
ESSAM: A Novel Competitive Evolution Strategies Approach to Reinforcement Learning for Memory Efficient LLMs Fine-Tuning

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Reinforcement learning (RL) has become a key training step for improving math-
2 ematical reasoning in large language models (LLMs), but it often has high GPU
3 memory usage, which makes it hard to use in settings with limited resources.
4 To reduce these issues, we propose Evolution Strategies with Sharpness-Aware
5 Maximization (ESSAM), a full parameter fine-tuning framework that tightly com-
6 bines the zero-order search in parameter space from Evolution Strategies (ES)
7 with the Sharpness-Aware Maximization (SAM) to improve generalization. We
8 conduct fine-tuning experiments on the mainstream mathematica reasoning task
9 GSM8K. The results show that ESSAM achieves an average accuracy of 78.27%
10 across all models and its overall performance is comparable to RL methods. It
11 surpasses classic RL algorithm PPO with an accuracy of 77.72% and is comparable
12 to GRPO with an accuracy of 78.34%, and even surpassing them on some models.
13 Further generalization experiments show that the models trained with ESSAM
14 exhibit stronger generalization ability. Their average performance achieves the
15 best results on 5 out of 6 datasets, indicating that ESSAM can effectively improve
16 the generalization performance of fine-tuned models. In terms of GPU memory
17 usage, ESSAM reduces the average GPU memory usage by 18 \times compared to PPO
18 and by 10 \times compared to GRPO, achieving an extremely low GPU memory usage.
19 In addition, we design an accelerated variant of ESSAM, which achieves nearly
20 a twofold speedup while maintaining the same GPU memory usage as ESSAM,
21 and attains an average accuracy of 78.02% across all models, outperforming PPO.
22 Code:<https://anonymous.4open.science/r/ESSAM-3F4F/>

23 1 Introduction

24 Recent breakthroughs in the mathematical reasoning ability of large language models (LLMs) have
25 led to their deep integration with many scientific fields such as mathematics and physics, making
26 them an indispensable part of modern life [7, 18, 11, 3]. These gains in mathematical reasoning are
27 largely driven by the rise of Reinforcement learning (RL) as a fine-tuning method [14, 15, 20, 21].
28 However, despite its effectiveness, applying RL to fine-tune LLMs requires extremely high GPU
29 memory resources [4], such as fine-tuning an 8B model with PPO, 314.44 GiB of GPU memory is
30 required (Appendix G), which is not friendly to resource constrained open-source communities, and
31 limits the further development of using RL to fine-tune LLMs.

32 To mitigate the issue that RL requires extremely high GPU memory resources, [13] explore using
33 Evolution Strategies (ES) to fine-tune LLMs. ES is a population based zero-order method, which
34 does not rely on gradients and updates model parameters using only forward generation and reward
35 evaluation. Therefore, ES can significantly reduce GPU memory usage and training cost. However,

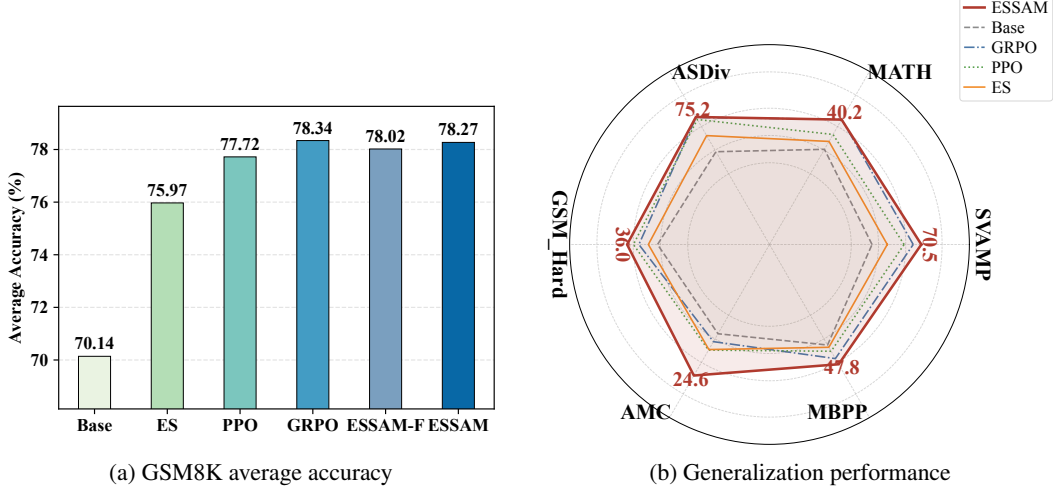


Figure 2: (a) The average accuracy of each algorithm across all models on the GSM8K task (%). (b) The average performance of each algorithm across datasets over all models.

36 on mainstream mathematical reasoning tasks such as GSM8K [2], the overall performance of ES
 37 is often limited and difficult to match the performance of RL methods. The reason lies in the fact
 38 that ES only explores within the model’s parameter space and then directly updates the parameters.
 39 This update approach tends to drive the model toward sharp minima, resulting in poor generalization
 40 performance. In addition, in their ES experiments, [13] use only 200 examples for training and 2,000
 41 examples for evaluation. This split is not a standard setup, and during training the 200 examples
 42 are used as a whole without the common practice of shuffling the training data and iterating with
 43 small mini batches. This can introduce more randomness and differs from standard LLMs fine-tuning
 44 pipelines. So the practical performance of ES still needs further study.

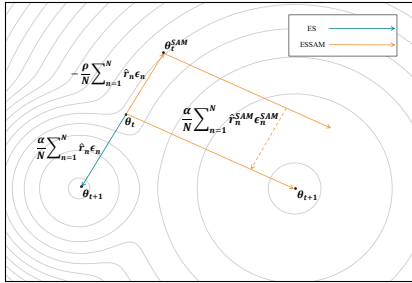


Figure 1: An illustration of the ESSAM parameter update.

54
 55
 56
 57
 58
 59 perform a reverse update, obtains a new update direction at the perturbed parameters, and then uses
 60 this new direction to update the original parameters, so that the solution of the model is guided to
 61 converge to flatter regions, enhancing the model’s robustness and generalization ability. In addition,
 62 we further propose an accelerated variant of ESSAM. This accelerated algorithm adopts a SAM
 63 mechanism based on Frobenius norm normalization during the reverse update and uses a smaller
 64 SAM population to improve training efficiency while maintaining low GPU memory usage.

65 We follow a more standard training procedure. We use a standard split between training data and
 66 evaluation data, shuffle the training data, and perform multi step updates with small mini batches,
 67 rather than training once on a small set of samples as a whole. This reduces randomness from small
 68 sample training and better averages the noise induced variation across iterations, and the training
 69 signal is smoother and the optimization process is more stable.

70 Experimental results show that ESSAM extends ES to the mathematical reasoning task GSM8K and
 71 significantly outperforms standard ES while maintaining low GPU memory usage. As shown in
 72 Figure 2a and Table 1, ESSAM achieves an average accuracy of 78.27% across all models, surpassing
 73 PPO with 77.72% and remaining close to GRPO with 78.34%, and it even outperforms RL methods

74 on some models. In addition, we evaluate the fine-tuned models on five more challenging out-of-
75 distribution mathematical reasoning benchmarks and one verifiable code benchmark. The models
76 fine-tuned with ESSAM achieve the best average performance on 5 out of these 6 benchmarks
77 (Figure 2b), demonstrating stronger generalization ability. We also evaluate the accelerated variant
78 of ESSAM, namely ESSAM-F. As shown in Table 2, ESSAM-F achieves nearly a twofold speedup
79 while maintaining the same GPU memory usage as ESSAM, and still attains an average accuracy of
80 78.02% across all models, outperforming PPO overall. We summarize our contributions as follows:

- 81 • We propose a new zeroth-order full-parameter fine-tuning framework, ESSAM. It combines
82 ES with the Sharpness-Aware Maximization mechanism, introduces neighborhood probing
83 and a two-stage evaluation update in parameter space, and guides optimization toward flatter
84 and more robust solutions.
- 85 • We extend zeroth-order full-parameter fine-tuning to the mainstream mathematical reasoning
86 task GSM8K for the first time. We also adopt a more standard data usage and training
87 procedure, including a standard split of training and evaluation sets, shuffling the training
88 data, and performing multi-step updates with small mini-batches, which improves training
89 stability and reproducibility.
- 90 • Experimental results show that ESSAM achieves performance comparable to mainstream RL
91 methods on GSM8K and significantly outperforms standard ES. Further out-of-distribution
92 generalization experiments show that ESSAM effectively improves the generalization ability
93 of fine-tuned models.
- 94 • We systematically evaluate GPU memory usage. As shown in Figure 4, ESSAM maintains
95 the same inference-level GPU memory usage as ES, and on average reduces GPU memory
96 usage by $18\times$ compared with PPO and by $10\times$ compared with GRPO, greatly reducing the
97 demand for computational resources.
- 98 • We further propose an accelerated variant of ESSAM, namely ESSAM-F. In the first-stage
99 reverse update, it adopts a SAM mechanism based on Frobenius norm normalization and
100 uses a smaller SAM population, achieving nearly a twofold speedup while maintaining the
101 same GPU memory usage.

102 2 Related Work

103 **Evolution Strategies.** Evolution Strategies (ES) are population-based zeroth-order optimization
104 methods that update parameters using fitness scores from randomly perturbed candidates. Classic
105 methods such as CMA-ES [8] and Natural Evolution Strategies [17] laid the foundation for applying
106 evolutionary search to neural network optimization. In LLM settings, early studies mainly focused
107 on small-scale or parameter-efficient subspaces, such as fine-tuning the last layer of mT5 [16] or
108 optimizing LoRA parameters [9]. Recent work [13] scales ES to full-parameter fine-tuning of
109 billion-parameter LLMs through engineering and parallelization advances, but its generalization on
110 mainstream mathematical reasoning tasks remains limited. Therefore, we introduce a sharpness-aware
111 mechanism into ES to improve its generalization performance.

112 **Reinforcement learning.** RL fine-tuning has become an important approach for improving the
113 performance of LLMs [1, 23, 12], especially their mathematical reasoning ability [7]. PPO [14] and
114 GRPO [15] are two widely used RL methods for LLM training. PPO stabilizes policy updates with a
115 clipped surrogate objective and usually relies on value function estimation, while GRPO improves
116 efficiency by removing the value model and using group-based rewards as a baseline. Although these
117 methods can effectively improve LLM performance, they still require substantial GPU memory due
118 to backpropagation, policy optimization, and additional training states.

119 **Sharpness-Aware Minimization (Maximization).** SAM [5] improves generalization by guiding
120 models toward flatter regions. In language model settings, prior work shows that sharpness-aware
121 training can reduce overfitting and improve downstream performance. Several studies further improve
122 SAM from the perspectives of stability and efficiency, such as selecting perturbed parameters based on
123 Fisher information [22] or reducing the extra training cost by modifying the perturbation and update
124 procedure [10]. When gradients are unavailable or only forward passes can be used, sharpness-aware
125 ideas have also been extended to black-box and zeroth-order optimization [19, 6]. Inspired by these
126 works, we combine SAM with ES and propose a zeroth-order sharpness-aware method for improving
127 the mathematical reasoning ability of LLMs.

Algorithm 1 Evolution Strategies with Sharpness-Aware Maximization (ESSAM)

Input: initial parameters θ_0 , reward function R , population size N , noise scale σ , hyperparameter ρ , learning rate α , iterations T , number of parallel processes P

Output: fine-tuned parameters θ_T

Create P processes, each instantiates a model with the same initial parameters θ_0 , with one process as the main process

for $t = 0$ **to** $T - 1$ **do**

(Stage 1) Build a SAM neighborhood point θ_t^{SAM}

 Sample N random seeds s_1, \dots, s_N

$\{r_n\}_{n=1}^N \leftarrow \text{SRE}(\theta_t, R, \sigma, \{s_n\}_{n=1}^N, N, P)$

 Compute mean μ_r , std s_r

 Normalize $\hat{r}_n = \frac{r_n - \mu_r}{s_r + \delta}$

$\theta_t^{\text{SAM}} \leftarrow \text{DIPU}(\theta_t, \{s_n\}_{n=1}^N, \{\hat{r}_n\}_{n=1}^N, -\rho, N)$

 Broadcast parameters of all processes to θ_t^{SAM}

(Stage 2) Evaluate around θ_t^{SAM} **and update** θ_t

 Sample N random seeds $s_1^{\text{SAM}}, \dots, s_N^{\text{SAM}}$

$\{r_n^{\text{SAM}}\}_{n=1}^N \leftarrow \text{SRE}(\theta_t^{\text{SAM}}, R, \sigma, \{s_n^{\text{SAM}}\}_{n=1}^N, N, P)$

 Compute mean μ_r^{SAM} , std s_r^{SAM}

 Normalize $\hat{r}_n^{\text{SAM}} = \frac{r_n^{\text{SAM}} - \mu_r^{\text{SAM}}}{s_r^{\text{SAM}} + \delta}$

$\theta_t \leftarrow \text{DIPU}(\theta_t^{\text{SAM}}, \{s_n\}_{n=1}^N, \{\hat{r}_n\}_{n=1}^N, \rho, N)$

$\theta_{t+1} \leftarrow \text{DIPU}(\theta_t^{\text{SAM}}, \{s_n^{\text{SAM}}\}_{n=1}^N, \{\hat{r}_n^{\text{SAM}}\}_{n=1}^N, \alpha, N)$

 Broadcast parameters of all processes to θ_{t+1}

end for

128 3 Method

129 In this section, we detailed the ESSAM algorithm flow (Sec. 3.1), and present the efficient memory-
130 saving mechanism used in the algorithm (Sec. 3.2). Finally, we explain why our proposed algorithm
131 can be viewed as performing Sharpness-Aware Maximization (Sec. 3.3).

