ESSA: Evolutionary Strategies for Scalable Alignment

Anonymous ACL submission

Abstract

Online alignment of large language models (LLMs) is dominated by reinforcement learning from human feedback (RLHF) with gradient-based optimizers such as PPO or GRPO. While effective, these pipelines require backpropagation through long rollouts, gradient synchronization across devices, and careful hyperparameter tuning, all of which become increasingly costly at scale. We present **ESSA** (Evolutionary Strategies for Scalable Alignment), a gradient-free post-SFT online alignment stage that replaces the gradient loop with inference-only black-box search. ESSA optimizes only the singular values of low-rank adapters after a short SFT warm-start, restricting the search to a compact, task-aligned subspace where evolutionary search is practical even for 72B-parameter models. Because the loop is inference-only, ESSA is compatible with INT4/INT8 weight quantization and reduces inter-GPU communication to a few bytes per iteration. Across instruction following (IFEval), preference-based assistant tuning (HelpSteer2, HH-RLHF), and mathematical reasoning (GSM8K, MATH500), ESSA matches or exceeds GRPO, Online DPO, and PPO under matched LoRA budgets, and at scale reaches a fixed MATH500 accuracy on Qwen2.5-32B/PRM800K up to 6× faster than GRPO on 128 GPUs.

1 Introduction

Aligning large language models (LLMs) to follow instructions, respect safety constraints, and produce preferred answers is now a standard step in modern NLP pipelines (Ouyang et al., 2022a; Grattafiori et al., 2024; Lambert et al., 2024). The dominant tool for online alignment is reinforcement learning from human feedback (RLHF), typically with PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024). These methods train an actor (and often a critic) through backpropagation over long rollouts, require all-reduce of model-sized gradient tensors at every step, and rely on careful tuning of learning rate, KL penalty, clipping, and batch size. As model size grows, these mechanics become a dominant practical cost: communication, memory, and engineering overhead all scale with the parameter count rather than with task difficulty (Zheng et al., 2023; Sheng et al., 2024).

We revisit a much simpler primitive: evolutionary search with only forward passes. Evolutionary strategies (ES) are gradient-free, embarrassingly parallel, communicate only random seeds and scalar rewards, and place no requirements on the inference engine beyond the ability to sample from the model (Salimans et al., 2017; Hansen and Ostermeier, 2001). Their classical limitation is the curse of dimensionality: applied naively to billions of parameters, ES is sample-inefficient and unstable. The question is whether a sufficiently compact, task-aligned parameterization exists where ES becomes a competitive replacement for the gradient loop within the modern post-SFT alignment recipe.

We answer this question affirmatively with **ESSA** (Evolutionary Strategies for Scalable Alignment). ESSA targets the online alignment stage that follows supervised fine-tuning (SFT)[1] and combines three ingredients: (i) restrict updates to LoRA adapters of the attention matrices (Hu et al., 2021); (ii) decompose each LoRA factor via SVD and freeze the singular vectors, leaving only $\mathcal{O}(r)$ singular values per factor as trainable parameters (Sun et al., 2025); (iii) optimize this compact vector with CMA-ES (Hansen and Ostermeier, 2001), evaluating each population member with a single inference pass. The resulting loop has three practical consequences (Figure 1):

[1] Standard alignment pipelines, such as InstructGPT (Ouyang et al., 2022b), Tulu 3 (Lambert et al., 2024), and Llama 3 (Grattafiori et al., 2024), apply an SFT warm-start before the RL stage, and we follow this convention. ESSA replaces only the RL stage; the SFT cost is shared with all baselines and reported separately.

- **Inference-only.** Because no backward pass is performed, ESSA runs natively under INT8/INT4 weight quantization, enabling one-candidate-per-GPU evaluation up to 72B parameters on a single 80GB device.
- **Communication-minimal.** Workers exchange a 32-bit seed and a 32-bit scalar reward per evaluation, eliminating the all-reduce of model-sized tensors that bottlenecks gradient RLHF.
- **Hyperparameter-light.** ESSA has three knobs (LoRA rank, population size, fraction α of singular values optimized) and we observe broad plateaus of stable accuracy for all three (Section 4.6).

Contributions. (1) We introduce ESSA, a gradient-free post-SFT online alignment stage built on SVD-parameterized LoRA singular values, and motivate it with a precise scope: the LoRA / cheap-reward regime that dominates modern open-source alignment. (2) We give a fair, matched-parameterization comparison against Online DPO, PPO, and full and SVD-restricted GRPO across instruction following (IFEval), preference-based assistant tuning (HelpSteer2, Anthropic HH-RLHF) and verifiable-reward math (GSM8K, MATH500); ESSA is competitive or better than all LoRA baselines and matches full-parameter GRPO on math while training $\sim 10^6\times$ fewer parameters. (3) We provide a wall-clock scaling study (16→128 GPUs) and a theoretical iteration-time bound that explains it, plus a precision study showing <1pp accuracy loss when moving from BF16 to INT4 weights.

2 Related Work

Online vs. offline alignment. Online RLHF (Ouyang et al., 2022a) with PPO (Schulman et al., 2017), REINFORCE (Sutton et al., 1999), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), REINFORCE++ (Hu, 2025), or DAPO (Yu et al., 2025) remains the standard recipe for verifiable-reward and preference-based tasks. Offline preference learning (DPO (Rafailov et al., 2024), ORPO (Hong et al., 2024), SimPO (Meng et al., 2024)) removes online rollouts but is bounded by the static preference dataset and underperforms online methods (Tang et al., 2024; Xu et al., 2024a). We benchmark against the strongest online LoRA baselines (Online DPO, PPO, GRPO) because they share ESSA's online setting; offline methods are an orthogonal design point.

Parameter-efficient and singular-value adaptation. LoRA (Hu et al., 2021), adapters (Houlsby et al., 2019), and prefix-tuning (Li and Liang, 2021) all reduce trainable parameters. A separate line of work (Transformer^2 (Sun et al., 2025), SVFT (Lingam et al., 2024), SVFit (Sun et al., 2024), and Singular Value Fine-Tuning (Sun et al., 2022)) freezes singular vectors and updates only singular values with gradients, in supervised fine-tuning settings. ESSA shares the singular-value parameterization but differs in two crucial ways. (i) *Online RL setting:* prior singular-value methods are evaluated for supervised fine-tuning, not for online policy optimization under scalar rewards. (ii) *Optimizer:* we show empirically that gradient-based optimizers fail in this parameterization at GRPO-scale RL: our SVD-GRPO ablation plateaus at $\sim$0.5 accuracy across all LoRA ranks (Section 4.2). In contrast, CMA-ES, which adapts its covariance toward the inverse Hessian (Hansen and Ostermeier, 2001), succeeds. ESSA's contribution is therefore not the parameterization itself, but the demonstration that it unlocks gradient-free RL alignment.

Evolutionary strategies for LLMs. Classical ES variants (CMA-ES (Hansen and Ostermeier, 2001), NES (Wierstra et al., 2011), ARS (Mania et al., 2018), GES (Maheswaranathan et al., 2018)) offer gradient-free, highly parallel optimization with linear scaling in the number of evaluators. Zero-order methods (Zhang et al., 2024) approximate gradients from loss differences but underperform on complex reasoning. Recent ES-for-LLM work (GENOME/GENOME+ (Zhang et al., 2025b), LoRAHub (Huang et al., 2024), DFO (Jin et al., 2024), ES-FT (Qiu et al., 2025), and EGGROLL (Sarkar et al., 2025)) either targets hyperscale full-parameter search or restricts ES to model merging (GENOME/GENOME+, LoRAHub), i.e. searching only for coefficients that combine pretrained checkpoints rather than learning new behaviour. Among ES-based methods for LLM fine-tuning (Appendix Table 5), ESSA induces the strongest dimension reduction, a key factor for ES sample efficiency (Salimans et al., 2017), and is the first ES-based method we are aware of to be applied to LLMs at the 72B parameter scale on a single 80 GB GPU. We also outperform EGGROLL and match ES-FT on Countdown while using a much smaller search space (Appendix E).

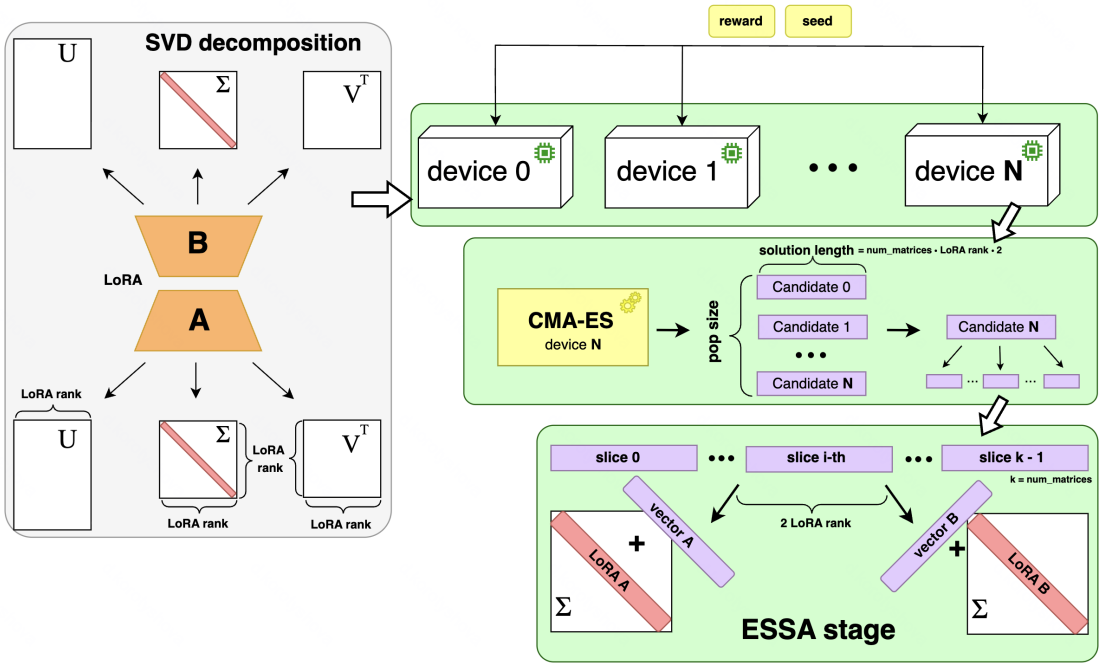


Figure 1: ESSA framework. After SFT, each LoRA factor A, B is decomposed by SVD; orthogonal bases U, V are frozen, and only the top-α fraction of the singular values is optimized by CMA-ES. Each device evaluates one candidate by reconstructing the LoRA factors and running an inference pass; workers exchange only random seeds and scalar rewards. **Slice** n denotes the contiguous block of singular values corresponding to one LoRA matrix (e.g. W_Q, W_K, W_V, W_O at a given transformer layer), and **device** N denotes the GPU worker.

3 Method

Setting and scope. We target the online alignment stage that follows SFT, where the model has already learned the task's input/output format and the goal is to push it along a scalar reward signal (a rule-based verifier or a reward model). ESSA does not attempt to replace SFT, and is not intended for from-scratch pretraining: its singular-value updates rely on the SVD basis already encoding task-relevant directions, which we verify on a controlled MNIST replica of the pipeline (Appendix Figure 6). The pipeline is therefore: (1) standard SFT on a small data fraction; (2) ESSA on a disjoint online split.

LoRA + SVD parameterization. For each attention matrix $W_0 \in \mathbb{R}^{m\times n}$ we introduce a LoRA update $\Delta W = BA$ with $B \in \mathbb{R}^{m\times r}$, $A \in \mathbb{R}^{r\times n}$, $r \ll \min(m, n)$ (Hu et al., 2021). After SFT, we SVD-decompose each factor:

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

The orthogonal bases U_A, V_A, U_B, V_B are frozen; only the top-α fraction of the singular values in Σ_A, Σ_B is exposed to the optimizer. This shrinks the trainable space per factor from $\mathcal{O}(r(m+n))$ to $\mathcal{O}(r)$.

