COMPRESS TO IMPROVE: RECLAIMING OUTPUT BUDGET FROM ON-POLICY DISTILLATION FOR STRONGER REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) teaches a student model to reason through teacher feedback on its own generated responses. We find that this process also produces substantial length inflation: across six OPD-family methods evaluated from a common base model on competition mathematics, responses are consistently longer than those produced by supervised fine-tuning (SFT) and reinforcement learning (RL). Under a finite output budget, longer initial solutions leave less room for checking, revision, and further exploration. We introduce concise post-training as an intermediate stage that favors shorter correct solutions and frees output space for further reasoning. Across OPD, ExOPD, and OPSD, it reduces mean output length by 29.6–36.6% while improving both sample accuracy and problem coverage. We then reuse the released budget through prompted continuation, which asks the compressed model to check and revise its answers with fixed parameters, or renewed OPD, which resumes teacher-guided learning. Continuation improves sample accuracy in every route, while renewed distillation increases problem coverage. The strongest final endpoint, ExOPD followed by compression and continuation, raises average@8 from 25.2% to 27.3% and pass@8 from 36.7% to 40.8% relative to the original ExOPD checkpoint. Both complete paths outperform their original distilled models without exceeding their mean output lengths, supporting compression as an intermediate stage for further reasoning and learning gains.

1 INTRODUCTION

A reasoning model can improve its answers by continuing to think with its parameters fixed. For example, Muennighoff et al. (2025) use a simple "Wait" intervention to extend a model's reasoning and improve its mathematical accuracy. This opportunity depends on having room to continue. In ordinary generation that retains the full reasoning history, the input and generated tokens must fit within a finite context window, and an output-token limit can further restrict generation (Muennighoff et al., 2025; Metel et al., 2026). A longer initial solution leaves fewer tokens for checking an approach, revising an error, or exploring another line of reasoning. Learning to reach an answer more concisely can therefore create space for further improvement.

This consideration is especially relevant to OPD, which transfers teacher capabilities by providing feedback along prefixes generated by the student (Agarwal et al., 2024; Gu et al., 2024). We observe substantial output growth across six distillation methods (Agarwal et al., 2024; Yang et al., 2026a; Zhao et al., 2026; Zhu et al., 2026; Jang et al., 2026; Ko et al., 2025), all initialized from the same Qwen3-1.7B base and evaluated on the same mathematics problems. Every distilled endpoint generates longer responses than SFT and all five evaluated RL methods (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025a; Xi et al., 2026; Shrivastava et al., 2026). The original OPD and ExOPD training trajectories also show diminishing late-stage gains. We ask whether concise post-training can free output space for further reasoning and learning (Section 3.1).

We apply concise post-training to an already distilled model, favoring shorter correct solutions before further inference or distillation. Each route retains its original OPD recipe and teacher feedback. Teacher guidance specifies how to continue at a given prefix; an explicit preference among correct solutions can additionally favor those that reach an answer with fewer tokens. DPO tests

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.

whether fixed rollouts suffice for compression. Online training lets us compare correctness-only feedback with a score-conditioned reward that favors shorter correct solutions more strongly when a larger share of the model's sampled responses are correct. These alternatives turn conciseness into a learning objective while retaining correctness supervision. A training-order comparison further shows why placement matters: applying these objectives after OPD yields compact policies, whereas applying them before OPD leaves final responses close to the original distilled model's length (Section 4.6).

We evaluate the compressed policy through two independent paths. The training-free path prompts every initial response to check for mistakes and continue reasoning with fixed parameters. The post-training path resumes the route's original distillation procedure from the compressed checkpoint. This second path asks whether an already-completed distillation recipe can make further progress from a policy shaped by a different intermediate objective.

Across OPD, ExOPD, and OPSD, the compressed policies improve both solved-problem coverage and sample accuracy with shorter responses. Prompted continuation improves sample accuracy in every route, and renewed distillation improves solved-problem coverage in every route. Both complete paths exceed the original distilled models on both measures while remaining below their original mean output lengths.

We make three contributions:

- **Identifying consistent length inflation and late-training saturation in OPD.** Our common-base comparison covers six distillation methods, SFT, and five RL methods. All six distilled models generate over twice the base response length, while original OPD and ExOPD runs exhibit diminishing late-stage gains, motivating an intermediate objective that frees output space and changes the starting policy for further learning.
- **Concise post-training with broader problem coverage.** We develop a concise post-training stage after distillation with score-conditioned feedback that favors shorter correct responses more strongly when a larger share of the model's sampled responses are correct. It reduces mean output length by 29.6–36.6% and improves both pass@8 and average@8 across all three routes.
- **Further inference and learning gains after compression.** The prompted-continuation and renewed-distillation endpoints both exceed the corresponding original distilled checkpoints on both accuracy metrics, at 94.9–99.8% of those checkpoints' mean output lengths. Renewed distillation also yields further gains from the compressed starting policies.

2 Related Work

On-policy distillation and output length. Generalized Knowledge Distillation trains on student-generated trajectories, while MiniLLM combines on-policy sampling with reverse-KL optimization and length-aware stabilization (Agarwal et al., 2024; Gu et al., 2024). Later methods change the divergence, teacher signal, or supervision mixture, including DistiLLM and DistiLLM-2, HPD, Veto, ExOPD, and OPSD (Ko et al., 2024; 2025; Zhu et al., 2026; Jang et al., 2026; Yang et al., 2026a; Zhao et al., 2026). Their effects on length need not agree: MiniLLM corrects a short-response bias, whereas ExOPD and recent OPD analyses report length growth or late instability in other settings (Gu et al., 2024; Yang et al., 2026a; Li et al., 2026; Wang et al., 2026b). We measure output length across six OPD-family methods under a common initialization and evaluation setting, then study concise post-training after distillation.

Controlling reasoning length. Additional reasoning can improve problem solving but can also introduce repetition, accumulated errors, and overthinking (Wei et al., 2022; Chen et al., 2025; Wu et al., 2025). Offline methods learn from shorter-correct preferences using DPO-style or likelihood-anchored objectives (Rafailov et al., 2023; Chen et al., 2025; Meng et al., 2024); online methods target requested lengths, shorter correct group samples, or correctness-aware brevity rewards (Aggarwal & Welleck, 2025; Yi et al., 2025; Arora & Zanette, 2025; Xiang et al., 2025; Song & Zheng, 2025; Fatemi et al., 2025). CRISP and Implicit Compression Regularization likewise learn from concise self-policy targets or shorter on-policy responses (Sang et al., 2026; Wang et al., 2026a). These works establish shorter-correct learning and success-aware length control. We evaluate com-

pression across OPD routes and use the resulting policies for sequential answer revision and renewed teacher-guided learning.

Reusing released budget for inference and learning. Inference budget can support parallel sampling and aggregation, as in self-consistency and ExpThink, or sequential revision, as in s1, Self-Refine, and SCoRe (Wang et al., 2023; Bian et al., 2026; Muennighoff et al., 2025; Madaan et al., 2023; Kumar et al., 2025). Sequential revision can also turn correct answers into incorrect ones (Huang et al., 2024). On the learning side, OPD is initialization-sensitive: prolonged optimization may plateau or reduce large-k coverage, while later training can recover capabilities lost earlier (Li et al., 2026; Liu et al., 2025a; Yue et al., 2025; Jin et al., 2025). We therefore separately test whether the compressed checkpoint supports fixed-policy continuation and renewed distillation from a changed starting policy.

3 Compression for Further Reasoning and Learning

Given a distilled checkpoint, we apply concise post-training to obtain a shorter-response policy π_C. The compressed checkpoint then supports two independent branches: prompted continuation with fixed parameters and renewed distillation with parameter updates (Figure 1). Section 3.1 presents the observations motivating this pipeline.

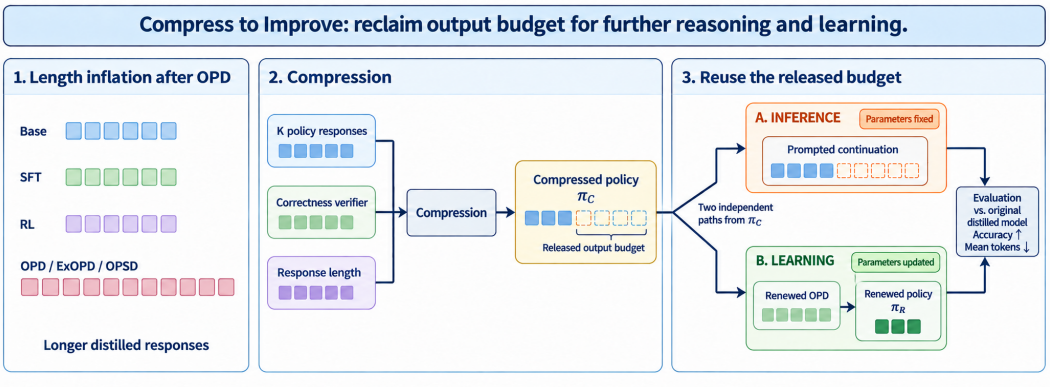

Figure 1: Method overview. OPD-family policies produce longer responses than the shared-base alternatives. Compression releases output budget, which is reused through prompted continuation with fixed parameters or renewed OPD with parameter updates. Both complete paths are compared with the original distilled checkpoint in accuracy and mean output length.

