

ESSA: EVOLUTIONARY STRATEGIES FOR SCALABLE ALIGNMENT

000
001
002
003
004
005 **Anonymous authors**
006 Paper under double-blind review
007
008
009
010

ABSTRACT

011 Alignment of Large Language Models (LLMs) typically relies on Reinforcement
012 Learning from Human Feedback (RLHF) with gradient-based optimizers such
013 as Proximal Policy Optimization (PPO) or Group Relative Policy Optimization
014 (GRPO). While effective, these methods require complex distributed training,
015 large memory budgets, and careful hyperparameter tuning, all of which become
016 increasingly difficult at billion-parameter scale. We present ESSA, Evolutionary
017 Strategies for Scalable Alignment, a gradient-free framework that aligns LLMs
018 using only forward inference and black-box optimization. ESSA focuses optimi-
019 zation on Low-Rank Adapters (LoRA) and further compresses their parameter
020 space by optimizing only the singular values from an singular value decomposi-
021 tion (SVD) of each adapter matrix. This dimensionality reduction makes evolu-
022 tionary search practical even for very large models and allows efficient operation
023 in quantized INT4 and INT8 inference mode. Across these benchmarks ESSA im-
024 proves the test accuracy of Qwen2.5-Math-7B by 12.6% on GSM8K and 14.8%
025 on PRM800K, and raises the accuracy of LLaMA3.1-8B on IFEval by 22.5%,
026 all compared with GRPO. In large-scale settings ESSA shows stronger scaling
027 than gradient-based methods: on Qwen2.5-32B for PRM800K it reaches near-
028 optimal accuracy twice as fast on 16 GPUs and six times as fast on 128 GPUs
029 compared with GRPO. These results position evolutionary strategies as a com-
030 pelling, hardware-friendly alternative to gradient-based LLM alignment, combin-
031 ing competitive quality with substantially reduced wall-clock time and engineer-
032 ing overhead.

1 INTRODUCTION

033 Large Language Models (LLMs) have made significant progress thanks to alignment techniques that
034 guide the model’s behavior toward human preferences. Online methods, predominantly Reinforce-
035 ment Learning from Human Feedback (RLHF) with Group Relative Policy Optimization (GRPO),
036 Proximal Policy Optimization (PPO), or REINFORCE Leave-One-Out (RLOO), remain the de facto
037 standard in practice (Ouyang et al., 2022a; Schulman et al., 2017; Ahmadian et al., 2024; Shao et al.,
038 2024). However, these pipelines are complex to implement: they involve actor/critic training, trajec-
039 tory generation, backpropagation through long sequences, and distributed synchronization. As mod-
040 els grow larger, practical considerations about how to distribute components across GPUs become
041 critical, with sensitivity to numerous interacting hyperparameters and communication bottlenecks
042 (Zheng et al., 2023; Sheng et al., 2024).

043 To address these issues, we revisit evolutionary strategies as a scalable, gradient-free alternative.
044 These methods require only forward passes with perturbed parameters and simple aggregation of
045 scalar fitness values, enabling near-parallel training, low memory usage, and robustness to sparse
046 or noisy rewards (Salimans et al., 2017). The classic concern of poor efficiency in very high-
047 dimensional spaces can be addressed through aggressive search space reduction.

048 We introduce **ESSA (Evolutionary Strategies for Scalable Alignment)**, which pairs Evolutionary
049 Strategies (ES) with parameter-efficient adaptation. We limit optimization to low-rank adapters of
050 attention matrices (Q/K/V/O) and further compress them via singular value decomposition (SVD)
051 parameterization of singular values, making black-box search practical and interpretable (Hu et al.
052 (2021), Vaswani et al. (2017)).

054 In this work, ESSA is introduced as a gradient-free online alignment procedure applied strictly after
 055 a supervised fine-tuning (SFT) warm-start. The method replaces only the post-SFT stage. ESSA can
 056 operate fully in quantized INT4 or INT8 inference mode, enabling efficient adaptation of models up
 057 to approximately 72B parameters on a single high-memory GPU Dettmers et al. (2022). Across
 058 mathematics of varying difficulty, instruction following, and general assistant tasks, ESSA matches
 059 baselines trained by GRPO, while offering stronger system scalability and reduced hyperparameter
 060 fragility (Shao et al. (2024)).

061 Taken together, ESSA turns alignment into a simple, highly parallel evaluation loop with minimal
 062 synchronization – an attractive fit for modern clusters and continual training settings – while retaining
 063 the quality expected from state-of-the-art online methods.

065 2 RELATED WORKS

067 **Alignment.** Alignment of large language models is commonly based on RLHF (Ouyang et al.,
 068 2022a), typically optimized with PPO (Schulman et al., 2017) or REINFORCE (Sutton et al., 1999).
 069 Variants such as RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), REINFORCE++ (Hu,
 070 2025) and DAPO (Yu et al., 2025) stabilize training by using relative advantages within groups, yet
 071 still inherit gradient estimation variance and substantial memory cost. Offline preference learning
 072 (Rafailov et al., 2024; Hong et al., 2024; Meng et al., 2024)) removes online rollouts but is bounded
 073 by dataset coverage and preference noise, limiting generalization (Tang et al., 2024; Xu et al., 2024).

074 **Parameter-efficient training of LLMs.** Parameter-efficient fine-tuning (PEFT) techniques reduce
 075 the cost of adapting large language models by updating only a small subset of parameters. Beyond
 076 classic approaches such as adapters (Houlsby et al., 2019), prefix-tuning (Li & Liang, 2021), and
 077 LoRA (Hu et al., 2021), more recent methods include DoRA (Liu et al., 2024), VeRA (Kopczko
 078 et al., 2024), and tensor-based approaches like LoTR (Bershatsky et al., 2024). Another way to
 079 reduce the number of trainable parameters is to optimize only the eigenvalues in the SVD decomposi-
 080 tion of transformer matrices, as done in Transformer² (Sun et al., 2025), which also inspired our
 081 method.

082 **Evolution strategies.** ES, including Covariance Matrix Adaptation (CMA-ES) (Hansen & Os-
 083 termeier, 2001), Natural Evolution Strategies (NES) (Wierstra et al., 2011), Augmented Random
 084 Search (ARS) (Mania et al., 2018), and Guided Evolutionary Strategies (GES) (Maheswaranathan
 085 et al., 2018), provide powerful gradient-free optimization that is highly parallel and robust to sparse
 086 rewards (Salimans et al., 2017). Zero-order optimizers (Zhang et al., 2024) approximate gradi-
 087 ents of loss differences but tend to lag in complex reasoning tasks. Applications of ES to large
 088 language model alignment remain rare due to high-dimensionality challenges: existing works like
 089 GENOME/GENOME+ (Zhang et al., 2025), LoRAHub (Huang et al., 2024), and DFO (Jin et al.,
 090 2024) reduce parameter space but are still limited to a variety of experiments.

091 3 EVOLUTIONARY STRATEGIES FOR SCALABLE ALIGNMENT

092 3.1 MOTIVATION

095 Modern online alignment of LLMs is dominated by gradient-based RLHF variants. In practice, these
 096 pipelines are costly: they require long rollouts, backpropagation through large contexts, optimizer-
 097 state synchronization across devices, and careful hyperparameter tuning. As model size grows,
 098 memory pressure and training fragility increase, and sparse or noisy rewards further destabilize
 099 learning.

101 ESSA offer the opposite trade-off. ESSA restricts learning to low-rank LoRA adapters represented
 102 in a compact SVD parameterization, where a few singular values is optimized. In this setting, evolu-
 103 tionary updates require nothing beyond forward evaluations under parameter perturbations together
 104 with aggregated scalar rewards, which yields a naturally parallelizable, memory-efficient training
 105 loop. The standard drawback of evolutionary methods, namely poor efficiency in extremely high-
 106 dimensional search spaces, is addressed here by forcing the search to remain within a compact and
 107 task-aligned low-rank subspace.

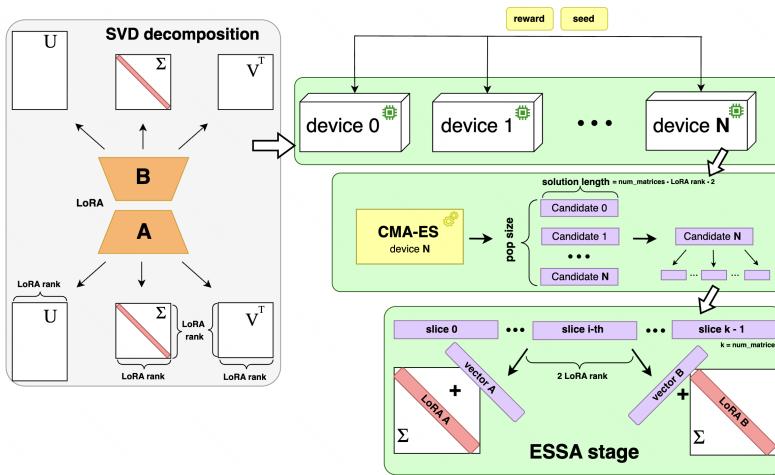


Figure 1: Illustration of the ESSA framework. LoRA adapters are first initialized via SFT and decomposed into fixed SVD bases with trainable singular values. The term device N denotes the GPU worker in distributed evaluation. CMA-ES receives a seed at each device, generates a population of size $N+1$ locally, evaluates a different candidate, and returns a reward. Each candidate is a vector $\sigma_i \in \mathbb{R}^{\text{solution_length}}$, $i = 0, 1, \dots, N$ and is added to SVD vectors of the training matrices. It is partitioned into contiguous slices, each of which corresponds to one LoRA matrix (e.g. W_Q, W_K, W_V, W_O for each transformer layer) and contains $2 \cdot \text{LoRA_rank}$ singular values (for factors A and B). The solution length is the dimensionality of the candidate vector, i.e., the concatenation of all perturbation of the trainable LoRA singular values across all matrices and layers. With the number of layers (**num_layers**), number of matrices per layer (**num_matrices_per_layer**), LoRA_rank: $\text{num_matrices} = \text{num_layers} \cdot \text{num_matrices_per_layer}$ and $\text{solution_length} = \text{num_matrices} \cdot \text{LoRA_rank} \cdot 2$.

Since the loop is inference-only, ESSA runs efficiently under low-precision (INT4/INT8) inference. It also scales linearly with available hardware by evaluating candidates in parallel with minimal communication (seed + reward) (Salimans et al., 2017). This design not only yields a compact gradient-free search space but also addresses the practical problem of optimizing the alignment stage in large-scale, multi-GPU training pipelines. ESSA integrates established components (LoRA, SVD compression, evolutionary search) into a unified framework that enables inference-only, quantized, distributed alignment with minimal engineering and synchronization overhead.

Crucially, ESSA is not a full training pipeline: it is a gradient-free alignment stage applied on top of a SFT initialization. The SFT warm-start preserves model expressivity and places ES in a task-aware region of parameter space. Without SFT, the LoRA matrices begin as an unstructured Gaussian matrix paired with a zero matrix, whose singular vectors are effectively identity-based and uninformed. We demonstrate the importance of SFT stage using a toy example on MNIST (LeCun & Cortes, 2005) in Appendix E. SFT is also inexpensive compared to online RL alignment. Prior work reports that SFT requires only a small fraction of the compute used for RL-based alignment (Ouyang et al., 2022b). The expensive component of alignment is the online RL loop, and ESSA eliminates this phase while preserving the lightweight warm-start. Experiments without SFT are provided in Appendix F.1 and demonstrate that ESSA can operate without SFT, while being primarily intended as a gradient-free replacement for the online alignment phase.

3.2 METHOD

Initialization. First, we reduce optimization problem to low-rank updates of each attention projection matrix $W_0 \in \mathbb{R}^{m \times n}$:

$$\Delta W = BA, \quad B \in \mathbb{R}^{m \times r}, A \in \mathbb{R}^{r \times n}, r \ll \min(m, n).$$

Then, we run a short SFT stage to initialize the LoRA adapters with task-aware parameters. For every backbone-task pair in the paper, we train exactly one SFT LoRA adapter.

162 **SVD.** To shrink the trainable space even further, we decompose each SFT-LoRA factor separately:
 163

$$164 \quad A = U_A \Sigma_A V_A^\top, \quad B = U_B \Sigma_B V_B^\top.$$

165 The orthogonal matrices U_A, V_A, U_B, V_B are kept fixed after the initial SFT step, while only the
 166 top singular values in Σ_A and Σ_B remain trainable. This SVD-LoRA representation preserves the
 167 expressive power of LoRA while reducing the number of variables that ES must explore.
 168

169 **Evolutionary Optimization.** We use CMA-ES (Hansen & Ostermeier, 2001) as the optimizer,
 170 maintaining and updating a multivariate normal search distribution over the selected singular values.
 171 Each ESSA iteration proceeds as follows (more detailed algorithm is provided in Figure 1 and in
 172 Appendix A):
 173

- 174 1. CMA-ES samples $\lambda \geq 2$ (population size) candidate singular-value vectors.
 175 $x_{i+1}^{(k)} \sim m_i + \sigma_i \mathcal{N}(0, C_i)$ for $k = 1, \dots, \lambda$, so that $x_{i+1}^{(k)} \sim \mathcal{N}(m_i, \sigma_i^2 C_i)$,
 176 where $m_i \in \mathcal{R}^n$ is a current mean of the search distribution; $\sigma_i \in \mathcal{R} > 0$ is an “overall”
 177 standard deviation; $C_i \in \mathcal{R}^{n \times n}$ is a covariance matrix encoding anisotropic search direc-
 178 tions; $n = \text{solution_length}$. Up to the scalar factor σ_i^2 , C_i is the covariance matrix of the
 179 search distribution
- 180 2. For each candidate, we reconstruct A and B by adding the candidate’s singular-value offsets
 181 to the fixed SVD decomposition, forming updated low-rank factors and computing $\Delta W =$
 182 BA , and evaluate the model on the alignment objective to obtain a scalar reward.
- 183 3. After all candidates are evaluated in parallel, CMA-ES updates m_i , σ_i , and C_i . This allows
 184 the search distribution to gradually align itself with beneficial directions in the objective
 185 landscape. Communication between workers is limited to random seeds and scalar rewards,
 186 allowing near-linear scaling across many GPUs.

188 3.3 THEORETICAL ANALYSIS

190 We compare the per-iteration latency of gradient-based online alignment (e.g.,
 191 RLHF/GRPO/PPO/RLOO) with a single ESSA update, taking into account both computa-
 192 tion and inter-GPU communication. The key observation is that gradient methods require expensive
 193 all-reduce of model-size gradients, whereas ESSA communicates only a random seed and scalar
 194 rewards. Consequently, the communication cost of ESSA scales essentially independently of model
 195 size.

196 A simple latency model shows that for any realistic cluster bandwidth, there exists a conservative
 197 population size threshold such that when the ESSA population size multiplied by the batch size
 198 processed by the single population instance $N_{\text{pop}} B^{\text{essa}}$ is below this bound, ESSA achieves a strictly
 199 lower per-iteration time than the idealized gradient pipeline. Moreover, because real clusters seldom
 200 achieve perfect device splitting or peak network bandwidth, the practical speed advantage of ESSA
 201 is typically even larger.

202 Full mathematical notation, the precise expressions for computation and communication time, and
 203 the formal proof of the population size bound are provided in Appendix B.

205 4 EXPERIMENTS

207 **Tasks and Models.** We evaluate ESSA on three categories of alignment workloads: (i) School-
 208 level math reasoning. We train and evaluate on GSM8K with accuracy as the primary metric. Back-
 209 bones: Qwen2.5-7B, Qwen2.5-Math-7B. (ii) Advanced math reasoning. We train on PRM800K and
 210 evaluate on MATH500, AIME’24/’25, MinervaMath, and OlympiadBench using pass@k / avg@16.
 211 Backbones: Qwen2.5-Math-7B, Qwen2.5-32B, Qwen2.5-72B. (iii) Instruction following. We train
 212 on an if-eval-like dataset and evaluate on IFEval. Backbones: Llama-3.1-8B. (iv) General-Purpose
 213 Assistant. We train and evaluate on HelpSteer2. Backbones: Llama-3.1-8B.

