LEARNING FROM HINTS, FORGETTING LESS: REWARD-DIRECTED TRANSFER FOR CONTINUAL POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training-time hints can help models solve difficult tasks, creating successful experience from which to learn. In continual post-training, the challenge is to turn these assisted successes into independent capabilities while retaining earlier ones. However, solely imitating hint-assisted success can pull the model toward particular answers or hint-dependent predictions, introducing behavioral changes that may interfere with previously acquired capabilities. We introduce Hint-Gap Weighted Transfer (HiGHT), which combines online reinforcement learning (RL) with selective transfer from hinted rollouts. Its central signal is the *hint gap*: the signed change in a sampled token's probability induced by the hint at the same response prefix. Positive trajectory advantages select reward-supported experience, while the hint gap modulates the direction and strength of each token's transfer contribution. Raw-prompt reinforcement learning continues to optimize task reward without hints. Across five real-world vision-language domains, HiGHT improves mean final accuracy from 49.94% to 55.12% and backward transfer from -6.03 to -3.18 percentage points over hinted-policy distillation combined with online RL (Distill + RL). Controlled local updates show better early-stage retention at matched update norms. These results support learning from hint-assisted experience with less forgetting.

1 INTRODUCTION

A hint can help a learner solve a difficult problem, but learning from that hint means being able to solve similar problems independently. Training-time hints offer language models a similar opportunity: they can elicit successful responses on tasks the model cannot yet solve reliably on its own (Zelikman et al., 2022). In continual post-training, this opportunity comes with a second demand: models must acquire new capabilities while retaining earlier ones. Sequential fine-tuning can erode previously acquired knowledge and reasoning abilities (Luo et al., 2025), and the same acquisition–retention challenge arises across visual-language domains (Zhao et al., 2025). The goal is therefore to turn assisted successes into independent capabilities while preserving what the model already knows.

Reinforcement learning (RL) learns from task outcomes while allowing multiple successful solutions. Online RL can favor successful policies close to the starting distribution, helping preserve previously acquired capabilities (Shenfeld et al., 2026b). Yet this learning process depends on obtaining informative reward feedback. On difficult tasks, rollout groups with uniformly incorrect answers yield zero correctness-based advantages in group-relative methods such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Xia et al., 2026). Training-time hints can make successful rollouts accessible (Qu et al., 2026), but the model must ultimately perform at test time without that additional information. This raises the central question: *how can these assisted successes improve performance under the original, raw prompt while preserving previously acquired capabilities?*

The difficulty lies in choosing what to transfer from hint-assisted success. A successful response offers one way to solve the task, while its token probabilities also reflect the additional information supplied by the hint. Supervised fine-tuning (SFT) on correct hinted rollouts encourages reproducing particular successful responses (Zelikman et al., 2022); distillation encourages matching the hinted policy's predictive distribution (Hübotter et al., 2026). These targets can favor particular solutions

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.

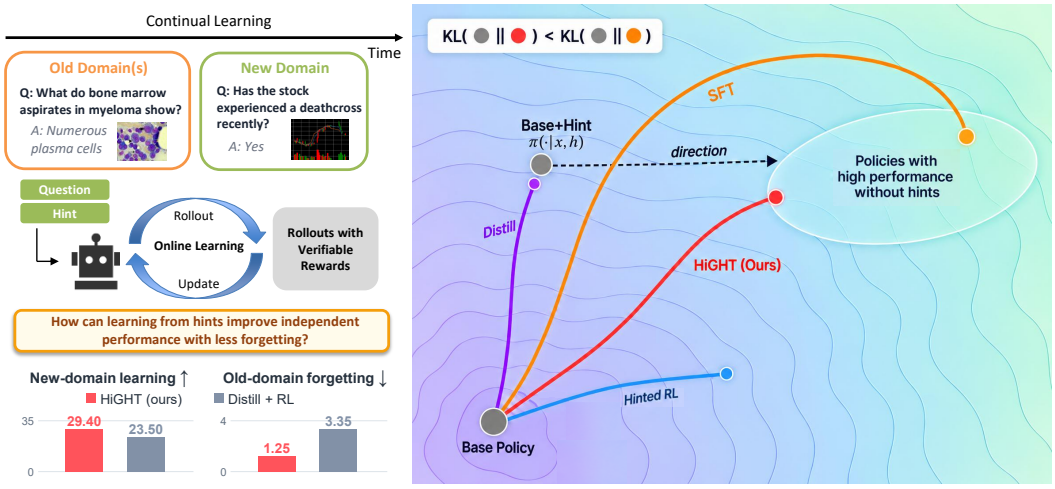

Figure 1: **How can learning from hints improve independent performance with less forgetting? Top left:** Continual learning with training-time hints and evaluation without hints. **Bottom left:** HiGHT improves new-domain learning and reduces old-domain forgetting compared with hinted-policy distillation combined with online RL (Distill + RL). Bars show new-domain accuracy gains and mean old-domain accuracy losses after 100 steps of finance-domain adaptation, relative to each method's own initial checkpoint, in percentage points (Section 4.2). **Right:** Schematic comparison of learning dynamics between our method, HiGHT, and alternative training strategies. SFT (orange) fits successful hinted responses; distillation (purple) targets hinted predictions; hinted-prompt RL (blue) optimizes reward with hints and affects the raw policy through shared parameters. HiGHT (red) uses rewards and signed hint gaps to guide transfer toward successful performance without hints.

or context-dependent preferences beyond what is needed for raw-prompt success. Such additional changes matter in continual learning: fitting response distributions can induce greater policy drift and forgetting (Shenfeld et al., 2026b), and dense self-distillation can reinforce incidental behaviors alongside useful ones (Wang et al., 2026a). The challenge is to retain the value of hinted experience while making its influence responsive to task reward and the policy being trained.

Our key insight is that a successful hinted response need not be reinforced uniformly: reward evaluates the response as a whole, while the hint's effect on individual token probabilities provides a finer-grained signal for transfer. We call this signed probability difference the *hint gap*. Evaluated under the current policy at the same response prefix, it is positive when the hint makes the sampled token more likely, negative when it makes that token less likely, and zero when the two probabilities agree. The gap makes the hint's influence explicit at each token, allowing a selected response to provide both encouragement and suppression during transfer.

We introduce *Hint-Gap Weighted Transfer* (HiGHT) to combine this token-level guidance with response-level reward selection. Positive reward-derived advantages select which hinted responses contribute to transfer. For each selected response, the signed hint gap modulates its token weights: a positive gap contributes to increasing raw-prompt log-likelihood, whereas a negative gap contributes to decreasing it. These auxiliary updates accompany online RL under raw and hinted prompts, with raw-prompt RL directly optimizing reward without hints. This design uses hint-assisted experience to support independent task success while making transfer selective at both the response and token levels. Figure 1 illustrates the continual-learning setting and key results, and compares the learning dynamics of HiGHT with alternative training strategies.

