

CE-GAD: Curriculum-Efficient On-Policy Adversarial Distillation of Large Language Models

Wenhui Zhu^{*1}, Hejian Sang^{*1}, Xiwen Chen^{*2}, Zhipeng Wang¹,
Shayan Mohajer Hamidi¹, Han Yu¹, Jingjing Wang², Yuanda Xu¹,
Xuanzhao Dong³, Zhengze Zhou¹, Ran He¹, Jelena Markovic-Voronov¹,
Kayhan Behdin¹, Sayan Ghosh¹, Alborz Geramifard¹

¹ LinkedIn, ² Clemson University, ³ Arizona State University.

^{*} Equal contribution. [†] Correspondence: wenhzhu@linkedin.com, xiwenchen.cs@outlook.com

Abstract

Generative adversarial distillation (GAD) provides on-policy feedback for black-box distillation by training a discriminator that ranks teacher and student responses to the same prompt. While effective, GAD is also costly: every step pays for a discriminator update, an actor update over the full response, and the indirect cost of recovering whenever the discriminator and the actor progress at incompatible rates. We argue that the right way to make GAD efficient is not to make individual updates cheaper but to decide *when each update is worth its cost*. To this end, we introduce curriculum-efficient GAD (CE-GAD), an adaptive recipe that monitors a small set of training signals and uses them to (i) skip discriminator updates that would not change the reward, (ii) constrain actor updates and prune per-token gradients to the subset that actually carries signal, and (iii) rescale rewards and normalize advantages to preserve within-batch contrast under saturation. Evaluations across four models from two open-source families (Qwen2.5 at 3B/7B/14B and Llama-3.2-3B) demonstrate that CE-GAD matches or surpasses vanilla GAD on quality while winning on 12 out of 16 individual benchmark configurations. Specifically, on Qwen2.5-7B, CE-GAD improves the average score by up to +3% relative compared to the base model, while reducing per-step wall-clock time by up to 8× relative to vanilla GAD.

1 Introduction

Knowledge distillation (KD) is a paradigm for compressing resource-intensive large language models (LLMs) into smaller, deployable student models while retaining much of the original predictive performance (Hinton et al., 2015). In traditional white-box distillation, the student has full access to the teacher’s internal probability distributions or hidden states. This allows for direct alignment of the models, typically by matching output distributions via Kullback-Leibler (KL) divergence (Hin-

ton et al., 2015) or by aligning intermediate representations (Sun et al., 2019; Wang et al., 2020).

However, white-box access is increasingly impractical, as state-of-the-art teachers are often proprietary models secured behind APIs (e.g., GPT-5). Under this restrictive black-box distillation setting, the teacher is accessible solely through sampled textual responses. The absence of fine-grained, token-level probabilities renders classical logit-based distillation (Hinton et al., 2015) inapplicable. To adapt, the standard offline baseline known as sequence-level knowledge distillation (SeqKD) (Kim and Rush, 2016a) trains the student directly on *cached* teacher outputs. Yet, this purely *offline* paradigm completely deprives the student of on-policy feedback, leaving compounding errors and distributional drift uncorrected during inference (Agarwal et al., 2024b).

To overcome this limitation, it is necessary to incorporate on-policy generation into the black-box distillation process. Generative adversarial distillation (GAD) (Ye et al., 2025) provides a natural framework for bridging this gap. In this paradigm, a discriminator evaluates and scores both teacher and student/actor¹ responses to the same prompt. These scores are then utilized as a reward signal, which the student learns to maximize through reinforcement learning (RL). Yet, GAD is severely bottlenecked by extreme computational overhead and training instability. Specifically, the training pipeline is cumbersome, as each iteration requires a fragile three-step process: (i) updating the discriminator, (ii) generating full-length on-policy student rollouts, and (iii) performing the student RL update.

This instability stems from a static, alternating update schedule that presumes a symmetric optimization problem when it is in fact deeply asym-

¹We use the terms “student” and “actor” interchangeably throughout this paper.

