Review



# A Survey on Hint-Based RLVR: Overcoming Zero-Advantage Failures with External Textual Signals

Wenyuan Zhang [*,†] , Shuaiyi Nie [†] , Zhengyang Ai [†] , Chengguang Tang , Xinghua Zhang , Yi Liu , Tingwen Liu , Pinyan Lu

Posted Date: 12 June 2026

doi: 10.20944/preprints202606.1050.v1

Keywords: large language model; reinforcement learning; hint-based RL; zero-advantage failures

 doi:10.20944/preprints202606.1050.v1



*Review*

# A Survey on Hint-Based RLVR: Overcoming Zero-Advantage Failures with External Textual Signals

**Wenyuan Zhang [1,*,†], Shuaiyi Nie [1,†], Zhengyang Ai [2,†], Chengguang Tang [3], Xinghua Zhang [1], Yi Liu[3], Tingwen Liu [1] and Pinyan Lu [2,4]**

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] Huawei Taylor Lab, China
[3] Tencent, Shenzhen, China
[4] Shanghai University of Finance and Economics, Shanghai, China
* Correspondence: zhangwenyuan@iie.ac.cn
† These authors contributed equally to this work.

**Abstract**

Reinforcement Learning with Verifiable Rewards (RLVR) has become a central paradigm for post-training large language models, yet group-relative methods often suffer from zero-advantage failures, where identical rollout rewards erase the policy-gradient signal. A growing body of work addresses this bottleneck by intervening in rollout-group construction to restore learnable contrasts. Among these efforts, methods that introduce external textual signals beyond the model's own distribution, such as reference trajectories, abstract scaffolds, and reusable experience, have emerged as a key branch, as they can restore learnable contrasts while expanding the model's capability boundary. This survey provides the first systematic survey of this branch: we introduce ***Hint*** as a unifying concept for such external textual signals and organize hint-based RL methods into sample-level hints, covering trajectory-based and scaffold-based guidance, and task-level hints, covering static and evolving experience bases. Beyond taxonomy, we further clarify the boundaries, cross-level analysis of construction and utilization, and future directions. We maintain an up-to-date resource list at: https://github.com/WYRipple/Awesome-Hint-Based-RL.

**Keywords:** large language model; reinforcement learning; hint-based RL; zero-advantage failures

## 1. Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) Wen et al. (2026) has emerged as a central paradigm for post-training large language models (LLMs) Xu et al. (2025), with group-relative advantage methods such as Group Relative Policy Optimization (GRPO) Shao et al. (2024) achieving strong results on mathematical reasoning Zhang et al. (2025), code generation Wang et al. (2025), and agentic reasoning Zhang et al. (2026). Despite their simplicity and effectiveness, these methods rely on a stringent condition: sampled rollouts within each training query must exhibit sufficient reward variation to define a meaningful update direction. Otherwise, when all rollouts in a group receive identical rewards, the advantage collapses to zero and the policy-gradient signal vanishes Liu et al. (2025); Qu et al. (2026). Consequently, zero-advantage groups incur both computational and data waste: they consume rollout budget without producing gradients, while filtering them squanders valuable verifiable training questions, especially hard questions that are most valuable for learning.

Recent work has approached this bottleneck from two directions. One refines reward signals within sampled trajectories, converting sparse outcome-level rewards into finer-grained feedback Nie et al. (2026); Ai et al. (2026); such process-reward methods have been systematically surveyed elsewhere Zheng et al. (2026). The other line intervenes in rollout-group construction itself to obtain higher-quality responses that restore learnable contrasts. Within this line, methods divide further by how such rollouts are obtained. The first searches more aggressively within the current policy's own

 doi:10.20944/preprints202606.1050.v1



distribution, using methods such as resampling Yu et al. (2025) or tree-structured search Li et al. (2025). The second introduces external information beyond the model's own distribution, such as reference trajectories, abstract knowledge, or accumulated experience. The former can only sharpen the policy's existing distribution, whereas the latter expands it, directly addressing a well-recognized ceiling of on-policy RLVR Yue et al. (2025). Therefore, recent RLVR research has increasingly turned to externally guided branch, which is the focus of our survey.