Our experiments cover controlled Hidden-Operator Continual Learning (HOCL), five-domain MLLM-CL (Zhao et al., 2025), and integration with Privileged On-Policy Exploration (POPE) (Qu et al., 2026) on hard mathematics. POPE constructs training-time guides from oracle-solution prefixes and applies RL to both raw and guided prompts. Hints are used only during training; all reported accuracy evaluations use raw prompts. On HOCL, HiGHT reaches 100% accuracy on all three tasks with zero backward transfer at the reported stage checkpoints (Table 1). On MLLM-CL, it improves mean finetune accuracy from 54.76% to 57.66%, mean final accuracy from 49.94% to

55.12%, and backward transfer from -6.03 to -3.18 percentage points over a baseline combining hinted-policy distillation with online RL (Distill + RL; Table 2). Controlled local updates at shared checkpoints, using the same rollouts and matched update norms, yield an average early-stage old-domain retention advantage of 0.436 percentage points over Distill + RL (Figure 3). With POPE's guide construction held fixed, the same transfer rule raises raw-prompt pass@1 from 4.64% to 8.48% over our implementation of POPE after 200 training steps under a common evaluation token cap (Table 3).

Our contributions are as follows.

- We introduce HiGHT, a transfer rule that combines positive trajectory advantages with signed token-level hint gaps to learn from hinted rollouts alongside raw-prompt RL.
- We characterize the effect of the transfer objective through paired local updates with shared checkpoints, identical rollouts, and matched update norms. These comparisons show better early-stage retention with HiGHT than with Distill + RL.
- We validate the transfer rule in controlled continual learning, five real-world vision-language domains, and integration with POPE. These evaluations show improved learning and retention over the compared online SFT and distillation objectives, and better utilization of fixed hints on hard mathematics.

2 Related Work

Retention in continual post-training. Continual learning preserves earlier capabilities through parameter-importance regularization (Kirkpatrick et al., 2017; Zenke et al., 2017), distillation of earlier outputs (Li & Hoiem, 2017), and exemplar or episodic-memory constraints (Rebuffi et al., 2017; Lopez-Paz & Ranzato, 2017). The acquisition–retention problem also arises in language-model fine-tuning (Luo et al., 2025) and across multimodal domains (Zhao et al., 2025). In RL post-training, CPO regularizes sparse parameter movement (Luo et al., 2026), while RAPO combines risk-aware update scaling and sampling (Liu et al., 2026). These approaches motivate controlling how new-task updates affect earlier capabilities. HiGHT addresses a complementary choice within the learning objective: which parts of newly generated, hint-assisted experience should guide the policy used without hints. Our matched-norm local comparisons examine how this transfer choice affects retention when update magnitude is controlled.

Hints for exploration and learning. Rationale bootstrapping turns answer-assisted generations into supervised training data (Zelikman et al., 2022). Privileged On-Policy Exploration (POPE) constructs guides from oracle-solution prefixes and trains on raw and guided prompts (Qu et al., 2026); HiLL learns adaptive hints with an explicit transferability reward (Xia et al., 2026); HINT combines abstract Meta-Hints with affinity-aware policy optimization (Wang et al., 2026b). Hint construction and learner optimization jointly determine the value of assisted experience. HiGHT focuses on the learner's transfer rule: rewarding hinted responses supply raw-prompt likelihood updates whose token contributions depend on the hint gap. Holding POPE's guides fixed in our integration experiment isolates this choice from improvements to hint construction.

Privileged self-distillation. Self-Distillation Fine-Tuning (SDFT) uses a demonstration-conditioned self-teacher for low-forgetting adaptation (Shenfeld et al., 2026a). Other approaches condition the teacher on feedback (Hübotter et al., 2026), filter distillation by correctness (Ding, 2026), or evaluate privileged predictions on student-generated trajectories (Zhao et al., 2026). Such conditioning provides richer supervision, but its effects depend on the transfer objective: Wang et al. (2026a) find that dense self-distillation can amplify incidental behaviors and forgetting. HiGHT uses the change in the sampled token's probability under a hint to construct a signed likelihood coefficient, gated by trajectory advantage. This makes the hinted prediction a token-level transfer signal within reward-based learning, while raw-prompt RL directly optimizes the deployment task.

Selective trajectory and token supervision. Prior work already combines trajectory assessment with token-dependent guidance. GATES uses tutor consensus to gate distillation and includes a clipped tutor–student log-probability-gap objective on student rollouts (Stein et al., 2026). HiGHT instead selects hinted rollouts through task-reward advantages and uses the signed probability gap at

each sampled token. SCOPE routes correct and incorrect student trajectories to differently weighted MLE and KL objectives (Zheng et al., 2026); DASH aggregates divergence-dependent supervision across token positions (Hou et al., 2026). Our transfer coefficient combines a response-level positive-advantage gate with a same-prefix hint gap, permitting encouragement and suppression within a selected hinted response. The resulting auxiliary update accompanies raw-prompt RL, and our evaluation measures both independent task learning and retention.

RL with auxiliary experience. LUFFY combines on-policy rollouts with off-policy demonstrations through regularized importance-sampling policy shaping (Yan et al., 2026). In HiGHT, auxiliary experience comes from the same model under a hinted prompt, and transfer compares its raw and hinted predictions at a shared response prefix. Advantage-weighted regression uses positive exponential advantage weights to fit sampled actions (Peng et al., 2019). Our centered exponential coefficient additionally depends on the signed hint gap: even a positive-advantage response can contribute negative token coefficients. This distinction determines how assisted experience modifies the raw policy alongside its own reward-driven updates.

3 Method

3.1 Problem Setup

Let x denote the raw prompt and h denote auxiliary training-only context. The *hinted prompt* (x, h) augments or modifies the original input with an answer (Zelikman et al., 2022), a task-specific mapping, a rationale prefix (Qu et al., 2026), a demonstration (Shenfeld et al., 2026a), or an exploration guide (Xia et al., 2026). Both prompt conditions share model parameters. At each continual post-training stage, the model learns a new task and is evaluated without hints on new and previous tasks. For the current task distribution $\mathcal{D}$, the raw-prompt objective is

$$J_{\text{raw}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r(x,y)], \tag{1}$$

where $r(x, y)$ is the evaluated task reward, including any configured auxiliary components.

Group-relative RL compares responses through their rewards. For example, GRPO standardizes rewards within a group $\{y_i\}_{i=1}^{G}$ sampled under the same prompt:

$$\hat{A}_i = \frac{r(x,y_i) - \text{mean}_j\, r(x,y_j)}{\text{std}_j\, r(x,y_j) + \epsilon}. \tag{2}$$

Here $\epsilon > 0$ stabilizes the denominator. These advantages guide updates toward above-average responses, with proximal policy optimization (PPO) clipping (Schulman et al., 2017) and reference-policy Kullback–Leibler (KL) regularization (Shao et al., 2024). Equal rewards yield zero advantages; in particular, uniformly incorrect answers provide no advantage signal from correctness. Hints can make successful rollouts accessible (Qu et al., 2026), providing additional experience for learning. Within-prompt RL also optimizes reward under the hinted condition,

$$J_{\text{hint}}(\theta) = \mathbb{E}_{x,h}\,\mathbb{E}_{y\sim\pi_\theta(\cdot|x,h)}[r(x,y)], \tag{3}$$

whose updates affect raw-prompt predictions through shared parameters. To make hinted experience directly guide the raw-prompt policy, HiGHT adds a transfer term that selects responses by reward and adjusts their token-level contributions using the hint gap.