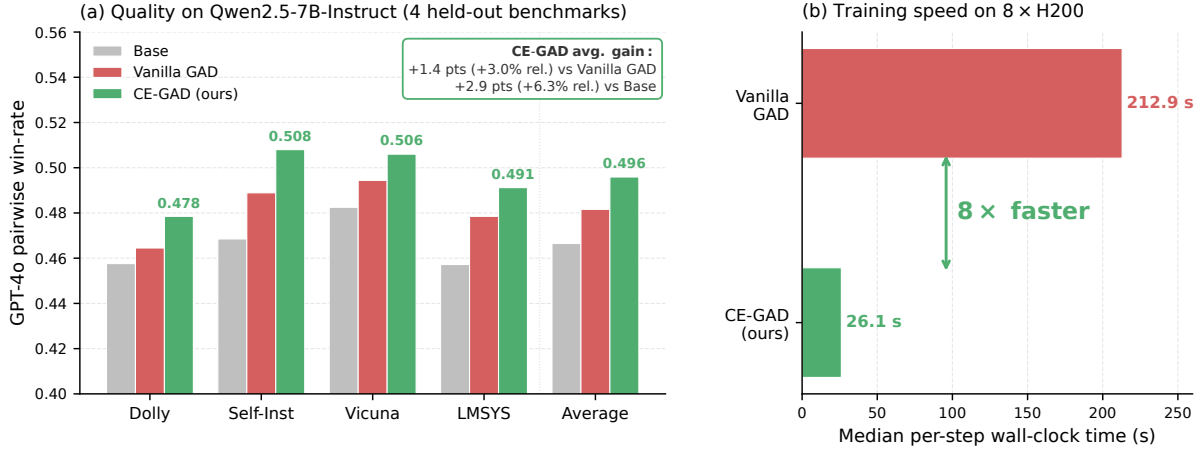


Figure 1: CE-GAD improves both quality and training cost on Qwen2.5-7B-Instruct. **(a)** GPT-4o pairwise win-rate across four held-out benchmarks plus their average (y-axis truncated at 0.40 for legibility). On average, CE-GAD improves +1.4 pts (+3.0% relative) over vanilla GAD and +2.9 pts (+6.3% relative) over the base model. Gains generalize across model families and benchmarks; see Table 1, where CE-GAD wins on 12/16 benchmark cells. **(b)** Median per-step wall-clock time on 8×H200: 8× speed-up over vanilla GAD.

metric. Adversarial dynamics are inherently non-stationary (Arjovsky and Bottou, 2017): discriminator informativeness, policy divergence, and reward calibration all drift during training, and a rigid schedule cannot react. We identify three resulting failure modes. *(i) Information vanishing.* A generator-scale discriminator faces a much simpler sequence-level ranking task and quickly locks into a high-accuracy band (0.7–0.9); subsequent backward passes yield near-zero information gain at full per-step cost. *(ii) Policy drift, token sparsity, and length collapse.* Optimizing against this saturated boundary pushes the actor outside its KL trust region and induces severe length collapse, while gradient mass concentrates in a minority of high-signal tokens, leaving most of each roll-out’s backward pass uninformative. *(iii) Reward de-alignment.* Even when ranking is preserved, the absolute teacher–student reward gap inflates by an order of magnitude; on skewed batches, the resulting global shift flattens within-batch advantage variance and erases the contrastive signal the actor needs.

Additionally, the above-mentioned failures translate directly into *wasted compute*: a saturated discriminator still pays for a full backward pass carrying no new information, and a length-collapsed student still incurs the full $\mathcal{O}(T)$ cost of backpropagating through tokens that lack any useful signal. GAD’s inefficiency is therefore not inherent to adversarial training but a symptom of *algorithmic redundancy*: much of the per-step budget is spent

on updates the current state has rendered uninformative.

To address these failure modes, we propose **curriculum-efficient generative adversarial distillation (CE-GAD)**, transitioning the paradigm from static scheduling to adaptive resource allocation. The name reflects a curriculum learning intuition: rather than presenting all training signals at uniform intensity, CE-GAD progressively adjusts the difficulty and frequency of each optimization component based on the learner’s current state. By monitoring real-time training signals (discriminator accuracy, policy divergence, and advantage statistics), CE-GAD introduces three mechanisms, each targeting one failure mode along an orthogonal axis of the training loop: *when* to update the discriminator, *whether* to update the actor, and *how* the reward shapes that update. *(i) Lightweight and adaptive scheduling* targets *information vanishing* by skipping discriminator backward passes once classification accuracy saturates. *(ii) Gated stability optimization* targets *policy drift, token sparsity, and length collapse* via a multi-trigger freeze gate, an adaptive trust-region penalty, and advantage-based gradient pruning. *(iii) Invariant reward and GRPO alignment* targets *reward de-alignment* by damping inflated reward magnitudes and restoring within-batch contrast through adaptive advantage normalization. Extensive experiments across the Qwen2.5 (3B, 7B, 14B) and Llama-3.2-3B families (Grattafiori et al., 2024) show that CE-GAD matches or exceeds vanilla

GAD’s instruction-following quality while delivering a per-step wall-clock speedup of up to $8\times$ (Figure 1).

2 Related Work

Sequence-level distillation. Knowledge distillation for LLMs often uses sequence-level supervised fine-tuning (SeqKD; Kim and Rush, 2016b) on teacher outputs. This recipe is used by instruction-tuning pipelines such as Alpaca, Vicuna, LIMA, and instruction tuning with GPT-4 (Taori et al., 2023; Chiang et al., 2023; Peng et al., 2023; Zhou et al., 2023). While simple, SeqKD is off-policy: the student trains only on teacher demonstrations and does not directly correct its own mistakes under its rollout distribution (Gu et al., 2024; Wen et al., 2023). This limitation is closely related to exposure bias and error accumulation in sequence generation (Bengio et al., 2015; Arora et al., 2022).