In this work, we use ***Hint*** as a unifying term for these external textual signals that reshape rollout-group construction. We systematically survey hint-based RL methods, examining how such hints are constructed and exploited to restore learnable contrasts on otherwise zero-advantage groups. We organize these methods into two principal branches. **Sample-level hints** are tailored to a single training query, providing precise per-problem guidance that does not transfer across questions. We further divide them into two types: (1) *Trajectory-based guidance* Yan et al. (2026); Huang et al. (2025) supplies complete or partial reference solutions; it provides direct signals on hard problems but constrains exploration. (2) *Scaffold-based guidance* Zhang et al. (2026); Yu et al. (2026) instead supplies higher-level abstractions, constraints, or feedback; it preserves exploratory autonomy, but is harder to construct and may be insufficient to push through the hardest problems. **Task-level hints**, by contrast, are stored in shared bases and retrieved across questions, turning accumulated experience into reusable priors. Among them, *static bases* Lu et al. (2026) are fixed before training, offering stability and predictability, whereas *evolving bases* Xia et al. (2026) are updated during training to adapt to the changing policy.

In summary, our contributions are as follows:

- We present the first survey of hint-based RL, consolidating works that introduce external textual signals into rollout-group construction to overcome zero-advantage failure.
- We propose a taxonomy of hints across two granularity levels, covering sample-level (trajectory-based and scaffold-based) and task-level (static and evolving experience bases).
- We further review domain-specific instantiations and discuss the boundaries, open problems, and future directions of hint-based RL.

## 2. Preliminary

We take GRPO as the primary focus of analysis, since it and its variants serve as the base algorithm for many recent hint-based RL methods. Our discussion also applies to policy-gradient methods whose updates depend on reward contrast among sampled responses.

Given a prompt $x$, the current policy $\pi_\theta$ samples $G$ rollouts $\{y_i\}_{i=1}^{G}$, each scored by a verifier as $r_i = R(x, y_i)$. GRPO uses the group-relative advantage $A_i = (r_i - \mu_G)/(\sigma_G + \epsilon)$, where $\mu_G$ and $\sigma_G$ are the within-group mean and standard deviation of rewards, and $\epsilon$ is a small constant. The advantage $A_i$ then weights each rollout's token-level contribution to the clipped policy-gradient update. When all rewards in a group are identical, $\sigma_G = 0$ and all advantages collapse to zero, causing the policy-gradient signal to vanish. This *zero-advantage failure* typically manifests as all-failure groups on hard samples or all-success groups on saturated ones. In both cases the sample yields no effective update.

We define *Hint-based RL* as a class of post-training methods that guide RL training with *hints*, explicit textual signals carrying task-relevant information beyond what ordinary rollouts under the current policy can reliably produce. By supplying this, hints make otherwise uninformative samples more likely to yield useful updates. Figure A1 presents our taxonomy. Sample-level hints provide guidance for individual training samples, while task-level hints maintain reusable guidance across samples through persistent experience repositories. The following sections elaborate these two branches.

## 3. Sample-Level Hints

*Sample-level hints* are the foundation of Hint-based RL. Each hint is exclusive to a single training sample and not shared across samples. Figure 1 illustrates how sample-level hints are constructed and utilized. Hints can originate from offline human annotations, verifiable text, or text selected and processed from the policy or teacher model during training. Their utilization falls into four categories,

 doi:10.20944/preprints202606.1050.v1



spanning all stages of the rollout. We further divide hints by content refinement into trajectories and scaffolds, and organize the taxonomy around the basic utilization and construction mechanisms.