3.1 Length Inflation and Training Saturation in OPD

Common-base comparisons establish the phenomena motivating this pipeline. On the same 120 problems from AIME24, AIME25, HMMT-Feb, and HMMT-Nov (Mathematical Association of America, 2026; Harvard–MIT Mathematics Tournament, 2026), we compare SFT; five RL methods—GRPO, DAPO, ProRL, BAPO, and GFPO (Shao et al., 2024; Yu et al., 2025; Liu et al., 2025a; Xi et al., 2026; Shrivastava et al., 2026); and six distillation methods—OPD, ExOPD, OPSD, HPD, Veto, and DistiLLM-2 (Agarwal et al., 2024; Yang et al., 2026a; Zhao et al., 2026; Zhu et al., 2026; Jang et al., 2026; Ko et al., 2025). All methods start from Qwen3-1.7B, with lengths measured in our reproductions as mean generated tokens relative to the base model's 3,938 tokens.

Figure 2 shows a clear separation: the six distilled endpoints generate 2.05–$2.39\times$ the base output length, and even the shortest is longer than every evaluated RL endpoint. ExOPD, the shortest distilled endpoint at 8,057 tokens, still exceeds the longest RL endpoint, DAPO at 6,435 tokens, by 1,622 tokens (25.2%). SFT reaches $1.05\times$ the base length and the RL methods 1.29–$1.63\times$, so the length inflation is consistent across the evaluated distillation family rather than being specific to one OPD recipe. Appendix A reports the complete numerical comparison.

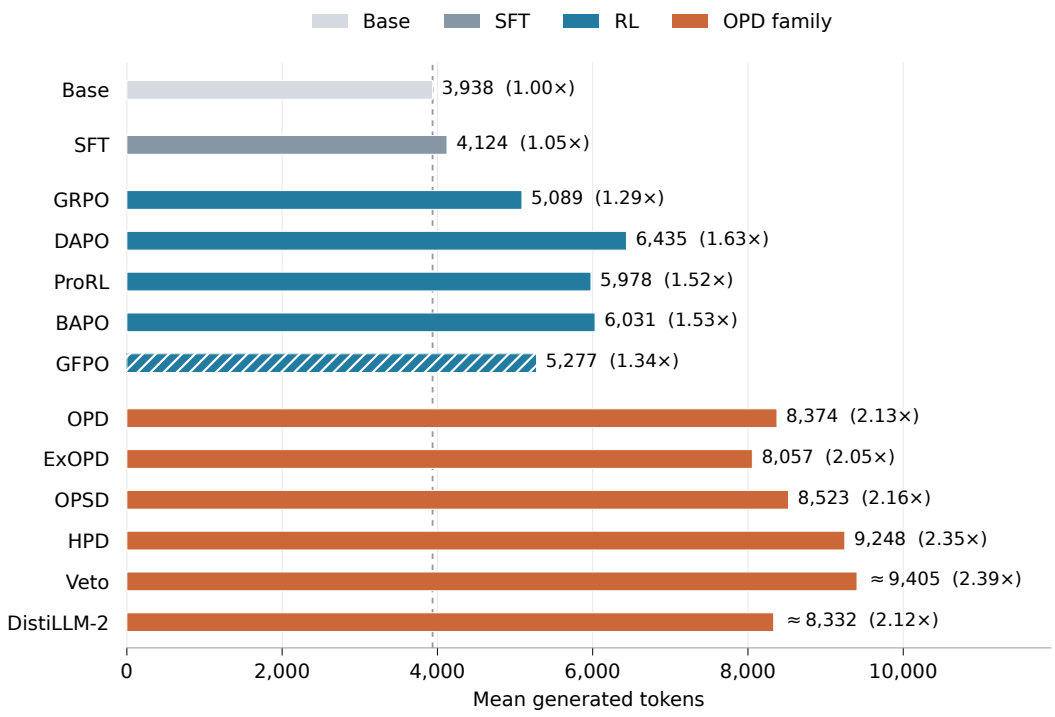

Figure 2: Mean output length on the common mathematics suite. Labels show tokens and ratios to the Qwen3-1.7B base; hatched GFPO marks the length-aware RL baseline.

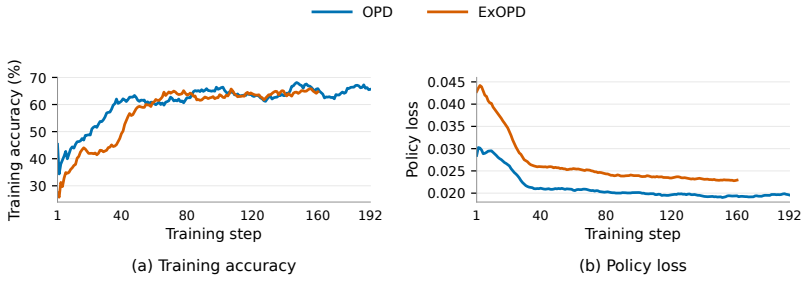

Figure 3: Training dynamics from the Qwen3-1.7B base during the original OPD and ExOPD runs, shown as causal 20-step trailing means. Accuracy stabilizes at approximately 40 and 60 steps, respectively, while policy loss changes slowly thereafter.

Figure 3 reveals a second pattern in the original, base-started training trajectories. Training accuracy largely stabilizes after approximately 40 steps for OPD and 60 steps for ExOPD, while policy loss changes only gradually thereafter. These curves provide evidence of training-side saturation in the two original runs. Related work further shows that prolonged OPD-family optimization can move from saturation into degradation: unclipped OPSD exhibits performance collapse, while length inflation during OPD can induce truncation collapse and sharp validation degradation (Zhao et al., 2026; Luo et al., 2026).

Teacher matching along student-generated prefixes does not explicitly prefer a shorter correct solution over a longer one.[1] Together, the longer endpoints and diminishing late-training gains motivate changing the objective rather than simply extending the same base-started trajectory. We therefore

[1]Recent concurrent work identifies EOS-token mismatch as one source of OPD length inflation (Yang et al., 2026b); complementarily, we learn concise policies after distillation and reuse the saved output budget.

add a separate compression stage after distillation to favor shorter correct responses before reusing the released output space.

3.2 Concise Post-Training

Let π_{D} be a distilled checkpoint and π_{C} its compressed policy. We compare three alternatives with distinct roles. DPO tests whether preferences over fixed rollouts suffice to learn shorter, accurate responses. Correctness-only GRPO tests whether online correctness learning alone produces such behavior. Score-conditioned GRPO adds explicit length feedback to test whether favoring concise correct responses improves the accuracy–length trade-off. The two online alternatives, defined below as RL1 and RL2, share their optimization settings; each alternative starts independently from π_{D} and uses it as a frozen reference π_{ref}.

For a prompt x, let K sampled completions have correctness $c_i \in \{0, 1\}$ and generated length $L_i \geq 1$. Define the correct set $\mathcal{C} = \{i : c_i = 1\}$, its size $m = |\mathcal{C}|$, and the group success rate $p = m/K$. Compression changes the policy without lowering the generation cap.

DPO. DPO combines correct–wrong and short–long preferences from frozen rollouts with a correct-response likelihood anchor (Rafailov et al., 2023). Appendix B.1 specifies the preference construction and objective, and Appendix B.2 gives the fixed initialization calibration.

Correctness-only GRPO (RL1). RL1 learns only from final-answer correctness and contains no length signal (Shao et al., 2024). It therefore isolates how much compression arises from online correctness learning alone; its reward and advantages are given in Appendix B.3.

Score-conditioned compression with GRPO (RL2). RL2 adds length feedback within the correct set, with stronger feedback when a problem is solved more reliably. Assign each correct response a centered efficiency score e_i that is positive for relatively shorter correct solutions and set $e_i = 0$ for incorrect responses. The reward is

$$\lambda(p) = \lambda_{\max} p^{\gamma}, \qquad r_i^{\mathrm{conc}} = c_i\big[1 + \lambda(p) e_i\big], \qquad \gamma > 0. \tag{1}$$

RL2 shifts reward from longer correct responses to shorter ones. The shift grows with the sampled success rate: $\lambda_{\max}$ bounds its scale, and γ controls how quickly it increases. Centering the efficiency scores preserves the group's mean reward at p, so the advantage contains a correctness term and a length term. When few sampled responses are correct, p^{γ} weakens the length preference so that the objective emphasizes learning to answer correctly. When all responses are correct, the correctness advantages cancel, while differences in correct-response length can still guide learning toward shorter solutions. Appendices B.3 and B.5 provide the score definition, decomposition, and bounds.