On-policy and adversarial distillation. The off-policy gap motivates on-policy distillation methods that learn from model-generated responses (Agarwal et al., 2024a; Gu et al., 2024). GAD (Ye et al., 2025) instantiates this idea in a black-box setting by training a discriminator to separate teacher and student responses and using its score as an RL reward, building on discriminator-based learning (Goodfellow et al., 2014; Yu et al., 2017). However, fully on-policy adversarial training can be unstable and expensive in memory and compute (Ye et al., 2025; Chen et al., 2026).

Preference optimization. In parallel, preference optimization provides another route to learning from comparisons. Preference-based alignment methods are widely used in post-training (Ouyang et al., 2022), and DPO (Rafailov et al., 2023) and follow-ups such as RPO (Liu et al., 2024) show that preference losses can be stable, especially when combined with supervised signals. Preference-based alignment has also been used as a practical distillation tool (Tunstall et al., 2023). In black-box distillation, teacher versus student comparisons naturally define preferences, yielding a contrastive signal that resembles on-policy correction but fits black-box constraints (Tunstall et al., 2023; Agarwal et al., 2024a; Gu et al., 2024; Chen et al., 2026). CE-GAD is complementary to these lines: we keep the GAD objective but reduce redundant work by adapting when to update the discriminator and how aggressively to update the actor.

3 Analysis

The inefficiency of vanilla GAD is typically attributed to its heavy per-step computational cost. However, empirical profiling reveals that the primary bottleneck is **redundancy**: a foundational resource misallocation that forces the system to spend massive compute on uninformative states due to the *non-stationary* nature of adversarial training.

3.1 The Standard GAD Setup

To contextualize this redundancy, we first spell out the standard GAD pipeline (Ye et al., 2025). Let x be an instruction prompt, y^T a cached response sampled from the black-box teacher, and $y \sim \pi_\theta(\cdot | x)$ a response generated by the student. GAD treats the student as a generator and trains a discriminator D_ϕ as an online reward model. Given the paired responses (y^T, y) for the same prompt, the discriminator is optimized to assign a higher score to the teacher response than to the student response, typically with a Bradley-Terry preference loss (Bradley and Terry, 1952):

$$\mathcal{L}_D(\phi) = -\log \sigma(D_\phi(x, y^T) - D_\phi(x, y)). \quad (1)$$

The discriminator score on the student response is then converted into a reward $R_\phi(x, y)$ for policy optimization. Since the sampled response y is not differentiable with respect to the student parameters, the student is trained with a policy-gradient objective, such as PPO or GRPO, to maximize $R_\phi(x, y)$ while staying close to a reference policy through a KL penalty, $\beta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$. In vanilla GAD, this produces a fixed adversarial loop: sample student rollouts, update the discriminator on teacher–student pairs, and update the student against the current discriminator reward. This setup is powerful because the reward model is not frozen: as the student distribution changes, the discriminator also changes, giving on-policy feedback in a purely black-box setting. The same property also makes training expensive and fragile. Every training step must pay for student generation, discriminator optimization, and actor optimization, while the relative learning speeds of the discriminator and actor can change over time. Based on profiling of vanilla GAD using Qwen2.5-7B-Instruct, we identify three critical failure modes where this fixed loop wastes compute and degrades final alignment quality. Figures 2 and 3 track these dynamic anomalies.

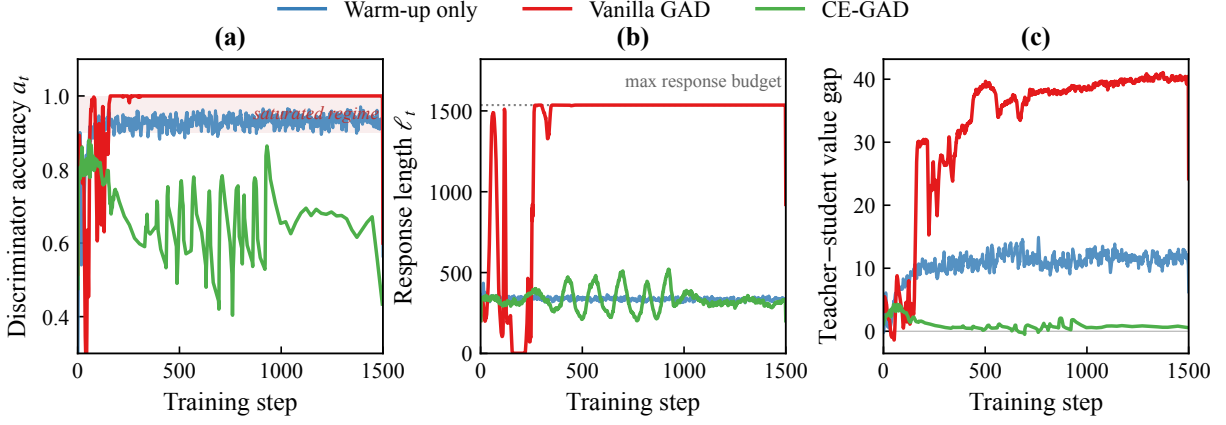


Figure 2: Training-time trajectories of vanilla GAD (with and without discriminator warm-up) versus CE-GAD. **(a) Information Vanishing:** Discriminator accuracy swiftly saturates under vanilla GAD, whereas CE-GAD maintains it within an informative, dynamic band. **(b) Policy Drift:** Vanilla GAD triggers severe mode collapse with response lengths plunging from ~ 300 to < 50 tokens; CE-GAD stabilizes generations within the reference trust region. **(c) Reward De-alignment:** The absolute reward gap $|R_T - R_S|$ expands by an order of magnitude under vanilla GAD, whereas CE-GAD consistently preserves the relative ranking gap.

