DiffGate: Difficulty-Gated Teacher Guidance for On-Policy Distillation

Anonymous authors
Paper under double-blind review

Abstract

On-policy distillation (OPD) has emerged as a widely used paradigm for post-training large language models, reducing the train–test mismatch of conventional distillation by supervising the student on its own generated trajectories. However, existing OPD objectives remain largely token-local and outcome-agnostic, optimizing teacher–student agreement at each prefix despite reasoning quality being determined at the trajectory level. Reinforcement learning with verifiable rewards (RLVR), particularly Group Relative Policy Optimization (GRPO), provides complementary outcome-level supervision but suffers from sparse rewards and coarse credit assignment. We show that OPD and RLVR exhibit complementary blind spots: teacher signals provide dense local guidance but are weakly aligned with rollout correctness, whereas group-relative rewards capture task success but provide coarse token-level credit and vanish on all-failure groups. We introduce DiffGate, an outcome-gated objective that combines GRPO with selective, bounded teacher guidance. Teacher supervision is applied only to failed trajectories, scaled by group difficulty, and smoothly bounded to prevent extreme teacher–student discrepancies from dominating optimization. The verifier therefore determines *which trajectories* receive teacher guidance, while the teacher provides dense token-level update directions within those trajectories. Across Qwen3-0.6B and Qwen3-1.7B students, DiffGate improves code avg@8 over matched GRPO by $+1.7$ and $+1.8$ points and pass@8 by $+1.6$ and $+5.7$ points, respectively. On mathematics, avg@8 remains within 0.5 points of GRPO while pass@8 improves by $+1.1$ and $+3.9$ points. Overall, DiffGate improves pass@8 across all four model–domain settings, demonstrating improved solution coverage under our evaluation protocol.

1 Introduction

Knowledge distillation (KD) transfers capabilities from a teacher to a smaller student by matching their predictive distributions. On-policy distillation (OPD) reduces the train–test mismatch of conventional KD by training on trajectories sampled from the student and querying the teacher on the states the student actually visits. A common OPD formulation minimizes the reverse KL divergence between the student and teacher on these visited states, yielding dense token-level supervision from their probability differences. However, this supervision remains *local and outcome-agnostic*: tokens are updated based on distributional disagreement regardless of whether the complete trajectory is correct. Consequently, teacher disagreement may reflect stylistic differences rather than reasoning errors, while large probability gaps can produce disproportionately large updates. Thus, OPD provides dense on-policy supervision, but not necessarily supervision aligned with task success (Hinton et al., 2015; Lu & Lab, 2025; Li et al., 2026a; Gan et al., 2026; Fu et al., 2026; Yang et al., 2026).

Reinforcement learning with verifiable rewards (RLVR) provides a complementary source of supervision. Methods such as GRPO directly optimize sequence-level correctness using verifiable outcomes, including mathematical answers and executable code. However, this supervision is inherently coarse: a single trajectory-level advantage is propagated to all tokens, limiting fine-grained credit assignment. More critically, the group-relative advantage vanishes when all sampled trajectories receive the same reward, leaving difficult all-failure groups without a learning signal. RLVR can further suffer from sparse rewards and policy entropy collapse during optimization. OPD and GRPO therefore exhibit complementary blind spots: OPD is *dense but correctness-blind*, whereas GRPO is *correctness-aware but coarse and sparse* (Nie et al., 2026; Liu et al., 2026a; Mishra et al., 2026; Mroueh, 2025; Shao et al., 2024; Wen et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

This complementarity motivates integrating OPD with RLVR; however, naively combining their learning signals can introduce conflicting supervision and hinder optimization. As illustrated in Figure 1, teacher–student disagreement alone cannot reliably distinguish correct from incorrect trajectories or localize the decisive reasoning error. We therefore use outcome feedback to route teacher supervision at the trajectory level rather than treating teacher disagreement as a token-level credit-assignment signal. Uniform distillation may therefore modify already-correct solutions, while large teacher–student discrepancies can dominate updates even when outcome feedback is informative. Moreover, strong teacher imitation may restrict exploration and interfere with reward-driven learning. Our training diagnostics further show that teacher disagreement can be larger on correct trajectories than on failed ones, while reward-only training can substantially reduce policy entropy (see Appendix Section E.6). These observations motivate a simple principle: apply bounded teacher guidance selectively to failed trajectories, concentrate it where outcome supervision is weakest, and reduce its influence as the student improves (Wang et al., 2026a; Cai et al., 2026).

We introduce DiffGate, a unified post-training objective that injects teacher knowledge directly into the RL update through *selective, bounded teacher guidance*. For each failed trajectory, DiffGate converts the teacher–student log-probability gap into a bounded token-level signal and scales it by the difficulty of the sampled group. Teacher guidance is therefore strongest on all-failure groups, where GRPO provides no signal, decreases as the group solve rate improves, and vanishes on successful trajectories. Bounding the teacher term preserves its local update direction while preventing extreme discrepancies from dominating optimization. In this way, the verifier determines *which failed trajectories* receive teacher guidance, while the teacher provides dense token-level update directions within those trajectories. Our main contributions are summarized as follows:

- **Local–global supervision gap.** We identify complementary blind spots in OPD and GRPO: OPD provides dense but outcome-agnostic token supervision, whereas GRPO captures trajectory correctness but offers coarse credit and vanishes on reward-uniform groups.
- **Difficulty-gated teacher guidance.** We introduce DiffGate, which integrates bounded teacher guidance directly into the GRPO update, activating it only on failed trajectories and scaling it by group difficulty. This concentrates local supervision where outcome feedback is weakest while preserving successful trajectories.
- **Comprehensive empirical evaluation.** We evaluate DiffGate against GRPO and distillation baselines on mathematical reasoning and code generation across Qwen3 student sizes, assess out-of-distribution transfer to science benchmarks, and analyze the effects of difficulty gating, bounded teacher guidance, rollout group size, and teacher strength.

2 Related Work

On-Policy Distillation (OPD). OPD trains the student on its own generated trajectories and queries a teacher on the visited states, reducing the distribution mismatch of conventional distillation (Agarwal et al., 2024; Li et al., 2026a; Xu et al., 2026; Team et al., 2026). Recent work has focused on improving the reliability of this dense token-level supervision. EOPD adapts the divergence according to teacher uncertainty to mitigate the mode-seeking behavior of reverse KL (Jin et al., 2026), while Uni-OPD identifies insufficient exploration and unreliable teacher supervision as key bottlenecks and calibrates token guidance using outcome information (Hou et al., 2026). Other analyses show that OPD can fail when teacher and student reasoning patterns are poorly aligned, when the teacher provides little additional capability, or when teacher gradients are noisy on already-correct trajectories (Li et al., 2026a; Armandpour et al., 2026). TGPO further studies unreliable negative supervision under large teacher–student policy divergence (Liu et al., 2026b). These methods improve *which* teacher signals are trusted, but standard OPD remains primarily token-local: teacher–student disagreement alone does not directly encode whether the full trajectory succeeds.

Reinforcement Learning with Verifiable Rewards (RLVR). RLVR directly optimizes task success using automatically checkable outcomes, with GRPO becoming a widely used critic-free approach for mathematical and code reasoning (Shao et al., 2024; Guo et al., 2025). However, outcome rewards provide sparse and coarse supervision: GRPO assigns the same trajectory-level advantage to every token and yields no gradient when all responses in a group receive the same reward (Mishra et al., 2026; Nie et al., 2026). Prior work addresses related limitations through dynamic sampling (Yu et al., 2026), correcting optimization and length biases (Liu et al., 2025; Sui et al., 2025), or introducing finer-grained credit assignment (Li et al., 2026b; Jiang et al., 2025; Wang et al., 2026b). These approaches improve outcome supervision, whereas DiffGate complements it with dense teacher guidance where the group-relative signal is weak.

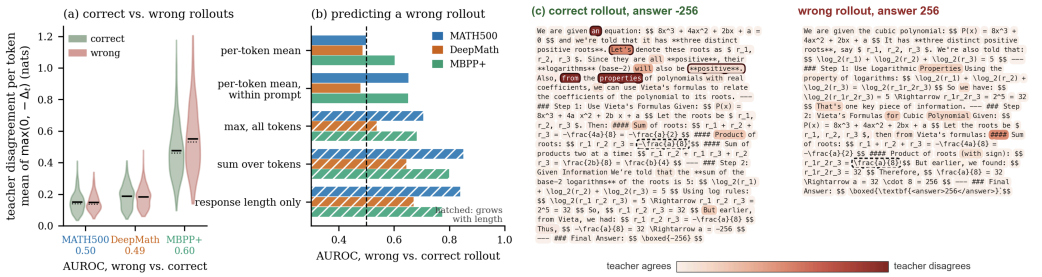

Figure 1: **Teacher disagreement alone does not reliably identify incorrect rollouts.** For a distillation-only Qwen3-1.7B student evaluated with a frozen Qwen3-8B teacher: (a) average token-level teacher–student disagreement distinguishes correct from incorrect rollouts only at chance level on mathematics (AUROC 0.50 and 0.49) and weakly on code (AUROC 0.60); (b) aggregate disagreement becomes more predictive mainly when it also reflects response length; and (c) within an incorrect rollout, large disagreement can occur on ordinary tokens rather than on the token associated with the decisive error. These observations motivate DiffGate to use the verifier to determine *which* trajectories receive teacher guidance, while the teacher provides the local token-level update direction. Further analysis is provided in Appendix Section B.

Combining OPD and RLVR. Recent work has explored combining teacher supervision with outcome-based policy optimization. KDRL jointly optimizes GRPO and reverse-KL distillation, but its teacher signal is not explicitly bounded at the token level (Xu et al., 2025). RL-aware distillation uses selective teacher imitation and a trust-region objective to balance distillation and reward optimization (Zhang et al., 2026). CoDistill-GRPO jointly trains large and small policies using outcome-based and distillation rewards, while Uni-OPD calibrates teacher supervision using outcome information (Kwon et al., 2026; Hou et al., 2026). In contrast, DiffGate combines bounded teacher guidance on failed trajectories with group-difficulty weighting, providing supervision when GRPO has no signal and reducing it as the group solve rate increases.

3 Background and Preliminaries

Problem Setup and Notations. Let π_{T} denote a fixed teacher policy and π_θ a student policy parameterized by θ. Given a prompt $x \sim \mathcal{D}$, the student autoregressively generates a response as $y = (y_1, \ldots, y_L) \sim \pi_\theta(\cdot \mid x)$, where $y_t \sim \pi_\theta(\cdot \mid c_t)$, $c_t = (x, y_{<t})$, and $\pi_\theta(y \mid x) = \prod_{t=1}^{L} \pi_\theta(y_t \mid c_t)$. Because y is sampled from the student, the visited contexts c_t are induced by its own policy. We consider two sources of supervision on these trajectories: a teacher distribution $\pi_{\mathrm{T}}(\cdot \mid c_t)$ that provides token-level feedback, and a verifier reward $R(x, y)$ that evaluates the complete trajectory.

On-Policy Distillation. OPD supervises the student on prefixes generated by the student itself, reducing the mismatch between training and inference-time states. A standard formulation minimizes the reverse KL between the student and teacher distributions at student-visited contexts as follows:

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D},\\ y\sim\pi_\theta(\cdot|x)}} \left[\frac{1}{L} \sum_{t=1}^{L} \log \frac{\pi_\theta(y_t \mid c_t)}{\pi_{\mathrm{T}}(y_t \mid c_t)} \right]. \tag{1}$$

Rather than evaluating the divergence over the full vocabulary, sampled-token OPD uses the student-generated token y_t to construct a Monte Carlo approximation of the learning signal. For a fixed context c, maximizing the negative reverse KL yields a policy-gradient form as follows:

$$\nabla_\theta J_{\mathrm{OPD}}(\theta; c) = \mathbb{E}_{a\sim\pi_\theta(\cdot|c)} \left[A(a, c)\, \nabla_\theta \log \pi_\theta(a \mid c) \right], \tag{2}$$

where the token-level OPD advantage is $A(a, c) := \log \pi_{\mathrm{T}}(a \mid c) - \log \pi_\theta(a \mid c)$. For a sampled token y_t, we therefore use $A_t = \mathrm{sg}[\log \pi_{\mathrm{T}}(y_t \mid c_t) - \log \pi_\theta(y_t \mid c_t)]$, optimized via the Monte Carlo sampled gradient estimator. During training, the sampled-token advantage is computed from the behavior policy and treated as a stop-gradient quantity. Thus, $A_t > 0$ increases the probability of tokens favored more by the teacher, while $A_t < 0$ suppresses tokens over-weighted by the student. This provides a dense token-level learning signal at every generation step.

RL with Verifiable Rewards. RL with verifiable rewards (RLVR) optimizes sequence-level task success using an automatic verifier. For reasoning and code tasks, we consider a binary reward $R(x, y) \in \{0, 1\}$ where $R(x, y) = 1$ if the generated response y is correct. The objective is:

$$J_{\mathrm{RLVR}}(\theta) = \mathbb{E}_{\substack{x\sim\mathcal{D},\\ y\sim\pi_\theta(\cdot|x)}} \left[R(x, y) \right]. \tag{3}$$

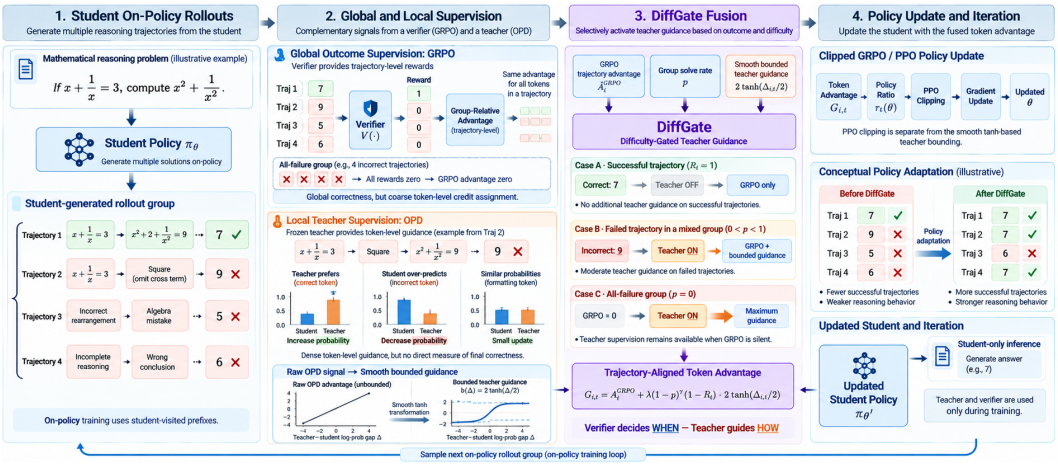

Figure 2: **Overview of DiffGate.** The student generates on-policy trajectories that receive sequence-level correctness feedback from a verifier and token-level guidance from a frozen teacher. DiffGate bounds the sampled-token teacher–student log-probability gap, applies teacher guidance only to failed trajectories, and scales its strength according to the group solve rate. The resulting token-level advantage combines local teacher guidance with the global GRPO outcome signal to update the student policy. See Algorithm 1 for details.

Using REINFORCE, the corresponding policy gradient is given by:

$$\nabla_\theta J_{\mathrm{RLVR}}(\theta) = \mathbb{E}\left[R(x,y)\sum_{t=1}^{L}\nabla_\theta \log \pi_\theta(y_t \mid c_t)\right]. \tag{4}$$

Unlike the token-level OPD signal, the verifier reward evaluates the complete response and does not identify which individual tokens contributed to its success or failure. Thus, unlike OPD, RLVR assigns the same trajectory-level signal to all tokens in a response. GRPO constructs a prompt-relative learning signal by comparing rewards within each rollout group, by sampling a group of n responses $y^{(i)} \sim \pi_\theta(\cdot \mid x)$ for the same prompt, with rewards $R_i = R(x, y^{(i)})$. Let $\bar{R} = \frac{1}{n}\sum_{j=1}^{n} R_j, \sigma_R = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(R_j-\bar{R})^2}$. The trajectory-level GRPO advantage is $A_i^{\mathrm{GRPO}} = \frac{R_i-\bar{R}}{\sigma_R+\varepsilon}$.Since A_i^{GRPO} is defined for the complete rollout, it is shared across all tokens in trajectory i. The corresponding on-policy surrogate is:

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{L_i}\sum_{t=1}^{L_i}\mathrm{sg}\left[A_i^{\mathrm{GRPO}}\right]\log \pi_\theta\left(y_t^{(i)} \mid c_{i,t}\right). \tag{5}$$

In practice, we optimize its PPO-style clipped surrogate using importance ratios with respect to the frozen behavior policy. When all trajectories receive the same reward, their GRPO advantages vanish.