Evolutionary optimization. Let $\sigma \in \mathbb{R}^d$ concatenate all trainable singular-value offsets ($d \approx 1{,}800$ for Qwen2.5-7B with r=8 on the four attention matrices per layer). At iteration t, CMA-ES (Hansen and Ostermeier, 2001) samples a population $\{x_t^{(k)}\}_{k=1}^{\lambda}$ from $\mathcal{N}(m_t, \sigma_t^2 C_t)$, evaluates each candidate's scalar reward F, and updates $(m_{t+1}, \sigma_{t+1}, C_{t+1})$ from the ranked rewards. Crucially, the workers share a single random-number stream, so each worker can locally regenerate any population member from a seed: communication is reduced to $\langle$seed, reward$\rangle$ pairs. The full procedure is given in Algorithm 1 (Appendix A).

Iteration-time bound. Let G be the number of GPUs, M_{params} the model size in bytes, and $\eta_{\text{gen}}, \eta_{\text{fb}}$ the per-sample generation and forward-backward times. Under mild assumptions on tensor-parallel sizes (Appendix B), the ESSA iteration is strictly faster than the idealized gradient pipeline whenever

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

The right-hand side grows linearly in M_{params} and G, so the ESSA-feasible population scales with model and cluster size (Section 4.3).

4 Experiments

Tasks and backbones. We evaluate on four task families that span the modern alignment spectrum: *(i) Instruction following:* train on an IFEval-like dataset (Xu et al., 2024b), evaluate on IFEval (Zhou et al., 2023); backbone Llama-3.1-8B (Grattafiori et al., 2024). *(ii) Preference-based assistant tuning:* train and evaluate on HelpSteer2 (Wang et al., 2024b) with the ArmoRM reward model (Wang et al., 2024a); safety/helpfulness on Anthropic HH-RLHF (Bai et al., 2022) with the OpenAssistant Deberta-v3 reward model, backbone Mistral-7B (Jiang et al., 2023). *(iii) Mathematical reasoning:* GSM8K (Cobbe et al., 2021) and PRM800K→MATH500 (Lightman et al., 2023) plus AIME'24/'25 (Zhang and Math-AI, 2024, 2025), MinervaMath (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AMC'23; backbones Qwen2.5-7B/-Math-7B/-32B/-72B (Team, 2024; Yang et al., 2024). *(iv) Symbolic puzzle:* Countdown (Appendix E).

Baselines. We compare against four online LoRA baselines: Online DPO, PPO, GRPO, and SVD-GRPO (GRPO restricted to singular-value updates), as well as full-parameter GRPO. All gradient baselines are trained with the verl library (Sheng et al., 2024) and a learning-rate sweep over {1e-6, 5e-6, 1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2}. SFT checkpoints are shared between ESSA and all baselines; the SFT cost is reported separately (Appendix H) and contributes an identical constant to every method. Unless noted, all methods run in BFLOAT16; the 72B experiments additionally report ESSA under INT4 (see Section 4.6 for the precision study).

4.1 LoRA-budget alignment

Table 1 reports the best accuracy reached by each LoRA method on GSM8K (verifiable rewards, Qwen2.5-7B) and IFEval (rule-based rewards, Llama-3.1-8B). ESSA, optimizing ∼2K–4K parameters with only forward passes, is the strongest method in the LoRA budget on GSM8K and IFEval. Full-parameter GRPO, which updates ∼2000× more parameters and requires backpropagation, is the only baseline that beats ESSA on IFEval (0.70 vs. 0.61), and even there ESSA matches it on GSM8K (0.88 vs. 0.86). Within the LoRA regime that dominates open-source alignment, ESSA is the strongest method we tested.

Method	# Params	GSM8K	IFEval
SFT only	—	0.72	0.45
Online DPO (LoRA)	13/17 M	0.76	0.56
PPO (LoRA)	26/34 M	0.74	0.57
GRPO (LoRA)	13/17 M	0.86	0.55
SVD-GRPO (LoRA, ours)	2/4 K	0.51	0.46
GRPO (full param.)	7/8 B	0.86	**0.70**
ESSA (ours)	2/4 K	**0.88**	0.61

Table 1: Best validation accuracy on GSM8K (Qwen2.5-7B) and IFEval (Llama-3.1-8B). # Params reports the number of optimized parameters for Qwen / Llama. PPO additionally stores a value head of comparable size. SVD-GRPO is GRPO restricted to ESSA's singular-value parameterization. All gradient baselines use the best learning rate from a sweep of nine values.

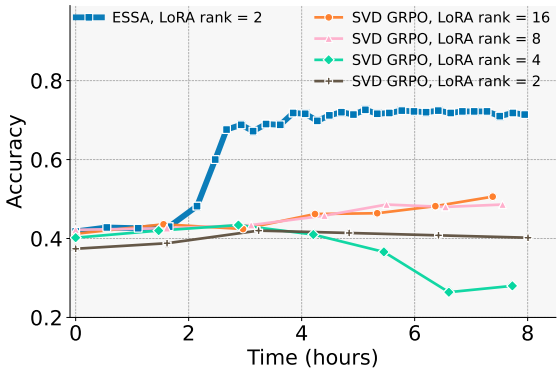


Figure 2: MATH500 accuracy over time on PRM800K with Qwen2.5-7B under the same SVD-LoRA parameterization. ESSA (blue, r=2, pop. 24, batch 300) reaches ∼0.72 while SVD-GRPO (red, $r \in \{2, 4, 8, 16\}$, best LR 1×10^{-2}) saturates below 0.5 at every rank.

4.2 Why a gradient-free optimizer? The SVD-GRPO ablation

To isolate the contribution of evolutionary search from the contribution of the singular-value parameterization, we run GRPO with exactly the same SVD-LoRA parameterization and SFT initialization as ESSA, sweeping LoRA ranks $\{2, 4, 8, 16\}$ and the learning rate over the nine values listed above. As shown in Figure 2 (and Table 1, row SVD-GRPO), GRPO restricted to singular values plateaus at ∼0.5 accuracy on PRM800K (MATH500) across all ranks, more than 20 percentage points below ESSA at the same rank.

Why does this happen? We offer a hypothesis consistent with classical optimization theory. In high-dimensional LoRA, the curvature of the RL objective is spread across many weakly correlated coordinates, and diagonal first-order methods like Adam (Kingma and Ba, 2015) suffice. Af-

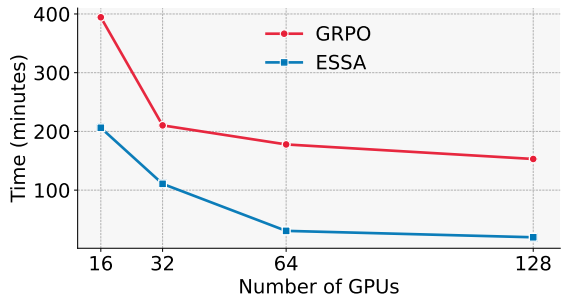


Figure 3: Wall-clock time to reach MATH500 accuracy 0.835 on Qwen2.5-32B/PRM800K. ESSA (blue): r=16, pop. 128, batch 256. GRPO (red): r=16, global batch 512. Both methods use the same SFT checkpoint and the same hardware.

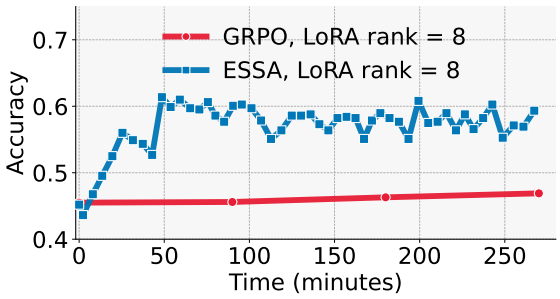


Figure 4: Validation accuracy on IFEval with Llama-3.1-8B. ESSA: r=8, pop. 24, α=1.0, batch 500. GRPO: r=8, global batch 512.

ter SVD compression, the same curvature is concentrated into a very small subspace whose conditioning becomes adversarial for first-order methods. CMA-ES adapts its full covariance matrix toward the inverse Hessian (Hansen and Ostermeier, 2001; Hansen, 2023), making it parameterization-invariant in expectation. Measuring the Hessian of an RL objective at 7B scale is itself a non-trivial research undertaking; we therefore offer the curvature argument as a hypothesis and treat the >20pp gap across all ranks and learning rates as the empirical evidence. Crucially, ESSA's value does not hinge on this hypothesis: even if a hypothetical curvature-aware gradient method matched ESSA's accuracy, it would still require backpropagation, ruling out INT4/INT8 inference-only deployment and the seed-only communication that drives ESSA's scaling story.

4.3 Parallel scaling

Figure 3 compares the wall-clock time required by GRPO and ESSA to reach a fixed MATH500 accuracy of 0.835 on Qwen2.5-32B/PRM800K, as the number of GPUs grows from 16 to 128. ESSA goes from $\sim$200 min at 16 GPUs to under 20 min at 128 GPUs ($\approx$10$\times$), while GRPO goes from $\sim$400 min to $\sim$150 min ($\approx$2.6$\times$). The gap is 2$\times$ at 16 GPUs and grows to 6$\times$ at 128 GPUs.

A note on GRPO's scaling. A natural concern is whether GRPO's flatter curve is a tuning artifact: at 128 GPUs, a global batch of 512 leaves little per-device work and may be communication-bound. We hold the GRPO batch fixed for two reasons. (1) The critical batch size literature (McCandlish et al., 2018; Merrill et al., 2025; Zhang et al., 2025a) shows that beyond a task-dependent critical batch, larger batches yield diminishing returns. (2) Xie et al. (2025) report that scaling the RL batch from 32 to 2048 yields only 0.00–0.05 Pass@1 improvement in RL fine-tuning, indicating that batch size is not a critical lever for RL alignment specifically. The structural advantage of ESSA, communicating $\mathcal{O}(1)$ bytes per evaluation versus $\mathcal{O}(M_{\text{params}})$ for gradient all-reduce, is not an artifact of GRPO's batch and is predicted by the bound in Eq. (1). We additionally verify on a single GPU with Qwen2.5-3B that ESSA's wall-clock advantage persists in the no-parallelism regime (Appendix I.1).

Reward-evaluation accounting. ESSA does pay for its forward-only loop with more reward queries. On Qwen2.5-7B/GSM8K (8$\times$H100), ESSA uses $\sim$816K reward evaluations to reach its best accuracy ($\sim$step 85, pop. 96 $\times$ 100 prompts), versus $\sim$165K for GRPO ($\sim$step 40, batch 512 $\times$ 8 rollouts). The $\sim$5$\times$ overhead is irrelevant when rewards are cheap (string matching, rule-based verification, or a small reward model), as they are throughout our experiments. In that regime ESSA still finishes 3$\times$ faster in wall-clock time on the same hardware. For expensive reward models, this overhead becomes a genuine cost that we discuss in the Limitations section.

4.4 Instruction following, preference, and safety

Instruction following (IFEval). Figure 4 compares ESSA and GRPO on Llama-3.1-8B/IFEval. ESSA reaches $\approx$0.6 accuracy within 60 minutes and stays there; GRPO climbs slowly and saturates near 0.45 after four hours. The advantage is therefore not specific to verifiable-reward math: it also holds for open-ended instruction following, where the reward signal is rule-based but non-trivial.

Method	MATH	Min.	Olym.	AI24	AI25	AMC	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 2: Qwen2.5-32B trained on PRM800K, evaluated on advanced-math benchmarks. avg@k with k=16 for AIME'24/'25/AMC'23 and k=8 otherwise.

Method	MATH	Min.	Olym.	AI24	AI25	AMC	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 3: Qwen2.5-72B (ESSA under INT4, GRPO under BF16+TP). avg@k with k=16 for AIME'24/'25/AMC'23 and k=8 otherwise. ESSA achieves competitive avg@k and slightly higher aggregate pass@8 while running one candidate per GPU.

Preference-based assistant tuning (HelpSteer2). On Llama-3.1-8B/HelpSteer2 with the ArmoRM reward model (a noisy, learned signal), ESSA and GRPO improve at comparable rates and reach comparable final reward (Appendix Figure 28), confirming that ESSA's behavior is not specific to rule-based rewards.