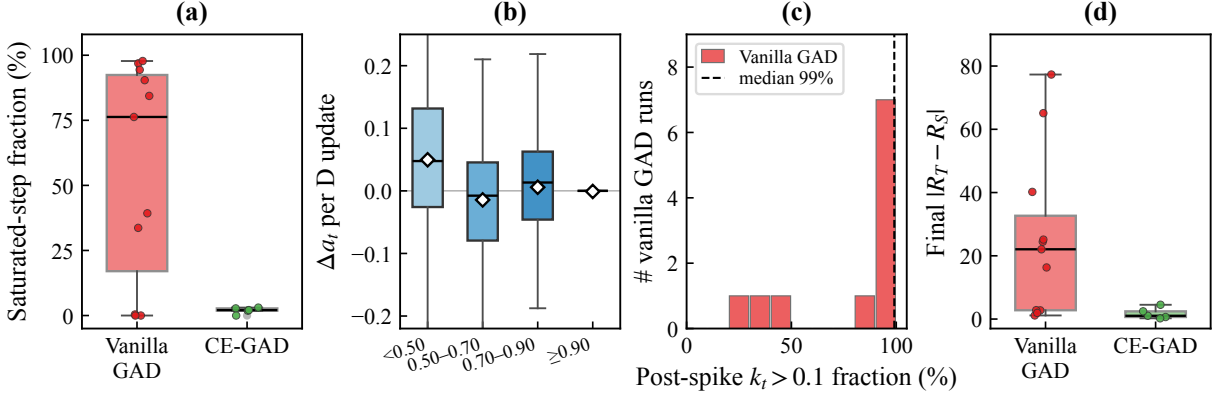


Figure 3: Aggregate empirical evidence of the three failure modes across training runs. **(a)** Percentage of total training steps trapped in the saturated discriminator regime. **(b)** Per-step marginal information gain (accuracy delta) binned by baseline accuracy, highlighting near-zero value in the saturated zone (> 0.7). **(c)** Distribution of post-spike KL persistence, showcasing vanilla GAD’s inability to recover from divergence. **(d)** Final teacher-student reward gap at convergence, demonstrating CE-GAD’s tight concentration in the healthy ranking regime.

3.2 Three Failure Modes of Vanilla GAD

Information Vanishing via Task-Asymmetry Over-computation. Vanilla GAD treats the discriminator as a symmetric, full-scale counterpart to the generator. In reality, the discriminator’s task is strictly lower in complexity. It acts as a sequence-level ranker rather than a fluent text generator. Under a rigid update schedule, a generator-scale discriminator achieves near-perfect classification boundaries almost immediately after the critic warm-up phase. As shown in Fig. 2a, a_t swiftly locks into a saturated band ($0.7 \sim 0.9$). In this regime, subsequent discriminator backward passes yield near-zero marginal information gain (Fig. 3b), moving accuracy by a negligible $0.01 \sim 0.02$ while

continuing to extract the full per-step cost c_D . This motivates our *Information-Guided Scheduling* and architectural decoupling (Section 4.1).

2. Policy Drift, Token Sparsity, and Degenerate Length Collapse.

Advantage-based actor updates in vanilla GAD operate under the flawed assumption that every token rollout is highly informative and that the optimization landscape remains safe. Empirically, we observe that the effective gradient mass is heavily concentrated in a sparse subset (the top-40% of tokens), meaning substantial wall-clock time is wasted calculating gradients for low-signal or padding tokens. Furthermore, when the discriminator saturates, it presents an aggressive, uncalibrated landscape. The actor quickly

violates its reference trust region, triggering permanent KL spikes from which it rarely recovers (Fig. 3c). Crucially, a leading indicator for this catastrophic quality collapse is a sudden, severe **length collapse**, where the mean student response length abruptly drops from ~ 300 tokens to under 50 tokens (Fig. 2b). This motivates our *Stability-Constrained Optimization and Multi-Trigger Gating* (Section 4.2). When the discriminator boundary becomes sharp, the model relies on statistical features in the student generations (e.g., punctuation density), structural markers, or termination tokens, instead of semantic quality. Consequently, the policy learns that shorter sequences reduce error probability and maximize token-level advantage estimates. This creates a feedback loop where the model terminates generation early. This shortcut reduces vocabulary diversity and leaves large parts of the sequence space unoptimized.