DiffGate Motivation. OPD and GRPO provide complementary supervision. OPD supplies dense token-level teacher guidance but does not directly encode whether the complete trajectory is correct. GRPO captures trajectory-level correctness, but assigns the same advantage to every token and provides no learning signal on reward-uniform groups. Moreover, Figure 1 shows that teacher–student disagreement alone does not reliably distinguish correct from incorrect trajectories or localize the decisive reasoning error. These observations motivate DiffGate: the verifier determines *which* trajectories receive additional guidance, while the teacher provides dense token-level update directions within the selected failed trajectories. Importantly, DiffGate performs trajectory-level routing and does not attempt to identify the causal error token within a response.

4 Methodology

As discussed in Section 3, OPD and GRPO provide complementary supervision: OPD gives dense token-level guidance but is local and unbounded, while GRPO captures trajectory correctness but is coarse and vanishes on reward-uniform groups. We introduce DiffGate, which combines both, using teacher guidance when outcome supervision is weak. Figure 2 summarizes the method.

Local versus Global Supervision. OPD and RLVR admit a similar policy-gradient form,

$$\widehat{\nabla_\theta J} = \sum_t A_t \nabla_\theta \log \pi_\theta(y_t \mid c_t), \tag{6}$$

but differ in how the advantage is constructed. OPD provides the token-level advantage $A_{i,t}^{\mathrm{OPD}} = \log \pi_{\mathrm{T}}(y_t^{(i)} \mid c_{i,t}) - \log \pi_\theta(y_t^{(i)} \mid c_{i,t})$, whereas GRPO assigns the same trajectory-level advantage A_i^{GRPO} to every token in rollout i. Thus, OPD provides *dense but local* supervision, while GRPO provides *global but coarse* feedback. However, the OPD advantage is an unbounded log-probability ratio: large teacher–student discrepancies can produce large token-level signals, potentially dominating the GRPO advantage. Moreover, the signals may disagree: teacher-preferred tokens can occur in incorrect trajectories, while correct trajectories may contain teacher-disfavored tokens.

Bounded Local Teacher Guidance. For each prompt x, we sample a group of n trajectories $\{y^{(1)}, \ldots, y^{(n)}\}$ from the behavior policy $\pi_{\theta_{\mathrm{old}}}$. For token $y_t^{(i)}$ in trajectory i, with context $c_{i,t} = (x, y_{<t}^{(i)})$, we define the sampled-token OPD gap as follows:

$$\Delta_{i,t} = \log \pi_{\mathrm{T}}(y_t^{(i)} \mid c_{i,t}) - \log \pi_{\theta_{\mathrm{old}}}(y_t^{(i)} \mid c_{i,t}). \tag{7}$$

Although $\Delta_{i,t}$ provides dense teacher supervision, it is unbounded and can attain large magnitude when the teacher and student strongly disagree. To prevent a small number of tokens from dominating the update, we apply a bounded transformation that keeps the OPD signal on a scale comparable to the GRPO advantage and defined as follows:

$$B_{i,t} = \phi(\Delta_{i,t}) \coloneqq 2 \tanh\left(\frac{\Delta_{i,t}}{2}\right), \qquad B_{i,t} \in (-2, 2). \tag{8}$$

The transformation preserves the sign of the original OPD advantage and is approximately linear around zero, $\phi(\Delta) \approx \Delta$ for small $|\Delta|$, while smoothly saturating for large teacher–student discrepancies.

Difficulty-Gated Trajectory Guidance. Let $R_i \in \{0, 1\}$ denote the verifier reward of trajectory i and let $p = \frac{1}{n}\sum_{j=1}^{n} R_j$ denote the group solve rate. We use the standard group-relative advantage A_i^{GRPO} defined in Section 3 as the global outcome signal.

A central design choice is that teacher guidance should not be applied uniformly. In particular, we want the teacher to guide trajectories that fail the verifier, with stronger guidance on prompts that the student finds difficult. We therefore define the trajectory gate as follows:

$$w_i = (1 - p)^\gamma (1 - R_i), \tag{9}$$

where $\gamma \geq 0$ controls the dependence on group difficulty. The factor $(1 - R_i)$ removes teacher guidance from successful trajectories, while $(1 - p)^\gamma$ increases its strength as the group's solving rate decreases. Combining the global and local signals gives the proposed token-level advantage as:

$$G_{i,t} = \underbrace{A_i^{\mathrm{GRPO}}}_{\text{global outcome}} + \underbrace{\lambda(1 - p)^\gamma (1 - R_i)}_{\text{trajectory gate}} \underbrace{2 \tanh\left(\frac{\Delta_{i,t}}{2}\right)}_{\text{bounded local guidance}}, \tag{10}$$

where $\lambda \geq 0$ controls the overall strength of teacher supervision.

The components serve complementary roles: (1) *Trajectory correctness*: $(1 - R_i)$ activates teacher guidance only for failed trajectories. (2) *Group difficulty*: $(1 - p)^\gamma$ scales teacher guidance more strongly when fewer rollouts succeed. (3) *Local direction*: $2 \tanh(\Delta_{i,t}/2)$ provides a bounded token-level teacher signal. (4) *Global supervision*: A_i^{GRPO} provides trajectory-level outcome feedback whenever the rollout group contains reward variation.

Together, the verifier determines *whether* correction is needed, the group solve rate determines *how much* teacher guidance to apply, and the teacher determines *how* to update the student locally. The gate operates at the trajectory level rather than the token level, since teacher–student disagreement alone does not reliably localize the exact reasoning error within a failed response.

The objective has useful boundary behavior. For a successful trajectory, $R_i = 1$, and therefore

$$G_{i,t} = A_i^{\mathrm{GRPO}}, \tag{11}$$

so DiffGate does not impose additional teacher pressure on a solution that already satisfies the verifier. Conversely, if all trajectories fail, then $p = 0$ and $A_i^{\mathrm{GRPO}} = 0$, giving

$$G_{i,t} = 2\lambda \tanh\left(\frac{\Delta_{i,t}}{2}\right). \tag{12}$$

Thus, the teacher gate is maximal on all-failure groups, where GRPO provides no learning signal.

Policy Optimization. We treat $G_{i,t}$ as a stop-gradient token-level advantage. For trajectories sampled from $\pi_{\theta_{\text{old}}}$, we define the importance ratio $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)}|c_{i,t})}{\pi_{\theta_{\text{old}}}(y_t^{(i)}|c_{i,t})}$. The student is optimized using the clipped policy surrogate as follows:

$$\mathcal{L}_{DiffGate}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{L_i}\sum_{t=1}^{L_i}\min\Big(\rho_{i,t}(\theta)\ \text{sg}[G_{i,t}], \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}}\right)\ \text{sg}[G_{i,t}]\Big). \tag{13}$$

Here, $\text{sg}[\cdot]$ denotes the stop-gradient operator, so gradients are taken only through the current policy π_θ. Setting $\lambda = 0$ recovers GRPO, while setting $\gamma = 0$ removes group-difficulty weighting but retains teacher guidance on failed trajectories, yielding the additive GRPO+OPD baseline. These controls isolate the effects of teacher guidance and difficulty gating, respectively.

Theoretical Analysis. We establish three properties of DiffGate. First, the bounded teacher signal admits a fixed-prefix policy-gradient interpretation through an f-divergence while limiting large teacher–student discrepancies. Second, difficulty gating provides guidance on all-failure groups, where GRPO is silent, and reduces its weight as the solve rate increases. Third, both the teacher contribution and combined token-level advantage are bounded. These results characterize how DiffGate balances local teacher guidance and outcome supervision, without implying convergence of the full PPO objective. Proofs are provided in Appendix Section C.

5 Experiments

We evaluate DiffGate on mathematics and code against both distillation-only and outcome-only baselines, focusing on four central questions: **(1)** Does combining outcome-level RLVR with token-level teacher guidance improve over OPD and GRPO alone? **(2)** Does the method improve out-of-domain transfer to unseen tasks? **(3)** When does teacher guidance contribute during training, and can generic regularization explain its gains? **(4)** Which design choices matter most, including difficulty gating, bounded teacher guidance, rollout group size, and teacher strength?

Models and Training Data. We use Qwen3-0.6B and Qwen3-1.7B students with a fixed Qwen3-8B teacher (Yang et al., 2025). We train on DeepMath for mathematics (He et al., 2026) and TACO for code (Li et al., 2023), using answer verification and execution-based tests, respectively.

Evaluation Benchmarks. For mathematics, we evaluate on five challenging benchmarks: Minerva-Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24 (Zhang & Team, 2024), AIME25 (Zhang & Team, 2025), and HMMT25 (Dekoninck et al., 2026). We exclude easier benchmarks such as GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2024), and AMC23 (Math-AI, 2023) from the main evaluation. For code generation, we use MBPP+ and HumanEval+ (Liu et al., 2023), together with LiveCodeBench (Jain et al., 2025). To evaluate out-of-distribution transfer, we directly test the mathematics-trained models on science benchmarks without any additional science training, using GPQA (Rein et al., 2023), MMLU-STEM (Hendrycks et al., 2020), and OlympiadBench-Physics (He et al., 2024).

Metrics. We report *avg@8* and *pass@8*. Avg@8 measures mean correctness across sampled responses, while pass@8 measures whether at least one of eight samples is correct, capturing solution coverage. Both metrics are estimated from the same pool of 16 generations per problem. Unless otherwise stated, we use temperature 0.6, top-p 1.0 (no nucleus truncation), repetition penalty 1.15, and a maximum generation length of 16,384 tokens (More details are in Appendix Section D).

Baselines. We compare DiffGate with forward- and reverse-KL on-policy distillation (OPD) (Lu & Lab, 2025), supervised fine-tuning (SFT) on teacher-generated solutions, recent OPD frameworks such as EOPD (Jin et al., 2026), Uni-OPD (Hou et al., 2026), and GRPO (Shao et al., 2024). GRPO serves as the matched outcome-only baseline and is recovered from DiffGate by setting $\lambda = 0$. Both methods use the same training data, rollout configuration, optimizer, and training-step budget.

5.1 Main Results

Tables 1 and 2 report the main results on mathematical reasoning, code generation, and out-of-distribution (OOD) transfer to science benchmarks. We summarize three key findings below.

Student	Method	Minerva		Olympiad		AIME24		AIME25		HMMT25		Average		Science (OOD)	
		avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
0.6B	Base	10.7	27.7	15.1	37.9	2.5	6.6	0.8	6.8	0.4	3.3	5.9	16.4	21.7	48.8
	OPD (forward KL)	14.1	35.7	23.5	49.8	3.8	15.6	3.8	16.4	0.8	7.6	9.2	25.0	29.4	**59.1**
	OPD (reverse KL)	18.2	36.7	29.7	54.7	2.5	15.4	7.9	19.9	1.7	9.1	12.0	27.2	30.1	58.2
	EOPD	17.1	35.2	29.9	53.9	6.2	17.8	5.4	15.1	2.1	6.3	12.2	25.7	30.2	58.1
	Uni-OPD	18.9	36.9	30.5	**55.8**	6.2	15.9	8.8	**24.0**	0.8	5.0	13.0	27.5	30.6	57.7
	SFT on teacher traces	10.9	31.5	19.6	45.6	5.0	12.2	2.5	11.4	0.4	3.3	7.7	20.8	–	–
	GRPO	**20.0**	34.9	**31.6**	51.7	**6.7**	**24.0**	**9.2**	19.9	**3.8**	11.6	**14.2**	28.4	22.8	50.1
	DiffGate	19.4	**38.4**	30.3	54.6	6.2	21.1	8.8	19.2	**3.8**	**14.1**	13.7	**29.5**	**31.2**	58.3
1.7B	Base	28.6	46.6	38.1	63.2	14.6	33.7	10.0	21.5	4.2	14.2	19.1	35.8	27.5	56.5
	OPD (forward KL)	25.0	47.4	31.2	59.6	10.8	32.5	6.7	19.0	2.1	9.1	15.2	33.5	35.8	**63.0**
	OPD (reverse KL)	29.9	49.6	39.0	63.1	12.5	32.2	11.7	26.1	3.3	10.9	19.3	36.4	36.7	62.1
	EOPD	28.9	46.7	38.2	62.6	12.5	36.3	10.4	23.8	2.5	13.0	18.5	36.5	36.2	**63.0**
	Uni-OPD	28.7	47.4	38.1	63.9	13.8	38.0	10.0	27.1	4.6	**15.2**	19.0	38.3	36.5	**63.0**
	SFT on teacher traces	26.0	48.0	33.1	59.8	6.2	20.8	7.9	23.8	1.7	8.2	15.0	32.1	–	–
	GRPO	**31.2**	46.9	**41.8**	61.1	**15.4**	28.9	**14.6**	27.2	4.2	11.9	**21.4**	35.2	14.3	35.4
	DiffGate	**31.2**	**52.9**	41.1	**64.2**	**15.4**	**39.6**	11.7	**28.1**	**5.8**	10.8	21.1	**39.1**	**39.2**	62.4
	Teacher 8B	38.3	52.7	45.1	71.9	23.3	45.7	17.1	36.9	7.1	17.5	26.2	44.9	47.6	69.7

Table 1: **Mathematical reasoning results.** We report avg@8 and pass@8, with GRPO serving as the outcome-only baseline. **Bold** and underlined values indicate the best and second-best student results, respectively. The science (OOD) columns report average performance on GPQA, MMLU-STEM, and OlympiadBench-Physics (Table 6), evaluated using the mathematics-trained students without additional science training.

Mathematical reasoning and code generation. Across both 0.6B and 1.7B students, DiffGate outperforms all OPD baselines in both avg@8 and pass@8, while also improving pass@8 over the GRPO baseline in all four settings. On mathematics, pass@8 improves over GRPO by 1.1 and 3.9 points, with only small changes in avg@8 (-0.5 and -0.3 points). On code, DiffGate improves GRPO on both metrics: avg@8 increases by 1.7 and 1.8 points, while pass@8 increases by 1.6 and 5.7 points. Overall, DiffGate combines the strengths of distillation and outcome-based learning, consistently improving solution coverage while outperforming OPD baselines across both domains.

Out-of-distribution. We evaluate the mathematics-trained students on GPQA, MMLU-STEM, and OlympiadBench-Physics without additional science training. DiffGate achieves average science avg@8 scores of 31.2 and 39.2 for the 0.6B and 1.7B students, improving over the corresponding base models (21.7 and 27.5) and the best OPD baselines (30.6 and 36.7). The gains are driven primarily by GPQA and MMLU-STEM, while OlympiadBench-Physics changes relatively little. These results suggest that the benefits of difficulty-gated teacher guidance extend beyond the mathematics training domain, although the magnitude of transfer varies across science tasks (Table 6; Appendix E.1).

Scaling to a 4B student. We further evaluate DiffGate with a Qwen3-4B student, using a Qwen3-14B teacher for mathematics and a Qwen3-8B teacher for code. On mathematics, DiffGate achieves 30.0 avg@8 and 47.2 pass@8, outperforming the strongest OPD baseline by 2.9 avg@8 and 2.2 pass@8 points (Table 7). On code, DiffGate reaches 57.3 avg@8 and 70.5 pass@8, improving over the strongest OPD baseline by 2.5 and 2.8 points, respectively (Table 8). These results suggest that the benefits of combining outcome supervision with difficulty-gated teacher guidance extend to the larger 4B student setting, while the magnitude of the gains varies across tasks (Appendix Section E.2).

5.2 Ablation Studies

To validate the design of DiffGate, we conduct controlled ablations against matched baselines, isolating the effects of difficulty gating, bounded teacher guidance, rollout group size, and teacher strength.

Difficulty gating. Setting $\gamma = 0$ removes difficulty weighting and yields a flat GRPO+OPD objective. Across all four settings, difficulty gating improves avg@8 by 1.3–2.4 points, with gains of 2.4 points at both mathematics model sizes and 1.3–1.4 points on code (Table 11). Teacher guidance is therefore more effective when concentrated on harder groups than when applied uniformly, and most of what the gate contributes is on the domain where the reward signal becomes dense early.