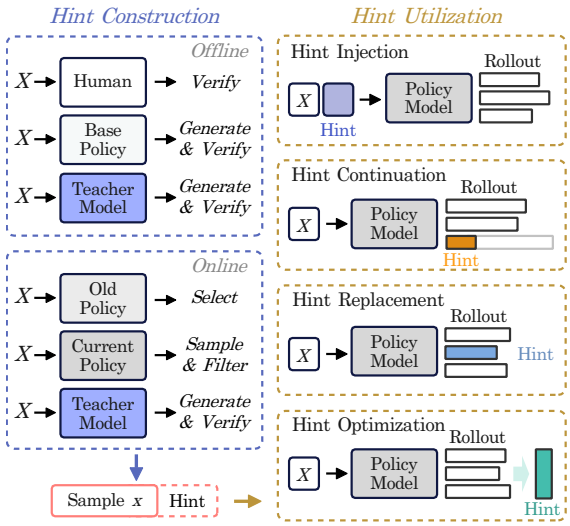


**Figure 1.** Construction and utilization taxonomy of sample-level hints.

### *3.1. Trajectory-Based Hints*

Trajectory-based hints take the form of policy-like responses, typically full solution trajectories or their prefixes. They expose concrete reasoning paths that the current policy may fail to sample reliably, making informative states more reachable and giving low-variance samples new rollout variations. We organize them by where the trajectory signal enters the RL training process.

#### 3.1.1. Trajectory Injection

Trajectory Injection places a trajectory segment in the input context while the policy still generates a full response. We distinguish these methods by the source of the injected segment.

##### Reference Prefix

These methods use early steps from verified reference trajectories to reduce the effective difficulty of hard training queries. The main design choice is how to calibrate the prefix length so that the augmented query remains challenging enough to yield a useful learning signal. QuestA Li et al. (2026), POPE Qu et al. (2026), CCL Wu et al. (2025), and SEELE Li et al. (2025) calibrate offline through pass-rate filtering, shortest-useful-prefix search, threshold-based prefix growth, and an accuracy-conditioned hint model targeting half-correct groups, respectively. GHPO Liu et al. (2025) and BREAD Zhang et al. (2026) calibrate online, triggering reference-prefix resampling when the original rollout group has no or too few successful responses; GHPO uses pass-rate-proportional checkpoints, while BREAD locates episode-level truncation points.

##### Self-Replay Prefix

These methods construct prefixes from the policy's own current or historical rollouts, keeping the hint close to the policy distribution. Most methods reuse successful trajectories to bring difficult samples back into a learnable range. HiPO Qiyuan et al. (2026) draws successful prefixes from the current batch to replace zero-variance groups with hinted groups, PROS Huang and Wan (2026) stores historical correct rollouts and truncates them at uncertain positions to expand the training pool, and RPO Yi et al. (2026) maintains per-sample historical responses as a refreshable prefix cache. Failure-Prefix Kim et al. (2026) takes the opposite direction, extracting rare incorrect prefixes from saturated samples to make over-easy queries informative again.

 doi:10.20944/preprints202606.1050.v1



Hybrid Prefix

These methods lie between the previous two sources, using online trajectories from another policy solving the same query. For example, CORE Mishra et al. (2026) lets multiple policies sample independently, extracts answer-stripped reasoning context from a successful policy, and provides it to failed policies for resampling.

#### 3.1.2. Trajectory Continuation

Trajectory Continuation uses a trajectory prefix as a fixed generation context and trains the policy to complete the suffix. The main design variable is how much prefix to reveal during training.

Scheduled Prefix Decay

These methods decay the given prefix under a preset curriculum, moving from strong continuation support toward prefix-free RL. UFT Liu et al. (2026a) and Prefix-RFT Huang et al. (2025) use cosine decay, with Prefix-RFT keeping most rollouts hint-free. EvoCoT Liu et al. (2026b) uses a discrete step curriculum, removing trailing reasoning steps and refreshing the reference with the updated policy.