Safety / helpfulness (HH-RLHF). Beyond aggregate helpfulness, we extend ESSA to the safety domain by training Mistral-7B on Anthropic HH-RLHF (Bai et al., 2022) with the OpenAssistant Deberta-v3 reward model. ESSA reaches a higher in-domain reward than GRPO (Appendix Figure 29). A GPT-4o side-by-side judge on a held-out set of safety/helpfulness prompts reports **12%** preference for ESSA, **76%** ties, and **12%** preference for GRPO, confirming parity on an out-of-distribution preference judgment that did not produce the training reward signal.

4.5 Scaling to 32B/72B on advanced math

Tables 2 and 3 report avg@k and pass@8 on six advanced-math benchmarks with Qwen2.5-32B and Qwen2.5-72B. ESSA matches or exceeds GRPO on the aggregate scores, including on AIME'24/'25 where backbone capacity matters most. At 72B we run ESSA under INT4 with one candidate per GPU; GRPO requires tensor parallelism for the backward pass.

4.6 Sensitivity and precision

Hyperparameters. A grid sweep over LoRA rank, population size, and α for five (backbone, task) pairs (full plots in Appendix F) reveals three consistent trends: (i) accuracy plateaus for $r \geq 2$ in most settings; (ii) populations of 24–96 are sufficient; (iii) accuracy is stable for $\alpha \geq 0.4$. ESSA therefore avoids the multi-knob sensitivity (KL coefficient, clipping, learning rate, mini-batch size) of PPO/GRPO.

Precision. Because the loop is inference-only, ESSA runs natively under INT8/INT4 weight quantization. On Qwen2.5-32B/PRM800K (Table 4), moving from BF16 to INT4 costs less than 1 percentage point of MATH500 accuracy; this is what allows the 72B model to fit on a single 80GB GPU per candidate.

Precision	BF16	INT8	INT4
Max MATH500 accuracy	0.847	0.844	0.838

Table 4: ESSA on Qwen2.5-32B/PRM800K under three precisions (r=8, pop. 64, α=1, batch 256).

5 Discussion

ESSA shows that the modern post-SFT alignment stage admits a gradient-free realization that is competitive in quality, friendly to quantization, and substantially better-suited to multi-GPU scaling. The three design choices (restricting to LoRA, parameterizing by singular values, and optimizing with CMA-ES) are individually old, but combined they turn online alignment into a problem where the per-iteration cost is dominated by inference rather than by gradient communication. Our matched-budget comparison against Online DPO, PPO, and GRPO (full and SVD-restricted) shows that ESSA is the strongest LoRA method on instruction following and verifiable-reward math, and parity with GRPO on a preference-based safety task as judged by GPT-4o. The SVD-GRPO ablation (Section 4.2) further isolates the gradient-free optimizer as the essential ingredient: the parameterization alone is not enough.

Limitations

SFT initialization is required. ESSA is positioned as a post-SFT alignment stage and relies on the structure that SFT introduces into the LoRA singular vectors; without any SFT, ESSA still improves over the base model but does not reach the SFT→ESSA accuracy (Appendix Figure 32). This dependence is shared by essentially all modern alignment pipelines (Ouyang et al., 2022b; Lambert et al., 2024; Grattafiori et al., 2024), but it does mean that ESSA does not replace SFT.

Reward-evaluation cost. ESSA uses roughly $5\times$ more reward evaluations than GRPO to reach the same accuracy. With cheap rewards (rule-based verifiers, small reward models) this is more than offset by the savings on the backward pass; with very expensive reward models, the higher evaluation count could erode the wall-clock advantage. The break-even point depends on the ratio between reward model cost and policy model cost and is a useful subject for future work.

Low-rank assumption. The singular-value parameterization assumes the alignment update lies close to a low-rank subspace seeded by SFT. This holds across the eight (backbone, task) pairs we evaluate and is consistent with the broader LoRA literature, but it is a real assumption: tasks that genuinely require a high-rank update may not be reachable by ESSA at the LoRA ranks we explored.

Curvature argument is a hypothesis. Our explanation for why SVD-GRPO fails while ESSA succeeds (Section 4.2) is offered as a hypothesis grounded in classical optimization theory and the >20pp empirical gap across ranks and learning rates. We did not directly measure Hessian condition numbers at GRPO scale; doing so is itself a non-trivial research project.

Backbone diversity. We covered Qwen-2.5 (7B/Math-7B/32B/72B), Llama-3.1-8B, and Mistral-7B, but a broader study across additional families (Gemma, Phi, Olmo) would further strengthen the generality claim.

Ethical Considerations

ESSA is a generic alignment method; it inherits the well-known dual-use considerations of any RLHF-style technique. Because ESSA reduces the engineering cost of online alignment, it may lower the barrier to both beneficial and harmful applications. Our HH-RLHF and HelpSteer2 experiments use only publicly available, license-compatible datasets and reward models, and our reward signals correspond to helpfulness/harmlessness preferences released by their original authors. We do not introduce any new dataset or release any new model weights with this work. We did not perform any human-subjects evaluation. The GPT-4o side-by-side comparison in Section 4.4 was performed via API access in compliance with OpenAI's usage policies. All datasets (GSM8K, PRM800K, IFEval, HelpSteer2, HH-RLHF, AIME, AMC, MinervaMath, OlympiadBench), backbone models (Qwen2.5, Llama-3.1, Mistral-7B), reward models (ArmoRM, OpenAssistant Deberta-v3), and software (verl, vLLM, EvoTorch) are released under permissive licenses (MIT, Apache 2.0, CC-BY-4.0) or their respective model community licenses (Llama 2/3), and our academic-research use complies with each artifact's terms.

Use of AI assistants. We used AI assistants (general-purpose LLMs) for grammar checks and short-form text polishing throughout the manuscript, and as coding aides for plotting and table formatting. All scientific claims, experiments, analyses, and design decisions were authored by the human authors, and we take full responsibility for the content.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur

Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Nikolaus Hansen. 2023. The cma evolution strategy: A tutorial. *Preprint*, arXiv:1604.00772.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. *Preprint*, arXiv:2403.07691.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

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

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

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2024. Lorahub: Efficient cross-task generalization via dynamic lora composition. *Preprint*, arXiv:2307.13269.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and 1 others. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Feihu Jin, Yin Liu, and Ying Tan. 2024. Derivative-free optimization for low-rank adaptation in large language models. *Preprint*, arXiv:2403.01754.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *Preprint*, arXiv:2101.00190.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Vijay Lingam, Atula Bhaskar, Hong Yin, Aran Komatsuzaki, Pavlo Molchanov, and Hieu Phan. 2024. SVFT: Parameter-efficient fine-tuning with singular vectors. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Niru Maheswaranathan, Luke Metz, Dami Choi, George Tucker, and Jascha Sohl-Dickstein. 2018. Guided evolutionary strategies: escaping the curse of dimensionality in random search.

Horia Mania, Aurelia Guy, and Benjamin Recht. 2018. Simple random search provides a competitive approach to reinforcement learning. *Preprint*, arXiv:1803.07055.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. 2018. An empirical model of large-batch training. *Preprint*, arXiv:1812.06162.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *Preprint*, arXiv:2405.14734.

William Merrill, Alexander Wettig, Jacob Goldstein, and 1 others. 2025. Critical batch size revisited: A simple empirical approach to large-batch language model training. *Preprint*, arXiv:2505.23971.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. 2025. Evolution strategies at scale: Llm fine-tuning beyond reinforcement learning. *Preprint*, arXiv:2509.24372.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. 2017. Evolution strategies as a scalable alternative to reinforcement learning. *Preprint*, arXiv:1703.03864.

Bidipta Sarkar, Mattie Fellows, Juan Agustin Duque, Alistair Letcher, Antonio León Villares, Anya Sims, Dylan Cope, Jarek Liesen, Lukas Seier, Theo Wolf, Uljad Berdica, Alexander David Goldie, Aaron Courville, Karin Sevegnani, Shimon Whiteson, and Jakob Nicolaus Foerster. 2025. Evolution strategies at the hyperscale. *Preprint*, arXiv:2511.16652.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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

Qi Sun, Edoardo Cetin, and Yujin Tang. 2025. Transformer-squared: Self-adaptive llms. *Preprint*, arXiv:2501.06252.

Yanpeng Sun, Qiang Chen, Xiangyu He, Jian Wang, Haocheng Feng, Junyu Han, and 1 others. 2022. Singular value fine-tuning: Few-shot segmentation requires few-parameters fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 37484–37496.

Yiren Sun, Zerui Li, Zihan Liu, and Bo Chen. 2024. SVFit: Parameter-efficient fine-tuning of large pretrained models using singular values. *Preprint*, arXiv:2409.05926.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, and Will Dabney. 2024. Understanding the performance gap between online and offline alignment algorithms. *Preprint*, arXiv:2405.08448.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

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

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024b. Helpsteer2: Open-source dataset for training top-performing reward models. *Preprint*, arXiv:2406.08673.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, and Jürgen Schmidhuber. 2011. Natural evolution strategies. *Preprint*, arXiv:1106.4487.

Tian Xie and 1 others. 2025. Reinforcement fine-tuning naturally mitigates forgetting. *Preprint*, arXiv:2503.02875.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. 2024a. Is dpo superior to ppo for llm alignment? a comprehensive study. *Preprint*, arXiv:2404.10719.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024b. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *Preprint*, arXiv:2406.08464.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, and 1 others. 2025a. How does critical batch size scale in pre-training? *Preprint*, arXiv:2410.21676.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. 2024. Revisiting zeroth-order optimization for

memory-efficient llm fine-tuning: A benchmark. *Preprint*, arXiv:2402.11592.

Yiqun Zhang, Peng Ye, Xiaocui Yang, Shi Feng, Shufei Zhang, Lei Bai, Wanli Ouyang, and Shuyue Hu. 2025b. Nature-inspired population-based evolution of large language models. *Preprint*, arXiv:2503.01155.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, and 8 others. 2023. Secrets of rlhf in large language models part i: Ppo. *Preprint*, arXiv:2307.04964.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A ESSA Algorithm

Algorithm 1 formalises the full ESSA training loop sketched in Section 3: SVD initialisation of the LoRA factors from the SFT checkpoint, parallel candidate sampling across N workers, fitness evaluation by a single forward pass per candidate, and a CMA-ES update per generation. Workers exchange only $\langle \text{seed}, F \rangle$ pairs, keeping per-step communication constant in model size.

Algorithm 1 ESSA: Distributed Evolutionary Search over LoRA Singular Values

Require: Training set $\mathcal{D}$; SFT-initialized LoRA factors $\{A_i, B_i\}$; fraction α; population P; GPUs N; generations E.

SVD init: for each A_i, B_i compute $A_i = U_{A_i}\Sigma_{A_i}V_{A_i}^\top$, $B_i = U_{B_i}\Sigma_{B_i}V_{B_i}^\top$; freeze U, V; concatenate top-α singular values into $\sigma \in \mathbb{R}^d$.
Initialize N workers with identical seeds and a common CMA-ES state (m_0, σ_0, C_0).
for $t = 0, \ldots, E-1$ **do**
  for worker $k = 1, \ldots, N$ **in parallel do**
    Sample candidate offsets $\{\delta^{(k,j)}\}$ from shared seeds.
    for each local candidate j **do**
      Reconstruct $A_i' = U_{A_i}\mathrm{diag}(\sigma_{A_i} + \delta_{A_i})V_{A_i}^\top$ and similarly B_i'; set $W_i' = W_i + B_i'A_i'$.
      Sample mini-batch $\mathcal{D}_{\text{mini}} \subset \mathcal{D}$ and compute reward F.
    end for
  end for
  All workers exchange $\langle \text{seed}, F \rangle$ pairs.
  Each worker reconstructs the same population and performs one CMA-ES update of (m, σ, C).
end for
Return aligned model with final singular values.

B Detailed Iteration-Time Analysis

This appendix expands the iteration-latency analysis sketched in Section 4.3, comparing one optimization step of an idealized gradient-based RLHF pipeline (e.g. PPO/GRPO) against one ESSA step. Gradient methods incur an all-reduce of a model-sized tensor, whereas ESSA exchanges constant-size information (seeds and scalar rewards). The proposition below isolates a sufficient condition under which ESSA is strictly faster per iteration.

