Does SAM Pay Off? A Cost-Adjusted Analysis of Sharpness-Aware Pretraining

Ivan Maksimov[1,2], **Joqsan Azocar**[1], **Vasily Konovalov**[3,1]
[1]MIRAI [2]SaluteDevices [3]AXXX
maksimov.ivan.v@phystech.edu

Abstract

The high computational cost of pretraining generative language models has motivated interest in optimizers that improve generalization without scaling model size. Sharpness-Aware Minimization (SAM) encourages convergence to flatter minima, a property theoretically linked to better generalization and potentially beneficial for the diverse outputs required in natural language generation. Yet, whether SAM's double compute per step is justified for pretraining generative models remains insufficiently studied. We evaluate SAM during the pretraining of a 100M-parameter LLaMa-2-like model on 150B tokens from SlimPajama against AdamW trained on 300B tokens, approximately compute-matched by design. In this setting, SAM exhibits worse training dynamics, yields approximately 1.8% higher training loss at $\rho = 0.15$, and fails to improve downstream reasoning and question-answering performance. While smaller ρ values (0.15) reduce loss spikes, they do not recover SAM's efficiency trade-off. These findings offer a clear cautionary result for the NLG community: theoretically motivated optimizers do not automatically transfer to autoregressive pretraining.

1 Introduction

The recent proliferation of open-source generative language models has democratized access to powerful natural language generation (NLG) capabilities, enabling applications ranging from dialogue systems and story generation to summarization and machine translation (Rykov et al., 2025). However, large token budgets can make the pretraining of these models computationally expensive, imposing substantial financial and environmental costs (Muennighoff et al., 2023). This resource barrier poses a particular challenge for the NLG community, where developing specialized generative models for low-resource languages, domains, or applications often requires pretraining from scratch or extensive continued pretraining.

Among the techniques proposed to improve generalization without scaling model size, Sharpness-Aware Minimization (SAM) (Foret et al., 2021) has garnered significant attention. SAM modifies the optimization objective to encourage convergence to flatter minima in the loss landscape regions where the loss is insensitive to small perturbations in model weights. This property is theoretically attractive for NLG: flatter minima have been linked to better generalization (Bartlett et al., 2023), and for generative models, robust optimization could lead to more diverse and coherent outputs by avoiding overfitting to narrow patterns in the training data. In vision and fine-tuning settings, SAM has demonstrated notable improvements (Bahri et al., 2022; Zhong et al., 2022), leading to growing interest in applying it to language modeling.

Despite this promise, the applicability of SAM to the pretraining of generative language models under realistic resource constraints remains underexplored. Earlier applications of SAM in NLP focused on fine-tuning (Li et al., 2024), while recent work also investigates language-model pretraining (Watts et al., 2026). Crucially, SAM requires two forward-backward passes per optimization step, approximately doubling its cost. Does SAM's theoretical benefit justify this overhead under limited pretraining budgets?

In this work, we conduct a controlled, cost-adjusted study of SAM in the pretraining of a generative language model. We train a 100M-parameter LLaMa-2-like architecture on 150B tokens from the SlimPajama dataset (Soboleva et al., 2023), comparing SAM against standard AdamW trained on 300B tokens. The comparison is approximately compute-matched by design: SAM runs for half the number of update steps to account for its doubled per-step cost. We evaluate on two standard benchmarks: HellaSwag (Zellers et al., 2019) for commonsense reasoning and BoolQ (Clark et al.,

Proceedings of the AAP-INLG 2026 Workshop (in preparation for publication), pages 42–47
October 18, 2026 ©2026 Association for Computational Linguistics

2019) for question answering. While these tasks are not generative themselves, they serve as robust proxies for the underlying language understanding capabilities that underpin high-quality text generation.

Contrary to expectations, our findings reveal a clear negative result in this setting:

1. SAM with $\rho = 0.15$ yields 1.8% higher training loss and fails to improve next-token prediction accuracy compared to AdamW.

2. Downstream accuracy on HellaSwag plateaus below the AdamW baseline, while BoolQ accuracy degenerates to random guessing level (approximately 50%).

3. Reducing the perturbation radius to $\rho = 0.15$ mitigates loss spikes but does not close the performance gap with AdamW under our approximately compute-matched budget.