Both online alternatives use the same clipped policy surrogate and frozen-reference KL regularizer; implementation details and settings appear in Appendices B.4 and C.1. The evaluated RL2 uses $\lambda_{\max} = 0.10$ and $\gamma = 2$.

3.3 Further Reasoning and Renewed Learning

Prompted continuation and renewed distillation start independently from the same compressed checkpoint. We compare the final accuracy and output length of each branch with the corresponding original distilled checkpoint.

Prompted continuation. Following the idea of extending reasoning through a continuation cue (Muennighoff et al., 2025), we append `but wait, let me check if there is a mistake` to every initial response and let the model continue with fixed parameters. We score the final answer and count tokens from both the initial response and the continuation.

Renewed on-policy distillation. We resume each route's original OPD, ExOPD, or OPSD procedure from its compressed checkpoint, independently of continuation. This tests whether a completed distillation recipe yields further gains from a policy shaped by a different intermediate objective. The renewed policy is evaluated against both its compressed starting point and the original distilled checkpoint; training and checkpoint-selection settings appear in Section 4.4 and Appendix C.3.

Comparing accuracy at the original output budget. Let $\overline{L}(\pi)$ denote mean generated length. Each route's original distilled checkpoint π_{D} defines its reference budget $\overline{L}(\pi_{\mathrm{D}})$. For endpoint π_E, we report $q_E = \overline{L}(\pi_E)/\overline{L}(\pi_{\mathrm{D}})$ and the accuracy change $Q(\pi_E) - Q(\pi_{\mathrm{D}})$. A positive change at $q_E \leq 1$ indicates improvement without exceeding this budget. We report gains over the compressed checkpoint separately. All comparisons use the same problems and sample count. Renewed distillation is evaluated after additional training, whose cost is outside this output-length comparison.

4 Experiments

We evaluate whether concise post-training frees output space while improving solved-problem coverage, compare compression objectives and training orders, and test whether further reasoning and renewed distillation turn the shorter policies into stronger final answers.

4.1 Experimental Setup

Models and data. To compare post-training strategies from a common starting point, we evaluate the Qwen3-1.7B base, SFT, GRPO, three distilled checkpoints (OPD, ExOPD, and OPSD), and each distilled checkpoint after compression. The GRPO baseline is trained from the base; RL1 in the compression ablation instead starts from a distilled checkpoint. The distilled and compressed models use thinking mode disabled during training and inference. We measure compression relative to each route's own original distilled checkpoint because teacher setups differ. The DeepMath training pool contains 57,130 problems, not all consumed by every run (Yang et al., 2026a). A held-out set of 500 relatively difficult DeepMath problems is used for validation and checkpoint selection. The test suite contains 120 problems from AIME24, AIME25, HMMT-Feb, and HMMT-Nov (Mathematical Association of America, 2026; Harvard–MIT Mathematics Tournament, 2026).

Training and metrics. In the compression ablation, DPO uses 50 training steps, while RL1 and RL2 share a 120-step online budget; the training-order experiments use the route-specific settings reported in Appendix C.4. Additional optimization and checkpoint-selection details are provided in Appendix C.1. Standard evaluation uses a maximum generation length of 16K tokens. We sample eight responses per problem. Average@8 is the percentage of correct responses among all 960 samples, and pass@8 is the percentage of the 120 problems solved at least once. We report percentages to one decimal place and differences in percentage points (pp). Length is the mean generated token count over all responses, including incorrect ones. These metrics distinguish how often the model answers correctly, how many problems it can solve, and how much output it uses.

4.2 Main Results: Compressed Checkpoints and Final Endpoints

To evaluate compression and the subsequent use of the released output space, Table 1 compares the complete paths with SFT and GRPO from the shared base. The main experiments use score-conditioned GRPO (RL2) for compression; Section 4.5 compares it with DPO and correctness-only GRPO. Each compressed checkpoint supports two independent endpoints: prompted continuation with fixed parameters and renewed distillation with parameter updates.

Every compressed checkpoint and all six final endpoints exceed SFT and GRPO on both accuracy metrics. Across routes, compression improves both metrics while reducing mean length by 29.6–36.6%; after budget reuse, all endpoints remain within 94.9–99.8% of their original distilled checkpoints' mean lengths and outperform those checkpoints on both metrics.

Gains from compression and budget reuse. ExOPD illustrates both stages' contributions. Compression raises average@8 from 25.2% to 26.7% and pass@8 from 36.7% to 39.2%, while shortening responses from 8,057 to 5,670 tokens. Prompted continuation achieves the strongest Qwen3-1.7B endpoint: 27.3% average@8 and 40.8% pass@8 at 7,946 tokens, or 98.6% of the original ExOPD mean length. Using these reported means, compression releases 2,387 tokens and continuation uses 2,276 of them, leaving the complete path 111 tokens below the original endpoint. Compression provides an immediate accuracy gain and output space for a further gain during inference.

Recipe	Average@8 (↑)	Pass@8 (↑)	Mean tokens (↓)
Base model	7.8	16.7	3,938
SFT	12.9	15.0	4,124
GRPO	14.1	23.3	5,089
OPD	22.5	30.8	8,374
OPD → compression	23.0 (+0.5)	33.3 (+2.5)	**5,307** (−36.6%)
OPD → compression → continuation	**24.5** (+2.0)	32.5 (+1.7)	8,129 (−2.9%)
OPD → compression → renewed OPD	23.3 (+0.8)	**34.2** (+3.3)	7,946 (−5.1%)
OPSD	21.7	30.0	8,523
OPSD → compression	22.7 (+1.0)	31.7 (+1.7)	**5,408** (−36.5%)
OPSD → compression → continuation	23.2 (+1.6)	33.3 (+3.3)	8,509 (−0.2%)
OPSD → compression → renewed OPSD	**24.0** (+2.3)	**35.0** (+5.0)	8,320 (−2.4%)
ExOPD	25.2	36.7	8,057
ExOPD → compression	26.7 (+1.5)	39.2 (+2.5)	**5,670** (−29.6%)
ExOPD → compression → continuation	**27.3** (+2.1)	**40.8** (+4.2)	7,946 (−1.4%)
ExOPD → compression → renewed ExOPD	26.7 (+1.5)	40.0 (+3.3)	7,729 (−4.1%)

Table 1: Shared-base baselines and complete training recipes through compression. Parentheses report changes relative to the original distilled model in each route: percentage points for Average@8 and Pass@8, and percent for mean tokens. Continuation and renewed distillation are independent branches from the corresponding compressed checkpoint; continuation tokens include both initial and appended generation. Bold marks the best result within each route for every column, including the shortest output.

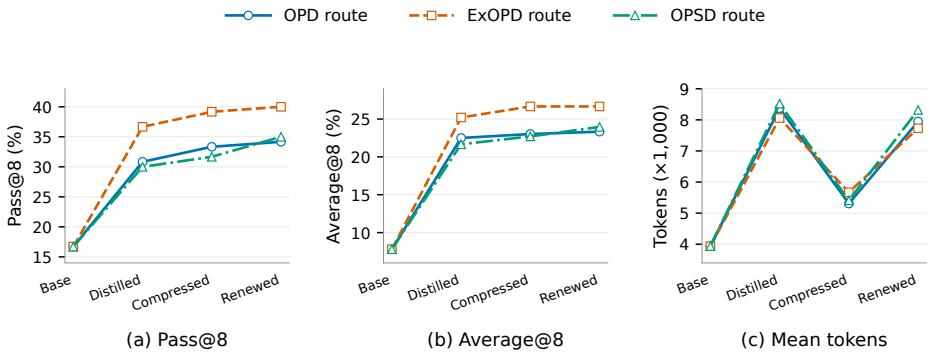

Figure 4: Compression and renewed learning across three distillation routes. Compression raises both accuracy metrics while shortening responses; resuming the original distillation procedure then raises pass@8 in every route. Positions mark successive training stages. Section 4.4 describes the renewed-distillation experiment.

4.3 Prompted Continuation After Concise Post-Training

To test whether compressed models benefit from additional reasoning, we compare their initial and continued answers in Table 1, using the protocol in Section 3.3. Continuation raises average@8 in all three routes, and every continued endpoint exceeds its original distilled checkpoint on both accuracy metrics within the original mean output budget.

Relative to the compressed ExOPD checkpoint, continuation increases average@8 from 26.7% to 27.3% and pass@8 from 39.2% to 40.8%. The higher sample accuracy and broader coverage are obtained with fixed model parameters and the same eight responses per problem. The released output space therefore supports a useful additional pass over the solution, with gains beyond those already obtained during compression.