Proposition 1 (ESSA iteration-time advantage)**.** *If the ESSA population size N_{pop} and per-candidate batch size B^{essa} satisfy*

$$N_{\text{pop}} B^{\text{essa}} < B^{\text{grad}}\left(1 + \frac{\eta_{\text{fb}}}{G\,\eta_{\text{gen}}}\right) + \frac{(G-1)\,M_{\text{params}}}{\text{peak_bw}\,G\,\eta_{\text{gen}}}, \tag{2}$$

then the ESSA iteration time is strictly smaller than that of the idealized gradient pipeline: $T^{\text{essa}} < T^{\text{grad}}$.

To prove the proposition we (i) introduce notation, (ii) bound per-iteration computation and communication, (iii) optimize the device split between training and generation, and (iv) derive a conservative population-size bound under which ESSA is guaranteed to be faster.

Notation. Let B^{grad} be the global batch size used in gradient methods, B^{essa} the batch size processed by a single population instance, $b_{\text{fb}}, b_{\text{gen}}$ the microbatch sizes, $m_{\text{fb}}, m_{\text{gen}}$ GPUs per model 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 forward-backward and generation microbatch times, $\eta_{\text{fb}} = \tau_{\text{fb}}(b_{\text{fb}})/b_{\text{fb}}$ and $\eta_{\text{gen}} = \tau_{\text{gen}}(b_{\text{gen}})/b_{\text{gen}}$ the per-sample times, and $k_{\text{fb}}^{\text{par}} = G_{\text{fb}}/m_{\text{fb}}$, $k_{\text{gen}}^{\text{par}} = G_{\text{gen}}/m_{\text{gen}}$ the numbers of microbatches processed in parallel. Then $T^{\text{grad}} = T_{\text{fb-gen}}^{\text{grad}} + T_{\text{comm}}^{\text{grad}}$ and $T^{\text{essa}} = T_{\text{gen}}^{\text{essa}} + T_{\text{comm}}^{\text{essa}}$, with asynchronous training and generation giving $T_{\text{fb-gen}}^{\text{grad}} = \max(T_{\text{fb}}^{\text{grad}}, T_{\text{gen}}^{\text{grad}})$.

Computation. Processing B^{grad} samples requires $B^{\text{grad}}/b_{\text{fb}}$ microbatches, of which $k_{\text{fb}}^{\text{par}}$ run in parallel:

$$T_{\text{fb}}^{\text{grad}} = \frac{B^{\text{grad}}\,m_{\text{fb}}}{G_{\text{fb}}}\,\eta_{\text{fb}},$$
$$T_{\text{gen}}^{\text{grad}} = \frac{B^{\text{grad}}\,m_{\text{gen}}}{G_{\text{gen}}}\,\eta_{\text{gen}}.$$

For ESSA, the entire population is evaluated by generation only:

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

Communication. Let M_{params} be the model-parameter size in bytes. Gradient methods all-reduce gradients in two collectives (reduce-scatter + all-gather), each moving M_{params}/G bytes across $G-1$ devices at sustained per-GPU bandwidth peak_bw:

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

ESSA communicates only a 32-bit random seed and a 32-bit scalar reward via one all-gather, $M_{\text{essa}} = 8$ bytes:

$$T^{\text{essa}}_{\text{comm}} = \frac{M_{\text{essa}}(G-1)}{G\,\text{peak_bw}}.$$

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

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

Lemma 2 (Optimal split)**.** *The minimum of* $T^{\text{grad}}_{\text{fb-gen}}(\theta)$ *over* $\theta \in (0,1)$ *is attained at*

$$\theta^\star = \frac{m_{\text{fb}}\eta_{\text{fb}}}{m_{\text{fb}}\eta_{\text{fb}} + m_{\text{gen}}\eta_{\text{gen}}},$$
$$T^{\text{grad}}_{\text{fb-gen}}(\theta^\star) = \frac{B^{\text{grad}}}{G}(m_{\text{fb}}\eta_{\text{fb}} + m_{\text{gen}}\eta_{\text{gen}}).$$

Proof. $\max(a/\theta, b/(1-\theta))$ is minimized when both arguments are equal: $a/\theta = b/(1-\theta)$ with $a = m_{\text{fb}}\eta_{\text{fb}}$, $b = m_{\text{gen}}\eta_{\text{gen}}$. □

Define the ideal gradient iteration time under perfect scheduling:

$$T^{\text{grad}}_\star = \frac{B^{\text{grad}}}{G}(m_{\text{fb}}\eta_{\text{fb}} + m_{\text{gen}}\eta_{\text{gen}}) + 2 \cdot \frac{M_{\text{params}}(G-1)}{G\,\text{peak_bw}}.$$

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

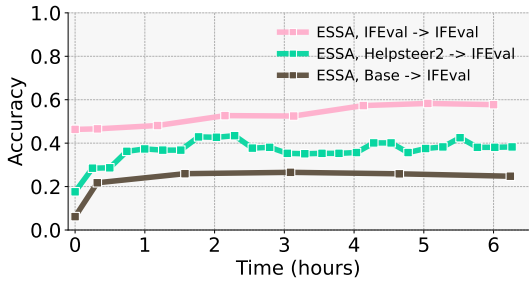
Theorem 3 (ESSA iteration is faster under a conservative bound)**.** *Assume* $m_{\text{fb}} \geq 1$, $m_{\text{gen}} \leq G$, *and* $2M_{\text{params}} - M_{\text{essa}} \geq M_{\text{params}}$. *If the population size satisfies* (2)*, then* $T^{\text{essa}} < T^{\text{grad}}_\star \leq T^{\text{grad}}$.

Proof. Starting from $T^{\text{essa}} < T^{\text{grad}}_\star$ and substituting the per-component expressions above, then applying the assumed bounds on $m_{\text{fb}}, m_{\text{gen}}$, and M_{essa}, yields inequality (2). □

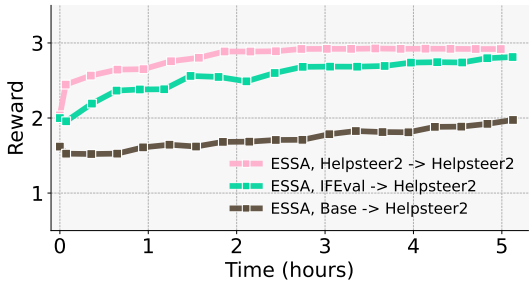
Discussion. Theorem 3 gives a conservative threshold on the ESSA population size below which it is guaranteed to be faster than the idealized gradient pipeline. Since real clusters rarely achieve the perfect split $\theta^\star$ and typically operate below nominal peak_bw, the practical advantage of ESSA is often even larger than predicted by (2). The right-hand side grows linearly in M_{params} and in G, so the ESSA-feasible population grows with model and cluster size; this matches the empirical scaling in Section 4.3.

C Cross-Domain Robustness

To assess robustness under distribution shift between SFT and ESSA, we instantiate the four cells $\{\text{HelpSteer2}, \text{IFEval}\}^2$ on Llama-3.1-8B. For each target dataset B we run: (i) in-domain $B \to B$, (ii) cross-domain $A \to B$ ($A \neq B$), and (iii) Base$\to B$ (random LoRA, no SFT). Across both directions, ESSA reliably improves the target metric even when the SFT checkpoint comes from a different domain (Figure 5). In the IFEval→HelpSteer2 direction the cross-domain gap is small; in the HelpSteer2→IFEval direction the gap is larger but cross-domain SFT still substantially outperforms no-SFT. This demonstrates that ESSA can re-use any reasonable SFT checkpoint and does not require a perfectly task-matched warm-start.



(a) Target: IFEval



(b) Target: HelpSteer2

Figure 5: Cross-domain ESSA on Llama-3.1-8B. In-domain (pink), cross-domain (green), and no-SFT (brown) configurations.

D Singular-Value Adaptation Requires Pretrained Structure (MNIST)

We verify the intuition that singular-value-only optimization is effective precisely when the singular vectors already encode meaningful directions. In a minimal logistic-regression replica of the LLM pipeline (pretrain on MNIST digits 0–5; SFT on digits 6–9; align via singular values), ESSA tracks SGD when there is a meaningful SFT update or pretrained structure (Figure 6a,b) and lags when the matrix is random Gaussian (Figure 6c). This

isolates the role of pretrained + SFT structure, the same condition that holds for LLMs, and explains why ESSA succeeds in the LLM setting despite its restricted parameter space.

E Other ES Methods on Countdown

We position ESSA against the two closest ES baselines for LLM fine-tuning: ES-FT (Qiu et al., 2025), which performs full-parameter ES, and EGGROLL (Sarkar et al., 2025), which performs ES over LoRA coefficients. Table 5 summarises their search dimensions and task coverage. Countdown is the only task common to all three methods, and Figure 7 reports the head-to-head comparison on Qwen2.5-7B / 8 GPUs.

Method	Search dim.	Tasks
ESSA (ours)	$\mathcal{O}(r)$	GSM8K, PRM800K, IFEval, HelpSteer2, HH-RLHF, Countdown
ES-FT (Qiu et al., 2025)	$\mathcal{O}(nm)$	Countdown, MATH, ARC-AGI, Sudoku
EGGROLL (Sarkar et al., 2025)	$\mathcal{O}(r(n+m))$	Countdown, GSM8K, MATH500, AIME, OlympiadBench, AMC

Table 5: ES-based methods for LLM fine-tuning. ESSA's search dimension is $\mathcal{O}(r)$ per LoRA factor, orders of magnitude smaller than full-parameter ES (ES-FT, $\mathcal{O}(M_{\text{params}})$) and LoRA-coefficient ES (EGGROLL, $\mathcal{O}(r(m{+}n))$ per matrix). Concretely, for Qwen2.5-7B (28 layers, four attention matrices per layer) at $r{=}8$: ESSA exposes $28 \times 4 \times 2r \approx 1.8K$ trainable singular values (two SVD factors A, B per matrix, each contributing r values), ES-FT searches the full $\sim$7B model parameters, and EGGROLL searches $28 \times 4 \times r(m{+}n) \approx 5M$ LoRA coefficients.

F Hyperparameter Sensitivity

For each (backbone, task) pair we report a 2-D heatmap over LoRA rank and population size, and a 1-D sweep over the fraction α of trainable singular values at the best (rank, pop.) cell. The bottom-left cell at $r{=}8, \alpha{=}0.1$ is marked N/A because $\lfloor r\alpha \rfloor = 0$ singular values are trainable. Figures 8–12 present the full sweep for all five (backbone, task) pairs.

The five sweeps confirm three trends used in the main text: (i) accuracy peaks at low LoRA ranks (typically $r \in \{2, 4, 8\}$) and degrades for over-large adapters; (ii) populations of 24–96 are sufficient, with diminishing returns beyond 96; (iii) accuracy is stable across $\alpha \geq 0.4$, indicating broad insensitivity to the SVD sparsification.

F.1 Training Dynamics

Figures 13–26 report full training-dynamics for each (backbone, task) pair, sweeping the population size at fixed LoRA rank and vice versa. These curves complement the static heatmaps above by showing how ESSA validation accuracy evolves over time as each knob is varied.

F.1.1 Qwen2.5-7B on GSM8K

Figures 13, 14, and 15 sweep population size, LoRA rank, and singular-value fraction α respectively, holding the other two knobs at their best-cell values.

F.1.2 Qwen2.5-Math-7B on GSM8K

Figures 16, 17, and 18 report the analogous population, rank, and α sweeps for the math-specialised Qwen2.5-Math-7B backbone on GSM8K.

F.1.3 Qwen2.5-7B on PRM800K

Figures 19, 20, and 21 show the rank, population, and α sweeps for Qwen2.5-7B on the harder PRM800K (MATH500) task.

F.1.4 Qwen2.5-Math-7B on PRM800K

Figures 22, 23, and 24 repeat the rank, population, and α sweeps on the math-specialised backbone.

