energy-efficient computation (Roy et al., 2019; Woźniak et al., 2020; Schuman et al., 2022; Davies et al., 2018). Meanwhile, large language models (LLMs) acquire broad capabilities in language understanding and generation through extensive pretraining, but entail substantial computational and energy costs (Brown et al., 2020; Touvron et al., 2023; Achiam et al., 2023; Yang et al., 2025). Recent work has therefore explored spiking LLMs through ANN-to-SNN conversion, which reuses pretrained weights while encoding intermediate activations with spiking neurons. This approach extends spiking computation to models with billions of parameters without repeating pretraining (Xing et al., 2025; Chen et al., 2025a; 2026; Zheng et al., 2026). Spiking LLMs thus retain pretrained language capabilities while offering the potential to reduce inference energy through spiking computation.

However, realizing this potential across downstream applications requires adapting spiking LLMs to task-specific data and objectives. For conventional LLMs, fine-tuning is the standard approach for adapting pretrained models to downstream tasks (Howard & Ruder, 2018; Hu et al., 2021; Dettmers et al., 2023). For spiking LLMs, this requires effective learning under the temporal and discrete computation introduced by spiking neurons. Backpropagation through time (BPTT) with surrogate gradients is widely used to train SNNs, as surrogate functions provide approximate gradients for the non-differentiable spike generation function (Wu et al., 2018; Neftci et al., 2019; Wu et al.,

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

2019). However, BPTT unfolds the computational graph across time and retains intermediate states for gradient computation (Xiao et al., 2026; Qin et al., 2026). When applied to spiking LLMs, this memory cost grows with time steps and compounds the substantial memory requirements of LLMs, limiting the scalability of BPTT-based fine-tuning.

Forward-only zeroth-order stochastic gradient descent (ZO-SGD) offers an attractive alternative by constructing update directions from finite differences of task loss values evaluated at randomly perturbed model parameters, without requiring backpropagation (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017). Building on this forward-only formulation, MeZO adapts ZO-SGD to LLM fine-tuning by applying parameter perturbations in place and regenerating them from random seeds, reducing the memory footprint to a level comparable to inference (Malladi et al., 2023). This memory-efficient formulation makes MeZO a natural candidate for spiking LLM adaptation. Nevertheless, spiking neurons introduce distinct challenges for ZO gradient estimation (Qin et al., 2026), leaving it unclear whether the effectiveness of ZO fine-tuning on conventional LLMs extends to ANN-to-SNN converted spiking LLMs.

We therefore analyze ZO-SGD fine-tuning through the relationship between the original spiking objective and its Gaussian smoothing. Under the stated assumptions, we derive a convergence bound that explicitly accounts for conversion error accumulated across transformer blocks. This error contributes to the gap between the two objectives and enlarges the bound on the final optimization error when other quantities are held fixed. Neuron resolution is controlled by settings such as the number of spiking time steps and the available spike output levels. Reducing this resolution can lower the cost of spike coding, but our analysis suggests that the associated conversion error may restrict fine-tuning gains. Empirical results across the SpikeLLM (Xing et al., 2025) and SpikingLLM (Zheng et al., 2026) conversion frameworks are consistent with this analysis, showing that ZO-SGD remains effective at high neuron resolutions but yields substantially smaller fine-tuning gains under limited-resolution settings.

Building on these theoretical and empirical findings, we propose a two-stage framework that combines task learning with knowledge transfer across neuron resolutions. The task learning stage applies ZO-SGD at high neuron resolution to acquire task-specific knowledge. The resulting model serves as a teacher whose outputs reflect downstream task adaptation. The knowledge transfer stage freezes this teacher and performs progressive block-wise representation alignment, matching the intermediate representations of a student with limited-resolution neurons to the corresponding teacher outputs. Each block is aligned using the output of the previously aligned student prefix, allowing the alignment to account for mismatch inherited from preceding blocks. This procedure is designed to preserve task knowledge while compensating for the representation changes introduced by lower neuron resolution. These stages provide a route to task-adapted spiking LLMs with limited-resolution neurons by transferring knowledge acquired through high-resolution ZO fine-tuning.

Our main contributions are as follows:

- We derive a convergence bound for ZO-SGD on spiking LLMs that accounts for conversion error and empirically examine the sensitivity of fine-tuning to neuron resolution. Results across two conversion frameworks are consistent with the analysis, showing smaller fine-tuning gains under limited-resolution settings.
- We propose a two-stage framework that combines high-resolution task learning with block-wise knowledge transfer across neuron resolutions to obtain task-adapted spiking LLMs with limited-resolution neurons.
- We conduct extensive experiments across downstream tasks and conversion frameworks, demonstrating that the proposed framework improves downstream task performance over direct ZO-SGD fine-tuning under limited-resolution neuron settings.

2 PRELIMINARIES

2.1 ZEROTH-ORDER PARAMETER UPDATES

ZO-SGD (Ghadimi & Lan, 2013; Nesterov & Spokoiny, 2017) constructs parameter updates from loss evaluations without backpropagating through the model. Let $\theta \in \mathbb{R}^d$ denote the trainable parameters of a language model and let $\mathcal{L}(\theta; B)$ denote its task loss on a mini-batch B. At iteration

t, a two-point estimator samples n independent Gaussian directions $z_{t,1}, \ldots, z_{t,n} \sim \mathcal{N}(0, I_d)$ and estimates an update direction using finite differences of the loss:

$$G_t := \frac{1}{n} \sum_{j=1}^{n} \frac{\mathcal{L}(\theta_t + \varepsilon z_{t,j}; B_t) - \mathcal{L}(\theta_t - \varepsilon z_{t,j}; B_t)}{2\varepsilon} z_{t,j}, \tag{1}$$

where $\varepsilon > 0$ is the perturbation scale. Given a learning rate η, ZO-SGD updates the parameters as

$$\theta_{t+1} = \theta_t - \eta G_t, \tag{2}$$

MeZO (Malladi et al., 2023) implements this update by perturbing parameters in place and regenerating random directions from their seeds. This avoids storing a full perturbation vector or retaining activations for backpropagation, bringing fine-tuning memory usage close to that of inference.

2.2 Spike Coding and Neuron Resolution

In ANN-to-SNN converted LLMs, spiking neurons approximate intermediate activations through finite sequences of discrete outputs. We use *neuron resolution* to describe the representation capacity of this activation coding, as determined by the number, spacing, and range of attainable reconstructed values. Let $\tau \in \{1, \ldots, T_{\text{sp}}\}$ index spiking time steps. For a scalar activation a, its reconstruction is $\widehat{a} = \sum_{\tau=1}^{T_{\text{sp}}} O_\tau$, where O_τ is the output contribution at step τ. The membrane potential V_τ and spike state q_τ determine these contributions according to the neuron dynamics. Below, we consider a nonnegative input with $V_1 = a$, omitting scaling and sign handling specific to each framework, and characterize resolution through two coding schemes used in spiking LLMs.

Multi-level spike coding. The Generalized Integrate-and-Fire (GIF) neurons used in SpikeLLM (Xing et al., 2025) allow multiple spike levels within each time step. For a threshold $\vartheta > 0$ and an integer $N \geq 1$ controlling spike levels, the encoding and subtractive reset can be written as

$$q_\tau = \text{clip}\left(\left\lfloor \frac{NV_\tau}{\vartheta} \right\rfloor, 0, N \right), \qquad O_\tau = \frac{\vartheta}{N} q_\tau, \qquad V_{\tau+1} = V_\tau - O_\tau, \tag{3}$$

where $\text{clip}(u, 0, N) = \min\{N, \max\{0, u\}\}$ and $q_\tau \in \{0, \ldots, N\}$. Under this scalar formulation, $\widehat{a}$ lies on the grid $\{m\vartheta/N : m = 0, \ldots, NT_{\text{sp}}\}$. The two parameters control complementary properties of this grid. The spacing between adjacent values is ϑ/N, so a larger N allows the neuron to distinguish more closely spaced activation values. Each time step contributes at most ϑ, giving a maximum representable value of $T_{\text{sp}}\vartheta$. Accordingly, N controls the value spacing and T_{sp} controls the representable range, with the pair (N, T_{sp}) jointly specifying the neuron resolution.