214 **Baseline.** We use GRPO as a baseline. GRPO is the standard and most robust online RLHF method
 215 widely used in both industry and open-source practice (Xi et al., 2025; DeepSeek-AI et al., 2025;

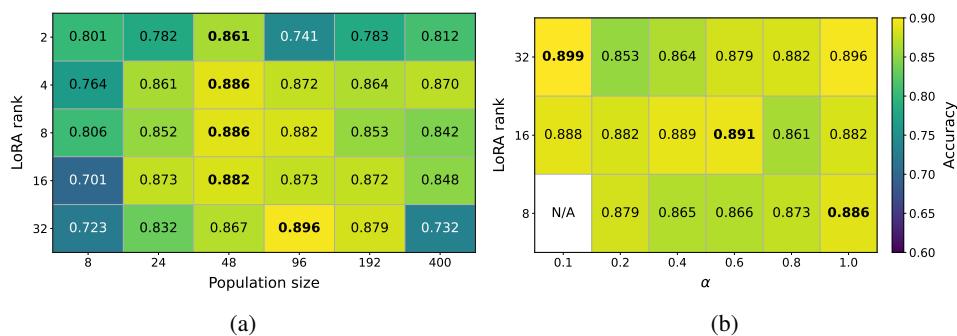
216 Yang et al., 2025), making it the natural baseline for comparison. Our goal with ESSA is not to
 217 outperform specialized gradient optimizers, but to evaluate whether a gradient-free approach can
 218 match the quality of GRPO-based RLHF while operating purely in inference mode.
 219

220 **SFT.** We start from SFT checkpoints appropriate for each dataset. Experiments without SFT are
 221 provided in Appendix F.1. We do not include the time spent on the SFT in all the presented results.
 222 It is presented separately in Appendix H.2. Since ESSA and GRPO use the same SFT checkpoint
 223 within each backbone-task pair, the SFT cost contributes an identical constant to both methods and
 224 does not affect their relative performance.
 225

226 **Other Details.** We use standard BFLOAT16 precision unless otherwise specified. Qwen2.5-72B
 227 is also trained under INT4 for per-device evaluation in ESSA. For training the models with the
 228 GRPO algorithm, we use the verl library (Sheng et al., 2024), which the authors describe as the
 229 most efficient in terms of model allocation and interaction speed. All experiments, unless otherwise
 230 noted, are conducted on 8 GPUs. For larger models Qwen2.5-32B and Qwen2.5-72B we use 16
 231 and 32 GPUs, respectively. The sizes of the training and validation datasets, as well as some other
 232 details of the experiments, are given in Appendix H.
 233

234 4.1 SENSITIVITY TO HYPERPARAMETERS

235 **ESSA hyperparameter sensitivity.** We investigate how ESSA accuracy depends on its key hy-
 236 perparameters – LoRA rank, population size, and the fraction α of singular values optimized in each
 237 SVD factor. A full grid search is performed on five settings Qwen2.5-7B and Qwen2.5-Math-7B on
 238 GSM8K and PRM800K, and LLaMA-3.1-8B on IFEval. Figure 2 shows the results for Qwen2.5-
 239 Math-7B on GSM8K as a representative example.
 240



253 Figure 2: Hyperparameter sensitivity of ESSA on **Qwen2.5-Math-7B** for **GSM8K**. Batch size 100.
 254 (a) Accuracy when varying LoRA rank and population size. (b) For each LoRA rank, the population
 255 size is fixed to the best value found in (a), while the percentage α of trainable singular values is
 256 varied. This illustrates how ESSA performance depends jointly on adapter rank and the fraction
 257 of singular values optimized. The single white cell occurs because for LoRA rank 8 and $\alpha = 0.1$,
 258 rounding down yields zero trainable singular values, so no valid accuracy is reported.
 259

260 This example highlights three consistent trends observed across all tasks: (i) increasing LoRA rank
 261 beyond moderate values does not necessarily improve accuracy, and in some PRM800K setups the
 262 best results occur with rank as low as 2; (ii) very small populations (≈ 8) underperform, while the
 263 benefit of larger populations levels off between about 24 and 96; (iii) for the fraction α of trainable
 264 singular values, accuracy remains stable once α reaches moderate levels (≥ 0.4), showing that ESSA
 265 can achieve its best quality without updating all singular values. The complete set of sensitivity plots
 266 for the remaining four tasks, which confirm these conclusions and show the cases where rank 2 is
 267 optimal, are provided in Appendix C.
 268

269 **Effect of SFT initialization.** We also examine how the maximum ESSA accuracy depends on the
 270 quality of the initial LoRA matrices A and B , which are obtained from the SFT stage.

270 To vary the initialization quality, we train the SFT model on different fractions of the
 271 GSM8K dataset and then run ESSA with identical hyperparameters. The setup is as follows:
 272 Qwen2.5-Math-7B, LoRA rank 16, population size 192, and $\alpha = 1.0$. Table 1 reports the final
 273 ESSA accuracy as a function of the percentage of SFT data used.
 274

275 The results show that reducing the SFT dataset from 100% to only 5% lowers the final ESSA accuracy
 276 by more than 15 percentage points. Thus, the quality of the initial supervised fine-tuning
 277 plays a key role in the ultimate performance of the aligned model. We also conduct an experiment,
 278 running the SFT on one domain and training ESSA on another. The results demonstrate ESSA’s
 279 robustness to the OOD shift in Appendix G.
 280

281 4.2 PARALLELIZATION

282 Figure 3 compares the wall-clock time required by GRPO and ESSA to reach a fixed test accuracy
 283 of 0.835 on Qwen2.5-32B for the PRM800K benchmark, when training is distributed across an
 284 increasing number of GPUs. For ESSA we use a LoRA rank 16 and an population size 128. For
 285 GRPO we match the LoRA rank (16) and use a learning rate 1×10^{-5} , global batch size 512, and
 286 mini batch size 64.
 287

288 As the number of GPUs grows from 16 to 128, both methods benefit from additional parallelism,
 289 but the gains differ substantially: GRPO decreases from nearly 400 minutes at 16 GPUs to roughly 150 minutes at 128 GPUs ($\approx 2.6 \times$
 290 speed-up); ESSA drops from about 200 minutes to under 20 minutes over the same range
 291 ($\approx 10 \times$ speed-up). On the smallest cluster (16 GPUs) ESSA is already about twice as fast as
 292 GRPO. This gap grows with increasing parallelism and reaches a factor of roughly six on
 293 128 GPUs.
 294

295 ESSA’s scaling advantage follows directly from its inference-only optimization loop: evaluation
 296 of population members is embarrassingly parallel and requires communication of only random
 297 seeds and scalar rewards. In contrast, GRPO must synchronize large gradient tensors each step,
 298 leading to communication bottlenecks that limit scaling efficiency.
 299

300 We also conduct an experiment on a single GPU using the smaller Qwen2.5-3B model, where no
 301 parallelism is available. The result is presented in Appendix F.1. This experiment demonstrate that
 302 despite the limited hardware ESSA converges faster than GRPO in this setting as well, while the
 303 primary goal of this work is to introduce a method that utilizes multi-GPU resources more efficiently
 304 than gradient-based training.
 305

306 4.3 PRECISION ANALYSIS

307 Because ESSA uses the model purely in inference mode, it can be trained even when the underlying
 308 model weights are quantized. We evaluate this capability on Qwen2.5-32B trained on
 309 PRM800K with LoRA rank 8, population size 64 and $\alpha = 1.0$. Table 2 reports the best validation
 310 accuracy achieved during training for three numerical precisions. Full training curves are
 311 provided in Appendix D.
 312

SFT dataset fraction	5%	25%	50%	75%	100%
Max ESSA accuracy	0.713	0.731	0.807	0.863	0.872

Table 1: ESSA validation maximum accuracy as a function of **GSM8k** dataset fraction used to initialize **Qwen2.5-Math-7B** LoRA. Settings: LoRA rank 16, pop. 192, batch size 100, $\alpha = 1.0$.

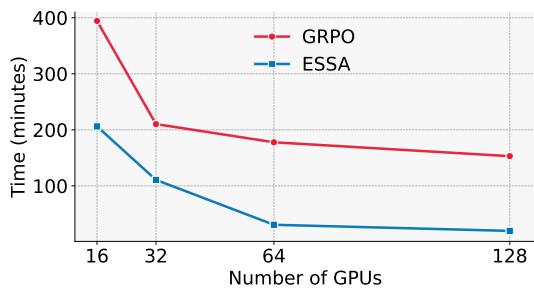


Figure 3: GRPO and ESSA scaling on **PRM800K** with **Qwen2.5-32B**: time to reach 0.835 accuracy vs. GPU count. ESSA: LoRA rank 16, pop. 128, batch size 256, $\alpha = 1.0$. GRPO: LoRA rank 16, lr 1×10^{-5} , global batch 512, mini batch 64.

Precision	BFLOAT16	INT8	INT4
Max ESSA accuracy	0.847	0.844	0.838

Table 2: ESSA validation maximum accuracy as a function of **Qwen2.5-32B** weight precision on PRM800K. Settings: LoRA rank 8, pop. 64, $\alpha = 1.0$, batch size 256.

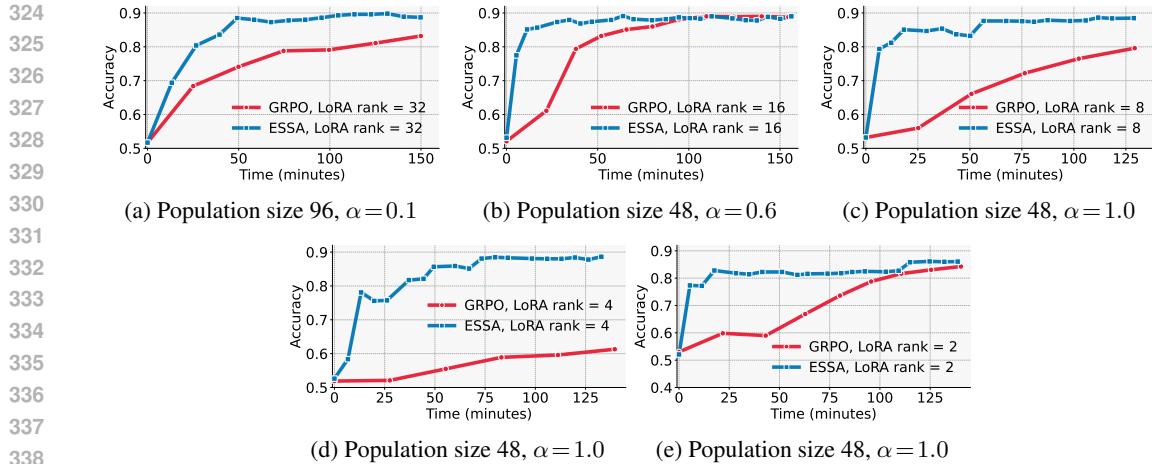


Figure 4: Validation accuracy over time on **GSM8K** with **Qwen2.5-Math-7B**. Panels (a)-(e) correspond to LoRA ranks 32, 16, 8, 4, and 2, respectively. ESSA (blue): batch size 100. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64. ESSA rises rapidly and plateaus early across all ranks, while GRPO improves more gradually.

Although the maximum accuracy decreases slightly as precision is reduced, the drop is minor (less than one percentage point from BFLOAT16 to INT4). This enables substantial savings in compute and memory: with INT4 quantization, a model as large as 72B parameters can fit on a single GPU for processing each population member, as demonstrated in our large-scale experiments later in the paper.

4.4 COMPARISON TO BASELINE (GRPO)

4.4.1 SCHOOL MATH

Figure 4 shows the validation accuracy versus wall-clock time for Qwen2.5-Math-7B on the GSM8K benchmark, comparing ESSA with GRPO across different LoRA ranks. Across all ranks, ESSA rises sharply during the first 10-20 minutes and reaches accuracies near 0.85-0.90 significantly earlier than GRPO, which typically requires 60-100 minutes to approach the same level.

For moderate and large ranks (32, 16) both methods eventually converge to a similar final accuracy (≈ 0.88 -0.90), but ESSA attains this plateau far sooner and with less fluctuation. At smaller ranks ESSA maintains accuracy close to its high-rank plateau, while GRPO lags behind for most of training. The trajectories demonstrate that ESSA is considerably less sensitive to LoRA rank: lowering the rank from 32 to 2 has only a mild effect on both early-time growth and final accuracy, whereas GRPO’s convergence speed degrades markedly as rank decreases.

On school-level math reasoning tasks, ESSA consistently delivers a faster time-to-quality than GRPO across all LoRA ranks, making it well suited for rapid iteration or training under tight time budgets. For completeness, the same comparison performed with the Qwen2.5-7B model on GSM8K is reported in Appendix F.1. Those curves confirm the same pattern: ESSA consistently converges faster than GRPO while matching final accuracy.

4.4.2 BEYOND SCHOOL MATH

Advanced-math benchmarks require multi-step symbolic reasoning, long derivations, and careful numeric manipulation. To faithfully measure the benefit of alignment strategies we therefore employ large backbones, Qwen2.5-32B and Qwen2.5-72B, which (i) possess enough raw capacity to tackle these high-difficulty problems and (ii) let us observe how training methods scale with model size.

Figure 5 illustrate the validation accuracy over time for Qwen2.5-32B and Qwen2.5-72B, respectively. Across both models the curves show the same pattern as on school-math: ESSA rises sharply early, reaches its plateau considerably sooner than GRPO, and maintains that level with low variance.

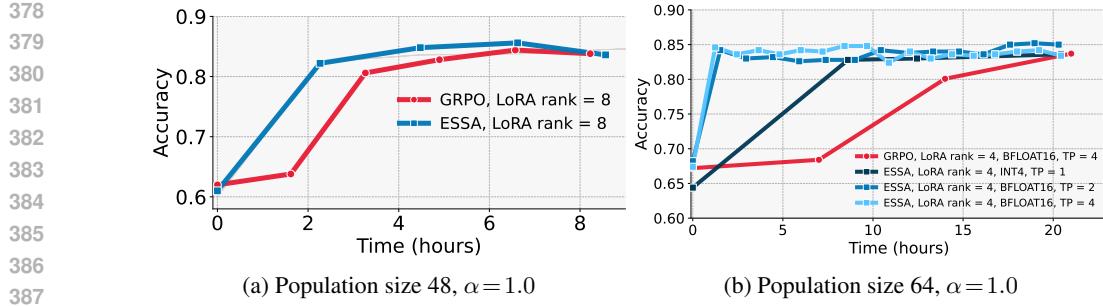


Figure 5: Validation accuracy over time on **PRM800K**. **Qwen2.5-32B** with LoRA rank 8 for both methods (a) and **Qwen2.5-72B** with LoRA rank 4 for both methods (b). For Qwen2.5-72B we run ESSA under BFLOAT16 with tensor parallelism (TP): $TP = 2$ and $TP = 4$, and under INT4 with $TP = 1$, keeping the total GPU budget at 32 for both methods. ESSA (blue): batch size 256. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64. Across both scales, ESSA reaches strong validation accuracy earlier and matches or exceeds GRPO throughout.

Method	MATH500	MinervaMath	OlympiadBench	AIME'24	AIME'25	AMC'23	Avg
avg@k							
GRPO	81.8	41.2	45.7	14.6	10.0	61.7	42.5
ESSA	82.1	41.8	47.6	17.3	12.1	63.3	44.0
pass@8							
GRPO	93.6	53.3	65.3	32.9	29.2	87.3	60.2
ESSA	92.6	52.2	66.4	35.3	28.0	86.9	60.2

Table 3: Results on advanced-math benchmarks with **Qwen2.5-32B**. Rows are grouped by metric: avg@k and pass@8. For **AIME'24**, **AIME'25**, and **AMC'23**, avg@k is reported at $k = 16$; for other benchmarks, $k = 8$. The final Avg averages across available entries.

Method	MATH500	MinervaMath	OlympiadBench	AIME'24	AIME'25	AMC'23	Avg
avg@k							
GRPO	83.0	43.0	49.1	18.8	11.0	62.2	44.5
ESSA	82.8	43.5	48.4	17.9	13.3	62.5	44.7
pass@8							
GRPO	94.2	51.8	66.7	37.5	23.5	85.6	59.9
ESSA	93.0	52.9	66.8	37.4	26.8	85.8	60.5

Table 4: Results on advanced-math benchmarks with **Qwen2.5-72B**. Rows are grouped by metric: avg@k and pass@8. For **AIME'24**, **AIME'25**, and **AMC'23**, avg@k is reported at $k = 16$; for other benchmarks, $k = 8$. The final Avg averages across available entries.

In total, we allocate 16 and 32 GPUs to the 32B and 72B models, respectively, across all configurations. In the BFLOAT16 precision regime the 72B model no longer fits on a single GPU, so we use tensor parallelism (TP) with size of 2 and 4. GRPO requires at least 4 GPUs for this configuration because both forward and backward passes must be distributed. In contrast, ESSA can operate with INT4 weights, allowing a full 72B model instance to reside on a single GPU and enabling one-candidate-per-GPU evaluation.

