ConMeZO: Adaptive Directional Sampling for Gradient-Free Finetuning of Language Models

Lejs Deen Behric [1] **Liang Zhang** [1] **Bingcong Li** [1] **Kiran Koshy Thekumparampil** [2]

Abstract

Zeroth-order optimization (MeZO) is an attractive strategy for finetuning large language models (LLMs) because it eliminates the memory overhead of storing intermediate activations required by backpropagation. However, it converges slowly due to the inherent curse of dimensionality when searching for descent directions in the higher-dimensional parameter space of billion-scale LLMs. We propose ConMeZO, a novel zeroth-order optimizer that accelerates convergence by adaptive directional sampling. Instead of drawing the direction uniformly at random, ConMeZO restricts the sampling to a cone centered around a momentum estimate. This concentrates the search in the directions where the true gradient is more likely to lie and thus reduces the effect of higher dimensions. We analytically prove that ConMeZO achieves the same worst-case convergence rate as MeZO. Empirically, when finetuning LLMs on natural language benchmarks, ConMeZO is up to 2x faster than MeZO while retaining the low-memory footprint of zeroth-order methods.

1. Introduction

Fine-tuning LLMs enables pre-trained models such as LLaMA (Touvron et al., 2023a;b; Grattafiori et al., 2024) and Gemma (Team et al., 2024a;b; 2025) to excel in diverse tasks. However, fine-tuning methods face significant challenges due to their high computational and memory demands. Substantial GPU resources are required for gradient computation and storing activation, often exceeding the budgets when only consumer-grade GPUs are available.

Zeroth-order optimization (ZO) methods, such as those employed by MeZO (Malladi et al., 2023), offer a promising alternative. By relying only on forward passes to estimate gradients, ZO methods bypass the memory-intensive backward pass, facilitating fine-tuning in resource-constrained scenarios. However, ZO methods suffer from high variance in gradient estimates, leading to slower convergence. As shown (Malladi et al., 2023, Table 15), while it takes 1K iterations with Adam to fine-tune a RoBERTa-large model to desirable accuracy, MeZO requires 100K iterations for comparable performance. Consequently, the overall runtime of MeZO can be significantly longer than Adam.

This work aims to address the runtime inefficiency of ZO methods while preserving their memory benefits. Traditional ZO methods typically rely on random search directions sampled from either a sphere or Gaussian distribution. Such random strategies, especially in the high-dimensional regime, is the bottleneck for the slow convergence of ZO. We propose reducing gradient variance by constraining random search directions within a cone centered on a promising search direction, defined by a momentum vector. This strategy improves convergence by narrowing the search space while maintaining the flexibility of ZO optimization. The proposed approach, coined ConMeZO, significantly reduces iteration counts while retaining their memory efficiency. Combining theoretical analysis and empirical validation, we contribute to advancing efficient and accessible fine-tuning methods for LLMs. Our contributions are summarized as:

1. **Algorithm design:** A distinctive cone-sampling strategy inspired by geometrical principles, which focuses search directions to areas more likely to yield productive updates. This approach not only reduces noise but also preserves the simplicity of ZO optimization, making it both efficient and theoretically sound.
2. **Theoretical analysis:** Unlike optimizers such as MeZO, whose convergence rates suffer from curse of dimensionality, we show that under benign settings our ConMeZO can provide up to $O(d)$ speed up.
3. **Improved practical performance:** Experiments on fine-tuning LLMs demonstrate faster convergence of ConMeZO, especially in early iterations. ConMeZO ultimately achieves up to 2x speedup over MeZO.

[1] Department of Computer Science, ETH Zurich [2] Amazon Search. Correspondence to: Kiran Koshy Thekumparampil <kkt@amazon.com>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

Related work. Our work falls in the broad field of zeroth-order (ZO) optimization (Nesterov & Spokoiny, 2017) and its application to LLM fine-tuning (Malladi et al., 2023). Given the space limitation, a more careful treatment of related work can be found in Appendix A.3.

Notation We use $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ for the Euclidian norm and the inner product, respectively. Let $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d \mid \|x\| = 1\}$ be the unit sphere in $\mathbb{R}^d$ and $r\,\mathbb{S}^{d-1}$ the sphere of radius $r > 0$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is ℓ-smooth iff it is differentiable and $\|\nabla f(x_1) - \nabla f(x_2)\| \leq \ell\|x_1 - x_2\|$ for all $x_1, x_2 \in \mathbb{R}^d$. The orthogonal complement of a vector $x \in \mathbb{R}^d$, denoted $(x)^\perp$ is given by $(x)^\perp = \{v \in \mathbb{R}^d \mid \langle x, v\rangle = 0\}$. $\mathcal{N}(0, I_d)$ and $\mathcal{U}(\mathcal{S})$ denotes the standard d-dimennsional Gaussian or Uniform distribution over a set $\mathcal{S}$, respectively. All proofs are deffered to appendix.

2. Problem Formulation

We consider the *zeroth-order (ZO) optimization* problem $\min_{x\in\mathbb{R}^d} f(x)$, which emerges when direct access to the subgradient is unavailable due to e.g., memory constraints on GPUs. We assume that f is differentiable, and it can be accessed via a ZO oracle which computes $f(x)$ at any given x (Nesterov & Spokoiny, 2017). ZO problem is usually solved by applying gradient descent (GD) using a ZO gradient estimator. We use the following popular stochastic ZO estimator which perturbs the point along randomly sampled directions (Nesterov & Spokoiny, 2017; Duchi et al., 2015).

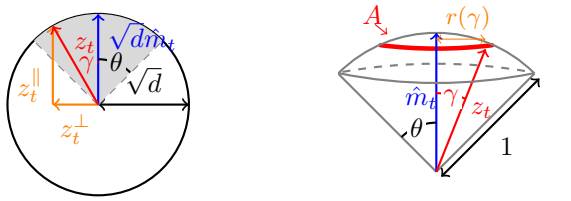
Definition 2.1. Stochastic ZO gradient estimate (ZOGE) of a function f at x using z randomly from an isotropic distributions like $\mathcal{N}(0, I_d)$ or $\mathcal{U}(\sqrt{d}\mathbb{S}^{d-1})$ and a smoothing parameter $\lambda > 0$ is given by

$$g_\lambda(x, z) = (1/2\lambda) \cdot (f(x + \lambda z) - f(x - \lambda z)) \cdot z$$

This estimator requires only two evaluations of $f(x)$ regardless of the dimensionality d. It avoids explicit gradient computation, reducing memory and compute overheads. Further, it is known that $\lim_{\lambda\to 0} g_\lambda(x, z)$ is unbiased.

Lemma 2.2. *(Zhang et al., 2024a) When λ is sufficiently small, $g_\lambda(x, z) \approx (z^\top \nabla f(x))z$. Further, first two moments of this term satisfies $\mathbb{E}_z[(z^\top \nabla f(x))z] = \nabla f(x)$ and $\mathbb{E}_z[\|(z^\top \nabla f(x))z\|^2] \leq 2d\|\nabla f(x)\|^2$.*

Despite the benefits mentioned above, this gradient estimator suffers from high $O(d)$ variance, especially in high-dimensional settings of LLM finetuning. This variance leads to $O(d)$ slower convergence rate than first-order methods. Addressing this limitation is crucial for making ZO optimization competitive in practical scenarios (Malladi et al., 2023). In next sections we overcome this limitation by constraining the search direction z around the true gradient direction estimated via momentum, while still retaining the memory advantage of ZO methods over first-order approaches.



(a) (b)

Figure 1: 2D- and 3D-representation of the cone-sampling approach. (a) Sphere with radius $\sqrt{d}$ and (gray) search space cone of half-angle θ around $\hat{m}_t$. (b) 3D representation of cone sampling in red area.

3. The Cone Sampling Approach

The key idea is to leverage a promising search direction, captured by a momentum m_t accumulating past gradients, and constrain the random search direction z_t at step t to a cone of fixed half-angle θ around the unit vector $\hat{m}_t = m_t/\|m_t\|$ along m_t. In Section 4, we show that cone sampling reduces the variance of the gradient estimate by focusing the search in a region of higher likelihood for productive updates, thus balancing exploration and exploitation more effectively.