Phase weight spike coding. SpikingLLM (Zheng et al., 2026) represents activations using binary spike states with output weights that vary across time steps. Writing the firing threshold, reset strength, and output weight as ϑ_τ, r_τ, and w_τ, respectively, its neuron dynamics take the form

$$q_\tau = \mathbb{I}\{V_\tau \geq \vartheta_\tau\}, \qquad O_\tau = w_\tau q_\tau, \qquad V_{\tau+1} = V_\tau - r_\tau q_\tau, \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Each neuron parameter is parameterized as $\gamma^{-\tau}$ with its own learnable base γ, initialized to 2 by default. For fixed output weights, each reconstructed activation is determined by a binary code $\mathbf{b} = (b_1, \ldots, b_{T_{\text{sp}}})$ as

$$\widehat{a}(\mathbf{b}) = \sum_{\tau=1}^{T_{\text{sp}}} w_\tau b_\tau, \qquad \mathbf{b} \in \{0,1\}^{T_{\text{sp}}}, \qquad \left|\{0,1\}^{T_{\text{sp}}}\right| = 2^{T_{\text{sp}}}. \tag{5}$$

Each of the T_{sp} time steps contributes a binary state, giving $2^{T_{\text{sp}}}$ possible spike codes. Increasing T_{sp} therefore provides more codes for representing activations, whereas decreasing T_{sp} yields a coarser representation. Accordingly, the neuron resolution is specified by T_{sp}.

3 ZO Fine-Tuning in Spiking LLMs across Neuron Resolutions: Theory and Empirical Evidence

Neuron resolution determines how finely a converted spiking LLM can represent intermediate activations, raising the question of how the resulting conversion error affects ZO-SGD fine-tuning.

We study this question by relating conversion error accumulated across transformer blocks to the discrepancy between the original spiking objective and its Gaussian smoothing. This relationship yields a convergence guarantee that accounts for conversion error, and we examine its implications empirically at different neuron resolutions.

3.1 Problem Setup

We consider ZO-SGD fine-tuning with a fixed neuron-resolution setting and trainable full-precision model weights. The reference ANN and its spiking counterpart share the same weights, so their forward difference measures the effect of activation conversion at each parameter iteration.

ANN and spiking forward passes. For a training example $\xi = (x, y) \sim \mathcal{D}$, consider a model with L transformer blocks and weights $\theta = (\theta_1, \dots, \theta_L) \in \mathbb{R}^d$. We write $f_\ell(\cdot; \theta_\ell)$ for the mapping implemented by the ℓ-th ANN block and $h_\ell(\theta; \xi)$ for its output representation. The reference ANN computes these representations sequentially as

$$h_0(\theta; \xi) = x, \quad h_\ell(\theta; \xi) = f_\ell\big(h_{\ell-1}(\theta; \xi); \theta_\ell\big), \quad \ell = 1, \dots, L. \tag{6}$$

At the same parameter vector θ, the converted model unfolds each block over T_{sp} spiking time steps. We denote the output of block ℓ at time step τ by $s_{\ell,\tau}(\theta; \xi)$ and the corresponding temporal block map by $f^s_{\ell,\tau}(\cdot; \theta_\ell)$. The temporal outputs and their aggregation are

$$s_{\ell,\tau}(\theta; \xi) = f^s_{\ell,\tau}\big(s_{\ell-1}(\theta; \xi); \theta_\ell\big), \quad s_\ell(\theta; \xi) = \sum_{\tau=1}^{T_{\mathrm{sp}}} s_{\ell,\tau}(\theta; \xi). \tag{7}$$

For compactness, we absorb the temporal dynamics and aggregation into $f^s_\ell(\cdot; \theta_\ell)$, giving

$$s_0(\theta; \xi) = x, \quad s_\ell(\theta; \xi) = f^s_\ell\big(s_{\ell-1}(\theta; \xi); \theta_\ell\big), \quad \ell = 1, \dots, L. \tag{8}$$

The difference between the final representations affects the downstream task loss. For a shared loss function ℓ_{task}, define the sample losses and their expectations as

$$\begin{aligned} \ell_a(\theta; \xi) &:= \ell_{\mathrm{task}}\big(h_L(\theta; \xi), y\big), \quad F_a(\theta) := \mathbb{E}_\xi[\ell_a(\theta; \xi)], \\ \ell_s(\theta; \xi) &:= \ell_{\mathrm{task}}\big(s_L(\theta; \xi), y\big), \quad F_s(\theta) := \mathbb{E}_\xi[\ell_s(\theta; \xi)]. \end{aligned} \tag{9}$$

For a mini-batch B, ZO-SGD uses the empirical spiking loss $\mathcal{L}_s(\theta; B) := |B|^{-1} \sum_{\xi_i \in B} \ell_s(\theta; \xi_i)$.

Gaussian smoothing and ZO-SGD. The discrete responses of spiking neurons mean that F_s need not be differentiable. Gaussian smoothing provides a differentiable objective through which we can characterize the expected ZO-SGD update. For $z \sim \mathcal{N}(0, I_d)$ and perturbation scale $\varepsilon > 0$, define

$$F_{s,\varepsilon}(\theta) := \mathbb{E}_z[F_s(\theta + \varepsilon z)], \quad F_{a,\varepsilon}(\theta) := \mathbb{E}_z[F_a(\theta + \varepsilon z)]. \tag{10}$$

We apply the ZO-SGD estimator and update in Equations 1 and 2 with $\mathcal{L} = \mathcal{L}_s$. While the expected ZO-SGD direction is characterized through $F_{s,\varepsilon}$, our goal is to bound the optimization error on the original spiking objective F_s. Relating these two objectives requires accounting for ANN-to-SNN conversion error.

Block-wise conversion error. To relate conversion to the task objective, we track how representation mismatch arises and propagates across blocks. Suppressing $(\theta; \xi)$ for compactness, we decompose the mismatch at block ℓ as

$$\Delta_\ell := s_\ell - h_\ell = \underbrace{f_\ell(s_{\ell-1}; \theta_\ell) - f_\ell(h_{\ell-1}; \theta_\ell)}_{u_\ell} + \underbrace{f^s_\ell(s_{\ell-1}; \theta_\ell) - f_\ell(s_{\ell-1}; \theta_\ell)}_{v_\ell}. \tag{11}$$

Here, u_ℓ measures the change in the reference block's output caused by mismatch in its input. The term v_ℓ measures the additional difference introduced by the spiking block when both block maps receive the same input $s_{\ell-1}$. We define the squared error components and their interaction term as

$$R_\ell := \|\Delta_\ell\|^2, \quad A_\ell := \|u_\ell\|^2, \quad B_\ell := \|v_\ell\|^2, \quad C_\ell := 2\langle u_\ell, v_\ell \rangle. \tag{12}$$

Thus, $R_\ell = A_\ell + B_\ell + C_\ell$ separates the total squared mismatch into propagated error, newly introduced error, and their interaction. Let L_ℓ denote the input Lipschitz constant in Assumption B.2, and define the corresponding squared propagation factor as $\Gamma_\ell := L_\ell^2$.

Lemma 3.1 (Accumulation of block-wise conversion error). *Fix $\theta \in \mathbb{R}^d$ and ξ. Under Assumption B.2, $R_1 = B_1$ and, for every $\ell = 2, \dots, L$,*

$$A_\ell \le \Gamma_\ell R_{\ell-1}, \qquad R_\ell \le \Gamma_\ell R_{\ell-1} + B_\ell + |C_\ell|.$$

Consequently, the final representation mismatch satisfies

$$R_L(\theta;\xi) = \|s_L(\theta;\xi) - h_L(\theta;\xi)\|^2 \le \underbrace{\sum_{\ell=1}^{L} \big(B_\ell(\theta;\xi) + |C_\ell(\theta;\xi)|\big) \prod_{j=\ell+1}^{L} \Gamma_j}_{=:\, \mathcal{E}_{\mathrm{conv}}(\theta;\xi)}, \tag{13}$$

where an empty product equals one.

The proof is provided in Appendix B. Equation 13 bounds the squared final representation mismatch by newly introduced conversion errors and their interactions with propagated errors, weighted by propagation factors. For the ZO-SGD analysis, we define the expected conversion-error bounds

$$\mathcal{C}(\theta) := \mathbb{E}_\xi \left[\mathcal{E}_{\mathrm{conv}}(\theta;\xi)\right], \quad \mathcal{C}_\varepsilon(\theta) := \mathbb{E}_{z,\xi} \left[\mathcal{E}_{\mathrm{conv}}(\theta + \varepsilon z;\xi)\right]. \tag{14}$$

These quantities bound the expected squared final representation mismatch at θ and under Gaussian parameter perturbations, respectively. They will be used to bound the gap between F_s and $F_{s,\varepsilon}$.

3.2 Theoretical Analysis

The full technical assumptions used in the following analysis are provided in Appendix B.1. We analyze ZO-SGD through the Gaussian-smoothed objective $F_{s,\varepsilon}$, for which the estimator satisfies $\mathbb{E}[G_t \mid \theta_t] = \nabla F_{s,\varepsilon}(\theta_t)$ under Assumption B.1, even when F_s is nondifferentiable (Nesterov & Spokoiny, 2017). The formal statement and proof of the Gaussian smoothing identity are provided in Appendix B.3. To obtain a convergence guarantee for the original objective F_s, we then bound its discrepancy from $F_{s,\varepsilon}$ using the accumulated conversion error.