132 3.1 Evolution Strategies with Sharpness-Aware Maximization

133 ES generates a population of perturbation models with Gaussian noise and updates parameters in a
134 reward-weighted manner, exploring and updating the parameter space only once in each iteration
135 (Appendix A). Different from standard ES, our ESSAM method introduces the SAM mechanism.
136 It first uses the current reward weighted aggregation to move the model parameters in the opposite
137 direction of reward increase to a nearby neighborhood point, and then uses the perturbations and
138 rewards computed at this point to update the parameters. This process helps avoid sharp solutions
139 and drives the model toward flatter and more robust regions, thereby improving generalization.

140 Specifically, given a pretrained LLM with parameters θ_0 and a reward function $R(\theta)$, we aim
141 to maximize the expected reward by iteratively updating model parameters. At iteration t , we
142 sample N i.i.d. Gaussian noises $\epsilon_n \sim \mathcal{N}(0, I)$ for $n = 1, \dots, N$, and construct perturbed models
143 $\theta_t^{(n)} = \theta_t + \sigma \epsilon_n$. We evaluate each perturbed model to obtain reward scores $r_n = R(\theta_t^{(n)})$ for
144 $n = 1, \dots, N$. We then apply z-score normalization [13] to keep the reward scale consistent:

$$\mu_r = \frac{1}{N} \sum_{n=1}^N r_n, \quad s_r = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (r_n - \mu_r)^2}, \quad \hat{r}_n = \frac{r_n - \mu_r}{s_r + \delta}, \quad n = 1, \dots, N. \quad (1)$$

145 where δ is a small constant for numerical stability.

146 Next, different from standard ES, which directly updates parameters using \hat{r}_n , we use \hat{r}_n to compute
147 a weighted aggregation of the corresponding noise directions. This forms a combined perturbation
148 for neighborhood probing, and we control its magnitude using the hyperparameter ρ , which perturbs
149 the parameters from θ_t to a neighborhood point θ_t^{SAM} . The perturbation is given by:

$$\theta_{t+1}^{\text{SAM}} = \theta_t - \rho \cdot \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \hat{r}_n \epsilon_n. \quad (2)$$

150 Above operations are shown in the first stage of ESSAM shown in Algorithm 1.

151 After obtaining $\theta_{t+1}^{\text{SAM}}$, we sample another N i.i.d. Gaussian noises $\epsilon_n^{\text{SAM}} \sim \mathcal{N}(0, I)$ at $\theta_{t+1}^{\text{SAM}}$ and
 152 repeat the same steps of perturbation, evaluation, and z-score normalization, producing normalized
 153 rewards \hat{r}_n^{SAM} at $\theta_{t+1}^{\text{SAM}}$. Finally, we restore the parameters back to the original θ_t , and use \hat{r}_n^{SAM}
 154 together with their corresponding noise to form a weighted aggregation for the final update with
 155 learning rate α , where we have absorbed $1/\sigma$ into α :

$$\theta_{t+1} = \theta_t + \alpha \cdot \frac{1}{N} \sum_{n=1}^N \hat{r}_n^{\text{SAM}} \epsilon_n^{\text{SAM}}. \quad (3)$$

156 Above operations are shown in the second stage of ESSAM shown in Algorithm 1.

157 In addition, we propose an accelerated variant of ESSAM, called ESSAM-F (Algorithm 3). The
 158 difference between ESSAM-F and ESSAM lies in how the SAM neighborhood point is constructed
 159 in the first stage. ESSAM-F introduces a SAM mechanism based on Frobenius norm normalization.
 160 It first uses a SAM population N_{SAM} , which is much smaller than N , and applies a centered reward
 161 weighted aggregation of the sampled noise directions to move the model parameters to a nearby
 162 neighborhood point. At this point, the perturbation magnitude is normalized by the Frobenius norm
 163 of the aggregated direction. It then uses the perturbations and rewards computed at this neighborhood
 164 point to update the original parameters. The Appendix B presents the detailed algorithmic procedure
 165 of ESSAM-F.

166 3.2 Memory Saving Mechanism

167 To make ESSAM practical for fine-tuning LLMs, we adopt the memory saving techniques proposed
 168 by [13] to reduce GPU memory usage during training. The detailed implementation is shown in
 169 Algorithms 4, 5 and 6 in Appendix C. Both stages of ESSAM and its accelerated variant use these
 170 memory saving techniques. Together with the zero-order update that relies on forward generation and
 171 reward evaluation rather than backpropagation, their greatly reduce GPU memory usage and avoid
 172 the high memory cost seen in RL methods.

173 Algorithm 1 and Algorithm 3 shows the detailed procedure of ESSAM and ESSAM-F.

174 3.3 Understanding the algorithm design of ESSAM

175 In the SAM setting, when constructing the SAM neighborhood point in the first stage, its structural
 176 form is as follows [5]: $\theta_{t+1}^{\text{SAM}} = \theta_t - \rho \cdot \frac{g}{\|g\|}$, Where g is a gradient direction.

177 In our setting, the gradient estimate is computed as:

$$g_t = \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N (r_n - \mu_r) \epsilon_n, \quad (4)$$

178 The estimated variance s_r^2 is computed as:

$$s_r^2 = \frac{1}{N-1} \sum_{n=1}^N (r_n - \mu_r)^2, \quad (5)$$

179 In our algorithm, the SAM neighborhood point is constructed as follows: $\theta_{t+1}^{\text{SAM}} = \theta_t - \rho \cdot \frac{g_t}{s_r}$.

180 Proposition 1 shows why the first stage update of ESSAM is essentially performing SAM.

181 **Proposition 1.** *Let the stochastic gradient estimation g_t defined in Eq. (4) and the variance s_r^2 defined
 182 in Eq. (5). Then it holds that*

Table 1: Accuracy of different algorithms on different models and average accuracy for the GSM8K task (%). Best results are in bold, and second-best results are underlined.

METHOD	QWEN2.5-INSTRUCT				LLAMA-INSTRUCT			AVERAGE
	0.5B	1.5B	3B	7B	3.2-1B	3.2-3B	3.1-8B	
ORIGINAL	46.47	73.77	84.84	90.52	39.04	74.60	81.73	70.14
PPO	53.90	<u>78.85</u>	87.19	91.28	61.49	83.02	88.29	77.72
GRPO	54.24	<u>78.39</u>	<u>87.34</u>	93.02	62.32	84.00	89.08	78.34
ES	52.46	75.97	86.66	90.98	57.24	80.97	87.54	75.97
ESSAM(OURS)	54.06	78.92	87.71	<u>92.57</u>	<u>61.79</u>	<u>83.93</u>	<u>88.93</u>	<u>78.27</u>
ESSAM-F(OURS)	<u>54.06</u>	78.70	87.19	<u>92.27</u>	61.41	83.85	88.63	78.02

$$\mathbb{E}[g_t] = \frac{N-1}{N} \nabla R(\theta) + \tau, \quad \tau = O(\sigma) \quad (6)$$

$$\mathbb{E}[\|g_t\|^2] = \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \cdot \|\nabla R(\theta)\|^2 + \gamma, \quad \gamma = O(\sigma). \quad (7)$$

$$\mathbb{E}[s_r^2] = \sigma^2 \|\nabla R(\theta)\|^2 + \zeta, \quad \zeta = O(\sigma^3). \quad (8)$$

$$\mathbb{E}[\|g_t\|^2] \approx \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \cdot \sigma^{-2} \cdot \mathbb{E}[s_r^2].$$

183 where d denotes the number of model parameters.

184 Since $\frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \cdot \sigma^{-2}$ is a constant, $\frac{g_t}{s_r}$ and $\frac{g_t}{\|g_t\|}$ can be regarded as equivalent, scaled
 185 only by a constant factor. Therefore, the first stage update of our proposed algorithm is essentially
 186 performing Sharpness-Aware Maximization. Detailed proof of Proposition 1 is in Appendix D.

187 4 Experiment

188 We conduct experiments on a mainstream mathematical reasoning task GSM8K. Section 4.1 presents
 189 the detailed results and the generalization results. Section 4.2 provides an analysis of GPU memory
 190 usage and computational. Section 4.3 presents the ablation study and analyzes the sensitivity to
 191 hyperparameters. Section 4.4 experimentally evaluates the performance, runtime, and GPU memory
 192 usage of ESSAM-F. The Appendix E presents the detailed experimental setup.

193 4.1 Main Results

194 **On the mainstream mathematical reasoning task GSM8K, ESSAM significantly improves**
 195 **the performance of all tested models and achieves performance on par with mainstream RL**
 196 **methods.** As shown in Table 1, we systematically compare the fine-tuning performance of ESSAM
 197 with ES, PPO, and GRPO across different model families and model sizes. Overall, ESSAM
 198 brings stable improvements on all tested models and significantly outperforms standard ES. The
 199 average accuracy of ESSAM reaches 78.27%, showing a clear improvement over the original models
 200 with 70.14% and standard ES with 75.97%. This shows that, after introducing the sharpness-
 201 aware mechanism, ESSAM can effectively alleviate the performance limitation of standard ES on
 202 mathematical reasoning tasks and improve the overall effectiveness of zeroth-order fine-tuning.

203 Compared with mainstream RL methods, ESSAM also shows strong competitiveness. Its average
 204 accuracy surpasses PPO with 77.72% and remains close to GRPO with 78.34%. More specifically, on
 205 the Qwen2.5 family, ESSAM achieves the best results on the 1.5B and 3B models, reaching 78.92%
 206 and 87.71%, respectively. On the 0.5B and 7B models, ESSAM also outperforms PPO and remains
 207 close to GRPO. On the LLaMA family, ESSAM brings stable improvements on the 1B, 3B, and 8B
 208 models, and its results are all higher than PPO and close to GRPO.

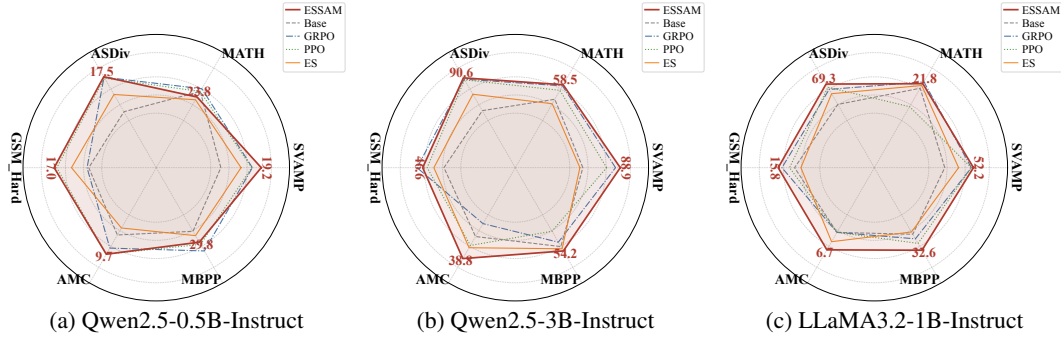


Figure 3: Performance of models fine-tuned with different algorithms on out-of-distribution benchmarks. The generalization results for the other models are presented in Appendix F.