We observe that the BFLOAT16 model converges faster than the INT4 version. This happens because placing a 72B model in an INT4 representation leaves less space for KV-cache and INT4 matrix multiplications are slower on standard accelerators. Crucially, increasing tensor parallelism shows minimal impact on the accuracy growth trajectory in this experiment. This indicates that, even when the inference worker does not fit on a single device, that is, when the per-device evaluation rule is broken, ESSA remains efficient.

We also evaluate the trained models on the advanced-math benchmarks using both ESSA and GRPO. The detailed scores are reported in Table 3 for Qwen2.5-32B and in Table 4 for Qwen2.5-72B. For Qwen2.5-32B, ESSA improves or matches GRPO on avg@k across all benchmarks and shows comparable pass@8 and avg@k performance. For Qwen2.5-72B, where the tasks are especially challenging, ESSA again achieves competitive or better results: it delivers slightly higher pass@8 and avg@k.

4.4.3 INSTRUCTION FOLLOWING

We next evaluate ESSA in a domain that is qualitatively different from mathematics: instruction following.

Here the goal is to align the model to follow natural language instructions rather than to perform structured reasoning. We use the IFEval benchmark with LLaMA3.1-8B and fix the LoRA rank to 8, comparing ESSA and GRPO under identical data and initialization.

Figure 6 shows that ESSA reaches roughly 0.6 accuracy within the first 60 minutes and maintains that level for the remainder of training. GRPO, in contrast, exhibits a much slower and steadier increase, saturating around 0.45 even after more than four hours. Although the ESSA trajectory displays higher short-term variance, its early and sustained advantage demonstrates that gradient-free evolutionary optimization is effective even in open-ended, non-mathematical instruction-following tasks.

4.4.4 GENERAL-PURPOSE ASSISTANT SETUP

To evaluate alignment in a general-purpose assistant scenario, we employ a more sophisticated reward model rather than simple verifiable rewards. Specifically, we use the RLHFlow/ArmoRM-Llama3-8B-v0.1(Wang et al., 2024a)¹ reward model to provide nuanced, preference-based feedback signals.

The instruction prompts for this setting were drawn from the HelpSteer2 (Wang et al., 2024b) dataset, which contains diverse user instructions and serves as a strong benchmark for open-domain assistant alignment. ESSA and GRPO are both trained with this preference-based reward signal and evaluated under the same protocol to assess their ability to align large language models to general-purpose assistant behavior.

Figure 7 reports the validation reward as a function of wall-clock time for LoRA ranks 8, 16, and 32. Across all ranks, ESSA and GRPO achieve similar final reward levels, with their learning curves showing comparable overall trends.

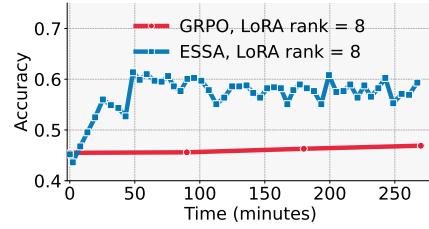


Figure 6: Validation accuracy over time on **IFEval** with **LLaMA-3.1-8B**. ESSA (blue): LoRA rank 8, pop. 24, batch size 500, $\alpha = 1.0$. GRPO (red): LoRA rank 8, lr 1×10^{-5} , global batch 512, mini batch 64. ESSA improves around 0.6-0.65, while GRPO remains nearly flat near 0.45 throughout training.

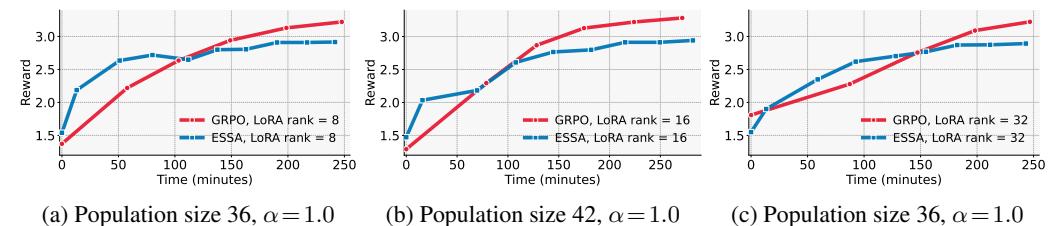


Figure 7: Validation reward over time on **HelpSteer2** with **LLaMA-3.1-8B**. Panels (a)-(c) correspond to LoRA ranks 8, 16, 32 respectively. ESSA (blue): batch size 100. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64. Both methods improve steadily, with ESSA showing faster early gains and GRPO slightly higher final reward. Overall, the plots indicate that ESSA achieves comparable alignment quality to GRPO with lower training complexity and similar convergence behavior.

¹<https://huggingface.co/RLHFlow/ArmoRM-Llama3-8B-v0.1>

486 4.4.5 SVD-LoRA IN GRPO
487

488 In all previous experiments the LoRA factors A and B were fully trainable in GRPO, whereas ESSA
489 optimized only the singular values after an SVD decomposition. To make the comparison strictly
490 fair we repeat the GRPO baseline with the same restriction: only the singular values of the SVD of
491 A and B are updated while their singular vectors were kept fixed, so that both methods have exactly
492 the same number of trainable parameters. This SVD-GRPO variant is evaluated on Qwen2.5-7B
493 with PRM800K, sweeping LoRA ranks 16,8,4,2 and keeping all other GRPO hyperparameters and
494 the SFT initialization identical to the main baseline.

495 Figure 8 shows that SVD-GRPO struggles to learn effectively: even at rank 16 it plateaus around
496 0.5 accuracy and degrades further as the rank decreases. By contrast, ESSA with only rank 2 rapidly
497 reaches about 0.72 accuracy and remains stable, outperforming SVD-GRPO by more than twenty
498 percentage points despite operating in an equally low-dimensional parameter space.

499 This phenomenon can be explained. Deep networks
500 typically exhibit dense, rotated curvature of the
501 objective function, but in high-dimensional parameter-
502izations such as full LoRA, these interactions are
503 spread across many weakly correlated coordinates.
504 Each parameter sees only a small fraction of the
505 curvature. After SVD compression, the same curvature
506 is concentrated into a very small subspace, and
507 diagonal first-order optimizers like Adam fail to make
508 progress. ES, by contrast, never computes gradients,
509 its updates depend only on scalar rewards. Moreover
510 CMA-ES adapts its covariance matrix in a way that
511 aligns, in expectation, with the inverse Hessian of
512 the objective function.

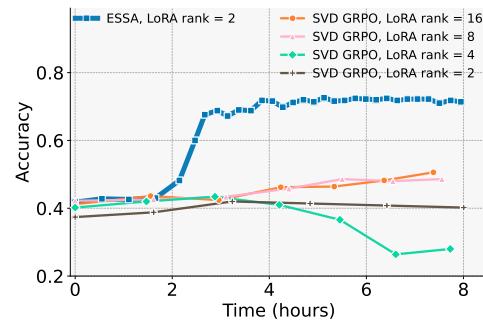
513 Thus, for ESSA, moving from direct LoRA pa-
514 rameters to SVD-LoRA simply reduces dimension-
515 ality without making the search problem harder.
516 For GRPO, however, the same reparameterization
517 changes both the curvature and the gradient statis-
518 tics in a way that standard first-order updates are not
519 well adapted to.

520 5 DISCUSSION
521

522 ESSA shows that scalable and efficient LLM align-
523 ment is possible without gradients or backward
524 passes, relying instead on inference-only evolution-
525 ary search in a compact, hardware-friendly pa-
526 rameter space. Across advanced math and instruc-
527 tion-following benchmarks, ESSA consistently
528 matches or outperforms state-of-the-art gradient-based approaches such as GRPO, while delivering
529 faster time-to-quality, greater robustness to hyperparameters, and dramatically reduced engineering
530 complexity.

531 Our theoretical analysis further supports these empirical findings, demonstrating that ESSA’s itera-
532 tion time and parallel efficiency scale substantially better with model and cluster size, thanks to min-
533 imal synchronization and communication overhead. The ability to operate natively in low-precision
534 (INT8/INT4) mode enables alignment of very large models – up to 72B parameters – using only a
535 single GPU per candidate, with negligible accuracy loss. These results position evolutionary strate-
536 gies, when paired with parameter-efficient adaptation, as a compelling alternative to classic RLHF
537 pipelines – offering a simple, scalable, and broadly applicable framework for LLM alignment.

538 **Limitations & Future Work.** ESSA still depends on a decent SFT warm-start and is ultimately
539 bounded by the expressivity of fixed-rank LoRA; with weak seeds or tiny ranks accuracy can plateau
early. Very large populations also raise total FLOPs even though communication stays cheap. Future
work will explore hybrid ES-gradient phases, adaptive rank expansion, and fully on-device / feder-
ated evolution in which edge GPUs or phones evaluate candidates and return only scalar rewards.



520 Figure 8: Validation accuracy over time on
521 **PRM800K with Qwen2.5-7B.** ESSA (blue):
522 LoRA rank 2, pop. 24, batch size 300, $\alpha =$
523 1.0. SVD-GRPO (only singular values are
524 updated): LoRA ranks 16/8/4/2, lr 1×10^{-2} ,
525 global batch 512, mini batch 64. ESSA
526 achieves ≈ 0.72 accuracy while SVD-GRPO
527 saturates at or below 0.5 even at LoRA rank
528 16.

540 REFERENCES
541

542 Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin,
543 Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learn-
544 ing from human feedback in llms, 2024. URL <https://arxiv.org/abs/2402.14740>.

545 Daniel Bershatsky, Daria Cherniuk, Talgat Daulbaev, Aleksandr Mikhalev, and Ivan Oseledets. Lotr:
546 Low tensor rank weight adaptation, 2024. URL <https://arxiv.org/abs/2402.01376>.

547 DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu,
548 Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu,
549 Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao
550 Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan,
551 Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao,
552 Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding,
553 Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang
554 Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai
555 Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang,
556 Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang,
557 Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang,
558 Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang,
559 R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhua Chen, Shengfeng
560 Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing
561 Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen
562 Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong
563 Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu,
564 Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xi-
565 aoshua Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia
566 Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng
567 Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong
568 Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong,
569 Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou,
570 Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying
571 Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda
572 Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu,
573 Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu
574 Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforce-
575 ment learning, 2025. URL <https://arxiv.org/abs/2501.12948>.

576 Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multi-
577 plication for transformers at scale. *arXiv preprint arXiv:2208.07339*, 2022.

578 Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution
579 strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398.

580 Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without
581 reference model, 2024. URL <https://arxiv.org/abs/2403.07691>.

582 Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, An-
583 drea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp,
584 2019. URL <https://arxiv.org/abs/1902.00751>.

585 Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang,
586 and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.

587 Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv
588 preprint arXiv:2501.03262*, 2025.

589 Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. Lo-
590 rahub: Efficient cross-task generalization via dynamic lora composition, 2024. URL <https://arxiv.org/abs/2307.13269>.

594 Feihu Jin, Yin Liu, and Ying Tan. Derivative-free optimization for low-rank adaptation in large
 595 language models, 2024. URL <https://arxiv.org/abs/2403.01754>.

596

597 Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. Vera: Vector-based random matrix
 598 adaptation, 2024. URL <https://arxiv.org/abs/2310.11454>.

599

600 Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005. URL <https://api.semanticscholar.org/CorpusID:60282629>.

601

602 Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.
 603 URL <https://arxiv.org/abs/2101.00190>.

604

605 Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-
 606 Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation, 2024. URL
 607 <https://arxiv.org/abs/2402.09353>.

608

609 Niru Maheswaranathan, Luke Metz, Dami Choi, George Tucker, and Jascha Sohl-Dickstein. Guided
 610 evolutionary strategies: escaping the curse of dimensionality in random search. 2018. URL
<https://arxiv.org/abs/1806.10230>.

611

612 Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive
 613 approach to reinforcement learning, 2018. URL <https://arxiv.org/abs/1803.07055>.

614

615 Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a
 616 reference-free reward, 2024. URL <https://arxiv.org/abs/2405.14734>.

617

618 Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong
 619 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kel-
 620 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike,
 621 and Ryan Lowe. Training language models to follow instructions with human feedback, 2022a.
 622 URL <https://arxiv.org/abs/2203.02155>.

623

624 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
 625 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kel-
 626 ton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike,
 627 and Ryan Lowe. Training language models to follow instructions with human feedback. In
 628 S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in
 629 Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc.,
 630 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

631

632 Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and
 633 Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model,
 634 2024. URL <https://arxiv.org/abs/2305.18290>.

635

636 Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as
 637 a scalable alternative to reinforcement learning, 2017. URL <https://arxiv.org/abs/1703.03864>.

638

639 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
 640 optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.

641

642 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 643 Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathe-
 644 matical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.

645

646 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
 647 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint
 arXiv: 2409.19256*, 2024.

648

649 Qi Sun, Edoardo Cetin, and Yujin Tang. Transformer-squared: Self-adaptive llms, 2025. URL
<https://arxiv.org/abs/2501.06252>.

648 Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient
 649 methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and
 650 K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press,
 651 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

652

653 Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov,
 654 Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, and Will Dabney. Understanding
 655 the performance gap between online and offline alignment algorithms, 2024. URL <https://arxiv.org/abs/2405.08448>.

656

657

658 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
 659 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural informa-*
 660 *tion processing systems*, 30, 2017.

661

662 Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences
 663 via multi-objective reward modeling and mixture-of-experts. In *EMNLP*, 2024a.

664

665 Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang,
 666 Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer2: Open-source dataset for training
 667 top-performing reward models, 2024b.

668

669 Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, and Jürgen Schmidhuber. Natural evolu-

670 *tion strategies*, 2011. URL <https://arxiv.org/abs/1106.4487>.

671

672 Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Jun-

673 jie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin
 674 Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and
 675 Yu-Gang Jiang. Agentgym-rl: Training llm agents for long-horizon decision making through
 676 multi-turn reinforcement learning, 2025. URL <https://arxiv.org/abs/2509.08755>.

677

678 Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu,
 679 and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study, 2024. URL
 680 <https://arxiv.org/abs/2404.10719>.

681

682 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 683 Gao, Chengan Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,
 684 Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin
 685 Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,
 686 Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui
 687 Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang
 688 Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger
 689 Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan
 690 Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.

691

692 Qiyi Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai,
 693 Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guang-
 694 ming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu,
 695 Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao
 696 Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingx-
 697 uan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL
 698 <https://arxiv.org/abs/2503.14476>.

699

700 Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu
 701 Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen.
 702 Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark, 2024.
 703 URL <https://arxiv.org/abs/2402.11592>.

704

705 Yiqun Zhang, Peng Ye, Xiaocui Yang, Shi Feng, Shufei Zhang, Lei Bai, Wanli Ouyang, and Shuyue
 706 Hu. Nature-inspired population-based evolution of large language models, 2025. URL <https://arxiv.org/abs/2503.01155>.

702 Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin,
703 Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu,
704 Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun,
705 Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. Secrets of rlfh in large language
706 models part i: Ppo, 2023. URL <https://arxiv.org/abs/2307.04964>.

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756 **Algorithm 1** ESSA: Distributed Evolutionary Search over LoRA Singular Values

757 **Require:** Training set \mathcal{D} , LoRA factors $A_i \in \mathbb{R}^{r \times n}$, $B_i \in \mathbb{R}^{m \times r}$ for all attention matrices, fraction
 758 α of top singular values to optimize, population size P , number of GPUs N , number of ES
 759 generations E .

760 1: **SVD initialization:**

761 2: **for** each LoRA factor A_i, B_i **do**

762 3: $A_i = U_{A_i} \Sigma_{A_i} V_{A_i}^\top$, $B_i = U_{B_i} \Sigma_{B_i} V_{B_i}^\top$

763 4: Keep $U_{A_i}, V_{A_i}, U_{B_i}, V_{B_i}$ fixed; collect singular values $\sigma_{A_i} = \text{diag}(\Sigma_{A_i})$, $\sigma_{B_i} = \text{diag}(\Sigma_{B_i})$

764 5: **end for**

765 6: Concatenate the top $\alpha \times 100\%$ singular values of each $\sigma_{A_i}, \sigma_{B_i}$ into a single parameter vector
 766 σ .

767 7: Initialize N workers with identical random seeds and a common CMA-ES state.

768 8: **for** $t = 0, 1, \dots, E - 1$ **do**

769 9: **for** each worker $k = 1, \dots, N$ **in parallel do**

770 10: Sample candidate perturbations of σ using the shared random seeds.