F.1.5 Llama-3.1-8B on IFEval

Figures 25 and 26 present the rank and population-size sweeps for Llama-3.1-8B on IFEval.

G Precision Analysis

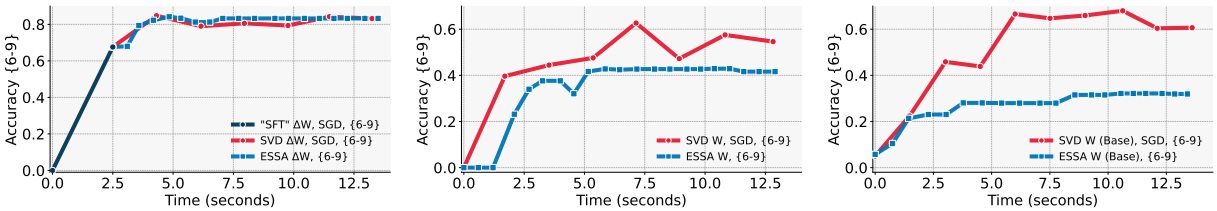
Figure 27 reports the full training dynamics behind the precision sweep in Section 4.6, comparing ESSA on Qwen2.5-32B/PRM800K under BF16, INT8, and INT4 weight quantisation.

These curves confirm that ESSA can safely exploit aggressive weight quantization to cut memory and compute costs while maintaining near-identical performance, which is what enables the 72B/INT4/one-candidate-per-GPU configuration reported in Section 4.5.

H SFT Cost and Implementation Details

Datasets.

SFT cost. SFT is shared between ESSA and all baselines and is a much smaller share of the total alignment cost than online RL (Ouyang et al., 2022b). Table 7 reports the SFT GPU-hours per (backbone, task) pair; these are constants that



(a) Pretrain + SFT (b) Pretrain only (c) Random init

Figure 6: MNIST toy: ESSA (blue) vs. SGD (red) on singular values of $W \in \mathbb{R}^{784\times 10}$.

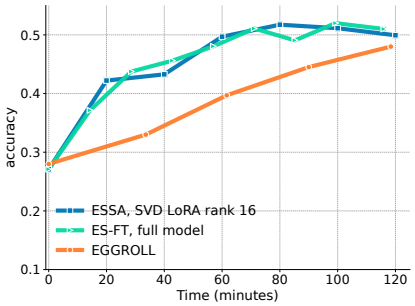
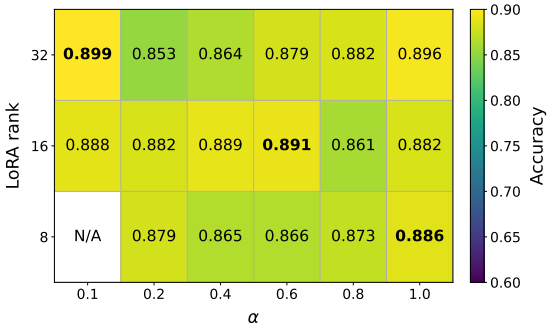


Figure 7: Countdown with Qwen2.5-7B on 8 GPUs. ESSA (blue) matches the much larger ES-FT (green, full-parameter ES) and clearly outperforms EGGROLL (orange), confirming that the dimension reduction in Table 5 translates into a real efficiency gain.

Dataset	SFT	ESSA/GRPO	Validation
GSM8K	1.5K	6.0K	1.3K
PRM800K	3.6K	7.9K	500
IFEval-like	1.7K	3.4K	541
HelpSteer2	19K	1.3K	1.0K
HH-RLHF	5K	10K	1.0K
Countdown	—	1.0K	1.0K

Table 6: Disjoint SFT and ESSA/GRPO splits.

(a) LoRA rank vs. population size



(b) Fraction α at the best (rank, pop.) cell

Figure 8: Qwen2.5-Math-7B on GSM8K.

should be added uniformly to every method's wall-clock number.

Best ESSA hyperparameters. For each (backbone, task) pair we select the best (rank, pop., α) from the grid in Appendix F.

HelpSteer2 dynamics. For completeness, Figure 28 reports the HelpSteer2 reward over time at three LoRA ranks. ESSA and GRPO improve at similar rates and reach comparable final reward.

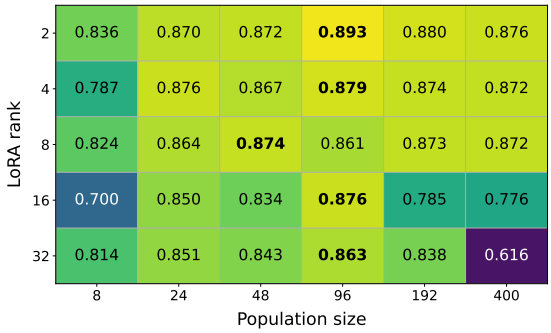
HH-RLHF dynamics. Figure 29 shows the in-domain reward over time for Mistral-7B trained on Anthropic HH-RLHF with the OpenAssistant Deberta-v3 reward model. ESSA crosses GRPO within the first hour and reaches a higher plateau by the end of training.

System prompts.

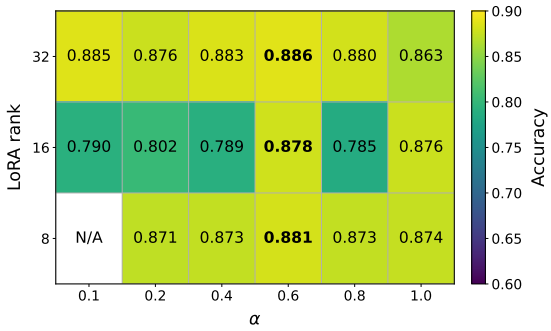
- **Math reasoning:** `Please reason step by step and place your final answer in \boxed{}.`
- **Countdown:** `You are a helpful assistant.`
- **Instruction following:** `You are a helpful, honest, and concise assistant.`

ESSA hyperparameter grid. For ESSA we performed a hyperparameter grid search over the values in Table 9. For each experiment the best hyperparameters were selected by validation performance.

Software. Gradient baselines (GRPO, Online DPO, PPO, full-parameter GRPO, SVD-GRPO) are trained with the verl library (Sheng et al., 2024).

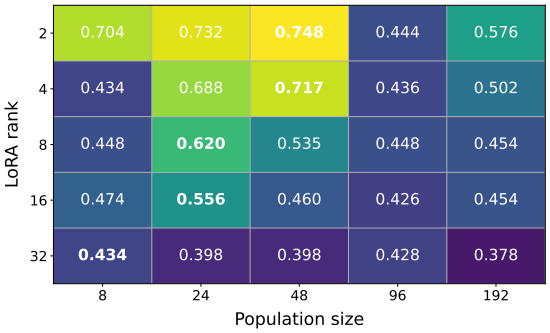


(a) LoRA rank vs. population size

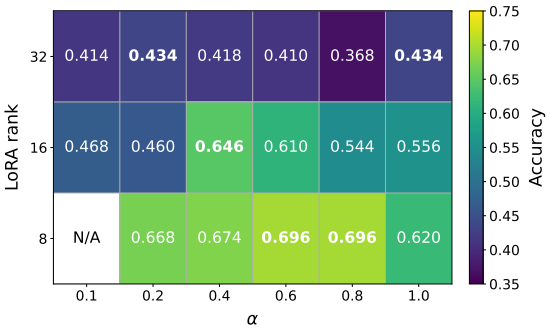


(b) Fraction α at the best cell

Figure 9: Qwen2.5-7B on GSM8K. Performance peaks at r=2, pop. 96 (acc. 0.893); accuracy is within ∼2pp across $\alpha \in [0.2, 1.0]$.



(a) LoRA rank vs. population size



(b) Fraction α at the best cell

Figure 10: Qwen2.5-7B on PRM800K (evaluated on MATH500). Accuracy peaks at r=2, pop. 48 (acc. 0.748); moderate $\alpha \in [0.4, 0.8]$ is best.

Backbone (task)	Epochs	GPU-hrs
Qwen2.5-7B (GSM8K)	1	0.18
Qwen2.5-Math-7B (GSM8K)	1	0.16
Qwen2.5-7B (PRM800K)	1	0.43
Qwen2.5-Math-7B (PRM800K)	1	0.40
Qwen2.5-32B (PRM800K)	3	9.80
Qwen2.5-72B (PRM800K)	3	51.97
Llama-3.1-8B (IFEval-like)	1	0.50
Llama-3.1-8B (HelpSteer2)	3	3.74

Table 7: SFT GPU-hours (H100-80GB).

ESSA is implemented on top of vLLM (for inference) and EvoTorch (for CMA-ES). All experiments use NVIDIA H100 80GB GPUs.

I Full GRPO Comparison Grids

This appendix collects the full per-rank ESSA-vs-GRPO comparisons for the school-math (GSM8K) and advanced-math (PRM800K) tasks, complementing the representative panels in Sections 4.1 and the no-SFT / single-GPU controls discussed below.

I.1 Qwen2.5-7B on GSM8K

Figure 30 compares ESSA and GRPO on Qwen2.5-7B/GSM8K across LoRA ranks 32, 16, 8, 4, 2. Figure 31 shows the single-GPU Qwen2.5-3B/GSM8K run: ESSA converges faster than GRPO even without parallelism. Figure 32 shows ESSA without any SFT warm-start; both methods reach lower absolute accuracy, but ESSA's relative behavior versus GRPO is preserved, confirming that the comparison is not an artifact of SFT initialization.

I.2 Advanced Math (PRM800K) Grids

Figures 33 and 34 present the full per-rank ESSA-vs-GRPO comparison for PRM800K (evaluated on MATH500) on Qwen2.5-Math-7B and Qwen2.5-7B respectively. Across both backbones, ESSA climbs to high accuracy much more rapidly than GRPO and reaches an equal or higher plateau, especially at small to moderate LoRA ranks.

(a) LoRA rank vs. population size

(b) Fraction α at the best cell

Figure 11: Qwen2.5-Math-7B on PRM800K (evaluated on MATH500). Accuracy peaks at r=2, pop. 24 (acc. 0.784); intermediate α values are competitive but slightly lower.

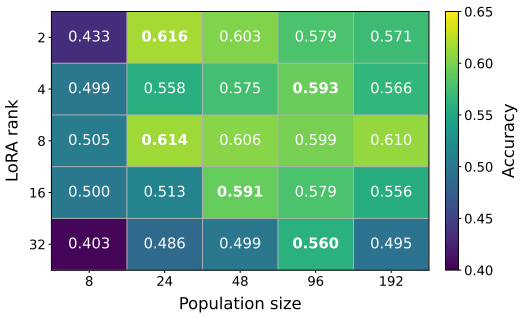


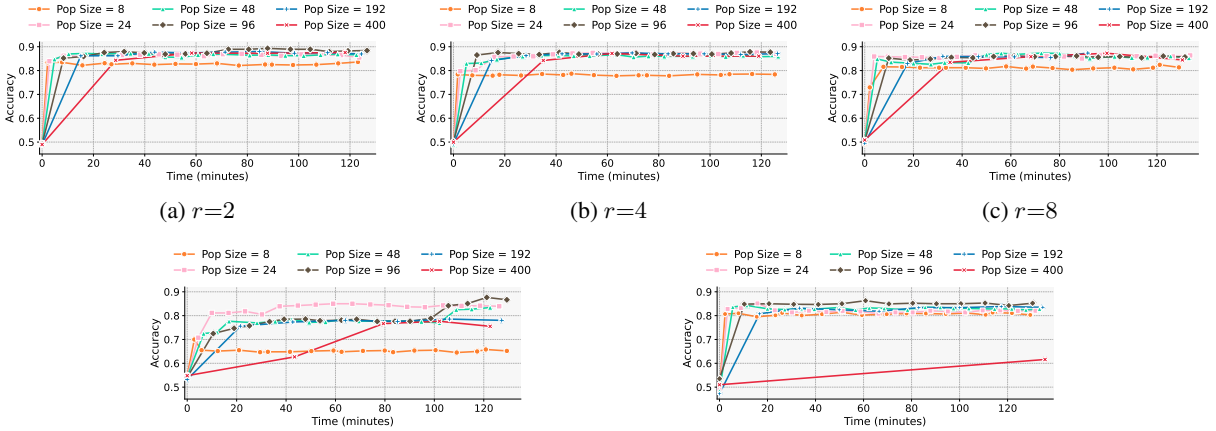
Figure 12: Llama-3.1-8B on IFEval. Peak accuracy 0.614 at r=8, pop. 24; low and medium ranks ($r \in \{4, 8, 16\}$) all perform comparably.