3. Reward Signal De-alignment and Within-Batch Blindness. While a saturated discriminator may maintain correct sequence-level ranking, its gradient signal often degenerates. An overconfident discriminator expands the absolute scale of its outputs, causing the absolute teacher–student reward gap $|R_T - R_S|$ to swell by an order of magnitude (Fig. 2c and Fig. 3d). Consequently, policy gradients become dominated by a massive global scalar shift rather than fine-grained contrastive directions. This issue worsens during frequent training batches where the positive advantage ratio drops precipitously ($p_t < 0.15$). In these highly skewed scenarios, the actor is uniformly penalized, destroying all within-batch contrastive signals and stalling directional policy improvement. This motivates our *Ranking-Invariant Reward Alignment* and GRPO-style normalization (Section 4.3). In RL, policy updates depend on the variance of advantages within a single batch. However, when a global scalar shift inflates the absolute reward space, the relative variance within the batch shrinks. This mathematical flattening applies the same penalty to different error types, preventing the actor from distinguishing between a response that is off-topic and one that contains minor grammatical errors.

4 The CE-GAD Framework

The primary objective of CE-GAD is to transform the static adversarial loop of **vanilla GAD** into a dynamic, state-aware process. Instead of committing

to a fixed, symmetrically heavy update schedule, CE-GAD continuously monitors instantaneous information gain, policy stability, and rolling reward distributions to adaptively modulate distillation intensity and conserve computational budget.

4.1 Lightweight and Adaptive Scheduling

As identified in Section 3.2, vanilla GAD inherently suffers from resource misallocation due to its architectural symmetry, leveraging a generator-scale model to optimize a sequence-level binary ranking objective. This structural redundancy induces rapid discriminator saturation, wherein the gradient $\nabla_{\phi} \mathcal{L}_D$ yields diminishing directional information once the classification boundary becomes overly sharp ($a_t \in [0.7, 0.9]$). To resolve this asymmetry, CE-GAD decouples the optimization scales by parameterizing the discriminator with a lightweight critic network², stabilized via label smoothing and a conservative cosine learning rate decay. Furthermore, we modulate the optimization pacing through an information-guided saliency mechanism. Formally, we introduce an indicator variable $\alpha_{D,t} \in \{0, 1\}$ to govern whether a discriminator backward pass is executed at step t . The updating interval r_t is dynamically computed as a piecewise function of the rolling classification accuracy a_t :

$$r_t = f(a_t) = \begin{cases} r_{\max}, & \text{if } a_t \geq \tau_{\text{sat}} \\ r_{\min}, & \text{if } a_t < \tau_{\text{sat}} \end{cases} \quad (2)$$

where τ_{sat} is the pre-defined saturation threshold, and $r_{\min}, r_{\max} \in \mathbb{N}^+$ denote the bounds of the optimization stride. The update gate $\alpha_{D,t}$ is then activated if and only if $t \equiv 0 \pmod{r_t}$. When the discriminator exhibits near-perfect separation ($a_t \geq \tau_{\text{sat}}$), the update interval is extended to $r_{\max}$ to bypass uninformative backward passes, thereby eliminating the redundant compute cost. Conversely, if a_t drops, the schedule aggressively reduces the interval to $r_{\min}$ to tighten the reward boundary. In scenarios of irreversible convergence or absolute saturation where $a_t \rightarrow 1.0$ persistently, the framework seamlessly transitions into a critic-free regime, completely bypassing the forward and backward paths of D_{ϕ} to yield a constant computational saving of c_D per step.

4.2 Gated Stability Optimization

To counteract the catastrophic recovery costs (c_{rec}) arising from length collapse and explosive KL drift,

²We implement this lightweight critic using Qwen2.5-1.5B-Instruct across all experimental setups.

CE-GAD treats the student optimization as a highly conditional, stability-constrained problem. We abandon the assumption that every actor update is fundamentally beneficial.

Multi-Trigger Actor Freeze Gating. We introduce a hard binary gate $b_t \in \{0, 1\}$ that monitors the health of the optimization landscape. The actor gradient update is executed if and only if $b_t = 1$, where:

$$b_t = \mathbb{I}(a_t \geq \tau_{\text{col}} \wedge k_t \leq \tau_{\text{kl}} \wedge L_t \geq \tau_{\text{len}}) \quad (3)$$

Here, a_t tracks discriminator collapse ($\tau_{\text{col}} = 0.35$), $k_t = \text{KL}(\pi_\theta \parallel \pi_{\text{ref}})$ guards against reference distribution divergence, and L_t monitors the rolling response token length. Because a sudden drop in generation length serves as a leading indicator for mode collapse, this multi-trigger gate immediately freezes actor updates ($b_t = 0$) upon violation, preventing the policy from chasing corrupt reward landscapes.