771 11: For each candidate:

772 1. Reconstruct A'_i, B'_i by adding perturbation vector to σ and computing

773
$$A'_i = U_{A_i} \text{diag}(\tilde{\sigma}_{A_i}) V_{A_i}^\top, B'_i = U_{B_i} \text{diag}(\tilde{\sigma}_{B_i}) V_{B_i}^\top.$$

774 2. Update the model weights $W'_i = W_i + B'_i A'_i$.

775 3. Draw a random mini-batch $\mathcal{D}_{\text{mini}} \subset \mathcal{D}$ of fixed size and compute the reward
 776 $F = F(W'; \mathcal{D}_{\text{mini}})$.

777 12: **end for**

778 13: All workers exchange seeds and the corresponding rewards so that each worker knows the
 779 full set of evaluations.

780 14: Each worker reconstructs the same perturbations from the shared seeds and performs one
 781 CMA-ES update of the search distribution.

782 15: **end for**

783 16: Reconstruct the final LoRA factors from the evolved singular values and return the aligned
 784 model.

785

786

787 **A ESSA ALGORITHM**

788

789

790 The overall procedure of ESSA is summarized in Algorithm 1, which outlines the main steps of the
 791 distributed evolutionary search over LoRA singular values.

792

793

794 **B DETAILED THEORETICAL ANALYSIS**

795

796

797 This appendix provides the complete derivation of the latency model summarized in Section 3.3.
 798 We first introduce the notation and expressions for per-iteration computation and communication
 799 time for both gradient-based methods and ESSA, then establish the optimal split of devices between
 800 training and generation, and finally prove a conservative bound on the ESSA population size under
 801 which ESSA is guaranteed to be faster.

802

803 **Notation.** Let B^{grad} be the global batch size used in gradient methods, B^{essa} the batch size pro-
 804 cessed by a single population instance, $b_{\text{fb}}, b_{\text{gen}}$ the microbatch sizes, $m_{\text{fb}}, m_{\text{gen}}$ GPUs per model
 805 instance for training and generation, G total GPUs with $G_{\text{fb}}, G_{\text{gen}}$ allocated to each, $\tau_{\text{fb}}(b), \tau_{\text{gen}}(b)$ the
 806 forward-backward and generation microbatch times, $\eta_{\text{fb}} = \tau_{\text{fb}}(b_{\text{fb}})/b_{\text{fb}}$ and $\eta_{\text{gen}} = \tau_{\text{gen}}(b_{\text{gen}})/b_{\text{gen}}$
 807 the per-sample times, and $k_{\text{fb}}^{\text{parallel}} = G_{\text{fb}}/m_{\text{fb}}$, $k_{\text{gen}}^{\text{parallel}} = G_{\text{gen}}/m_{\text{gen}}$ the numbers of microbatches
 808 processed in parallel. $T^{\text{grad}} = T_{\text{fb}-\text{gen}}^{\text{grad}} + T_{\text{comm}}^{\text{grad}}$, $T^{\text{essa}} = T_{\text{gen}}^{\text{essa}} + T_{\text{comm}}^{\text{essa}}$, with asynchronous training
 809 and generation so that $T_{\text{fb}-\text{gen}}^{\text{grad}} = \max(T_{\text{fb}}^{\text{grad}}, T_{\text{gen}}^{\text{grad}})$.

810 **Computation.** Processing B^{grad} samples by gradient training requires $B^{\text{grad}}/b_{\text{fb}}$ microbatches;
 811 $k_{\text{fb}}^{\text{parallel}}$ run in parallel:
 812

$$813 \quad T_{\text{fb}}^{\text{grad}} = \frac{B^{\text{grad}}}{b_{\text{fb}} k_{\text{fb}}^{\text{parallel}}} \tau_{\text{fb}}(b_{\text{fb}}) = \frac{B^{\text{grad}} m_{\text{fb}}}{G_{\text{fb}}} \eta_{\text{fb}}. \quad T_{\text{gen}}^{\text{grad}} = \frac{B^{\text{grad}} m_{\text{gen}}}{G_{\text{gen}}} \eta_{\text{gen}}.$$

816 For ESSA, a population of N_{pop} candidates is evaluated purely by generation:

$$817 \quad T_{\text{gen}}^{\text{essa}} = \frac{N_{\text{pop}} B^{\text{essa}} m_{\text{gen}}}{G} \eta_{\text{gen}}.$$

820 **Communication.** Let M_{params} be the model-parameter size (bytes). Gradient methods communicate
 821 gradients using all-reduce which consists of two collective operations: reduce-scatter and
 822 all-gather. Each moves a block of size M_{params}/G across the $G - 1$ other devices. With effective
 823 interconnect peak bandwidth peak_bw (bytes/s) – the sustained per-GPU bandwidth for large
 824 collective messages:

$$825 \quad T_{\text{comm}}^{\text{grad}} = 2 \cdot \frac{M_{\text{params}}(G - 1)}{G \text{ peak_bw}}.$$

827 ESSA communicates only a random seed and the resulting reward, requiring a single all-gather of
 828 size $M_{\text{essa}} = 2 \times 4$ bytes:

$$829 \quad T_{\text{comm}}^{\text{essa}} = \frac{M_{\text{essa}}(G - 1)}{G \text{ peak_bw}}.$$

832 **Optimal device split.** Let $\theta \in (0, 1)$ be the fraction of devices used for training ($G_{\text{fb}} = \theta G$,
 833 $G_{\text{gen}} = (1 - \theta)G$). Then

$$834 \quad T_{\text{fb-gen}}^{\text{grad}}(\theta) = \frac{B^{\text{grad}}}{G} \max\left(\frac{m_{\text{fb}} \eta_{\text{fb}}}{\theta}, \frac{m_{\text{gen}} \eta_{\text{gen}}}{1 - \theta}\right).$$

837 **Lemma B.1** (Optimal split). *The minimum of $T_{\text{fb-gen}}^{\text{grad}}(\theta)$ over $\theta \in (0, 1)$ is attained at*

$$839 \quad \theta^* = \frac{m_{\text{fb}} \eta_{\text{fb}}}{m_{\text{fb}} \eta_{\text{fb}} + m_{\text{gen}} \eta_{\text{gen}}}, \quad T_{\text{fb-gen}}^{\text{grad}}(\theta^*) = \frac{B^{\text{grad}}}{G} (m_{\text{fb}} \eta_{\text{fb}} + m_{\text{gen}} \eta_{\text{gen}}).$$

842 *Proof.* $\max(a/\theta, b/(1 - \theta))$ is minimized when the two arguments are equal: $a/\theta = b/(1 - \theta)$,
 843 with $a = m_{\text{fb}} \eta_{\text{fb}}$, $b = m_{\text{gen}} \eta_{\text{gen}}$. \square

845 Define the ideal gradient iteration time (perfect scheduling):

$$847 \quad T_{\star}^{\text{grad}} = T_{\text{fb-gen}}^{\text{grad}}(\theta^*) + T_{\text{comm}}^{\text{grad}} = \frac{B^{\text{grad}}}{G} (m_{\text{fb}} \eta_{\text{fb}} + m_{\text{gen}} \eta_{\text{gen}}) + 2 \cdot \frac{M_{\text{params}}(G - 1)}{G \text{ peak_bw}}.$$

849 Clearly $T^{\text{grad}} \geq T_{\star}^{\text{grad}}$.

851 **Theorem B.2** (ESSA iteration is faster under a conservative bound). *Suppose we only assume
 852 $m_{\text{fb}} \geq 1$, $m_{\text{gen}} \leq G$, and $2M_{\text{params}} - M_{\text{essa}} \geq M_{\text{params}}$. If the population size satisfies*

$$853 \quad N_{\text{pop}} B^{\text{essa}} < B^{\text{grad}} \left(\frac{\eta_{\text{fb}}}{G \eta_{\text{gen}}} + 1 \right) + \frac{(G - 1) M_{\text{params}}}{\text{peak_bw} G \eta_{\text{gen}}}, \quad (1)$$

856 then $T^{\text{essa}} < T_{\star}^{\text{grad}} \leq T^{\text{grad}}$.

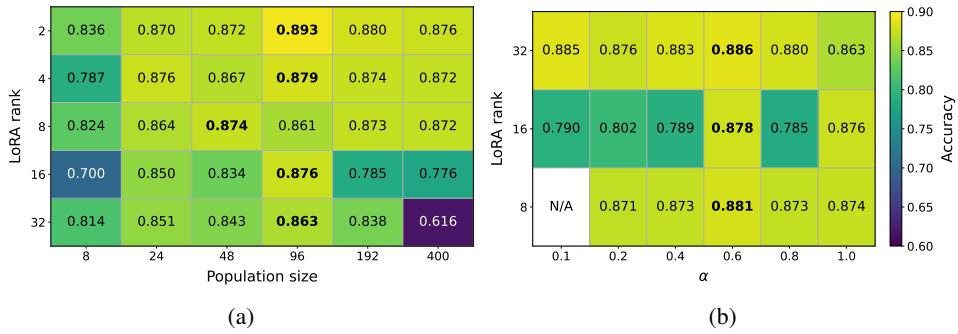
858 *Proof.* Starting from $T^{\text{essa}} < T_{\star}^{\text{grad}}$ and applying the bounds $m_{\text{fb}} \geq 1$, $m_{\text{gen}} \leq G$,
 859 $2M_{\text{params}} - M_{\text{essa}} \geq M_{\text{params}}$ to the exact inequality yields equation 1. \square

861 **Discussion.** Theorem B.2 provides a conservative population-size threshold below which ESSA
 862 is guaranteed to be faster than the idealized gradient pipeline. Because real clusters rarely achieve
 863 the perfect split θ^* and typically operate below the nominal peak_bw , the practical advantage of
 ESSA is often even larger than predicted by equation 1.

864 C SENSITIVITY TO HYPERPARAMETERS

866 C.1 HEATMAPS

868 For completeness, Figures 9-12 present the full hyperparameter sensitivity for Section 4.1. These
 869 include Qwen2.5-7B on both GSM8K and PRM800K, Qwen2.5-Math-7B on PRM800K, as well as
 870 LLaMA-3.1-8B on IFEval.

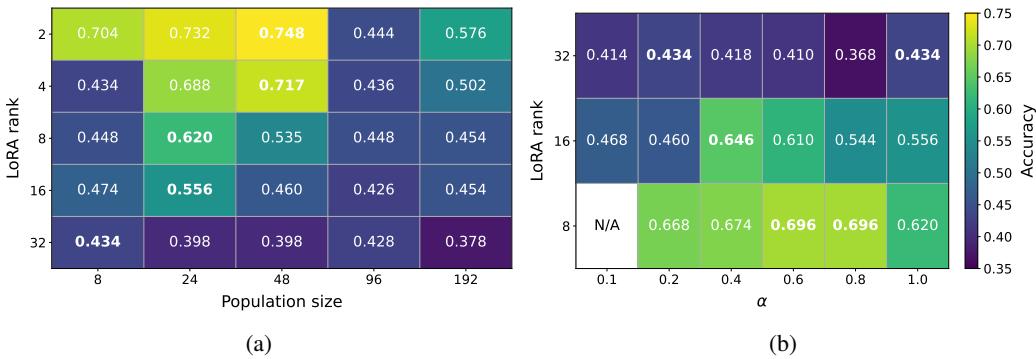


(a)

(b)

882 Figure 9: Hyperparameter sensitivity of ESSA on **Qwen2.5-7B** for **GSM8K**. Batch size 100. **(a)**
 883 Accuracy when varying LoRA rank and population size. **(b)** For each LoRA rank, the population
 884 size is fixed to the best value found in (a), while the percentage α of trainable singular values is
 885 varied. This illustrates how ESSA performance depends jointly on adapter rank and the fraction
 886 of singular values optimized. The single white cell occurs because for LoRA rank 8 and $\alpha = 0.1$,
 887 rounding down yields zero trainable singular values, so no valid accuracy is reported.

889 Figure 9 examines the hyperparameter sensitivity of ESSA on Qwen2.5-7B for GSM8K. Varying the
 890 LoRA rank and population size shows that performance remains stable across a broad range, with
 891 accuracy peaking at 0.893 for LoRA rank 2 and population size 96. Notably, even very low-ranks (2-
 892 4) achieve top performance, while larger ranks yield diminishing or slightly degraded results. When
 893 fixing the population size and varying the fraction α of trainable singular values, ESSA maintains
 894 consistently high accuracy (≈ 0.87 -0.89) across all α values. This indicates strong robustness to the
 895 degree of SVD sparsification.



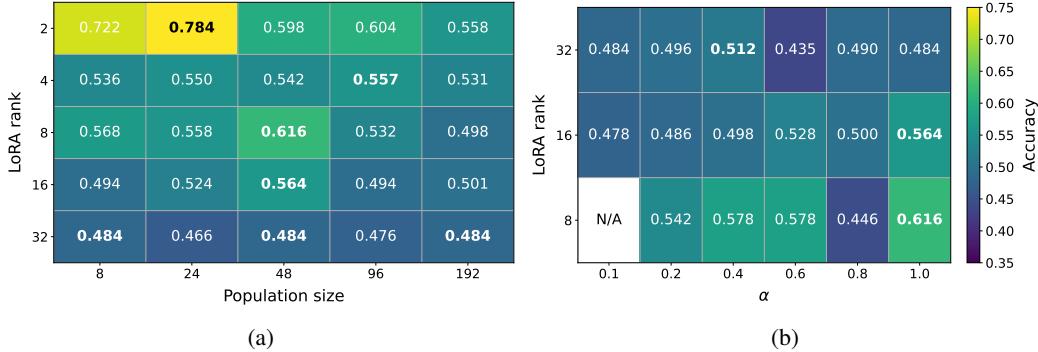
(a)

(b)

896 Figure 10: Hyperparameter sensitivity of ESSA on **Qwen2.5-7B** for **PRM800K**. Batch size 300. **(a)**
 897 Accuracy when varying LoRA rank and population size. **(b)** For each LoRA rank, the population
 898 size is fixed to the best value found in (a), while the percentage α of trainable singular values is
 899 varied. This illustrates how ESSA performance depends jointly on adapter rank and the fraction
 900 of singular values optimized. The single white cell occurs because for LoRA rank 8 and $\alpha = 0.1$,
 901 rounding down yields zero trainable singular values, so no valid accuracy is reported.

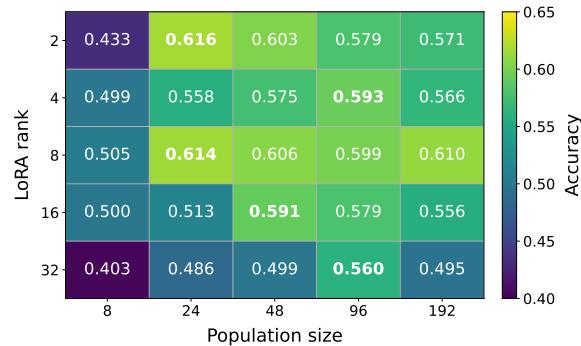
914 Figure 10 analyzes the hyperparameter sensitivity of ESSA on Qwen2.5-7B for PRM800K. When
 915 varying LoRA rank and population size, accuracy peaks at 0.748 for LoRA rank 2 and population
 916 size 48, showing that small adapter ranks with moderate population sizes are most effective. Per-
 917 formance drops notably for larger ranks (16) or oversized populations (96), indicating diminishing

918 returns beyond a compact search space. With population size fixed, varying the fraction α of trainable
 919 singular values shows that moderate values ($\alpha \approx 0.4$ -0.8) yield the best results – up to 0.696
 920 accuracy for LoRA rank 8 – while very low or full updates slightly reduce performance. These
 921 results confirm that ESSA performs best with low-rank adapters, moderate population sizes, and
 922 partial singular value optimization.