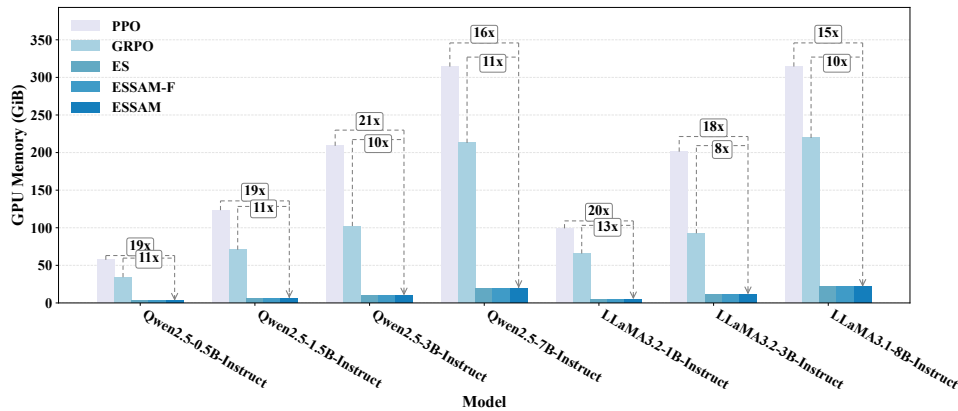


Figure 4: GPU memory usage when fine-tuning different LLMs with different algorithms. More details can be found in Appendix G.

209 **ESSAM can improve the generalization ability of the model.** We further evaluate the fine-tuned
 210 models on five more challenging out-of-distribution mathematical reasoning benchmarks and one
 211 verifiable code generation benchmark. The results show that ESSAM exhibits stronger generalization
 212 ability on these downstream tasks. After averaging the results over all models, ESSAM achieves the
 213 best average performance on 5 out of the 6 benchmarks, and its overall average performance across
 214 all models and all datasets is also higher than ES, PPO, and GRPO (Figure 2b). More detailed results
 215 are shown in Figure 3, on Qwen2.5-0.5B-Instruct, ESSAM performs best on SVAMP, ASDiv, and
 216 GSM-Hard, and is tied for the best result on AMC. On Qwen2.5-3B-Instruct, ESSAM achieves the
 217 best results on 5 out of the 6 benchmarks. On LLaMA3.2-1B-Instruct, ESSAM also achieves the best
 218 results on 5 out of the 6 benchmarks. These results show that the advantage of ESSAM is not limited
 219 to the training task GSM8K, but can generalize well to more challenging out-of-distribution tasks,
 220 demonstrating stronger cross-task transfer ability and generalization ability.

221 4.2 GPU Memory Usage and Time Efficiency Analysis

222 **ESSAM can fine-tune LLMs using only inference-level GPU memory, greatly reducing GPU**
 223 **memory usage.** We measure and compare the GPU memory usage of base models with different
 224 sizes and architectures during training. The results in Figure 4 show a highly consistent and clear
 225 trend. Across all models, ESSAM maintains the same extremely low, inference-level GPU memory
 226 usage as standard ES. This indicates that, with the help of memory-saving techniques, introducing the
 227 SAM mechanism, neighborhood probing, and the two-stage evaluation update into the ES framework
 228 does not introduce additional GPU memory overhead. In contrast, the GPU memory usage of PPO
 229 and GRPO increases rapidly as the model size grows, often reaching hundreds of GiB on larger
 230 models. This is related to their reliance on backpropagation and policy updates, as well as the need to
 231 store value networks and additional optimizer states. Compared with PPO, ESSAM achieves about
 232 $15\times$ – $21\times$ GPU memory savings across different models; compared with GRPO, it achieves about
 233 $8\times$ – $13\times$ savings. On average, ESSAM reduces GPU memory usage by $18\times$ compared with PPO and

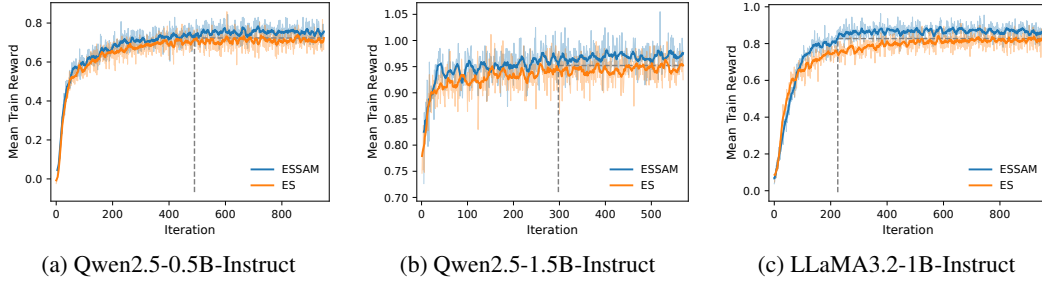


Figure 5: The training mean reward curves of ESSAM and ES. These curves show that ESSAM has better training trend and converges earlier than ES, leading to better computational efficiency. More results are presented in the Appendix H.

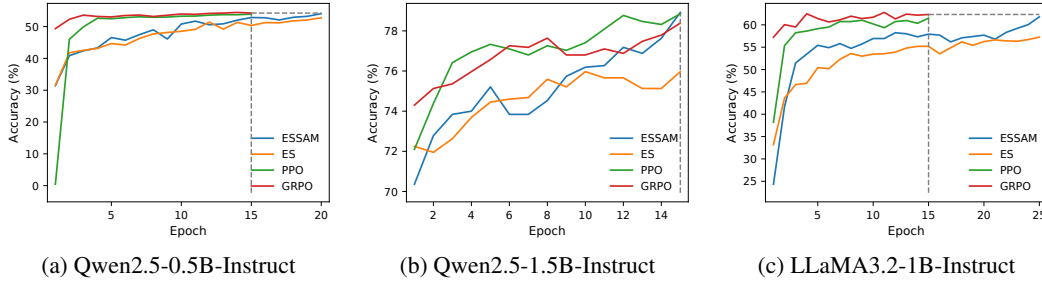


Figure 6: A schematic illustration of how the test set accuracy of models fine-tuned with different algorithms evolves over training. More results are shown in the Appendix I.

234 by $10\times$ compared with GRPO. This substantially reduces the demand for computational resources
 235 while maintaining inference-level GPU memory usage, making stable LLM reasoning fine-tuning
 236 feasible in resource constrained settings.

237 **ESSAM has good time efficiency and shows a clear advantage in iteration efficiency compared**
 238 **with standard ES.** As shown in Figure 5, ESSAM converges faster than ES and reaches, at an early
 239 stage of training, the reward level that ES only approaches near the end of training. This indicates
 240 that, under the same target reward, ESSAM needs fewer iterations to achieve a similar training result.
 241 Figure 6 further shows how model accuracy changes with training progress. From the accuracy
 242 perspective, ESSAM can achieve fine-tuning performance comparable to RL methods with no more
 243 than twice the training progress, or even with the same training progress, while ES struggles to reach
 244 this level under the same training progress as ESSAM. Although ESSAM introduces an additional
 245 round of neighborhood probing and reward evaluation in each iteration, which slightly increases the
 246 per-iteration cost, its faster convergence leads to better overall training efficiency. In other words,
 247 the extra cost in each iteration is partly offset by the reduced number of iterations needed to reach a
 248 strong training result. This makes ESSAM more efficient than standard ES when the goal is to reach
 249 a target reward or target accuracy. This also suggests that the SAM mechanism helps the optimization
 250 reach flatter and more robust regions faster, thereby improving time efficiency in practice.

251 Table 2 reports the runtime of ES-
 252 SAM and GRPO when fine-tuning dif-
 253 ferent LLMs. In Table 2, although
 254 ESSAM takes longer than GRPO on
 255 smaller models, the increase is not
 256 large. Except for Qwen2.5-0.5B-
 257 Instruct, where ESSAM takes about
 258 3.5 times the runtime of GRPO, the
 259 runtime of ESSAM on other models is
 260 approximately 2 times that of GRPO.
 261 This extra runtime is acceptable be-
 262 cause ESSAM can achieve GRPO-
 263 level fine-tuning performance while
 264 using only inference-level GPU memory.

Table 2: Training time in hours when fine-tuning different LLMs with different algorithms.

BASE MODEL	GRPO	ESSAM	ESSAM-F
QWEN-2.5-0.5B-IT	13.5	45	27
QWEN-2.5-1.5B-IT	26.5	34	20
QWEN-2.5-3B-IT	48	91	41
QWEN-2.5-7B-IT	106	107	67
LLAMA-3.2-1B-IT	19	33	20
LLAMA-3.2-3B-IT	49	116	52
LLAMA-3.1-8B-IT	105	233	113

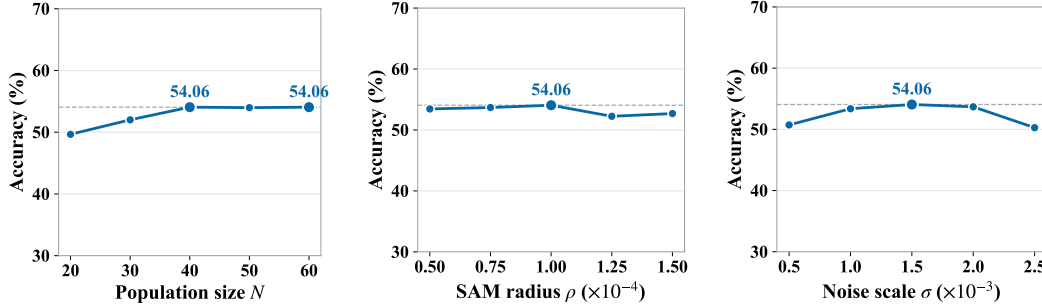


Figure 7: Ablation study of the population size N , noise scale σ , and SAM radius ρ on Qwen2.5-0.5B-Instruct. The results show that ESSAM is sensitive to these hyperparameters but remains relatively stable within a reasonable range.

265 4.3 Ablation Study

266 We conduct an ablation study on Qwen2.5-0.5B-Instruct to investigate the sensitivity of ESSAM
 267 to different hyperparameters. The experimental results are shown in Figure 7. Overall, ESSAM
 268 is sensitive to the choices of the population size N , noise scale σ , and SAM radius ρ , but remains
 269 relatively stable within a reasonable range. For N , performance improves as the population size
 270 increases from 20 to 40, but the gain becomes saturated when N is further increased to 50 and 60,
 271 suggesting that a moderate population size can provide effective search directions while balancing
 272 performance and efficiency. For σ , overly small values lead to insufficient exploration, while overly
 273 large values introduce excessive perturbations, and both degrade performance. For ρ , ESSAM is
 274 relatively stable when ρ is small, but performance starts to fluctuate and drop when ρ exceeds a certain
 275 value. The best performance is achieved at $\rho = 1.0 \times 10^{-4}$, indicating that a proper neighborhood
 276 perturbation strength helps guide the model toward flatter and more robust regions.

277 4.4 ESSAM-F

278 We evaluate the accelerated variant of ESSAM. As shown in Table 1, although ESSAM-F suffers
 279 a slight performance drop compared with the original ESSAM, its overall performance remains
 280 highly competitive. In terms of average accuracy across all models, ESSAM-F achieves 78.02%,
 281 which is only 0.25 percentage points lower than ESSAM’s 78.27%, while still outperforming PPO’s
 282 77.72% and remaining close to GRPO’s 78.34%. Looking at the results on individual models,
 283 ESSAM-F matches ESSAM on Qwen2.5-0.5B-Instruct, matches PPO on Qwen2.5-3B-Instruct, and
 284 still outperforms PPO on Qwen2.5-7B-Instruct, LLaMA3.2-3B-Instruct, and LLaMA3.1-8B-Instruct,
 285 indicating that it can preserve performance well while reducing computational cost. At the same time,
 286 ESSAM-F has a clear advantage in runtime efficiency. Since it uses a smaller SAM population in the
 287 reverse update of the first stage, it achieves nearly a twofold speedup (Table 2) while maintaining the
 288 same GPU memory usage as ESSAM (Figure 4). These results show that ESSAM-F further improves
 289 the practical training efficiency of the method with almost no sacrifice in performance, and achieves a
 290 better balance among performance, runtime, and GPU memory usage.

291 5 Conclusion

292 In this paper, we propose ESSAM, a zeroth-order full-parameter fine-tuning method that combines
 293 Evolution Strategies with Sharpness-Aware Maximization to improve the mathematical reasoning
 294 ability of LLMs under limited computational resources. ESSAM relies only on forward generation and
 295 rule-based reward evaluation, maintaining inference-level GPU memory usage that is much lower than
 296 PPO and GRPO. Experiments show that ESSAM consistently improves all tested models, outperforms
 297 standard ES, and achieves performance comparable to PPO and GRPO. Further generalization
 298 experiments show that ESSAM fine-tuned models perform strongly on out-of-distribution benchmarks.
 299 We also propose an accelerated variant, ESSAM-F, which achieves nearly a twofold speedup with
 300 the same GPU memory usage while still outperforming PPO overall. Therefore, our work offers a
 301 practical and efficient option for resource-constrained LLM math reasoning fine-tuning.