Dual Adaptive Hyperparameter Tuning. The dynamic adjustment of the trust-region coefficient β_t and the policy learning rate η_t is formally governed by the tracking of instantaneous KL drift:

$$\beta_t = \beta_0 \cdot \max\left(1, \frac{k_t}{\tau_{\text{kl}}}\right), \quad (4)$$

$$\eta_t = \frac{\eta_0}{1 + \gamma \max(0, k_t - \tau_{\text{kl}})} \quad (5)$$

where β_0 and η_0 denote the baseline initialization values, and γ is a scaling penalty factor. This formulation ensures that when the policy begins to escape the reference anchor boundary, the optimization pressure is smoothly self-throttled.

Sparsity-Aware Gradient Pruning. To minimize the per-token computational overhead ($\alpha_G \cdot n \cdot T \cdot c_G$), we filter out low-signal tokens. We isolate a high-signal token subset $\mathcal{V}_t \subseteq \{1, \dots, T\}$ where the absolute advantage exceeds a dynamic quantile q_t . The resulting clipped policy loss is formulated as:

$$\mathcal{L}_{\text{policy}}(\theta) = \mathbb{E}_{t \in \mathcal{V}_t} [\min(\rho_t \tilde{A}_t, \text{clip}(\rho_t, 1 - \epsilon, 1 + \epsilon) \tilde{A}_t)] \quad (6)$$

where $\rho_t = \pi_\theta(y_t \mid x, y_{<t}) / \pi_{\theta_{\text{old}}}(y_t \mid x, y_{<t})$ represents the standard importance ratio, and $\tilde{A}_t$ is the aligned advantage. This gradient denoising narrows the backward pass to the top token mass, drastically cutting wall-clock time.

4.3 Invariant Reward and GRPO Alignment

The final failure mode stems from the divergence between discriminator ranking correctness and raw reward magnitude. Overconfident discriminators often inflate reward variances, generating massive gradients that trigger the sharp reward $\rightarrow$ drift $\rightarrow$ recovery chain.

Reward Rescaling. To safeguard gradient stability, we apply an empirical damping factor $s_t = (1 - a_t)$ directly to the raw discriminator outputs. As the discriminator confidence a_t climbs, the absolute reward magnitude is compressed, ensuring that overconfidence does not skew the optimization trajectory while strictly preserving the relative sequence ranking.

Advantage Normalization. In highly skewed training batches where the positive advantage ratio p_t falls below a critical floor (e.g., $p_t < 0.15$), the actor receives an uninformative uniform penalty (“everything is wrong”), destroying intra-batch contrastive signals. To restore relative ranking within the batch without inflating forwarding costs, we dynamically blend raw advantages with standardized within-batch advantages:

$$\tilde{A}_t = (1 - \lambda_t) A_t + \lambda_t \left(\frac{A_t - \mu_A}{\sigma_A} \right) \quad (7)$$

The mixing weight $\lambda_t \in [0, 1]$ is adaptively driven by p_t . When p_t is dangerously low, $\lambda_t \rightarrow 1$, enforcing GRPO-style contrastive normalization. This architectural alignment prevents the student from collapsing into a generic high-reward mode and ensures robust, directionally sound policy updates at zero additional forward/backward pass cost.

5 Experiments

5.1 Setup

Dataset. Following Ye et al. (2025), we use LMSYS-Chat-1M-Clean, a curated subset of the LMSYS-Chat-1M dataset (Zheng et al., 2024). We distill from cached teacher responses and train on LMSYS-style conversational prompts.

Teacher and student models. We adopt GPT-5-Chat (OpenAI, 2025) as the black-box teacher, accessed only through its text API (no logits, hidden states, or parameters). For students, we use instruction-tuned variants from the Qwen2.5 family (Qwen2.5-3B/7B/14B-Instruct) and the Llama-3 family (Llama-3.2-3B-Instruct; Grattafiori et al.,

2024). This suite spans two architectures and three parameter scales (3B, 7B, 14B).

Training. Unless otherwise stated, we follow the training and evaluation protocol of Ye et al. (2025). We implement CE-GAD in verl with GRPO updates, where π_θ denotes the student policy and D_ϕ denotes the online discriminator. We use train batch size 256, validation batch size 600, policy mini-batch size 256, maximum prompt length 2,048, maximum response length $T = 1,536$, and sampling temperature 0.8. For each prompt, we sample $n = 4$ rollouts with vLLM using tensor parallel size 2, and train for 2 epochs on 8 H200 GPUs. The actor uses base learning rate $\eta_0 = 1 \times 10^{-6}$ and initial KL coefficient $\beta_0 = 0.03$; the discriminator uses learning rate 5×10^{-6} and 80 warm-up steps. The base generator-to-discriminator update ratio is 3 before applying the adaptive schedule, and the controller tracks a_t , k_t , L_t , and advantage statistics online. Checkpoints and evaluations are run every 50 steps.

Evaluation. We evaluate on held-out instruction-following benchmarks (Dolly Databricks, 2023, Self-Instruct Wang et al., 2023, Vicuna Chiang et al., 2023, and an LMSYS-Chat-1M-Clean split Zheng et al., 2024) using GPT-4o as judge, and report per-set scores together with their average.