2 Related Work

Sharpness-Aware Minimization (SAM) has emerged as a powerful optimization technique for improving model generalization across various deep learning tasks (Foret et al., 2021). The dynamics of SAM have been extensively analyzed by (Bartlett et al., 2023), who demonstrated that for convex quadratic objectives, SAM converges to cycles oscillating in the direction of greatest curvature. In the non-quadratic setting they study, these oscillations can induce drift toward a lower Hessian spectral norm through a third-derivative effect. This characteristic is particularly beneficial for deep neural networks, which often suffer from sharp minima that lead to poor generalization.

While initially demonstrating success in computer vision tasks, SAM has been increasingly adapted for natural language processing applications. Experimental evidence shows that Fisher-masked SAM (FSAM) significantly improves performance on benchmark language understanding tasks such as GLUE and SuperGLUE (Zhong et al., 2022). However, applications to *natural language generation* have received less attention. A few studies have explored SAM for fine-tuning generative models (Li et al., 2024), but these focus on adapting pretrained models rather than pretraining them from scratch, a setting where the double computational cost of SAM is particularly burdensome.

Watts et al. (2026) study SAM pretraining of 20M–150M-parameter language models and mid-training of OLMo-2-1B. In their token-matched pretraining comparisons, SAM achieves similar or higher pretraining loss than AdamW across the model sizes and token budgets studied. We complement these findings with an approximately compute-matched comparison of SAM and AdamW for a 100M-parameter model pretrained on SlimPajama, evaluating next-token prediction, HellaSwag, and BoolQ. Work on data-constrained pretraining (Muennighoff et al., 2023) provides further context for our token budgets and repeated-data training.

3 Sharpness-Aware Minimization

Sharpness-Aware Minimization (SAM) is an optimization technique designed to improve the generalization of deep neural networks by minimizing not only the training loss but also the sharpness of the loss landscape around the model parameters. Sharpness here refers to how sensitive the loss is to small perturbations in the model weights; flatter minima (low sharpness) tend to generalize better.

SAM aims to find parameters $\mathbf{w}$ that minimize the maximum loss within a neighborhood around $\mathbf{w}$, encouraging solutions that are robust to small changes in weights:

$$\min_{\mathbf{w}} \max_{\|\epsilon\| \leq \rho} \mathcal{L}_{D_{\text{tr}}}(\mathbf{w} + \epsilon) \tag{1}$$

where $\mathcal{L}_{D_{\text{tr}}}$ is the training loss for a training dataset D_{tr}, ϵ is a perturbation vector bounded by radius ρ, and $\mathbf{w}$ are the model parameters. This can be interpreted as minimizing both the loss value and the sharpness, defined as:

$$\text{sharpness}(\mathbf{w}) = \max_{\|\epsilon\| \leq \rho} \mathcal{L}_{D_{\text{tr}}}(\mathbf{w} + \epsilon) - \mathcal{L}_{D_{\text{tr}}}(\mathbf{w}) \tag{2}$$

The full SAM objective often includes a regularization term:

$$\mathcal{L}_{\text{SAM}}(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2, \quad \text{where} \tag{3}$$

$$\mathcal{L}_{\text{SAM}}(\mathbf{w}) = \max_{\|\epsilon\| \leq \rho} \mathcal{L}_{D_{\text{tr}}}(\mathbf{w} + \epsilon) \tag{4}$$

Since directly solving the max problem is expensive, SAM approximates the worst-case perturbation $\epsilon(\mathbf{w})$ using the gradient of the loss:

$$\hat{\epsilon}(\mathbf{w}) = \rho \frac{\nabla_{\mathbf{w}} \mathcal{L}_{D_{\text{tr}}}(\mathbf{w})}{\|\nabla_{\mathbf{w}} \mathcal{L}_{D_{\text{tr}}}(\mathbf{w})\|_2} \quad (5)$$

The optimization update at step t becomes:

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla \mathcal{L}_{D_{\text{tr}}} \left(\mathbf{w}^{(t)} + \hat{\epsilon}(\mathbf{w}^{(t)}) \right) \quad (6)$$

where η is the learning rate. This update illustrates SAM with SGD as the base optimizer; our experiments use AdamW (Loshchilov and Hutter, 2019).

This two-step process involves:

- Perturbing the weights in the direction of the gradient scaled by ρ.
- Computing the gradient at the perturbed weights and updating the original weights accordingly.