Bounded teacher guidance. Replacing $2\tanh(\Delta_{i,t}/2)$ with the raw OPD log-probability gap lowers mathematics avg@8 by 1.0–1.2 points and pass@8 by 1.1–1.3 points, while lowering code avg@8 by 1.1–1.5 points and pass@8 by 0.8–1.1 points. The raw signal also increases median gradient norms by 1.4–3.3$\times$ and produces substantially larger gradient spikes during training (Table 10). Bounding therefore limits extreme teacher updates, stabilizes optimization, and improves held-out accuracy.

Smaller rollout groups strengthen the benefit of teacher guidance. Reducing the group size from $n = 16$ to $n = 4$ makes reward-uniform groups more frequent during training. GRPO drops by 1.9

Student	Method	MBPP+		HumanEval+		LCB		Average	
		avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
0.6B	Base	30.1	53.4	22.6	45.5	6.1	14.1	19.6	37.7
	OPD (forward KL)	37.8	63.7	31.4	60.8	7.7	18.7	25.6	47.7
	OPD (reverse KL)	46.0	**65.8**	33.4	56.5	10.4	20.8	29.9	47.7
	EOPD	46.0	65.3	38.5	58.8	10.3	21.5	31.6	48.6
	Uni-OPD	45.2	65.5	36.9	58.1	10.8	20.5	31.0	48.0
	GRPO	40.9	58.5	40.1	**62.2**	**13.1**	**23.8**	31.3	48.2
	DiffGate	**46.1**	65.5	**41.9**	61.8	11.0	22.0	**33.0**	**49.8**
1.7B	Base	51.6	61.5	48.2	61.6	14.7	25.5	38.2	49.5
	OPD (forward KL)	47.4	69.6	48.9	78.6	12.4	27.7	36.2	58.6
	OPD (reverse KL)	54.4	70.0	59.6	81.8	15.7	28.5	43.2	60.1
	EOPD	**55.9**	**72.1**	58.0	81.3	16.1	29.7	43.3	61.0
	Uni-OPD	53.1	69.3	60.5	78.8	16.3	28.9	43.3	59.0
	SFT on teacher traces	48.7	69.6	50.6	79.2	13.6	26.9	37.6	58.6
	GRPO	55.5	65.6	55.8	70.2	**19.0**	**31.4**	43.4	55.7
	DiffGate	55.5	70.4	**63.2**	**84.1**	16.9	29.5	**45.2**	**61.4**
	Teacher 8B	70.8	81.6	80.0	90.2	28.1	43.1	59.6	71.7

Table 2: **Code generation results.** We report avg@8 and pass@8 on MBPP+, HumanEval+, and LiveCodeBench. **Bold** and underlined values indicate the best and second-best student results, respectively.

avg@8 points on average across the four settings, compared with only 1.2 for DiffGate. On 0.6B mathematics, DiffGate moves from 0.5 points below GRPO at $n = 16$ to 1.2 points above it at $n = 4$ (Table 9; Appendix E.3). These results are consistent with the intended role of the teacher term, suggesting that it can be especially helpful when group-relative reward feedback is less informative.

The method is not highly sensitive to the exact gate strength. Sweeps over λ and γ on the 1.7B student show that relatively small teacher weights are sufficient in practice (see Appendix D.4, Figure 5). Mathematics is nearly flat for $\lambda \in \{0.25, 0.5, 1\}$ and degrades for larger values, while code peaks near $\lambda = 0.5$. Removing difficulty weighting clearly hurts mathematics, while its effect on code is smaller, whereas varying $\gamma \in \{1, 2, 3\}$ produces only small, non-monotonic changes across settings. Overall, whether teacher guidance is gated matters more than the precise gate sharpness.

Why is naive GRPO+OPD insufficient? Appendix Table 11 compares DiffGate with a matched additive GRPO+OPD baseline that uses the same bounded teacher signal on failed trajectories but removes difficulty weighting by setting $\gamma = 0$. On mathematics, difficulty gating improves avg@8 by 2.4 points at both model sizes and pass@8 by 2.8 and 4.1 points for the 0.6B and 1.7B students, respectively. On code, the gains are smaller but remain positive: avg@8 improves by 1.3 and 1.4 points, while pass@8 improves by 0.1 and 1.4 points. These results indicate that teacher guidance alone provides much of the benefit on code, whereas adapting its strength to group difficulty is substantially more important on mathematics.

5.3 Why Does DiffGate Help?

We examine the training dynamics to understand when teacher guidance is most useful and how it interacts with the GRPO signal. We focus on reward-uniform groups, teacher–student disagreement, policy entropy, and gradient behavior (Figure 3; Appendix Sections E.6–E.8).

Teacher guidance remains available when GRPO has little or no signal. Early in training, 14–36% of prompts form all-failure groups, for which the group-relative GRPO advantage is exactly zero (Table 12). These groups quickly become rare on mathematics but persist on code; for example, about 20% of 0.6B code prompts remain all-failure near the end of training. DiffGate continues to provide bounded token-level supervision on these failed trajectories. Accordingly, on 0.6B code, the teacher–student discrepancy on failed rollouts decreases from 1.04 to 0.37 nats/token under DiffGate, compared with 0.56 under GRPO; at 1.7B, it falls to 0.19 versus 0.30. On mathematics, GRPO alone already closes much of this gap. This is consistent with the larger gains observed on code, where the reward blind spot persists longer. Gating is what keeps the teacher there: on all-fail groups the gate is one and DiffGate coincides with the additive GRPO+OPD objective; as the blind spot closes, the gate turns the teacher down 10×, from 0.061 to 0.006 per token, while the additive rule still applies

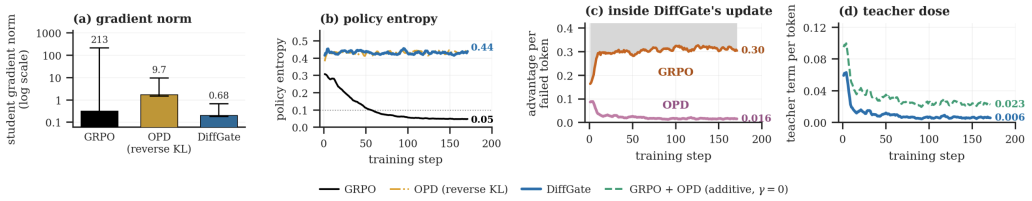

Figure 3: **Why DiffGate helps. Maths, 1.7B student; all methods use the same data, teacher, and step budget.** *(a)* Student gradient norm over training: the bar denotes the median and the whisker spans minimum to maximum, shown on a log scale. *(b)* Policy entropy, shown as a five-step rolling mean; the dotted line marks 0.1. *(c)* The two components of DiffGate's update on failed rollouts: the applied teacher term per token (dark blue) and the magnitude of the GRPO advantage per token (light blue). The GRPO advantage is zero for an all-failure group and, with $n = 16$, is approximately 0.25 or larger for groups with 1–8 successes and approximately 1.10 or larger for groups with 9 or more successes, ignoring the small ε term; values are weighted by the measured frequency of each group type. The grey region indicates this approximate lower-bound contribution. *(d)* Applied teacher term per response token under difficulty-gated and additive weighting on the same rollouts.

four times that at the end, most of it on mixed-success groups where GRPO already has a nonzero signal (see Figure 3c,d), and it trails DiffGate (see Table 11).

DiffGate exhibits more stable gradient behavior and higher policy entropy. This effect is clear for the 1.7B student. Late in training, GRPO entropy falls below 0.1, coinciding with student gradient spikes as large as 213 on mathematics and 65 on code. In contrast, DiffGate maintains higher policy entropy and limits the corresponding peaks to 0.7 and 2.6, respectively (Figure 3a,b; Table 13). Bounding the teacher signal is important for this stability: replacing $2\tanh(\Delta/2)$ with the raw log-probability gap increases median gradient norms by 1.4–3.3× and, on 0.6B code, increases the maximum gradient norm from 4.3 to 342. Thus, the bounded teacher term supplies useful local guidance without allowing large teacher–student disagreements in optimization.

Performance over training. On the 1.7B code benchmarks, DiffGate shows an early and persistent advantage in solution coverage. In the matched checkpoint, DiffGate has higher pass@8 than GRPO at the evaluated code checkpoints, while reaching most of its avg@8 improvement within the first few updates. Compared with OPD baselines, DiffGate also improves more rapidly and reaches stronger final performance, suggesting that combining outcome supervision with difficulty-gated teacher guidance is more effective than distillation alone. Overall, the gains emerge early and remain consistent across training rather than appearing only at the final checkpoint (Figure 4; Appendix E.10).

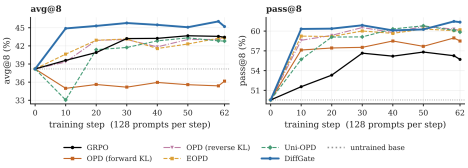

Figure 4: **Performance over training.** avg@8 (top) and pass@8 (bottom) across training checkpoints for the 1.7B student on the code benchmarks in Table 2. Step 0 denotes the base model (dotted line), enabling comparison of how each method improves accuracy and solution coverage.

Preserving policy diversity is consistent with improved solution coverage. On mathematics, GRPO achieves a higher training solve rate but converges to a substantially sharper policy, consistent with prior observations of entropy collapse during RLVR training (Wen et al., 2026). This advantage largely disappears on held-out avg@8, whereas DiffGate improves pass@8. The higher entropy under DiffGate is therefore consistent with retaining a broader set of plausible solution modes, increasing the chance of solving a problem across multiple samples. This entropy gap appears in all four settings (see Figure 3b; Figure 9), alongside higher pass@8 (see Tables 1 and 2).

6 Conclusion and Limitations

We identify a local–global mismatch between OPD and RLVR: OPD provides dense token-level guidance but is only weakly tied to trajectory correctness, while RLVR captures task success with coarse credit assignment. We introduce DiffGate, which integrates bounded, difficulty-gated teacher guidance into GRPO, concentrating supervision on failed and difficult trajectories. Across Qwen3 students, DiffGate improves solution coverage and shows more robust behavior than OPD and GRPO.

Limitations. We evaluate DiffGate on mathematical reasoning and code generation across three student model sizes. While the results demonstrate its applicability to these settings, its effectiveness on other tasks, such as agentic reasoning and long-horizon decision-making, remains unexplored. Computational constraints limit our evaluation to students of up to 4B parameters.

Ethics Statement

This work investigates post-training methods for language-model reasoning using publicly available models, datasets, and benchmarks in mathematics, code, and science. The study does not involve human participants, private user data, or sensitive demographic information. As with other post-training approaches, the behavior of the resulting models depends on the quality and reliability of both teacher and reward signals. Any downstream deployment should therefore consider potential model errors, reward misspecification, and limitations arising from imperfect supervision.

Reproducibility Statement

We provide detailed information on the training and evaluation datasets, including their sources, selection criteria, preprocessing procedures, and task-specific reward verification, along with the software environment and GPU configurations. We also describe the evaluation protocol, including benchmark settings, sampling and decoding configurations, correctness criteria, and the definitions of avg@8 and pass@8. These details are documented in the main manuscript and Appendix Section D, with the metric, benchmark versions, answer checking, and seeds.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, pp. 21246–21263, 2024. 2

Mohammadreza Armandpour, Fatih Ilhan, David Harrison, Ajay Jaiswal, Duc NM Hoang, Fartash Faghri, Yizhe Zhang, Minsik Cho, and Mehrdad Farajtabar. Unmasking on-policy distillation: Where it helps, where it hurts, and why. *arXiv preprint arXiv:2605.10889*, 2026. 2

Qiye Cai, Yichuan Ma, Linyang Li, Peiji Li, Yongkang Chen, Qipeng Guo, Yicheng Zou, Tao Gui, Xiaocheng Feng, and Bing Qin. HΘ2 sd: Hybrid hindsight self-distillation. *arXiv preprint arXiv:2607.18955*, 2026. 2

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. 22

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. 6

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. *arXiv preprint arXiv:2605.00674*, 2026. 6

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026. 1

Siyuan Gan, Yuhan Li, Xiran Wang, Linjian Meng, Boyan Wang, Zhen Zhao, Jing Huo, and Yang Gao. When teacher guidance misleads: Reward-aligned on-policy distillation. *arXiv preprint arXiv:2608.27960*, 2026. 1

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 3828–3850, 2024. 6

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026. 6, 20, 23

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020. 6

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026. 2, 3, 6, 21

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025. 6

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*, 2025. 2

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026. 2, 6, 21

Soo Min Kwon, Ziteng Sun, Ananda Theertha Suresh, Himanshu Jain, and Sanjiv Kumar. Codistill-grpo: A co-distillation recipe for efficient group relative policy optimization. *arXiv preprint arXiv:2605.08873*, 2026. 3

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, 2022. 6

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023. 6, 20

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026a. 1, 2

Ziheng Li, Liu Kang, Feng Xiao, Luxi Xing, Qingyi Si, Zhuoran Li, Weikang Gong, Deqing Yang, Yanghua Xiao, and Hongcheng Guo. Outcome-grounded advantage reshaping for fine-grained credit assignment in mathematical reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24681–24693, 2026b. 2

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. 6

Jiacai Liu, Chaojie Wang, Chris Liu, Liang Zeng, Rui Yan, Yiwen Sun, and Yang Liu. Dapo: Improving multi-step reasoning abilities of large language models with direct advantage-based policy optimization. *Advances in Neural Information Processing Systems*, 38:71445–71477, 2026a. 1

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023. 6

Xinyu Liu, Kechen Jiao, Chunyang Xiao, Runsong Zhao, Junhao Ruan, Bei Li, Jiahao Liu, Qifan Wang, Xin Chen, Jingang Wang, et al. Teacher-guided policy optimization for on-policy reasoning distillation under large policy divergence. *arXiv preprint arXiv:2605.13230*, 2026b. 2

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. 2

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation. 1, 6

Math-AI. American mathematics competition 2023. Hugging Face dataset, 2023. 6

Amritansh Mishra, Supriyo Chakraborty, and Berkcan Kapusuzoglu. On the policy gradient foundations of group relative policy optimization: Credit assignment, gradient sparsity, and rank collapse. *arXiv preprint arXiv:2606.29238*, 2026. 1, 2

Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025. 1

Wenhua Nie, Jianan Wu, Junlin Liu, Ziwei Li, Zheng Lin, Zhang Zijian, Yilong Fan, Haoran Zheng, and Jyh-Shing Roger Jang. Gradient starvation in binary-reward grpo: Why group-mean centering fails and why the simplest fix works. *arXiv preprint arXiv:2605.07689*, 2026. 1, 2

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023. 6

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1, 2, 6

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025. 2

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026. 2

Chen Wang, Zhaochun Li, Jionghao Bai, Yining Zhang, Hexuan Deng, Ge Lan, and Yue Wang. Distilled reinforcement learning for llm post-training. *arXiv preprint arXiv:2607.17247*, 2026a. 2

Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Ge Lan, and Yue Wang. Targeted exploration via unified entropy control for reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16789–16802, 2026b. 2

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. In *International Conference on Learning Representations*, volume 2026, pp. 49450–49483, 2026. 1, 9

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. 2

Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. Kdrl: Post-training reasoning llms via unified knowledge distillation and reinforcement learning. *arXiv preprint arXiv:2506.02208*, 2025. 3

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 6

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026. 1

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026. 2

Yifan Zhang and Math-AI Team. American invitational mathematics examination (aime) 2024, 2024. 6

Yifan Zhang and Math-AI Team. American invitational mathematics examination (aime) 2025, 2025. 6

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for llm reasoning. *arXiv preprint arXiv:2602.22495*, 2026. 3

Appendix Contents

A Notation and Conventions **15**

B Failure Modes of On-Policy Distillation **16**

C Theoretical Analysis **16**

C.1 Bounded Teacher Guidance as a Policy Gradient 16

C.2 Difficulty Gating and Relative Supervision Strength 18

C.3 Bounded Combined Token-Level Advantage 19

D Implementation Details **20**

D.1 Models, Data and Baselines 20

D.2 Algorithm 21

D.3 Training Cost 21

D.4 Hyperparameter Sweep 21

D.5 Evaluation Protocol 22

E Additional Experimental Results **23**

E.1 Out-of-Distribution Transfer to Science 24

E.2 Scaling to Qwen3-4B Students 24

E.3 Rollout Group Size 25

E.4 Bounded versus Unbounded Teacher Term 25

E.5 Difficulty Gating and the Additive GRPO+OPD Baseline 26

E.6 Advantage Dynamics: DiffGate versus OPD and GRPO 26

E.7 Training Reward and Response Length 28

E.8 Why the Gate, in One View 29

E.9 Where Does DiffGate Help? Difficulty and Teacher Reliability 31

E.10 Training Dynamics and Performance 31

F Qualitative Examples **32**

F.1 Code Examples 32

F.2 Mathematics Examples 34

G Limitations and Future Work **36**

H Use of Large Language Models (LLMs) **36**

A Notation and Conventions

Table 3 provides a compact reference for the notation used throughout the paper, covering the policy setup, outcome-based supervision, teacher guidance, and the key components of DiffGate.