Baselines. We compare three regimes: **Base**, the original instruction-tuned student; **GAD** (Ye et al., 2025); and **CE-GAD**. Our adaptive controller on top of the same objective.

5.2 Main Results

Table 1 presents the evaluation scores across the four datasets for the Base, GAD, and CE-GAD models. The empirical data yields two primary findings regarding alignment quality and architectural scalability, with a particular emphasis on relative performance trajectories.

Comparison with the Base Models. CE-GAD consistently outperforms the base instruction-tuned models across all evaluated configurations, delivering substantial relative gains in final average scores. Specifically, CE-GAD yields relative improvements of +5.1% for Qwen2.5-3B, +4.8% for Llama-3.2-3B, +6.3% for Qwen2.5-7B, and +3.1% for Qwen2.5-14B. These robust increments across the board indicate that the adversarial distillation objective effectively transfers teacher knowledge

regardless of the underlying architecture or parameter scale.

Comparison with Vanilla GAD. When evaluated directly against the standard GAD baseline, CE-GAD matches or exceeds its performance while operating at a lower computational cost, demonstrating superior efficiency-to-reward scaling. This advantage is most prominent on the Qwen2.5-7B backbone, where CE-GAD outperforms vanilla GAD across every individual benchmark, culminating in a +3.0% relative improvement in the overall average score. On a per-dataset level for the 7B model, the most notable surge occurs on Self-Instruct with a +3.9% relative increase, accompanied by solid relative gains on Dolly (+3.0%), LMSYS (+2.7%), and Vicuna (+2.4%). For the remaining Qwen families, CE-GAD secures localized advantages on three out of four datasets, driving minor relative lifts of +0.6% and +0.3% in final averages for the 3B and 14B models, respectively. On the Llama-3.2-3B architecture, CE-GAD maintains a comparable average performance to GAD (with a negligible -0.04% relative difference) while highlighting a distinct +2.9% relative boost specifically on the Vicuna task.

Analysis of Stability across Scales. The relative improvement patterns over vanilla GAD reveal a distinct, non-linear relationship with model capacity, peaking at the median scale (Qwen2.5-7B). At the smaller 3B scale, standard adversarial loops often suffer from high baseline variance; here, the steady relative progress (+0.6% for Qwen2.5-3B) demonstrates that the stability gates in CE-GAD successfully anchor a stable optimization trajectory. Conversely, for the largest configuration (Qwen2.5-14B), the strong intrinsic capabilities of the base model naturally compress the headroom for further expansion, resulting in a modest +0.3% relative gain over GAD. These scaling trends confirm that the dynamic scheduling and reward stabilization mechanisms do not compromise alignment quality, proving that optimization updates can be executed selectively without sacrificing distillation accuracy.

5.3 Training Efficiency

CE-GAD converts the static GAD loop into an adaptive resource-allocation process. To quantify this, we profile the per-step wall-clock time of both vanilla GAD and CE-GAD on the Qwen2.5-7B-Instruct model using identical hardware configurations.

Table 1: Performance Comparison between Base, GAD , and CE-GAD across different model families.

Family	Model Type	Dolly	Self-Inst	Vicuna	LMSYS	Average
Qwen2.5-3B	Base	0.4239	0.4487	0.4675	0.4334	0.4434
	GAD	0.4378	0.4641	0.4873	0.4625	0.4629
	CE-GAD	0.4472	0.4706	0.4803	0.4656	0.4659
Llama-3.2-3B	Base	0.4463	0.4545	0.4729	0.4129	0.4467
	GAD	0.4625	0.4815	0.4721	0.4574	0.4684
	CE-GAD	0.4537	0.4748	0.4859	0.4582	0.4682
Qwen2.5-7B	Base	0.4576	0.4685	0.4825	0.4572	0.4665
	GAD	0.4645	0.4889	0.4944	0.4785	0.4816
	CE-GAD	0.4785	0.5080	0.5060	0.4912	0.4959
Qwen2.5-14B	Base	0.4750	0.4808	0.4904	0.4727	0.4797
	GAD	0.4904	0.4934	0.4962	0.4936	0.4934
	CE-GAD	0.4889	0.4940	0.5007	0.4957	0.4948

Per-step Component	Vanilla GAD	CE-GAD	Speedup
Wall-clock total (s)	212.9	26.1	8×
Actor update (s)	49.0	7.5	6.54×
Discriminator update (s)	66.1	4.2	15.73×
Rollout (s)	57.7	13.5	4.29×
Update frequency $\bar{\alpha}_D$	1.00	0.19	n/a
Response length T	1536	324	n/a

Table 2: Per-step wall-clock breakdown on Qwen2.5-7B-Instruct. CE-GAD achieves an 8× overall speedup by maintaining optimal sequence lengths and reducing discriminator update frequency.