References

- 302
- 303 [1] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep
304 reinforcement learning from human preferences. *Advances in neural information processing*
305 *systems*, 30, 2017.
- 306 [2] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
307 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John
308 Schulman. Training verifiers to solve math word problems, 2021.
- 309 [3] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit
310 Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the
311 frontier with advanced reasoning, multimodality, long context, and next generation agentic
312 capabilities. *arXiv preprint arXiv:2507.06261*, 2025.
- 313 [4] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen
314 Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf,
315 2024.
- 316 [5] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware mini-
317 mization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
- 318 [6] Yao Fu, Yihang Jin, Chunxia Zhang, Junmin Liu, and Haishan Ye. Zero-order sharpness-aware
319 minimization. *arXiv preprint arXiv:2511.09156*, 2025.
- 320 [7] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
321 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in
322 llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- 323 [8] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolu-
324 tion strategies. *Evolutionary computation*, 9(2):159–195, 2001.
- 325 [9] Feihu Jin, Yifan Liu, and Ying Tan. Derivative-free optimization for low-rank adaptation in
326 large language models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*,
327 2024.
- 328 [10] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and
329 scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on*
330 *Computer Vision and Pattern Recognition*, pages 12360–12370, 2022.
- 331 [11] OpenAI. O3 and o4 mini system card, 2025.
- 332 [12] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin,
333 Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to
334 follow instructions with human feedback. *Advances in neural information processing systems*,
335 35:27730–27744, 2022.
- 336 [13] Xin Qiu, Yulu Gan, Conor F Hayes, Qiyao Liang, Elliot Meyerson, Babak Hodjat, and Risto
337 Miikkulainen. Evolution strategies at scale: Llm fine-tuning beyond reinforcement learning.
338 *arXiv preprint arXiv:2509.24372*, 2025.
- 339 [14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal
340 policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- 341 [15] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
342 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical
343 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- 344 [16] Orlando Grabiél Toledano-López, Julio Madera, Hector González, Alfredo Simón-Cuevas,
345 Thomas Demeester, and Erik Mannens. Fine-tuning mt5-based transformer via cma-es for
346 sentiment analysis. In *IberLEF@ SEPLN*, 2022.
- 347 [17] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber.
348 Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

- 349 [18] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
350 Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*
351 *arXiv:2505.09388*, 2025.
- 352 [19] Feiyang Ye, Yueming Lyu, Xuehao Wang, Masashi Sugiyama, Yu Zhang, and Ivor Tsang.
353 Sharpness-aware black-box optimization. *arXiv preprint arXiv:2410.12457*, 2024.
- 354 [20] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai,
355 Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement
356 learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- 357 [21] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang,
358 Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint*
359 *arXiv:2507.18071*, 2025.
- 360 [22] Qihuang Zhong, Liang Ding, Li Shen, Peng Mi, Juhua Liu, Bo Du, and Dacheng Tao. Improving
361 sharpness-aware minimization with fisher mask for better generalization on language models.
362 *arXiv preprint arXiv:2210.05497*, 2022.
- 363 [23] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei,
364 Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences.
365 *arXiv preprint arXiv:1909.08593*, 2019.

366 **A Evolution Strategies**

Algorithm 2 Evolution Strategies (ES)

Input: initial parameters θ_0 , reward function R , population size N , noise scale σ , learning rate α , iterations T
Output: model parameters θ_T
for $t = 0$ **to** $T - 1$ **do**
 for $n = 1$ **to** N **do**
 Sample Gaussian noise $\epsilon_n \sim \mathcal{N}(0, I)$
 Construct perturbed parameters $\theta_t^{(n)} = \theta_t + \sigma \epsilon_n$
 Evaluate reward $r_n = R(\theta_t^{(n)})$
 end for
 Compute mean $\mu_r = \frac{1}{N} \sum_{n=1}^N r_n$
 Compute std $s_r = \sqrt{\frac{1}{N} \sum_{n=1}^N (r_n - \mu)^2}$
 for $n = 1$ **to** N **do**
 Normalize reward $\hat{r}_n = \frac{r_n - \mu_r}{s_r + \varepsilon}$
 end for
 Update parameters $\theta_{t+1} = \theta_t + \alpha \cdot \frac{1}{N} \sum_{n=1}^N \hat{r}_n \epsilon_n$
end for

367 Algorithm 2 presents the basic framework of using ES to fine-tune LLMs.

368 Specifically, Given a pretrained LLM with parameters θ_0 and a reward function $R(\theta)$, we aim to
369 maximize the expected reward by iteratively updating model parameters. At iteration t , we sample N
370 i.i.d. Gaussian noises $\epsilon_n \sim \mathcal{N}(0, I)$ for $n = 1, \dots, N$, and construct perturbed models:

$$\theta_t^{(n)} = \theta_t + \sigma \epsilon_n, \quad n = 1, \dots, N. \quad (9)$$

371 We evaluate each perturbed model to obtain reward scores:

$$r_n = R(\theta_t^{(n)}), \quad n = 1, \dots, N. \quad (10)$$

372 We then apply z-score normalization to keep the reward scale consistent:

$$\mu_r = \frac{1}{N} \sum_{n=1}^N r_n, \quad s_r = \sqrt{\frac{1}{N} \sum_{n=1}^N (r_n - \mu_r)^2}, \quad (11)$$

373

$$\hat{r}_n = \frac{r_n - \mu_r}{s_r + \varepsilon}, \quad n = 1, \dots, N, \quad (12)$$

374 where ε is a small constant for numerical stability. Finally, we update the parameters by aggregating
375 perturbations weighted by normalized rewards:

$$\theta_{t+1} = \theta_t + \alpha \cdot \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \hat{r}_n \epsilon_n. \quad (13)$$

376 To reduce tuning complexity, we absorb $1/\sigma$ into the learning rate α . The update rule becomes:

$$\theta_{t+1} = \theta_t + \alpha \cdot \frac{1}{N} \sum_{n=1}^N \hat{r}_n \epsilon_n. \quad (14)$$

B ESSAM with Frobenius-Normalized SAM Perturbation

Algorithm 3 ESSAM with Frobenius-Normalized SAM Perturbation (ESSAM-F)

Input: initial parameters θ_0 , reward function R , ES population size N , SAM population size N_{SAM} , noise scale σ , SAM radius ρ , learning rate α , iterations T , number of parallel processes P

Output: fine-tuned parameters θ_T

Create P processes, each instantiates a model with the same initial parameters θ_0 , with one process as the main process

for $t = 0$ to $T - 1$ **do**

(Stage 1) Build a Frobenius-normalized SAM neighborhood point θ_t^{SAM}

 Sample N_{SAM} random seeds $s_1, \dots, s_{N_{\text{SAM}}}$

$\{r_n\}_{n=1}^{N_{\text{SAM}}} \leftarrow \text{SRE}(\theta_t, R, \sigma, \{s_n\}_{n=1}^{N_{\text{SAM}}}, N, P)$

 Compute mean μ_r

$g_t^{\text{norm}} \leftarrow \text{WEIGHTEDNOISEFROBNORM}(\{s_n\}_{n=1}^{N_{\text{SAM}}}, \{r_n - \mu_r\}_{n=1}^{N_{\text{SAM}}}, N)$

$\theta_t^{\text{SAM}} \leftarrow \text{DIPU}\left(\theta_t, \{s_n\}_{n=1}^{N_{\text{SAM}}}, \{r_n - \mu_r\}_{n=1}^{N_{\text{SAM}}}, -\frac{\rho}{g_t^{\text{norm}} + \epsilon}, N\right)$

 Broadcast parameters of all processes to θ_t^{SAM}

(Stage 2) Evaluate around θ_t^{SAM} **and update** θ_t

 Sample N random seeds $s_1^{\text{SAM}}, \dots, s_N^{\text{SAM}}$

$\{r_n^{\text{SAM}}\}_{n=1}^N \leftarrow \text{SRE}(\theta_t^{\text{SAM}}, R, \sigma, \{s_n^{\text{SAM}}\}_{n=1}^N, N, P)$

 Compute mean μ_r^{SAM} , std s_r^{SAM}

 Normalize $\hat{r}_n^{\text{SAM}} = \frac{r_n^{\text{SAM}} - \mu_r^{\text{SAM}}}{s_r^{\text{SAM}} + \delta}$

$\theta_t \leftarrow \text{DIPU}\left(\theta_t^{\text{SAM}}, \{s_n\}_{n=1}^{N_{\text{SAM}}}, \{r_n - \mu_r\}_{n=1}^{N_{\text{SAM}}}, \frac{\rho}{g_t^{\text{norm}} + \epsilon}, N\right)$

$\theta_{t+1} \leftarrow \text{DIPU}(\theta_t, \{s_n^{\text{SAM}}\}_{n=1}^N, \{\hat{r}_n^{\text{SAM}}\}_{n=1}^N, \alpha, N)$

 Broadcast parameters of all processes to θ_{t+1}

end for

378 Algorithm 3 presents the detailed algorithmic procedure of ESSAM-F, the accelerated variant of
 379 ESSAM. The difference between ESSAM-F and ESSAM lies in the way the SAM neighborhood
 380 point is constructed in the first stage, it introduces a SAM mechanism based on Frobenius norm
 381 normalization. It first uses a smaller SAM population and a centered reward weighted aggregation of
 382 the sampled noise directions to move the model parameters to a nearby neighborhood point, where the
 383 perturbation magnitude is normalized by the Frobenius norm of the aggregated direction. It then uses
 384 the perturbations and rewards computed at this neighborhood point to update the original parameters.

385 Specifically, given a pretrained LLM with parameters θ_0 and a reward function $R(\theta)$, we aim to
 386 maximize the expected reward by iteratively updating model parameters. At iteration t , we first
 387 sample $N_{\text{SAM}} (N_{\text{SAM}} \ll N)$ i.i.d. Gaussian noises $\epsilon_n \sim \mathcal{N}(0, I)$ for $n = 1, \dots, N_{\text{SAM}}$, and
 388 construct perturbed models:

$$\theta_t^{(n)} = \theta_t + \sigma \epsilon_n, \quad n = 1, \dots, N_{\text{SAM}}. \quad (15)$$

389 We evaluate each perturbed model to obtain reward scores:

$$r_n = R(\theta_t^{(n)}), \quad n = 1, \dots, N_{\text{SAM}}. \quad (16)$$

390 We compute the reward mean:

$$\mu_r = \frac{1}{N_{\text{SAM}}} \sum_{n=1}^{N_{\text{SAM}}} r_n. \quad (17)$$

391 Then, instead of applying z-score normalization, we directly use centered rewards to compute a
 392 weighted aggregation of the sampled noise directions:

$$g_t = \frac{1}{N_{\text{SAM}}} \sum_{n=1}^{N_{\text{SAM}}} (r_n - \mu_r) \epsilon_n. \quad (18)$$

393 To normalize its scale, we compute the Frobenius norm

$$g_t^{\text{norm}} = \|g_t\|_F, \quad (19)$$

394 and construct the SAM neighborhood point as

$$\theta_t^{\text{SAM}} = \theta_t - \rho \cdot \frac{g_t}{g_t^{\text{norm}} + \epsilon}, \quad (20)$$

395 where ρ is the SAM radius and ϵ is a small constant for numerical stability.

396 After obtaining θ_t^{SAM} , The procedure of the second stage is exactly the same as that of the second
397 stage in ESSAM, and the population size is N . The detailed procedure of ESSAM-F is shown in
398 Algorithm 3.