For this purpose, we use the reference ANN objective F_a to isolate the effect of ANN-to-SNN conversion. Let $D_s(\theta) := F_s(\theta) - F_a(\theta)$ denote the resulting objective gap, and define its expected absolute change under Gaussian smoothing as

$$\Omega_\varepsilon(\theta) := \mathbb{E}_z \left[|D_s(\theta + \varepsilon z) - D_s(\theta)|\right]. \tag{15}$$

Lemma 3.2 (Smoothing gap induced by conversion error). *Fix $\theta \in \mathbb{R}^d$ and $\varepsilon > 0$. Under Assumptions B.2–B.4, the original spiking objective and its Gaussian smoothing satisfy*

$$|F_{s,\varepsilon}(\theta) - F_s(\theta)| \le \frac{L_a \varepsilon^2 d}{2} + \Omega_\varepsilon(\theta) \le \frac{L_a \varepsilon^2 d}{2} + \rho \left(\sqrt{\mathcal{C}(\theta)} + \sqrt{\mathcal{C}_\varepsilon(\theta)}\right). \tag{16}$$

The term $L_a \varepsilon^2 d/2$ bounds the smoothing error of the ANN reference, while $\Omega_\varepsilon(\theta)$ measures the expected absolute change in the conversion-induced objective gap under Gaussian smoothing. The second inequality bounds this change using $\mathcal{C}(\theta)$ and $\mathcal{C}_\varepsilon(\theta)$. These quantities account for conversion error when deriving a guarantee for F_s from the analysis of $F_{s,\varepsilon}$. Let $F^\star_{s,\varepsilon} := \inf_\theta F_{s,\varepsilon}(\theta)$ and $F^\star_s := F_s(\theta^\star_s) = \inf_\theta F_s(\theta)$, where $\theta^\star_s$ is a minimizer of F_s.

Theorem 3.1 (Non-asymptotic convergence for the original spiking objective). *Let $\{\theta_t\}_{t=0}^K$ be the sequence generated by the ZO-SGD update in Equation 2. Suppose Assumptions B.1–B.6 hold. For a constant step size $0 < \eta \le 1/L_\varepsilon$ and every integer $K \ge 0$, the final iterate satisfies*

$$\begin{aligned} \mathbb{E}[F_s(\theta_K) - F^\star_s] \le{}& (1 - \eta\mu_{\mathrm{PL}})^K \big(F_{s,\varepsilon}(\theta_0) - F^\star_{s,\varepsilon}\big) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}} + L_a \varepsilon^2 d \\ &+ \rho\, \mathbb{E}\left[\sqrt{\mathcal{C}(\theta_K)} + \sqrt{\mathcal{C}_\varepsilon(\theta_K)}\right] + \rho \left(\sqrt{\mathcal{C}(\theta^\star_s)} + \sqrt{\mathcal{C}_\varepsilon(\theta^\star_s)}\right), \end{aligned} \tag{17}$$

where the expectation is over the mini-batches and Gaussian directions sampled up to iteration K.

Theorem 3.1 links accumulated ANN-to-SNN conversion error to the convergence guarantee for the original spiking objective F_s. This dependence arises from bounding the discrepancy between the original and Gaussian-smoothed objectives. Reducing the optimization error on $F_{s,\varepsilon}$ does not by itself eliminate the terms involving $\mathcal{C}$ and $\mathcal{C}_\varepsilon$, which need not vanish as the number of updates increases. The following corollary simplifies these terms under a uniform bound on $\mathcal{C}(\theta)$ and $\mathcal{C}_\varepsilon(\theta)$.

Corollary 3.1 (Convergence under uniform control of conversion error)**.** *Under the conditions of Theorem 3.1, suppose further that there exist constants $U \geq 0$ and $\Psi \geq 0$ such that*

$$\mathcal{C}(\theta_K), \mathcal{C}_\varepsilon(\theta_K) \leq U^2\Psi \quad \textit{almost surely}, \qquad \mathcal{C}(\theta_s^\star), \mathcal{C}_\varepsilon(\theta_s^\star) \leq U^2\Psi. \tag{18}$$

Here, U and Ψ jointly specify the uniform conversion-error bound $U^2\Psi$. Then, for every $K \geq 0$, the original spiking objective satisfies

$$\mathbb{E}[F_s(\theta_K) - F_s^\star] \leq (1 - \eta\mu_{\mathrm{PL}})^K \left(F_{s,\varepsilon}(\theta_0) - F_{s,\varepsilon}^\star\right) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}} + 4\rho U\sqrt{\Psi} + L_a \varepsilon^2 d. \tag{19}$$

Equation 19 contains a geometrically decaying term and three terms due to estimator variance, ANN-to-SNN conversion, and Gaussian smoothing. The conversion contribution is $4\rho U\sqrt{\Psi}$, which increases with the uniform conversion-error bound. Complete proofs are deferred to Appendix B.

3.3 Empirical Results

The preceding analysis identifies accumulated conversion error as a term in the convergence bound. Since reducing neuron resolution generally increases conversion error, we vary neuron resolution and compare the gains from direct ZO-SGD fine-tuning to examine this theoretical prediction.

Tasks and evaluation. We evaluate SST-2 and RTE from GLUE (Wang et al., 2018), BoolQ, WSC, WIC, MultiRC, and COPA from SuperGLUE (Wang et al., 2019), and SQuAD (Rajpurkar et al., 2016). These tasks cover sentiment classification, natural language inference, word sense disambiguation, commonsense reasoning, and reading comprehension. For each task, we use 1,000 labeled training examples and 1,000 test examples. We report F1 for SQuAD and accuracy for all other tasks. The average score is computed over the tasks reported in each table.

Models and neuron configurations. We evaluate Llama2-7B (Touvron et al., 2023) and OPT-6.7B (Zhang et al., 2022) under SpikeLLM (Xing et al., 2025), and Llama2-7B under SpikingLLM (Zheng et al., 2026). For SpikeLLM, both models use the high-resolution configuration $(N, T_{\mathrm{sp}}) = (256, 8)$, while the limited-resolution configurations are $(16, 4)$ for Llama2-7B and $(64, 4)$ for OPT-6.7B. For SpikingLLM, the corresponding configurations use $T_{\mathrm{sp}} = 8$ and $T_{\mathrm{sp}} = 6$. These comparisons examine the effect of neuron resolution within each model and framework.

Compared methods. Zero-Shot and MeZO report the performance of the reference ANN before and after ZO-SGD fine-tuning, respectively. For the spiking models, Zero-Shot-S evaluates the converted pretrained model without downstream fine-tuning. MeZO-S first fine-tunes the ANN and then converts it to a spiking model, whereas S-MeZO first converts the pretrained ANN and then directly fine-tunes the resulting spiking model. S-MeZO therefore corresponds to optimization of the spiking objective F_s analyzed above. We measure the benefit of direct fine-tuning by comparing S-MeZO with Zero-Shot-S at the same neuron resolution.

Effect of neuron resolution on ZO-SGD fine-tuning. Table 1 reports the average score under each setting, while Tables 6–8 in Appendix C.2 provide the corresponding results for individual tasks. Direct ZO-SGD fine-tuning remains effective at high neuron resolution and achieves performance comparable to ANN MeZO under SpikeLLM. When neuron resolution is reduced, its improvement over Zero-Shot-S becomes smaller for Llama2-7B and OPT-6.7B under SpikeLLM, as well as for Llama2-7B under SpikingLLM. These observations are consistent with Equation 19, in which accumulated conversion error enlarges the bound on the final optimization error. MeZO-S further shows that performance loss can occur after task learning has been completed in the ANN, suggesting that the benefits of task learning may not be fully preserved after conversion.

4 Two-Stage Framework: Task Learning and Knowledge Transfer across Neuron Resolutions

4.1 Framework Overview

The analysis and empirical results in Section 3 show that accumulated conversion error enlarges the convergence bound and that direct ZO-SGD gains decrease when neuron resolution is limited.

Model	Conversion	Resolution	ANN reference		Spiking LLM		
			Zero-Shot	MeZO	Zero-Shot-S	MeZO-S	S-MeZO
Llama2-7B	SpikeLLM	High	56.97	70.08	56.34	69.72	70.71
		Limited			52.99	58.21	58.53
OPT-6.7B	SpikeLLM	High	53.35	69.80	53.86	67.46	69.98
		Limited			52.57	67.72	64.95
Llama2-7B	SpikingLLM	High	56.47	68.47	56.02	67.87	67.58
		Limited			55.63	65.60	61.47

Table 1: Average performance across neuron resolution settings. SpikeLLM averages cover eight tasks, whereas SpikingLLM averages cover the seven tasks excluding SQuAD.