Adaptive Prefix Control

These methods set prefix length from online estimates of policy competence. G$^2$RPO-A Guo et al. (2025) uses recent reward trends to adjust the next prefix length, while ADHint Zhang et al. (2026) estimates query difficulty from naive rollouts, assigns longer prefixes to harder queries, and reweights advantages by posterior difficulty. Other methods adapt by rollout success under candidate prefixes. Hint-GRPO Huang et al. (2025) increases the hint ratio until a correct group appears, BHA Xie et al. (2026) maintains bucket-level hint ratios over verified teacher trajectories, and TRAPO Su et al. (2026) lengthens the expert prefix within a group when early hint-free rollouts have low average reward.

Intra-group Prefix Mixing

This branch contrasts different prefix strengths within the same rollout group. StepHint Zhang et al. (2025) segments a reference trajectory into reasoning steps, mixes continuations from different step boundaries with hint-free rollouts, and clips negative advantages on failed continuations to protect correct prefixes.

#### 3.1.3. Trajectory Replacement

Trajectory Replacement modifies the rollout group with trajectory-level samples before the RL update. It changes the candidate set by inserting reliable trajectories or repairing failed ones.

Unconditional Augmentation

These methods add a verified reference trajectory to every rollout group as a positive anchor. ANCHOR Liu et al. (2025) and LUFFY Yan et al. (2026) prepare such trajectories offline for each training sample and mix them with on-policy rollouts during training.

Triggered Substitution

These methods substitute trajectories only when the original group is uninformative. AMPO Yuan et al. (2025), S-GRPO Yan et al. (2026), and HAPO Wu et al. (2026) use external reference trajectories under different triggers, replacing failed or low-reward rollouts in all-fail or low-confidence groups. HiPO Qiyuan et al. (2026) instead derives the replacement from the current policy, using successful self-hinted groups to replace zero-advantage groups in the batch.

Reconstructive Repair

These methods repair failed trajectories by keeping the valid prefix and regenerating the suffix. SCOPE Ren et al. (2026) uses a PRM to locate the first erroneous step, preserves the preceding on-policy prefix, and lets a teacher generate the corrective suffix.

 doi:10.20944/preprints202606.1050.v1



#### 3.1.4. Trajectory Optimization

Trajectory Optimization converts trajectory hints into auxiliary update signals. ExPO Zhou et al. (2026) generates a self-explanation trajectory conditioned on the ground-truth answer and uses it as a low-weight SFT target during GRPO training. MENTOR Jiang et al. (2026) samples rollouts from a mixture of the policy and an expert model, lets the expert intervene more heavily at high-uncertainty token positions, and assigns positive advantage only when these rollouts outperform the on-policy baseline.

### *3.2. Scaffold-Based Hints*

Scaffold-based hints are high-level textual guidance expressed as abstractions, constraints, or feedback. Compared with trajectory-based hints, they are better suited for steering how the policy approaches a problem while keeping multiple valid reasoning paths open.

#### 3.2.1. Scaffold Injection

Scaffold Injection places a scaffold in the input prompt as an additional condition for rollout sampling, guiding the policy toward responses that better satisfy the intended target. We divide it into four types according to the guidance form.

##### Answer-Level Scaffold

These methods use the ground-truth answer as input guidance. By conditioning generation on the target answer, they turn open-ended problem solving into answer-conditioned reasoning, making it easier for the policy to sample reasoning paths that support the correct outcome. RAVR Lin et al. (2025) and CoVRL Wen et al. (2026) apply this guidance to each query, whereas HDPO Ding (2026) triggers it only for all-fail rollout groups and retains the correct answer-conditioned trajectories.