Algorithm 4 Seed Replay Eval (SRE)

Input: parameters θ , reward function R , noise scale σ , seeds $\{s_n\}_{n=1}^N$, population size N , number of parallel processes P
Output: rewards $\{r_n\}_{n=1}^N$
Assign seeds $\{s_n\}_{n=1}^N$ to P processes for fully-parallel evaluations
for $n = 1$ **to** N **do**
 For the process handling s_n , reset its random number generator using seed s_n
 for each LLM layer **do**
 Using seed s_n , sample noise $\epsilon_{n,l} \sim \mathcal{N}(0, I)$
 Perturb in-place: $\theta_l \leftarrow \theta_l + \sigma \epsilon_{n,l}$
 end for
 Evaluate reward $r_n = R(\theta)$ using greedy decoding
 For the process handling s_n , reset its random number generator using seed s_n
 for each LLM layer **do**
 Using seed s_n , sample noise $\epsilon_{n,l} \sim \mathcal{N}(0, I)$
 Restore in-place: $\theta_l \leftarrow \theta_l - \sigma \epsilon_{n,l}$
 end for
end for

Algorithm 5 Decomposed In-place Update (DIPU)

Input: parameters θ , seeds $\{s_n\}_{n=1}^N$, weights $\{w_n\}_{n=1}^N$, step size η , population size N
Output: updated parameters θ
for $n = 1$ **to** N **do**
 For the process handling s_n , reset its random number generator using seed s_n
 for each LLM layer **do**
 Using seed s_n , sample noise $\epsilon_{n,l} \sim \mathcal{N}(0, I)$
 Update in-place: $\theta_l \leftarrow \theta_l + \eta \cdot \frac{1}{N} w_n \epsilon_{n,l}$
 end for
end for

Algorithm 6 Weighted Noise Frobenius Norm

Input: seeds $\{s_n\}_{n=1}^N$, coefficients $\{c_n\}_{n=1}^N$, population size N
Output: Frobenius norm $g^{\text{norm}} = \left\| \sum_{n=1}^N c_n \epsilon_n \right\|_F$
total_sq $\leftarrow 0$
for each LLM layer l **do**
 $u_l \leftarrow 0$
 for $n = 1$ **to** N **do**
 Reset random number generator using seed s_n
 Using seed s_n , sample noise $\epsilon_{n,l} \sim \mathcal{N}(0, I)$
 $u_l \leftarrow u_l + c_n \epsilon_{n,l}$
 end for
 total_sq \leftarrow total_sq + $\|u_l\|_F^2$
end for
 $g^{\text{norm}} \leftarrow \sqrt{\text{total_sq}}$

400 **D Proof**

401 By the Taylor's expansion, we have

$$R(\theta + \sigma\epsilon_n) = R(\theta) + \sigma \langle \nabla R(\theta), \epsilon_n \rangle + \alpha(\theta, \sigma\epsilon_n), \quad (21)$$

402 where

$$\alpha(\theta, \sigma\epsilon_n) \triangleq R(\theta + \sigma\epsilon_n) - \left(R(\theta) + \sigma \langle \nabla R(\theta), \epsilon_n \rangle \right). \quad (22)$$

403 By the \mathcal{L} -smoothness, we can obtain that

$$\mathbb{E}[|\alpha(\theta, \sigma\epsilon_n)|] \leq \mathbb{E}\left[\frac{\mathcal{L}\sigma^2}{2} \|\epsilon_n\|^2\right] \leq \frac{(d+2)\mathcal{L}\sigma^2}{2}, \quad (23)$$

404 **Lemma 1.** *Let the estimated variance s_r be defined in Eq. (5) Then it holds that*

$$\mathbb{E}[s_r^2] = \sigma^2 \|\nabla R(\theta)\|^2 + \zeta, \quad (24)$$

405 with

$$\begin{aligned} \zeta = & \mathbb{E}\left[\frac{1}{N-1} \sum_{i=1}^N \left(\alpha(\theta, \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N \alpha(\theta, \sigma\epsilon_j) \right)^2\right] \\ & + \mathbb{E}\left[\frac{2\sigma}{N-1} \sum_{i=1}^N \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \left(\alpha(\theta, \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N \alpha(\theta, \sigma\epsilon_j) \right)\right]. \end{aligned}$$

406 Moreover,

$$|\zeta| \leq \frac{(d+2)^2 \mathcal{L}^2 \sigma^4}{4} + \frac{4N}{N-1} (d+1)^{\frac{1}{2}} (d+2) \mathcal{L} \sigma^3 \|\nabla R(\theta)\|. \quad (25)$$

407 *Proof.* First, we have

$$\begin{aligned} & R(\theta + \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N R(\theta + \sigma\epsilon_j) \\ & \stackrel{\text{Eq. (21)}}{=} R(\theta) + \sigma \langle \nabla R(\theta), \epsilon_i \rangle + \alpha(\theta, \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N \left(R(\theta) + \sigma \langle \nabla R(\theta), \epsilon_j \rangle + \alpha(\theta, \sigma\epsilon_j) \right) \\ & = \sigma \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle + \alpha(\theta, \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N \alpha(\theta, \sigma\epsilon_j). \end{aligned} \quad (26)$$

408 Define

$$\beta_i \triangleq \alpha(\theta, \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N \alpha(\theta, \sigma\epsilon_j).$$

409 Then

$$\begin{aligned} \sum_{i=1}^N \left(R(\theta + \sigma\epsilon_i) - \frac{1}{N} \sum_{j=1}^N R(\theta + \sigma\epsilon_j) \right)^2 &= \sum_{i=1}^N \left(\sigma \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle + \beta_i \right)^2 \\ &= \sum_{i=1}^N \left(\sigma^2 \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle^2 + \beta_i^2 \right. \\ & \quad \left. + 2\sigma \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \beta_i \right). \end{aligned} \quad (27)$$

410 Furthermore, it holds that

$$\mathbb{E} \left[\sum_{i=1}^N \beta_i^2 \right] = \mathbb{E} \left[\left(1 - \frac{1}{N} \right) \sum_{i=1}^N \alpha^2(\theta, \sigma \epsilon_i) \right] \leq (N-1) \frac{\mathcal{L}^2 \sigma^4 (d+2)^2}{4}, \quad (28)$$

411 and

$$\mathbb{E} \left[\sum_{i=1}^N \left| \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \beta_i \right| \right] \leq \mathbb{E} \left[N \left\| \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \right\| \|\beta_i\| \right] \\ \stackrel{\text{Eq. (23)}}{\leq} 2N(d+1)^{\frac{1}{2}}(d+2)\mathcal{L}\sigma^2 \|\nabla R(\theta)\|. \quad (29)$$

412 Therefore, we can obtain that

$$\mathbb{E}[s_r^2] = \mathbb{E} \left[\frac{1}{N-1} \sum_{i=1}^N \left(\sigma^2 \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle^2 + \beta_i^2 + 2\sigma \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \beta_i \right) \right] \\ = \sigma^2 \|\nabla R(\theta)\|^2 + \zeta. \quad (30)$$

413 Finally,

$$|\zeta| \leq \mathbb{E} \left[\frac{1}{N-1} \sum_{i=1}^N \beta_i^2 + \frac{2\sigma}{N-1} \sum_{i=1}^N \left| \left\langle \nabla R(\theta), \epsilon_i - \frac{1}{N} \sum_{j=1}^N \epsilon_j \right\rangle \beta_i \right| \right] \\ \leq \frac{(d+2)^2 \mathcal{L}^2 \sigma^4}{4} + \frac{4N}{N-1} (d+1)^{\frac{1}{2}} (d+2) \mathcal{L} \sigma^3 \|\nabla R(\theta)\|. \quad (31)$$

414

□

415 **Lemma 2.** *Let the stochastic gradient estimation g_t be defined in Eq. (4) Then its norm satisfies*

$$\mathbb{E}[\|g_t\|^2] = \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \|\nabla R(\theta)\|^2 + \gamma, \quad (32)$$

416 with

$$\gamma = \mathbb{E} \left[\frac{1}{\sigma^2} \frac{1}{N^2} \left(\sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n \right)^2 + \frac{1}{\sigma^2} \frac{1}{N^4} \left(\sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right)^2 \right. \\ \left. - 2 \frac{1}{\sigma^2} \frac{1}{N^3} \left(\sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n \right)^\top \left(\sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right) \right. \\ \left. + 2 \left\langle \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \langle \nabla R(\theta), \epsilon_i \rangle \epsilon_n, \right. \right. \\ \left. \left. \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right\rangle \right]. \quad (33)$$

417 Furthermore, it holds that

$$|\gamma| \leq \frac{(d+2)^2 \mathcal{L}^2 \sigma^2}{N} + \frac{4(d+2)\mathcal{L}\sigma}{N} \|\nabla R(\theta)\|. \quad (34)$$

418 *Proof.* First, we have

$$\begin{aligned}
g_t &= \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle \nabla R(\theta), \epsilon_j \rangle \epsilon_i \\
&\quad + \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n
\end{aligned} \tag{35}$$

419 Thus,

$$\begin{aligned}
\mathbb{E}[g_t] &= \mathbb{E}\left[\frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle \nabla R(\theta), \epsilon_j \rangle \epsilon_i\right] \\
&\quad + \mathbb{E}\left[\frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n\right] \\
&= \frac{N-1}{N} \nabla R(\theta) + \tau
\end{aligned} \tag{36}$$

$$|\tau| = \mathbb{E}\left[\frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n\right] \leq \mathcal{L} \sigma (d+2) \tag{37}$$

420 Define

$$A \triangleq \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle \nabla R(\theta), \epsilon_j \rangle \epsilon_i, \tag{38}$$

$$B \triangleq \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n. \tag{39}$$

$$\begin{aligned}
\|g_t\|^2 &= \left\| \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle \nabla R(\theta), \epsilon_j \rangle \epsilon_i \right\|^2 \\
&\quad + \left\| \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right\|^2 \\
&\quad + 2 \left\langle \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \langle \nabla R(\theta), \epsilon_i \rangle \epsilon_n, \right. \\
&\quad \quad \left. \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right\rangle.
\end{aligned} \tag{40}$$

421 Then

$$\begin{aligned}
\|A\|^2 &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \langle \epsilon_i \epsilon_i^\top \nabla R(\theta), \epsilon_j \epsilon_j^\top \nabla R(\theta) \rangle \\
&\quad + \frac{1}{N^4} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N \sum_{l=1}^N \langle \langle \nabla R(\theta), \epsilon_i \rangle \epsilon_j, \langle \nabla R(\theta), \epsilon_k \rangle \epsilon_l \rangle \\
&\quad - \frac{2}{N^3} \left\langle \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta), \sum_{i=1}^N \sum_{j=1}^N \langle \nabla R(\theta), \epsilon_i \rangle \epsilon_j \right\rangle. \tag{41}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E} \left[\|A\|^2 \right] &= \frac{N+d}{N} \|\nabla R(\theta)\|^2 + \frac{3(N-1)+d+1}{N^3} \|\nabla R(\theta)\|^2 - \frac{2(N+d)}{N^2} \|\nabla R(\theta)\|^2 \\
&= \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \|\nabla R(\theta)\|^2 \tag{42}
\end{aligned}$$

$$\begin{aligned}
\|B\|^2 &= \frac{1}{\sigma^2} \frac{1}{N^2} \left\| \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n \right\|^2 + \frac{1}{\sigma^2} \frac{1}{N^4} \left\| \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right\|^2 \\
&\quad - 2 \frac{1}{\sigma^2} \frac{1}{N^3} \left(\sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n \right)^\top \left(\sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right). \tag{43}
\end{aligned}$$

$$\begin{aligned}
\langle A, B \rangle &= \left\langle \frac{1}{N} \sum_{n=1}^N \epsilon_n \epsilon_n^\top \nabla R(\theta) - \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \langle \nabla R(\theta), \epsilon_i \rangle \epsilon_n, \right. \\
&\quad \left. \frac{1}{\sigma} \frac{1}{N} \sum_{n=1}^N \alpha(\theta, \sigma \epsilon_n) \epsilon_n - \frac{1}{\sigma} \frac{1}{N^2} \sum_{n=1}^N \sum_{i=1}^N \alpha(\theta, \sigma \epsilon_i) \epsilon_n \right\rangle. \tag{44}
\end{aligned}$$

$$\gamma = \mathbb{E} \left[\|B\|^2 \right] + 2\mathbb{E} \left[\langle A, B \rangle \right] \tag{45}$$

422 Thus,

$$\begin{aligned}
\mathbb{E} [\|g_t\|^2] &= \mathbb{E} \left[\|A\|^2 \right] + \mathbb{E} \left[\|B\|^2 \right] + 2\mathbb{E} \left[\langle A, B \rangle \right] \\
&= \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \|\nabla R(\theta)\|^2 + \gamma \tag{46}
\end{aligned}$$

423 There are also,

$$\mathbb{E} \left[\|B\|^2 \right] \leq \frac{\mathcal{L}^2 \sigma^2 (d+2)^2}{4N} + \frac{\mathcal{L}^2 \sigma^2 (d+2)^2}{4N} + \frac{\mathcal{L}^2 \sigma^2 (d+2)^2}{2N} = \frac{\mathcal{L}^2 \sigma^2 (d+2)^2}{N} \tag{47}$$

$$\begin{aligned}
\mathbb{E} \left[\langle A, B \rangle \right] &\leq \frac{\mathcal{L} \sigma (d+2)}{2N} \|\nabla R(\theta)\| + \frac{\mathcal{L} \sigma (d+2)}{2N} \|\nabla R(\theta)\| \\
&\quad + \frac{\mathcal{L} \sigma (d+2)}{2N} \|\nabla R(\theta)\| + \frac{\mathcal{L} \sigma (d+2)}{2N} \|\nabla R(\theta)\| \\
&= \frac{2\mathcal{L} \sigma (d+2)}{N} \|\nabla R(\theta)\|. \tag{48}
\end{aligned}$$

424 Finally,

$$|\gamma| \leq \frac{(d+2)^2 \mathcal{L}^2 \sigma^2}{N} + \frac{4(d+2)\mathcal{L}\sigma}{N} \|\nabla R(\theta)\|. \quad (49)$$

425

□

426 **D.1 Proof of Proposition 1**

427 *Proof.* From Lemma 1 and Lemma 2, we know

$$\mathbb{E}[\|g_t\|^2] = \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \|\nabla R(\theta)\|^2 + \gamma, \quad \gamma = O(\sigma), \quad (50)$$

$$\mathbb{E}[s_r^2] = \sigma^2 \|\nabla R(\theta)\|^2 + \zeta, \quad \zeta = O(\sigma^3). \quad (51)$$

428 Comparing Eq. 50 and Eq. 51, and ignoring the perturbation terms ζ and γ (which are higher-order
429 terms with respect to σ), we can obtain that:

$$\mathbb{E}[\|g_t\|^2] = \frac{N^3 + (d-2)N^2 + (3-2d)N + d-2}{N^3} \cdot \sigma^{-2} \cdot \mathbb{E}[s_r^2]. \quad (52)$$

430

□

431 **E Hyperparameter Settings**

432 **Models.** In our experiments, we train the following models.