4 Datasets

As for pre-training we used SlimPajama (Soboleva et al., 2023). SlimPajama is a large, open-source dataset designed for training large language models, created by extensively cleaning and deduplicating the 1.2T token RedPajama dataset (Weber et al., 2024) down to 627B tokens. It combines multiple high-quality sources such as web text, Wikipedia, GitHub, books, and more, and applies both global and local deduplication using advanced methods like MinHashLSH to maximize information density and minimize duplicate or low-quality data.

We tested the pre-trained models on two popular NLP benchmarks: HellaSwag (Zellers et al., 2019) and BoolQ (Clark et al., 2019). HellaSwag is a challenge dataset for evaluating commonsense NLI that is especially hard for state-of-the-art models with a comparable number of parameters, though its questions are trivial for humans. BoolQ is a question answering dataset for yes/no questions containing about 16k examples.

5 Experimental Setup

Due to the computation resource limitation for pre-training we employed an LLaMA-2 architecture (Touvron et al., 2023) with the hyperparameters from Table 1 (Appendix B), which also provides batching, hardware, and resource-budget details.

We used 150B tokens for training with SAM and 300B tokens without SAM, since SAM requires roughly double the computational resources per optimization step (Xie et al., 2024). The number of epochs for SAM training was set to 4, following recommendations (Muennighoff et al., 2023). According to the methodology for assessing the effectiveness of SAM (Foret et al., 2021), the epochs without SAM should be double those with SAM, totaling 8 for approximate compute matching. For the neighborhood radius ρ, we explored two values: 0.15 and 2.0, as they have provided satisfactory fine-tuning results (Bahri et al., 2022; Li et al., 2024). The batch size was about 1 million tokens.

6 Experimental Results

Training Dynamics Figure 1 presents the training loss and next-token prediction accuracy during pretraining. SAM consistently underperforms AdamW across both metrics. At the end of pretraining, SAM with $\rho = 2.0$ achieves a final loss of 2.71, compared with AdamW at 2.58. SAM with $\rho = 0.15$ partially mitigates this degradation, reaching 2.63, or approximately 1.8% higher training loss than the baseline. For next-token accuracy, AdamW reaches 49%, while SAM ($\rho = 2.0$) reaches 46.5%, and SAM ($\rho = 0.15$) reaches 48%.

Downstream Evaluation Figures 2 and 3 (Appendix C) show accuracy on HellaSwag and BoolQ. On HellaSwag, AdamW reaches 31.8%, improving steadily throughout training. SAM ($\rho = 2.0$) reaches 29.4%, while SAM ($\rho = 0.15$) reaches 30.4%, plateauing after approximately 70% of training. On BoolQ, the results are more striking: AdamW achieves approximately 60%, while SAM ($\rho = 2.0$) reaches 51–52%, and SAM ($\rho = 0.15$) reaches approximately 50%, both near the uniform-random-guessing level.

These results are particularly concerning for NLG: BoolQ, which requires nuanced semantic understanding, is precisely where SAM performs worst. An hypothesis is that SAM's flatness penalty may overly constrain representations needed for high-quality generation.

Analysis Why does SAM fail in this setting? We consider three hypotheses not tested by these experiments:

First, the *autoregressive loss landscape* may differ fundamentally from classification tasks. Language modeling optimizes a sequential distribution over tokens, which may already produce relatively flat minima, leaving little room for SAM's sharp-

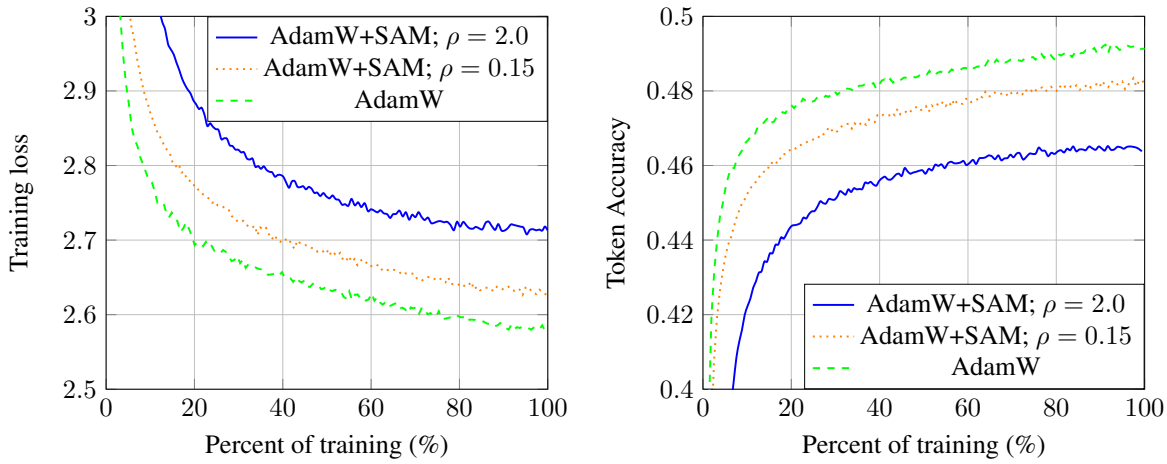

Figure 1: Loss function and token accuracy (smoothed values) for next token prediction during the pre-training.