3.2 Hint-Gap Weighted Transfer

Selecting responses by reward. Hinted rollouts can differ in reward, so transfer should reflect their quality. To assess their value relative to either prompt condition's experience, we compare rewards across raw and hinted rollouts for the same item. For a hinted rollout $y = (y_1, \ldots, y_T)$, let $\hat{A}(y)$ denote its reward standardized across this combined group. We use the positive part $[\hat{A}(y)]_+ = \max(\hat{A}(y), 0)$ to select responses whose reward exceeds the shared baseline and scale their transfer contribution. Reward thus determines which hinted experience guides the raw policy.

Using the hint gap for token-level guidance. A hint can raise the probability of some tokens and lower that of others, even within a rewarding response. To capture both effects, we compare raw and hinted probabilities for the same sampled token, fixing the model and response prefix to measure the hint-induced prediction change. At an update, define

$$q_t = \pi_{\bar{\theta}}(y_t \mid x, h, y_{<t}), \qquad p_t = \pi_{\bar{\theta}}(y_t \mid x, y_{<t}), \qquad \Delta_t = \max(q_t, \rho) - \max(p_t, \rho), \tag{4}$$

where $\bar{\theta}$ denotes detached current parameters and $\rho \geq 0$ is a numerical probability floor. We call Δ_t the *hint gap*. Its sign and magnitude describe the difference between the floored hinted and raw probabilities. For a reward-selected response, we use a positive gap to encourage the token under the raw prompt and a negative gap to suppress it. The gap supplies relative token guidance; the reward gate determines whether the response contributes.

Constructing signed transfer weights. Transfer coefficients should preserve the gap's sign for selected responses and vanish when either the advantage gate or gap is zero. We therefore use a centered exponential weight, combining response selection and token-level guidance as

$$a_t(y) = [\hat{A}(y)]_+ \, \mathrm{sg}(\Delta_t), \qquad w_t(y) = \exp(\mathrm{clip}(\eta a_t(y), \ell, u)) - 1, \tag{5}$$

where $\eta > 0$ controls the weighting scale, $\ell < 0 < u$, and sg is the stop-gradient operator. Subtracting one gives the desired zero point and permits negative coefficients. Clipping with $\ell = \log(0.9)$ and $u = \log(11)$ keeps coefficients in $[-0.1, 10]$ for numerical stability. We apply this guidance to the policy used without hints by weighting raw-prompt token log-likelihoods along each hinted rollout:

$$\mathcal{J}_{h \to x}(\theta) = \mathbb{E}_{y \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid x, h)} \left[\sum_{t=1}^{T} w_t(y) \log \pi_\theta(y_t \mid x, y_{<t}) \right]. \tag{6}$$

Each coefficient sets the sign and magnitude of its token's contribution to the raw-prompt log-likelihood gradient. Trajectories are sampled from $\pi_{\theta_{\mathrm{old}}}$; all coefficients are held fixed during differentiation, while token probabilities used to form them may be refreshed between optimizer passes. The gradient therefore acts on the raw-prompt log-likelihood in Equation 6.

Combined objective. Transfer learns from responses sampled with additional context, while the deployment objective is successful behavior under raw prompts. We therefore retain raw-prompt RL as a direct task-reward objective alongside transfer and hinted-prompt RL:

$$\mathcal{J}_{\mathrm{total}} = \mathcal{J}_{\mathrm{within}} - \mathcal{R}_{\mathrm{KL}} + \lambda_{h \to x} \mathcal{J}_{h \to x} + \lambda_{x \to h} \mathcal{J}_{x \to h}. \tag{7}$$

Here $\mathcal{J}_{\mathrm{within}}$ contains raw- and hinted-prompt RL terms with prompt-specific baselines and PPO-style clipping, while $\mathcal{R}_{\mathrm{KL}}$ collects reference-policy penalties. The rule applies in either source–target direction, with nonnegative coefficients $\lambda_{h \to x}$ and $\lambda_{x \to h}$ controlling their contributions. We maximize $\mathcal{J}_{\mathrm{total}}$, combining reward, transfer, and reference-policy penalties. Equation 6 uses a token sum; the appendices specify direction-specific coefficients, reduction conventions, and numerical settings.

As training proceeds, fresh rollouts provide new experience and current predictions determine the hint gaps. Reward and hint influence jointly govern transfer: a response with nonpositive advantage contributes zero transfer even when its hinted and raw predictions differ. Section 4.2 reports learning and retention results, including controlled comparisons at matched update norms. Appendix G discusses exponential weighting and bounds the local transfer gradient.

4 Experiments

We evaluate three aspects of transfer: independent task learning and retention, local retention at matched update norms, and utilization of a fixed hint source. Hidden-Operator Continual Learning (HOCL; Section 4.1) evaluates knowledge acquisition and retention across three controlled tasks with Qwen/Qwen3-8B (Yang et al., 2025). MLLM-CL domain continual learning (Zhao et al., 2025) (Section 4.2) provides the main acquisition–retention and controlled local-update evaluations with Qwen/Qwen3-VL-4B-Instruct (Bai et al., 2025). The POPE integration (Section 4.3) evaluates transfer with guide construction held fixed on hard mathematical reasoning. All accuracy evaluations use raw prompts without hints. Appendix C documents the MLLM-CL training, evaluation, and method-specific checkpoint protocols.

Table 1: Three-task HOCL results at the reported stage checkpoints. All metrics use the $[0, 1]$ accuracy scale; BWT is the mean change on earlier tasks. Bold marks the best value, including ties. The full stage-by-task matrix appears in Appendix B.

Method	MFT	MFN	MAA	BWT
SFT	0.985	0.790	0.856	-0.293
Distill-only	0.973	0.977	0.973	**0.005**
Distill + RL	0.995	0.985	0.991	-0.015
HiGHT	**1.000**	**1.000**	**1.000**	0.000

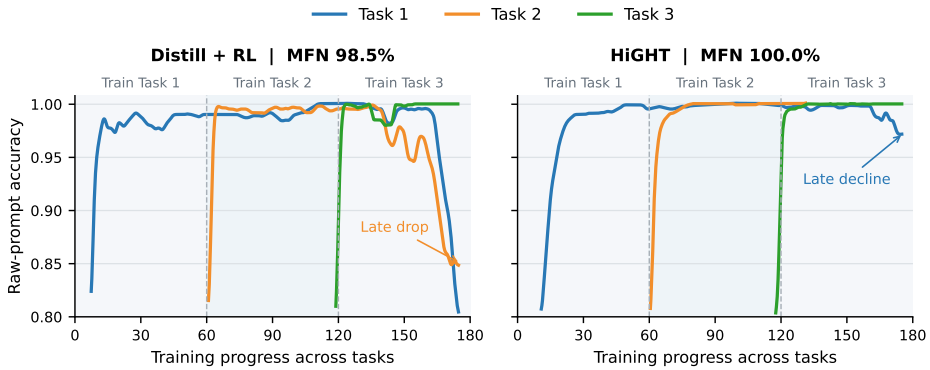

Figure 2: HOCL raw-prompt accuracy during sequential training. Blue, orange, and green denote Tasks 1–3. The panels resolve differences near saturation: HiGHT achieves 100% accuracy on all three tasks at third-stage step 400, whereas Distill + RL averages 98.5% at its corresponding checkpoint. The curves also show subsequent late-stage declines (Appendix B).