Experiment	r	Pop.	Batch	α
GSM8K (Qwen-Math-7B)	32	96	100	0.1
GSM8K (Qwen-7B)	2	96	100	1.0
PRM800K (Qwen-Math-7B)	2	24	300	1.0
PRM800K (Qwen-7B)	2	48	300	1.0
PRM800K (Qwen-32B)	8	48	256	1.0
PRM800K (Qwen-72B, INT4)	4	64	256	1.0
IFEval (Llama-8B)	8	24	500	1.0
HelpSteer2 (Llama-8B)	8	36	100	1.0
HH-RLHF (Mistral-7B)	8	32	200	1.0
Countdown (Qwen-7B)	16	48	200	1.0

Table 8: Best ESSA hyperparameters per experiment. The CMA-ES initial step size is fixed to $\sigma_0 = 0.32$ throughout.

Parameter	Values tested	Default
LoRA rank (r)	2, 4, 8, 16, 32	8
Population	8, 24, 36, 48, 64, 96, 192, 400	96
Fraction α	0.1, 0.2, 0.4, 0.6, 0.8, 1.0	1.0
Batch size	100, 256, 300, 500	100
Precision	INT4, INT8, BF16	BF16
CMA-ES σ_0	0.32	0.32

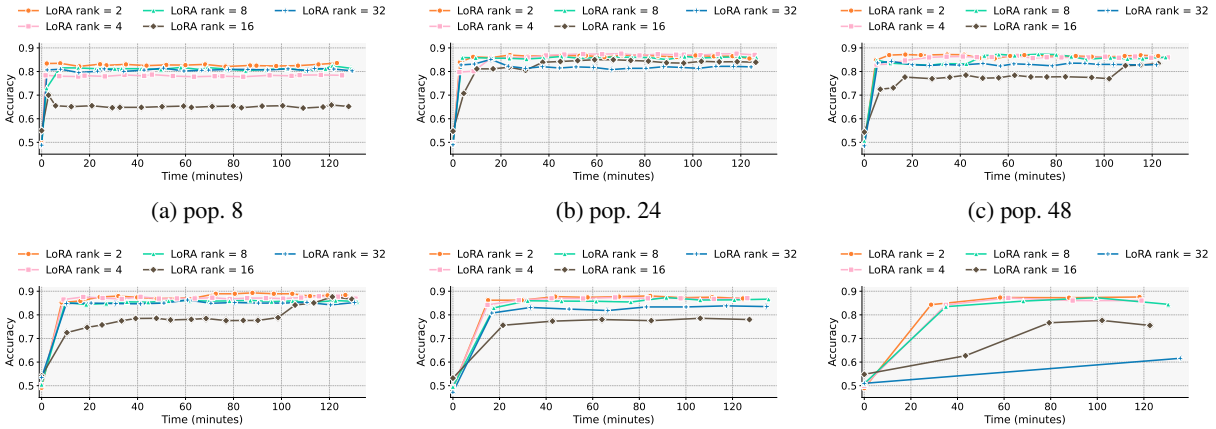
Table 9: ESSA grid for ablation studies. Up to 128 NVIDIA H100 80GB GPUs were used per experiment.



(a) r=2 (b) r=4 (c) r=8

(d) r=16 (e) r=32

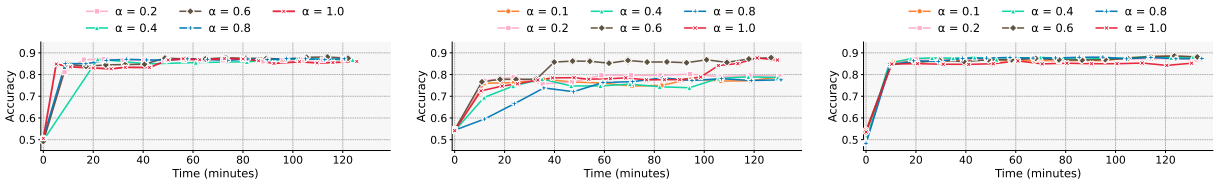
Figure 13: GSM8K with Qwen2.5-7B: validation accuracy vs. time as ESSA population size varies (8, 24, 48, 96, 192, 400) at fixed LoRA ranks. Batch size 100.



(a) pop. 8 (b) pop. 24 (c) pop. 48

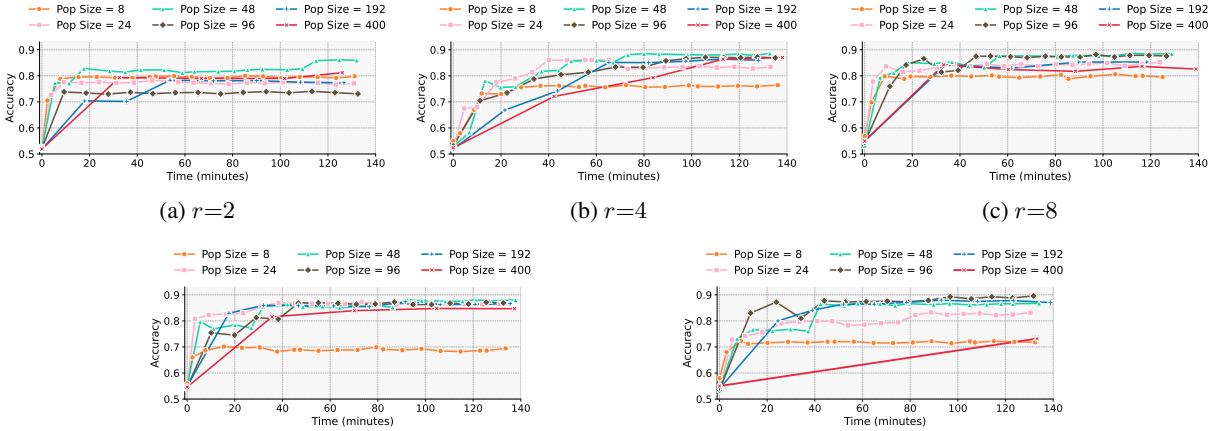
(d) pop. 96 (e) pop. 192 (f) pop. 400

Figure 14: GSM8K with Qwen2.5-7B: validation accuracy vs. time as LoRA rank varies (2, 4, 8, 16, 32) at fixed population sizes. Batch size 100.



(a) r=8, pop. 48 (b) r=16, pop. 96 (c) r=32, pop. 96

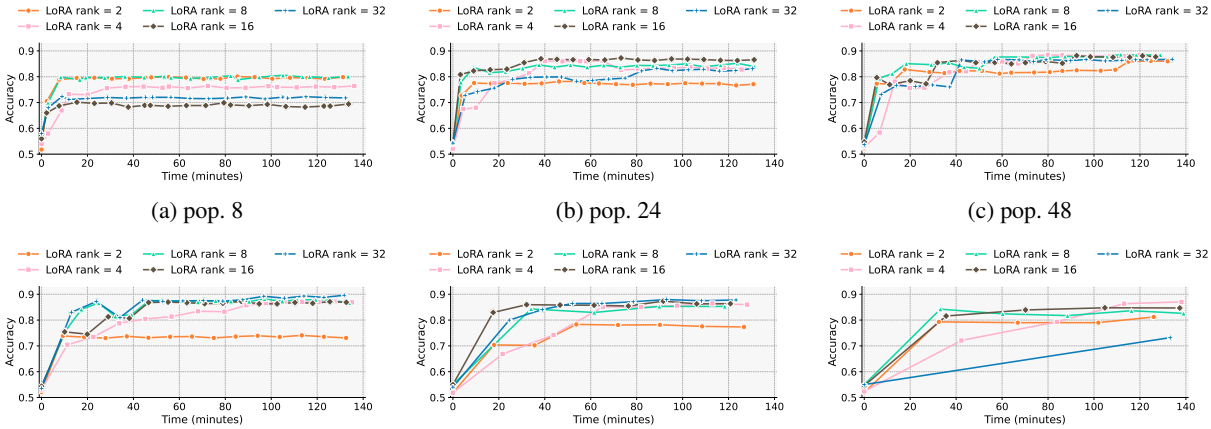
Figure 15: GSM8K with Qwen2.5-7B: validation accuracy vs. time while varying α. Rank and population size are fixed to the best cells in Figure 9.



(a) r=2 (b) r=4 (c) r=8

(d) r=16 (e) r=32

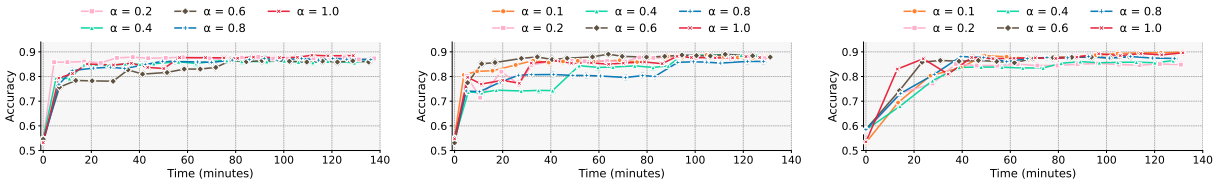
Figure 16: GSM8K with Qwen2.5-Math-7B: validation accuracy vs. time as ESSA population size varies at fixed LoRA ranks. Batch size 100.



(a) pop. 8 (b) pop. 24 (c) pop. 48

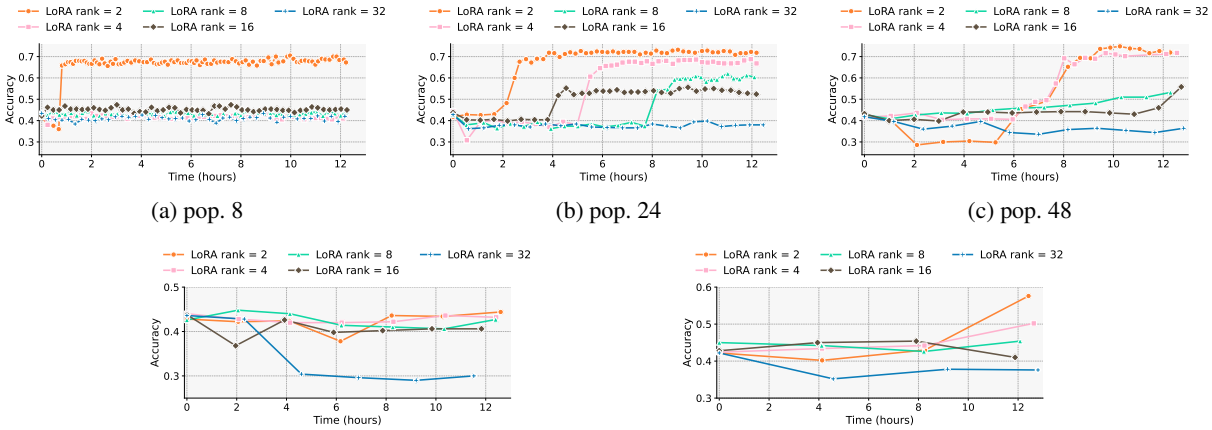
(d) pop. 96 (e) pop. 192 (f) pop. 400

Figure 17: GSM8K with Qwen2.5-Math-7B: validation accuracy vs. time as LoRA rank varies at fixed population sizes. Batch size 100.