ZO-SGD nevertheless remains effective at high neuron resolution, suggesting that adaptation in this setting can serve as a basis for addressing the difficulty encountered at limited resolution. Based on these observations, our framework adopts a two-stage design. The task learning stage adapts a high-resolution spiking LLM to the downstream task using ZO-SGD, whereas the knowledge transfer stage uses the adapted model as a frozen teacher to progressively align the intermediate representations of a limited-resolution student. The teacher and student share the same transformer backbone and ANN-to-SNN conversion framework but differ in neuron resolution. We use superscripts H and L for the forward passes at high and limited neuron resolutions, with $f_\ell^{s,\mathrm{H}}$ and $f_\ell^{s,\mathrm{L}}$ denoting the corresponding spiking block maps.

4.2 Task Learning through ZO-SGD Fine-Tuning at High Neuron Resolution

We optimize the converted pretrained model while keeping its high-resolution neuron configuration fixed. Using the task distribution $\mathcal{D}$ and loss ℓ_{task} from Section 3.1, the objective is

$$F_s^{\mathrm{H}}(\theta) := \mathbb{E}_{\xi=(x,y)\sim\mathcal{D}}\left[\ell_{\mathrm{task}}\big(s_L^{\mathrm{H}}(\theta;x),y\big)\right], \tag{20}$$

where $s_L^{\mathrm{H}}(\theta;x)$ is the final representation produced by the high-resolution spiking model. We optimize this objective using ZO-SGD, with each perturbed mini-batch loss evaluated through the high-resolution spiking forward pass. After K updates, we denote the adapted weights by $\theta^{\mathrm{H}} := \theta_K$ and freeze the model as the teacher for the next stage. This teacher choice is motivated by the effectiveness of ZO-SGD under high-resolution neuron settings. Because ZO-SGD relies only on forward evaluations, this stage avoids the activation storage across time steps required by BPTT and remains memory-efficient at the scale of LLMs. ZO-SGD fine-tuning at high neuron resolution produces intermediate representations adapted to the downstream task. The resulting model supplies these representations at every transformer block as direct supervision for the subsequent knowledge transfer stage.

4.3 Knowledge Transfer through Block-Wise Representation Alignment

We initialize the student's model weights from θ^{H} and set its neurons to the target limited-resolution configuration. Although the weights initially coincide, the change in neuron resolution can alter the intermediate representations and impair the task performance acquired during the first stage. We correct this mismatch by aligning the student blocks from shallow to deep against the frozen teacher. For an alignment set $\mathcal{D}_{\mathrm{cal}}$, the teacher produces the block-wise targets

$$s_0^{\mathrm{H}}(x) = x, \quad s_\ell^{\mathrm{H}}(x) = f_\ell^{s,\mathrm{H}}\big(s_{\ell-1}^{\mathrm{H}}(x);\theta_\ell^{\mathrm{H}}\big), \quad \ell = 1,\dots,L. \tag{21}$$

The fixed teacher parameters are omitted from the state notation. When aligning block ℓ, $s_{\ell-1}^{\mathrm{L,cal}}(x)$ denotes the output of the already aligned student prefix, with $s_0^{\mathrm{L,cal}}(x) = x$. We update the current block by minimizing the mean squared error between its output and the teacher representation,

$$\mathcal{J}_\ell(\theta_\ell^{\mathrm{L}}) := \frac{1}{|\mathcal{D}_{\mathrm{cal}}|}\sum_{x\in\mathcal{D}_{\mathrm{cal}}} \mathrm{MSE}\Big(f_\ell^{s,\mathrm{L}}\big(s_{\ell-1}^{\mathrm{L,cal}}(x);\theta_\ell^{\mathrm{L}}\big), s_\ell^{\mathrm{H}}(x)\Big), \tag{22}$$

We backpropagate the alignment loss only through the current student block using surrogate gradients to approximate derivatives of the spike function. The teacher, the student prefix, and the neuron configuration remain fixed throughout this optimization. After alignment, we freeze the resulting weights $\theta_\ell^{\mathrm{L,cal}}$ and compute

$$s_\ell^{\mathrm{L,cal}}(x) = f_\ell^{s,\mathrm{L}}\big(s_{\ell-1}^{\mathrm{L,cal}}(x); \theta_\ell^{\mathrm{L,cal}}\big), \tag{23}$$

which becomes the input for aligning block $\ell + 1$. This propagation mirrors the student's inference process, in which each block receives representations produced by its preceding limited-resolution blocks. Proceeding through all L blocks transfers the teacher's task-adapted representations to the student at the target neuron resolution.

5 Experiments

We evaluate the framework using the models, neuron configurations, and downstream evaluation protocol in Section 3.3. The high-resolution configurations are used for task learning, and the corresponding limited-resolution configurations are used for the final students. Our experiments examine downstream performance, cross-resolution representation mismatch before and after alignment, the contribution of block-wise alignment, and sensitivity to its learning rate. Additional implementation details, including the alignment data and hyperparameter settings, are provided in Appendix C.1.

5.1 Downstream Task Performance

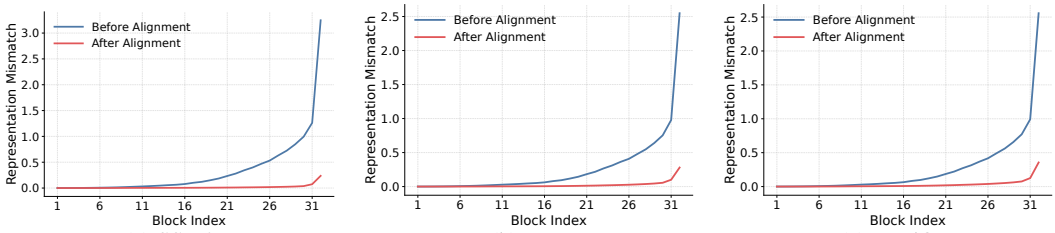
We compare our final students with Zero-Shot-S and S-MeZO at the same target neuron resolution. These comparisons assess whether high-resolution task learning followed by cross-resolution alignment improves downstream performance over direct ZO-SGD fine-tuning with limited-resolution neurons. Table 2 reports performance on individual tasks for Llama2-7B and OPT-6.7B under SpikeLLM and for Llama2-7B under SpikingLLM, with SQuAD excluded from the latter. Across all three settings, our framework achieves higher average performance than Zero-Shot-S and S-MeZO. It also achieves or ties for the best performance among the compared methods on the vast majority of reported tasks when evaluated at the same target neuron resolution. These results support the effectiveness of high-resolution task learning followed by cross-resolution representation alignment across multiple downstream tasks.

Setting	Method	SST-2	RTE	BoolQ	WSC	WIC	MultiRC	COPA	SQuAD	**Average**
Llama2-7B SpikeLLM	Zero-Shot-S	59.52	46.93	50.70	57.69	54.39	45.90	68.00	40.76	52.99
	S-MeZO	78.56	48.38	51.80	57.69	49.22	49.90	75.00	57.72	58.53
	Ours	83.72	56.68	57.40	60.58	53.61	56.60	81.00	59.68	**63.66**
OPT-6.7B SpikeLLM	Zero-Shot-S	60.21	50.90	58.60	37.50	51.88	42.90	79.00	39.60	52.57
	S-MeZO	88.88	59.21	62.90	49.04	51.88	57.30	79.00	71.42	64.95
	Ours	92.32	61.73	65.40	54.81	57.05	57.50	80.00	76.97	**68.22**
Llama2-7B SpikingLLM	Zero-Shot-S	57.11	54.87	62.90	38.46	50.00	46.10	80.00	–	55.63
	S-MeZO	67.55	56.32	64.50	54.81	52.82	51.30	83.00	–	61.47
	Ours	80.05	63.54	66.90	58.65	55.96	60.90	83.00	–	**67.00**

Table 2: Limited-resolution downstream performance across model and conversion settings.

5.2 Block-Wise Representation Mismatch

We examine whether alignment reduces cross-resolution representation mismatch using Llama2-7B under SpikingLLM on SST-2, RTE, and BoolQ. At each transformer block, we compute the cross-resolution representation MSE between the limited-resolution student's output and the corresponding representation of the frozen high-resolution teacher. Figure 1 shows that the representation MSE increases with network depth before alignment and rises sharply near the output. After alignment, it remains small through most of the network and is substantially reduced in the final blocks.



(a) SST-2 (b) RTE (c) BoolQ

Figure 1: Block-wise cross-resolution representation MSE before and after alignment for Llama2-7B under SpikingLLM on SST-2, RTE, and BoolQ.

The reduction across network depth is consistent with the intended role of progressive alignment in reducing cross-resolution representation mismatch.