The cone sampling approach introduces two main algorithmic components: i) **Promising search direction**, where the momentum vector m_t acts as a predictor of beneficial future search directions; and ii) **Cone restriction** that constrains the search space of the perturbation direction z_t to a cone with an apex at the origin, central axis $\sqrt{d}\,\hat{m}_t$, and half-angle θ. This reduces the variance by limiting the set of possible directions to those more aligned with $\hat{m}_t$. The following subsections derive the mathematical framework for cone sampling. The proposed approach seamlessly integrates with the ZOGE (Definition 2.1).

3.1. Search Direction

We construct the promising search direction using a momentum m_t computed as the exponentially moving averaging of past gradient estimates $g(x_t, z_t)$ as follows

$$m_{t+1} \leftarrow \beta \cdot m_t + (1 - \beta) \cdot g(x_t, z_t). \tag{1}$$

We choose momentum since it reduces variance of stochastic GD when training neural networks (Tieleman, 2012; Kingma & Ba, 2014; Cutkosky & Orabona, 2019).

3.2. Sampling Method

Next, we discuss how to sample from intersection of the sphere $\sqrt{d}\mathbb{S}^{d-1}$ and the cone with a central axis along the direction $\hat{m}_t$ and half-angle θ. Using the 2D Figure 1a as a geometrical reference, we see that the random vector z_t can be split into two additive parts: $z_t^{\parallel}$ and $z_t^{\perp}$, representing the component of z_t parallel and perpendicular to m_t, respec-

Algorithm 1 ConMeZO

Input: Cone angle $\theta \in [0, \frac{\pi}{2}]$, momentum parameter $\beta \in [0, 1]$, learning rate η, smoothing parameter $\lambda > 0$.
for $t = 0, \dots, T$ **do**
  $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$ // Sample u_t
  $[m_0 \leftarrow u_0]_{t=0}$
  $z_t \leftarrow \cos(\theta)\sqrt{d} \cdot \hat{m}_t + \sin(\theta) \cdot u_t$ // $\hat{m}_t \leftarrow m_t/\|m_t\|$
  $x \leftarrow x - \eta \cdot g_\lambda(x, z_t)$
  $m_{t+1} \leftarrow \beta \cdot m_t + (1-\beta) \cdot g_\lambda(x, z_t)$
end for

tively. First consider the case when the angle γ between z_t and m_t is fixed. We can see that $\cos(\gamma) = \|z_t^{\parallel}\|/\sqrt{d}$ and $\sin(\gamma) = \|z_t^{\perp}\|/\sqrt{d}$. Then, it is easy to argue that $z_t^{\parallel} = \cos(\gamma)\sqrt{d}\hat{m}_t$ and $z_t^{\perp}$ can be sampled as $z_t^{\perp} = \sin(\gamma)u_t^{\perp}$, where $u_t^{\perp} \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1} \cap (\hat{m}_t)^{\perp})$, and $(\hat{m}_t)^{\perp}$ denotes the subspace orthogonal to $\hat{m}_t$. Since $u_t^{\perp} \perp \hat{m}_t$, we can easily verify $z_t \in \sqrt{d}\mathbb{S}^{d-1}$. Now the questions boil down to sample such a $u_t^{\perp}$ and to introduce randomness in the angle γ. To achieve these, we make two justified simplifications.

Sampling a $u_t^{\perp} \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1} \cap (\hat{m}_t)^{\perp})$. Instead of sampling a random vector $u_t^{\perp}$ of magnitude $\sqrt{d}$ orthogonal to $\hat{m}_t$, we sample $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$. This is an appropriate simplification, because in very-high d dimensions almost all vectors in $\sqrt{d}\,\mathbb{S}^{d-1}$ are orthogonal to any fixed unit vector.

Proposition 3.1. *The cosine similarity between a randomly sampled vector $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$ and a fixed unit vector $\hat{m}_t$ satisfies $\langle \hat{m}_t, u_t \rangle / \|u_t\| \to 0$ as $d \to \infty$.*

Introducing randomness into the sampling angle γ. Instead of fixing a specific angle between $\hat{m}_t$ and z_t, we originally wanted this γ to be a variable value in $[0, \theta]$. Again due to the high dimensionality, most of the probability mass of the distribution of γ concentrates sharply near the edge of the cone, making it suitable to set $\gamma = \theta$ in practice.

To formally prove this, we look at the cumulative density of the angle γ if z_t is uniformly sampled from the intersection of the d-dimensional unit (for simplicity) sphere with the cone of angle θ. A graphical illustration can be found in Figure 1b. This cumulative density $p(\gamma \leq \theta')$ is equal to the area on the higher dimensional spherical cap (denoted with A in Figure 1b) divided by the overall area of the surface . Generalizing this to higher dimensions, the distribution of γ converges to the Dirac delta centered around θ.

Proposition 3.2. *Consider a cone $\mathcal{C}$ with apex at the origin, central axis aligned with a unit vector $\hat{m}_t$, and half-angle θ. If a random vector z_t is sampled uniformly from $\mathcal{C} \cap \mathbb{S}^{d-1}$, then as $d \to \infty$, the angle γ between z_t and $\hat{m}_t$ satisfies $\gamma = \theta$ almost surely.*

A direct consequence of Proposition 3.2 is that γ can be treated as equal to θ given a large d. Let $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$, the random direction z_t can thus be obtained by:

$$z_t \leftarrow \cos(\theta)\sqrt{d} \cdot \hat{m}_t + \sin(\theta) \cdot u_t. \tag{2}$$

This greatly simplifies the conceptually complicated cone sampling, facilitating its practical implementation. As zeroth-order gradient estimate, we use the ZEGO (Definition 2.1) with this newly constructed search direction z_t. The resultant ConMeZO method is given in Algorithm 1.

4. Theoretical Analysis

Next, we analyze Algorithm 1 on ℓ-smooth (potentially nonconvex) minimization problems. Our analysis shows that ConMeZO may decrease the objective at a step faster than MeZO (Malladi et al., 2023), when the momentum and the true gradient align. Further, we show that MeZO does not have better worst-case convergence rate than ConMeZO.

We begin with insights on the variance properties of ConMeZO. For notational convenience, we denote $a_t = \nabla f(x_t)$ and assume that the smoothing parameter λ of the ZOGE (Definition 2.1) is infinitesimally small, i.e., $\lambda \to 0$.

Lemma 4.1. *Suppose that z_t is given by (2). Then, the first moment of the ZOGE can be bounded via*

$$\mathbb{E}_{u_t}\left[(z_t^\top a_t) z_t\right] = d\cos^2(\theta)(\hat{m}_t^\top a_t) \cdot \hat{m}_t + \sin^2(\theta) \cdot a_t.$$

Moreover, the second moment can be bounded by

$$\mathbb{E}_{u_t}\left[\left\|(z_t^\top a_t) z_t\right\|^2\right] \leq d\left((d+4)\cos^2(\theta)\cos^2(\rho_t) + \sin^2(\theta)\right)\|a_t\|^2,$$

where ρ_t denotes the angle between a_t and $\hat{m}_t$.

This lemma reveals that the estimator introduces a bias that is influenced by the momentum vector m_t. This direction alignment bias wrt the true gradient a_t is minimized when $\hat{m}_t$ is well aligned with a_t. The parameter θ adjusts the relative magnitude of these components to optimize the estimator's behavior. Building on this, we show that the per-step objective decrease of ConMeZO is $O(\|a_t\|^2)$ whenever $\hat{m}_t$ and a_t are reasonably aligned ($\rho_t \gg 0$) and when $\sin\theta$ is small. Note that this is better than the $O(\|a_t\|^2/d)$ descent of MeZO (Nesterov & Spokoiny, 2017).

4.1. Convergence Guarantee

This subsection tackles theoretical guarantees on the descent per iteration and the global convergence of the algorithm.