4.1 Hidden-Operator Continual Learning

Task. Each HOCL task defines a hidden mapping from symbolic operators to arithmetic operations. Multiple-choice questions use expressions with placeholders such as `operator_2`; the raw prompt requests the answer without revealing the mapping. Training and test questions share the task mapping, requiring the model to internalize this knowledge for raw-prompt success. Training hints reveal the mapping to enable reward-bearing trajectories; evaluation measures raw-prompt multiple-choice accuracy.

Baselines and metrics. Baselines compare uses of the same training information. SFT trains on supervised raw-prompt examples. Distill-only imitates hinted responses without reward-dependent weighting. Distill + RL combines hinted distillation with within-prompt reward optimization but omits the proposed signed transfer rule. We report four metrics: Mean Finetune Accuracy (MFT) averages diagonal performance when each task is first learned; Mean Final Accuracy (MFN) averages performance after the last task; Mean Average Accuracy (MAA) averages seen-task accuracy across stages; and Backward Transfer (BWT) measures the final change on earlier tasks relative to when each was learned.

Result. Table 1 summarizes the three-stage results, with accuracy curves in Figure 2. HiGHT achieves 100% accuracy on all three tasks with zero BWT. In contrast, SFT learns the latest task but loses 59.0 percentage points on Task 1, producing BWT of -0.293. The distillation baselines retain substantially more than SFT but finish below HiGHT: final mean accuracy on the two previous tasks is 97.8% for Distill + RL and 97.3% for Distill-only, versus 100.0% for HiGHT and 69.5% for SFT.

4.2 MLLM-CL Domain Continual Learning

Protocol and baselines. We train sequentially on Remote Sensing (RS), Medical (Med), Autonomous Driving (AD), Science (Sci), and Finance (Fin), using 1% of each MLLM-CL training

Table 2: MLLM-CL results under the common 1,024-token evaluation protocol, using 1% training data and seed 42. Blocks separate reference methods, sequential supervised CL baselines, and online objectives. MAA averages seen-task accuracy equally across stages and is omitted for references without a sequential trajectory. Bold marks the strongest online result.

Method	RS	Med	AD	Sci	Fin	MFT	MFN	MAA	BWT
Zero-shot	50.1	35.0	20.7	48.6	32.2	–	–	–	–
Cumulative LoRA	63.8	38.2	37.5	51.5	60.4	51.00	50.28	52.54	-0.90
Joint LoRA	62.9	38.9	40.5	53.1	58.9	50.86	50.86	–	–
Oracle LoRA	63.8	38.2	40.7	59.0	58.8	52.10	52.10	–	–
MR-LoRA	63.8	38.2	40.7	59.0	58.8	52.10	52.10	–	–
Sequential LoRA (Hu et al., 2021)	55.1	35.4	33.3	50.9	60.1	52.38	46.96	51.36	-6.78
LoRA + replay	56.2	37.3	33.3	51.4	62.4	52.66	48.12	51.54	-5.68
MoELoRA (Luo et al., 2024)	56.6	36.0	31.4	51.2	60.5	51.82	47.14	51.18	-5.85
MoELoRA + replay	56.7	36.3	30.4	51.2	61.5	51.60	47.22	51.07	-5.48
O-LoRA (Wang et al., 2023)	56.3	35.6	32.8	51.4	62.2	51.78	47.66	51.20	-5.15
O-LoRA + replay	55.7	36.5	33.1	50.8	59.8	50.94	47.18	50.75	-4.70
Sequential Full FT	58.0	36.5	31.0	50.4	57.4	48.96	46.66	50.28	-2.88
Full FT + replay	59.8	37.5	29.6	50.6	55.8	49.12	46.66	50.50	-3.08
Online Distillation	51.3	38.0	33.6	46.8	41.9	48.20	42.32	50.45	-7.35
Online Rejection SFT	40.2	29.4	26.4	35.5	76.4	53.80	41.58	51.22	-15.28
Distill + RL	58.0	38.8	30.1	51.6	71.2	54.76	49.94	52.97	-6.03
HiGHT	**59.6**	**38.9**	**33.9**	**61.4**	**81.8**	**57.66**	**55.12**	**54.28**	**-3.18**

split. All methods share the evaluation prompts, answer checker, and seed-42 subset of 1,000 official test examples per domain, using raw prompts, greedy decoding, and a 1,024-token limit. Offline baselines train on correctness-filtered hinted reasoning records, whereas online methods generate fresh trajectories. We compare full-model supervised fine-tuning (Full FT), adapter-based CL methods, Online Rejection SFT, reward-filtered Online Distillation, Distill + RL, and HiGHT. We report each training stage's final HiGHT checkpoint. Method-specific training configurations and checkpoint protocols appear in Appendix C.

Metrics. We use HOCL's MFT, MFN, MAA, and BWT definitions, with accuracy in percentages and BWT in percentage points (pp). MAA averages seen-task accuracy within each stage, then across stages. Table 2 reports final-stage domain scores; Appendix D provides both central online objectives' complete post-task trajectories.

Learning and retention. Among the online objectives, HiGHT obtains the strongest aggregate scores. Relative to Distill + RL, MFT rises from 54.76 to 57.66, MFN from 49.94 to 55.12, and MAA from 52.97 to 54.28, while BWT improves from -6.03 to -3.18 pp. Final accuracy is higher on all five domains, with the largest gains on Sci (+9.8 pp) and Fin (+10.6 pp). Full FT reaches MFN 46.66 with or without the configured minimal replay. Our primary comparison is with Distill + RL, which shares the online training setup; Appendix C gives the method-specific training settings.

Training dynamics and local interventions. Figure 3 traces Fin adaptation from each method's own post-Sci checkpoint. At step 100, HiGHT gains 29.40 pp on Fin while losing an average of 1.25 pp across the four old domains; Distill + RL gains 23.50 pp and loses 3.35 pp. To compare objective-induced update directions, we perform local updates across 30 distinct rollout seeds per shared checkpoint, sharing questions, rollouts, evaluation data, and target step norm 0.05 between methods. The mean early-stage old-domain retention difference is +0.436 pp in favor of HiGHT, with a 90% confidence interval (CI) of [0.202, 0.674] pp. At matched update norms, HiGHT thus preserves old-domain accuracy better during early adaptation. Figure 3(d) shows all four phase–metric mean differences with 90% CIs; Appendix E details the aggregation.