5.3 Ablation and Sensitivity Analysis

Block-wise representation alignment. We omit block-wise alignment and directly evaluate the task-adapted teacher weights at the target neuron resolution. Table 3 reports the accuracy gain of the complete framework over this unaligned student, isolating the contribution of representation alignment after task learning. All measured entries show positive gains, confirming that high-resolution task learning alone does not fully preserve downstream performance after reducing neuron resolution. The improvements across multiple tasks demonstrate the contribution of block-wise representation alignment to the final student's performance.

Alignment learning rate. Using Llama2-7B under SpikingLLM, we vary the learning rate for block-wise alignment while keeping the alignment data, optimizer, batch size, and number of updates per block fixed. For each setting, we report downstream accuracy after all blocks have been aligned, as summarized in Table 4. The average accuracy over the four tasks spans 0.45 points across the tested rates. Average performance therefore varies modestly over this range, while individual tasks exhibit different sensitivities.

Framework	SST-2	RTE	BoolQ	MultiRC
SpikeLLM	+8.38	+1.81	+2.30	+1.50
SpikingLLM	+0.58	+2.53	+5.10	+2.10

Table 3: Performance gains from block-wise representation alignment on Llama2-7B across both frameworks at a limited-resolution setting.

Setting	SST-2	RTE	BoolQ	MultiRC
1×10^{-6}	79.24	63.54	66.90	58.60
2×10^{-6}	79.13	61.37	65.50	60.90
5×10^{-6}	80.05	59.93	66.80	59.70

Table 4: Effect of alignment learning rate on Llama2-7B under the SpikingLLM framework.

6 Conclusion

We investigate the downstream adaptation of ANN-to-SNN converted spiking LLMs with forward-only ZO-SGD, focusing on the effect of neuron resolution. Our convergence analysis links block-wise accumulated conversion error to the gap between the original spiking objective and its Gaussian smoothing, resulting in a convergence guarantee with terms that depend on conversion error. Results across two conversion frameworks are consistent with this analysis, showing that ZO-SGD remains effective at high resolution but its gains diminish with limited-resolution neurons. Motivated by these findings, we propose a two-stage framework that acquires task knowledge through high-resolution ZO-SGD and transfers it to a limited-resolution model through teacher-guided progressive block-wise representation alignment. Experiments across multiple downstream tasks show that the framework improves performance over direct ZO-SGD fine-tuning at limited neuron resolution under both ANN-to-SNN conversion frameworks considered. The representation analysis and ablation results further show that block-wise alignment reduces mismatch with the teacher and improves task performance beyond directly reusing the adapted weights at limited resolution.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, manuscript refinement, and assistance with mathematical proofs. We have not used generative AI tools for conducting experiments or generating experimental results, and human-subject data collection and participant interaction are not applicable to this work. Additionally, we used generative AI tools for LaTeX formatting, proofreading, and terminology consistency. We have reviewed all AI-assisted work. We checked all AI-assisted revisions against the mathematical derivations, experimental records, cited literature, and implementation, and manually verified the final claims and references. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Malyaban Bal and Abhronil Sengupta. Spikingbert: Distilling bert to train spiking language models using implicit differentiation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 10998–11006, 2024.

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

Long Chen, Xiaotian Song, Andy Song, BaDong Chen, Jiancheng Lv, and Yanan Sun. Fas: Fast ann-snn conversion for spiking large language models. *arXiv preprint arXiv:2502.04405*, 2025a.

Long Chen, Xiaotian Song, and Yanan Sun. Las: Loss-less ann-snn conversion for fully spike-driven large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 1730–1738, 2026.

Yiming Chen, Yuan Zhang, Liyuan Cao, Kun Yuan, and Zaiwen Wen. Enhancing zeroth-order finetuning for language models with low-rank structures. In *International Conference on Learning Representations*, volume 2025, pp. 62581–62607, 2025b.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee micro*, 38(1):82–99, 2018.

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

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Tanmay Gautam, Youngsuk Park, Hao Zhou, Parameswaran Raman, and Wooseok Ha. Variance-reduced zeroth-order methods for fine-tuning language models. *arXiv preprint arXiv:2404.08080*, 2024.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM journal on optimization*, 23(4):2341–2368, 2013.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, 2018.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 795–811. Springer, 2016.

Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *Advances in Neural Information Processing Systems*, 38:21139–21169, 2026.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Mingyue Qin, Shuyu Yin, Qinghai Guo, Peilin Liu, Xiaolin Huang, and Fei Wen. Zeroth-order forward-only snn training inspiring neuromorphic on-chip learning. In *Forty-third International Conference on Machine Learning*, 2026.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. *Nature*, 575(7784):607–617, 2019.

Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature computational science*, 2(1):10–19, 2022.

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

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Stanisław Woźniak, Angeliki Pantazi, Thomas Bohnstingl, and Evangelos Eleftheriou. Deep learning incorporating biologically inspired neural dynamics and in-memory computing. *Nature Machine Intelligence*, 2(6):325–336, 2020.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, Yuan Xie, and Luping Shi. Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 1311–1318, 2019.

Mingqing Xiao, Qingyan Meng, Zongpeng Zhang, Di He, Dongsheng Li, and Zhouchen Lin. Online pseudo-zeroth-order training of neuromorphic spiking neural networks. In *International Conference on Learning Representations*, volume 2026, pp. 135364–135391, 2026.

Xingrun Xing, Boyan Gao, Zheng Liu, David Clifton, Shitao Xiao, Wanpeng Zhang, Li Du, Zheng Zhang, Guoqi Li, and Jiajun Zhang. Spikellm: Scaling up spiking neural network to large language models via saliency-based spiking. In *International Conference on Learning Representations*, volume 2025, pp. 32880–32899, 2025.

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

Ziming Yu, Pan Zhou, Sike Wang, Jia Li, Mi Tian, and Hua Huang. Zeroth-order fine-tuning of llms in random subspaces. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4475–4485. IEEE, 2025.

Pengyun Yue, Xuanlin Yang, Mingqing Xiao, and Zhouchen Lin. Pseuzo: Pseudo-zeroth-order algorithm for training deep neural networks. *Advances in Neural Information Processing Systems*, 38:107625–107645, 2026.

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

Yanjun Zhao, Sizhe Dang, Haishan Ye, Guang Dai, Yi Qian, and Ivor Tsang. Second-order fine-tuning without pain for llms: A hessian informed zeroth-order optimizer. In *International Conference on Learning Representations*, volume 2025, pp. 43496–43520, 2025.

Hanyuan Zheng, Haozhen Zhang, Tianshuo Chen, Zhaogeng Liu, Yi Chang, and Bin Gu. Distribution-aware multi-granularity phase coding: Towards lower conversion error for spike-driven large language models. In *International Conference on Learning Representations*, volume 2026, pp. 80051–80076, 2026.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K Eshraghian. Spikegpt: Generative pre-trained language model with spiking neural networks. *arXiv preprint arXiv:2302.13939*, 2023.

APPENDIX

A RELATED WORKS

Zeroth-Order Fine-Tuning of LLMs. The memory requirements of backpropagation have motivated ZO approaches to downstream adaptation of pretrained LLMs. MeZO (Malladi et al., 2023) implements ZO-SGD through in-place parameter perturbations and seed-based reconstruction, demonstrating effective language model fine-tuning with memory usage comparable to inference. Subsequent work addresses the limitations of ZO fine-tuning from several perspectives. MeZO-SVRG (Gautam et al., 2024) incorporates stochastic variance reduction to improve optimization stability and convergence. LOZO (Chen et al., 2025b) exploits low-rank gradient structure through a low-rank ZO estimator, while SubZero (Yu et al., 2025) uses layer-wise low-rank perturbations to reduce variance in gradient estimation and memory overhead. HiZOO (Zhao et al., 2025) incorporates estimated diagonal Hessian information to account for heterogeneous curvature. Sparse-MeZO (Liu et al., 2026) restricts perturbations to selected parameters to mitigate the effects of estimation error. PseuZO (Yue et al., 2026) combines estimates of the model output Jacobian with derivatives of the task loss and uses exponential moving averages to reduce estimator variance.

Spiking LLMs. Research on spiking LLMs has explored both dedicated spiking architectures and ANN-to-SNN conversion of pretrained models. SpikeGPT (Zhu et al., 2023) integrates binary spiking activations into a recurrent architecture based on RWKV for autoregressive language modeling. SpikingBERT (Bal & Sengupta, 2024) combines implicit differentiation with knowledge distillation from a pretrained BERT teacher to train a spiking language model. ANN-to-SNN conversion further extends spiking computation to larger pretrained language models. SpikeLLM (Xing et al., 2025) introduces Generalized Integrate-and-Fire neurons and uses Optimal Brain Spiking to allocate spikes according to saliency, supporting models with billions of parameters while reducing the cost of spike coding. FAS (Chen et al., 2025a) combines full-parameter fine-tuning of the ANN with coarse-to-fine calibration of spiking thresholds and initial membrane potentials to reduce conversion error. LAS (Chen et al., 2026) develops specialized neurons for activation outliers and nonlinear operations, enabling fully spike-driven Transformer components. SpikingLLM (Zheng et al., 2026) uses distribution-aware multi-granularity phase coding to better match discrete neural outputs to activation distributions, with neuron parameters optimized while model weights remain fixed.