##### Solution-Blueprint Scaffold

These methods inject a blueprint to clarify the key idea needed for solving the problem. Some methods precompute the blueprint offline and inject it during training when needed. Guide-GRPO Nath et al. (2025) uses teacher-generated high-level guidance for all-fail groups, AVATAR Kulkarni and Fazli (2026) uses precomputed strategy hints when exploration on a hard sample stagnates, and PieceHint Fang et al. (2026) selects high-value reasoning pieces from reference trajectories and gradually withdraws them during training. Other methods learn or dynamically generate the blueprint. RLAD Qu et al. (2025) and A2D Chen et al. (2026) train scaffold generators to produce abstractions and sub-questions, while SAGE$_{scaf}$[1] Liao et al. (2026) uses the policy model as a self-hinter and increases the hint level until a correct response is sampled.

##### Knowledge-Level Scaffold

These methods introduce problem-specific meta-knowledge needed for solving the problem, such as relevant theorems, minimal concepts, or techniques. Both KnowRL Yu et al. (2026) and NuRL Chen et al. (2026) construct such scaffolds by distilling reference trajectories into compact, non-leaking knowledge cues. KnowRL further filters candidate knowledge points and selects a minimal compatible subset through offline evaluation, while NuRL obtains the cue by compressing an answer-conditioned reference trajectory into abstract knowledge guidance.

##### Format-Level Scaffold

These methods inject formatting specifications before rollout sampling. MeRF Zhang et al. (2026) appends a natural-language reward specification to the prompt, while RuscaRL Zhou et al. (2026) converts high-quality responses into checklist-style rubrics, varies the proportion of injected rubric across rollouts, and gradually reduces the scaffold during training.

[1] We use the subscripts *scaf* and *exp* to distinguish SAGE variants that share the same name.

 doi:10.20944/preprints202606.1050.v1



#### 3.2.2. Scaffold Replacement

Scaffold Replacement uses scaffolds to generate replacement responses for failed trajectories, offering more flexible and diverse forms of intervention than trajectory replacement.

##### Pedagogical Guidance

This branch can be viewed as a replacement-oriented extension of scaffold injection. HINT Wang et al. (2025) uses teacher-generated heuristic hints, Scaf-GRPO Zhang et al. (2026) uses hierarchical hints ranging from knowledge to solution steps, and KEPO Yang et al. (2026) uses answer-conditioned teacher reasoning hints. The hinted trajectories then replace failed rollouts or enter the rollout buffer. HiLL Xia et al. (2026) further trains a hinter to generate concise pedagogical hints from the current failed rollout and reference solution, using the selected hint to replace the original all-fail group with a hinted rollout group.

##### Critique-Driven Refinement

These methods treat the initial rollout as a diagnostic target. The scaffold specifies what went wrong or what remains missing, then conditions a regeneration step whose output replaces the failed response or the erroneous suffix. LTE Tang et al. (2026) instantiates the lightest form, converting wrong final answers from an all-fail group into negative hints that condition the regeneration. RGR-GRPO Bi et al. (2025) and GOLF Huang et al. (2026) use richer feedback, with the former refining the current best rollout according to unmet rubric criteria and the latter aggregating failed attempts with external critiques into group-level feedback. Beyond reshaping feedback, $R^3L$ Shi et al. (2026) makes the repair local, diagnosing the error point and regenerating only the suffix from there. ECHO Li et al. (2026) further learns the critic itself, using the reward gain of refined trajectories to train both the policy and the feedback generator.

##### Action-Level Guidance

This branch moves replacement to the finer-grained action level. InfoFlow Luo et al. (2025) uses pathfinding hints as search queries, replacing a stalled search action with a more informative query so that the agent can continue the rollout from a better retrieved state.

#### 3.2.3. Scaffold Optimization

Scaffold Optimization keeps the policy scaffold-free at inference, converting behavior generated under scaffold conditions into auxiliary training signals for the unguided policy.