ness penalty to improve generalization (Li et al., 2024). Forcing additional flatness may overly constrain the model's capacity to capture the diverse patterns needed for generation.

Second, SAM applies a *uniform perturbation radius* ρ across all parameters. In language modeling, different tokens (content words vs. function words) and positions likely have different loss sensitivities. A shared radius does not imply identical perturbations across parameters, but could still over-regularize some while under-regularizing others. Fisher-masked SAM (Zhong et al., 2022) may be more suitable for generation.

Third, the 100M-parameter model and 150B token budget may be *insufficient for SAM to reach wide minima*. While this scale follows established practices for controlled experiments (Allen-Zhu and Li, 2024; Muennighoff et al., 2023), improvements in the base-model metrics evaluated here may require different scales or training budgets.

We tested a higher learning rate ($4 \cdot 10^{-4}$) with $\rho = 0.15$, but this did not improve performance. This limited comparison does not rule out improvements from other hyperparameter choices.

Conclusion

Under the tested configuration (100M parameters, 150B tokens for SAM and 300B for AdamW), approximately compute-matched by design, SAM did not outperform AdamW on training loss, next-token accuracy, HellaSwag, or BoolQ.

For the NLG community, this finding carries three practical implications. First, SAM's benefits in other settings do not necessarily imply improved base-model performance under an approximately compute-matched budget. Second, claims of optimizer improvements for generative models should be validated with generation-specific metrics (e.g., diversity, MAUVE) rather than reasoning benchmarks alone. Third, practitioners should prioritize cost-adjusted comparisons (performance per FLOP or wall-clock hour) when evaluating optimizers with doubled per-step cost.

Limitations

Our study is limited to one 100M-parameter architecture, token budgets of 150B for SAM and 300B for AdamW, two perturbation radii, limited learning-rate exploration, and no reported repeated runs; evaluation covers two understanding benchmarks rather than direct generation tasks. Whether larger models or longer training would improve SAM's relative performance on our evaluated metrics remains an open question. We encourage future work to evaluate SAM on generation-specific tasks, explore variants such as Fisher-masked SAM (Zhong et al., 2022), and extend our cost-adjusted methodology to larger scales.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of Language Models: Part 1, Learning Hierarchical Language Structures. *Preprint*, arXiv:2305.13673.

Dara Bahri, Hossein Mobahi, and Yi Tay. 2022. Sharpness-aware Minimization Improves Language Model Generalization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7360–7371. Association for Computational Linguistics.

Peter L. Bartlett, Philip M. Long, and Olivier Bousquet. 2023. The dynamics of sharpness-aware minimization: Bouncing across ravines and drifting towards wide minima. *J. Mach. Learn. Res.*, 24:316:1–316:36.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2924–2936. Association for Computational Linguistics.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting Catastrophic Forgetting in Large Language Model Tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 4297–4308. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A. Raffel. 2023. Scaling Data-constrained Language Models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Elisei Rykov, Kseniia Petrushina, Maksim Savkin, Valerii Olisov, Artem Vazhentsev, Kseniia Titova, Alexander Panchenko, Vasily Konovalov, and Julia Belikova. 2025. When models lie, we learn: Multilingual span-level hallucination detection with PsiloQA. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11663–11682, Suzhou, China. Association for Computational Linguistics.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288v2.

Ishaan Watts, Catherine Li, Sachin Goyal, Jacob Mitchell Springer, and Aditi Raghunathan. 2026. Sharpness-aware pretraining mitigates catastrophic forgetting. *Preprint*, arXiv:2605.02105v1.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Wanyun Xie, Thomas Pethick, and Volkan Cevher. 2024. Sampa: Sharpness-aware minimization parallelized. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4791–4800. Association for Computational Linguistics.