Existing ZO fine-tuning methods mainly target conventional LLMs, whereas research on spiking LLMs has focused primarily on constructing spiking architectures and reducing ANN-to-SNN conversion error. The downstream adaptation of converted spiking LLMs remains underexplored. Although forward-only optimization has been studied for general SNNs through OPZO (Xiao et al., 2026) and SZO (Qin et al., 2026), these methods do not address ANN-to-SNN converted LLMs or the sensitivity of ZO fine-tuning to neuron resolution. Meanwhile, the distillation strategy of SpikingBERT (Bal & Sengupta, 2024) and the ANN-guided calibration strategy of FAS (Chen et al., 2025a) suggest that supervision from a reference model can mitigate discrepancies introduced by spiking computation. Building on these complementary directions, our framework first acquires downstream task knowledge through ZO-SGD at a high-resolution neuron setting and then transfers it to a limited-resolution spiking LLM through progressive block-wise representation alignment.

B PROOFS FOR THE CONVERGENCE ANALYSIS

This section states the technical assumptions and provides proofs for the results in Section 3.

B.1 ASSUMPTIONS

Assumption B.1 (Sampling and unbiased loss estimates)**.** *For every $\theta \in \mathbb{R}^d$, the mini-batch loss satisfies $\mathbb{E}_B[\mathcal{L}_s(\theta; B)] = F_s(\theta)$. At each iteration t, B_t and the Gaussian directions are sampled independently of θ_t. The directions $z_{t,j} \sim \mathcal{N}(0, I_d)$, $j = 1, \ldots, n$, are mutually independent and independent of B_t.*

Assumption B.2 (Input Lipschitz continuity of ANN blocks)**.** *For each reference ANN block f_ℓ, there exists a constant $L_\ell > 0$ such that*

$$\|f_\ell(p; \theta_\ell) - f_\ell(q; \theta_\ell)\| \le L_\ell \|p - q\|. \tag{24}$$

The inequality holds uniformly over parameter vectors and block inputs considered in the analysis.

Assumption B.3 (Lipschitz continuity of the task loss)**.** *The task loss is ρ-Lipschitz with respect to the final representation for some $\rho > 0$, i.e.,*

$$|\ell_{\text{task}}(p, y) - \ell_{\text{task}}(q, y)| \le \rho \|p - q\|, \tag{25}$$

for all final representations p, q and targets y.

Assumption B.4 (Smoothness of the ANN reference objective)**.** *The reference ANN objective F_a is differentiable and L_a-smooth for some $L_a > 0$, i.e.,*

$$\|\nabla F_a(\theta) - \nabla F_a(\theta')\| \le L_a \|\theta - \theta'\|, \tag{26}$$

for all $\theta, \theta' \in \mathbb{R}^d$.

Assumption B.5 (Smoothness and the PL condition)**.** *The Gaussian-smoothed objective $F_{s,\varepsilon}$ is differentiable and L_ε-smooth for some $L_\varepsilon > 0$, with finite infimum $F^\star_{s,\varepsilon} := \inf_\theta F_{s,\varepsilon}(\theta) > -\infty$. It satisfies the Polyak–Łojasiewicz (PL) condition (Karimi et al., 2016) with parameter $\mu_{\text{PL}} > 0$:*

$$\frac{1}{2}\|\nabla F_{s,\varepsilon}(\theta)\|^2 \ge \mu_{\text{PL}}\big(F_{s,\varepsilon}(\theta) - F^\star_{s,\varepsilon}\big), \tag{27}$$

for all $\theta \in \mathbb{R}^d$. The original spiking objective F_s attains its minimum at $\theta^\star_s$, with $F^\star_s := F_s(\theta^\star_s)$.

Assumption B.6 (Bounded conditional variance)**.** *There exists a constant $\sigma^2_\varepsilon \ge 0$ such that the ZO-SGD estimator satisfies*

$$\mathbb{E}\left[\|G_t - \mathbb{E}[G_t \mid \theta_t]\|^2 \middle| \theta_t\right] \le \sigma^2_\varepsilon \tag{28}$$

for every iteration t. This bound accounts for randomness in both the mini-batch and the Gaussian directions.

B.2 Propagation of the ANN-to-SNN conversion error

Proof of Lemma 3.1. For $\ell \ge 2$, the input mismatch induces the following output difference in the reference ANN block:

$$u_\ell = f_\ell(s_{\ell-1}; \theta_\ell) - f_\ell(h_{\ell-1}; \theta_\ell). \tag{29}$$

The input Lipschitz bound in Assumption B.2 gives

$$\|u_\ell\| \le L_\ell \|s_{\ell-1} - h_{\ell-1}\| = L_\ell \|\Delta_{\ell-1}\|. \tag{30}$$

Squaring both sides gives

$$A_\ell \le L^2_\ell R_{\ell-1} = \Gamma_\ell R_{\ell-1}. \tag{31}$$

Since $\Delta_\ell = u_\ell + v_\ell$, expanding the squared norm gives

$$R_\ell = A_\ell + B_\ell + C_\ell \le \Gamma_\ell R_{\ell-1} + B_\ell + |C_\ell|. \tag{32}$$

For the first block, $s_0 = h_0 = x$ and hence $u_1 = 0$, so that $A_1 = C_1 = 0$ and $R_1 = B_1$. Repeatedly applying Equation 32 proves by induction that

$$R_L \le \sum_{j=1}^{L} \left(\prod_{k=j+1}^{L} \Gamma_k \right) \big(B_j + |C_j|\big) = \mathcal{E}_{\text{conv}}(\theta; \xi). \tag{33}$$

This is Equation 13 and completes the proof. □

The next elementary result translates conversion error in the representations into a discrepancy between the objectives.

Lemma B.1 (Representation-to-objective conversion bound)**.** *Fix $\theta \in \mathbb{R}^d$ and $\varepsilon > 0$. Under Assumption B.3, suppose the block-wise conversion bound in Lemma 3.1 holds at θ and at the perturbed parameters $\theta + \varepsilon z$. Then*

$$|F_s(\theta) - F_a(\theta)| \le \rho \sqrt{\mathcal{C}(\theta)}, \tag{34}$$

$$|F_{s,\varepsilon}(\theta) - F_{a,\varepsilon}(\theta)| \le \rho \sqrt{\mathcal{C}_\varepsilon(\theta)}. \tag{35}$$

Proof. Assumption B.3 implies, pointwise in ξ,

$$|\ell_s(\theta;\xi) - \ell_a(\theta;\xi)| \le \rho\|\Delta_L(\theta;\xi)\|. \tag{36}$$

Consequently, the triangle inequality for expectations followed by Cauchy–Schwarz gives

$$|F_s(\theta) - F_a(\theta)| \le \rho\mathbb{E}_\xi\left[\|\Delta_L(\theta;\xi)\|\right] \tag{37}$$

$$\le \rho\sqrt{\mathbb{E}_\xi\left[\|\Delta_L(\theta;\xi)\|^2\right]} \tag{38}$$

$$\le \rho\sqrt{\mathcal{C}(\theta)}. \tag{39}$$

This proves Equation 34. Applying the same argument jointly to (ξ, z) at the perturbed parameter $\theta + \varepsilon z$ proves Equation 35. □

These bounds connect the accumulated representation error to the ANN-to-SNN objective discrepancy, thereby supporting the subsequent analysis of the smoothing gap.