Table 3: **Summary of notation.** Symbols used for the policy setup, outcome supervision, teacher guidance, and DiffGate optimization.

Notation	Description
General setup	
$\mathcal{D}$	Distribution over training prompts.
x	Prompt sampled from $\mathcal{D}$.
π_θ	Student policy parameterized by θ.
$\pi_{\theta_{\text{old}}}$	Frozen behavior policy used to generate trajectories for the current policy update.
π_{T}	Fixed teacher policy.
$y^{(i)} = (y_1^{(i)}, \ldots, y_{L_i}^{(i)})$	i-th student-generated trajectory of length L_i.
$c_{i,t} = (x, y_{<t}^{(i)})$	Prefix context preceding token $y_t^{(i)}$ in trajectory i.
n	Number of trajectories sampled per prompt.
$\text{sg}[\cdot]$	Stop-gradient operator; its argument is treated as constant during differentiation.
Trajectory-level outcome supervision	
$R_i = R(x, y^{(i)})$	Binary verifier reward for trajectory i: $R_i = 1$ if the response passes the verifier and $R_i = 0$ otherwise.
$\bar{R}$	Mean reward within a rollout group, $\bar{R} = \frac{1}{n}\sum_{i=1}^{n} R_i$.
σ_R	Sample standard deviation of rewards within the rollout group, $\sigma_R = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(R_i - \bar{R})^2}$.
p	Group solve rate, $p = \frac{1}{n}\sum_{i=1}^{n} R_i$. For binary rewards, $p = \bar{R}$.
A_i^{GRPO}	Group-relative trajectory advantage, $A_i^{\text{GRPO}} = \frac{R_i - \bar{R}}{\sigma_R + \varepsilon}$.
ε	Positive constant used to stabilize reward normalization when σ_R is small or zero.
Token-level teacher guidance	
$\Delta_{i,t}$	Sampled-token teacher–student log-probability gap, $\Delta_{i,t} = \log \pi_{\text{T}}(y_t^{(i)} \mid c_{i,t}) - \log \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid c_{i,t})$.
$B_{i,t}$	Bounded teacher signal, $B_{i,t} = 2\tanh\left(\frac{\Delta_{i,t}}{2}\right) \in (-2, 2)$.
DiffGate components and optimization	
w_i	Trajectory-level teacher gate, $w_i = (1-p)^\gamma(1 - R_i)$, which activates teacher guidance only on failed trajectories and scales it by group difficulty.
γ	Nonnegative exponent controlling the dependence of teacher guidance on group difficulty. Setting $\gamma = 0$ removes difficulty weighting while retaining failed-trajectory selection.
λ	Nonnegative coefficient controlling the overall strength of teacher guidance. Setting $\lambda = 0$ recovers GRPO.
$G_{i,t}$	Combined token-level advantage used by DiffGate, $G_{i,t} = A_i^{\text{GRPO}} + \lambda w_i B_{i,t}$.
$\rho_{i,t}(\theta)$	Token-level importance ratio, $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)} \mid c_{i,t})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid c_{i,t})}$.
ϵ_{clip}	PPO clipping parameter used in the clipped policy surrogate.

B Failure Modes of On-Policy Distillation

Analysis setup. We next examine whether teacher–student disagreement can itself indicate when distillation is needed. Using a Qwen3-1.7B student and a frozen Qwen3-8B teacher, we evaluated rollouts from MATH500, DeepMath, and MBPP+. For each trajectory, we measured the average token-level teacher disagreement and compared it with verifier correctness using AUROC, where 0.5 indicates chance-level discrimination.

Teacher disagreement is largely correctness-blind. On mathematics, correct and incorrect trajectories show almost identical levels of disagreement. On MATH500, the averages are 0.150 and 0.148 nats/token, giving an AUROC of 0.50; on DeepMath, they are 0.188 and 0.184, with AUROC 0.49. Thus, the magnitude of the local teacher signal provides essentially no useful information about whether a mathematical trajectory is correct. On MBPP+, incorrect trajectories have somewhat larger disagreement than correct ones (0.551 versus 0.478 nats/token), but the signal remains only moderately predictive, with AUROC 0.60.

Large disagreement does not necessarily indicate the reasoning error. Aggregate disagreement can appear more predictive, but it is also strongly affected by response length: longer trajectories naturally accumulate more disagreement, and response length alone is already predictive of correctness. Moreover, the largest teacher–student discrepancies frequently occur on ordinary words or identifiers rather than on tokens directly associated with the reasoning failure. Hence, a large local disagreement does not reliably tell us either *whether* the trajectory is wrong or *where* the error occurs.

Implication for DiffGate. These observations motivate explicitly separating two complementary roles. The verifier provides trajectory-level information about *when* correction is needed, while the teacher provides dense token-level information about *how* the student should be updated. DiffGate therefore activates teacher guidance only on failed trajectories, where additional corrective supervision is most useful, rather than applying the teacher signal uniformly to both correct and incorrect rollouts.

C Theoretical Analysis

We establish three properties of DiffGate. First, the bounded teacher signal admits an exact fixed-prefix policy-gradient interpretation through an f-divergence. Second, the difficulty gate increases the relative importance of teacher guidance on harder groups and supplies a learning signal on all-failure groups where GRPO is silent. Third, both the teacher contribution and the resulting token-level advantage are bounded. These results characterize the learning signal used by DiffGate; they do not establish convergence of the complete clipped PPO objective.

C.1 Bounded Teacher Guidance as a Policy Gradient

Problem Setup. Consider a fixed rollout prefix c and a finite token vocabulary $\mathcal{V}$. Let

$$p_{\mathrm{old}}(a) \coloneqq \pi_{\theta_{\mathrm{old}}}(a \mid c), \qquad p_\theta(a) \coloneqq \pi_\theta(a \mid c), \qquad q(a) \coloneqq \pi_{\mathrm{T}}(a \mid c), \qquad a \in \mathcal{V}, \tag{14}$$

where p_{old} is the frozen behavior policy, p_θ is the student being optimized, and q is the fixed teacher. We assume positive probability on every token. Consistent with Equation equation 7, define

$$\Delta_{\mathrm{old}}(a) \coloneqq \log q(a) - \log p_{\mathrm{old}}(a), \tag{15}$$

and its bounded transformation

$$B_{\mathrm{old}}(a) \coloneqq 2 \tanh\left(\frac{\Delta_{\mathrm{old}}(a)}{2}\right) = 2\frac{q(a) - p_{\mathrm{old}}(a)}{q(a) + p_{\mathrm{old}}(a)}. \tag{16}$$

Hence $|B_{\mathrm{old}}(a)| < 2$, and its sign agrees with the teacher–student log-probability gap.

Proposition C.1 (Fixed-prefix gradient identity)**.** *Define*

$$f(r) \coloneqq 2(r-1) - 4\log\left(\frac{r+1}{2}\right), \qquad r > 0, \tag{17}$$

and the corresponding f-divergence

$$D_f(p_\theta \| q) \coloneqq \sum_{a \in \mathcal{V}} q(a) f\left(\frac{p_\theta(a)}{q(a)}\right). \tag{18}$$

Consider the detached sampled-token surrogate

$$\mathcal{L}_{\mathrm{BPG},c}(\theta) \coloneqq -\mathbb{E}_{a\sim p_{\mathrm{old}}}\left[\mathrm{sg}[B_{\mathrm{old}}(a)]\log p_\theta(a)\right]. \tag{19}$$

Then f is convex, $f(1)=0$, and

$$\nabla_\theta \mathcal{L}_{\mathrm{BPG},c}(\theta)|_{\theta=\theta_{\mathrm{old}}} = \nabla_\theta D_f(p_\theta\|q)|_{\theta=\theta_{\mathrm{old}}}. \tag{20}$$

Proof. Differentiating f gives

$$f'(r) = 2\frac{r-1}{r+1}, \qquad f''(r) = \frac{4}{(r+1)^2} > 0. \tag{21}$$

Thus f is convex, and direct substitution gives $f(1)=0$.

Using Eq. equation 16,

$$B_{\mathrm{old}}(a) = -f'\left(\frac{p_{\mathrm{old}}(a)}{q(a)}\right). \tag{22}$$

Since q is fixed,

$$\nabla_\theta D_f(p_\theta\|q) = \sum_{a\in\mathcal{V}} f'\left(\frac{p_\theta(a)}{q(a)}\right)\nabla_\theta p_\theta(a) \tag{23}$$

$$= \mathbb{E}_{a\sim p_\theta}\left[f'\left(\frac{p_\theta(a)}{q(a)}\right)\nabla_\theta \log p_\theta(a)\right]. \tag{24}$$

Evaluating at $\theta=\theta_{\mathrm{old}}$ and applying Eq. equation 22,

$$\nabla_\theta D_f(p_\theta\|q)|_{\theta=\theta_{\mathrm{old}}} = -\mathbb{E}_{a\sim p_{\mathrm{old}}}\left[B_{\mathrm{old}}(a)\,\nabla_\theta \log p_\theta(a)|_{\theta=\theta_{\mathrm{old}}}\right]. \tag{25}$$

This is exactly the gradient of Eq. equation 19 at $\theta=\theta_{\mathrm{old}}$, proving the result. □

Interpretation. Proposition C.1 gives a local interpretation of the bounded teacher signal: at a fixed prefix and at the behavior policy, minimizing the detached policy-gradient surrogate follows the same first-order direction as minimizing $D_f(p_\theta\|q)$.

The divergence also has the equivalent form

$$D_f(p_\theta\|q) = 4D_{\mathrm{KL}}\left(q\,\middle\|\,\frac{p_\theta+q}{2}\right). \tag{26}$$

Indeed, the linear term in f vanishes after summing over the two normalized distributions. Thus, the bounded teacher signal corresponds locally to a valid divergence while avoiding an unbounded sampled-token log-probability ratio.

Connection to reverse-KL OPD. At the same fixed prefix,

$$\nabla_\theta D_{\mathrm{KL}}(p_\theta\|q)|_{\theta=\theta_{\mathrm{old}}} = -\mathbb{E}_{a\sim p_{\mathrm{old}}}\left[\Delta_{\mathrm{old}}(a)\,\nabla_\theta \log p_\theta(a)|_{\theta=\theta_{\mathrm{old}}}\right], \tag{27}$$

where the additive constant from differentiating the reverse KL vanishes by the score-function identity. Moreover,

$$2\tanh\left(\frac{z}{2}\right) = z - \frac{z^3}{12} + O(z^5), \qquad z\to 0. \tag{28}$$

Hence the bounded and standard OPD signals agree to first order for small teacher–student gaps, while the bounded signal saturates for large discrepancies.

Remark C.2 (Relation to the implemented objective). The identity above is local: it holds at a fixed prefix and at $\theta=\theta_{\mathrm{old}}$. In DiffGate, $B_{i,t}$ is evaluated only on sampled student tokens, exactly as in Eq. equation 8; the full-vocabulary divergence in Eq. equation 18 is never computed.

The implemented method additionally multiplies $B_{i,t}$ by the trajectory-dependent gate w_i, adds the GRPO advantage, and optimizes the resulting detached signal using PPO clipping. Therefore, Proposition C.1 provides a fixed-prefix interpretation of the teacher component only; it does not imply that the complete DiffGate objective globally minimizes Eq. equation 18.

C.2 Difficulty Gating and Relative Supervision Strength

Consider a group of $n \geq 2$ trajectories with binary rewards $R_i \in \{0, 1\}$ and group solve rate

$$p = \frac{1}{n} \sum_{i=1}^{n} R_i. \tag{29}$$

For binary rewards, $p = \bar{R}$. Since we use the sample standard deviation defined in Section 3,

$$\sigma_R = \sqrt{\frac{n}{n-1} p(1-p)} = \sqrt{\kappa_n p(1-p)}, \qquad \kappa_n \coloneqq \frac{n}{n-1}. \tag{30}$$

The GRPO advantage, difficulty gate, and combined DiffGate signal are therefore

$$A_i^{\mathrm{GRPO}} = \frac{R_i - p}{\sqrt{\kappa_n p(1-p)} + \varepsilon}, \tag{31}$$

$$w_i = (1-p)^{\gamma}(1 - R_i), \tag{32}$$

$$G_{i,t} = A_i^{\mathrm{GRPO}} + \lambda w_i B_{i,t}. \tag{33}$$

For a failed trajectory, $R_i = 0$, these reduce to

$$A_i^{\mathrm{GRPO}} = -\frac{p}{\sqrt{\kappa_n p(1-p)} + \varepsilon}, \qquad w_i = (1-p)^{\gamma}. \tag{34}$$

Proposition C.3 (Difficulty-dependent relative teacher strength)**.** *For a failed trajectory in a mixed group, $0 < p < 1$, the magnitude of the teacher contribution relative to the GRPO advantage satisfies*

$$\frac{|\lambda w_i B_{i,t}|}{|A_i^{\mathrm{GRPO}}|} \leq 2\lambda(1-p)^{\gamma} \frac{\sqrt{\kappa_n p(1-p)} + \varepsilon}{p} =: U_n(p). \tag{35}$$

For fixed $\lambda \geq 0$, $\gamma \geq 0$, $\varepsilon > 0$, and $n \geq 2$, $U_n(p)$ is non-increasing on $p \in (0, 1)$.

Proof. For a failed trajectory, $w_i = (1-p)^{\gamma}$. Since $|B_{i,t}| < 2$,

$$|\lambda w_i B_{i,t}| < 2\lambda(1-p)^{\gamma}. \tag{36}$$

Moreover,

$$|A_i^{\mathrm{GRPO}}| = \frac{p}{\sqrt{\kappa_n p(1-p)} + \varepsilon}. \tag{37}$$

Combining the two expressions gives Eq. equation 35.

Expanding the upper bound yields

$$U_n(p) = 2\lambda \left[\sqrt{\kappa_n} \frac{(1-p)^{\gamma+1/2}}{\sqrt{p}} + \varepsilon \frac{(1-p)^{\gamma}}{p} \right]. \tag{38}$$

For $\gamma \geq 0$, both terms are non-increasing on $p \in (0, 1)$; therefore, their nonnegative sum is also non-increasing. □

Boundary behavior. When $p = 0$, every rollout fails. The GRPO advantages vanish and $w_i = 1$, giving the following advantage:

$$G_{i,t} = \lambda B_{i,t}. \tag{39}$$

Thus, the teacher receives its maximum *gate weight* on all-failure groups, where GRPO provides no learning signal. For a mixed group, $0 < p < 1$, both signals are available on failed trajectories, and Proposition C.3 shows that an upper bound on the teacher contribution relative to GRPO decreases as the group solve rate increases.

At the opposite boundary, $p = 1$, every trajectory succeeds. In this case, $A_i^{\mathrm{GRPO}} = 0$ and $w_i = 0$, so

$$G_{i,t} = 0. \tag{40}$$

The average gate weight across the group also has a simple form:

$$\frac{1}{n} \sum_{i=1}^{n} w_i = (1-p)^{\gamma+1}. \tag{41}$$

Hence, the average teacher weight decreases as the group solve rate increases.

Remark C.4 (Scope of the relative-strength result). Proposition C.3 concerns an upper bound on the relative magnitudes of the two scalar advantage terms. It does not imply that the realized teacher contribution is monotonic at every token, since $|B_{i,t}|$ varies across tokens and policies. It also does not establish agreement between teacher and outcome supervision, improvement in verifier reward, or a bound on the resulting parameter update.

C.3 Bounded Combined Token-Level Advantage

The bounded teacher transformation also provides direct control over the scale of the scalar learning signal supplied to the policy surrogate.