(a) r=8, pop. 48 (b) r=16, pop. 48 (c) r=32, pop. 96

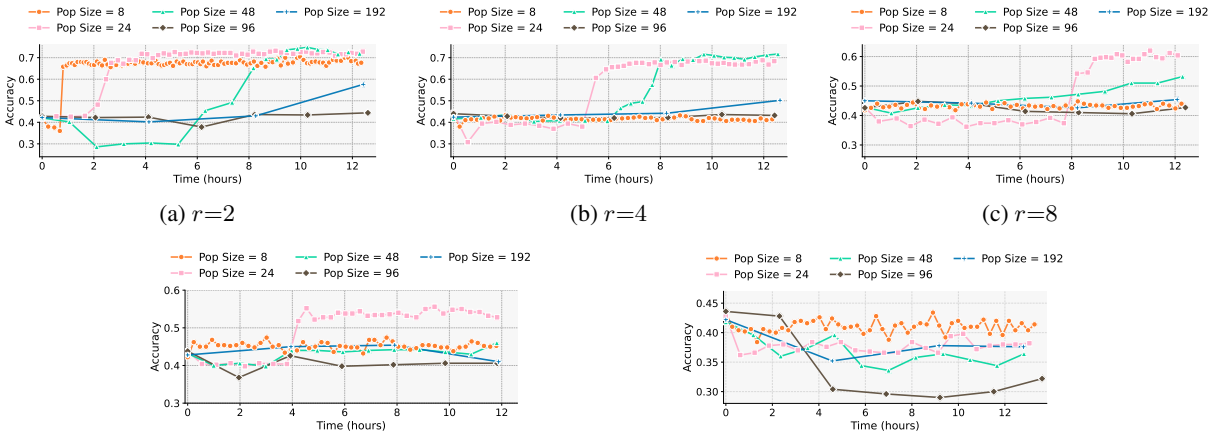
Figure 18: GSM8K with Qwen2.5-Math-7B: validation accuracy vs. time while varying α, at the best (rank, pop.) cells.



(a) pop. 8 (b) pop. 24 (c) pop. 48

(d) pop. 96 (e) pop. 192

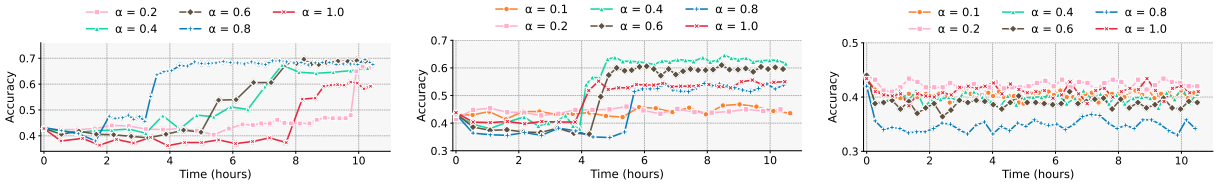
Figure 19: PRM800K (eval. MATH500) with Qwen2.5-7B: validation accuracy vs. time as LoRA rank varies at fixed population sizes. Batch size 300.



(a) r=2 (b) r=4 (c) r=8

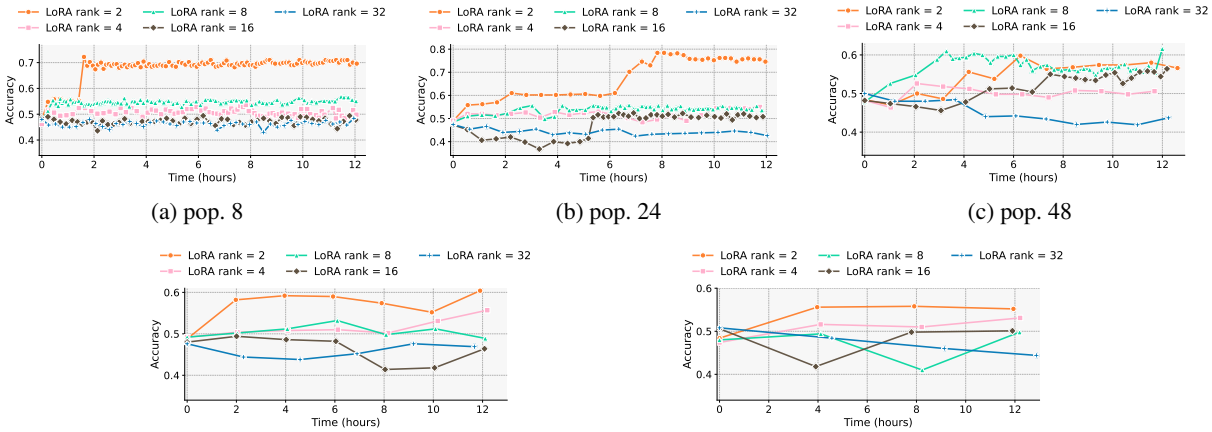
(d) r=16 (e) r=32

Figure 20: PRM800K (eval. MATH500) with Qwen2.5-7B: validation accuracy vs. time as population size varies at fixed LoRA ranks. Batch size 300.



(a) r=8, pop. 24 (b) r=16, pop. 24 (c) r=32, pop. 8

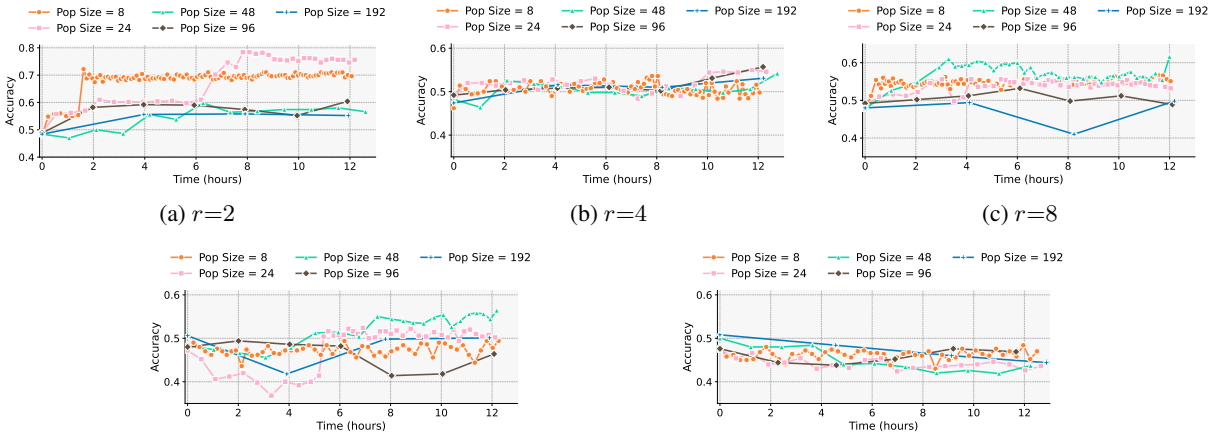
Figure 21: PRM800K (eval. MATH500) with Qwen2.5-7B: validation accuracy vs. time while varying α, at the best (rank, pop.) cells.



(a) pop. 8 (b) pop. 24 (c) pop. 48

(d) pop. 96 (e) pop. 192

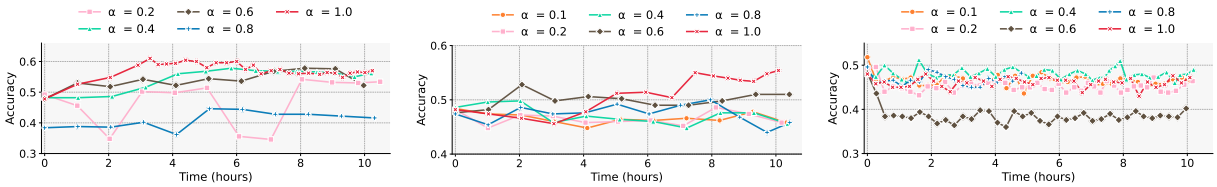
Figure 22: PRM800K (eval. MATH500) with Qwen2.5-Math-7B: validation accuracy vs. time as LoRA rank varies at fixed population sizes. Batch size 300.



(a) r=2 (b) r=4 (c) r=8

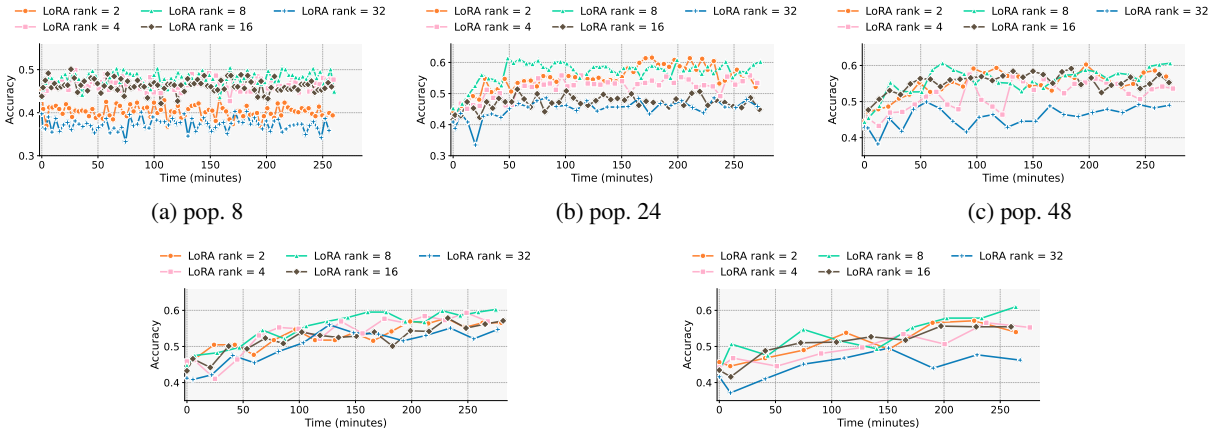
(d) r=16 (e) r=32

Figure 23: PRM800K (eval. MATH500) with Qwen2.5-Math-7B: validation accuracy vs. time as population size varies at fixed LoRA ranks. Batch size 300.



(a) r=8, pop. 48 (b) r=16, pop. 48 (c) r=32, pop. 8

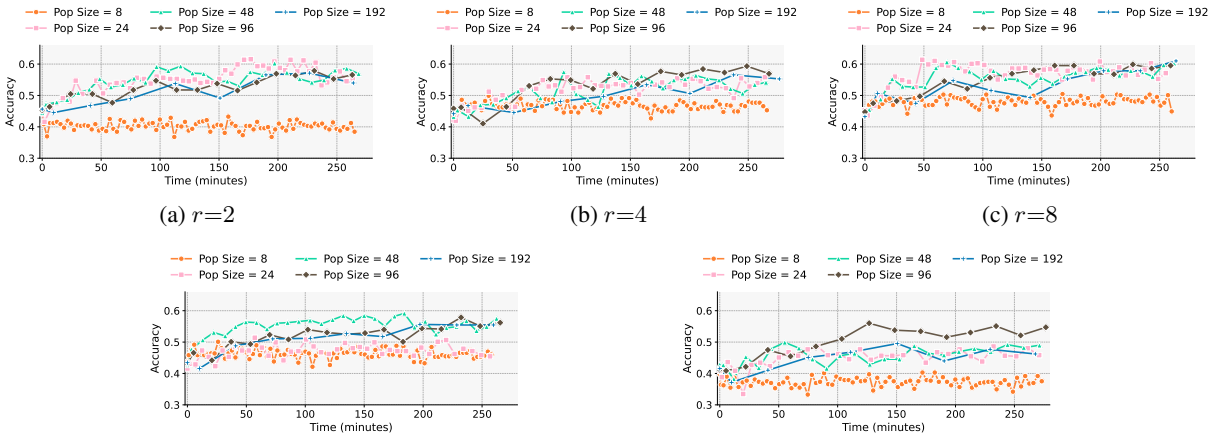
Figure 24: PRM800K (eval. MATH500) with Qwen2.5-Math-7B: validation accuracy vs. time while varying α, at the best (rank, pop.) cells.



(a) pop. 8 (b) pop. 24 (c) pop. 48

(d) pop. 96 (e) pop. 192

Figure 25: IFEval with Llama-3.1-8B: validation accuracy vs. time as LoRA rank varies at fixed population sizes. Batch size 500, α=1.0.



(a) r=2 (b) r=4 (c) r=8

(d) r=16 (e) r=32

Figure 26: IFEval with Llama-3.1-8B: validation accuracy vs. time as population size varies at fixed LoRA ranks. Batch size 500, α=1.0.