4.4 Renewed Distillation from Compressed Checkpoints

We test whether the original distillation procedures yield further gains when initialized from compressed checkpoints. Figure 3 shows diminishing late-stage training-accuracy gains in the original OPD and ExOPD runs. We resume each route's procedure from its compressed checkpoint, independently of continuation, and select checkpoints at approximately 160 steps after validation performance stabilizes. Appendix C.3 gives the training settings.

Table 1 shows that renewed distillation increases pass@8 in all three routes; average@8 rises for OPD and OPSD and is maintained for ExOPD. Every renewed endpoint exceeds its original distilled checkpoint on both metrics while remaining shorter.

ExOPD makes the coverage gain particularly clear: pass@8 rises from 39.2% to 40.0% while average@8 remains at 26.7%. The renewed stage therefore increases the number of problems solved within eight samples while preserving pooled sample accuracy. Together, the three routes show that compressed policies support further learning under their original teacher-guided procedures as responses expand again.

4.5 Ablating the Compression Method

Recipe	Average@8 (↑)	Pass@8 (↑)	Mean tokens (↓)
OPD	22.5	30.8	8,374
OPD → DPO	**23.0** (+0.5)	31.7 (+0.8)	5,563 (−33.6%)
OPD → correctness-only GRPO (RL1)	Collapsed under the evaluated configuration		
OPD → score-conditioned GRPO (RL2)	**23.0** (+0.5)	**33.3** (+2.5)	**5,307** (−36.6%)
OPSD	21.7	30.0	8,523
OPSD → DPO	22.0 (+0.3)	30.8 (+0.8)	**5,324** (−37.5%)
OPSD → correctness-only GRPO (RL1)	Collapsed under the evaluated configuration		
OPSD → score-conditioned GRPO (RL2)	**22.7** (+1.0)	**31.7** (+1.7)	5,408 (−36.5%)
ExOPD	25.2	36.7	8,057
ExOPD → DPO	25.7 (+0.5)	38.3 (+1.7)	**5,030** (−37.6%)
ExOPD → correctness-only GRPO (RL1)	25.4 (+0.2)	38.3 (+1.7)	5,469 (−32.1%)
ExOPD → score-conditioned GRPO (RL2)	**26.7** (+1.5)	**39.2** (+2.5)	5,670 (−29.6%)

Table 2: Compression-objective ablation. Parentheses show changes from the original checkpoint (percentage points for accuracy; percent for tokens); bold marks the best per route and column.

To isolate the compression objective, Table 2 compares DPO, correctness-only RL1, and RL2 from the same original distilled checkpoints, with shared online settings for RL1 and RL2. DPO improves both accuracy metrics while reducing length across all routes. RL2 attains the highest pass@8 among the compression alternatives in every route and higher average@8 than DPO on ExOPD and OPSD while matching it on OPD.

The objectives favor different accuracy–length trade-offs. DPO produces the shortest responses on OPSD and ExOPD. On ExOPD, for example, DPO reaches 38.3% pass@8 with 5,030 tokens, while RL2 reaches 39.2% with 5,670 tokens and also achieves higher average@8. RL2 thus combines substantial compression with the strongest solved-problem coverage among the evaluated compression objectives, motivating its use in the main paths.

Correctness-only learning also compresses ExOPD and improves its accuracy, but RL1 collapses from OPD and OPSD under the same evaluated online settings. Here, collapse denotes explosive response-length growth or a breakdown in task performance that prevents training from continuing; Appendix C.1 records this criterion. The outcome of correctness-only learning therefore depends on the distilled starting point in this comparison. Adding the explicit length-feedback objective yields shorter responses and higher accuracy across all three starting checkpoints.

4.6 Training Order and Room for Further Reasoning

Table 3 compares applying RL2 or DPO before versus after OPD from the same Qwen3-1.7B base. The OPD → RL2 row reuses the main compression checkpoint from the 120-step RL2 budget; the

Training sequence	Average@8	Pass@8	Mean tokens	Original OPD output budget used (%)
OPD	22.5	30.8	8,374	100.0
RL2 ordering				
RL2 → OPD	23.6	30.8	8,231	98.3
OPD → RL2	23.0	33.3	5,307	63.4
DPO ordering				
DPO → OPD	20.4	28.3	7,879	94.1
OPD → DPO	23.0	31.7	5,563	66.4

Table 3: Complete training-order comparison. All paths start from Qwen3-1.7B, and accuracy metrics are percentages. The final column divides each endpoint's mean output length by the original OPD model's 8,374 tokens. Applying the compression objective after OPD preserves substantially more output space for subsequent reasoning.

pre-OPD RL2 stage uses 200 steps (Appendix C.4). RL2 → OPD and DPO → OPD use 98.3% and 94.1% of the original OPD output budget, whereas OPD → RL2 and OPD → DPO use only 63.4% and 66.4%. The post-OPD variants also attain higher pass@8; average@8 is comparable across the two RL2 orders and higher for post-OPD DPO.

For RL2, ending with OPD gives 23.6% average@8 and 30.8% pass@8, while ending with compression gives 23.0% and 33.3%, respectively. Ending with compression combines broader solved-problem coverage with roughly one third of the original mean output budget available for further inference or learning.

4.7 Additional Results on Qwen3-4B

To examine whether the accuracy gains also occur at a second model scale, we evaluate selected OPD and ExOPD paths with Qwen3-4B on the same competition-mathematics suite. Table 7 in Appendix D reports average@8 over 960 sampled responses.

Compression raises OPD's average@8 from 42.3% to 43.6% and ExOPD's from 45.2% to 46.3%. Starting from compressed ExOPD, prompted continuation reaches 47.3% and renewed distillation reaches 47.8%. These results extend the accuracy findings to Qwen3-4B: both evaluated distillation routes improve after compression, and the compressed ExOPD checkpoint supports further gains through either inference or teacher-guided learning.

5 Limitations

We evaluate all compression and budget-reuse paths on Qwen3-1.7B in competition mathematics, with selected Qwen3-4B results in Appendix D. The length comparison covers six distillation methods; compression and budget reuse cover OPD, ExOPD, and OPSD. Budget comparisons use mean generated length across all sampled responses, including correct and incorrect answers, not per-problem hard limits. Prompted continuation requires additional inference, while renewed distillation also requires training that is outside the inference output budget.

6 Conclusion

In our common-base mathematics comparison, all six evaluated OPD-family methods generate longer responses than SFT and the RL baselines, exceeding twice the base model's mean output length. The original OPD and ExOPD training trajectories additionally exhibit diminishing late-stage gains. Concise post-training shortens responses and improves accuracy across OPD, ExOPD, and OPSD. Continuing inference or distillation from the compressed checkpoints yields further gains: both complete paths outperform the original distilled models on average@8 and pass@8 at lower mean output length. These results support compression as an intermediate stage for improving reasoning within a limited output budget.

AI USE STATEMENT

OpenAI Codex assisted with literature synthesis, citation organization, manuscript structure, drafting and editing, experimental-design feedback, mathematical formalization, and interpretation of researcher-supplied results. It also edited the method overview figure and generated LaTeX tables and scripts for checking transcription, arithmetic, mathematical properties, and compilation. Researchers supplied the model-training and benchmark-evaluation results; Codex checks covered manuscript consistency. The authors are responsible for the claims, citations, and final content.

ETHICS STATEMENT

This study aims to improve mathematical reasoning within a limited output budget. Concise responses may omit steps needed to verify a solution, and a correct final answer does not establish the validity of the full reasoning trace. Performance on mathematics benchmarks alone does not establish reliability in high-stakes applications. Both additional training and inference consume computational resources. Model and data redistribution must follow the applicable licenses and usage conditions.

REPRODUCIBILITY STATEMENT

Section 3 describes concise post-training, prompted continuation, renewed distillation, and output-budget accounting. Appendix B gives the preference construction, losses, calibration, and reward derivations. Section 4.1 defines the data and evaluation metrics, and Appendix C reports the training and continuation settings. The result tables identify each distillation route and training stage, including collapsed runs.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649v3.

Pranjal Aggarwal and Sean Welleck. L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2503.04697v2.

Daman Arora and Andrea Zanette. Training Language Models to Reason Efficiently. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.04463v4.

Tingcheng Bian, Yuzhe Zhang, Jing Jin, Jinchang Luo, MingQuan Cheng, Haiwei Wang, Wenyuan Jiang, and Miaohui Wang. ExpThink: Experience-Guided Reinforcement Learning for Adaptive Chain-of-Thought Compression. *arXiv preprint arXiv:2605.07501*, 2026. URL https://arxiv.org/abs/2605.07501v2.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT Think That Much for 2+3=? On the Overthinking of Long Reasoning Models. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/chen25bx.html.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2504.05185*, 2025. URL https://arxiv.org/abs/2504.05185v3.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge Distillation of Large Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543v6.