- 433 • **Qwen:** Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct;
- 434
- 435 • **Llama:** LLaMA3.2-1B-Instruct, LLaMA3.2-3B-Instruct, LLaMA3.1-8B-Instruct.

436 **Datasets.** In our experiments, we use the GSM8K dataset to train and evaluate LLMs. The problem and prompt template are shown in Figure 8. This task mainly tests text understanding, multi step reasoning, and arithmetic computation.

Problem: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

Template: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer.
Please solve the following problem: {question}
Show your work in <think> </think> tags. And put the final numerical answer after ####.

Figure 8: An example GSM8K problem and the prompt template.

439 **Reward function.** For verifiable math reasoning tasks such as GSM8K and Countdown, we use a
440 rule based reward function. The rules are as follows, we split the reward into an outcome accuracy
441 reward and an output format reward:

$$R_{\text{accuracy}}(\hat{y}, y) = \begin{cases} 1, & \text{is_equivalent}(\hat{y}, y), \\ 0, & \text{otherwise.} \end{cases} \quad (53)$$

$$R_{\text{format}}(a) = \begin{cases} 1.0, & \text{the output follows the full format;} \\ 0.6, & \text{it contains both a } \textit{thinking marker} \\ & \text{and an } \textit{answer marker}; \\ 0.1, & \text{it contains a } \textit{thinking marker} \text{ only;} \\ 0.5, & \text{it contains an } \textit{answer marker} \text{ only;} \\ 0.0, & \text{otherwise.} \end{cases} \quad (54)$$

443 The final reward we use is a weighted sum of the accuracy reward and the format reward:

$$R(a, y) = R_{\text{answer}}(\hat{y}, y) + 0.1 R_{\text{format}}(a), \quad (55)$$

444 where y is the ground truth answer, \hat{y} is the model predicted answer, and a is the full model response
445 to the question.

446 **Hyperparameter setting.** For all ES and ESSAM experiments, we set the hyperparameter $\rho = 10^{-4}$
447 and the learning rate $\alpha = 2.5 \times 10^{-4}$. The population size N and the noise scale σ are chosen
448 as shown in Table 3. For the PPO and GRPO experiments, we set the learning rate α and the
449 KL-divergence penalty coefficient β as shown in Table 4. In addition, for the GRPO experiments, we
450 set the group size to $\bar{N} = 8$.

451 **Compute resources.** Our experiments are conducted using four NVIDIA A100 80GB PCIe GPUs.

Table 3: Hyperparameter settings for ES and ESSAM across different models. Each pair (\cdot, \cdot) denotes (population size N , noise scale σ); the symbol \checkmark indicates the hyperparameter setting for each model-method combination.

METHOD	MODEL	$(40, 1.5 \times 10^{-3})$	$(40, 2 \times 10^{-3})$	$(60, 1.5 \times 10^{-3})$	$(60, 2 \times 10^{-3})$
ES	QWEN-0.5B-INSTRUCT	\checkmark			
	QWEN-1.5B-INSTRUCT			\checkmark	
	QWEN-3B-INSTRUCT		\checkmark		
	QWEN-7B-INSTRUCT				\checkmark
	LLAMA-1B-INSTRUCT	\checkmark			
	LLAMA-3B-INSTRUCT				\checkmark
	LLAMA-8B-INSTRUCT				\checkmark
ESSAM	QWEN-0.5B-INSTRUCT	\checkmark			
	QWEN-1.5B-INSTRUCT			\checkmark	
	QWEN-3B-INSTRUCT		\checkmark		
	QWEN-7B-INSTRUCT				\checkmark
	LLAMA-1B-INSTRUCT	\checkmark			
	LLAMA-3B-INSTRUCT				\checkmark
	LLAMA-8B-INSTRUCT				\checkmark

Table 4: Hyperparameter settings for PPO and GRPO across different models. Each pair (\cdot, \cdot) denotes (learning rate α , KL-divergence penalty coefficient β); the symbol \checkmark indicates the hyperparameter setting for each model-method combination.

METHOD	MODEL	$(10^{-6}, 10^{-3})$	$(10^{-5}, 10^{-3})$	$(10^{-6}, 5 \times 10^{-3})$	$(10^{-5}, 5 \times 10^{-3})$
PPO	QWEN-0.5B-INSTRUCT	\checkmark			
	QWEN-1.5B-INSTRUCT	\checkmark			
	QWEN-3B-INSTRUCT	\checkmark			
	QWEN-7B-INSTRUCT		\checkmark		
	LLAMA-1B-INSTRUCT	\checkmark			
	LLAMA-3B-INSTRUCT				\checkmark
	LLAMA-8B-INSTRUCT			\checkmark	
GRPO	QWEN-0.5B-INSTRUCT	\checkmark			
	QWEN-1.5B-INSTRUCT	\checkmark			
	QWEN-3B-INSTRUCT	\checkmark			
	QWEN-7B-INSTRUCT	\checkmark			
	LLAMA-1B-INSTRUCT				\checkmark
	LLAMA-3B-INSTRUCT		\checkmark		
	LLAMA-8B-INSTRUCT	\checkmark			

452 F Generalization Experiments

453 As shown in Figure 9, we further present the generalization performance of more fine-tuned models.

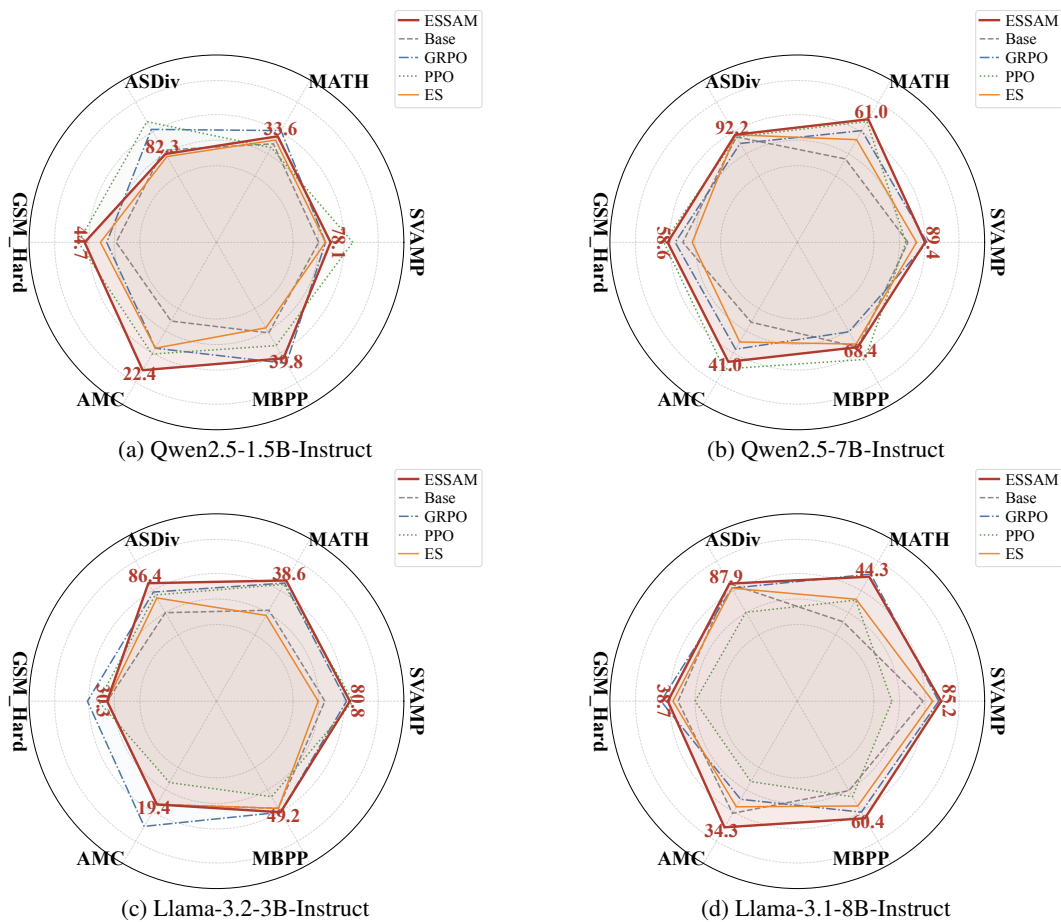


Figure 9: More generalization experiments.

454 G Detailed GPU Memory Usage

455 Table 5 shows the detailed GPU memory usage when fine-tuning different LLMs with different
 456 algorithms.

Table 5: The detailed GPU memory usage.

BASE MODEL	RL		ES	ESSAM
	PPO	GRPO		
QWEN-2.5-0.5B-INSTRUCT	57.32	34.20	2.99	2.99
QWEN-2.5-1.5B-INSTRUCT	123.40	70.80	6.57	6.57
QWEN-2.5-3B-INSTRUCT	208.96	102.24	9.96	9.96
QWEN-2.5-7B-INSTRUCT	314.40	212.88	19.85	19.85
LLAMA-3.2-1B-INSTRUCT	99.20	65.68	4.90	4.90
LLAMA-3.2-3B-INSTRUCT	201.24	92.60	11.49	11.49
LLAMA-3.1-8B-INSTRUCT	314.44	219.48	21.45	21.45

457 **H Training Reward Curve**

458 As shown in Figure 10, we present the curves of the evolution of the average training reward for more
459 models fine-tuned with ES and ESSAM.

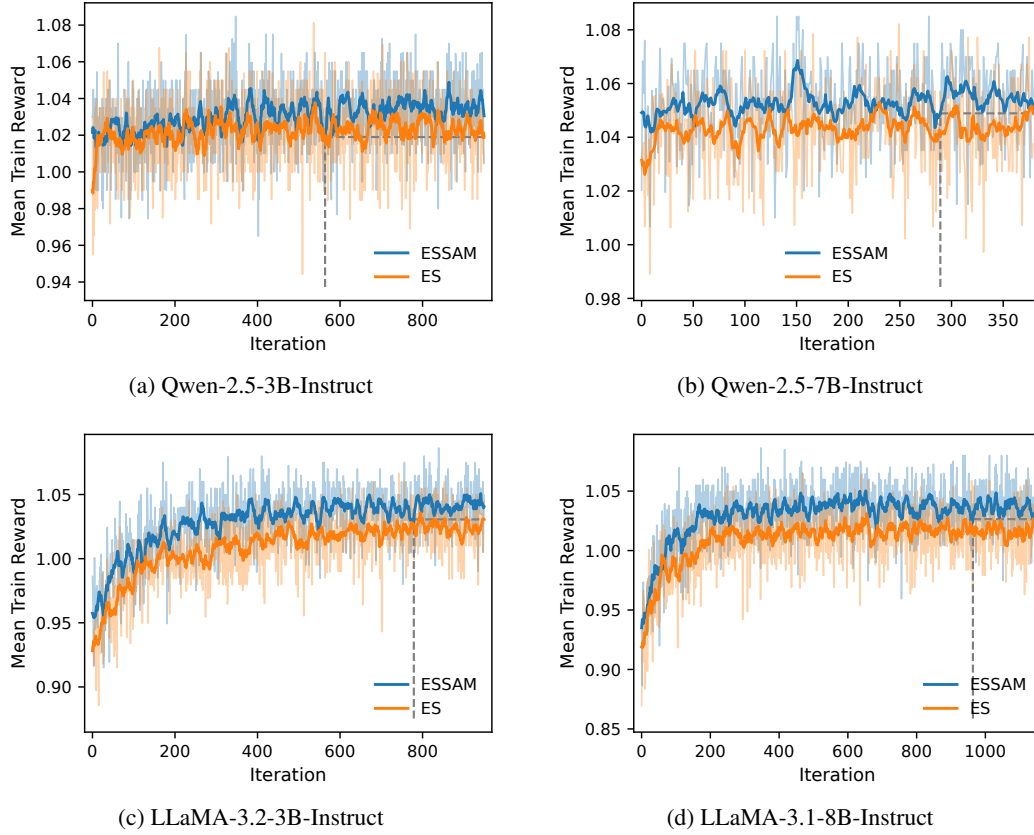


Figure 10: The training mean reward curves of ESSAM and ES.