Proposition C.5 (Bounded combined advantage)**.** *For a rollout group of size $n \geq 2$ with binary rewards and sample standard-deviation normalization,*

$$\left|A_i^{\mathrm{GRPO}}\right| \leq \frac{n-1}{\sqrt{n}}. \tag{42}$$

Since $0 \leq w_i \leq 1$ and $|B_{i,t}| < 2$, the teacher contribution satisfies

$$|\lambda w_i B_{i,t}| < 2\lambda, \tag{43}$$

and therefore

$$|G_{i,t}| < \frac{n-1}{\sqrt{n}} + 2\lambda. \tag{44}$$

Proof. If $p \in \{0, 1\}$, all rewards are identical and $A_i^{\mathrm{GRPO}} = 0$. We therefore consider $0 < p < 1$.

For a successful trajectory, $R_i = 1$,

$$|A_i^{\mathrm{GRPO}}| = \frac{1-p}{\sqrt{\kappa_n p(1-p)} + \varepsilon} \tag{45}$$

$$\leq \sqrt{\frac{1-p}{\kappa_n p}} = \sqrt{\frac{n-1}{n}\frac{1-p}{p}}. \tag{46}$$

A mixed group contains at least one successful rollout, so $p \geq 1/n$. Therefore,

$$|A_i^{\mathrm{GRPO}}| \leq \sqrt{\frac{n-1}{n}(n-1)} = \frac{n-1}{\sqrt{n}}. \tag{47}$$

For a failed trajectory, $R_i = 0$,

$$|A_i^{\mathrm{GRPO}}| = \frac{p}{\sqrt{\kappa_n p(1-p)} + \varepsilon} \tag{48}$$

$$\leq \sqrt{\frac{p}{\kappa_n(1-p)}} = \sqrt{\frac{n-1}{n}\frac{p}{1-p}}. \tag{49}$$

Because a mixed group contains at least one failed rollout, $p \leq (n-1)/n$, giving

$$|A_i^{\mathrm{GRPO}}| \leq \frac{n-1}{\sqrt{n}}. \tag{50}$$

Finally, $w_i \leq 1$ and $|B_{i,t}| < 2$ imply $|\lambda w_i B_{i,t}| < 2\lambda$. Applying the triangle inequality to $G_{i,t} = A_i^{\mathrm{GRPO}} + \lambda w_i B_{i,t}$ completes the proof. □

Remark C.6 (What the bound guarantees). Equation equation 44 bounds the scalar token-level advantage supplied to the PPO surrogate. It does not by itself bound parameter-gradient norms, parameter movement, KL drift between successive policies, or guarantee improvement in task reward. These quantities additionally depend on the policy Jacobian, optimizer, and PPO clipping dynamics.

Group	Setting	Value
Models	Student	Qwen3-0.6B / Qwen3-1.7B (also Qwen3-4B, App. E.2)
	Teacher	Qwen3-8B, frozen (also Qwen3-14B, App. E.2)
	Mode	Non-thinking; shared chat template
Optimization	Optimizer	AdamW
	Learning rate	3×10^{-6}, cosine schedule, no warmup
	Prompts per step	128; PPO mini-batch 32
	PPO clip ϵ	0.2
	Entropy bonus	0
Rollouts	Group size n	16 (DiffGate, GRPO); 1 (OPD baselines)
	Sampling	Temperature 1.0, top-p 1.0
	Lengths	Prompt $\leq$ 2048, response $\leq$ 4096 tokens
	Reward	Binary: answer matching (math), unit tests (code)
DiffGate	Teacher strength λ	1
	Difficulty exponent γ	1
	Bounded signal	$2\tanh(\Delta_{i,t}/2) \in (-2, 2)$
Regularization	KL in reward	None
	KL in loss	None
	Reference-policy KL	Not used
Budget	Mathematics	171 steps; 21,888 prompt presentations
	Code	62 steps; 7,936 prompt presentations
	Hardware	1 node, 4×A100-80GB, FSDP
Evaluation	Decoding	16 generations; $T = 0.6$, top-$p = 1.0$, repetition penalty 1.15
	Length	$\leq$ 16,384 new tokens; avg@8 and pass@8 from the same pool

Table 4: **Training and evaluation configuration.** Shared across methods unless otherwise specified. Training uses 128 prompts per step. The mathematics and code training sets contain 21,928 and 4,082 prompts, respectively; the reported budgets correspond to approximately 1.00 and 1.94 dataset passes. GRPO-based methods differ only in their teacher-guidance configuration.

D Implementation Details

This section presents the experimental setup and implementation details, including the models, datasets, baselines, training and evaluation configurations, computational cost, hyperparameter analysis, and the complete DiffGate algorithm.

D.1 Models, Data and Baselines

Models. We use Qwen3-0.6B and Qwen3-1.7B students with a frozen Qwen3-8B teacher, and extend the evaluation to Qwen3-4B students with Qwen3-8B and Qwen3-14B teachers in Appendix Section E.2. All models operate in non-thinking mode with a shared chat template, allowing the teacher to evaluate the student-generated tokens under the same prefix context.

Training Data. We train on DeepMath (He et al., 2026), retaining mathematics problems with difficulty greater than 6, and on TACO (Li et al., 2023) for code generation. Mathematics responses receive a binary reward based on final-answer matching, while code responses are evaluated using unit tests executed in a sandbox. The training sets contain 21,928 mathematics prompts and 4,082 code prompts. With 128 prompts per step, we train for 171 mathematics steps and 62 code steps, corresponding to 21,888 and 7,936 prompt presentations, respectively.

Baselines and Implementation. We implement GRPO, DiffGate, the boundedness and gating ablations (Appendix Sections E.4 and E.5), and the forward- and reverse-KL OPD baselines using a common training framework built on verl[1]. These methods share the data pipeline, rollout engine, verifier, and optimizer. GRPO is recovered by setting $\lambda = 0$, while setting $\gamma = 0$ yields the additive GRPO+OPD baseline without difficulty weighting. Thus, the GRPO-based variants differ only in

[1] https://github.com/volcengine/verl

their teacher-guidance configuration. We train EOPD (Jin et al., 2026) and Uni-OPD (Hou et al., 2026) using their original implementations, with the same training data and student–teacher pairs. The SFT baseline fine-tunes the student on teacher-generated solutions for the same prompts.

Training Configuration. Table 4 summarizes the training and evaluation settings. Within the common implementation, we use the same optimizer, prompt batch size, and training-step budget without method-specific optimizer tuning. GRPO-based methods sample $n = 16$ trajectories per prompt, whereas the OPD baselines use a single trajectory. We fix $\lambda = \gamma = 1$ for DiffGate across all model sizes and domains, without per-domain tuning. No method uses a reference-policy KL penalty in its reward or training loss.

Computational Resources and Software Environment. All training and evaluation experiments were conducted on Ubuntu 22.04 Linux nodes, each equipped with four NVIDIA A100 80GB GPUs. Each run reported in Tables 1 and 2 used a single node. The software environment comprised Python 3.12, PyTorch 2.8.0[2] with CUDA 12.9[3], FlashAttention 2.8.3[4], and Transformers 4.57.1[5]. We used vLLM 0.11.0 [6] for rollout generation during training and vLLM 0.18.0 for evaluation. Training was performed using our verl-based implementation, while EOPD and Uni-OPD were run using the original implementations released by their respective authors.

D.2 Algorithm

The verifier determines *which trajectories* receive teacher guidance, while the teacher supplies bounded token-level update directions within the selected failed trajectories. The gate therefore operates at the trajectory level; DiffGate does not attempt to localize the causal error token within a failed response. For successful trajectories, $R_i = 1$ and hence $w_i = 0$, so no teacher update is applied. For all-failure groups, $p = 0$ and the GRPO advantage vanishes, making teacher guidance maximal. Setting $\lambda = 0$ recovers GRPO, whereas $\gamma = 0$ removes difficulty weighting and yields a flat teacher-guidance baseline.

Arm	**Rollouts / prompt**	**Math 0.6B**	**Math 1.7B**	**Code 0.6B**	**Code 1.7B**
		171 steps		*62 steps*	
OPD (forward KL)	1	4	4.7	1.5	5
OPD (reverse KL)	1	4.1	4.5	1.4	1.5
EOPD	1	4.2	4.7	4	4
Uni-OPD	8	15	17	14	13
GRPO	16	36	39	16	19
DiffGate	16	38	37	14	19

Table 5: **Training cost.** Wall-clock hours on 4× A100-80GB for the runs behind main Tables 1 and 2.

D.3 Training Cost

Table 5 reports wall-clock training time on one node with 4× A100-80GB GPUs. DiffGate has comparable computational cost to GRPO, with measured wall-clock times differing by at most 12.5% across the four settings. Standard OPD baselines are generally cheaper because they use fewer rollouts per prompt, while DiffGate and GRPO both use grouped on-policy generation.

D.4 Hyperparameter Sweep

Figure 5 varies the teacher strength λ and the difficulty exponent γ on the 1.7B student. Every point is taken at the same training step within a domain (mathematics 100, code 62, the last step all sweep arms share), so the values differ from those of Tables 1 and 2; the no-gate control $\gamma = 0$ is reported at its final step, and $\lambda = 0.25$ was not run on code. We summarize the key insights as follows:

[2] https://pytorch.org/
[3] https://developer.nvidia.com/cuda-toolkit
[4] https://github.com/Dao-AILab/flash-attention
[5] https://github.com/huggingface/transformers
[6] https://github.com/vllm-project/vllm

Algorithm 1 DiffGate: Difficulty-Gated On-Policy Distillation

Require: Student policy π_θ, fixed teacher π_{T}, prompt distribution $\mathcal{D}$, verifier R, group size n, teacher weight λ, difficulty exponent γ, PPO clipping parameter ϵ

for each training iteration **do**
  Freeze behavior policy: $\pi_{\theta_{\mathrm{old}}} \leftarrow \pi_\theta$
  Sample a batch of prompts $x \sim \mathcal{D}$
  for each prompt x **do**
    Sample n on-policy trajectories: $y^{(i)} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x), \qquad i = 1, \ldots, n$
    Evaluate trajectory rewards: $R_i = R(x, y^{(i)}) \in \{0, 1\}$
    Compute group statistics:

$$p = \frac{1}{n}\sum_{i=1}^{n} R_i, \qquad A_i^{\mathrm{GRPO}} = \frac{R_i - \bar{R}}{\sigma_R + \varepsilon}, \qquad \bar{R} = \frac{1}{n}\sum_{j=1}^{n} R_j$$

    for each trajectory i and token t **do**
      Define context: $c_{i,t} = (x, y^{(i)}_{<t})$
      Compute teacher–student discrepancy:

$$\Delta_{i,t} = \log \pi_{\mathrm{T}}(y_t^{(i)} \mid c_{i,t}) - \log \pi_{\theta_{\mathrm{old}}}(y_t^{(i)} \mid c_{i,t})$$

      Bound the local teacher signal: $B_{i,t} = 2\tanh\left(\frac{\Delta_{i,t}}{2}\right)$
      Compute difficulty gate: $w_i = (1-p)^\gamma (1 - R_i)$
      Construct difficulty-gated token advantage: $G_{i,t} = A_i^{\mathrm{GRPO}} + \lambda w_i B_{i,t}$
    end for
  end for
  Compute token-level importance ratios:$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)} | c_{i,t})}{\pi_{\theta_{\mathrm{old}}}(y_t^{(i)} | c_{i,t})}$
  Update θ using the clipped surrogate:

$$\mathcal{L}_{DiffGate} = -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{L_i}\sum_{t=1}^{L_i} \min\left(\rho_{i,t}(\theta)\,\mathrm{sg}[G_{i,t}], \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\,\mathrm{sg}[G_{i,t}]\right)$$

end for

- **Moderate teacher strength.** Mathematics avg@8 is 21.4–21.9 for $\lambda \leq 1$ and falls to 19.6 and 19.0 at $\lambda = 2$ and 4. Code peaks at $\lambda = 0.5$ (46.2) and falls to 42.7 at $\lambda = 8$.
- **Gating matters more than its sharpness.** Removing the gate ($\gamma = 0$) costs 3.2 avg@8 against the λ=1 point of this sweep on mathematics (18.7 against 21.9) and is neutral on code (43.8 against 43.4–45.2); the matched comparison at the full step budget, in Table 11, puts the same gap at 2.4. Varying $\gamma \in \{1, 2, 3\}$ changes avg@8 by at most 1.8 on mathematics and 2.2 on code, with no consistent direction.
- **One setting for every cell.** All reported runs use $\lambda = \gamma = 1$. The best code setting in the sweep, $\lambda = 0.5$, is 1.0 avg@8 above the reported $\lambda = 1$ run (46.2 against 45.2) on a single seed; we did not tune per domain.

D.5 Evaluation Protocol

This section provides additional details on the evaluation protocol, including metric computation, benchmark composition, answer and code verification, randomness, and evaluation record keeping.

Metric computation. For each test problem, we generate 8 responses. We report two metrics: avg@8 and pass@8. Avg@8 is the mean correctness across the eight generated responses. Pass@8 measures whether at least one of the eight sampled responses is correct, following the standard pass@k formulation of Chen et al. (2021). When reporting results averaged across multiple benchmarks, we use the unweighted mean of the corresponding per-benchmark scores.

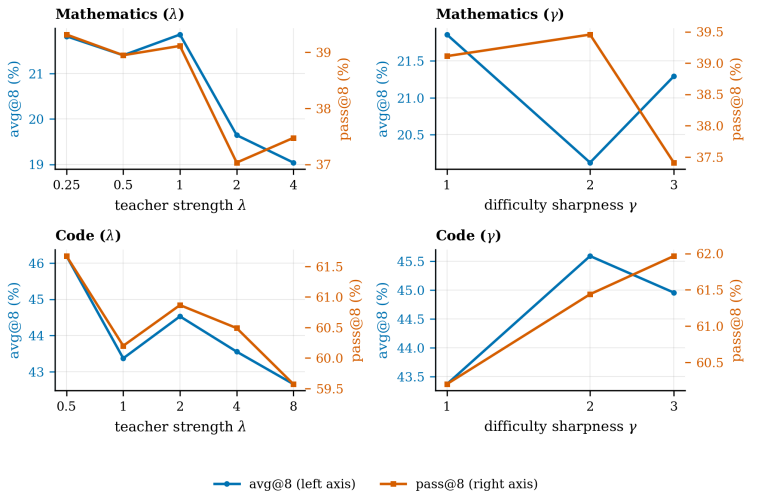

Figure 5: **Hyperparameter sweep, 1.7B student.** Teacher strength λ (left) and difficulty sharpness γ (right), for mathematics (top) and code (bottom); avg@8 on the left axis, pass@8 on the right. Every point is DiffGate at a different setting, so this figure varies one constant rather than comparing arms. Hollow markers are the reported configuration $\lambda = \gamma = 1$.

Randomness. Training uses seed 42 for data ordering and batching. Evaluation does not explicitly set a sampling seed in vLLM. Unless otherwise stated, each reported model corresponds to a single training run.

Benchmark composition. The mathematics evaluation contains Minerva-Math (272 problems), OlympiadBench English open-ended mathematics (674), AIME24 (30), AIME25 (30), and HMMT25 (30). Code evaluation uses MBPP+ (378), HumanEval+ (164), and 454 code-generation problems from LiveCodeBench. The science evaluation uses GPQA-Diamond (198), MMLU-STEM (3,153 questions across 19 STEM subjects), and OlympiadBench-Physics (233).

Mathematics and science verification. For mathematics, we first extract an answer from an `<answer>` tag when available, otherwise from the final `\boxed{}` expression, and finally from the last numerical expression in the response. We use Math-Verify[7] to determine equivalence with the reference answer after standard normalization. If symbolic verification fails, normalized strings are compared directly, with numerical answers accepted within a tolerance of 10^{-4}. For GPQA and MMLU-STEM, we extract the selected option from the answer tag or, if absent, the final stated option.

Code verification. Generated code is extracted from the final fenced Python block or, when no fenced block is present, from the first top-level definition or import onward. Each execution is isolated in a Python subprocess with fixed time and memory limits for safe and reproducible evaluation. HumanEval+ and MBPP+ use the EvalPlus test harnesses, and a response is marked correct only if it passes all tests. LiveCodeBench and TACO use their provided input–output tests; outputs are compared after removing leading and trailing whitespace.