Local effects across checkpoints. Figure 4 compares HiGHT, Distill + RL, and the distillation component at shared Fin checkpoints 0/10/20/30. At step 30, HiGHT averages +2.403 pp on Fin and

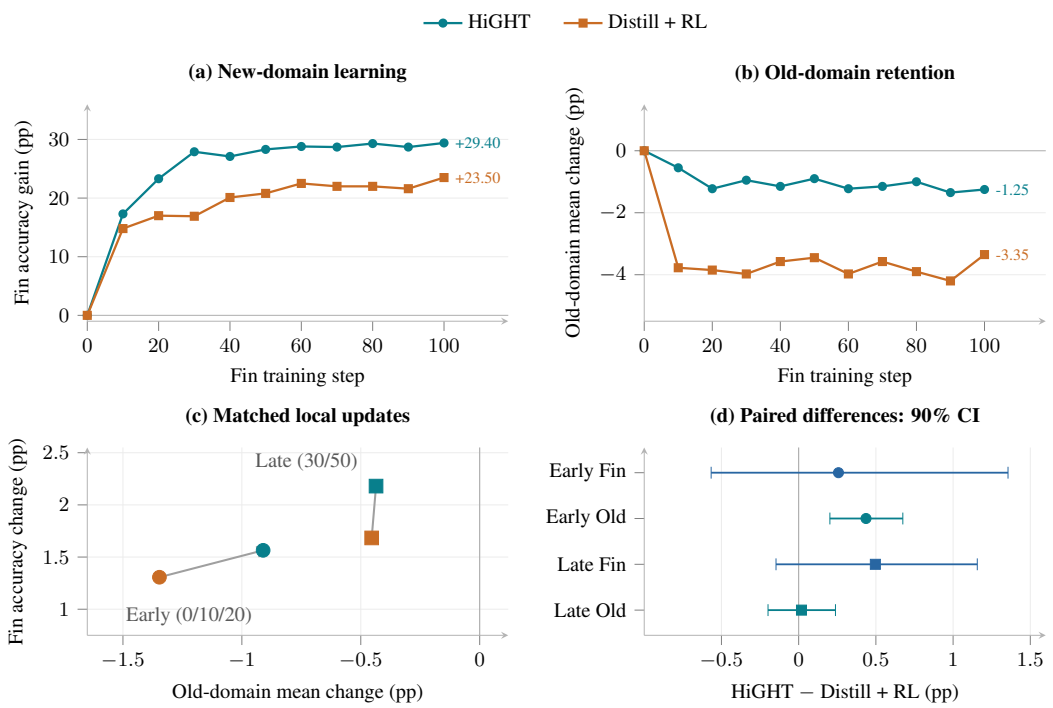

Figure 3: Learning–retention dynamics on the final MLLM-CL domain. (a–b) Accuracy changes along one training trajectory per method, relative to its own post-Sci checkpoint. (c) Mean effects of controlled local updates across 30 rollout seeds per checkpoint. Arrows connect Distill + RL to HiGHT; circles denote early steps 0/10/20 and squares denote late steps 30/50. (d) Paired mean differences with pointwise 90% bootstrap CIs over 30 batches, after averaging checkpoints within each batch. Intervals quantify uncertainty in the mean effects at the fixed evaluation set.

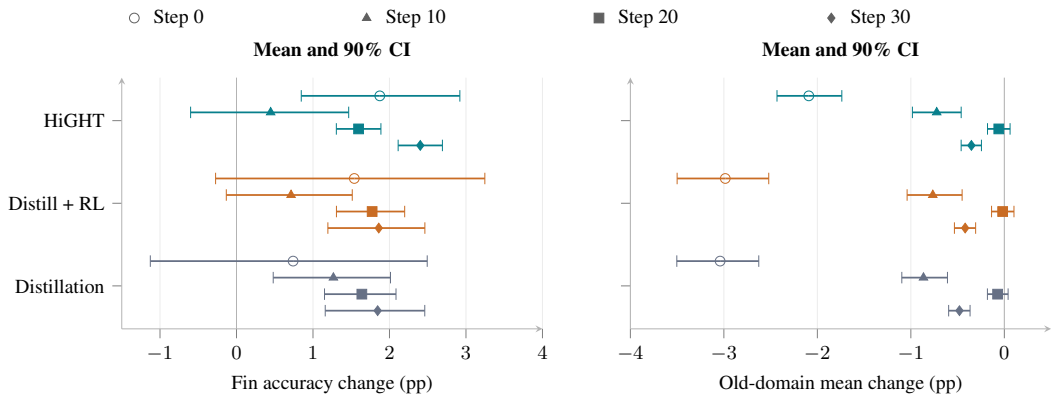

Figure 4: Local learning and retention at Fin checkpoints 0/10/20/30. Rows compare full HiGHT, full Distill + RL, and distillation updates, each normalized to target step norm 0.05. Points show means; bars show pointwise 90% bootstrap CIs over clusters of three batches. Steps 0/10 use 30 four-question batches each; steps 20/30 use 150 each, covering all 600 Fin examples. Each batch has a distinct rollout seed. Objectives share questions, rollouts, and evaluation data within each checkpoint.

-0.352 pp on old domains, compared with +1.860 pp and -0.419 pp for Distill + RL. Mean differences vary across checkpoints; error bars show 90% CIs for each mean. Appendix E gives the protocol and sample counts.

Table 3: POPE Hard raw-prompt pass@1 under the common 16,384-token evaluation protocol. Trained rows are means over seeds 42/43/44 at step 200. POPE (our implementation) shares the rollout and learner structure of POPE + HiGHT, with cross-prompt transfer set to zero.

Method	Training signal	pass@1 (%)
Base model	None	0.54
Original POPE	POPE guides	3.79
POPE (our implementation)	Raw/hinted RL	4.64
POPE + HiGHT	Raw/hinted RL + transfer	**8.48**

4.3 Compatibility with Privileged On-Policy Exploration

Setup. To evaluate transfer beyond domain continual learning, we add HiGHT to the released POPE framework (Qu et al., 2026) for hard mathematical reasoning, keeping its guides fixed as training-time hints. We compare guide utilization after 200 training steps under a common evaluation token cap. Using Qwen/Qwen3-4B-Instruct-2507 and the 256-problem POPE Hard set, we compare Original POPE, our implementation of POPE, and POPE + HiGHT. Our implementation of POPE uses the same raw/hinted rollout groups and counterfactual learner traces as POPE + HiGHT, with cross-prompt transfer set to zero. This isolates the transfer rule's contribution within our implementation; training and evaluation settings appear in Appendix F.

Guide utilization at matched training steps. At step 200, POPE + HiGHT improves pass@1 by 4.69 ± 0.96 pp over Original POPE and by 3.84 ± 1.60 pp over our implementation of POPE (mean $\pm$ sample standard deviation of paired differences across three training seeds), winning all three seeds. Pass@8 also rises from 12.89% for both POPE baselines to 23.44% for POPE + HiGHT. In the matched mechanism diagnostic, HiGHT reaches 50% and 90% raw-hard success at steps 35 and 41, respectively, compared with steps 50 and 60 for our implementation of POPE. With fixed guides, these results show higher raw-prompt performance at matched training steps and earlier success in the mechanism diagnostic. The Hard set supports both method development and this guide-utilization evaluation. Appendix F reports per-seed results, scoring robustness, and implementation details.

5 Limitations

Computational cost is a shared challenge for RL-based continual post-training. Online RL repeatedly generates and scores rollouts in addition to updating model parameters, making training substantially slower than SFT on fixed examples in our setting. We use a common 1% training subset to accommodate these costs for both the RL baselines and HiGHT within our computational budget. More efficient rollout generation and larger-budget studies are therefore priorities for this class of methods.

Our approach also relies on informative training-time hints, motivating further study of noisy or automatically generated hints. Broader model and domain coverage, together with independently held-out reasoning evaluations, would extend the evidence beyond the settings studied here.

6 Conclusion

Learning from hints requires turning assisted success into capabilities that remain useful without the hint. We presented HiGHT, which selects hinted experience through positive trajectory advantages and uses the signed hint gap to modulate token-level transfer alongside online RL. Controlled local updates show better early-stage retention than Distill + RL at matched update norms. Integrating HiGHT into POPE improves raw-prompt performance over our implementation of POPE with guide construction held fixed. Across five real-world vision-language domains, HiGHT improves new-task learning, final performance, and backward transfer over Distill + RL, acquiring new capabilities while retaining more of the previously learned ones.