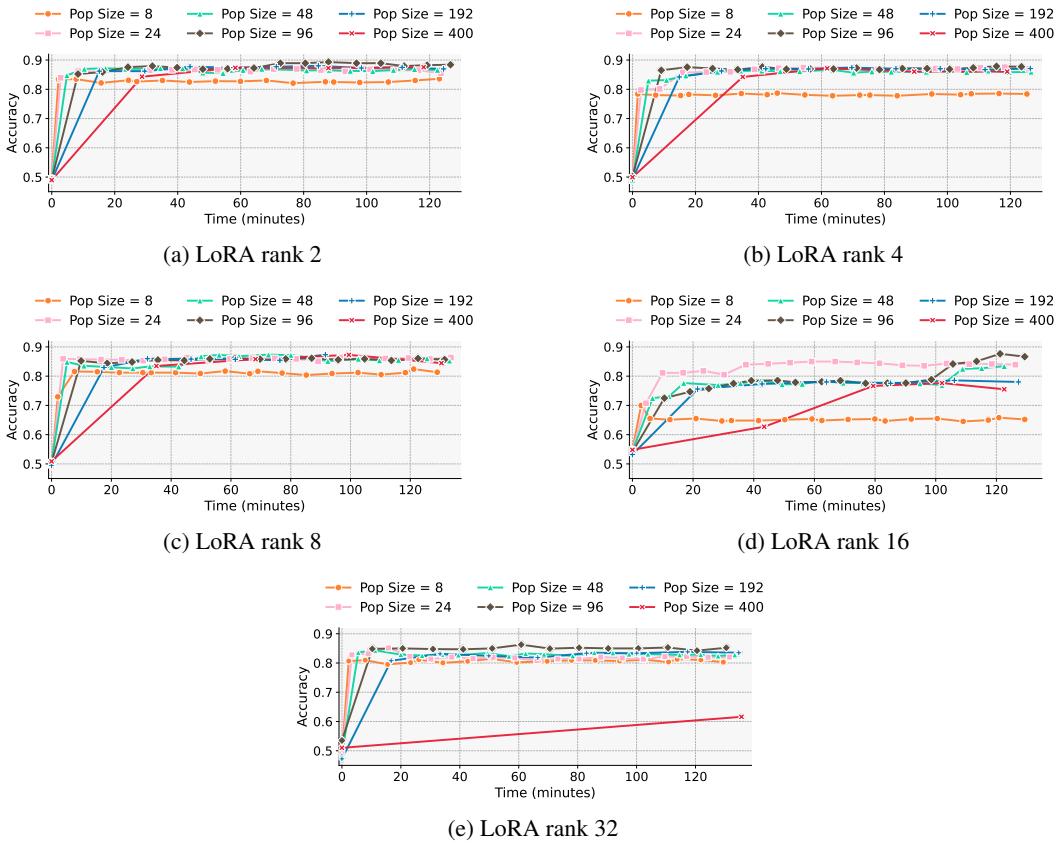
936 Figure 11: Hyperparameter sensitivity of ESSA on **Qwen2.5-Math-7B** for **PRM800K**. Batch size
 937 300. **(a)** Accuracy when varying LoRA rank and population size. **(b)** For each LoRA rank, the
 938 population size is fixed to the best value found in (a), while the percentage α of trainable singular
 939 values is varied. This illustrates how ESSA performance depends jointly on adapter rank and the
 940 fraction of singular values optimized. The single white cell occurs because for LoRA rank 8 and
 941 $\alpha = 0.1$, rounding down yields zero trainable singular values, so no valid accuracy is reported.

943 Figure 11 presents the hyperparameter sensitivity of ESSA on Qwen2.5-Math-7B for PRM800K.
 944 When varying LoRA rank and population size, accuracy peaks at 0.784 for LoRA rank 2 and popu-
 945 lation size 24, showing that compact low-rank adapters with small populations perform best. Higher
 946 ranks or very large populations lead to gradual accuracy degradation, indicating that excessive search
 947 dimensionality does not improve performance. Fixing population size and varying the fraction α of
 948 trainable singular values, ESSA achieves its highest score (0.616) at LoRA rank 8 and full update
 949 ($\alpha = 1.0$), while intermediate α values yield competitive but slightly lower results. The results high-
 950 light that ESSA performs optimally with lightweight low-rank configurations and remains robust to
 951 SVD sparsification, even on complex reasoning tasks.



964 Figure 12: Hyperparameter sensitivity of ESSA on **LLaMA-3.1-8B** for **IFeval**. Accuracy when
 965 varying LoRA rank and population size.

966 Figure 12 shows the hyperparameter sensitivity of ESSA on LLaMA-3.1-8B for IFeval. Varying
 967 LoRA rank and population size reveals stable performance across configurations, with the highest
 968 accuracy (0.614) achieved at LoRA rank 8 and population size 24. Low ranks (4-16) yield com-
 969 parable results, while very high-rank (32) adapters slightly underperform, suggesting a sweet spot
 970 around low-range ranks and moderate population sizes.

972 C.2 TRAINING DYNAMICS
973974 C.2.1 QWEN2.5-7B ON GSM8K
975976 Appendix Figures 13, 14 and 15 provide the complete training-dynamic study of ESSA on Qwen2.5-
977 7B for GSM8K. This analysis shows how validation accuracy evolves over time as we vary the three
978 key ESSA hyperparameters (Figure 9).
9791006 Figure 13: Validation accuracy over time on **GSM8K** with **Qwen2.5-7B** when varying the ESSA
1007 population size (8, 24, 48, 96, 192, 400) at fixed LoRA ranks (2, 4, 8, 16, 32). Batch size 100.
10081009 Figure 13 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-7B when varying
1010 the ESSA population size.
1011

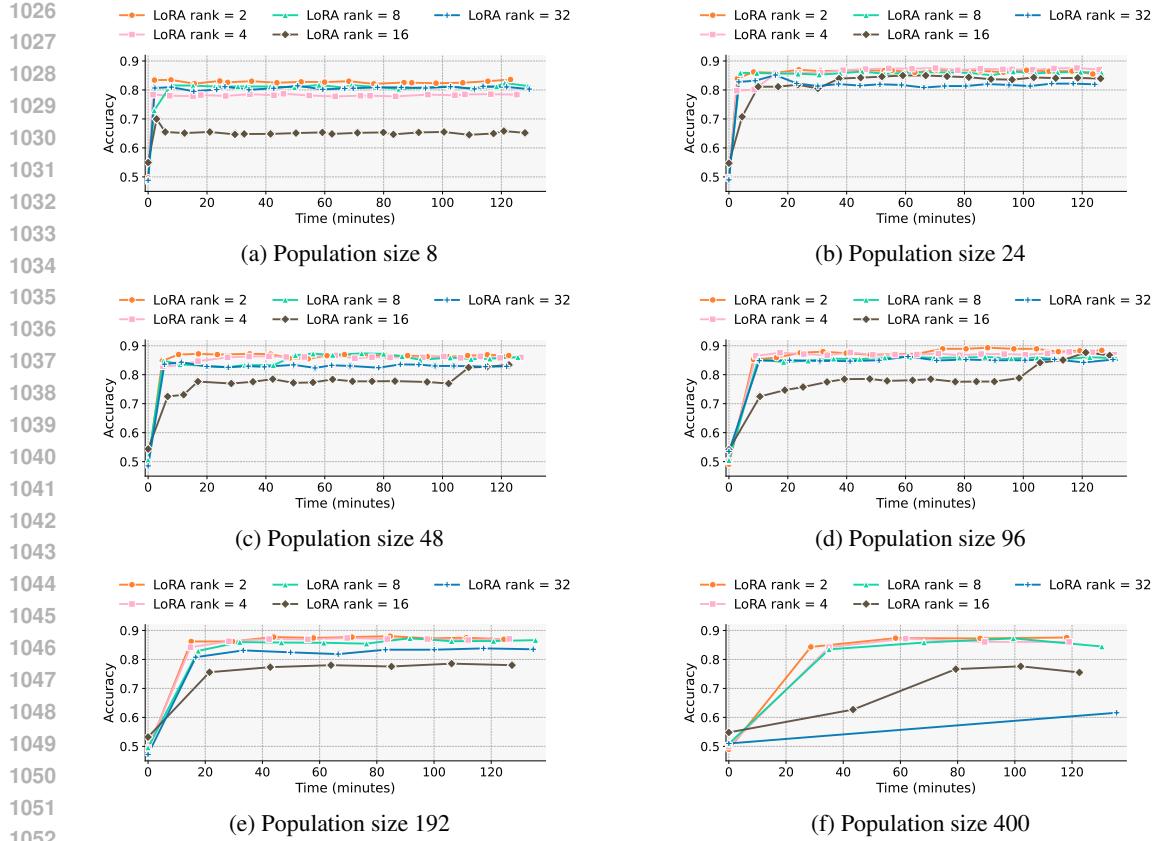


Figure 14: Validation accuracy over time on **GSM8K** with **Qwen2.5-7B** when varying the LoRA rank (2, 4, 8, 16, 32) at fixed population sizes (8, 24, 48, 96, 192, 400). Batch size 100.

Figure 14 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-7B when varying the LoRA rank.

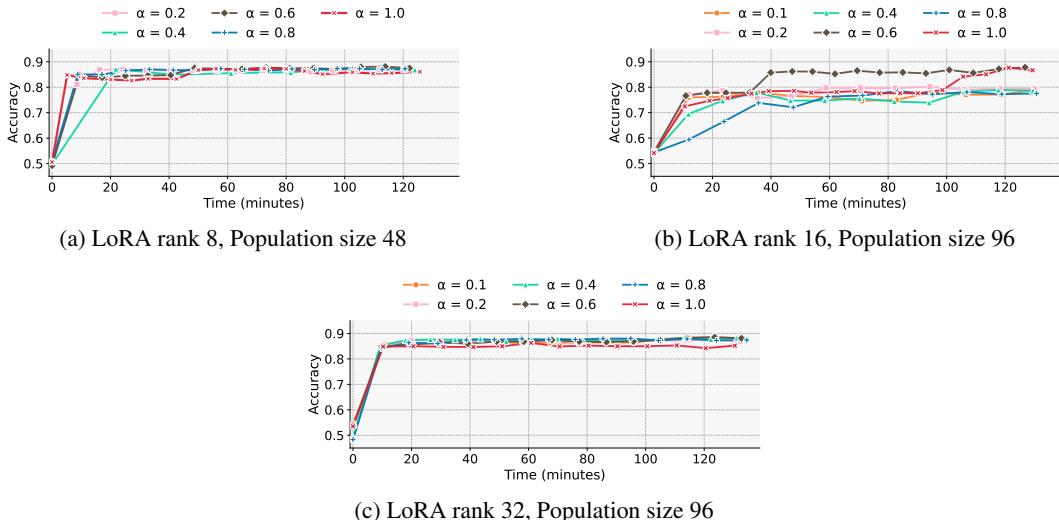


Figure 15: Validation accuracy on **GSM8K** with **Qwen2.5-7B** while varying α . LoRA rank and population size are fixed to the optimal choices from Figure 9. Batch size 100.

1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133

Figure 15 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-7B when varying the fraction α of trainable singular values.

C.2.2 QWEN2.5-MATH-7B ON GSM8K

Appendix Figures 16, 17 and 18 provide the complete training-dynamic study of ESSA on Qwen2.5-Math-7B for GSM8K. These graphs complement the main text sensitivity analysis by showing how validation accuracy evolves over time as we vary the three key ESSA hyperparameters (Figure 2).

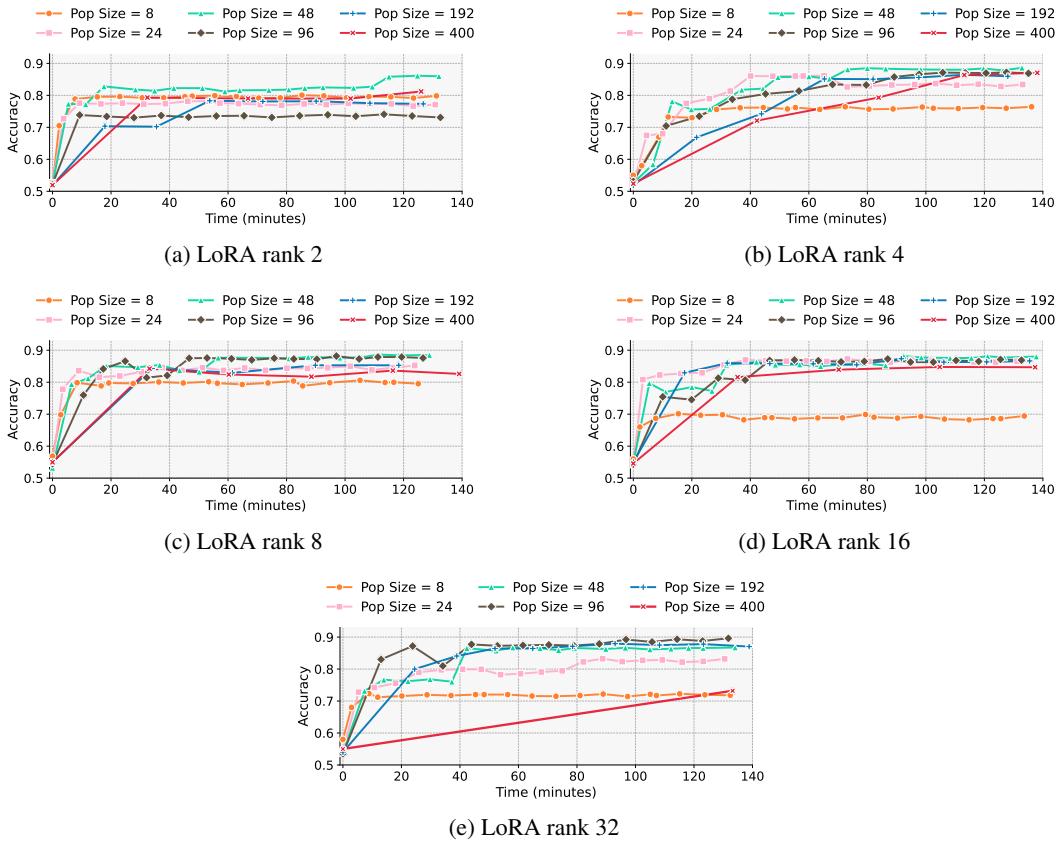


Figure 16: Validation accuracy over time on **GSM8K** with **Qwen2.5-Math-7B** when varying the ESSA population size (8, 24, 48, 96, 192, 400) at fixed LoRA ranks (2, 4, 8, 16, 32). Batch size 100.

Figure 16 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-Math-7B when varying the ESSA population size.

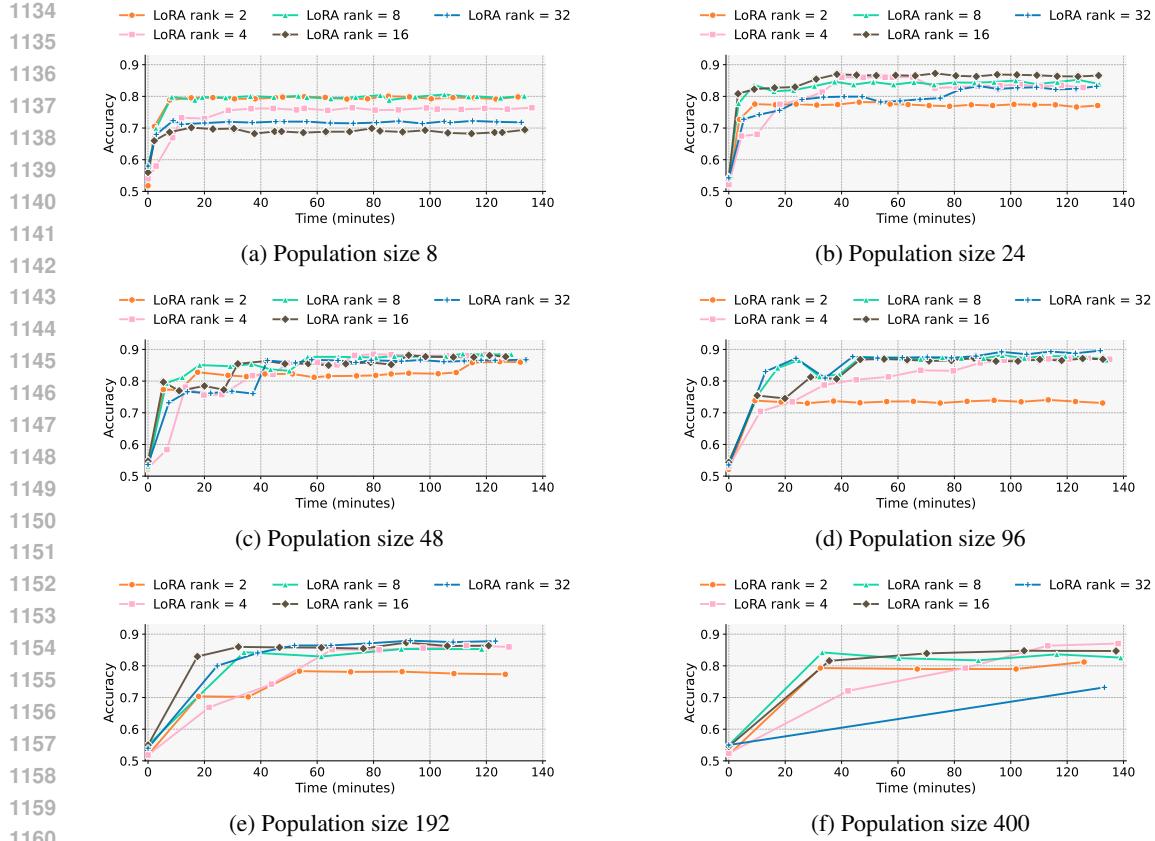


Figure 17: Validation accuracy over time on **GSM8K** with **Qwen2.5-Math-7B** when varying the LoRA rank (2, 4, 8, 16, 32) at fixed population sizes (8, 24, 48, 96, 192, 400). Batch size 100.