Contamination and evaluation records. DeepMath-103K was decontaminated by its authors against common mathematics benchmarks (He et al., 2026). We did not perform an additional decontamination analysis for DeepMath or TACO against the evaluation sets used here. For each checkpoint and benchmark, we store per-problem correctness for all 16 generations together with the extracted answer and first complete response.

E Additional Experimental Results

This section provides supplementary experimental results and analyses that complement the main paper. We report out-of-distribution evaluation on science benchmarks, additional ablations on key design choices, training-dynamics analyses, and rollout-group-size experiments. These results further characterize when DiffGate is most effective and help isolate the contribution of each component.

[7] https://github.com/huggingface/Math-Verify

Method	GPQA		MMLU-STEM		OlympiadBench-Physics		Average	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
Qwen3-0.6B mathematics student								
Base	13.0	50.8	48.5	86.2	3.6	9.4	21.7	48.8
OPD (forward KL)	26.1	**77.3**	58.5	**88.5**	3.6	11.4	29.4	**59.1**
OPD (reverse KL)	26.5	74.2	59.0	87.8	4.8	12.6	30.1	58.2
EOPD	26.1	74.3	59.7	88.1	4.7	12.0	30.2	58.1
Uni-OPD	27.0	73.2	60.2	87.8	4.7	12.0	30.6	57.7
GRPO	12.8	52.8	50.1	85.2	**5.6**	12.2	22.8	50.1
DiffGate	**28.3**	74.4	**60.6**	87.7	4.6	**12.7**	**31.2**	58.3
Qwen3-1.7B mathematics student								
Base	14.4	57.6	59.2	90.8	9.0	21.1	27.5	56.5
OPD (forward KL)	32.1	**78.5**	68.0	**92.4**	7.3	18.2	35.8	**63.0**
OPD (reverse KL)	30.2	73.0	70.4	91.9	9.3	21.5	36.7	62.1
EOPD	30.7	76.4	68.5	92.1	9.3	20.6	36.2	**63.0**
Uni-OPD	31.0	76.2	68.9	91.6	9.6	21.2	36.5	**63.0**
GRPO	7.8	29.1	26.7	56.9	8.4	20.1	14.3	35.4
DiffGate	**33.2**	73.0	**74.3**	92.2	**10.0**	**22.2**	**39.2**	62.4
Teacher 8B	44.8	81.7	82.1	95.5	15.9	31.9	47.6	69.7

Table 6: **Out-of-distribution transfer to Science.** avg@8 / pass@8 of mathematics-trained students on GPQA, MMLU-STEM and OlympiadBench-Physics, with no science training; the final columns average the three and are the science columns of Table 1. **Bold** / underline: best / second-best per student.

Method	Minerva		Olympiad		AIME24		AIME25		HMMT25		Average	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
Base (Qwen3-4B)	38.8	52.2	51.2	71.7	20.4	43.3	17.5	35.6	7.5	**24.6**	27.1	45.5
OPD (forward KL)	38.9	**56.3**	48.5	70.4	20.8	36.5	15.0	32.4	7.9	19.1	26.2	43.0
OPD (reverse KL)	39.0	54.5	49.4	70.3	20.8	40.3	17.5	39.2	8.8	20.8	27.1	45.0
DiffGate	**41.8**	54.3	**53.3**	**72.1**	**22.1**	**47.1**	**21.2**	**40.6**	**11.7**	21.7	**30.0**	**47.2**
Teacher 14B	43.6	57.4	55.4	74.8	30.0	49.2	20.0	34.5	9.2	22.1	31.6	47.6

Table 7: **Mathematical reasoning with a Qwen3-4B student and Qwen3-14B teacher.** Avg@8 and pass@8 across five mathematics benchmarks. **Bold** and underline denote the best and second-best student results, respectively.

E.1 Out-of-Distribution Transfer to Science

Table 6 gives the per-benchmark scores behind the science columns of Table 1: mathematics-trained students evaluated on GPQA, MMLU-STEM, and OlympiadBench-Physics with no science training. The gains over the base student come from GPQA and MMLU-STEM (at 1.7B, DiffGate reaches 33.2 and 74.3 avg@8 against 14.4 and 59.2), while OlympiadBench-Physics stays within two points of the base student for every method.

E.2 Scaling to Qwen3-4B Students

We further evaluate DiffGate with a Qwen3-4B student, using a Qwen3-14B teacher for mathematics and a Qwen3-8B teacher for code. Due to computational constraints, we compare against the base model and forward- and reverse-KL OPD baselines. On mathematics, DiffGate achieves the strongest overall student performance, improving the base model from 27.1 to 30.0 avg@8 and from 45.5 to 47.2 pass@8 (Table 7). It also outperforms the best OPD baseline by 2.9 avg@8 and 2.2 pass@8 points, reaching within 0.4 pass@8 points of the 14B teacher. On code, DiffGate similarly achieves the highest overall student scores, reaching 57.3 avg@8 and 70.5 pass@8 compared with 54.1 and 69.8 for the base model (Table 8). Relative to the strongest OPD baseline, this corresponds to gains of 2.5 avg@8 and 2.8 pass@8 points. These results indicate that the benefits of combining outcome supervision with difficulty-gated teacher guidance extend to larger student models.

Method	MBPP+		HumanEval+		LCB		Average	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
Base (Qwen3-4B)	62.4	<u>78.5</u>	<u>73.9</u>	**89.8**	<u>26.0</u>	<u>41.1</u>	54.1	<u>69.8</u>
OPD (forward KL)	64.0	**78.7**	67.8	86.2	22.8	38.1	51.6	67.7
OPD (reverse KL)	<u>66.9</u>	77.6	71.8	85.7	25.6	39.7	<u>54.8</u>	67.7
DiffGate	**67.1**	**78.7**	**77.1**	<u>89.5</u>	**27.6**	**43.3**	**57.3**	**70.5**
Teacher 8B	70.8	81.6	80.0	90.2	28.1	43.1	59.6	71.7

Table 8: **Code generation with a Qwen3-4B student and Qwen3-8B teacher.** Avg@8 and pass@8 on MBPP+, HumanEval+, and LiveCodeBench. **Bold** and <u>underline</u> denote the best and second-best student results, respectively.

Method	Mathematics				Code			
	0.6B		1.7B		0.6B		1.7B	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
$n = 16$ rollouts per prompt								
GRPO	**14.2**	28.4	**21.4**	35.2	31.3	48.2	43.4	55.7
DiffGate	13.7	**29.5**	21.1	**39.1**	**33.0**	**49.8**	**45.2**	**61.4**
$n = 4$ rollouts per prompt								
GRPO	10.9	23.9	**21.3**	36.0	29.8	48.2	40.8	52.3
DiffGate	**12.1**	**27.9**	20.6	**39.3**	**31.4**	**49.2**	**44.0**	**60.4**

Table 9: **Effect of rollout group size.** GRPO and DiffGate trained with n=16 and n=4 rollouts per prompt, holding data, teacher, optimizer, and budget fixed; averages over the benchmark sets of Tables 1 and 2. The n=16 rows are the runs of Tables 1 and 2. Bold marks the better of the two arms within each group size.

E.3 Rollout Group Size

DiffGate is motivated by a simple observation: group-relative rewards provide no learning signal when all sampled rollouts receive the same outcome. Smaller rollout groups make such reward-uniform groups more common, providing a direct test of this mechanism. We therefore retrained GRPO and DiffGate with $n = 4$ instead of $n = 16$, keeping the data, teacher, optimizer, and training budget fixed (see Table 9). Reducing n from 16 to 4 lowers GRPO by 1.9 avg@8 points on average across the four settings, compared with 1.2 for DiffGate. The difference is largest on 0.6B mathematics, where GRPO drops by 3.3 points while DiffGate drops by 1.6. On code, DiffGate also degrades less at both model sizes. We summarize the key points as follows:

- **The advantage appears where reward is sparse.** On 0.6B mathematics DiffGate moves from slightly below GRPO at $n = 16$ to $+1.2$ avg@8 ahead at $n = 4$, and it keeps the higher pass@8 in all four settings at both group sizes. Small groups produce more all-failure batches; there, the outcome signal is zero while DiffGate still receives dense teacher guidance.
- **The gap is largest early in training.** With $n = 4$ on 0.6B mathematics, DiffGate reaches 11.8 avg@8 by step 25 against 7.3 for GRPO. All-failure groups are most frequent at the start and become rarer as the student improves, and the gap narrows with them.

E.4 Bounded versus Unbounded Teacher Term

The unbounded variant replaces $2\tanh(\Delta_{i,t}/2)$ with the raw log-probability gap $\Delta_{i,t}$ and changes nothing else. Bounding improves avg@8 by 1.0–1.5 points and pass@8 by 0.8–1.3 points in all four settings (see Table 10); the gradient statistics are in the lower panel of the table and in Figure 6.

Method	Mathematics				Code			
	0.6B		1.7B		0.6B		1.7B	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
Held-out accuracy								
DiffGate, bounded $2\tanh(\Delta/2)$	**13.7**	**29.5**	**21.1**	**39.1**	**33.0**	**49.8**	**45.2**	**61.4**
DiffGate, unbounded (raw Δ)	12.7	28.2	19.9	38.0	31.9	49.0	43.7	60.3
bounded − unbounded	+1.0	+1.3	+1.2	+1.1	+1.1	+0.8	+1.5	+1.1
Training gradient norm *(median / maximum over steps)*								
Method	*median*	*max*	*median*	*max*	*median*	*max*	*median*	*max*
DiffGate, bounded	0.31	1.7	0.21	0.7	0.97	4.3	0.80	2.6
DiffGate, unbounded	0.50	3.4	0.29	1.0	3.22	342.3	1.98	10.2

Table 10: **Bounded versus unbounded teacher term.** Top: held-out avg@8 / pass@8 of the two DiffGate variants. Bottom: median and maximum student gradient norm over training.

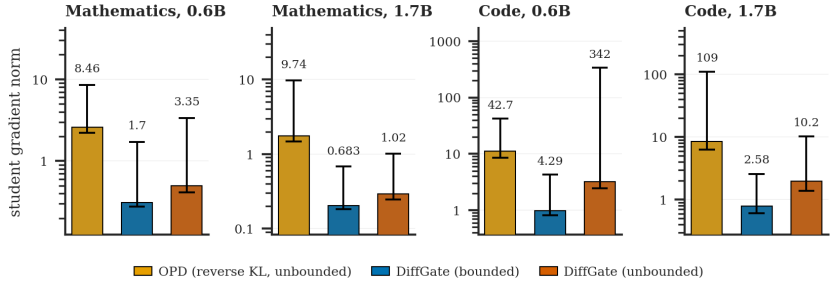

Figure 6: **Student gradient norm over training.** Bars: median. Whiskers: minimum to maximum step, with the maximum printed above. Log scale.

Why the bound matters. The teacher term $2\tanh(\Delta/2)$ passes small gaps through unchanged and caps large ones at 2. Most tokens are small gaps. A few are not: at the start of training, 2–9% of the tokens the teacher scores have a gap above 2 nats, and the largest in a step is 6–15 nats (Table 13). Unbounded, one such token enters the update with more weight than any GRPO advantage, which is at most 3.75; bounded, with at most 2. These tokens are most common on code, and that is where removing the bound does the most damage: the largest gradient step on 0.6B code grows from 4.3 to 342 (Figure 6). Accuracy is also higher with the bound in every setting (Table 10).

E.5 Difficulty Gating and the Additive GRPO+OPD Baseline

Table 11 compares DiffGate with the additive objective, which applies the bounded teacher term with a constant weight to every failed trajectory. The gate matters most on mathematics, where it raises avg@8 by 2.4 points at both model sizes and pass@8 by 2.8 and 4.1 points. On code, most of the effect of the teacher term is already present without the gate: gating adds 1.3 and 1.4 avg@8 and 0.1 and 1.4 pass@8. Appendix E.8 shows the mechanism: the gate concentrates the teacher on groups where the reward is silent, whereas the additive rule spends it on groups the reward already handles.

E.6 Advantage Dynamics: DiffGate versus OPD and GRPO

When all sixteen rollouts for a prompt receive the same reward, GRPO provides no learning signal because every trajectory has zero group-relative advantage. In an all-fail group, however, DiffGate can still use the teacher to provide token-level guidance; in an all-pass group, the teacher gate is intentionally zero. We use the training logs to examine how often such reward-uniform groups occur, whether restricting teacher guidance to failed rollouts leaves substantial teacher–student disagreement, and whether the gate concentrates supervision where GRPO is least informative. GRPO and DiffGate use the same data, teacher, group size, and training budget, and their statistics match at step 1, so subsequent differences reflect the training objective.

Reward-uniform groups are common, especially on code. At step 1, all rollouts fail for 32–36% of mathematics prompts and 14–35% of code prompts. This fraction decreases rapidly on mathematics

Objective	Mathematics				Code			
	0.6B		1.7B		0.6B		1.7B	
	avg@8	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*	*avg@8*	*pass@8*
GRPO + OPD (additive, γ=0)	11.3	26.7	18.7	35.0	31.7	49.7	43.8	60.0
DiffGate (γ=1)	**13.7**	**29.5**	**21.1**	**39.1**	**33.0**	**49.8**	**45.2**	**61.4**

Table 11: **Difficulty gating against the additive baseline.** Both arms are GRPO plus the same bounded teacher term on failed trajectories, matched on data, teacher, n=16 rollouts, and step budget. They differ only in that term's weight: constant (additive, γ=0) against the failed fraction of the group (DiffGate, γ=1).

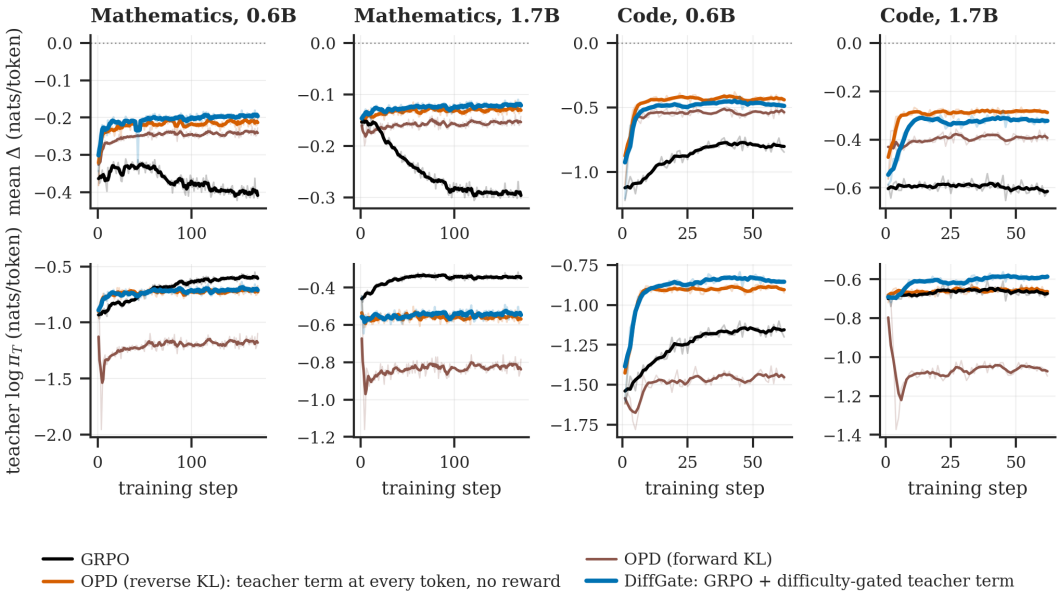

Figure 7: **Teacher-side training dynamics: GRPO, OPD and DiffGate.** Columns are the four model–domain settings. GRPO and DiffGate use n=16 rollouts and the outcome reward, DiffGate adding the gated teacher term; the two OPD baselines use one rollout per prompt and no reward. All four share data, teacher, and step budget. Faint traces are per-step values, bold lines are a five-step rolling mean. Rows: mean Δ over all response tokens, and the teacher's log-likelihood of the student's own samples.

as the student improves, but remains substantial on code: roughly 20% of 0.6B code prompts are still all-fail near the end of training. Including all-pass groups, 59–74% of code prompts can provide no GRPO signal at later checkpoints (Table 12). DiffGate addresses precisely the all-fail case by applying teacher guidance when GRPO is silent, while leaving already-solved all-pass groups unchanged. It directly explains why teacher guidance can be especially useful on code, where reward-uniform groups remain frequent throughout training.