Theorem 4.2 (Descent Lemma)**.** *Assume f is ℓ-smooth and let $\lambda \to 0$. Choose η via (4) in appendix, and let $\mathcal{D}_t := \mathbb{E}_{u_t}[f(x_{t+1})] - f(x_t)$. ConMeZO ensures that:*

$$\mathcal{D}_t \lesssim -\frac{(d\cos^2(\theta)\cos^2(\rho_t) + \sin^2(\theta))}{2\ell d}\|a_t\|^2.$$

Table 1: Test performance of ConMeZO on RoBERTa Large after 10K iterations, averaged over 5 seeds.

	SST-2	SST-5	SNLI	MNLI	RTE	TREC
AdamW	93.1	56.6	86.4	81.4	83.6	95.9
MeZO	92.5	50.8	80.4	69.2	72.8	88.9
Cone	**93.0**	49.8	**80.8**	**73.6**	**74.5**	**89.9**

When $\theta = 0$, Theorem 4.2 shows that the expected improvement per iteration is only determined by the squared cosine similarity $\cos^2(\rho_t)$ between the momentum and the true gradient. We find that $\cos^2(\rho_t)$ could be larger than $1/d$, especially during the early phase of convergence; see Figure 3 for an example. In such steps, the per-step decrease in the objective becomes dimension independent and $O(d)$ times larger than that of MeZO, leading to a faster convergence.

We now establish the overall convergence guarantee of our algorithm over T iterations. This guarantee not only matches the rate of MeZO but also demonstrates potential improvements under certain conditions related to the alignment between the momentum vector and the true gradient.

Corollary 4.3. *Assume f is ℓ-smooth, $\min_{x\in\mathbb{R}^d} f(x) > -\infty$, and $\lambda \to 0$. There exists a hyperparameter setting, such that after T iterations, ConMeZO ensures:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}_{u_t}\left[\|\nabla f(x_t)\|^2\right] \leq \frac{2\ell d(f(x_0) - f^\star)}{T}. \quad (3)$$

Theorem 4.2 and Corollary 4.3 hint that the convergence rate could be even faster than MeZO, especially when the momentum m_t can be well aligned with the gradient $\nabla f(x_t)$, i.e. $\cos^2 \rho_t \gg 1/d$, for a sufficient fraction of the iterations.

5. Experimental Results

Our ConMeZO is tested on fine-tuning tasks. For space limitation, experimental details can be found in Appendix.

5.1. RoBERTa

We start with RoBERTa-large (Liu et al., 2019b), a 355M model. Similar to (Malladi et al., 2023), we tackle a few-shot setting with 512 samples per class, on the GLUE benchmark (Wang et al., 2018a). Following common practice (Zhang et al., 2024a), random directions are sampled from a standard normal distribution instead of the sphere.

After 10K iterations, our ConMeZO achieves an average improvement of $\geq 1\%$ test accuracy over MeZO across the tasks, as shown in Table 1. This demonstrates its capability to achieve higher accuracy consistently across diverse tasks. Moreover, ConMeZO outperforms MeZO at every evaluated checkpoint (see Table 2 and Figure 4 in appendix).

Table 2: Accuracy (%) of ConMeZO vs. MeZO on RoBERTa at different steps for one fixed seed.

	SST-2		SST-5		SNLI		MNLI		RTE		TREC	
Steps	MeZO	Cone	MeZO	Cone	MeZO	Cone	MeZO	Cone	MeZO	Cone	MeZO	Cone
1500	88.3	**90.1**	44.7	**46.0**	68.7	**71.6**	57.2	**58.4**	63.2	**65.0**	53.6	**65.6**
3000	90.8	**92.1**	48.1	**49.5**	73.3	**76.7**	61.0	**64.3**	66.1	**68.2**	68.0	**78.4**
6000	91.4	**92.7**	50.3	**50.9**	77.6	**78.8**	65.5	**69.8**	69.3	**70.0**	84.2	**87.2**

Table 3: Comparison of ConMeZO vs. MeZO on OPT-1.3B.

Task	**Optim.**	4K	6K	14K	20K
SST2 (Acc)	MeZO	68.6	75.3	84.5	88.0
	Cone	**74.8**	**80.3**	**86.5**	**88.9**
BoolQ (Acc)	MeZO	60.7	62.3	62.6	62.6
	Cone	**61.3**	**62.3**	**63.3**	**63.3**
DROP (F1)	MeZO	23.1	24.6	**25.9**	25.9
	Cone	**24.7**	**25.3**	25.8	**26.3**
SQuAD (F1)	MeZO	56.7	62.2	70.0	73.0
	Cone	**64.5**	**68.3**	**73.5**	**75.4**

Overall, ConMeZO represents a sweet spot between convergence speed and memory efficiency: slightly higher memory consumption and runtime per iteration enable faster early-stage convergence and long-term accuracy improvements. ConMeZO consumes 4640MiB of memory and has a per-iteration runtime of 0.52s, compared to 2670MiB and 0.37s for MeZO. However, these differences remain within the same order as MeZO and are still far from the significantly higher demands of first-order optimizers. For instance, fine-tuning RoBERTa with AdamW requires 15820 MiB memory and incurs a per-iteration runtime of approximately 1.25s.

5.2. OPT

Fine-tuning is then conducted on the OPT-1.3B model (Zhang et al., 2022). Across all four benchmarks ConMeZO delivers the highest final accuracy/F1 on every task (Table 3). The gains are most crucial in the earliest stages of training. The steep slope of the SQuAD learning curve in Figure 5 (in appendix) highlights ConMeZO reaches MeZO's 16K step performance in less than 8K steps, yielding a 2× speed-up. For practitioners operating under strict compute budgets or early-stopping regimes, this translates directly into fewer forward/backward passes for the same accuracy.

6. Conclusion

This work explores the challenges and opportunities in fine-tuning LLMs using ZO optimization. By introducing a novel cone-sampling strategy, it mitigates the high variance of traditional random-direction estimators and leverages momentum to guide updates more effectively. Empirical evaluations on RoBERTa-large and the substantially larger OPT-1.3B model show that our method consistently outperforms state-of-the-art MeZO, achieving up to a 2× speedup in early convergence and accuracy gains across benchmarks.

References

Alabdulkareem, A. and Honorio, J. Information-theoretic lower bounds for zero-order stochastic gradient estimation. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pp. 2316–2321. IEEE, 2021.

Balasubramanian, K. and Ghadimi, S. Zeroth-order (non)-convex stochastic optimization via conditional gradient and gradient updates. *Advances in Neural Information Processing Systems*, 31, 2018.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, 2015.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the ACM Workshop on Artificial Intelligence and Security*, pp. 15–26, 2017.

Chen, X., Liu, S., Xu, K., Li, X., Lin, X., Hong, M., and Cox, D. ZO-AdaMM: Zeroth-order adaptive momentum method for black-box optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Chen, Y., Zhang, Y., Cao, L., Yuan, K., and Wen, Z. Enhancing zeroth-order fine-tuning for language models with low-rank structures. *arXiv preprint arXiv:2410.07698*, 2024.

Choromanski, K., Rowland, M., Sindhwani, V., Turner, R., and Weller, A. Structured evolution with compact architectures for scalable policy optimization. In *International Conference on Machine Learning*, pp. 970–978. PMLR, 2018.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2924–2936, 2019.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2368–2378, 2019.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., and Wibisono, A. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Fang, W., Yu, Z., Jiang, Y., Shi, Y., Jones, C. N., and Zhou, Y. Communication-efficient stochastic zeroth-order optimization for federated learning. *IEEE Transactions on Signal Processing*, 70:5058–5073, 2022.

Flaxman, A. D., Kalai, A. T., and McMahan, H. B. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 385–394, 2005.

Gautam, T., Park, Y., Zhou, H., Raman, P., and Ha, W. Variance-reduced zeroth-order methods for fine-tuning language models. In *International Conference on Machine Learning*, 2024.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Golovin, D., Karro, J., Kochanski, G., Lee, C., Song, X., and Zhang, Q. Gradientless descent: High-dimensional zeroth-order optimization. In *International Conference on Learning Representations*, 2020.