Jinyi Han, Ying Huang, Ying Liao, Zishang Jiang, Xikun Lu, Haiquan Zhao, Xinyi Wang, Guanghao Zhou, Sihang Jiang, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. Your Models Have Thought Enough: Training Large Reasoning Models to Stop Overthinking. *arXiv preprint arXiv:2509.23392*, 2025. URL https://arxiv.org/abs/2509.23392v3.

Harvard–MIT Mathematics Tournament. Past Tournaments: Problems and Results. HMMT official archive, 2026. URL https://www.hmmt.org/www/archive/problems. Accessed 2026-09-25.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.01798v2.

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable On-Policy Distillation through Adaptive Target Reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026. doi: 10.18653/v1/2026.findings-acl.2094. URL https://aclanthology.org/2026.findings-acl.2094/.

Hangzhan Jin, Sitao Luan, Tianwei Ni, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. RL Fine-Tuning Heals OOD Forgetting in SFT. *arXiv preprint arXiv:2509.12235*, 2025. URL https://arxiv.org/abs/2509.12235v3.

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. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.07079v3.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards Streamlined Distillation for Large Language Models. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.03898v2.

Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. DistiLLM-2: A Contrastive Approach Boosts the Distillation of LLMs. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.07067v2.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training Language Models to Self-Correct via Reinforcement Learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/871ac99fdc5282d0301934d23945ebaa-Abstract-Conference.html.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking On-Policy Distillation of Large Language Models: Phenomenology, Mechanism, and Recipe. *arXiv preprint arXiv:2604.13016*, 2026. URL https://arxiv.org/abs/2604.13016v2.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged Reinforcement Learning Expands Reasoning Boundaries in Large Language Models. In *Advances in Neural Information Processing Systems*, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/1a22b912945fb7c0bdd079e792b31b6f-Abstract-Conference.html.

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. In *Conference on Language Modeling*, 2025b. URL https://arxiv.org/abs/2503.20783v2.

Feng Luo, Yu-Neng Chuang, Guanchu Wang, Zicheng Xu, Xiaotian Han, Tianyi Zhang, and Vladimir Braverman. Demystifying OPD: Length Inflation and Stabilization Strategies for Large Language Models. *arXiv preprint arXiv:2604.08527*, 2026. URL https://arxiv.org/abs/2604.08527.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. CoT-Valve: Length-Compressible Chain-of-Thought Tuning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2502.09601v1.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651v2.

Mathematical Association of America. American Invitational Mathematics Examination. MAA Invitational Competitions, 2026. URL https://maa.org/maa-invitational-competitions/. Accessed 2026-09-25.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14734v3.

Michael R. Metel, Yufei Cui, Boxing Chen, and Prasanna Parthasarathi. Thinking Long, but Short: Stable Sequential Test-Time Scaling for Large Reasoning Models. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 2942–2951, Rabat, Morocco, March 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-eacl.153. URL https://aclanthology.org/2026.findings-eacl.153/.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling Length from Quality in Direct Preference Optimization. *arXiv preprint arXiv:2403.19159*, 2024. URL https://arxiv.org/abs/2403.19159v2.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290v3.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. CRISP: Compressed Reasoning via Iterative Self-Policy Distillation. *arXiv preprint arXiv:2603.05433*, 2026. URL https://arxiv.org/abs/2603.05433v7.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300v3.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-Distillation Enables Continual Learning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.19897v2.

Vaishnavi Shrivastava, Ahmed H. Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample More to Think Less: Group Filtered Policy Optimization for Concise Reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a00ef870c140f09263ba2288855be50b-Abstract-Conference.html.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2312.06585v4.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A Long Way to Go: Investigating Length Correlations in RLHF. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2310.03716v2.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314v1.

Mingyang Song and Mao Zheng. Walk Before You Run! Concise LLM Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2505.21178*, 2025. URL https://arxiv.org/abs/2505.21178v1.

Daniil Tiapkin, Daniele Calandriello, Johan Ferret, Sarah Perrin, Nino Vieillard, Alexandre Ramé, and Mathieu Blondel. On Teacher Hacking in Language Model Distillation. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.02671v1.

Chen Wang, Hexuan Deng, Yining Zhang, Yuchen Zhang, Jionghao Bai, Zhaochun Li, Ge Lan, and Yue Wang. Implicit Compression Regularization: Concise Reasoning via Internal Shorter Distributions in RL Post-Training. *arXiv preprint arXiv:2605.07316*, 2026a. URL https://arxiv.org/abs/2605.07316v1.

Rui Wang, Hongru Wang, Yi Chen, Boyang Xue, Tianqing Fang, Wenhao Yu, and Kam-Fai Wong. Demystifying On-Policy Distillation: Roles, Pathologies, and Regulations. *arXiv preprint arXiv:2607.13399*, 2026b. URL https://arxiv.org/abs/2607.13399v1.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171v4.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2201.11903v6.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models. *arXiv preprint arXiv:2408.00724*, 2024. URL https://arxiv.org/abs/2408.00724v3.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When More is Less: Understanding Chain-of-Thought Length in LLMs. *arXiv preprint arXiv:2502.07266*, 2025. URL https://arxiv.org/abs/2502.07266v3.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Xun Deng, Zhihao Zhang, Honglin Guo, Zhikai Lei, Miao Zheng, Guoteng Wang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, Qi Zhang, and Xuanjing Huang. BAPO: Stabilizing Off-Policy Reinforcement Learning for LLMs via Balanced Policy Optimization with Adaptive Clipping. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cca4df598d8066ac9e985852dd1d48b5-Abstract-Conference.html.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable Chain-of-Thought Compression in LLMs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://arxiv.org/abs/2502.12067v3.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just Enough Thinking: Efficient Reasoning with Adaptive Length Penalties Reinforcement Learning. *arXiv preprint arXiv:2506.05256*, 2025. URL https://arxiv.org/abs/2506.05256v2.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of Draft: Thinking Faster by Writing Less. *arXiv preprint arXiv:2502.18600*, 2025. URL https://arxiv.org/abs/2502.18600v2.

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*, 2026a. URL https://arxiv.org/abs/2602.12125v2.

Yuxiao Yang, Tianrun Yu, Shangzhe Li, Kaixiang Zhao, Xuchao Zhang, Chetan Bansal, Huaxiu Yao, Taylor W. Killian, and Weitong Zhang. When EOS Tokens Disagree: Understanding Length Inflation in On-Policy Distillation. *arXiv preprint arXiv:2609.20511*, 2026b. URL https://arxiv.org/abs/2609.20511.

Jingyang Yi, Jiazheng Wang, and Sida Li. ShorterBetter: Guiding Reasoning Models to Find Optimal Inference Length for Efficient Reasoning. *arXiv preprint arXiv:2504.21370*, 2025. URL https://arxiv.org/abs/2504.21370v4.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2503.14476v2.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.13837v5.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping Reasoning With Reasoning. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2203.14465v2.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2601.18734v3.

Wenhong Zhu, Ruobing Xie, Rui Wang, and Pengfei Liu. Hybrid Policy Distillation for LLMs. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2604.20244v2.

APPENDIX

A NUMERICAL OUTPUT-LENGTH COMPARISON

Family	Method	Mean tokens	Base ratio	Growth (%)
Base	Qwen3-1.7B	3,938	1.00×	0.0
SFT	SFT	4,124	1.05×	+4.7
RL	GRPO	5,089	1.29×	+29.2
	DAPO	6,435	1.63×	+63.4
	ProRL	5,978	1.52×	+51.8
	BAPO	6,031	1.53×	+53.1
	GFPO	5,277	1.34×	+34.0
OPD	OPD	8,374	2.13×	+112.6
	ExOPD	8,057	2.05×	+104.6
	OPSD	8,523	2.16×	+116.4
	HPD	9,248	2.35×	+134.8
	Veto	≈9,405	2.39×	+138.8
	DistiLLM-2	≈8,332	2.12×	+111.6

Table 4: Mean output length and growth in our common-base Qwen3-1.7B reproductions on the four-contest suite. Approximate measurements retain the ≈ mark; ratios and growth are calculated from the reported means.

Table 4 gives the numerical values underlying Figure 2, including each method's relative increase over the shared Qwen3-1.7B base. The comparison setup is described in Section 3.1.

B POST-TRAINING OBJECTIVES AND DERIVATIONS

This appendix specifies the offline objective, gradient calibration, and properties of the online rewards used in Section 3.