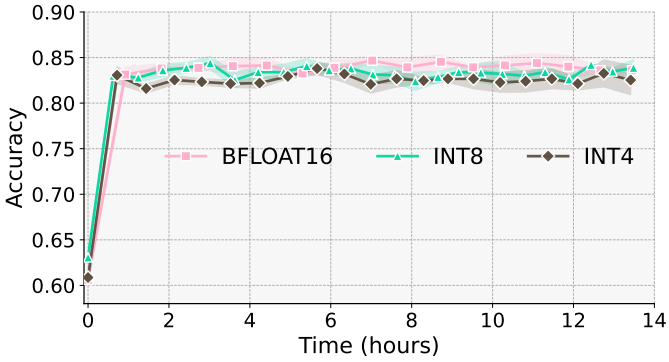
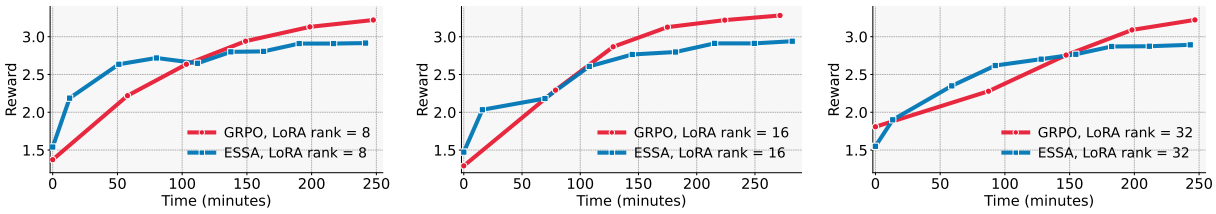


Figure 27: Full training curves for ESSA on Qwen2.5-32B/PRM800K (evaluated on MATH500) under BF16, INT8, and INT4 weight quantization. Settings: r=8, pop. 64, α=1.0, batch size 256. Shaded areas indicate one standard deviation across runs. Accuracy rises within the first hour and remains stable; INT4 trails BF16 by less than one percentage point at convergence (consistent with Table 4).



(a) r=8 (b) r=16 (c) r=32

Figure 28: HelpSteer2 reward on Llama-3.1-8B for $r \in \{8, 16, 32\}$. ESSA (blue) and GRPO (red) reach comparable final reward.

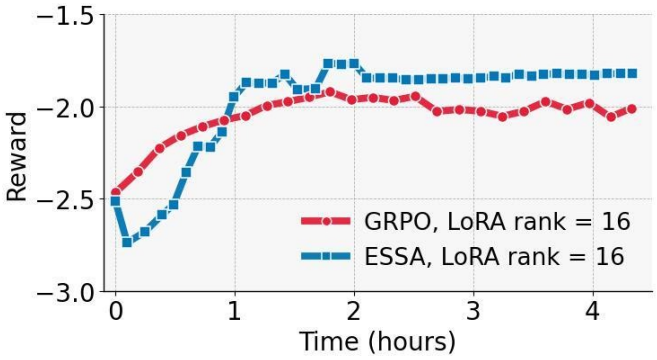
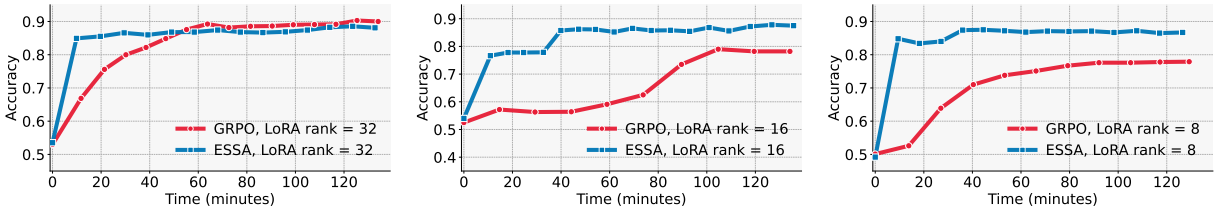
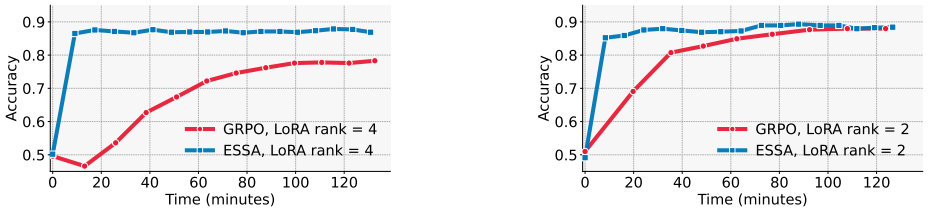


Figure 29: HH-RLHF reward over wall-clock time on Mistral-7B (LoRA r=16). ESSA (blue) reaches a higher in-domain reward than GRPO (red).



(a) r=32 (b) r=16 (c) r=8



(d) r=4 (e) r=2

Figure 30: GSM8K with Qwen2.5-7B: validation accuracy vs. time at LoRA ranks 32, 16, 8, 4, 2. ESSA (blue, batch 100) vs. GRPO (red, lr 10^{-5}, global batch 512, mini batch 64).

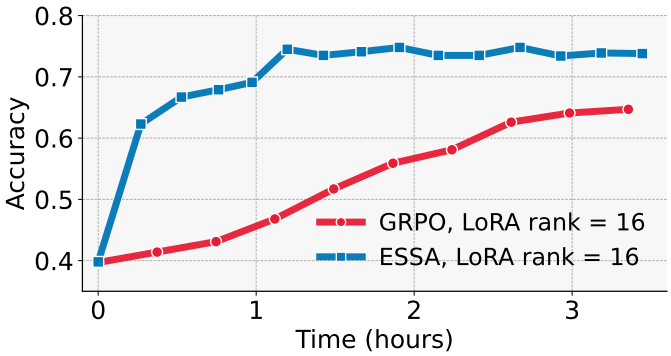
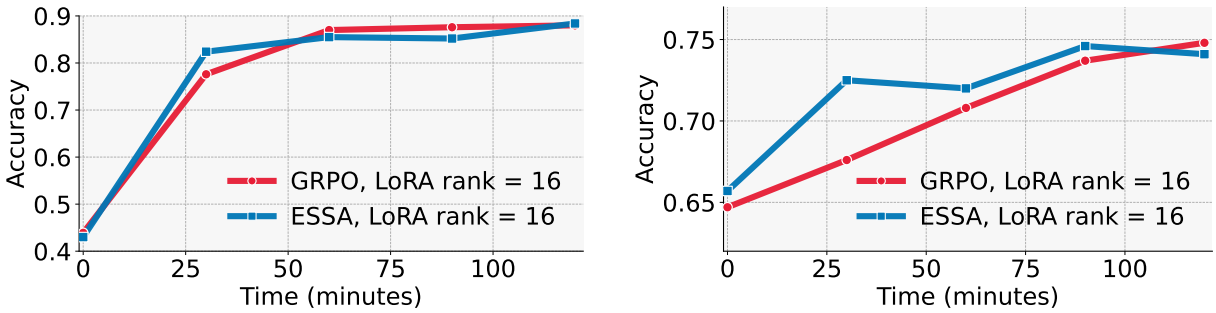
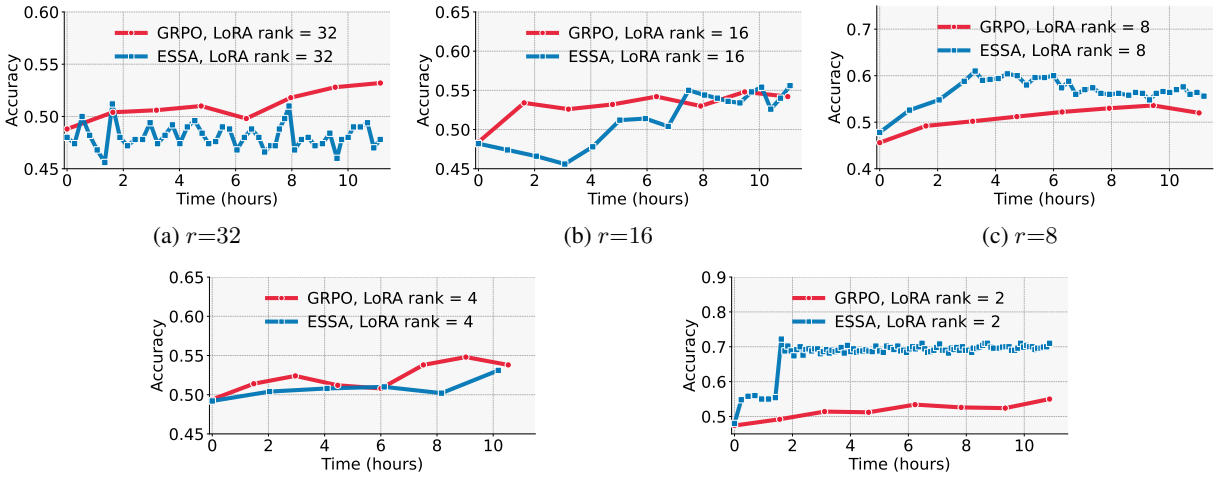


Figure 31: Single-GPU Qwen2.5-3B/GSM8K: ESSA (blue, pop. 48) converges faster than GRPO (red).



(a) Qwen2.5-7B (b) Qwen2.5-Math-7B

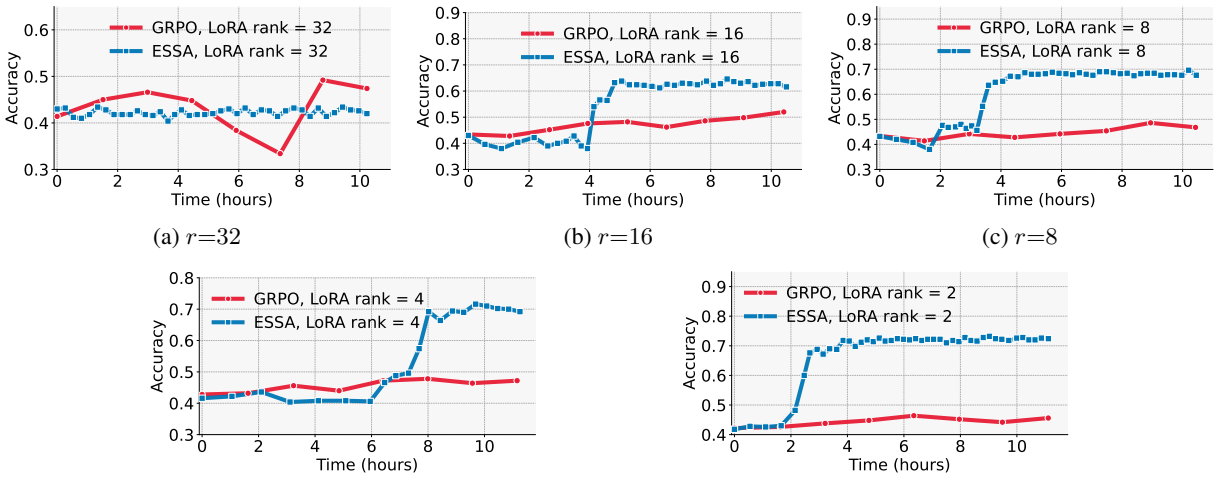
Figure 32: GSM8K without SFT warm-start (A Kaiming init, B=0). The relative ordering of ESSA and GRPO is preserved, showing that the comparison is not an artifact of SFT initialization.



(a) r=32 (b) r=16 (c) r=8

(d) r=4 (e) r=2

Figure 33: PRM800K with Qwen2.5-Math-7B: validation accuracy vs. time at LoRA ranks 32, 16, 8, 4, 2. ESSA (blue, batch 300) vs. GRPO (red, lr 10^{-5}, global batch 512, mini batch 64).



(a) r=32 (b) r=16 (c) r=8

(d) r=4 (e) r=2

Figure 34: PRM800K with Qwen2.5-7B: validation accuracy vs. time at LoRA ranks 32, 16, 8, 4, 2. ESSA (blue, batch 300) vs. GRPO (red, lr 10^{-5}, global batch 512, mini batch 64).