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

Grill, J.-B., Valko, M., and Munos, R. Black-box optimization of noisy functions with unknown smoothness. *Advances in Neural Information Processing Systems*, 28, 2015.

Jamieson, K. G., Nowak, R., and Recht, B. Query complexity of derivative-free optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

Ji, K., Wang, Z., Zhou, Y., and Liang, Y. Improved zeroth-order variance reduced algorithms and analysis for non-convex optimization. In *International Conference on Machine Learning*, pp. 3100–3109. PMLR, 2019.

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

Li, Z., Zhang, X., Zhong, P., Deng, Y., Razaviyayn, M., and Mirrokni, V. Addax: Utilizing zeroth-order gradients to improve memory efficiency and performance of sgd for fine-tuning language models. *arXiv preprint arXiv:2410.06441*, 2024.

Lin, T., Zheng, Z., and Jordan, M. Gradient-free methods for deterministic and stochastic nonsmooth nonconvex optimization. *Advances in Neural Information Processing Systems*, 35:26160–26175, 2022.

Liu, S., Kailkhura, B., Chen, P.-Y., Ting, P., Chang, S., and Amini, L. Zeroth-order stochastic variance reduction for nonconvex optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Liu, S., Chen, P.-Y., Chen, X., and Hong, M. SignSGD via zeroth-order oracle. In *International Conference on Learning Representations*, 2019a.

Liu, S., Chen, P.-Y., Kailkhura, B., Zhang, G., Hero III, A. O., and Varshney, P. K. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. *IEEE Signal Processing Magazine*, 37(5):43–54, 2020.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Liu, Y., Zhu, Z., Gong, C., Cheng, M., Hsieh, C.-J., and You, Y. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. *arXiv preprint arXiv:2402.15751*, 2024.

Ma, S. and Huang, H. Revisiting zeroth-order optimization: Minimum-variance two-point estimators and directionally aligned perturbations. In *The Thirteenth International Conference on Learning Representations*, 2025.

Malladi, S., Gao, T., Nichani, E., Damian, A., Lee, J. D., Chen, D., and Arora, S. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

Mania, H., Guy, A., and Recht, B. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Nesterov, Y. and Spokoiny, V. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

Park, S., Yun, J., Kim, S., Kundu, S., and Yang, E. Unraveling zeroth-order optimization through the lens of low-dimensional structured perturbations. *arXiv preprint arXiv:2501.19099*, 2025.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, 2016.

Salimans, T., Ho, J., Chen, X., Sidor, S., and Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Shamir, O. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, 2013.

Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., Sifre, L., Rivière, M., Kale, M. S., Love, J., et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024a.

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024b.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Tieleman, T. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26, 2012.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

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

Voorhees, E. M. and Tice, D. M. Building a question answering test collection. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 200–207, 2000.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2018a.

Wang, F., Shen, L., Ding, L., Xue, C., Liu, Y., and Ding, C. Simultaneous computation and memory efficient zeroth-order optimizer for fine-tuning large language models. *arXiv preprint arXiv:2410.09823*, 2024.

Wang, Y., Du, S., Balakrishnan, S., and Singh, A. Stochastic zeroth-order optimization in high dimensions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1356–1365. PMLR, 2018b.

Wang, Z., Balasubramanian, K., Ma, S., and Razaviyayn, M. Zeroth-order algorithms for nonconvex–strongly-concave minimax problems with improved complexities. *Journal of Global Optimization*, pp. 1–32, 2022.

Wibisono, A., Wainwright, M. J., Jordan, M., and Duchi, J. C. Finite sample convergence rates of zero-order stochastic optimization methods. *Advances in Neural Information Processing Systems*, 25, 2012.

Williams, A., Nangia, N., and Bowman, S. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 1112–1122, 2018.

Xu, M., Wu, Y., Cai, D., Li, X., and Wang, S. Federated fine-tuning of billion-sized language models across mobile devices. *arXiv preprint arXiv:2308.13894*, 2023.

Yue, P., Yang, L., Fang, C., and Lin, Z. Zeroth-order optimization with weak dimension dependency. In *Annual Conference on Learning Theory*, pp. 4429–4472. PMLR, 2023.

Zelikman, E., Huang, Q., Liang, P., Haber, N., and Goodman, N. D. Just one byte (per gradient): A note on low-bandwidth decentralized language model finetuning using shared randomness. *arXiv preprint arXiv:2306.10015*, 2023.

Zhang, L., Li, B., Thekumparampil, K. K., Oh, S., and He, N. DPZero: Private fine-tuning of language models without backpropagation. In *International Conference on Machine Learning*, 2024a.

Zhang, L., Li, B., Thekumparampil, K. K., Oh, S., Muehlebach, M., and He, N. Zeroth-order optimization finds flat minima. *arXiv preprint arXiv:2506.05454*, 2025.

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

Zhang, Y., Li, P., Hong, J., Li, J., Zhang, Y., Zheng, W., Chen, P.-Y., Lee, J. D., Yin, W., Hong, M., Wang, Z., Liu, S., and Chen, T. Revisiting zeroth-order optimization for memory-efficient LLM fine-tuning: A benchmark. In *International Conference on Machine Learning*, 2024b.

Zhao, Y., Dang, S., Ye, H., Dai, G., Qian, Y., and Tsang, I. W. Second-order fine-tuning without pain for llms: A Hessian informed zeroth-order optimizer. *arXiv preprint arXiv:2402.15173*, 2024.

A. Broader impact, limitation and future work

A.1. Limitations

Despite its significant advancements, the proposed cone-based ZO optimization method has several limitations that require further exploration.

First, the performance of the optimizer is highly dependent on the choice of hyperparameters such as the cone angle (θ) and momentum (β). Experiments reveal that certain configurations achieve faster convergence in the early stages of optimization (e.g., the first 2,000 steps), while others perform better when fine-tuning near the optimum. This suggests that static parameter choices may not fully exploit the potential of the optimizer across different optimization phases.

Another limitation lies in the lack of theoretical guarantees on the cosine similarity between the momentum vector and the true gradient. While empirical results indicate improved alignment, a rigorous analysis is missing, which leaves room for theoretical gaps in the method's convergence guarantees.

Finally, compared to MeZO, the proposed method requires slightly higher memory and runtime due to the additional computational overhead of the cone-sampling strategy and the storage of momentum. While these trade-offs are acceptable in many practical scenarios, they may limit applicability in extremely resource-constrained environments.

A.2. Broader Impact

This is a foundational research work studying the efficiency of optimization algorithms for finetuning large language models (LLMs). Therefore, we believe this work does not open any new avenues of broad impact which were not already touched by prior works on LLM and optimization algorithm efficiency. However, we would like to highlight that our work may increase the adoption of LLM finetuning by people or entities who have limited computational resources.

A.3. Related Work

Zeroth-order (ZO) optimization. The work of (Nesterov & Spokoiny, 2017) marked a foundational step in formally analyzing the convergence rate of zeroth-order methods, such as zeroth-order (stochastic) gradient descent (ZO-SGD). ZO-SGD substitutes gradients in SGD with their zeroth-order estimators. Building on this foundation, (Shamir, 2017) refined the analysis for nonsmooth convex functions, while (Lin et al., 2022) extended these insights to nonsmooth nonconvex functions. Contributions by (Ghadimi & Lan, 2013) further tackles smooth functions in stochastic settings. These works have show that for smooth problem, the squared norm of the gradient converges with a worst-case rate $O(d/T)$ where d is the number of dimensions in x (Nesterov & Spokoiny, 2017). In stark contrast, standard gradient descent has a rate of $O(1/T)$ (Nesterov, 2003). Further, there exists lower complexity results which prove that this dimension dependence is unavoidable (Jamieson et al., 2012; Wibisono et al., 2012; Duchi et al., 2015; Golovin et al., 2020; Alabdulkareem & Honorio, 2021) unless there are additional structural assumptions such as sparsity or low-rank Hessian (Wang et al., 2018b; Yue et al., 2023). More recently, Zhang et al. (2025) proved that zeroth-order methods converge to flat minima for convex and sufficiently smooth functions, where flat minima are defined as the minimizers that achieve the smallest trace of Hessian.