Qihuang Zhong, Liang Ding, Li Shen, Peng Mi, Juhua Liu, Bo Du, and Dacheng Tao. 2022. Improving sharpness-aware minimization with fisher mask for better generalization on language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4064–4085, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Sharpness-Aware Minimization Algorithm

The following algorithm illustrates SAM with SGD as the base optimizer; the pre-training experiments use AdamW:

Algorithm 1 SAM with SGD as the base optimizer (illustrative).

Input: Training set $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{n}$, loss function ℓ,
batch size b, step size $\eta > 0$, neighborhood size $\rho > 0$
Output: Model trained with SAM

1: Initialize weights $\mathbf{w}_0$, $t = 0$
2: **while** not converged **do**
3: Sample batch $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{b}$
4: Compute gradient $\nabla_{\mathbf{w}} \mathcal{L}_{\mathcal{B}}(\mathbf{w})$ for the batch
5: Compute $\hat{\epsilon}$ per equation 5
6: Compute gradient for SAM: $\mathbf{g} = \nabla_{\mathbf{w}} \mathcal{L}_{\mathcal{B}}(\mathbf{w})\big|_{\mathbf{w}+\hat{\epsilon}}$
7: Update weights: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{g}$
8: $t = t + 1$
9: **end while**

B Pre-training Hyperparameters

Hyperparameter	Value
Number of parameters	100M
Hidden size	600
Intermediate vector size	1800
Tokenizer	TinyLlama-1.1B
Optimizer	AdamW
Learning rate scheduler	Cosine
Learning Rate	$2 \cdot 10^{-4}$
Higher-LR trial ($\rho = 0.15$)	$4 \cdot 10^{-4}$
Weight decay value	$3 \cdot 10^{-5}$
Batch size	1M tokens
ρ	0.15 and 2.0

Table 1: LLaMA-2 pre-training hyperparameters.

The experiments used a sequence length of 2,048 tokens, a global batch of 512 sequences, micro-batch size 16, and `NO_SHARD`, with four A100-SXM4-40GB GPUs.

C Performance on BoolQ and HellaSwag benchmarks

Accuracy on BoolQ (Figure 2) and HellaSwag (Figure 3) benchmarks while pre-training. During training, the HellaSwag accuracy with SAM does not exceed that without SAM. For BoolQ, SAM accuracy remains near the random-guessing level (approximately 50%), whereas training without SAM plateaus at approximately 60% accuracy after 40% of training.

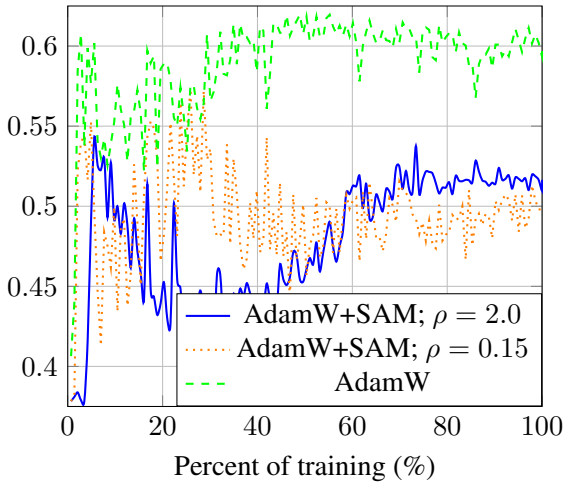

Figure 2: Accuracy on BoolQ.

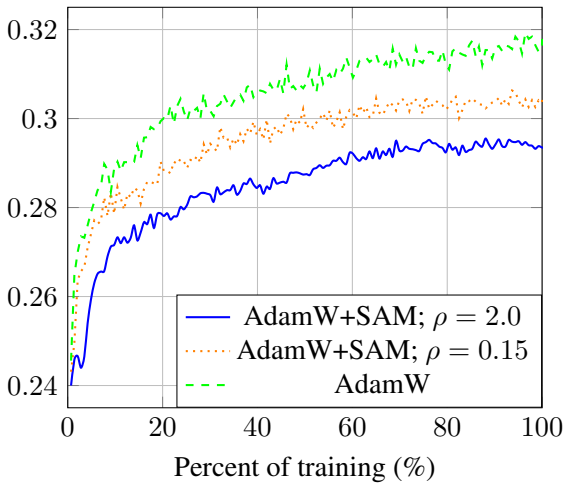

Figure 3: Accuracy on HellaSwag.