Selective teacher guidance remains effective. Figure 7 tracks the teacher–student gap and the teacher log-likelihood of student-generated samples throughout training. By the end of training, DiffGate reaches a teacher–student gap comparable to reverse-KL OPD, despite applying teacher guidance only to failed rollouts. In contrast, the gap under GRPO increases on mathematics, from 0.15 to 0.29 for the 1.7B student. At the same time, the teacher assigns increasing likelihood to GRPO samples, suggesting that the larger gap does not simply reflect poorer generations. Together with the entropy trends in Figure 3b, this pattern is consistent with GRPO becoming more concentrated on a narrower subset of teacher-supported responses. GRPO remains competitive on avg@8 while DiffGate achieves higher pass@8.

The gate focuses teacher guidance on failed and difficult rollouts. Figure 8 examines the quantities directly used by the gate. Teacher disagreement is initially similar on correct and failed rollouts but becomes larger on correct rollouts later in training. By the end, it is 1.3–1.7× higher on correct rollouts under DiffGate and 1.5–3.5× under GRPO (Table 13). Applying teacher guidance uniformly

Quantity	Mathematics, 0.6B			Mathematics, 1.7B			Code, 0.6B			Code, 1.7B		
	Start	*GRPO*	*DiffGate*	*Start*	*GRPO*	*DiffGate*	*Start*	*GRPO*	*DiffGate*	*Start*	*GRPO*	*DiffGate*
Group-level reward statistics												
All-fail groups, $A_{\mathrm{GRPO}} \equiv 0$ (%)	32.4	7.5	10.8	35.6	2.6	2.6	35.4	22.3	19.6	13.7	9.2	5.7
All-pass groups, $A_{\mathrm{GRPO}} \equiv 0$ (%)	22.6	60.9	49.1	32.1	70.8	62.1	24.0	36.5	37.4	49.8	55.8	56.3
Prompts with no group signal (%)	55.0	68.5	59.8	67.7	73.5	64.8	59.4	58.8	57.0	63.5	65.0	62.1
Token-weighted mean A_{GRPO}	-0.010	-0.051	-0.030	-0.010	-0.052	-0.037	+0.004	-0.065	-0.023	-0.014	-0.053	-0.023
Teacher signal on failed rollouts												
$\lvert\Delta\rvert$ on failed rollouts (nats/token)	0.386	0.170	0.146	0.166	0.087	0.078	1.039	0.556	0.374	0.315	0.295	0.188
Bounded $\lvert 2\tanh(\Delta/2)\rvert$ on failed rollouts	0.245	0.117	0.111	0.129	0.065	0.063	0.310	0.242	0.199	0.115	0.112	0.107
Difficulty gate $(1-p)$ on failed rollouts	0.69	0.33	0.41	0.63	0.21	0.26	0.69	0.55	0.54	0.43	0.36	0.33
Failed-rollout tokens with $\lvert\Delta\rvert > 2$ (%)	5.6	2.3	1.6	1.7	1.0	0.7	11.2	6.8	4.9	3.7	3.6	2.4
Policy and training statistics												
Mean Δ, all tokens (nats/token)	-0.343	-0.401	-0.195	-0.143	-0.293	-0.121	-1.213	-0.802	-0.479	-0.587	-0.609	-0.321
Mean $\log \pi_T$, all tokens (nats/token)	-0.886	-0.599	-0.708	-0.445	-0.344	-0.542	-1.609	-1.164	-0.851	-0.678	-0.669	-0.592
Train solve rate (%)	22.4	60.8	49.3	30.8	71.2	63.4	22.6	38.0	37.1	49.4	55.6	56.4

Table 12: **Advantage and teacher-signal statistics over training.** "Start" is the mean of the two arms at step 1; GRPO and DiffGate report averages over the final ten steps. Reward-uniform groups (all-fail or all-pass) have zero group-relative advantage. We measure teacher-signal statistics only on failed rollouts, and we average over the final block of all generated response tokens.

Quantity	Mathematics, 0.6B		Mathematics, 1.7B		Code, 0.6B		Code, 1.7B	
	GRPO	*DiffGate*	*GRPO*	*DiffGate*	*GRPO*	*DiffGate*	*GRPO*	*DiffGate*
Teacher disagreement and reward-signal availability								
Correct/failed disagreement ratio	0.99→2.41	0.83→1.34	0.85→3.48	0.80→1.57	1.10→1.45	1.13→1.29	1.68→2.07	1.62→1.72
Teacher dose on GRPO-silent groups (%)	55→58	52→52	78→41	69→26	69→64	63→58	58→56	52→39
GRPO-silent groups, share of prompts (%)	26→8	28→11	36→3	31→3	40→22	32→20	15→9	13→6
Applied teacher guidance								
Gate factor, fired $\times(1-p)$	0.48→0.13	0.54→0.21	0.47→0.06	0.44→0.10	0.58→0.35	0.50→0.34	0.25→0.16	0.25→0.14
Applied teacher dose per token	0.112→0.016	0.122→0.023	0.065→0.004	0.061→0.006	0.182→0.086	0.145→0.068	0.033→0.018	0.034→0.016
Failed tokens pushed up (%)	–	41→20	–	21→12	–	44→24	–	12→13
Teacher confidence on failed-rollout tokens								
Teacher prob. on pushed-up tokens	–	0.68→0.43	–	0.56→0.31	–	0.69→0.55	–	0.49→0.38
Teacher prob. on all failed tokens	0.54→0.30	0.56→0.38	0.57→0.25	0.54→0.30	0.61→0.45	0.55→0.47	0.46→0.37	0.45→0.35
Tail of the raw gap Δ, before the bound								
Tokens with $\lvert\Delta\rvert > 2$ nats (%)	5.2→2.3	4.1→1.6	1.8→1.0	1.7→0.7	11.1→6.8	8.7→4.8	4.3→3.6	3.8→2.4
Largest $\lvert\Delta\rvert$ in a step (nats)	8.9→4.9	9.3→5.5	6.3→3.3	6.3→3.1	17.7→13.6	14.6→10.6	10.2→8.5	9.6→6.3
Optimisation *(median / maximum over training)*								
Student gradient norm	0.22 / 0.7	0.31 / 1.7	0.31 / 212.9	0.21 / 0.7	0.67 / 1.2	0.97 / 4.3	1.36 / 64.9	0.80 / 2.6

Table 13: **Teacher-side diagnostics over training.** Each entry reports start → end, where start" averages the first five training steps and end" averages the final ten. For GRPO, teacher-derived quantities are computed only for diagnostic purposes and are never used for optimization; rows that require an applied teacher update are therefore left blank. The disagreement block reports the raw teacher–student log-probability difference Δ before bounding, evaluated on teacher-scored tokens. The gradient row instead reports the median/maximum gradient norm over the full training run.

would therefore spend substantial supervision on trajectories that already satisfy the verifier. The factor $(1 - R_i)$ avoids this by restricting teacher guidance to failed rollouts. The average applied teacher weight also decreases as performance improves, from roughly 0.5 to 0.1–0.2 on mathematics, consistent with the intended difficulty-dependent behavior described in Section C.2.

E.7 Training Reward and Response Length

Figure 9 tracks the verifier pass rate over the 16 rollouts, the mean response length, and the policy entropy for GRPO, reverse-KL OPD, DiffGate and the additive variant with no difficulty weighting ($\gamma = 0$). All arms share the training set and step budget.

Teacher guidance affects response behavior early in training. One of the clearest early changes is response length. Teacher-guided models show a rapid shift in response length during the first few training steps, coinciding with the decrease in teacher–student disagreement shown in Figure 7, while GRPO changes more gradually. After this initial adjustment, response lengths become relatively stable, consistent with later teacher guidance reflecting less of the early response-format differences.

We observe no clear evidence of verifier exploitation. Mathematics rewards use symbolic answer checks, while code rewards rely on hidden unit tests executed in a sandbox, avoiding reliance on a learned reward model. We do observe behaviors such as response-length growth and format

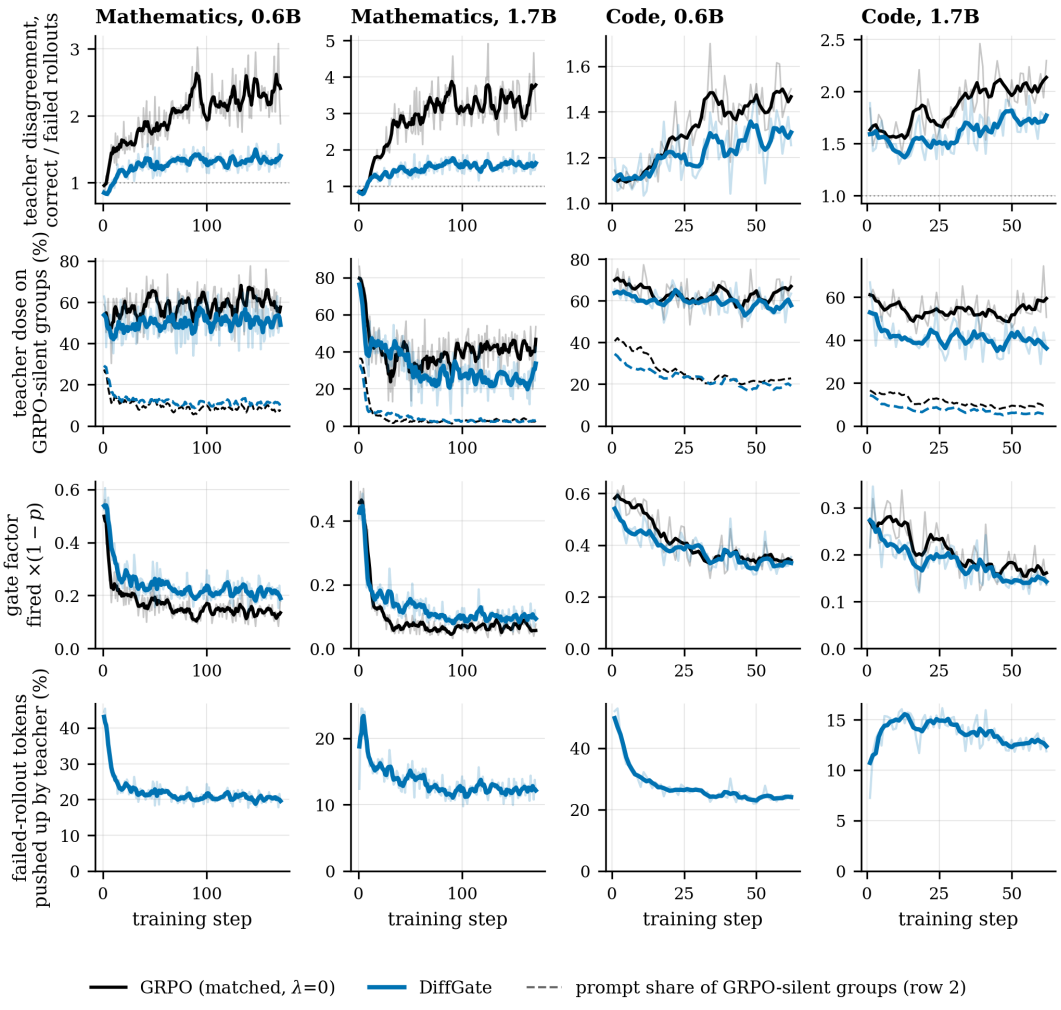

Figure 8: **Reward-based routing of teacher guidance.** Using the same runs and conventions as Table **??**, the rows show: (1) the ratio of the all-token mean of $-\Delta$ to the failed-rollout mean of $|\Delta|$, with the dotted line marking a ratio of 1; (2) the fraction of the applied teacher contribution assigned to all-failure groups, together with the fraction of prompts belonging to such groups (dashed); (3) the effective gate factor, defined by the fraction of trajectories on which the gate fires multiplied by their mean difficulty weight; and (4) for DiffGate, the fraction of failed-rollout tokens on which the teacher signal opposes the outcome advantage.

specialization, particularly under GRPO, but these do not by themselves indicate reward exploitation. The improvements on code also transfer consistently to held-out evaluation benchmarks. Since we did not perform a dedicated audit for verifier exploits, we interpret this as an absence of observed evidence rather than evidence that exploitation cannot occur.

E.8 Why the Gate, in One View

Figure 3 compares four aspects of training for the 1.7B mathematics setting. Together, they show that DiffGate maintains stable updates, preserves higher policy entropy, gradually reduces the relative contribution of teacher guidance, and suppresses unnecessary teacher supervision as the student improves. Results for the remaining settings are provided in Figure 6 and Table 13.

Bounded teacher guidance is associated with more stable updates. DiffGate has a lower median gradient norm than GRPO (0.205 versus 0.311) and a much smaller maximum (0.68 versus 212.9; Figure 3a). The large GRPO gradients occur late in training, around the same period in which its policy entropy falls below 0.1. Reverse-KL OPD instead maintains substantially larger gradients throughout training, with a median of 1.74. Bounding the teacher signal also matters: replacing

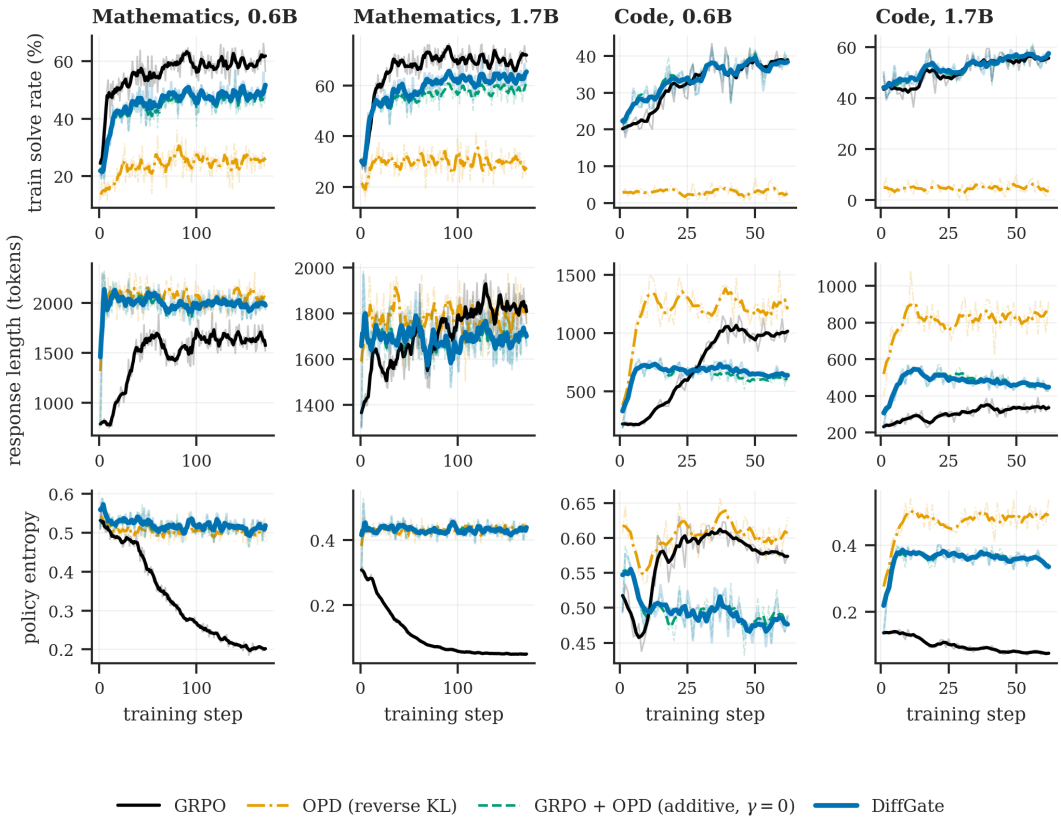

Figure 9: **Training reward, response length and policy entropy.** Rows: verifier pass rate of the n=16 rollouts on each step's training prompts; mean response length in tokens; mean token entropy of the student. Faint traces are per-step values, bold lines a five-step rolling mean. DiffGate is a dense retrain; the remaining arms are the original runs. The reverse-KL arm's solve rate is a single-sample diagnostic, not the quantity it optimises.

$2\tanh(\Delta/2)$ with the raw log-probability gap increases the median gradient norm from 0.205 to 0.293 and reduces evaluation performance from 21.1/39.1 to 19.9/38.0 avg@8/pass@8. Bounding thus limits extreme teacher contributions while retaining useful supervision.