B.1 OFFLINE PREFERENCE CONSTRUCTION

Comparisons use completion-mean log-probabilities

$$\bar{\ell}_\theta(y_i \mid x) = \frac{1}{L_i} \sum_{t=1}^{L_i} \log \pi_\theta(y_{it} \mid x, y_{i,<t}). \tag{2}$$

From a frozen rollout pool, select the shortest and longest distinct correct responses y_s and y_l, and up to two hard-negative incorrect responses $\mathcal{W}_h$ ranked by reference completion-mean likelihood. Let $\mathcal{C}'$ contain the selected correct responses without duplicates. We learn $y_c \succ y_w$ for $y_c \in \mathcal{C}'$, $y_w \in \mathcal{W}_h$, and $y_s \succ y_l$. Using a completion-mean variant of DPO (Rafailov et al., 2023), define

$$\mathcal{L}_{\text{pair}}(a \succ b) = -\log \sigma\big(\beta \left[(\bar{\ell}_\theta(a) - \bar{\ell}_{\text{ref}}(a)) - (\bar{\ell}_\theta(b) - \bar{\ell}_{\text{ref}}(b))\right]\big). \tag{3}$$

The prompt-level objective combines the mean correct–wrong pair loss $\mathcal{L}_{\text{cw}}$, the short–long pair loss $\mathcal{L}_{\text{sl}}$, and an absolute likelihood anchor:

$$\mathcal{L}_{\text{offline}} = \mathbb{E}_{x \sim \mathcal{D}_+}\left[\mathcal{L}_{\text{cw}} + \lambda_b \mathcal{L}_{\text{sl}} - \frac{\alpha}{m} \sum_{i \in \mathcal{C}} \bar{\ell}_\theta(y_i \mid x)\right]. \tag{4}$$

Here $\mathcal{D}_+$ contains prompts with at least one correct response. Appendix B.2 specifies the fixed loss weights.

For each prompt, $\mathcal{C}$ indexes all verifier-correct sampling records, including repeated samples of the same completion. The preference set $\mathcal{C}'$ contains the shortest and longest distinct correct token sequences, with at most two endpoints. Ties are resolved by token-ID hash. The hard-negative set

$\mathcal{W}_h$ contains up to two distinct incorrect sequences with highest frozen-reference completion-mean likelihood, using the same tie-breaking rule. The losses are

$$\mathcal{L}_{\mathrm{cw}}(x) = \frac{1}{|\mathcal{C}'|\,|\mathcal{W}_h|} \sum_{y_c \in \mathcal{C}'} \sum_{y_w \in \mathcal{W}_h} \mathcal{L}_{\mathrm{pair}}(y_c \succ y_w), \tag{5}$$

$$\mathcal{L}_{\mathrm{sl}}(x) = \mathcal{L}_{\mathrm{pair}}(y_s \succ y_l), \qquad \mathcal{L}_{\mathrm{anchor}}(x) = -\frac{1}{m} \sum_{i \in \mathcal{C}} \bar{\ell}_\theta(y_i \mid x). \tag{6}$$

An unavailable pair type contributes zero. A brevity pair requires two correct sequences of unequal length. Prompts with no correct sample are excluded from the eligible training set; all-correct prompts contribute the anchor and any available brevity pair. Each anchor term is a completion-token mean, followed by an average over correct sampling records. For example, a completion sampled three times receives weight $3/m$ in the anchor, independently of its participation in preference pairs. Prompt and padding tokens are excluded; generated EOS tokens are included when covered by the frozen completion mask.

All three losses use the same outer denominator, the number of eligible prompts in the batch. Thus, missing loss terms contribute zero rather than changing the prompt weighting. This gives each prompt equal outer weight regardless of its number of correct samples or preference pairs.

B.2 Fixed DPO Gradient Calibration

The three loss weights are calibrated at the unmodified distilled-checkpoint parameters θ_0 on a frozen set $\mathcal{D}_{\mathrm{cal}}$ of 64 eligible prompts. For $j \in \{\mathrm{cw}, \mathrm{sl}, \mathrm{anchor}\}$, define

$$g_j^0 = \nabla_\theta \left[\frac{1}{64} \sum_{x \in \mathcal{D}_{\mathrm{cal}}} \mathcal{L}_j(x; \theta) \right] \Bigg|_{\theta = \theta_0}, \qquad G_j = \|g_j^0\|_2. \tag{7}$$

These are norms of gradients of the mean losses over the same calibration prompts and full trainable parameter set, measured before clipping, Adam preconditioning, and weight decay. They determine the fixed weights

$$\lambda_b = 0.10 \frac{G_{\mathrm{cw}}}{G_{\mathrm{sl}}}, \qquad \alpha = 0.25 \frac{G_{\mathrm{cw}}}{G_{\mathrm{anchor}}}. \tag{8}$$

The calibration requires finite, nondegenerate norms. The resulting ratio $1 : 0.10 : 0.25$ applies to the three weighted aggregate gradient norms at initialization; the coefficients remain fixed throughout training. For a batch $\mathcal{B}$ of B eligible prompts, optimization uses

$$\mathcal{L}_B = \frac{1}{B} \sum_{x \in \mathcal{B}} \left[\mathcal{L}_{\mathrm{cw}}(x) + \lambda_b \mathcal{L}_{\mathrm{sl}}(x) + \alpha \mathcal{L}_{\mathrm{anchor}}(x) \right]. \tag{9}$$

We accumulate the combined weighted gradient over an effective batch of 16 prompts, apply one global gradient clip at norm 1.0, and take one AdamW step. All-correct prompts use the same fixed coefficients with zero correctness-pair loss. Table 5 summarizes the available training signals.

B.3 Online Reward Details

For a prompt x, let K sampled completions have correctness $c_i \in \{0, 1\}$ and generated length $L_i \geq 1$. Define the correct set $\mathcal{C} = \{i : c_i = 1\}$, its size $m = |\mathcal{C}|$, and the group success rate $p = m/K$. RL1 uses the correctness reward and group-mean-centered advantage

$$r_i^{\mathrm{corr}} = c_i, \qquad A_i^{\mathrm{corr}} = c_i - p. \tag{10}$$

Thus, mixed groups assign positive advantages to correct responses and negative advantages to incorrect responses, while all-correct and all-incorrect groups have zero reward-derived advantages.

RL2 uses Eq. (1). For $m \geq 2$, let $z_i = \log L_i$, let $\mu_{\mathcal{C}}$ be its mean over correct responses, and let $d_{\mathcal{C}}$ be its median absolute deviation. The centered efficiency score is

$$u_i = -\operatorname{clip}\left(\frac{z_i - \mu_{\mathcal{C}}}{d_{\mathcal{C}} + \epsilon_L}, -1, 1 \right), \qquad e_i = u_i - \frac{1}{m} \sum_{j \in \mathcal{C}} u_j, \quad i \in \mathcal{C}, \tag{11}$$

with $\epsilon_L > 0$. We set $e_i = 0$ for incorrect responses and set all efficiency scores to zero when $m \leq 1$. Subtracting the group mean from the RL2 reward yields

$$A_i^{\text{conc}} = c_i - p + \lambda(p) c_i e_i. \tag{12}$$

The first term learns correctness; the second redistributes reward among correct solutions according to length. The evaluated implementation subtracts the group mean without dividing by the group standard deviation, preserving the intended amplitude of the efficiency signal (Liu et al., 2025b).

B.4 Shared Online Surrogate

Let $h_{it} = (x, y_{i,<t})$ and $\rho_{it} = \pi_\theta(y_{it} \mid h_{it}) / \pi_{\text{old}}(y_{it} \mid h_{it})$. For detached response-level advantages A_i, the base clipped surrogate is

$$\begin{aligned}\mathcal{L}_{\text{online}} = & -\mathbb{E}_{x, \mathcal{Y} \sim \pi_{\text{old}}} \left[\frac{1}{K} \sum_{i=1}^{K} \frac{1}{L_i} \sum_{t=1}^{L_i} \min\{\rho_{it} A_i, \text{clip}(\rho_{it}, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}) A_i\} \right] \\ & + \eta_{\text{KL}} \mathcal{K}(\theta; \pi_{\text{ref}}).\end{aligned} \tag{13}$$

The rollout policy changes during training while the OPD reference remains frozen. $\mathcal{K}$ denotes the reference KL regularizer at rollout prefixes. The evaluated configuration additionally specifies a dual-clipping parameter (Appendix C); the expression above isolates the shared base surrogate. When all advantages are zero, the reward-derived gradient vanishes, while the KL regularizer can still update the policy.

B.5 Centered Efficiency and Reward Bounds

Correct count	Offline preference objective	Correctness-only advantage	Score-conditioned advantage
$m = 0$	Omit prompt	0	0
$m = 1$	Correct–wrong pairs and anchor	$c_i - p$	$c_i - p$
$2 \leq m < K$	Both pair types and anchor	$c_i - p$	$c_i - p + \lambda(p) c_i e_i$
$m = K$	Short–long pair and anchor	0	$\lambda_{\max} e_i$

Table 5: Training signals available from a group of $K \geq 2$ responses. Short–long pairs require unequal correct-response lengths. The anchor averages all correct sampling records; preference pairs use the selected distinct endpoints. Online entries describe reward-derived advantages; the reference KL term is separate.