AI USE STATEMENT

We used `gpt5.6-sol` and `gpt6` to refine terminology and wording and to check the consistency of terminology throughout the manuscript. An AI assistant also cross-checked reported metrics against archived evaluation records and helped prepare figures and captions. We also used `gpt5.6-sol` to search for related work and assist with reading the literature, to draft implementation details by summarizing our code, and to help refine mathematical proofs. An AI assistant additionally helped retrieve the ICLR 2027 submission requirements, migrate the LaTeX template, and draft accompanying statements.

We manually checked the mathematical proofs step by step, checked the implementation details against the code, and verified the related-work discussions and citations against the original sources. We also reviewed the AI-assisted wording and terminology changes. We used no AI tools for purposes beyond those disclosed above. The authors take responsibility for the final content of this work, including all AI-assisted text and mathematical claims.

REPRODUCIBILITY STATEMENT

Appendix A describes the models and data used in the experiments. Appendix B gives the metric definitions, HOCL stage results, and reported checkpoints. Appendix C documents the MLLM-CL training and evaluation settings. Appendix E specifies the shared-checkpoint diagnostics, paired sampling, and confidence-interval calculations. Appendix F provides details of the POPE integration and per-seed results. The assumptions and proof of Proposition 1 are given in Appendix G.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

ZhiYan Hou, Xinyu Tang, Hongyan An, Jianjin Zhang, Weizhen Wang, Yunyun Han, Gengsheng Li, Xiangzhao Hao, Haiyun Guo, Wenbin Hu, et al. Dash: Divergence-adaptive supervision horizons for on-policy self-distillation of reasoning models. *arXiv preprint arXiv:2608.06243*, 2026.

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

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Yibei Liu, Jiajun Chen, Qianle Zhang, Tangyue Jin, Mengying Zhu, Meng Xi, and Yangyang Wu. Taming the implicit: Dual-channel risk-aware reinforcement fine-tuning for continual multimodal post-training. *arXiv preprint arXiv:2608.03660*, 2026.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. volume 30, 2017.

Mao-Lin Luo, Zhe-Xu Wang, Zi-Hao Zhou, Bo Ye, Jian Zhao, Min-Ling Zhang, and Tong Wei. Rl forgets! towards continual policy optimization. *arXiv preprint arXiv:2607.04364*, 2026.

Tongxu Luo, Jiahe Lei, Fangyu Lei, Weihao Liu, Shizhu He, Jun Zhao, and Kang Liu. Moelora: Contrastive learning guided mixture of experts on parameter-efficient fine-tuning for large language models. *arXiv preprint arXiv:2402.12851*, 2024.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. Pope: Learning to reason on hard problems via privileged on-policy exploration. *arXiv preprint arXiv:2601.18779*, 2026.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

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

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.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026a.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. 2026:59839–59864, 2026b.

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Meng Wang, Haohan Zhao, Wenzhuo Liu, Lu Yang, Geng Liu, Haiyang Guo, Guo-Sen Xie, Gaofeng Meng, Hongbin Liu, and Fei Zhu. Denser $\neq$ better: Limits of on-policy self-distillation for continual post-training. *arXiv preprint arXiv:2607.01763*, 2026a.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Xinyi Wang, Jinyi Han, Zishang Jiang, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Ma Shuguang, Fei Yu, Yanghua Xiao, et al. Don't tell the answer, truly guide the reasoning during rl rollouts. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3437–3455, 2026b.

Yu Xia, Canwen Xu, Zhewei Yao, Julian McAuley, and Yuxiong He. Learning to hint for reinforcement learning. *arXiv preprint arXiv:2604.00698*, 2026.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

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.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. pp. 3987–3995, 2017.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025.

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. *arXiv preprint arXiv:2601.18734*, 2026.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

A Models and Data

This appendix lists the external models and datasets used in the experiments.

- **Qwen/Qwen3-8B** is used for HOCL. The model is released under the Apache-2.0 license and is cited as Qwen3 (Yang et al., 2025).
- **Qwen/Qwen3-VL-4B-Instruct** is the base multimodal model for the MLLM-CL study. The model is released under the Apache-2.0 license and is cited as Qwen3-VL (Bai et al., 2025).
- **MLLM-CL** is the public benchmark used for the domain sequence Remote Sensing (RS), Medical (Med), Autonomous Driving (AD), Science (Sci), and Finance (Fin). We use 1% of each training split and a fixed subset of each official test split. The benchmark is released under the Apache-2.0 license (Zhao et al., 2025).
- **Qwen/Qwen3-4B-Instruct-2507 and POPE Hard** are used for the guide-utilization study. POPE Hard contains 256 problems with fixed privileged guides (Qu et al., 2026).
- **HOCL data** are generated for the controlled hidden-operator experiment. Each of the three tasks defines a mapping from four symbolic operators to arithmetic operations.

B HOCL Stage Results

Table 4 reports the complete matrix underlying the aggregate results in Table 1. Rows indicate the most recently trained task, and columns show raw-prompt accuracy on each evaluation task.

Reported checkpoints and curves. The three-task study uses the archived stage checkpoints. For HiGHT, Distill + RL, and Distill-only, the first two stages are evaluated at step 599 and the third at the fixed reporting step 400. SFT uses its completed stage checkpoints at steps 197, 196, and 193. The 100% HOCL result refers to the reported third-stage checkpoint. The archive also contains later step-599 evaluations: HiGHT obtains 90.5%, 100%, and 100% on Tasks 1–3, while Distill + RL obtains 70%, 89%, and 100%. Figure 2 redraws the archived learning-curve image by digitizing its traces; the table values come from the numerical evaluation records.

Metric definitions. For K sequential tasks, let $a_{i,j}$ be accuracy on task j at the reported checkpoint after learning task i. Both HOCL and MLLM-CL use

$$\mathrm{MFT} = \frac{1}{K}\sum_{i=1}^{K} a_{i,i}, \qquad \mathrm{MFN} = \frac{1}{K}\sum_{j=1}^{K} a_{K,j},$$

$$\mathrm{MAA} = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{i}\sum_{j=1}^{i} a_{i,j}, \quad \mathrm{BWT} = \frac{1}{K-1}\sum_{j=1}^{K-1}(a_{K,j} - a_{j,j}).$$

HOCL reports fractions; MLLM-CL reports accuracies as percentages and BWT in percentage points. MAA weights training stages equally.

C MLLM-CL Implementation and Reproducibility

Data and evaluation. Training follows RS $\rightarrow$ Med $\rightarrow$ AD $\rightarrow$ Sci $\rightarrow$ Fin, starting from Qwen/Qwen3-VL-4B-Instruct and continuing each stage from the preceding checkpoint. Each domain uses a seed-42 1% training subset. All table rows use the same prompt, answer checker, and seed-42 subset of 1,000 official test examples per domain, with greedy decoding and a 1,024-token generation limit. The implementation snapshot is archived in `revision/RL4CL_revised_table2_20260911/code/awhr_may_source_snapshot`.