These research were motivated by the growing interest in zeroth-order methods, driven by practical challenges including the memory limitations imposed by fast differentiation techniques (Wang et al., 2018b; Liu et al., 2020). ZO has been enriched with various enhancements such as conditional gradient methods (Balasubramanian & Ghadimi, 2018), and variance reduction techniques (Liu et al., 2018; Fang et al., 2018; Ji et al., 2019). Other notable adaptations include the integration of SignSGD (Liu et al., 2019a) and applications to minimax optimization (Wang et al., 2022). Beyond algorithmic development, these methods have demonstrated utility across diverse domains, including black-box machine learning (Grill et al., 2015; Chen et al., 2017; 2019), bandit optimization (Flaxman et al., 2005; Shamir, 2017), reinforcement learning (Salimans et al., 2017; Choromanski et al., 2018; Mania et al., 2018), and distributed learning, where they mitigate communication overhead (Fang et al., 2022; Zelikman et al., 2023; Xu et al., 2023).

ZO for LLM fine-tuning. In the realm of ZO optimization for LLMs, various approaches have emerged, emphasizing memory efficiency and computational effectiveness. MeZO (Malladi et al., 2023) offers a breakthrough by eliminating backpropagation and significantly reducing memory requirements, but it suffers from slower convergence rates and sensitivity to high-dimensional noise. (Zhang et al., 2024b) provides a more comprehensive benchmark for evaluating the performance of ZO for LLMs fine-tuning, where they observe that directly combining ZO with momentum methods does not lead to

significant performance gain. (Liu et al., 2024) introduces Sparse MeZO, where only a carefully chosen subset of parameters is updated. LeZO (Wang et al., 2024) introduces a layer-wise sparse strategy to reduce computational overhead. (Gautam et al., 2024) integrate variance reduction to ZO optimizers and proposes MeZO-SVRG. (Zhao et al., 2024) proposes also the estimation of second-order information with ZO oracles to improve the performance of MeZO. Similarly, LOZO (Chen et al., 2024) incorporates low-rank gradient estimations, capturing the inherent low-dimensional structure of LLM gradients. The work of (Park et al., 2025) further develops a theoretical framework to characterize effectiveness of structural perturbations, such as sparsity and low rankness, in ZO approaches. It is also pointed out in (Ma & Huang, 2025) that effective perturbations in ZO should account for the (estimated) gradient directions, and they propose an approach that requires halving the minibatch data. DPZero (Zhang et al., 2024a) extends ZO optimization into the realm of differential privacy, addressing the dual challenges of memory efficiency and data privacy in fine-tuning LLMs. More recently, Addax strategically combined first-order and ZO steps to improve overall efficiency (Li et al., 2024).

B. Proofs

B.1. Proof of Proposition 3.1

Let $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$ and $\hat{m}_t \in \mathbb{R}^d$ with $\|\hat{m}_t\| = 1$ be a promising search direction. Instead of ensuring that the sampled random direction is orthogonal to $\hat{m}_t$, we show that it suffices to sample any random direction $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$. We show that the relative magnitude of the projection $\langle \hat{m}_t, u_t\rangle / \|u_t\|$ becomes negligible as $d \to \infty$.

Proof. Notice that $u_t = \sqrt{d}\frac{X}{\|X\|}$, where $X \sim \mathcal{N}(0, I_d)$. It holds that $\|u_t\| = \sqrt{d}$.

We have that

$$\frac{\langle \hat{m}_t, u_t\rangle}{\|u_t\|} = \frac{\langle \hat{m}_t, X\rangle}{\|X\|}.$$

$\langle \hat{m}_t, X\rangle$ is a $\mathcal{N}(0,1)$ random variable, since $\|\hat{m}_t\| = 1$, and $\|X\|^2$ is a χ^2-distributed random variable with d degrees of freedom. Therefore, for large d, the ratio $\langle \hat{m}_t, X\rangle / \|X\|$ is on the order of $\mathcal{N}(0,1)/\sqrt{d}$, which converges to 0 in probability as $d \to \infty$.

B.2. Proof of Proposition 3.2

Consider a cone $\mathcal{C}$ in $\mathbb{R}^d$ with apex at the origin, central axis aligned with a unit vector $\hat{m}_t$, and half-angle $\theta \in [0, \pi/2]$. We are interested in the distribution of the angle γ between a random vector z_t, sampled uniformly from the intersection of $\mathcal{C}$ with $\mathbb{S}^{d-1}$, and the axis $\hat{m}_t$. The following proof demonstrates that as the dimension $d \to \infty$, the angle γ becomes concentrated at θ.

Proof. We consider the case where $d \to \infty$:

$$\begin{aligned} p(\gamma \le \theta') &= \frac{\int_0^{\theta'} \text{ Surface area of hypercircle with radius } r(\alpha)\, d\alpha}{\int_0^{\theta} \text{ Surface area of hypercircle with radius } r(\beta)\, d\beta} \\ &= \frac{\int_0^{\theta'} C_d \cdot r(\alpha)^{d-1}\, d\alpha}{\int_0^{\theta} C_d \cdot r(\beta)^{d-1}\, d\beta} \end{aligned}$$

Where C_d is a constant dependent on d and independent of radius.

When inspecting Figure 1b it is simple to see that $r(\gamma) = \sin(\gamma)$. Now assume that $\theta' < \theta$. Further calculation yields

$$\begin{aligned} p(\gamma \le \theta') &= \frac{\int_0^{\theta'} (\sin(\alpha))^{d-1}\, d\alpha}{\int_0^{\theta} (\sin(\beta))^{d-1}\, d\beta} \\ &\le \frac{\theta' (\sin(\theta'))^{d-1}}{\int_0^{\theta} (\sin(\beta))^{d-1}\, d\beta} \end{aligned}$$

because $(\sin\alpha)^{d-1} \leq (\sin\theta')^{d-1}$ for $0 \leq \alpha \leq \theta'$, and the interval length is θ'. Let $s = \frac{\theta+\theta'}{2} \in (\theta', \theta)$. Now because $\sin(\beta)$ is increasing on $[0, \theta]$, on the sub-interval $[s, \theta]$ we have $(\sin(\beta))^{d-1} \geq (\sin(s))^{d-1}$.

Hence

$$\int_0^\theta (\sin(\beta))^{d-1}\, d\beta \geq \int_s^\theta (\sin(\beta))^{d-1}\, d\beta \geq (\theta - s)\,(\sin(s))^{d-1}.$$

We have that

$$\sin(s) > \sin(\theta'),$$

since $\theta > s > \theta'$.

Putting these together, we get

$$\begin{aligned} p(\gamma \leq \theta') &= \frac{\int_0^{\theta'} (\sin(\alpha))^{d-1}\, d\alpha}{\int_0^{\theta} (\sin(\beta))^{d-1}\, d\beta} \\ &\leq \frac{\theta'}{\theta - s} \left(\frac{\sin(\theta')}{\sin(s)}\right)^{d-1}. \end{aligned}$$

Since $\sin(s) > \sin(\theta')$, we have that

$$p(\gamma \leq \theta') \to 0 \text{ for } d \to \infty.$$

So instead of sampling γ, in practice we can set $\gamma = \theta$.