Figure 17 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-Math-7B when varying the the LoRA rank.

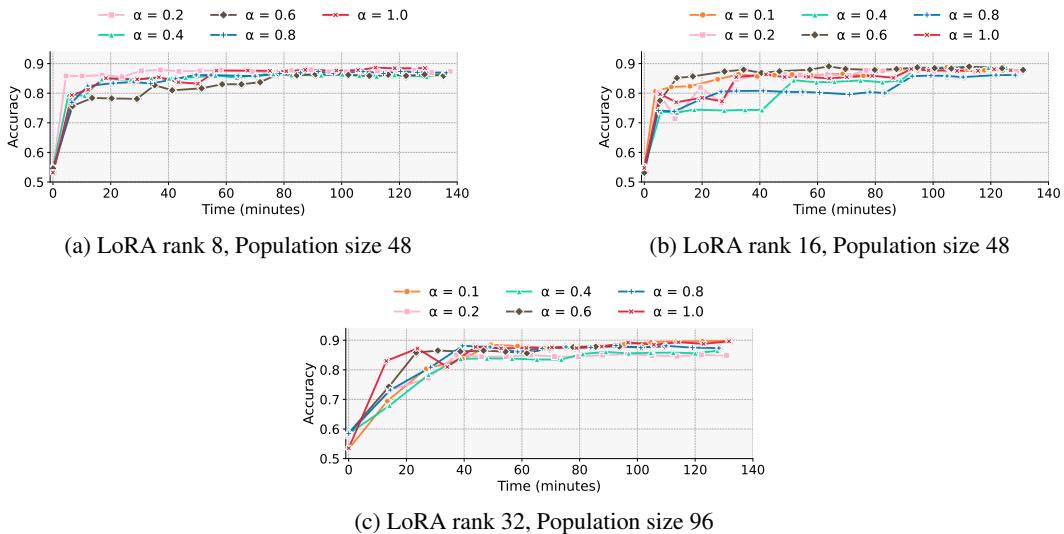


Figure 18: Validation accuracy on **GSM8K** with **Qwen2.5-Math-7B** while varying α . LoRA rank and population size are fixed to the optimal choices from Figure 2. Batch size 100.

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
Figure 18 shows the full training dynamics of ESSA on GSM8K with Qwen2.5-Math-7B when varying the fraction α of trainable singular values.

C.2.3 QWEN2.5-7B ON PRM800K

Appendix Figures 19, 20 and 21 provide the complete training-dynamic study of ESSA on Qwen2.5-7B for PRM800K. This analysis shows how validation accuracy evolves over time as we vary the three key ESSA hyperparameters (Figure 10)

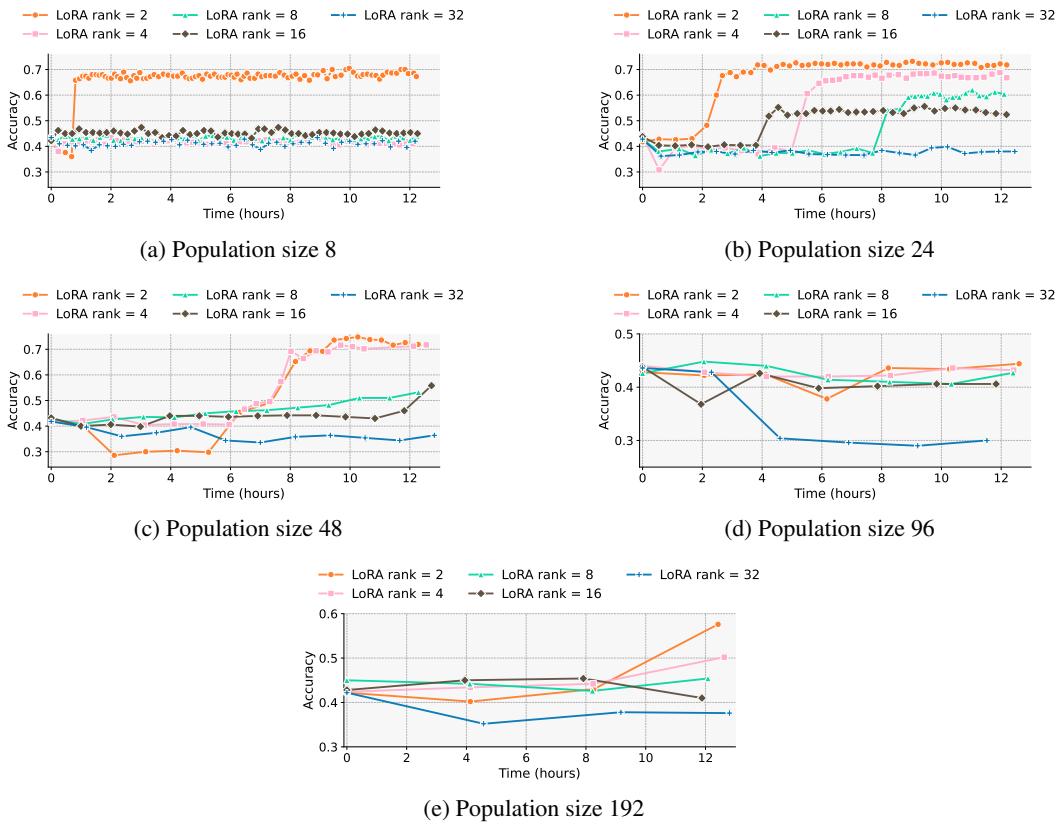


Figure 19: Validation accuracy over time on **PRM800K** with **Qwen2.5-7B** when varying the ESSA population size (8, 24, 48, 96, 192) at fixed LoRA ranks (2, 4, 8, 16, 32). Batch size 300.

Figure 19 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-7B when varying the ESSA population size.

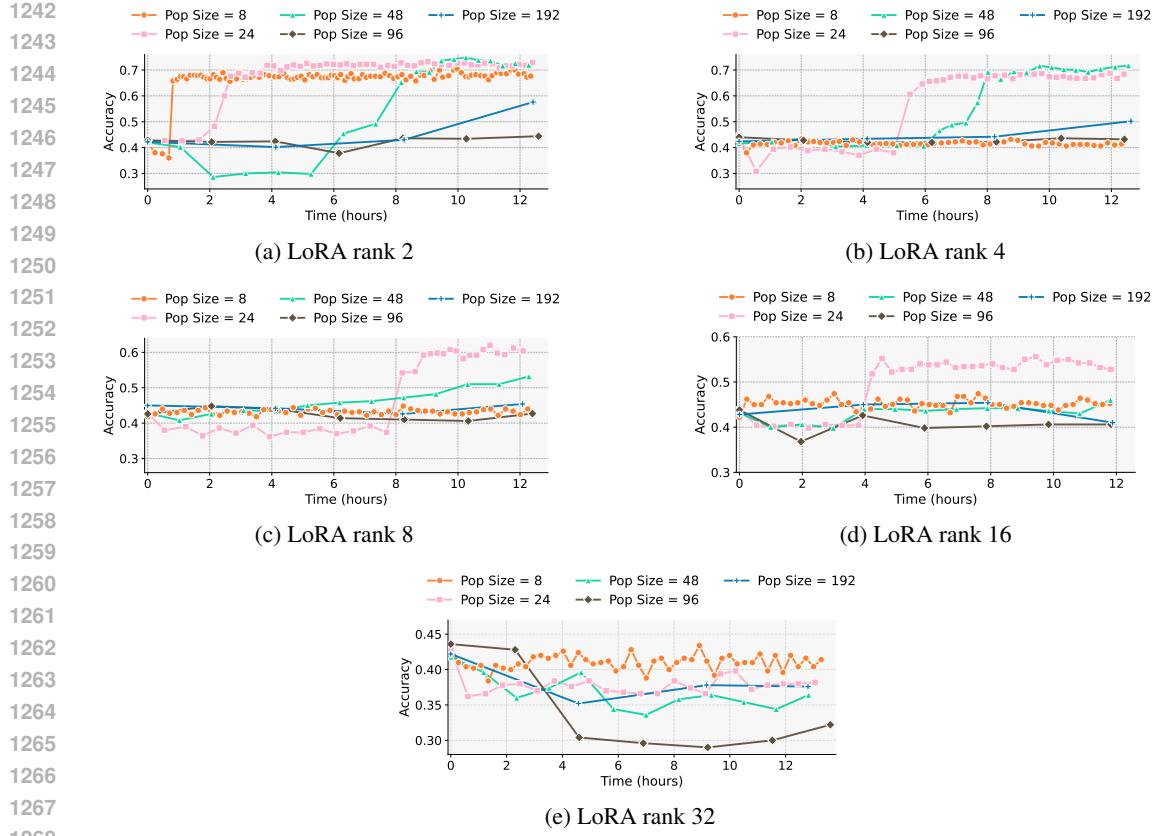


Figure 20: Validation accuracy over time on **PRM800K** with **Qwen2.5-7B** when varying the LoRA rank (2, 4, 8, 16, 32) at fixed population sizes (8, 24, 48, 96, 192). Batch size 300.

Figure 20 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-7B when varying the LoRA rank.

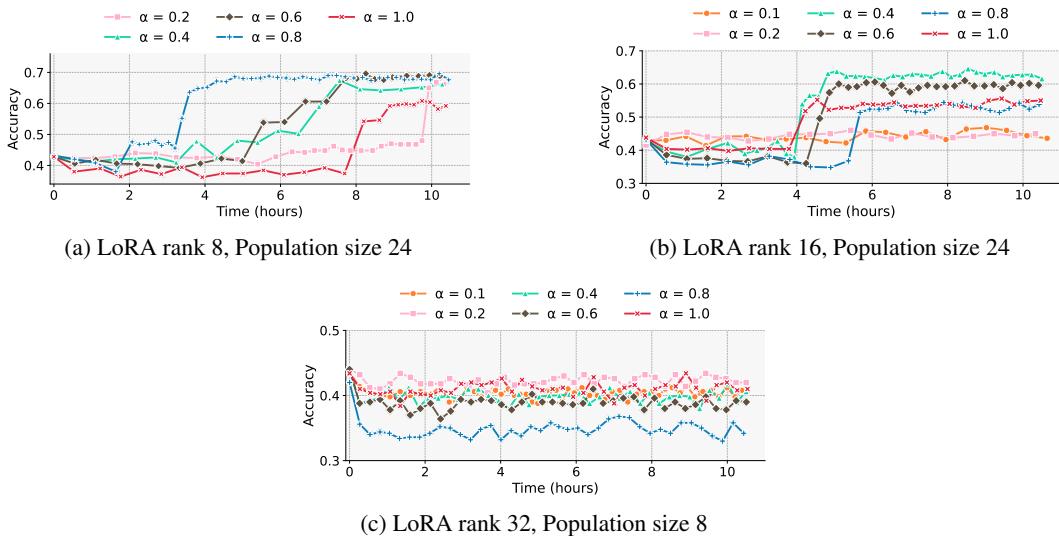
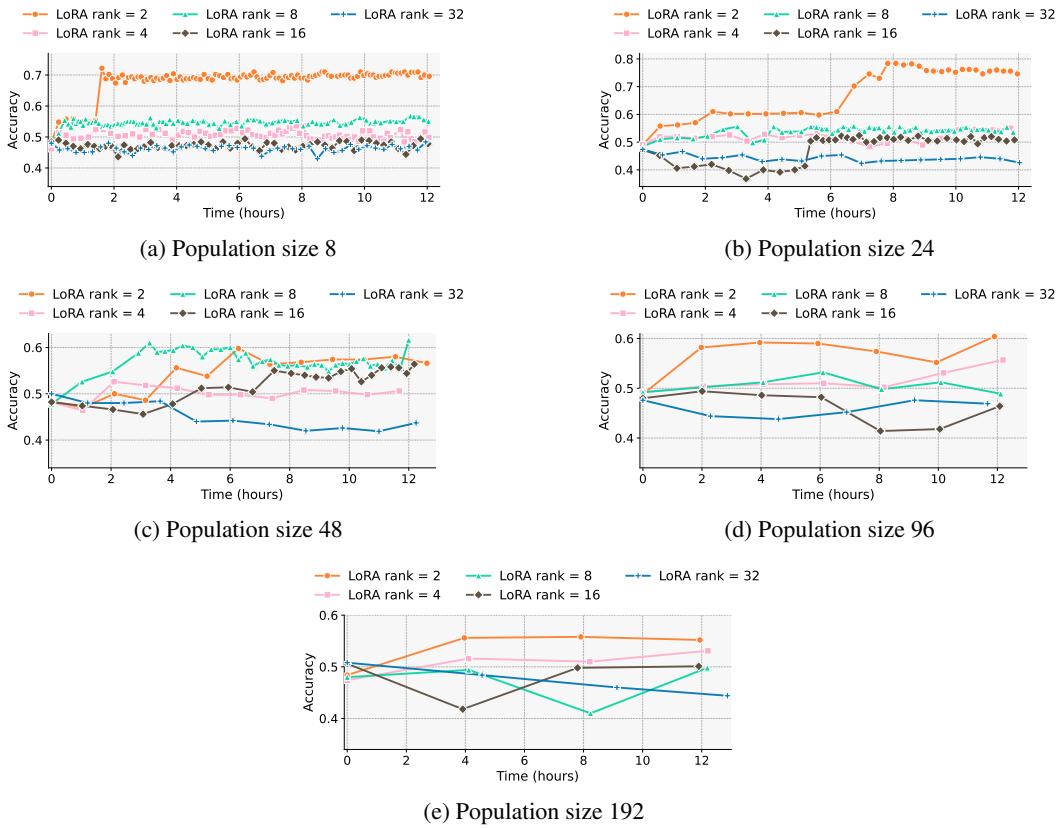


Figure 21: Validation accuracy over time on **PRM800K** with **Qwen2.5-7B** while varying α . LoRA rank and population size are fixed to the optimal choices from Figure 10. Batch size 300.

1296 Figure 21 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-7B when varying
 1297 the fraction α of trainable singular values.
 1298

1299 C.2.4 QWEN2.5-MATH-7B ON PRM800K

1300
 1301 Appendix Figures 22, 23 and 24 provide the complete training-dynamic study of ESSA on Qwen2.5-
 1302 Math-7B for PRM800K. These graphs complement the main text sensitivity analysis by showing
 1303 how validation accuracy evolves over time as we vary the three key ESSA hyperparameters
 1304 (Figure 11).



1323 Figure 22: Validation accuracy over time on **PRM800K** with **Qwen2.5-Math-7B** when varying the
 1324 ESSA population size (8, 24, 48, 96, 192) at fixed LoRA ranks (2, 4, 8, 16, 32). Batch size 300.