HiGHT training and objective. Each domain is trained for ten epochs with learning rate 5×10^{-6}, 50 warmup steps, seed 42, and CUDAAdamW on direct BF16 parameters and states without FP32 master weights. Each rollout step uses 16 examples, with eight raw and eight reveal-answer hinted responses per example. Policy updates use response minibatches of eight and 16 update passes per

Table 4: HOCL raw-prompt accuracy after each sequential training stage.

Method	Stage	Task 1	Task 2	Task 3
SFT	Task 1	0.985	0.265	0.345
	Task 2	0.595	0.990	0.400
	Task 3	0.395	0.995	0.980
Distill-only	Task 1	0.975	0.280	0.395
	Task 2	0.975	0.960	0.310
	Task 3	0.965	0.980	0.985
Distill + RL	Task 1	0.990	0.275	0.375
	Task 2	1.000	0.995	0.360
	Task 3	0.990	0.965	1.000
HiGHT	Task 1	1.000	0.310	0.345
	Task 2	1.000	1.000	0.310
	Task 3	1.000	1.000	1.000

rollout step. Sampling temperature is 0.9, and the training generation limit is 300 tokens, distinct from the 1,024-token evaluation limit. Checkpoints are saved every 100 outer steps. Raw- and hinted-prompt RL terms have unit weights; bidirectional inter-prompt transfer has weight 0.3 and score scale $\eta = 50$. PPO clipping is 0.28. Raw, hinted, and inter-prompt reference penalties each have coefficient 0.04 relative to the stage-initial policy. The loss first averages masked tokens within each response and then averages responses and prompt-mixture components. The archived implementation uses probability floor $\rho = 10^{-3}$ in Equation 4 and coefficient range $[-0.1, 10]$ in Equation 5.

Rewards parse the final `<answer>` tag and use the domain-specific answer matcher, together with auxiliary rewards for format, length, and avoiding explicit hint leakage. Cross-prompt advantages are standardized across the 16 responses; within-prompt RL subtracts the corresponding prompt-group mean without dividing by its standard deviation. Positive cross-prompt advantage therefore selects responses whose composite reward exceeds the sampled group baseline.

Reported checkpoints and supervised baselines. The HiGHT results use the final checkpoint of each recovered training stage, ending at Fin step 374. The five post-task evaluations use the common official-test protocol above. Distill + RL uses the recovered checkpoint chain documented in the experiment archive. Baseline hyperparameter and checkpoint selection uses the frozen 200-example-per-domain validation set. Validation access to old domains is distinct from replaying old-domain training examples.

Offline methods use correctness-filtered reasoning records. Sequential Full FT uses a validation-selected learning rate of 10^{-5} for ten epochs; effective per-domain record counts are 597/212/426/187/493. Replay adds one sampled record from each previous domain before correctness filtering, yielding a minimal-memory baseline. Online Rejection SFT trains on reward-selected online trajectories. Online Distillation filters hinted rollouts at correctness reward ≥ 0.7 and uses hinted-only transfer with validation-selected early stopping: Sci uses learning rate 2.5×10^{-6} at step 5, while Fin uses 7.5×10^{-7} at step 5 with 75% hint sampling.

D MLLM-CL Stage Trajectories

Table 5 reports the full post-task trajectories for Distill + RL and HiGHT. Dashes indicate domains that have not yet been introduced at a given stage.

E Training Dynamics and Local-Update Diagnostics

Dense trajectories. Each method starts from its own post-Sci checkpoint and trains independently on Fin, saving every ten steps through step 100 without sharing optimizer states. Each checkpoint is evaluated on 1,000 fixed test items per domain. We subtract each method's own step-0 accuracy, thereby isolating change during Fin adaptation without conflating it with absolute pre-Fin performance.

Table 5: Post-task trajectories under the common 1,024-token evaluator. Percentages; stage names indicate the most recently trained domain.

Method	Stage	RS	Med	AD	Sci	Fin
Distill + RL	RS	67.6	–	–	–	–
	Med	65.7	39.6	–	–	–
	AD	61.5	39.9	36.7	–	–
	Sci	60.8	39.6	35.4	58.7	–
	Fin	58.0	38.8	30.1	51.6	71.2
HiGHT	RS	67.6	–	–	–	–
	Med	65.0	39.8	–	–	–
	AD	64.0	36.4	37.7	–	–
	Sci	64.5	37.3	37.8	61.4	–
	Fin	59.6	38.9	33.9	61.4	81.8

Figure 3(a–b) shows the observed accuracy changes along one training trajectory per method. These dense trajectories are separate runs constructed to resolve the learning curve; their step-100 Fin accuracies are 82.5 for HiGHT and 73.0 for Distill + RL.

Shared-state local updates. The paired diagnostic uses Distill + RL checkpoints at Fin steps 0/10/20/30/50/100. At each checkpoint, 30 batches of four Fin questions use 30 distinct rollout seeds, providing identical raw/hinted rollouts to both objectives. These 120 question draws contain 106 unique questions. Each update has target parameter-step norm 0.05, and evaluation uses 500 fixed examples per domain. Early averages checkpoints 0/10/20 within each batch and late averages 30/50; the archive also retains step 100. The early/late definitions follow the independently observed dense trajectory. Batch b uses rollout seed $42 + 10{,}000{,}019b$; its questions and seed remain paired across checkpoints. The parent training trajectories use seed 42, while the local experiment varies the rollout-generation seed. Figure 3(d) reports mean paired effects and 90% CIs from resampling the 30 paired batches, conditional on these checkpoints and the fixed evaluation set.

Checkpoint-wise objective diagnostic. Figure 4 draws on the separate six-objective diagnostic and displays the full HiGHT objective, full Distill + RL objective, and distillation component. Steps 0 and 10 each use 30 four-question batches with distinct rollout seeds. Steps 20 and 30 each cover all 600 Fin training examples exactly once in 150 disjoint four-question batches with 150 distinct rollout seeds. Within each checkpoint, objectives reuse the same rollouts and are separately normalized to target norm 0.05. Evaluation uses 500 fixed examples per domain. Each marker is the mean across batches. Following the archived cluster-bootstrap design, 90% CIs resample clusters of three batches (shards): ten shards at steps 0/10 and fifty at steps 20/30. Equal shard sizes preserve the batch-level mean. These counts describe rollout-generation repeats at fixed checkpoints. In the full-600 runs, FP16 evaluation realizes norms about 0.046–0.047 and realized-to-ideal update cosines about 0.86–0.88.

Mean effects and intervals. For each metric, let $\Delta_{m,b}$ denote the local accuracy change for method m on batch b. Phase summaries first average checkpoints within each batch, then compute each method's mean and the mean paired difference $\bar{d}$, where $d_b = \Delta_{HiGHT,b} - \Delta_{\text{Distill+RL},b}$. All local-update CIs use 20,000 bootstrap resamples (seed 42), taking the 5th and 95th percentiles of resampled means. The same resampled units are used for both methods to preserve pairing. These pointwise 90% intervals quantify uncertainty in mean effects conditional on the checkpoints and fixed evaluation set. Table 6 reports all four phase–metric summaries. The script `scripts/plot_ci90_diagnostics.py` reproduces the figures and summaries; `revision/ci90_figures_20260918/` contains the numerical values and source checksums.