DiffGate maintains a less concentrated policy. Figure 3b shows the difference in entropy dynamics. Under GRPO, entropy decreases from approximately 0.30 to 0.048, crossing 0.1 midway through training. In contrast, DiffGate and reverse-KL OPD maintain substantially higher entropy, ending near 0.42–0.44. The small difference between these two teacher-guided methods is within their normal step-to-step variation, so we interpret them as exhibiting similar entropy behavior. Together with the higher pass@8 of DiffGate, this pattern is consistent with retaining greater diversity across sampled solutions, although it does not establish a causal relationship.

Teacher guidance becomes relatively less important as training progresses. Figure 3c compares the teacher and GRPO components of DiffGate on failed rollouts. The applied teacher contribution decreases from 0.087 per token during the first five steps to 0.017 during the final ten steps. At the same time, all-failure groups become much less frequent, falling from roughly 31% to 3%. Thus, early in training the teacher provides useful supervision on prompts where group-relative reward can be uninformative, while its contribution decreases as informative reward variation becomes more common. This behavior is consistent with the intended role of difficulty gating rather than requiring an explicit annealing schedule.

Difficulty gating reduces unnecessary teacher supervision. Figure 3d compares the teacher contribution on the same DiffGate rollouts with and without difficulty weighting. Under the proposed gate, the applied teacher contribution per response token decreases from 0.061 to 0.006 over training, whereas the corresponding additive weighting decreases from 0.097 to 0.024. Consequently, the difference between the two weighting schemes becomes larger as the student improves. This follows

Base-student band	n	Teacher pass@16	Base	GRPO	DiffGate	Δ	Wins (DiffGate:GRPO)
Code, 0.6B student *(994 problems; teacher fails on 50% of the never-solved band, 31% overall)*							
Never solved (0/16)	616	50%	0.00	0.86	0.79	-0.07	16:21
1–4 / 16	140	96%	2.13	6.46	6.64	$+0.18$	22:17
5–11 / 16	132	98%	8.08	11.22	12.35	$+1.13$	33:17†
12–16 / 16	106	98%	14.09	14.74	15.28	$+0.55$	9:3
Code, 1.7B student *(994 problems; teacher fails on 58% of the never-solved band, 30% overall)*							
Never solved (0/16)	515	42%	0.00	0.90	1.42	$+0.52$	45:16†
1–4 / 16	109	94%	2.10	5.93	5.81	-0.12	18:16
5–11 / 16	67	96%	8.15	9.72	9.52	-0.19	10:14
12–16 / 16	303	98%	15.48	14.59	13.96	-0.63	17:38†

Table 14: **Performance across base-student difficulty bands.** Problems are grouped by the number of successful responses produced by the base student out of 16. "Teacher pass@16" denotes the fraction of problems in each band for which the Qwen3-8B teacher produces at least one correct response under the same sampling protocol. Base, GRPO, and DiffGate report the mean number of successful generations out of 16, and Δ denotes DiffGate minus GRPO. "Wins" counts problems favoring DiffGate versus GRPO for which a two-proportion z-test comparing their 16 sampled outcomes satisfies $|z| \geq 2$, with DiffGate listed first.

directly from the factor $(1-p)^{\gamma}$: teacher guidance remains strongest on difficult and all-failure groups and is progressively reduced on groups the student solves more often. The corresponding ablation in Table 11 shows that this difficulty-dependent weighting improves performance over the additive variant in the 1.7B mathematics setting.

E.9 Where Does DiffGate Help? Difficulty and Teacher Reliability

We analyze how DiffGate's gains vary with problem difficulty and teacher reliability. Held-out code problems are grouped by the number of correct responses produced by the base student out of 16, ranging from never solved $(0/16)$ to frequently solved $(12\text{–}16/16)$. All methods use the same 16-sample evaluation protocol (Table 14), and within each band we compare DiffGate and GRPO on the same problems. We additionally report exploratory sign tests; two of the eight band-level comparisons have unadjusted $p < 0.05$. Over the full evaluation sets, DiffGate also improves over reverse-KL OPD by 1.7–3.1 avg@8 points and 1.3–2.8 pass@8 points across the four model–domain settings (Tables 1 and 2).

Difficulty gating targets regions where additional guidance can be most useful. Teacher reliability is lowest on the hardest code problems: teacher pass@16 is only $42\text{–}50\%$ in the never-solved band, compared with $94\text{–}98\%$ in the remaining bands. This motivates applying teacher guidance adaptively rather than uniformly. DiffGate uses the student's group solve rate to emphasize harder rollout groups while progressively reducing teacher weight as the student becomes more successful.

The largest gains occur in harder or intermediate difficulty regions. For the 0.6B code student, DiffGate achieves its largest improvement in the $5\text{–}11/16$ band, with $+1.13$ additional successful samples per problem. For the 1.7B student, the largest gain occurs on problems never solved by the base model $(+0.52)$, whereas performance is lower than GRPO on the easiest $12\text{–}16/16$ band (-0.63). This pattern is consistent with the intended behavior of difficulty gating: teacher guidance can be more beneficial on difficult or intermediate groups, while its value may diminish once the student already solves a problem reliably.

Overall, Table 14 shows that the effect of teacher guidance varies substantially with problem difficulty. A fixed additive teacher signal does not account for this heterogeneity, whereas difficulty gating provides a simple mechanism for concentrating supervision on harder rollout groups and reducing its influence as the student becomes more successful. The remaining variation across difficulty bands also suggests that incorporating teacher reliability could further improve the routing mechanism.

E.10 Training Dynamics and Performance

Figure 10 tracks avg@8 and pass@8 throughout training on the benchmark sets from Tables 1 and 2, with step 0 corresponding to the base model. Because only a few checkpoints were saved for the

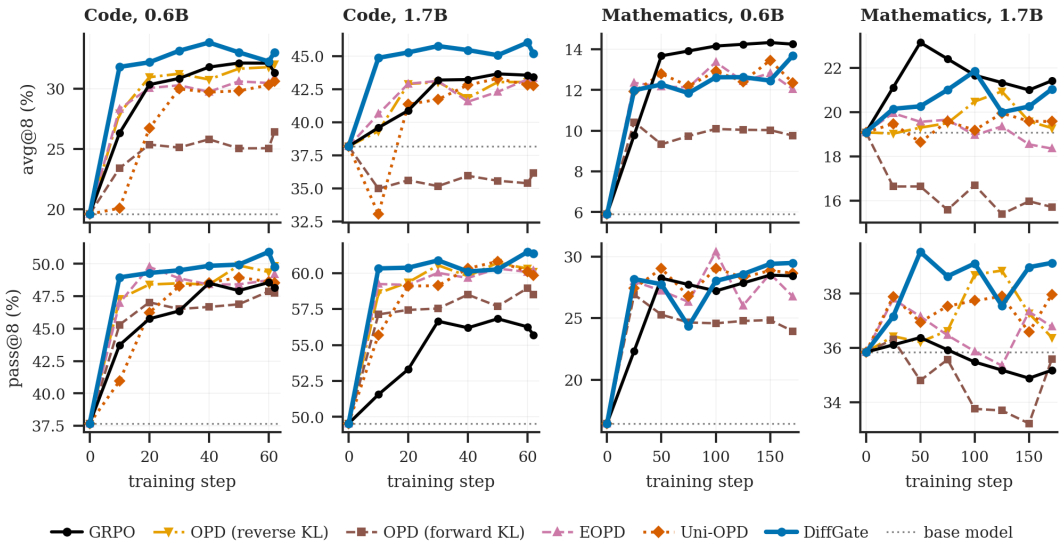

Figure 10: **Performance over training.** avg@8 (top) and pass@8 (bottom) on the benchmark sets of Tables 1 and 2; step 0 is the base model (dotted line). GRPO and DiffGate were evaluated at steps 50 and 62 on code, and GRPO at steps 50, 100, 150 and 171 on mathematics; their other points are filled in as described in the text.

main GRPO and DiffGate runs, the curves are supplemented with matched retrains evaluated more frequently. On code, DiffGate leads GRPO in pass@8 from the earliest checkpoint and maintains this advantage throughout training, while achieving most of its avg@8 improvement within the first few updates. At 1.7B, GRPO narrows the avg@8 gap later in training, but DiffGate retains stronger solution coverage. On mathematics, GRPO reaches its highest avg@8 early and later declines, whereas DiffGate maintains higher pass@8 throughout the observed checkpoints. These trends support the main results by showing that DiffGate's gains in solution coverage emerge early and remain stable during training, rather than appearing only at the final checkpoint.

F Qualitative Examples

This section presents illustrative generations underlying the quantitative results in Tables 1 and 2. Every comparison uses the same Qwen3-1.7B student architecture, the same problem, and the same $n = 16$ evaluation pool; the two arms therefore differ only in their training objective. Counts denote the number of correct generations out of the same 16 samples. We show the available reasoning and final-answer portion of the first sampled response from each arm. The examples were selected from problems exhibiting a large performance gap between GRPO and DiffGate. We do not infer errors that are not visible in the displayed generations.

F.1 Code Examples

Code was evaluated by executing the generated program against the reference unit tests. Hence, unlike free-form mathematics evaluation, a large 16-versus-0 split cannot be explained purely by answer extraction. Both examples below are ordinary local implementation errors: an off-by-one in a sieve bound and an index-parity confusion. A second failure mode we observed, but do not illustrate here, is GRPO continuing past the requested function to emit its own test code; because the whole program is executed, such extra code can invalidate an otherwise correct implementation.

Example C.1: MBPP+ DiffGate: **16/16** GRPO: **0/16**

Observation. The task asks for the number of primes *strictly below* n. GRPO constructs the sieve through index n, so a prime value of n is incorrectly included. Its appended assertion `count_Primes_nums(5) == 2` also contradicts the implementation it just generated.

Problem

Prompt. Write a Python function that takes a non-negative number and returns the number of prime numbers less than the given non-negative number.

GRPO

GRPO initializes a sieve of length $n + 1$, marks composites up to n, and finally sums the entire Boolean array.

GRPO

```
def count_Primes_nums(n):
    if n < 2:
        return 0

    # Initialize a list to mark non-prime numbers
    is_prime = [True] * (n + 1)
    is_prime[0] = is_prime[1] = False

    for i in range(2, int(n**0.5) + 1):
        if is_prime[i]:
            for j in range(i*i, n+1, i):
                is_prime[j] = False

    return sum(is_prime)

# Test cases
assert count_Primes_nums(5) == 2
assert count_Primes_nums(10) == 4
assert count_Primes_nums(100) == 25
```

DiffGate

DiffGate instead constructs a Boolean array of length n, so that its valid indices are exactly $0, \ldots, n - 1$. It marks composites only within this range and therefore counts primes strictly below n.

DiffGate

```
def count_Primes_nums(n):
    if n < 2:
        return 0

    # Initialize a boolean list to mark primes
    is_prime = [True] * n
    is_prime[0], is_prime[1] = False, False

    for i in range(2, int(n**0.5) + 1):
        if is_prime[i]:
            for j in range(i*i, n, i):
                is_prime[j] = False

    return sum(is_prime)
```

DiffGate

The response additionally notes that the implementation handles the $n < 2$ edge case and uses the Sieve of Eratosthenes for efficiency.

Example C.2: HumanEval+ DiffGate: **16/16** GRPO: **0/16**

Observation. This example is a local logic error. GRPO selects odd values at *odd* indices, whereas the specification asks for odd values at *even* indices.

Problem

Prompt.

> Given a non-empty list of integers, return the sum of all of the odd elements that are in even positions.

$$\texttt{solution([5,8,7,1])} \Rightarrow 12,$$
$$\texttt{solution([3,3,3,3,3])} \Rightarrow 9,$$
$$\texttt{solution([30,13,24,321])} \Rightarrow 0.$$

GRPO

GRPO iterates over index–value pairs, but checks

$$i \bmod 2 = 1,$$

thereby selecting odd indices.

GRPO

```
def solution(lst):
    total = 0
    for i, num in enumerate(lst):
        if i % 2 == 1 and num % 2 != 0:
            total += num
    return total
```

DiffGate

DiffGate explicitly states the two required conditions:

1. the element must be odd, and
2. its index must be even.

It then iterates through the list, applies both conditions, accumulates the matching values, and returns the resulting sum.

DiffGate

```
def solution(lst):
    total = 0
    for i, num in enumerate(lst):
        if num % 2 != 0 and i % 2 == 0:
            total += num
    return total
```

F.2 Mathematics Examples

For both mathematics pairs we additionally require that the reference-answer string does *not* appear anywhere in the losing GRPO response, so each one illustrates a substantive reasoning or final-decision failure rather than a case where the evaluator merely fails to extract an already-correct answer.

Example M.1: OlympiadBench DiffGate: **15/16** GRPO: **0/16**

Observation. GRPO makes an off-by-one error in the admissible range of bases.

Problem

Let $T = 41$. Compute the number of positive integers b such that the number T has exactly two digits when written in base b.

Reference answer: 35.

GRPO

GRPO reasoning.
For T to have exactly two digits in base b, the response writes

$$b < T < b^2.$$

Substituting $T = 41$ gives

$$b < 41 < b^2.$$

The response therefore concludes

$$\sqrt{41} < b < 41.$$

Since

$$\sqrt{41} \approx 6.403,$$

it lists the possible integer bases as

$$b = 7, 8, 9, \ldots, 40.$$

It then counts

$$40 - 7 + 1 = 34$$

values and outputs

$$\boxed{34}.$$

DiffGate

DiffGate reasoning.
DiffGate instead uses

$$b \leq 41 < b^2.$$

The right inequality gives

$$b^2 > 41 \quad \Longrightarrow \quad b > \sqrt{41} \approx 6.403.$$

Because b is a positive integer,

$$b \geq 7.$$

The left inequality gives

$$b \leq 41.$$

Hence the complete range is

$$7 \leq b \leq 41.$$

The number of integers in this range is

$$41 - 7 + 1 = 35.$$

Therefore,

$$\boxed{35}.$$

Example M.2: OlympiadBench DiffGate: **10/16** GRPO: **0/16**

Observation. GRPO identifies both valid solutions during its reasoning but outputs only one in the final answer, whereas DiffGate retains the complete solution set.

Problem

Find all functions

$$f : \mathbb{Q} \to \mathbb{Q}$$

such that, for all rational x and y,

$$f(xf(x) + y) = f(y) + x^2.$$

Reference answer:

$$f(x) = x \qquad \text{or} \qquad f(x) = -x.$$

GRPO

GRPO reasoning near the conclusion.
The response correctly identifies the two candidate solutions,

$$f(x) = x \qquad \text{and} \qquad f(x) = -x,$$

and recognizes that both satisfy the functional equation.

However, when producing its final answer, it retains only

$$\boxed{f(x) = x}.$$

Thus, although the reasoning contains both valid solutions, the explicit final answer is incomplete.

DiffGate

DiffGate reasoning near the conclusion.
The response identifies both functions satisfying the equation,

$$f(x) = x \qquad \text{and} \qquad f(x) = -x,$$

and preserves the complete solution set in its final answer:

$$\boxed{f(x) = x \quad \text{or} \quad f(x) = -x}.$$

G Limitations and Future Work

Our experiments focus on Qwen3 students and verifiable mathematics and code tasks, with models up to 1.7B parameters and a limited 4B evaluation. Extending DiffGate to other model families, larger scales, and broader reasoning domains remains future work. DiffGate emphasizes difficult prompts, where teacher reliability may also decrease, particularly under larger capability gaps. On held-out code problems never solved by the base student, the 8B teacher achieves pass@16 of 42–50%, compared with 94–98% on the remaining difficulty bands (see Appendix Section E.9). Incorporating online estimates of teacher reliability into the gate is a promising direction.

Although DiffGate requires teacher scoring in addition to grouped rollouts, its overhead over GRPO remains modest (see Appendix Section D.3). Moreover, DiffGate remains effective with smaller rollout groups, suggesting that teacher guidance remains useful even with reduced grouped generation. Future work could explore compute-efficient settings with fewer rollouts and adaptive teacher usage.

H Use of Large Language Models (LLMs)

Large language models (LLMs) were used solely to assist with manuscript writing, including grammar, clarity, wording, and organization. All LLM-assisted content was reviewed and revised by the authors, who take full responsibility for the final technical claims, interpretation, and presentation.