B.3 Gaussian smoothing and the ZO-SGD estimator

Lemma B.2 (Gaussian smoothing identity for ZO-SGD). *Let $F_s : \mathbb{R}^d \to \mathbb{R}$ be the original spiking objective and, for a fixed $\varepsilon > 0$, let $F_{s,\varepsilon}$ be its Gaussian smoothing defined in Equation 10. Assume that the Gaussian expectations below are finite and differentiation under the convolution integral is justified. Then $F_{s,\varepsilon}$ is differentiable and, for $z \sim \mathcal{N}(0, I_d)$,*

$$\nabla F_{s,\varepsilon}(\theta) = \frac{1}{\varepsilon}\mathbb{E}_z[F_s(\theta + \varepsilon z)z] = \mathbb{E}_z\left[\frac{F_s(\theta+\varepsilon z) - F_s(\theta - \varepsilon z)}{2\varepsilon}z\right]. \tag{40}$$

If Assumption B.1 also holds, the n-direction estimator in Equation 1 satisfies the following identity at every iteration t:

$$\mathbb{E}[G_t \mid \theta_t] = \nabla F_{s,\varepsilon}(\theta_t). \tag{41}$$

Proof. Let

$$\varphi_\varepsilon(v) = (2\pi\varepsilon^2)^{-d/2}\exp\left(-\frac{\|v\|^2}{2\varepsilon^2}\right) \tag{42}$$

be the density of $\mathcal{N}(0, \varepsilon^2 I_d)$. Writing Gaussian smoothing as a convolution gives

$$F_{s,\varepsilon}(\theta) = \int_{\mathbb{R}^d} F_s(u)\varphi_\varepsilon(u-\theta)\,du. \tag{43}$$

Because

$$\nabla_\theta\varphi_\varepsilon(u-\theta) = \frac{u-\theta}{\varepsilon^2}\varphi_\varepsilon(u-\theta), \tag{44}$$

differentiating under the integral and changing variables to $u = \theta + \varepsilon z$ yield

$$\nabla F_{s,\varepsilon}(\theta) = \int_{\mathbb{R}^d} F_s(u)\frac{u-\theta}{\varepsilon^2}\varphi_\varepsilon(u-\theta)\,du \tag{45}$$

$$= \frac{1}{\varepsilon}\mathbb{E}_z\left[F_s(\theta+\varepsilon z)z\right]. \tag{46}$$

The symmetry of the standard Gaussian distribution implies

$$\mathbb{E}_z\left[F_s(\theta-\varepsilon z)z\right] = -\mathbb{E}_z\left[F_s(\theta+\varepsilon z)z\right]. \tag{47}$$

Combining this equality with Equation 46 proves the two-sided identity

$$\nabla F_{s,\varepsilon}(\theta) = \mathbb{E}_z\left[\frac{F_s(\theta+\varepsilon z) - F_s(\theta-\varepsilon z)}{2\varepsilon}z\right]. \tag{48}$$

It remains to relate this identity to the mini-batch estimator. Conditional on θ_t and the sampled directions, Assumption B.1 gives

$$\mathbb{E}_{B_t}\left[G_t \mid \theta_t, \{z_{t,j}\}_{j=1}^n\right] = \frac{1}{n}\sum_{j=1}^n \frac{F_s(\theta_t + \varepsilon z_{t,j}) - F_s(\theta_t - \varepsilon z_{t,j})}{2\varepsilon}z_{t,j}. \tag{49}$$

Taking the expectation over the i.i.d. Gaussian directions and applying Equation 48 therefore yields

$$\mathbb{E}\left[G_t \mid \theta_t\right] = \nabla F_{s,\varepsilon}(\theta_t), \tag{50}$$

as claimed. □

B.4 SMOOTHING GAP INDUCED BY CONVERSION ERROR

Proof of Lemma 3.2. Let $S_\varepsilon(\theta)$ denote the difference between the Gaussian-smoothed and original spiking objectives:

$$S_\varepsilon(\theta) = F_{s,\varepsilon}(\theta) - F_s(\theta). \tag{51}$$

Recall from Equation 15 that $D_s(\theta) = F_s(\theta) - F_a(\theta)$. Insert the ANN objective at θ and its Gaussian smoothing to obtain the exact decomposition

$$\begin{aligned} S_\varepsilon(\theta) = F_{a,\varepsilon}(\theta) - F_a(\theta) \\ + \mathbb{E}_z \left[D_s(\theta + \varepsilon z) - D_s(\theta) \right]. \end{aligned} \tag{52}$$

We bound the two parts separately.

The L_a-smoothness of F_a gives the absolute remainder bound

$$|F_a(\theta + v) - F_a(\theta) - \langle \nabla F_a(\theta), v \rangle| \le \frac{L_a}{2} \|v\|^2. \tag{53}$$

Set $v = \varepsilon z$ and take the Gaussian expectation. Since $\mathbb{E}[z] = 0$ and $\mathbb{E}\|z\|^2 = d$,

$$|F_{a,\varepsilon}(\theta) - F_a(\theta)| \le \frac{L_a d \varepsilon^2}{2}. \tag{54}$$

By the triangle inequality, Jensen's inequality, and Lemma B.1,

$$\begin{aligned} \Omega_\varepsilon(\theta) &\le \mathbb{E}_z |D_s(\theta + \varepsilon z)| + |D_s(\theta)| \\ &\le \rho \mathbb{E}_{z,\xi} \|\Delta_L(\theta + \varepsilon z; \xi)\| + \rho \mathbb{E}_\xi \|\Delta_L(\theta; \xi)\| \\ &\le \rho \left(\sqrt{\mathcal{C}_\varepsilon(\theta)} + \sqrt{\mathcal{C}(\theta)} \right). \end{aligned} \tag{55}$$

Combining Equations 52, 54, and 55 proves

$$|S_\varepsilon(\theta)| \le \frac{L_a d \varepsilon^2}{2} + \Omega_\varepsilon(\theta) \le \frac{L_a d \varepsilon^2}{2} + \rho \left(\sqrt{\mathcal{C}_\varepsilon(\theta)} + \sqrt{\mathcal{C}(\theta)} \right), \tag{56}$$

which is the desired inequality for the smoothing gap. □

B.5 OPTIMIZATION OF THE GAUSSIAN-SMOOTHED SPIKING OBJECTIVE

We next bound the ZO-SGD optimization error on $F_{s,\varepsilon}$. The discrepancy between $F_{s,\varepsilon}$ and F_s is handled separately by Lemma 3.2.

Lemma B.3 (Convergence bound for the Gaussian-smoothed spiking objective)**.** *Let $\{\theta_t\}_{t=0}^K$ be generated by $\theta_{t+1} = \theta_t - \eta G_t$. Under Assumptions B.5–B.6 and the conditional unbiasedness established in Lemma B.2, if $0 < \eta \le 1/L_\varepsilon$ and $q := 1 - \mu_{\mathrm{PL}}\eta$, then for every $K \ge 0$,*

$$\mathbb{E}\left[F_{s,\varepsilon}(\theta_K) - F_{s,\varepsilon}^\star \right] \le q^K \left(F_{s,\varepsilon}(\theta_0) - F_{s,\varepsilon}^\star \right) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}}. \tag{57}$$

Proof. For brevity, write $H = F_{s,\varepsilon}$ and $H^\star = F_{s,\varepsilon}^\star$, and define the estimation error

$$\zeta_t = G_t - \nabla H(\theta_t). \tag{58}$$

Lemma B.2 and Assumption B.6 give

$$\mathbb{E}[\zeta_t \mid \theta_t] = 0, \qquad \mathbb{E}[\|\zeta_t\|^2 \mid \theta_t] \le \sigma_\varepsilon^2. \tag{59}$$

By L_ε-smoothness and the update $\theta_{t+1} = \theta_t - \eta G_t$,

$$H(\theta_{t+1}) \le H(\theta_t) - \eta \langle \nabla H(\theta_t), G_t \rangle + \frac{L_\varepsilon \eta^2}{2} \|G_t\|^2. \tag{60}$$

Conditioning on θ_t and using Equation 59,

$$\begin{aligned} \mathbb{E}[\|G_t\|^2 \mid \theta_t] &= \|\nabla H(\theta_t)\|^2 + \mathbb{E}[\|\zeta_t\|^2 \mid \theta_t] \\ &\le \|\nabla H(\theta_t)\|^2 + \sigma_\varepsilon^2. \end{aligned} \tag{61}$$

Taking the conditional expectation in Equation 60 therefore gives

$$\begin{aligned}\mathbb{E}[H(\theta_{t+1}) \mid \theta_t] &\le H(\theta_t) - \eta\left(1 - \frac{L_\varepsilon \eta}{2}\right)\|\nabla H(\theta_t)\|^2 + \frac{L_\varepsilon \eta^2 \sigma_\varepsilon^2}{2} \\ &\le H(\theta_t) - \frac{\eta}{2}\|\nabla H(\theta_t)\|^2 + \frac{L_\varepsilon \eta^2 \sigma_\varepsilon^2}{2},\end{aligned} \tag{62}$$

where the second inequality uses $\eta \le 1/L_\varepsilon$. The PL inequality in Assumption B.5 now yields

$$\mathbb{E}[H(\theta_{t+1}) - H^\star \mid \theta_t] \le (1 - \mu_{\mathrm{PL}}\eta)(H(\theta_t) - H^\star) + \frac{L_\varepsilon \eta^2 \sigma_\varepsilon^2}{2}. \tag{63}$$