F POPE Extension: Protocol and Audits

Protocol and development use. The base model is Qwen/Qwen3-4B-Instruct-2507. Training uses the released POPE Hard pairs of raw and hinted prompts, with guide construction fixed. Our implementation of POPE and POPE + HiGHT sample eight raw and eight hinted responses per

Table 6: Local accuracy changes in percentage points: means [pointwise 90% bootstrap CIs] over 30 paired batches. Checkpoints are averaged within each batch before resampling. The final column summarizes within-batch differences (HiGHT − Distill + RL).

Setting	Metric	HiGHT	Distill + RL	Paired difference
Early (0/10/20)	Fin	$+1.56$ $[+0.95, +2.18]$	$+1.31$ $[-0.00, +2.32]$	$+0.26$ $[-0.57, +1.36]$
Early (0/10/20)	Old	-0.91 $[-1.14, -0.68]$	-1.35 $[-1.61, -1.09]$	$+0.44$ $[+0.20, +0.67]$
Late (30/50)	Fin	$+2.18$ $[+1.59, +2.79]$	$+1.68$ $[+0.94, +2.40]$	$+0.50$ $[-0.15, +1.16]$
Late (30/50)	Old	-0.44 $[-0.63, -0.24]$	-0.45 $[-0.68, -0.23]$	$+0.02$ $[-0.20, +0.24]$

group. The learning rate is 10^{-6}, gradient clipping is 0.3, and generation uses temperature 0.8 with a 16,384-token limit and 20,000-token training sequences without packing. Transfer uses $\eta = 5$, weight 0.03, floor 10^{-5}, coefficient range $[-0.1, 10]$, and hint-to-raw direction only. Both within-prompt and transfer contributions use our implementation's scale factor 2.0; the primary runs use entropy bonus 10^{-4} and token-sum reduction.

The frozen raw-Hard evaluation uses eight samples per problem, temperature 0.8, top-p 0.95, top-k 50, generation seed 42, and a 16,384-token cap. Hyperparameter screening required improvement over our implementation of POPE together with finite gradients and limited clamp saturation. Step 200 was then fixed for the three-seed comparison. The same Hard problems support development and evaluation, making this a study of guide utilization at matched training steps on the development problem set. Reported dispersion is the sample standard deviation of paired pass@1 differences across training seeds 42/43/44.

Table 7: Per-seed POPE Hard step-200 diagnostic. Cells report pass@1 / pass@8 percentages.

Seed	Original POPE	POPE (our implementation)	POPE + HiGHT
42	5.03 / 16.41	6.54 / 15.62	8.69 / 24.22
43	2.64 / 10.55	4.15 / 12.50	8.20 / 24.22
44	3.71 / 11.72	3.22 / 10.55	8.54 / 21.88

Scoring and output-length audit. The official verifier grades the final boxed answer. As a robustness check, an option-aware audit grades the substantive boxed expression preceding any trailing standalone A–E option box; official results use the released verifier. Under official scoring, the mean HiGHT improvement over our implementation of POPE is +3.84 pp; under option-aware scoring, it is +1.89 pp. The corresponding mean gains over Original POPE are +4.69 pp and +2.16 pp. At seed 42, 912 of 2,048 HiGHT responses approach the 16,384-token limit, compared with 139 for Original POPE and 162 for our implementation of POPE. The accuracy gains therefore accompany longer generated responses, with method-dependent inference costs within the common token cap.

Matched mechanism diagnostic. The three-condition diagnostic fixes the hard problem and its guide and uses zero entropy bonus for every method. Original POPE reaches 50%/90% trailing raw-hard success at steps 50/62, our implementation of POPE at 50/60, and HiGHT at 35/41. All three methods eventually solve the problem. This controlled comparison measures how quickly each update converts the same guide into raw-prompt success under the matched settings.

G Exponential Weighting and Transfer-Gradient Bound

Connection to reward-weighted policy updates. At the sequence level, Equation 1 treats a complete response as an action and its evaluated reward as the return. For a fixed reference distribution π_0 and inverse temperature $\eta > 0$, the idealized regularized policy objective

$$\max_{\pi} \ \mathbb{E}_{y\sim\pi}[\hat{A}(y)] - \eta^{-1} D_{\mathrm{KL}}(\pi \| \pi_0) \tag{8}$$

has solution $\pi^*(y) = \pi_0(y)\exp(\eta\hat{A}(y))/Z$, provided the normalizer $Z = \mathbb{E}_{y\sim\pi_0}[\exp(\eta\hat{A}(y))]$ is finite. Here $\hat{A}$ is fixed and optimization is over distributions supported on π_0. This exponential

reward reweighting underlies advantage-weighted regression (Peng et al., 2019). Equation 5 uses a centered exponential to combine positive trajectory selection with signed token-level hint gaps. The sequence-level solution above describes the regularized objective; the proposition below characterizes the implemented transfer surrogate at a fixed update.

Local transfer sensitivity. The following proposition characterizes the added gradient at a fixed update under the stated assumptions.

Proposition 1 (Vanishing, bounded transfer)**.** *Fix a prompt pair, rollout distribution, and stopped weights. Assume* $T \le T_{\max}$, $|\Delta_t(y)| \le \epsilon$, *and* $\|\nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})\| \le G$ *uniformly over sampled trajectories and tokens in a neighborhood of the parameters at which the gradient is evaluated. Suppose* $\mathbb{E}[[\hat{A}(y)]_+] < \infty$ *and let* $c = \max(|\ell|, |u|)$. *Then*

$$\|\nabla_\theta \mathcal{J}_{h\to x}\| \le e^c \eta T_{\max} G \epsilon \, \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x,h)}[[\hat{A}(y)]_+]. \tag{9}$$

If every gap is zero, this transfer gradient is zero. With the other quantities uniformly bounded, it is $\mathcal{O}(\epsilon)$ *as the gaps close. The same bound without* $T_{\max}$ *holds for the response-averaged surrogate.*

Proof. Write $z_t = \text{clip}(\eta[\hat{A}(y)]_+\Delta_t, \ell, u)$ and $w_t = e^{z_t} - 1$. The rollout distribution and all coefficients are fixed during differentiation. The uniform score-gradient bound, bounded coefficient, and bounded response length justify differentiation under the expectation, giving

$$\nabla_\theta \mathcal{J}_{h\to x} = \mathbb{E}_{y\sim\pi_{\theta_{\text{old}}}(\cdot|x,h)}\left[\sum_{t=1}^{T} w_t \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})\right].$$

Since the clipping interval contains zero, $|z_t| \le \eta[\hat{A}(y)]_+|\Delta_t|$ and $|z_t| \le c$. The mean-value theorem gives $|e^{z_t} - 1| \le e^c|z_t|$, hence

$$|w_t| \le e^c \eta[\hat{A}(y)]_+\epsilon.$$

The triangle inequality and the uniform bounds $T \le T_{\max}$ and $\|\nabla \log \pi\| \le G$ yield Equation 9. For response averaging, division by T removes $T_{\max}$ from the bound. If every gap is zero, then $z_t = w_t = 0$ for every token. The flooring map is 1-Lipschitz because $|\max(q, \rho) - \max(p, \rho)| \le |q - p|$; convergence of the original probabilities therefore implies convergence of the floored gaps. These statements concern only the transfer gradient; the other terms of Equation 7 remain present. □