##### Auxiliary Supervision

These methods convert scaffold-derived signals into auxiliary targets. MEL Huang et al. (2026) learns verified meta-experience from success–failure contrasts, while ThinkTuning Rrv et al. (2025) injects teacher reflection tokens into selected student rollouts and updates them with advantage-aware shaping. RLTF Song et al. (2026) uses text feedback for critique prediction and feedback-conditioned self-distillation. KEPO Yang et al. (2026) distills only positive-reward trajectories.

##### Distribution Alignment

These methods use scaffold-guided generation to define a distributional target for the unguided policy. HDPO Ding (2026) builds an answer-conditioned teacher distribution and transfers it to the question-only policy through Jensen–Shannon divergence. RAVR Lin et al. (2025) and CoVRL Wen et al. (2026) align the question-only distribution with answer-guided reference behavior through KL regularization.

## 4. Task-Level Hints

*Task-level hints* extend sample-specific hints into a cross-sample hint base, enabling experience reuse across samples during training. As such hints are inherently cross-sample, they are also termed

 doi:10.20944/preprints202606.1050.v1



Experience. As shown in Figure 2, their distinctive feature over sample-level hints lies in the differentiated construction mechanisms of the experience base. We branch primarily by whether the base is updated during training, and further subdivide by utilization method and content.

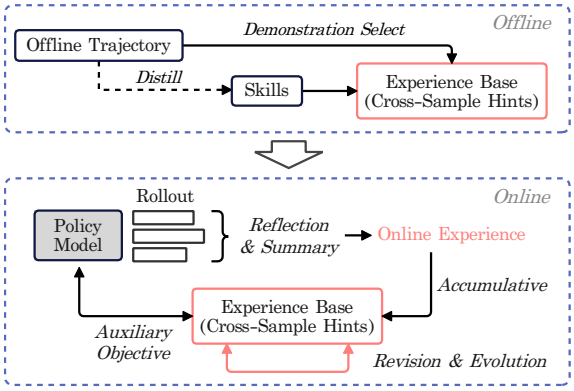


**Figure 2.** Offline and online construction mechanisms specific to task-level hints.

### *4.1. Static Experience Base*

Static experience base methods construct the experience base before target RL training and keep it fixed during policy optimization. During training, they retrieve relevant experience by task, domain, or similarity and use it as an additional condition for the current training sample.

#### Demonstration-Based Experience

These methods use cross-sample problems and solutions as few-shot hints. CBRL Agashe et al. (2026) retrieves relevant demonstrations by task tags and anneals the injection probability to zero during training. ICPO Huang et al. (2026) uses demonstrations to induce an expert-conditioned trajectory from the current policy; when the trajectory passes the verifier, it randomly replaces one ordinary trajectory in the rollout group.

#### Skill-based Experience

These methods use reusable skills or strategy templates as hints. TemplateRL Wu et al. (2025a) searches successful paths with MCTS on seed problems, removes problem-specific solution content, keeps the action sequence, and retrieves nearby templates by problem complexity. SKILL0 Lu et al. (2026) selects skill files by the performance gap on matched validation tasks with and without the skill, and gradually reduces the skill context until the policy can solve tasks without it.

### *4.2. Evolving Experience Base*

Evolving experience base methods add, modify, remove, or reconstruct experience entries during RL training, allowing hints to evolve with policy optimization. Compared with static experience bases, they adapt to the changing capability boundary of the policy and preserve the usefulness of experience hints.

#### 4.2.1. Accumulative Experience Base

Accumulative experience base methods continuously add new experience from training interactions into the experience base, ensuring that newly discovered hints can be retained and reused.

#### Reflection Experience

These methods write reflections over the agent's own interaction rollouts into the experience base, so later rollouts can reuse improvement signals distilled from prior trial and error. EvolveR Wu et al. (2025b) and RetroAgent Zhang et al. (2026) both accumulate reflective experience from rollout outcomes. The former distills successful and failed trajectories into guiding and cautionary principles, while the latter stores retrospective lessons and retrieves them by relevance and historical utility. ERL Shi et al. (2026)

 doi:10.20944/preprints202606.1050.v1



focuses on failure-driven correction, adding a reflection to the experience base only when it leads to a successful retry. MAGE lu Yang et al. (2026) keeps reflection within the same meta-rollout, using previous interaction history, rewards, and reflections as the experience context for the next rollout.