The smoothness and PL assumptions imply $0 < \mu_{\mathrm{PL}} \le L_\varepsilon$ whenever the objective is nonconstant, so $q \in [0, 1)$. Taking total expectations and unrolling Equation 63 gives

$$\begin{aligned}\mathbb{E}[H(\theta_K) - H^\star] &\le q^K (H(\theta_0) - H^\star) + \frac{L_\varepsilon \eta^2 \sigma_\varepsilon^2}{2} \sum_{i=0}^{K-1} q^i \\ &\le q^K (H(\theta_0) - H^\star) + \frac{L_\varepsilon \eta^2 \sigma_\varepsilon^2}{2(1-q)} \\ &= q^K (H(\theta_0) - H^\star) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}},\end{aligned} \tag{64}$$

which proves Equation 57. □

B.6 Proof of the main theorem

Proof of Theorem 3.1. Let $\theta_s^\star \in \arg\min_\theta F_s(\theta)$. Add and subtract the smoothed objective to obtain the exact decomposition

$$\begin{aligned}F_s(\theta_K) - F_s^\star = &\underbrace{F_{s,\varepsilon}(\theta_K) - F_{s,\varepsilon}^\star}_{T_1} \\ &+ \underbrace{F_s(\theta_K) - F_{s,\varepsilon}(\theta_K)}_{T_2} \\ &+ \underbrace{F_{s,\varepsilon}^\star - F_s^\star}_{T_3}.\end{aligned} \tag{65}$$

Lemma B.3 bounds the expectation of the optimization term by

$$\mathbb{E}[T_1] \le q^K \left(F_{s,\varepsilon}(\theta_0) - F_{s,\varepsilon}^\star\right) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}}. \tag{66}$$

Lemma 3.2 gives the pointwise bound

$$\begin{aligned}T_2 &\le |F_s(\theta_K) - F_{s,\varepsilon}(\theta_K)| \\ &\le \frac{L_a d \varepsilon^2}{2} + \rho\left(\sqrt{\mathcal{C}(\theta_K)} + \sqrt{\mathcal{C}_\varepsilon(\theta_K)}\right).\end{aligned} \tag{67}$$

Finally, by the definition of $F_{s,\varepsilon}^\star$,

$$\begin{aligned}T_3 &= F_{s,\varepsilon}^\star - F_s(\theta_s^\star) \\ &\le F_{s,\varepsilon}(\theta_s^\star) - F_s(\theta_s^\star) \\ &\le \frac{L_a d \varepsilon^2}{2} + \rho\left(\sqrt{\mathcal{C}(\theta_s^\star)} + \sqrt{\mathcal{C}_\varepsilon(\theta_s^\star)}\right),\end{aligned} \tag{68}$$

where the last inequality again uses Lemma 3.2. Combining Equations 65–68 and taking expectations proves

$$\begin{aligned}\mathbb{E}[F_s(\theta_K) - F_s^\star] \le\ & q^K \left(F_{s,\varepsilon}(\theta_0) - F_{s,\varepsilon}^\star\right) + \frac{L_\varepsilon \eta \sigma_\varepsilon^2}{2\mu_{\mathrm{PL}}} + L_a d \varepsilon^2 \\ &+ \rho\, \mathbb{E}\left[\sqrt{\mathcal{C}(\theta_K)} + \sqrt{\mathcal{C}_\varepsilon(\theta_K)}\right] \\ &+ \rho\left(\sqrt{\mathcal{C}(\theta_s^\star)} + \sqrt{\mathcal{C}_\varepsilon(\theta_s^\star)}\right).\end{aligned} \tag{69}$$

Substituting $q = 1 - \eta\mu_{\mathrm{PL}}$ yields Equation 17. □

Proof of Corollary 3.1. The uniform bounds in Equation 18 imply

$$\mathbb{E}\left[\sqrt{\mathcal{C}(\theta_K)}+\sqrt{\mathcal{C}_\varepsilon(\theta_K)}\right]\le 2U\sqrt{\Psi}, \qquad \sqrt{\mathcal{C}(\theta_s^\star)}+\sqrt{\mathcal{C}_\varepsilon(\theta_s^\star)}\le 2U\sqrt{\Psi}. \tag{70}$$

Substituting these inequalities into Equation 17 proves the corollary. □

C Additional Experimental Results

C.1 Additional Implementation Details

All experiments are conducted on a cluster of NVIDIA A100 GPUs (80GB). We follow the sampling and evaluation protocol for downstream tasks in Section 3.3. For block-wise representation alignment, we sample 512 unlabeled sequences of length 512 or 1024 from WikiText-2 (Merity et al., 2016) for SpikeLLM and from the Pile (Gao et al., 2020) for SpikingLLM. Table 5 summarizes the hyperparameters used for high-resolution task learning and block-wise alignment. Braced entries denote the candidate learning rates considered for the corresponding configuration. The source code required to conduct the experiments is anonymously available at https://anonymous.4open.science/r/ZOSGDICLR-B58D.

Framework	Model	High-resolution task learning				Block-wise alignment	
		Batch size	Learning rate	Steps	ε	Learning rate	Epochs
SpikeLLM	Llama2-7B	8	1×10^{-7}	20k	1×10^{-3}	$\{0.1,0.5,1\}\times10^{-5}$	20
SpikeLLM	OPT-6.7B	8	1×10^{-7}	20k	1×10^{-3}	$\{0.5,0.8,1\}\times10^{-5}$	20
SpikingLLM	Llama2-7B	8	$\{1,2,3\}\times10^{-8}$	20k	1×10^{-4}	$\{1,2,5\}\times10^{-6}$	20

Table 5: Hyperparameter settings for high-resolution ZO-SGD task learning and block-wise representation alignment.

C.2 Detailed Results for ZO-SGD Fine-Tuning

Tables 6–8 report the results for individual tasks summarized in Table 1.

Method	Config	SST-2	RTE	BoolQ	WSC	WIC	MultiRC	COPA	SQuAD	**Average**
Zero-Shot	–	58.03	62.09	66.10	37.50	50.16	42.40	79.00	60.45	56.97
MeZO	–	86.58	59.21	69.90	60.58	56.43	64.60	82.00	81.30	70.08
Zero-Shot-S	N=256, T_{sp}=8	57.80	60.29	65.70	36.54	50.16	42.60	78.00	59.64	56.34
	N=16, T_{sp}=4	59.52	46.93	50.70	57.69	54.39	45.90	68.00	40.76	52.99
MeZO-S	N=256, T_{sp}=8	84.98	60.65	69.40	59.62	55.17	64.00	83.00	80.96	69.72
	N=16, T_{sp}=4	75.92	53.07	55.60	58.65	52.19	53.10	77.00	40.14	58.21
S-MeZO	N=256, T_{sp}=8	85.09	62.09	70.20	62.50	57.21	65.70	83.00	79.91	70.71
	N=16, T_{sp}=4	78.56	48.38	51.80	57.69	49.22	49.90	75.00	57.72	58.53

Table 6: Performance comparison on Llama2-7B across different tasks under the SpikeLLM framework.

Method	Config	SST-2	RTE	BoolQ	WSC	WIC	MultiRC	COPA	SQuAD	**Average**
Zero-Shot	–	61.24	55.23	57.30	37.50	51.25	41.70	82.00	40.57	53.35
MeZO	–	94.04	65.34	66.40	55.77	58.62	61.40	79.00	77.81	69.80
Zero-Shot-S	N=256, T_{sp}=8	61.58	55.96	60.00	39.42	51.41	40.90	81.00	40.62	53.86
	N=64, T_{sp}=4	60.21	50.90	58.60	37.50	51.88	42.90	79.00	39.60	52.57
MeZO-S	N=256, T_{sp}=8	93.92	64.26	65.40	46.15	54.55	61.50	77.00	76.93	67.46
	N=64, T_{sp}=4	93.35	59.93	65.70	52.88	57.05	57.20	78.00	77.68	67.72
S-MeZO	N=256, T_{sp}=8	94.38	65.34	66.60	54.81	58.62	61.30	80.00	78.75	69.98
	N=64, T_{sp}=4	88.88	59.21	62.90	49.04	51.88	57.30	79.00	71.42	64.95

Table 7: Performance comparison on OPT-6.7B across different tasks under the SpikeLLM framework.

Method	Config	SST-2	RTE	BoolQ	WSC	WIC	MultiRC	COPA	**Average**
Zero-Shot-S	T_{sp}=8	57.80	55.60	64.80	38.46	50.16	43.30	82.00	56.02
	T_{sp}=6	57.11	54.87	62.90	38.46	50.00	46.10	80.00	55.63
MeZO-S	T_{sp}=8	83.60	59.57	68.90	59.62	55.49	63.90	84.00	67.87
	T_{sp}=6	82.34	55.60	66.60	58.65	54.23	59.80	82.00	65.60
S-MeZO	T_{sp}=8	80.50	62.82	67.20	59.62	56.74	64.20	82.00	67.58
	T_{sp}=6	67.55	56.32	64.50	54.81	52.82	51.30	83.00	61.47

Table 8: Performance comparison on Llama2-7B across different tasks under the SpikingLLM framework.