Proof of Lemma 4.1

Proof. We start with the first part of Lemma 4.1. It can be seen that

$$\begin{aligned} \mathbb{E}_{u_t}\left[(z_t^\top a_t) z_t\right] &= \mathbb{E}_{u_t}\left[\left(\left(\cos(\theta)\sqrt{d}\cdot \hat{m}_t + \sin(\theta)\cdot u_t\right)^\top a_t\right)\left(\cos(\theta)\sqrt{d}\cdot \hat{m}_t + \sin(\theta)\cdot u_t\right)\right] \\ &= d\cos^2(\theta)(\hat{m}_t^\top a_t)\cdot \hat{m}_t + \sin^2(\theta)\cdot \mathbb{E}\left[(u_t^\top a_t) u_t\right] \\ &= d\cos^2(\theta)(\hat{m}_t^\top a_t)\cdot \hat{m}_t + \sin^2(\theta)\cdot a_t. \end{aligned}$$

For the second part of Lemma 4.1, let $z_t = \alpha\hat{m}_t + \beta u_t$, where $u_t \sim \mathcal{U}(\sqrt{d}\,\mathbb{S}^{d-1})$, $\|\hat{m}_t\| = 1$, $\alpha = \cos(\theta)\cdot\sqrt{d}$ and $\beta = \sin(\theta)$. We now derive the second moment of $(z_t^\top a_t) z_t$.

$$\begin{aligned} & \mathbb{E}_{u_t}\left[\left\|(z_t^\top a_t) z_t\right\|^2\right] \\ = \;& \mathbb{E}_{u_t}\left[\left\|\left(\alpha(\hat{m}_t^\top a_t) + \beta(u_t^\top a_t)\right)(\alpha\hat{m}_t + \beta u_t)\right\|^2\right] \\ = \;& \mathbb{E}_{u_t}\left[\left(\alpha^2(\hat{m}_t^\top a_t)^2 + \beta^2(u_t^\top a_t)^2 + 2\alpha\beta(\hat{m}_t^\top a_t)(u_t^\top a_t)\right)\left(\alpha^2\|\hat{m}_t\|^2 + \beta^2\|u_t\|^2 + 2\alpha\beta(\hat{m}_t^\top u_t)\right)\right] \\ = \;& \alpha^2\left(\alpha^2(\hat{m}_t^\top a_t)^2 + \beta^2\|a_t\|^2 + 0\right) \\ & +\beta^2\left(\alpha^2(\hat{m}_t^\top a_t)^2 d + \beta^2 d\|a_t\|^2 + 0\right) \\ & +2\alpha\beta\left(0 + 0 + 2\alpha\beta(\hat{m}_t^\top a_t)^2\right) \\ = \;& d^2\cos^2(\theta)(\hat{m}_t^\top a_t)^2 + 4d\sin^2(\theta)\cos^2(\theta)(\hat{m}_t^\top a_t)^2 + d\sin^2(\theta)\|a_t\|^2 \\ = \;& d\cos^2(\theta)\left(d + 4\sin^2(\theta)\right)(\hat{m}_t^\top a_t)^2 + d\sin^2(\theta)\|a_t\|^2 \\ \leq \;& d(d+4)\cos^2(\theta)(\hat{m}_t^\top a_t)^2 + d\sin^2(\theta)\|a_t\|^2. \end{aligned}$$

Using $(\hat{m}_t^\top a_t)^2 = (\cos(\rho)\|\hat{m}_t\|\|a_t\|)^2 = \cos^2(\rho)\cdot\|a_t\|^2$, we have:

$$\mathbb{E}_{u_t}\left[\left\|(z_t^\top a_t) z_t\right\|^2\right] \leq d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\|a_t\|^2.$$

Proof of Theorem 4.2

We assume that the function f is ℓ-smooth, meaning:

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{\ell}{2}\|x_{t+1} - x_t\|^2.$$

Additionally, we use the update rule $x_{t+1} = x_t - \eta g_\lambda(x_t, z_t)$, where $g_\lambda(x_t, z_t)$ is the gradient estimate at x_t and $z_t = \cos(\theta)\sqrt{d} \cdot \hat{m}_t + \sin(\theta) \cdot u_t$, where $u_t \sim \mathcal{U}(\sqrt{d}\mathbb{S}^{d-1})$. Let $a_t = \nabla f(x_t)$. The proof derives the expected improvement in $f(x)$ per iteration under these assumptions.

Proof. Substituting the update rule $x_{t+1} = x_t - \eta g_\lambda(x_t, z_t)$ into the smoothness assumption:

$$f(x_{t+1}) \leq f(x_t) - \eta \nabla f(x_t)^\top g_\lambda(x_t, z_t) + \frac{\eta^2 \ell}{2}\|g_\lambda(x_t, z_t)\|^2.$$

Taking expectations with respect to u_t, the random search direction:

$$\mathbb{E}_{u_t}[f(x_{t+1})] \leq f(x_t) - \eta a_t^\top \mathbb{E}_{u_t}[g_\lambda(x_t, z_t)] + \frac{\eta^2 \ell}{2}\mathbb{E}_{u_t}[\|g_\lambda(x_t, z_t)\|^2].$$

Using the moments of $g_\lambda(x_t, z_t)$:

$$\mathbb{E}_{u_t}[(z_t^\top a_t) z_t] = d\cos^2(\theta)(\hat{m}_t^\top a_t) \cdot \hat{m}_t + \sin^2(\theta) \cdot a_t,$$

and

$$\mathbb{E}_{u_t}\left[\|(z_t^\top a_t) z_t\|^2\right] = d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\|a_t\|^2,$$

we substitute these into the inequality:

$$\begin{aligned}\mathbb{E}_{u_t}[f(x_{t+1})] \leq f(x_t) &- \eta\left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\|a_t\|^2 \\ &+ \frac{\eta^2 \ell}{2} d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\|a_t\|^2.\end{aligned}$$

This proves the first part of the theorem.

Next, rearranging the inequality to isolate $\|a_t\|^2$:

$$\begin{aligned}\mathbb{E}_{u_t}[f(x_t) - f(x_{t+1})] \geq \Big(&\eta\left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right) \\ &- \frac{\eta^2 \ell}{2} d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\Big)\|a_t\|^2.\end{aligned}$$

Thus,

$$\|a_t\|^2 \leq \frac{\mathbb{E}_{u_t}[f(x_t) - f(x_{t+1})]}{\eta\left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right) - \frac{\eta^2 \ell}{2} d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)}.$$

To maximize the denominator, observe that it is a concave quadratic function of η, achieving its maximum at:

$$\eta^* = \frac{d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)}{\ell d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)}. \tag{4}$$

Substituting $\eta = \eta^*$ and rearranging for $\mathbb{E}_{u_t}[f(x_{t+1})] - f(x_t)$ yields:

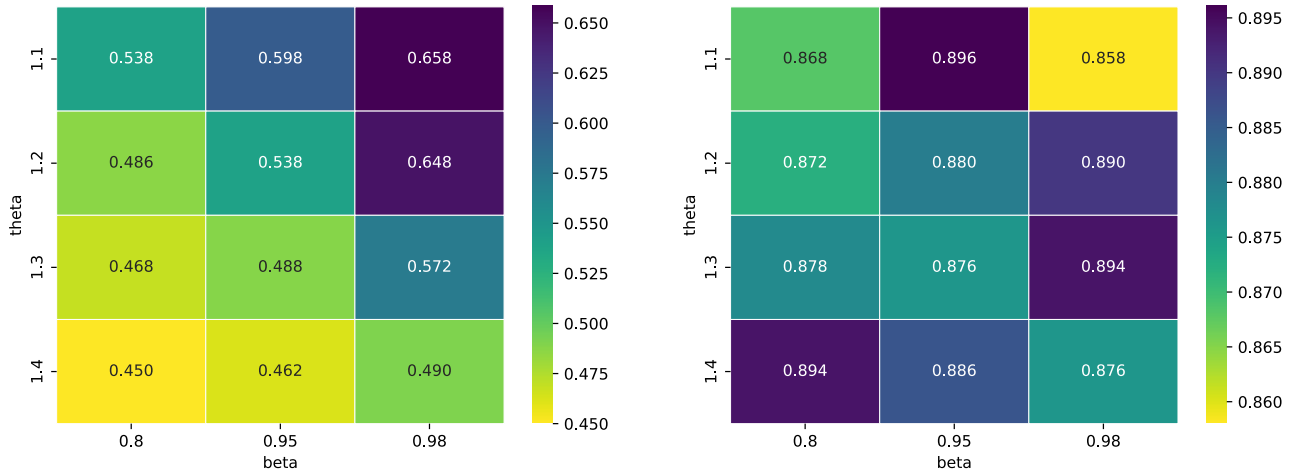
$$\begin{aligned}
\mathbb{E}_{u_t}[f(x_{t+1})] - f(x_t) &\leq -\left(\eta^* \left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right) - \frac{\eta^{*2}\ell}{2} d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)\right)\|a_t\|^2 \\
&= -\frac{\left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)^2}{2\ell d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)}\|a_t\|^2 \\
&\leq -\frac{\left(d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)\right)^2}{2\ell d\left((d+4)\cos^2(\theta)\cos^2(\rho) + \frac{d+4}{d}\sin^2(\theta)\right)}\|a_t\|^2 \\
&= -\frac{d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)}{2\ell d(1+4/d)}\|a_t\|^2 \\
&\approx -\frac{d\cos^2(\theta)\cos^2(\rho) + \sin^2(\theta)}{2\ell d}\|a_t\|^2
\end{aligned}$$

This completes the proof of the theorem.