As shown in Table 2, GAD requires 212.9 seconds per step, while CE-GAD reduces the total step time to 26.1 seconds, yielding an overall 8× speedup. This performance gain is distributed across all three primary computational phases. The discriminator update shows the largest acceleration, speeding up by 15.73× (from 66.1 to 4.2 seconds). The actor update accelerates by 6.54× (from 49.0 to 7.5 seconds), and the autoregressive rollout phase accelerates by 4.29× (from 57.7 to 13.5 seconds). These individual component speedups stem from the compounding effects of the two dynamic control levers. The 15.73× reduction in discriminator overhead is directly tied to Information-Guided Scheduling, which drops the average update frequency $\bar{\alpha}_D$ from 1.00 to 0.19, meaning backward updates are bypassed in 81% of the steps. Furthermore, the efficiency gains in the rollout and actor phases are a consequence of stabilizing the policy trajectory. In vanilla GAD, the student suffers from policy drift and reward hacking, causing it to continuously generate repetitive tokens until hitting the hard ceiling of the 1536 maximum sequence length. By enforcing stability

constraints, CE-GAD prevents this degenerate behavior and maintains a healthy average response length of 324 tokens. This structural stability shortens the generation sequence, which scales down both the forward rollout computation and the memory footprint required for actor gradient calculation.

We note that a substantial portion of the measured speedup arises because vanilla GAD has already entered a pathological regime (length collapse to the maximum context window), whereas CE-GAD maintains healthy generation lengths. In other words, the comparison reflects CE-GAD avoiding a failure mode rather than compressing an already well-functioning pipeline. This distinction should be considered when interpreting the absolute speedup figure.

6 Conclusion

We presented CE-GAD, an adaptive framework eliminating computational redundancy in black-box on-policy adversarial distillation (GAD). By transitioning from a static schedule to a dynamic, state-aware resource allocation scheme, CE-GAD mitigates three common failure modes: discriminator information vanishing, student length collapse, and reward magnitude de-alignment.

Empirical evaluations across Qwen2.5 and Llama-3.2 families demonstrate up to an 8× per-step wall-clock speedup while maintaining or exceeding vanilla GAD quality, especially in Qwen2.5-7b. These results prove that adversarial distillation’s prohibitive overhead stems from an inefficient update cadence rather than inherent objective limitations. This philosophy of state-aware pacing extends naturally to broader RL alignment pipelines, highlighting a promising avenue for adaptive compute allocation in post-training.

Limitations

While CE-GAD significantly reduces computational redundancy, we note several avenues for further extension.

Compute Characterization. Our wall-clock profiling captures overall training acceleration under standardized conditions. Future extensions will include a granular breakdown of per-component savings alongside transient hardware metrics such as peak memory utilization and multi-node communication overhead.

Evaluation Breadth. The current empirical evaluation focuses on standard instruction-following benchmarks under a representative teacher distribution. Validating the framework against specialized domains, including long-context reasoning, multilingual synthesis, and safety alignment, remains an important direction to confirm broader generalization.

Control Function Generality. The thresholds governing the update strides and optimization gates are currently instantiated via fixed heuristics. Exploring automated, dynamic parameterization through online control loops represents a promising next step to enhance out-of-the-box adaptability across diverse teacher-student pairs.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024a. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*.
- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024b. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.
- Martin Arjovsky and Léon Bottou. 2017. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*.
- Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Cheung. 2022. [Why exposure bias matters: An imitation learning perspective of error accumulation in language generation](#). In *Findings of ACL*.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. [Scheduled sampling for sequence prediction with recurrent neural networks](#). In *Proceedings of NeurIPS*.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Xiwen Chen, Jingjing Wang, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hejian Sang, Zhipeng Wang, Alborz Geramifard, and Feng Luo. 2026. Soda: Semi on-policy black-box distillation for large language models. *arXiv preprint arXiv:2604.03873*.
- Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. [Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality](#).
- Databricks. 2023. [Free dolly: Introducing the world’s first truly open instruction-tuned llm](#).
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- 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. *arXiv preprint arXiv:2407.21783*.
- Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.
- Yoon Kim and Alexander M Rush. 2016a. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327.
- Yoon Kim and Alexander M Rush. 2016b. [Sequence-level knowledge distillation](#). In *Proceedings of EMNLP*.
- Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. 2024. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. *Advances in Neural Information Processing Systems*, 37:138663–138697.
- OpenAI. 2025. [Introducing gpt-5](#).
- 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. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. [Instruction tuning with GPT-4](#). *arXiv preprint arXiv:2304.03277*.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 4323–4332.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.
- Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Cl  mentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [Self-instruct: Aligning language models with self-generated instructions](#). In *Proceedings of ACL*.
- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. [f-divergence minimization for sequence-level knowledge distillation](#). In *Proceedings of ACL*.
- Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. 2025. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. In *The Twelfth International Conference on Learning Representations*.
- Chunting Zhou, Pengfei Liu, Puxin Xu, Srinu Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. [LIMA: Less is more for alignment](#). In *Proceedings of NeurIPS*.