For $m \geq 2$, the scale in Eq. (11) is

$$d_{\mathcal{C}} = \text{median}_{j \in \mathcal{C}} \left| z_j - \text{median}_{k \in \mathcal{C}} \, z_k \right|. \tag{14}$$

Clipping gives $u_i \in [-1, 1]$, and subtracting its mean gives $|e_i| \leq 2$ and $\sum_{i \in \mathcal{C}} e_i = 0$. Consequently, for binary c_i,

$$\frac{1}{K} \sum_i r_i^{\text{conc}} = \frac{1}{K} \sum_i c_i + \frac{\lambda(p)}{K} \sum_{i \in \mathcal{C}} e_i = p. \tag{15}$$

Subtracting this mean yields Eq. (12). Incorrect-response advantages are $-p$ and independent of their lengths within a fixed group. The lower bound on a correct response's raw reward is $1 - 2\lambda(p)$, so $\lambda_{\max} < 1/2$ is sufficient for positivity. A sufficient condition for all correct responses to have positive *advantages* is stronger: $2\lambda(p) < 1 - p$. At high success rates, a long correct response can receive negative advantage even though its raw reward is positive.

When $p = 1$, the correctness component cancels, leaving $A_i = \lambda_{\max} e_i$. For nonzero efficiency variance, dividing by the group's reward standard deviation would cancel this amplitude, up to a numerical stabilizer. This motivates mean centering without per-group standardization. If $m \leq 1$ or all correct-response lengths are identical, efficiency is zero. Table 5 summarizes these support cases.

Setting	DPO	RL1	RL2
Training steps / budget	50	120	120
Learning rate	10^{-6}	10^{-7}	10^{-7}
Correctness feedback	Preference pairs	Binary reward	Binary reward
Explicit length feedback	Short–long pairs	None	Score-conditioned

Table 6: Optimization settings for concise post-training. RL entries give the configured step budget; collapsed runs are identified in Table 2. The DPO anchor and additional optimization parameters are described in Appendix C.

C Implementation and Optimization Details

C.1 Concise Post-Training

DPO uses 50 steps, learning rate 10^{-6}, $\beta = 0.1$, and an effective batch of 16 prompts. Its correctness:brevity:anchor ratio $1 : 0.10 : 0.25$ sets the fixed initialization-calibrated weights in Eq. (8). The RL configuration has a 120-step budget, learning rate 10^{-7}, group size $K = 8$, PPO clipping parameter 0.2, dual-clipping parameter 3.0, KL-loss coefficient 0.01, and batch-size setting 16. RL2 uses $\lambda_{\max} = 0.10$ and $\gamma = 2$. A run is marked as collapsed when explosive response-length growth or a breakdown in task performance prevents further training. This qualitative criterion identifies the RL1 failures reported in Table 2.

C.2 Prompted Continuation

Every initial response receives the prompt `but wait, let me check if there is a mistake`, irrespective of initial-answer correctness. The model continues with fixed parameters, and evaluation scores its final answer after continuation. No separate continuation-specific token cap is imposed. We record the generated length after continuation; the reported length includes the initial response and the appended generation.

C.3 Renewed Distillation

Each compressed checkpoint is trained further with its original OPD, ExOPD, or OPSD procedure, independently of the continuation branch. All three routes use batch size 64 and one rollout per prompt. Their evaluated checkpoints are selected at approximately 160 training steps, when performance on the validation set has stabilized.

C.4 Training-Order Comparison

All sequences in Table 3 begin from Qwen3-1.7B. RL2 $\rightarrow$ OPD uses 200 steps of RL2 followed by 160 steps of OPD. The OPD $\rightarrow$ RL2 row reuses the main compression checkpoint, obtained with a 120-step RL2 budget after the original 160-step OPD run. DPO $\rightarrow$ OPD and OPD $\rightarrow$ DPO use 50+160 and 160+50 steps, respectively.

D Additional Qwen3-4B Results

We additionally evaluate selected OPD and ExOPD trajectories with Qwen3-4B on the same four-benchmark competition-mathematics suite. Table 7 reports the pooled sample accuracy. The Qwen3-4B results reproduce the main within-scale trend: compression improves both OPD and ExOPD, while prompted continuation and renewed distillation further improve the compressed ExOPD policy.

Recipe	Average@8 (↑)
Base model	15.0
OPD	42.3
OPD → compression	**43.6** (+1.4)
ExOPD	45.2
ExOPD → compression	46.3 (+1.0)
ExOPD → compression → continuation	47.3 (+2.1)
ExOPD → compression → renewed ExOPD	**47.8** (+2.6)

Table 7: Additional Qwen3-4B results for the available OPD and ExOPD trajectories. Average@8 is computed over 960 sampled responses from AIME24, AIME25, HMMT-Feb, and HMMT-Nov. Parentheses report percentage-point changes relative to the corresponding original distilled checkpoint. Bold marks the best result within each route.

E Termination Mismatch and OPD Length Inflation

Yang et al. (2026b), in *When EOS Tokens Disagree: Understanding Length Inflation in On-Policy Distillation*, study how termination behavior changes under sampled-token OPD. A base student and a post-trained teacher can prefer different end-of-sequence (EOS) tokens, even when both tokens are recognized as valid stopping actions. At a student-generated stopping token, a low teacher probability produces a negative distillation signal. This suppresses the student's familiar way of stopping, while the teacher's preferred alternative may be sampled too rarely to learn reliably.

Their correction aggregates the probabilities of semantically equivalent EOS tokens into one stopping action for supervision. Across Qwen3, Llama, and Gemma, this reduces mismatch-induced length inflation; merely expanding the decoder's stopping set is insufficient. Their K2-Horizon experiments also exhibit late-stage length growth after termination alignment, separating this residual behavior from the initial token mismatch.

This distinction clarifies the motivation in Section 3.1: a response can grow because its stopping behavior changes, without a corresponding increase in useful reasoning. Our intervention operates on the resulting policy. Concise post-training favors shorter correct responses, and the two subsequent branches test whether the released output space supports additional inference or renewed teacher-guided learning. EOS alignment and post-distillation compression address different stages of this process: the former corrects a supervision mismatch, while the latter learns a compact starting policy for further improvement.

F Extended Related Work

Three questions connect the literature relevant to this work: how teacher-guided training changes a student's response distribution, how post-training can favor concise correct reasoning, and how a compact policy can use additional inference or training to improve. We review these connections below, with particular attention to the methods closest to our compression and budget-reuse stages.

F.1 Distillation on Student-Generated Trajectories

The defining choice in on-policy distillation is to obtain supervision at prefixes visited by the student. GKD formulates learning from student-generated trajectories with a choice of distribution-matching losses, addressing the mismatch between fixed training sequences and the student's own generations (Agarwal et al., 2024). MiniLLM uses reverse-KL distillation with policy-gradient optimization and stabilizing mechanisms, including teacher-mixed sampling and length normalization (Gu et al., 2024). These works distinguish the source of training trajectories from the divergence used to match the teacher. Both choices affect what the student learns.

Subsequent methods modify the matching objective and the use of generated data. DistiLLM combines skew divergences with adaptive reuse of student-generated samples, while DistiLLM-2 introduces contrastive signals from teacher- and student-generated sequences (Ko et al., 2024; 2025).

HPD combines forward- and reverse-KL guidance through token-level weighting of offline and student-sampled actions (Zhu et al., 2026). Veto constructs an adaptive target from teacher–student agreement to regulate unstable gradients and distributional concentration (Jang et al., 2026). These approaches improve the quality or stability of knowledge transfer through different forms of supervision.

Other approaches change the strength or information content of the teacher signal. ExOPD extrapolates the reward implicit in a teacher relative to a reference policy, allowing the student to move beyond direct imitation (Yang et al., 2026a). OPSD uses the same model under an informative teacher condition, such as access to a solution or a different reasoning mode, to supervise student rollouts (Zhao et al., 2026). Relatedly, SDFT conditions a self-teacher on demonstrations for continual learning (Shenfeld et al., 2026). This form of self-distillation provides teacher feedback along student prefixes, whereas supervised learning on a fixed demonstration trains on the demonstration's prefixes. Our post-distillation stage can follow different teacher-guided routes because it learns from the completed checkpoint's responses and their correctness.

F.2 Length Dynamics in Distillation and Reinforcement Learning

Output length reflects both the supervision signal and the way an objective is optimized. MiniLLM analyzes a short-response bias in its uncorrected objective (Gu et al., 2024), whereas ExOPD reports response growth as distillation and reward extrapolation proceed (Yang et al., 2026a). Recent analyses study long-response inflation, short-response degeneration, and sensitivity to teacher–student reasoning patterns and generation budgets (Wang et al., 2026b; Li et al., 2026). Luo et al. (2026) further connect OPD length inflation with repetition and truncation collapse. Appendix E discusses a distinct termination-level mechanism. Together, these studies motivate examining actual length trajectories and endpoints under specified training configurations.