1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332 Figure 22 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-Math-7B when
 1333 varying the ESSA population size.
 1334

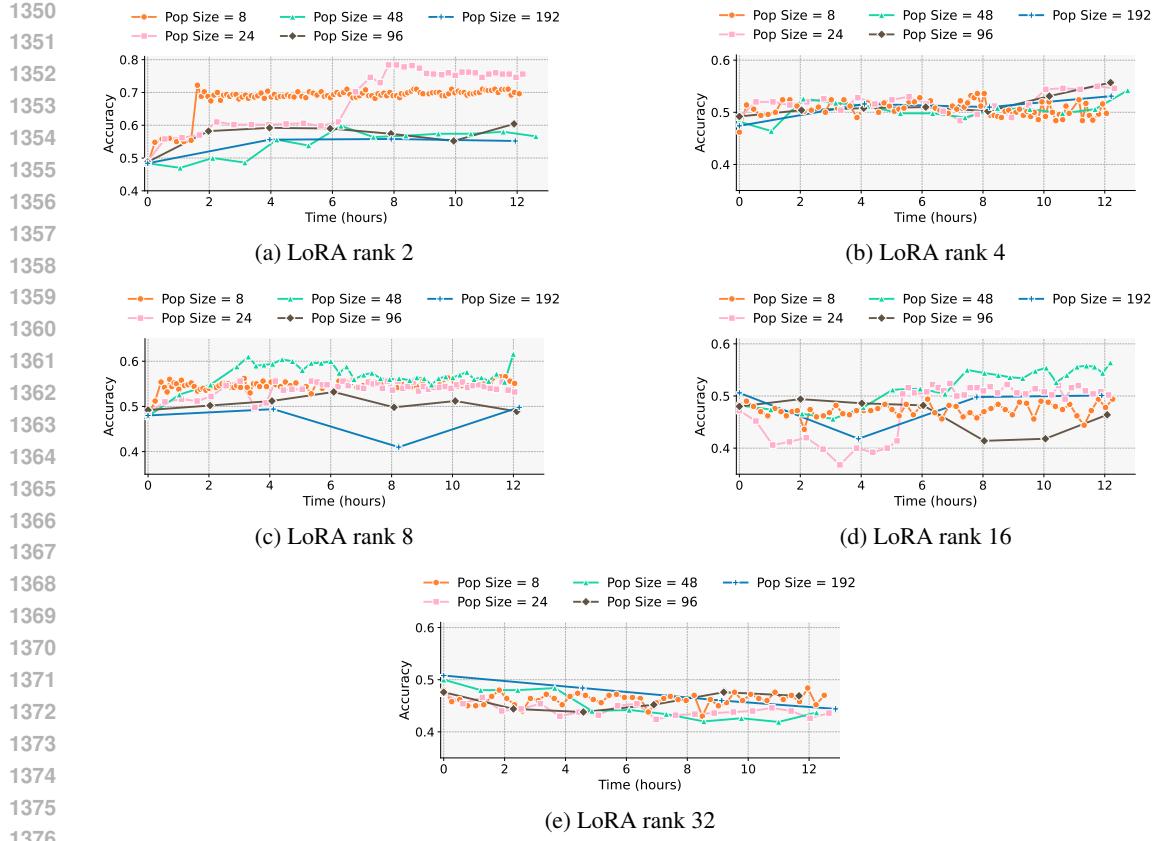


Figure 23: Validation accuracy over time on **PRM800K** with **Qwen2.5-Math-7B** when varying the LoRA rank (2, 4, 8, 16, 32) at fixed population sizes (8, 24, 48, 96, 192). Batch size 300.

Figure 23 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-Math-7B when varying the LoRA rank.

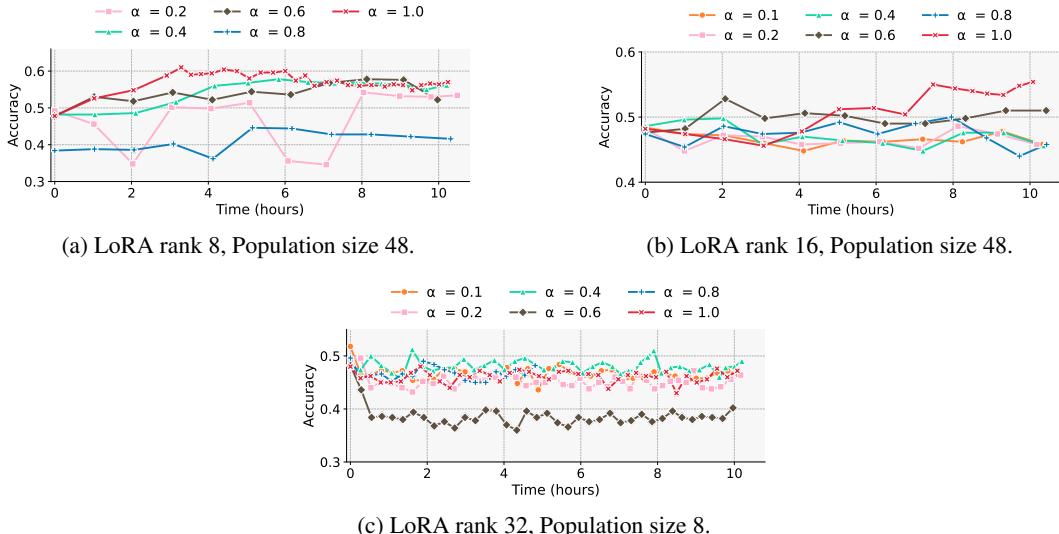


Figure 24: Validation accuracy over time on **PRM800K** with **Qwen2.5-Math-7B** when varying α . LoRA rank and population size are fixed to the optimal choices from Figure 11. Batch size 300.

1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457

Figure 24 shows the full training dynamics of ESSA on PRM800K with Qwen2.5-Math-7B when varying the fraction α of trainable singular values.

C.2.5 LLAMA3.1-8B ON IFEVAL

Appendix Figures 25 and 26 provide the complete training-dynamic study of ESSA on LLaMA3.1-8B for IFEval. Two graphs complement the main text sensitivity analysis by showing how validation accuracy evolves over time as we vary the two key ESSA hyperparameters (Figure 12)

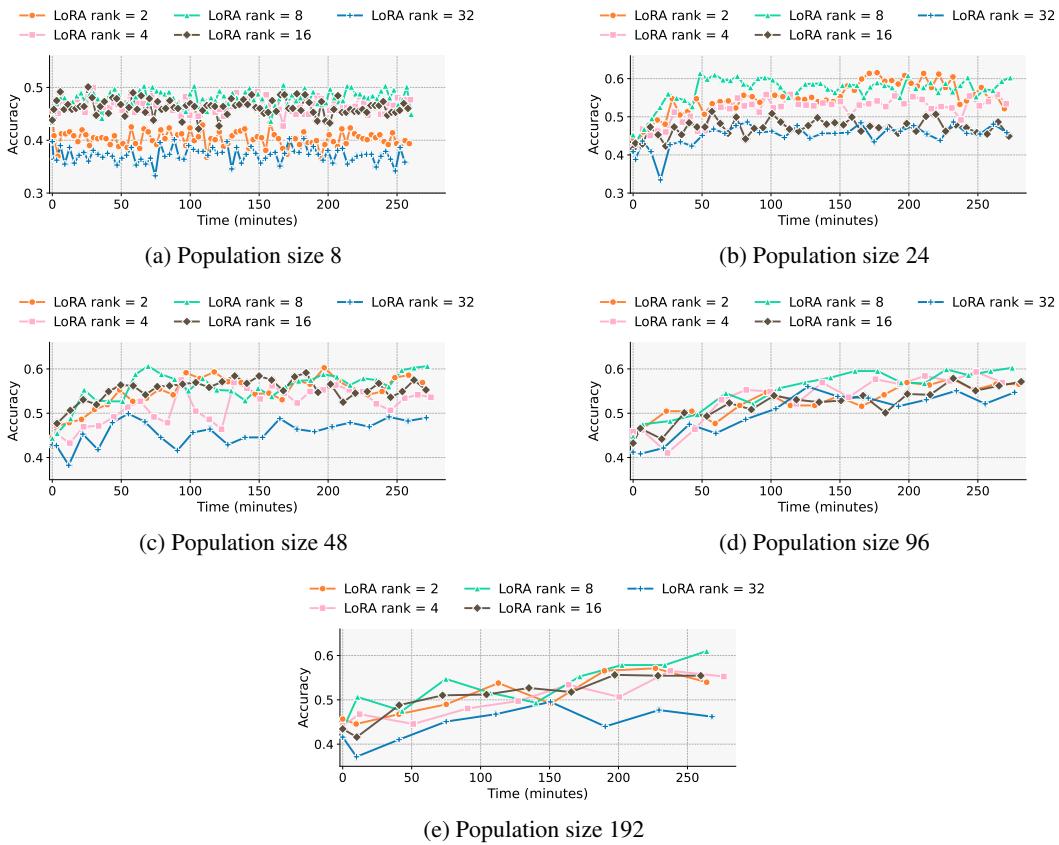


Figure 25: Validation accuracy over time on **IFEval** with **LLaMA3.1-8B** when varying the ESSA population size (8, 24, 48, 96, 192) at fixed LoRA ranks (2, 4, 8, 16, 32). Batch size 500, $\alpha = 1.0$.

Figure 25 shows the full training dynamics of ESSA on IFEval with LLaMA3.1-8B when varying the ESSA population size.

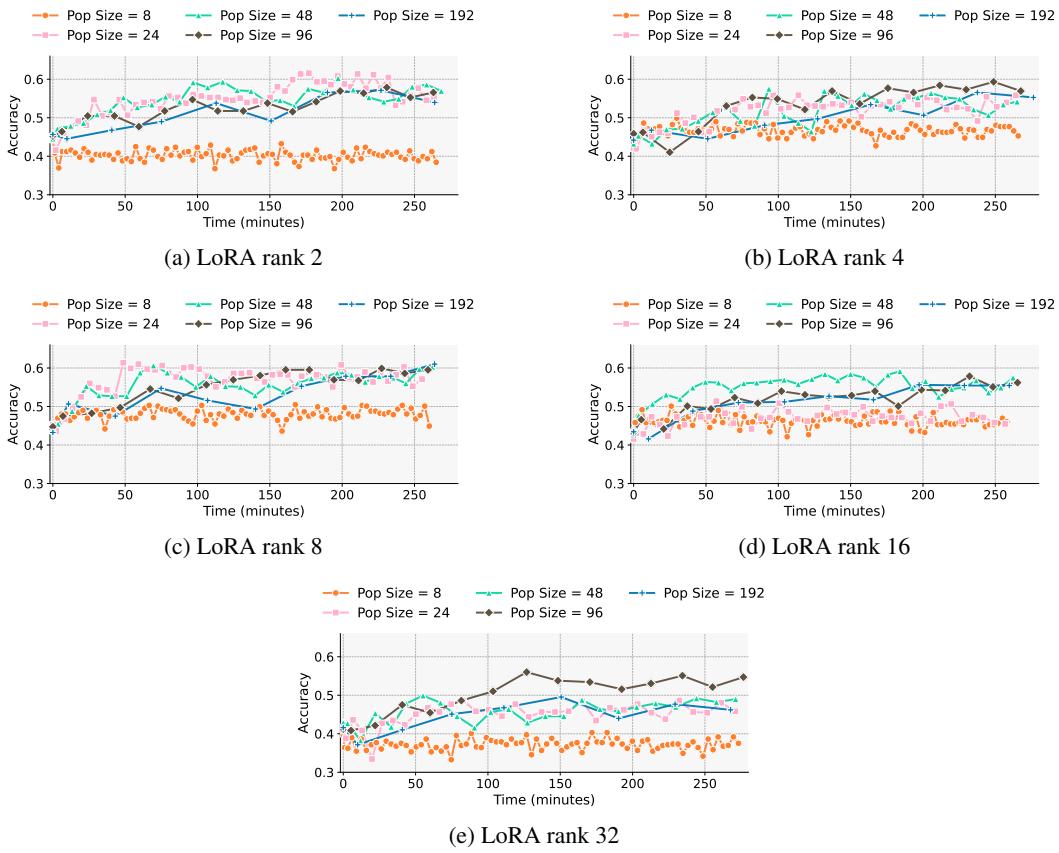


Figure 26: Validation accuracy over time on **IFEval** with **LLaMA3.1-8B** when varying the LoRA rank (2, 4, 8, 16, 32) at fixed population sizes (8, 24, 48, 96, 192). Batch size 500, $\alpha = 1.0$.

Figure 26 shows the full training dynamics of ESSA on IFEval with LLaMA3.1-8B when varying the LoRA rank.

D PRECISION ANALYSIS

Figure 27 shows the full training curves of ESSA when running Qwen2.5-32B on PRM800K with LoRA rank 8, population size 64, $\alpha = 1.0$, and per-candidate batch size 256, for three different numerical precisions: BFLOAT16, INT8, and INT4. Shaded areas indicate one standard deviation across runs.

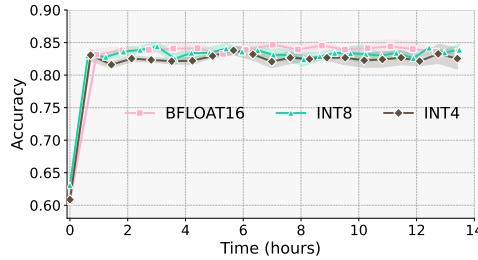


Figure 27: Validation accuracy over training time for Qwen2.5-32B on PRM800K under different weight precisions (BFLOAT16, INT8, INT4). Settings: LoRA rank 8, pop. 64, batch size 256, $\alpha = 1.0$.

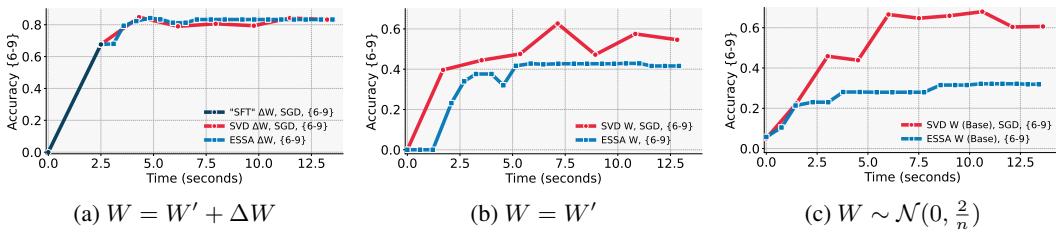
Across all precisions, accuracy rises quickly within the first hour and then remains stable. Reducing precision from BFLOAT16 to INT8 and even to INT4 produces only a slight reduction in final accuracy.

1512 racy, consistent with the summary in Table 2 of the main text. These results confirm that ESSA can
 1513 safely exploit aggressive quantization to cut memory usage and computation cost while maintaining
 1514 near-identical performance.
 1515

1516 E TOY EXAMPLE ON MNIST

1518 A natural concern arises when considering a singular-value-only parameterization. If we restrict the
 1519 problem to simple linear or logistic regression, scaling the rows of a fixed low-rank update may seem
 1520 too limited to support meaningful adaptation. In this setting only the singular magnitudes change
 1521 while all directions remain fixed, which can give the impression of insufficient flexibility.
 1522

1523 Motivated by this concern, we emphasize that ESSA’s effectiveness in LLMs relies on the structured
 1524 representations already present in the pretrained backbone and inherited through the downstream
 1525 SFT initialization. To make this distinction concrete, we designed an MNIST experiment that repli-
 1526 cates the LLM training pipeline as closely as possible within a minimal linear model.



1535 Figure 28: Validation accuracy over time on **MNIST** (on the test set of digits 6-9) with a single
 1536 matrix $\mathbf{W} \in \mathcal{R}^{n \times m}$, $n = 784$, $m = 10$. Panels (a)-(c) correspond to $W = W' + \Delta W$, $W = W'$
 1537 and $W \sim \mathcal{N}(0, \frac{2}{n})$ setups respectively, where W' is pretrained base weights. Train size 400. ESSA
 1538 (blue): solution length 10, population size 24, batch size 64. SGD (red): lr 0.1, batch size 64.

1539 Toy example on MNIST proceeds as follows:

1. At the pretrain stage (analog of LLM pretraining) we train a logistic regression model on MNIST digits 0-5 (train size 5000) using SGD optimizer. We obtain a "base model" $W = W'$, where W' is the single matrix of size 784 by 10, that knows something about the data distribution, similar to LLM pretraining.
2. At the "SFT" stage (analogue of LoRA SFT) we take 400 samples of digits 6-9, initialize a matrix update ΔW with Kaiming init, and train only ΔW using SGD: $W = W' + \Delta W$, where ΔW is the single matrix of size 784 by 10.
3. At the last stage we decomposed $\Delta W = BSA$ and train only the singular values S using ES or SGD optimizer. ESSA optimize singular values directly using train accuracy, while the SGD baseline optimized them through the standard cross-entropy loss.

1552 In Figure 28a, ESSA follows the behavior of SGD on the MNIST task with high fidelity. This
 1553 demonstrates that once the singular vectors encodes useful directions, modulating only the asso-
 1554 ciated singular amplitudes is already sufficient to achieve effective adaptation. In Figure 28b, we
 1555 remove the additive update entirely. We decompose the pretrained weights $W = W' = B'S'A'$,
 1556 and train only the singular values S' using ESSA and SGD. Even in this more constrained setup,
 1557 ESSA achieved performance comparable to SGD. The pretrained matrix contained enough structure
 1558 that rebalancing its singular directions remained a viable strategy for improving accuracy. Finally, in
 1559 Figure 28c, we examine the case without pretraining or SFT, where the model is initialized with
 1560 and the singular vectors carry no semantic information. In this setup ESSA improves only slowly, while
 1561 SGD makes substantially faster progress. Without a structured subspace, changing only singular
 1562 values is insufficient to emulate full learning dynamics.