Proof of Corollary 4.3

Proof. A proof directly follows from Theorem 4.2 by setting $\theta = \pi/2$ and telescoping across T iterations.

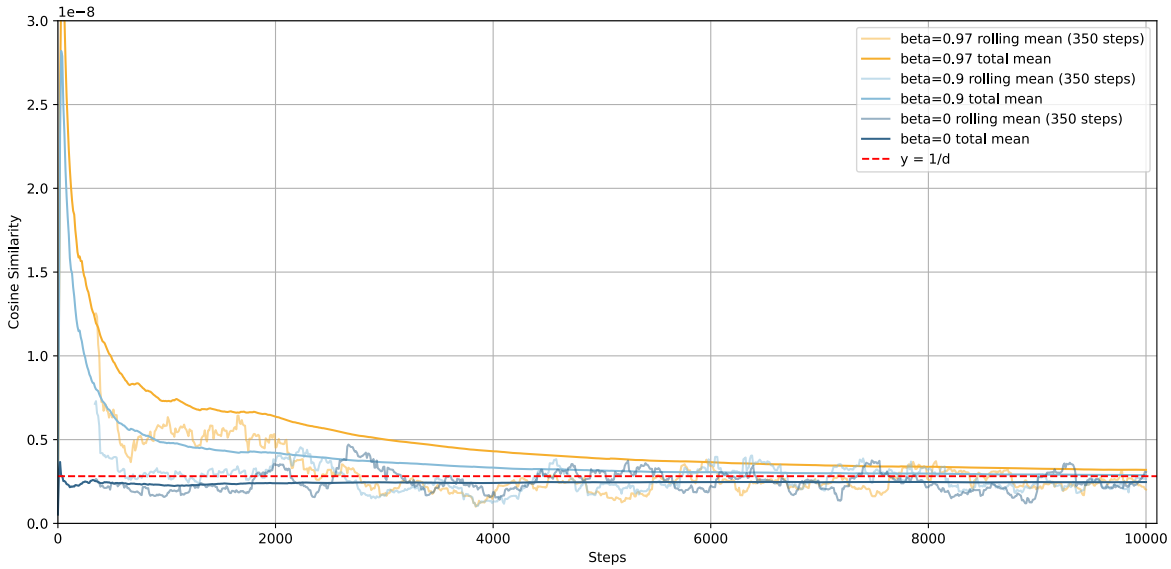
C. Figures



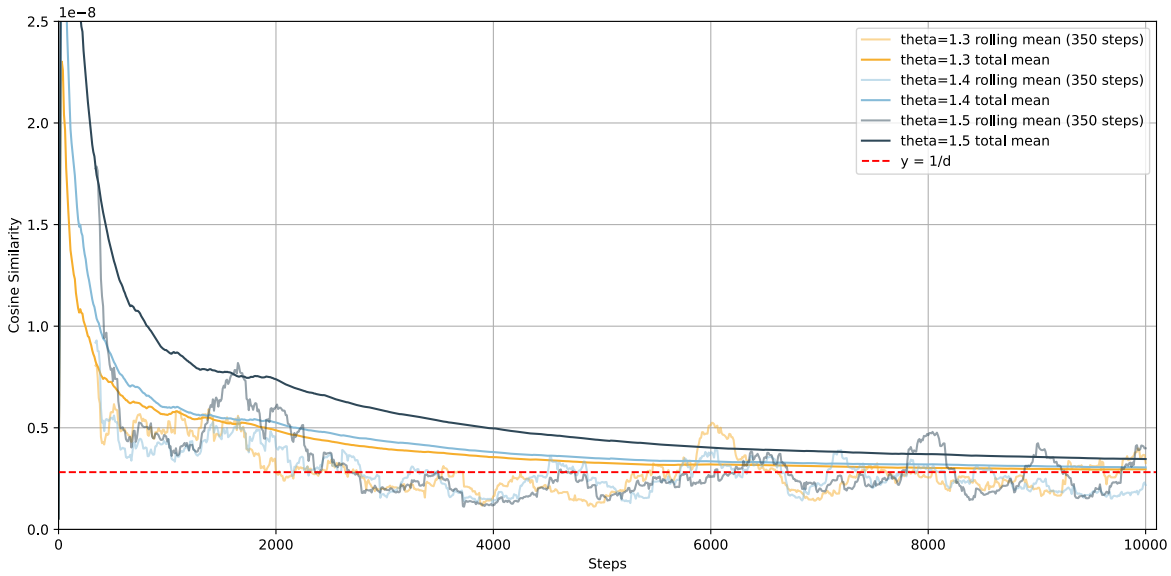
(a) After 1,000 iterations. MeZO's test accuracy after 1,000 iterations is 0.474.

(b) After 10,000 iterations. MeZO's test accuracy after 10,000 iterations is 0.89.

Figure 2: Heatmaps of Test Accuracy of ConMeZO on TREC dataset for different θ and β values and fixed learning rate $\eta = 10^{-6}$.



(a) TREC dataset for $\theta = 1.3$ and varying β.



(b) TREC dataset for $\beta = 0.95$ and varying θ.

Figure 3: Squared Cosine Similarity between real gradient and momentum vector during training.