Teacher agreement and downstream correctness also measure different aspects of learning. Work on teacher hacking shows how improving a distillation proxy can diverge from the desired target behavior (Tiapkin et al., 2025). Entropy-aware OPD adjusts distribution matching at uncertain positions to preserve useful alternatives (Jin et al., 2026). These perspectives help explain why a slowly changing teacher-matching objective need not establish that every part of a response remains useful. Our common-base comparison measures the response-length consequences across several distillation methods, and the compression experiments test whether their long outputs can be shortened while preserving or improving task performance.

RL and preference learning have their own length dynamics. Analyses of RLHF and DPO identify correlations between response length and learned preferences (Singhal et al., 2024; Park et al., 2024). For reasoning RL, Dr. GRPO studies how response-length and reward-variance normalization affect optimization (Liu et al., 2025b). DAPO combines token-level optimization, sampling changes, and overlong-response reward shaping (Yu et al., 2025); BAPO adjusts clipping to balance policy updates and stabilize learning from off-policy samples (Xi et al., 2026). GFPO explicitly filters larger rollout groups by length or token efficiency to train more concise policies (Shrivastava et al., 2026). This literature identifies several ways to influence length through rewards, sample selection, and optimization, which provide useful context for the SFT and RL endpoints in our comparison.

F.3 Concise Reasoning through Data and Preferences

Intermediate reasoning can improve problem solving by making useful computations explicit (Wei et al., 2022). The benefit depends on what the additional steps accomplish. Studies of overthinking identify repeated solutions and low-value continuations (Chen et al., 2025); analyses of reasoning length describe a trade-off between decomposing a problem into easier steps and introducing further opportunities for error (Wu et al., 2025). This motivates learning compact successful trajectories while retaining the steps needed to solve the problem.

Supervised learning offers a direct route through the choice of training responses. STaR and ReST-EM generate candidate solutions, select successful ones, and train on the selected trajectories (Zelikman et al., 2022; Singh et al., 2024). For explicit compression, TokenSkip constructs shorter reasoning traces by removing low-importance tokens before supervised training, while CoT-Valve learns controllable changes in chain length through parameter-efficient tuning (Xia et al., 2025; Ma

et al., 2025). These methods shape the response distribution through compact training targets or a learned control direction.

Preference learning instead compares alternative responses to the same question. DPO learns from preferred and rejected completions relative to a reference policy (Rafailov et al., 2023), and SimPO uses completion-mean log-probabilities in a reference-free preference objective (Meng et al., 2024). A close predecessor for our offline branch is the overthinking study of Chen et al. (2025), which constructs short–long correct-response preferences and evaluates SFT, DPO, likelihood-regularized preference learning, and SimPO. Our offline implementation combines such a brevity preference with correct–incorrect pairs and a likelihood anchor over correct responses. Its role in the present study is to test compression of already-distilled checkpoints alongside online alternatives, before evaluating the resulting policies' further inference and learning.

F.4 Online Objectives for Controlling Reasoning Length

Online methods can use newly sampled solutions to adjust how much reasoning a model produces. GRPO obtains a relative learning signal from groups of responses to the same prompt (Shao et al., 2024). L1 trains a model to follow a requested reasoning budget, with objectives for exact lengths or maximum lengths (Aggarwal & Welleck, 2025). ShorterBetter derives a target from sampled solution lengths, favoring responses near the shortest observed correct solution when one is available (Yi et al., 2025). These objectives illustrate two useful forms of control: following an externally specified budget and adapting to successful responses discovered by the current policy.

Correctness-aware objectives regulate when brevity should influence an update. Arora & Zanette (2025) apply relative length feedback within correct responses while maintaining their reward advantage over incorrect ones. Adaptive Length Penalties scale compression pressure with the current success rate (Xiang et al., 2025). ConciseR first strengthens reasoning, then activates its length signal on groups in which every response is correct (Song & Zheng, 2025). JET constructs shorter on-policy trajectories through truncation and resumed generation, together with correctness-gated length feedback (Han et al., 2025). Our score-conditioned GRPO follows this line of work: it uses a continuous success-rate weight and centered efficiency scores to redistribute reward among correct solutions. The reward decomposition in Appendix B.3 explains how correctness and brevity contribute in mixed and all-correct groups.

Compression can also emerge without a direct token penalty. Fatemi et al. (2025) study a two-stage RL procedure that develops reasoning and then encourages concise behavior through training on solvable problems. ICR adds a compression signal derived from the shortest correct on-policy samples (Wang et al., 2026a). These results motivate separating the effect of correctness learning from that of an explicit length objective. Our correctness-only and score-conditioned GRPO branches share their optimization settings, so their comparison isolates the added length feedback at the same distilled starting point.

F.5 Compression through Self-Distillation and Decoding

CRISP is a particularly relevant distillation-based approach: a conciseness instruction defines the self-teacher, which supervises student-generated trajectories, and periodic teacher refreshes sustain compression (Sang et al., 2026). It demonstrates that teacher-guided training can itself be directed toward shorter reasoning when conciseness is built into the teacher condition. Our pipeline studies a different stage arrangement: it takes an existing distilled policy, applies correctness-based concise post-training, and evaluates how the resulting checkpoint benefits from subsequent expansion. The two approaches connect through the response distribution used as the target, while assigning different roles to the compression stage.

Length can also be changed at inference time. Chain of Draft prompts compact intermediate steps (Xu et al., 2025), and s1 controls when reasoning ends through budget forcing (Muennighoff et al., 2025). Such interventions alter the text produced under fixed parameters. Training a concise policy changes the checkpoint from which later inference or learning begins. This distinction is central to our two follow-up branches: prompted continuation measures fixed-policy behavior, whereas renewed OPD measures learning from the compressed initialization.

F.6 Reallocating Inference Budget

Test-time scaling studies how additional computation should be distributed across attempts. Self-consistency samples multiple reasoning paths and aggregates their answers (Wang et al., 2023). Compute-allocation studies compare revisions, independent samples, verification, and search, showing that the useful allocation depends on the problem and the model (Snell et al., 2024; Wu et al., 2024). In a finite output window, a compact initial response leaves more space for additional reasoning along the same trajectory. This is the inference motivation for our compression stage.

Sequential extension has several established forms. s1 prolongs reasoning with a continuation cue (Muennighoff et al., 2025). Self-Refine explicitly generates feedback and revises an answer (Madaan et al., 2023), while SCoRe trains multi-turn self-correction through RL (Kumar et al., 2025). Work on intrinsic self-correction also shows that revision can turn a correct answer into an incorrect one (Huang et al., 2024). Accordingly, our continuation protocol applies the same cue to every response and evaluates the final answer, counting both the initial and appended tokens. This evaluates the policy's use of additional space without selecting which responses to continue using their correctness.

Compression has already been connected to inference scaling. ExpThink learns from historical short correct solutions and evaluates compressed policies under multi-sample answer aggregation (Bian et al., 2026). Its results establish a close precedent for reinvesting released tokens in more attempts. Our continuation branch instead follows each initial response with a sequential extension. Reporting both average@8 and pass@8 describes the final samples' mean correctness and solved-problem coverage; these measure different outcomes from majority-vote accuracy. The shared question is whether compression creates a better starting distribution for the chosen way of spending further output budget.

F.7 Initialization and Further Post-Training

The outcome of a later training stage depends on the policy from which it starts. Li et al. (2026) study how SFT initialization and teacher–student reasoning-pattern alignment affect OPD. HPD also evaluates its learned policy as an initialization for further OPD (Zhu et al., 2026). These studies connect the quality of subsequent teacher-guided learning to the trajectories made available by earlier training. Our renewed-distillation experiment examines this connection after an initial OPD run has reached its observed plateau: a concise intermediate objective changes the checkpoint, after which the original teacher-guided procedure is resumed.

Continued learning must also be understood through the evolution of the response distribution. ProRL studies sustained RL and expansion of the set of solvable problems (Liu et al., 2025a). Research on RL reasoning capacity distinguishes improvements at small sampling budgets from coverage at larger budgets (Yue et al., 2025). Studies of SFT followed by RL further show that later gains can depend on capabilities changed during the preceding stage (Jin et al., 2025). These findings motivate tracking the original distilled checkpoint, the compressed checkpoint, and the renewed endpoint together. In our pipeline, compression has two connected roles: improving the immediate accuracy–length trade-off and preparing a policy that can turn renewed output growth into further gains. The continuation and renewed-distillation experiments test those opportunities separately, with fixed parameters in the former and additional teacher-guided updates in the latter.