1563 These experiments demonstrate that the singular-value parameterization is effective precisely in the
 1564 settings where LLMs operate: pretrained models with downstream SFT produce meaningful low-
 1565 rank directions, and ESSA leverages this structure by adjusting their relative scaling. This behavior
 1566 is consistent with the results observed in our large-scale LLM experiments.

1566 **F COMPARISON TO GRPO**
 1567

1568 **F.1 SCHOOL MATH**
 1569

1570 Appendix Figure 29 presents the results of Qwen2.5-7B on GSM8K in addition to Qwen2.5-Math-
 1571 7B (Figure 4).

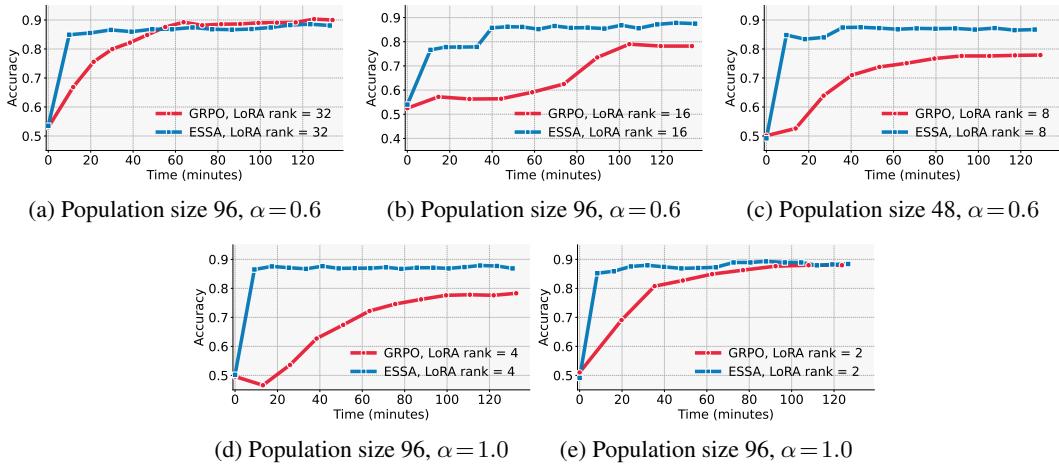


Figure 29: Validation accuracy over time on **GSM8K** with **Qwen2.5-7B**. Panels (a)-(e) correspond to LoRA ranks 32, 16, 8, 4, and 2, respectively. ESSA (blue): batch size 100. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64.

Appendix Figure 34 presents the results of Qwen2.5-7B and Qwen2.5-Math-7B on GSM8K without SFT warm-start in addition to School Math experiments (Section 4.4.1). LoRA matrix A is initialized using Kaiming initialization, matrix B is zero.

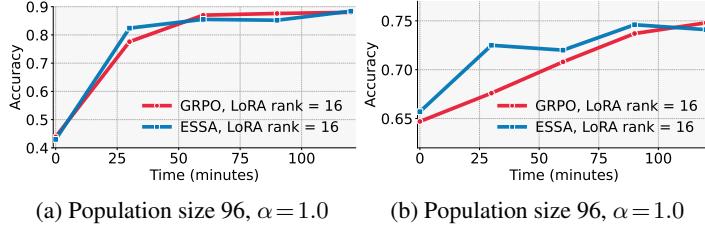


Figure 30: Validation accuracy over time on **GSM8K** with **Qwen2.5-7B** (a) and **Qwen2.5-Math-7B** (b) without SFT warm-start. ESSA (blue): train size 100. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64.

Appendix Figure 31 presents the results of Qwen2.5-3B on one GPU in addition to School Math experiments (Section 4.2).

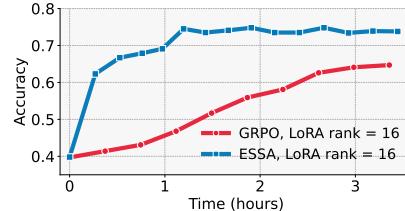
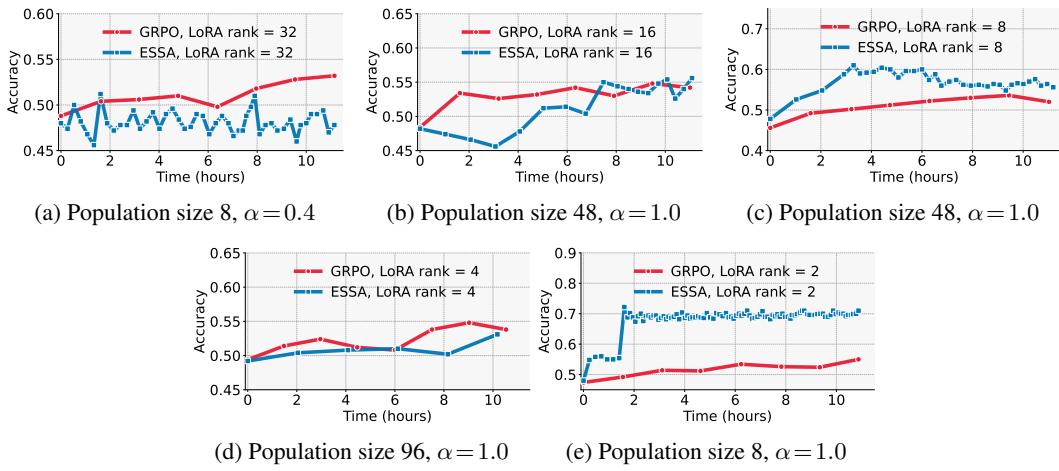
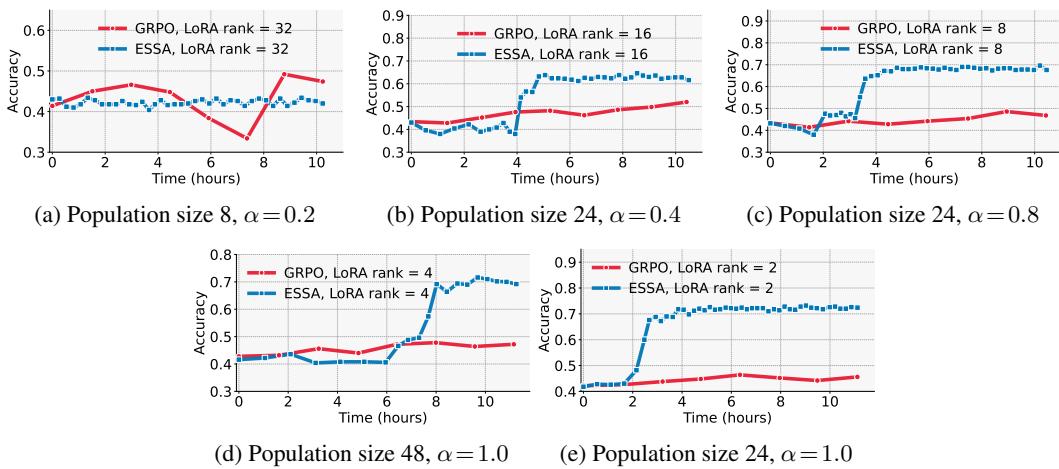
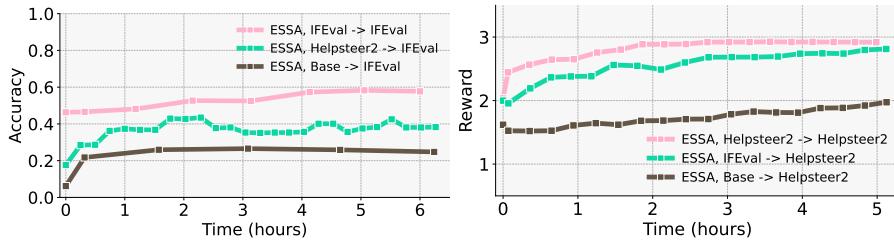


Figure 31: Validation accuracy over time on **GSM8K** with **Qwen2.5-3B** on one GPU. ESSA (blue): population size 48, $\alpha = 1.0$, batch size 100. GRPO (red): lr 1×10^{-5} , global batch 512, mini batch 64.

1620 F.2 BEYOND SCHOOL MATH
16211622 Appendix Figures 32 and 33 present the full results of our beyond school math experiments on
1623 PRM800K.
16241625 Figure 32: Validation accuracy over time on **PRM800K** with **Qwen2.5-Math-7B**. Panels (a)-(e)
1626 correspond to LoRA ranks 32, 16, 8, 4, and 2, respectively. ESSA (blue): batch size 300. GRPO
1627 (red): lr 1×10^{-5} , global batch 512, mini batch 64.
16281629 Figure 33: Validation accuracy over time on **PRM800K** with **Qwen2.5-7B**. Panels (a)-(e) corre-
1630 spond to LoRA ranks 32, 16, 8, 4, and 2, respectively. ESSA (blue): batch size 300. GRPO (red): lr
1631 1×10^{-5} , global batch 512, mini batch 64.
16321633 We evaluate both Qwen2.5-Math-7B and Qwen2.5-7B, sweeping LoRA ranks and comparing ESSA
1634 with GRPO. Across both model variants on GSM8K (Section 4.4.1, Figure 29) and PRM800K
1635 (Section 4.4.2, Section F.2) ESSA climbs to high accuracy much more rapidly than GRPO and often
1636 reaches a higher plateau, particularly when the LoRA rank is small or moderate. In experiments
1637 without SFT (Figure 34) runs starting from a random LoRA achieve lower absolute accuracy. The
1638 relative performance between ESSA and GRPO remains consistent. Experiment with Qwen2.5-3B
1639 (Figure 31) on one GPU confirms that ESSA’s benefits are not restricted to multi-GPU setups, but
1640 persist even when only one device is available.
1641

1674 G OOD DRIFT

1676 To assess the robustness of ESSA under distribution shift, we conduct an additional set of experiments that instantiate exactly the cross-domain setting proposed in the review. We select two datasets with markedly different reward structures: HelpSteer2 and IFEval. HelpSteer2 encourages general-purpose response quality, while IFEval enforces rigid constraint satisfaction. Moving between them therefore offers a direct test of how well ESSA adapts when the SFT warm-start and the target alignment domain differ. All HelpSteer2 experiments use the same reward model as in the main paper (RLHFlow/ArmoRM-Llama3-8B-v0.1). All IFEval experiments use if-eval-like dataset as a train dataset.



1684 (a) LoRA rank 8, population size 48, $\alpha = 1.0$ (b) LoRA rank 8, population size 36, $\alpha = 1.0$

1694 Figure 34: Validation metrics over time on **IFEval** (a) and **HelpSteer2** (b) with **Llama3.1-8B**. (a)
1695 IFEval → IFEval (pink), HelpSteer2 → IFEval (green), base model → IFEval (brown) experiments,
1696 batch size 500. (b): HelpSteer2 → HelpSteer2 (pink), IFEval → HelpSteer2 (green), base model →
1697 HelpSteer2 (brown) experiments, batch size 100.

1698 For each target dataset B we evaluate three configurations. The first performs SFT on dataset A and
1699 ESSA on B ($A \rightarrow B$). The second performs both SFT and ESSA on B ($B \rightarrow B$). The third applies
1700 ESSA directly to B without any SFT warm-start (Base model → B), initialized with default LoRA
1701 parameters (Kaiming-uniform for weight A and zeros for weight B).

1703 Across both directions, ESSA reliably improves performance on the target dataset, even when the
1704 SFT checkpoint originates from a mismatched domain. In the IFEval to HelpSteer2 condition the
1705 gap between cross-domain and in-domain SFT is modest, indicating that the structured initialization
1706 learned from if-eval-like dataset already provides useful axes for ESSA to exploit when optimizing
1707 for HelpSteer2. In the opposite direction the gap is larger, which is expected given that HelpSteer2
1708 does not teach the model the strict formatting and constraint-following behavior required by IFEval.
1709 Nevertheless ESSA exhibits stable monotonic improvement and substantially surpasses the Base →
1710 IFEval configuration that starts from random LoRA parameters. In both directions, SFT warm-starts
1711 clearly outperform training from scratch, confirming that even an out-of-distribution SFT checkpoint
1712 supplies meaningful structure for ESSA’s evolutionary search.

1713 These results demonstrate that ESSA maintains its effectiveness under distribution shift and that the
1714 method can leverage any reasonable SFT initialization, even when it originates from a domain with
1715 incompatible reward structure.

1716 H IMPLEMENTATION DETAILS

1719 H.1 DATASETS

1721 Dataset	1722 SFT Train Size	1723 ESSA/GRPO Train Size	1724 Validation
1725 GSM8K	1495	5978	1319
1726 PRM800K	3600	7900	500
1727 IFEval	1700	3414	541
1728 HelpSteer	19000	1324	1024

1729 Table 5: Dataset sizes across all experiments.

1728 **System Prompts**
1729

1730 • **Math-Reasoning:** Please reason step by step and place your
1731 final answer in `\boxed{}`.

1732 • **Instruction-Following:** You are a helpful, honest, and concise
1733 assistant.

1734
1735 **H.2 SFT HYPERPARAMETERS**
1736

1737 Model and Dataset	1738 SFT Batch Size	1738 Learning Rate	1738 Epochs	1738 SFT GPU Hours
1739 Qwen2.5-7B (GSM8K)	1740 16	1740 1×10^{-4}	1740 1	1740 0.18
1740 Qwen2.5-Math-7B (GSM8K)	1741 16	1741 1×10^{-4}	1741 1	1741 0.16
1741 Qwen2.5-7B (PRM800K)	1742 16	1742 5×10^{-4}	1742 1	1742 0.43
1742 Qwen2.5-Math-7B (PRM800K)	1743 16	1743 5×10^{-4}	1743 1	1743 0.40
1743 Qwen2.5-32B (PRM800K)	1744 128	1744 1×10^{-6}	1744 3	1744 9.80
1744 Qwen2.5-72B (PRM800K)	1745 128	1745 1×10^{-6}	1745 3	1745 51.97
1745 Llama3.1-8B (IF-Eval-like)	1746 16	1746 5×10^{-4}	1746 1	1746 0.50
1746 Llama3.1-8B (HelpSteer2)	1747 128	1747 1×10^{-6}	1747 3	1747 3.74

1747 Table 6: Summary of hyperparameters used for SFT across all experiments.

1748
1749 **H.3 ESSA HYPERPARAMETERS**
1750

1751 For ESSA, we performed a hyperparameter grid search covering various values of LoRA rank, pop-
1752 ulation size, fraction of singular values α , batch size, and precision (Table 7). For each experiment,
1753 the best ESSA hyperparameters were selected based on validation performance (Table 8). We use
1754 up to 128 NVIDIA H100 80GB GPUs per experiment.

1755 Parameter	1756 Values Tested	1756 Default
1756 LoRA Rank	1757 2, 4, 8, 16, 32	1757 8
1757 Population Size	1758 8, 24, 36, 48, 64, 96, 192, 400	1758 96
1758 Fraction of Singular Values (α)	1759 0.1, 0.2, 0.4, 0.6, 0.8, 1.0	1759 1.0
1759 Batch Size	1760 100, 256, 300, 500	1760 100
1760 Precision	1761 INT4, INT8, BFLOAT16	1761 BFLOAT16

1763 Table 7: ESSA hyperparameter grid and default values for ablation studies.

1764 Experiment	1765 LoRA Rank	1765 Population Size	1765 Batch Size	1765 Fraction α
1765 GSM8K (Qwen2.5-Math-7B)	1766 32	1766 96	1766 100	1766 0.1
1766 GSM8K (Qwen2.5-7B)	1767 2	1767 96	1767 100	1767 1.0
1767 PRM800K (Qwen2.5-Math-7B)	1768 2	1768 24	1768 300	1768 1.0
1768 PRM800K (Qwen2.5-7B)	1769 2	1769 48	1769 300	1769 1.0
1769 PRM800K (Qwen2.5-32B)	1770 8	1770 48	1770 256	1770 1.0
1770 PRM800K (Qwen2.5-72B, INT4)	1771 4	1771 64	1771 256	1771 1.0
1771 IFEval (LLaMA3.1-8B)	1772 8	1772 24	1772 500	1772 1.0
1772 HelpSteer (LLaMA3.1-8B)	1773 8	1773 36	1773 100	1773 1.0

1774 Table 8: Best ESSA hyperparameters for each experiment.