460 **I Test Set Accuracy**

461 As shown in Figure 11, we present schematic diagrams of the evolution of test set accuracy for more
 462 models fine-tuned with different algorithms.

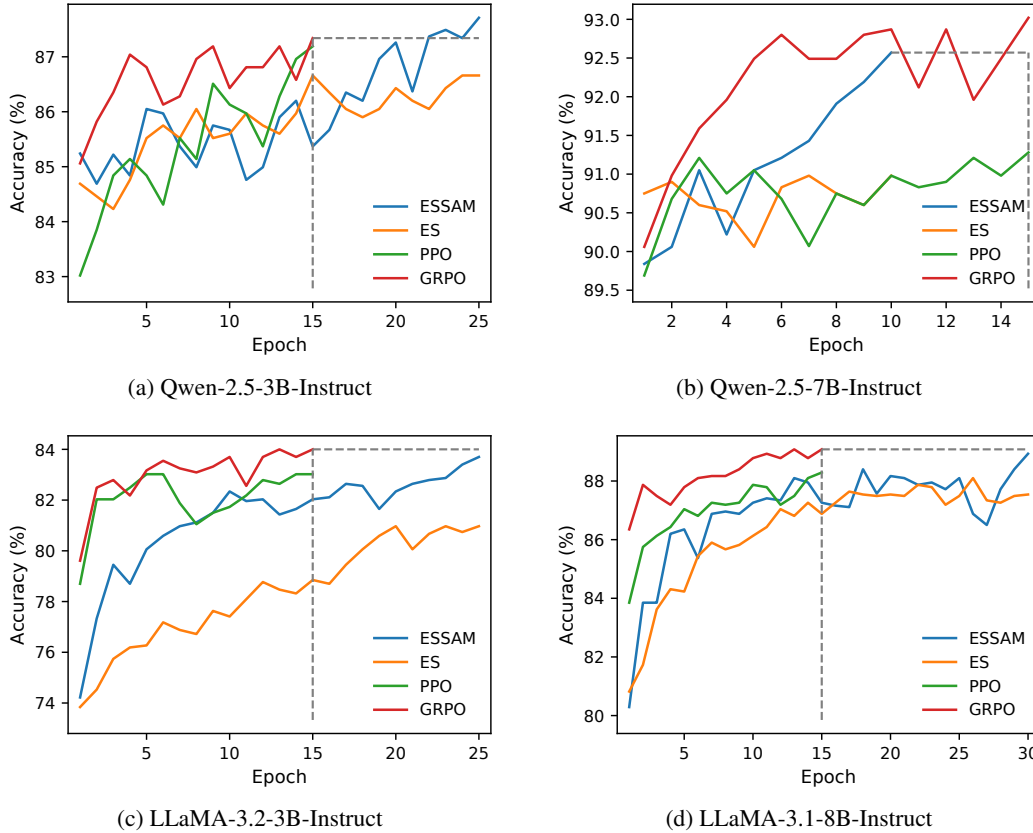


Figure 11: A schematic illustration of how the test set accuracy of models fine-tuned with different algorithms evolves over training.

463 **J Example**

464 In this section, we present example responses from models fine-tuned with ESSAM.

Qwen2.5-0.5b-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take? Show your work in `<think>` `</think>` tags. And put the final numerical answer after `####`.

Model Response: Let me solve this step by step. `<think>` First, we know that the robe takes 2 bolts of blue fiber. According to the problem, the white fiber is half the amount of blue fiber. So, the white fiber is $\frac{2}{2} = 1$ bolt. Therefore, the total number of bolts needed for the robe is the sum of the blue and white fibers, which is $2 + 1 = 3$ bolts. `<think>` `####` 3

465

Qwen2.5-1.5b-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for \$2 per fresh duck egg. How much in dollars does she make every day at the farmers' market? Show your work in `<think>` `</think>` tags. And put the final numerical answer after ####.

Model Response: Let me solve this step by step. `<think>` First, let's determine the total number of eggs laid by Janet's ducks per day. Janet's ducks lay 16 eggs per day. Janet eats 3 eggs for breakfast every morning. So, the number of eggs left after breakfast is $16 - 3 = 13$ eggs. Next, she bakes muffins for her friends with 4 eggs. So, the number of eggs left after baking muffins is $13 - 4 = 9$ eggs. These 9 eggs are sold at the farmers' market. Each egg is sold for \$2. Therefore, the total amount of money she makes from selling the eggs is $9 \times 2 = 18$ dollars. `<think>` #### 18

466

Qwen2.5-3b-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: Every day, Wendi feeds each of her chickens three cups of mixed chicken feed, containing seeds, mealworms and vegetables to help keep them healthy. She gives the chickens their feed in three separate meals. In the morning, she gives her flock of chickens 15 cups of feed. In the afternoon, she gives her chickens another 25 cups of feed. How many cups of feed does she need to give her chickens in the final meal of the day if the size of Wendi's flock is 20 chickens? Show your work in `<think>` `</think>` tags. And put the final numerical answer after ####.

Model Response: Let me solve this step by step. `<think>` First, we need to determine the total amount of feed Wendi gives to her chickens in a day. We know that each chicken gets 3 cups of feed per day. Wendi has a flock of 20 chickens. So, the total feed given to the chickens in a day is:

$$20 \text{ chickens} \times 3 \text{ cups/chicken} = 60 \text{ cups}$$

Next, we need to account for the feed given in the morning and the afternoon. Wendi gives 15 cups in the morning and 25 cups in the afternoon.

So, the total feed given in the morning and afternoon is:

$$15 \text{ cups} + 25 \text{ cups} = 40 \text{ cups}$$

To find out how much feed is needed for the final meal, we subtract the feed given in the morning and afternoon from the total daily feed:

$$60 \text{ cups} - 40 \text{ cups} = 20 \text{ cups}$$

Therefore, the amount of feed needed for the final meal is 20 cups. `<think>` #### 20

467

Qwen2.5-7b-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

468

Show your work in `<think>` `</think>` tags. And put the final numerical answer after `####`.

Model Response: Let me solve this step by step. `<think>` First, we need to calculate the new value of the house after the repairs. The original value of the house was \$80,000, and the repairs increased the value by 150%.

The increase in value due to repairs is:

$$80,000 \times 1.50 = 120,000$$

So, the new value of the house is:

$$80,000 + 120,000 = 200,000$$

Next, we need to calculate the total cost Josh incurred. This includes the purchase price of the house and the repair costs:

$$80,000 + 50,000 = 130,000$$

Now, we can find the profit by subtracting the total cost from the new value of the house:

$$200,000 - 130,000 = 70,000$$

So, the profit Josh made is \$70,000. `<think>` `####` 70000

469

LLaMA-3.2-1B-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week? Show your work in `<think>` `</think>` tags. And put the final numerical answer after `####`.

Model Response: Let me solve this step by step. `<think>` Step 1: Calculate the total number of sprints per week

James decides to run 3 sprints 3 times a week, so the total number of sprints per week is:

$$3 \text{ sprints} \times 3 \text{ times} = 9 \text{ sprints}$$

Step 2: Calculate the total meters run per sprint

Each sprint is 60 meters.

Step 3: Calculate the total meters run per week

To find the total meters run per week, multiply the total number of sprints per week by the meters run per sprint:

$$9 \text{ sprints} \times 60 \text{ meters/sprint} = 540 \text{ meters}$$

`<think>` `####` 540

470

LLaMA-3.2-3B-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: Toulouse has twice as many sheep as Charleston. Charleston has 4 times as many sheep as Seattle. How many sheep do Toulouse, Charleston, and Seattle have together if Seattle has 20 sheep? Show your work in `<think>` `</think>` tags. And put the final numerical answer after `####`.

471

Model Response: Let me solve this step by step. <think> First, let's determine the number of sheep Charleston has. Since Charleston has 4 times as many sheep as Seattle, and Seattle has 20 sheep.

So, the number of sheep Charleston has is:

$$4 \times 20 = 80 \text{ sheep}$$

Next, let's determine the number of sheep Toulouse has. Since Toulouse has twice as many sheep as Charleston, and Charleston has 80 sheep.

So, the number of sheep Toulouse has is:

$$2 \times 80 = 160 \text{ sheep}$$

Now, let's add up the number of sheep all three cities have together. Seattle has 20 sheep, Charleston has 80 sheep, and Toulouse has 160 sheep.

So, the total number of sheep is:

$$20 + 80 + 160 = 260 \text{ sheep}$$

<think> #### 260

472

LLaMA-3.1-8B-Instruct

Prompt: You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer. Please solve the following problem: John drives for 3 hours at a speed of 60 mph and then turns around because he realizes he forgot something very important at home. He tries to get home in 4 hours but spends the first 2 hours in standstill traffic. He spends the next half-hour driving at a speed of 30mph, before being able to drive the remaining time of the 4 hours going at 80 mph. How far is he from home at the end of those 4 hours? Show your work in <think> </think> tags. And put the final numerical answer after ####.

Model Response: Let me solve this step by step. <think> Step 1: Calculate the distance John drives for the first 3 hours at 60 mph. Distance = Speed \times Time, so the distance is:

$$60 \text{ mph} \times 3 \text{ hours} = 180 \text{ miles}$$

So, John is 180 miles away from home after driving for 3 hours.

Step 2: Since John turns around and tries to get home in 4 hours, we need to calculate the distance he covers in those 4 hours. However, he spends the first 2 hours in standstill traffic, which means he doesn't move at all during those 2 hours. So, for the first 2 hours, the distance covered is:

$$0 \text{ miles}$$

Step 3: Then, he spends the next half-hour (0.5 hours) driving at a speed of 30 mph.

Distance = Speed \times Time, so the distance covered is:

$$30 \text{ mph} \times 0.5 \text{ hours} = 15 \text{ miles}$$

So, in the next half-hour, he covers 15 miles.

Step 4: After the half-hour of driving at 30 mph, he has 1.5 hours left to reach his goal of getting home in 4 hours. During these remaining 1.5 hours, he drives at a speed of 80 mph. Distance = Speed \times Time, so the distance covered is:

$$80 \text{ mph} \times 1.5 \text{ hours} = 120 \text{ miles}$$

So, in the last 1.5 hours, he covers 120 miles.

Step 5: Now, let's calculate the total distance he covers in the 4 hours to get back home.

Distance covered in the first 2 hours of standstill traffic = 0 miles

Distance covered in the next half-hour at 30 mph = 15 miles

Distance covered in the last 1.5 hours at 80 mph = 120 miles

473

Total distance covered is:

$$0 + 15 + 120 = 135 \text{ miles}$$

Since John was initially 180 miles away from home and he covers a total distance of 135 miles to get back home, we need to calculate the remaining distance:

$$180 \text{ miles} - 135 \text{ miles} = 45 \text{ miles}$$

Therefore, John is 45 miles away from home at the end of those 4 hours. <think> #### 45.

474

475 **NeurIPS Paper Checklist**

476 **1. Claims**

477 Question: Do the main claims made in the abstract and introduction accurately reflect the
478 paper’s contributions and scope?

479 Answer: [Yes]

480 Justification: See abstract and introduction, we present the paper’s contributions and scope.

481 Guidelines:

- 482 • The answer [N/A] means that the abstract and introduction do not include the claims
483 made in the paper.
- 484 • The abstract and/or introduction should clearly state the claims made, including the
485 contributions made in the paper and important assumptions and limitations. A [No] or
486 [N/A] answer to this question will not be perceived well by the reviewers.
- 487 • The claims made should match theoretical and experimental results, and reflect how
488 much the results can be expected to generalize to other settings.
- 489 • It is fine to include aspirational goals as motivation as long as it is clear that these goals
490 are not attained by the paper.