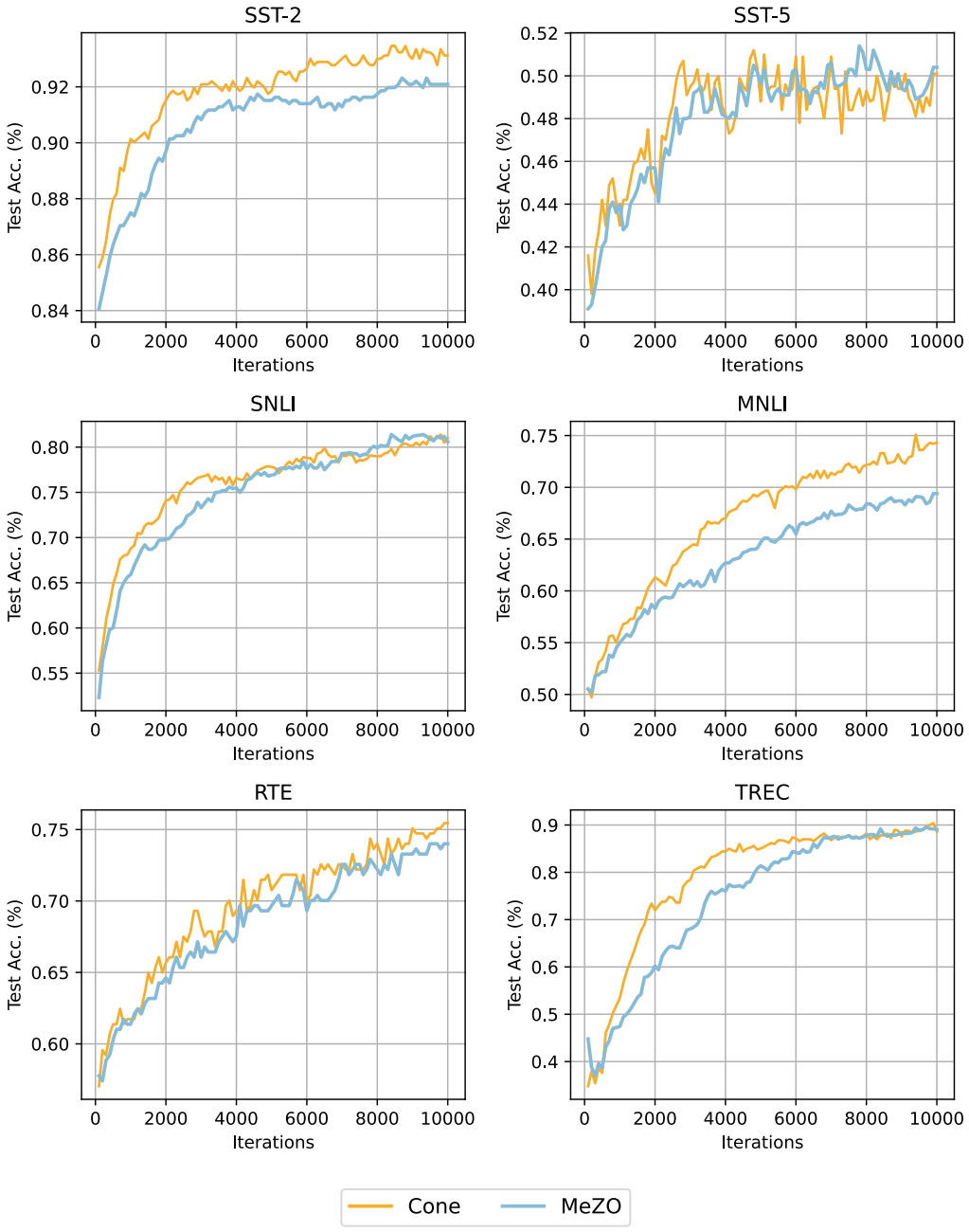


Figure 4: Test Accuracy of ConMeZO with settings mentioned in D.1 compared to MeZO over 10,000 iterations.

D. Experimental details

D.1. RoBERTa

The smoothing parameter λ was fixed to 10^{-3} for all experiments. Empirically best found hyperparameters for our optimizer are $\theta = 1.3, \beta = 0.97, \eta = 1.72 \cdot 10^{-6}$ and $\lambda = 10^{-3}$. All experiments are executed on a single NVIDIA RTX 3090 GPU with 24 GiB of memory.

Our implementation builds heavily on the framework provided by DPZero paper (Zhang et al., 2024a) and can be found at `https://anonymous.4open.science/r/conmezo`.

The optimizer's performance was analyzed under varying configurations of its hyperparameters: θ, β, and the learning rate η. These parameters were systematically adjusted to evaluate their sensitivity and impact on model performance.

A robust configuration we found was $\theta = 1.3584506581, \beta = 0.97, \eta = 1.71375681 \times 10^{-6}$. We used this configuration and the seeds $13, 21, 42, 87, 100$ to calculate values provided in Figure 1.

Parameter Sensitivity & Ablation Study. Understanding the sensitivity of the optimizer to its hyperparameters, particularly momentum (β) and cone angle (θ), provides critical insights into its performance across different phases of optimization. This section explores their roles in convergence acceleration and alignment with the true gradient, highlighting key patterns observed in the experiments.

Early-Phase Convergence and Momentum Alignment. Momentum plays a critical role in accelerating early convergence by aligning the optimizer's updates with the true gradient direction. High momentum values, such as $\beta = 0.98$, help maintain directional consistency in the initial optimization phase, where gradients tend to form a tightly clustered cone, driving rapid progress toward the optimum. This effect is even more pronounced when combined with a small cone angle (θ), as illustrated in early-phase heatmaps, while more balanced configurations yield better results (Figure 2a and Figure 2b). Our analysis of squared cosine similarity between the momentum vector and true gradients (Figure 3) confirms this: high momentum ($\beta = 0.97$) significantly improves alignment, with up to 2x better directional accuracy during the first 2,000 iterations compared to the expected alignment for random directions. However, as training progresses beyond 3,500 iterations, this alignment advantage diminishes, approaching the random baseline, suggesting that high momentum's benefits are primarily confined to the early phases of training. This supports the use of high initial momentum to accelerate early progress, followed by a gradual reduction to improve long-term stability.

Impact of Cone Angle: Exploration vs. Exploitation. The cone angle (θ) balances building an accurate gradient approximation against effectively exploiting it. Larger θ values sample broader directions, increasing cosine similarity with the true gradient but reducing reliance on the estimated gradient for updates. Figure 3b shows how varying θ affects cosine similarity over iterations.

D.2. OPT

We evaluated on four benchmarks: SST-2 (Socher et al., 2013) (binary sentiment classification), BoolQ (Clark et al., 2019) (boolean question answering), SQuAD v1.1 (Rajpurkar et al., 2016) (span-based QA), and DROP (Dua et al., 2019) (discrete reasoning QA). Each task was fine-tuned for 20,000 iterations to capture performance across early, mid, and late training phases. The best found values for the hyperparameters of our optimizer are $\theta = 1.35, \beta = 0.95$ and $\lambda = 10^{-3}$.

All OPT-1.3B experiments were conducted for 20 K iterations using both ConMeZO and MeZO with a fixed learning rate of $\eta = 10^{-7}$. Due to the very high cost of fine-tuning a 1.3 B-parameter model and rerunning multiple random seeds, we did not perform additional learning-rate searches or seed sweeps for OPT. Consequently, every OPT result reported uses a single seed and $\eta = 10^{-7}$ for both optimizers. These findings completely align with our RoBERTa results, demonstrating that the observed performance gains under fixed hyperparameter settings extend across different model scales.

Hyperparameters were set to $\theta = 1.35, \beta = 0.95$ and we used seed 29 for our reported results.

Our implementation builds on the DPZero framework (Zhang et al., 2024a), with all experiments run on a single NVIDIA H100 NVL GPU (˜95 GiB each). We fixed the smoothing parameter to $\lambda = 10^{-3}$ and, instead of sampling uniformly from a hypersphere, drew random direction from $\mathcal{N}(0, I_d)$, which is a valid simplification in high dimensions.

D.3. Licences

Our evaluations are carried out on commonly-used datasets in the literature.

Datasets. GLUE (Wang et al., 2018a) is designed to provide a general-purpose evaluation of language understanding. Those adopted in our work include MNLI (inference, (Williams et al., 2018)), SST-2/5 (sentiment analysis, (Socher et al., 2013)), SNLI (natural language inference) (Bowman et al., 2015), RTE[1] (inference), and TREC (question classification, (Voorhees & Tice, 2000)). These datasets are released under different permissive licenses.

[1] https://paperswithcode.com/dataset/rte

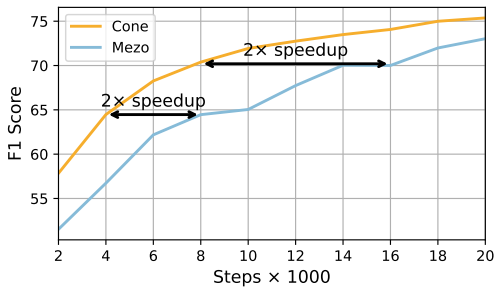


Figure 5: SQuAD F1 score over 20,000 iterations comparing ConMeZO vs. MeZO.

Models. RoBERTa-large. This is a 355M parameter model. The model checkpoint[2] is released under the MIT license.

OPT-1.3B. The model checkpoint[3] is released under a non-commercial license. [4]

[2] https://huggingface.co/FacebookAI/roberta-large
[3] https://huggingface.co/facebook/opt-1.3b
[4] https://github.com/facebookresearch/metaseq/blob/main/projects/OPT/MODEL_LICENSE.md