491 **2. Limitations**

492 Question: Does the paper discuss the limitations of the work performed by the authors?

493 Answer: [N/A]

494 Justification: The paper has no limitation.

495 Guidelines:

- 496 • The answer [N/A] means that the paper has no limitation while the answer [No] means
497 that the paper has limitations, but those are not discussed in the paper.
- 498 • The authors are encouraged to create a separate “Limitations” section in their paper.
- 499 • The paper should point out any strong assumptions and how robust the results are to
500 violations of these assumptions (e.g., independence assumptions, noiseless settings,
501 model well-specification, asymptotic approximations only holding locally). The authors
502 should reflect on how these assumptions might be violated in practice and what the
503 implications would be.
- 504 • The authors should reflect on the scope of the claims made, e.g., if the approach was
505 only tested on a few datasets or with a few runs. In general, empirical results often
506 depend on implicit assumptions, which should be articulated.
- 507 • The authors should reflect on the factors that influence the performance of the approach.
508 For example, a facial recognition algorithm may perform poorly when image resolution
509 is low or images are taken in low lighting. Or a speech-to-text system might not be
510 used reliably to provide closed captions for online lectures because it fails to handle
511 technical jargon.
- 512 • The authors should discuss the computational efficiency of the proposed algorithms
513 and how they scale with dataset size.
- 514 • If applicable, the authors should discuss possible limitations of their approach to
515 address problems of privacy and fairness.
- 516 • While the authors might fear that complete honesty about limitations might be used by
517 reviewers as grounds for rejection, a worse outcome might be that reviewers discover
518 limitations that aren’t acknowledged in the paper. The authors should use their best
519 judgment and recognize that individual actions in favor of transparency play an impor-
520 tant role in developing norms that preserve the integrity of the community. Reviewers
521 will be specifically instructed to not penalize honesty concerning limitations.

522 **3. Theory assumptions and proofs**

523 Question: For each theoretical result, does the paper provide the full set of assumptions and
524 a complete (and correct) proof?

525 Answer: [Yes]

526 Justification: The complete set of assumptions and the full and correct proof for the theory
527 in Section 3.3 are provided in the Appendix D.

528 Guidelines:

- 529 • The answer [N/A] means that the paper does not include theoretical results.
- 530 • All the theorems, formulas, and proofs in the paper should be numbered and cross-
531 referenced.
- 532 • All assumptions should be clearly stated or referenced in the statement of any theorems.
- 533 • The proofs can either appear in the main paper or the supplemental material, but if
534 they appear in the supplemental material, the authors are encouraged to provide a short
535 proof sketch to provide intuition.
- 536 • Inversely, any informal proof provided in the core of the paper should be complemented
537 by formal proofs provided in appendix or supplemental material.
- 538 • Theorems and Lemmas that the proof relies upon should be properly referenced.

539 4. Experimental result reproducibility

540 Question: Does the paper fully disclose all the information needed to reproduce the main ex-
541 perimental results of the paper to the extent that it affects the main claims and/or conclusions
542 of the paper (regardless of whether the code and data are provided or not)?

543 Answer: [Yes]

544 Justification: We provide the code link in the abstract and describe the experimental setup in
545 detail in the Appendix E.

546 Guidelines:

- 547 • The answer [N/A] means that the paper does not include experiments.
- 548 • If the paper includes experiments, a [No] answer to this question will not be perceived
549 well by the reviewers: Making the paper reproducible is important, regardless of
550 whether the code and data are provided or not.
- 551 • If the contribution is a dataset and/or model, the authors should describe the steps taken
552 to make their results reproducible or verifiable.
- 553 • Depending on the contribution, reproducibility can be accomplished in various ways.
554 For example, if the contribution is a novel architecture, describing the architecture fully
555 might suffice, or if the contribution is a specific model and empirical evaluation, it may
556 be necessary to either make it possible for others to replicate the model with the same
557 dataset, or provide access to the model. In general, releasing code and data is often
558 one good way to accomplish this, but reproducibility can also be provided via detailed
559 instructions for how to replicate the results, access to a hosted model (e.g., in the case
560 of a large language model), releasing of a model checkpoint, or other means that are
561 appropriate to the research performed.
- 562 • While NeurIPS does not require releasing code, the conference does require all submis-
563 sions to provide some reasonable avenue for reproducibility, which may depend on the
564 nature of the contribution. For example
 - 565 (a) If the contribution is primarily a new algorithm, the paper should make it clear how
566 to reproduce that algorithm.
 - 567 (b) If the contribution is primarily a new model architecture, the paper should describe
568 the architecture clearly and fully.
 - 569 (c) If the contribution is a new model (e.g., a large language model), then there should
570 either be a way to access this model for reproducing the results or a way to reproduce
571 the model (e.g., with an open-source dataset or instructions for how to construct
572 the dataset).
 - 573 (d) We recognize that reproducibility may be tricky in some cases, in which case
574 authors are welcome to describe the particular way they provide for reproducibility.
575 In the case of closed-source models, it may be that access to the model is limited in
576 some way (e.g., to registered users), but it should be possible for other researchers
577 to have some path to reproducing or verifying the results.

578 5. Open access to data and code

579 Question: Does the paper provide open access to the data and code, with sufficient instruc-
580 tions to faithfully reproduce the main experimental results, as described in supplemental
581 material?

582 Answer: [Yes]

583 Justification: We provide an anonymous link to the code used in our experiments in the
584 abstract. The anonymous repository includes the code, data, environment, and running
585 commands needed for reproduction.

586 Guidelines:

- 587 • The answer [N/A] means that paper does not include experiments requiring code.
- 588 • Please see the NeurIPS code and data submission guidelines ([https://neurips.cc/
589 public/guides/CodeSubmissionPolicy](https://neurips.cc/public/guides/CodeSubmissionPolicy)) for more details.
- 590 • While we encourage the release of code and data, we understand that this might not
591 be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not
592 including code, unless this is central to the contribution (e.g., for a new open-source
593 benchmark).
- 594 • The instructions should contain the exact command and environment needed to run to
595 reproduce the results. See the NeurIPS code and data submission guidelines ([https://neurips.cc/
596 //neurips.cc/public/guides/CodeSubmissionPolicy](https://neurips.cc/public/guides/CodeSubmissionPolicy)) for more details.
- 597 • The authors should provide instructions on data access and preparation, including how
598 to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- 599 • The authors should provide scripts to reproduce all experimental results for the new
600 proposed method and baselines. If only a subset of experiments are reproducible, they
601 should state which ones are omitted from the script and why.
- 602 • At submission time, to preserve anonymity, the authors should release anonymized
603 versions (if applicable).
- 604 • Providing as much information as possible in supplemental material (appended to the
605 paper) is recommended, but including URLs to data and code is permitted.

606 6. Experimental setting/details

607 Question: Does the paper specify all the training and test details (e.g., data splits, hyperpa-
608 rameters, how they were chosen, type of optimizer) necessary to understand the results?

609 Answer: [Yes]

610 Justification: We describe the experimental setup in detail in the Appendix E. We also
611 provide the code link in the abstract, and the code includes more experimental details.

612 Guidelines:

- 613 • The answer [N/A] means that the paper does not include experiments.
- 614 • The experimental setting should be presented in the core of the paper to a level of detail
615 that is necessary to appreciate the results and make sense of them.
- 616 • The full details can be provided either with the code, in appendix, or as supplemental
617 material.

618 7. Experiment statistical significance

619 Question: Does the paper report error bars suitably and correctly defined or other appropriate
620 information about the statistical significance of the experiments?

621 Answer: [No]

622 Justification: Due to the high computational cost, we do not report error bars.

623 Guidelines:

- 624 • The answer [N/A] means that the paper does not include experiments.
- 625 • The authors should answer [Yes] if the results are accompanied by error bars, confidence
626 intervals, or statistical significance tests, at least for the experiments that support the
627 main claims of the paper.
- 628 • The factors of variability that the error bars are capturing should be clearly stated (for
629 example, train/test split, initialization, random drawing of some parameter, or overall
630 run with given experimental conditions).

- 631 • The method for calculating the error bars should be explained (closed form formula,
632 call to a library function, bootstrap, etc.)
- 633 • The assumptions made should be given (e.g., Normally distributed errors).
- 634 • It should be clear whether the error bar is the standard deviation or the standard error
635 of the mean.
- 636 • It is OK to report 1-sigma error bars, but one should state it. The authors should
637 preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis
638 of Normality of errors is not verified.
- 639 • For asymmetric distributions, the authors should be careful not to show in tables or
640 figures symmetric error bars that would yield results that are out of range (e.g., negative
641 error rates).
- 642 • If error bars are reported in tables or plots, the authors should explain in the text how
643 they were calculated and reference the corresponding figures or tables in the text.

644 8. Experiments compute resources

645 Question: For each experiment, does the paper provide sufficient information on the com-
646 puter resources (type of compute workers, memory, time of execution) needed to reproduce
647 the experiments?

648 Answer: [Yes]

649 Justification: We provide information about the computational resources used for our
650 experiments in the Appendix E.

651 Guidelines:

- 652 • The answer [N/A] means that the paper does not include experiments.
- 653 • The paper should indicate the type of compute workers CPU or GPU, internal cluster,
654 or cloud provider, including relevant memory and storage.
- 655 • The paper should provide the amount of compute required for each of the individual
656 experimental runs as well as estimate the total compute.
- 657 • The paper should disclose whether the full research project required more compute
658 than the experiments reported in the paper (e.g., preliminary or failed experiments that
659 didn't make it into the paper).

660 9. Code of ethics

661 Question: Does the research conducted in the paper conform, in every respect, with the
662 NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

663 Answer: [Yes]

664 Justification: The research conducted in this paper conforms, in every respect, with the
665 NeurIPS Code of Ethics.

666 Guidelines:

- 667 • The answer [N/A] means that the authors have not reviewed the NeurIPS Code of
668 Ethics.
- 669 • If the authors answer [No], they should explain the special circumstances that require a
670 deviation from the Code of Ethics.
- 671 • The authors should make sure to preserve anonymity (e.g., if there is a special consid-
672 eration due to laws or regulations in their jurisdiction).

673 10. Broader impacts

674 Question: Does the paper discuss both potential positive societal impacts and negative
675 societal impacts of the work performed?

676 Answer: [N/A]

677 Justification: The work has no societal impact. This paper only studies a machine learning
678 algorithm and has no societal impact.

679 Guidelines:

- 680 • The answer [N/A] means that there is no societal impact of the work performed.

- 681
- 682
- 683
- 684
- 685
- 686
- 687
- 688
- 689
- 690
- 691
- 692
- 693
- 694
- 695
- 696
- 697
- 698
- 699
- 700
- 701
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
 - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
 - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
 - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
 - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

702 **11. Safeguards**

703 Question: Does the paper describe safeguards that have been put in place for responsible
704 release of data or models that have a high risk for misuse (e.g., pre-trained language models,
705 image generators, or scraped datasets)?

706 Answer: [N/A]

707 Justification: The paper poses no such risks.

708 Guidelines:

- 709
- 710
- 711
- 712
- 713
- 714
- 715
- 716
- 717
- 718
- The answer [N/A] means that the paper poses no such risks.
 - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
 - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
 - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

719 **12. Licenses for existing assets**

720 Question: Are the creators or original owners of assets (e.g., code, data, models), used in
721 the paper, properly credited and are the license and terms of use explicitly mentioned and
722 properly respected?

723 Answer: [Yes]

724 Justification: The models and datasets used in the paper are properly credited by citing their
725 original papers.

726 Guidelines:

- 727
- 728
- 729
- 730
- 731
- 732
- 733
- The answer [N/A] means that the paper does not use existing assets.
 - The authors should cite the original paper that produced the code package or dataset.
 - The authors should state which version of the asset is used and, if possible, include a URL.
 - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
 - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- 734
- 735
- 736
- 737
- 738
- 739
- 740
- 741
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
 - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
 - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

742 **13. New assets**

743 Question: Are new assets introduced in the paper well documented and is the documentation
744 provided alongside the assets?

745 Answer: [N/A]

746 Justification: The paper does not release new assets.

747 Guidelines:

- 748
- 749
- 750
- 751
- 752
- 753
- 754
- 755
- The answer [N/A] means that the paper does not release new assets.
 - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
 - The paper should discuss whether and how consent was obtained from people whose asset is used.
 - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

756 **14. Crowdsourcing and research with human subjects**

757 Question: For crowdsourcing experiments and research with human subjects, does the paper
758 include the full text of instructions given to participants and screenshots, if applicable, as
759 well as details about compensation (if any)?

760 Answer: [N/A]

761 Justification: The paper does not involve crowdsourcing or research with human subjects.

762 Guidelines:

- 763
- 764
- 765
- 766
- 767
- 768
- 769
- 770
- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
 - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
 - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

771 **15. Institutional review board (IRB) approvals or equivalent for research with human
772 subjects**

773 Question: Does the paper describe potential risks incurred by study participants, whether
774 such risks were disclosed to the subjects, and whether Institutional Review Board (IRB)
775 approvals (or an equivalent approval/review based on the requirements of your country or
776 institution) were obtained?

777 Answer: [N/A]

778 Justification: The paper does not involve crowdsourcing or research with human subjects.

779 Guidelines:

- 780
- 781
- 782
- 783
- 784
- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
 - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are only used for writing, editing, or formatting purposes and do not affect the core methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.