Strategy Experience

These methods store decision strategies formed during training. SGE Szot et al. (2026) stores strategy sequences from rollouts, indexed by task and outcome, later using successful strategies as references and failed strategies to trigger critique and revision. CRMWeaver Lai et al. (2025) distills advanced-model execution logs into workflow guidelines and retrieves them for similar business queries. IntPro Liu et al. (2026) stores intent explanations from user history, allowing the agent to retrieve historical intent patterns under uncertainty. MetaClaw Xia et al. (2026) generates skill instructions from failed trajectories, immediately saves them into a long-term experience base, and uses them in later tasks.

4.2.2. Curated Experience Base

Curated experience base methods go beyond appending new entries; they merge, revise, or replace existing ones to ensure experience reliability.

Skill Evolution

These methods treat skills as experience entries and maintain reusable action procedures. SkillRL Xia et al. (2026) and ARISE Li et al. (2026) generate structured skills from training rollouts. SkillRL updates the SkillBank by analyzing failures in underperforming task categories, while ARISE lets the policy handle skill generation, selection, and use in one loop. COS-PLAY Wu et al. (2026) and SAGE$_{exp}$ Wang et al. (2026) make skills executable. COS-PLAY turns interaction segments into skill protocols with effect contracts, while SAGE$_{exp}$ lets the agent create, call, repair, and save function skills. K$^2$-Agent Wu et al. (2026) combines task-level know-what rules with low-level know-how action hints, revising high-level rules after execution failures.

Experience Revision

These methods revise experience entries or their retrieved form using new rollouts, failures, or usage feedback. BEPA Wang et al. (2026) first converts expert trajectories into policy-compatible successful trajectory caches, then refreshes each task cache with new successes from the current policy during RL. DGO Bai et al. (2026) and Comp.RL Muhtar et al. (2026) both update experience from fresh interaction trajectories. DGO extracts strategies and pitfall warnings from correct and incorrect trajectories and renews the experience base after each RL round, while Comp.RL lets an extractor decide whether to add, update, or merge entries from the current trajectory, outcome, and previously used experience. INSPO Zhou et al. (2026), AgentEvolver Zhai et al. (2025), and PEARL Li et al. (2026) revise higher-level strategy text. INSPO rewrites the instruction population from failed trajectories, AgentEvolver reranks and rewrites retrieved experience into a task-specific hint, and PEARL explicitly reads and updates user preference strategies during calendar decisions.

4.2.3. Optimized Experience Base

Optimized experience base methods bring the utility of experience into training, so that experience selection, weighting, or generation modules are shaped by policy feedback.

Experience-Derived Utility

These methods turn retrieved experience into update signals while still using experience to guide rollout generation or teacher-side scoring. D2Skill Tu et al. (2026) and SLEA-RL Wang and Jiang (2026) both inject retrieved experience during rollout sampling and then estimate its utility from the resulting performance. D2Skill updates skill utility and forms hindsight reward from the success gap between skill and baseline groups. SLEA-RL retrieves strategy or warning experience by observation clusters and combines step-level experience with local advantage. Skill-SD Wang et al. (2026) uses per-task

 doi:10.20944/preprints202606.1050.v1



skills only on the teacher side, turning skill-conditioned teacher scores on student on-policy tokens into a self-distillation loss.

Trainable Experience Module

These methods train a module that produces or selects textual experience before it guides generation. UMEM Ye et al. (2026) trains a Mem-Optimizer to extract and manage generalizable memory entries using semantic-neighborhood utility. Trainable Graph Memory Xia et al. (2025) organizes historical trajectories into a weighted graph of query, transition, and meta-cognition nodes, and trains the weights so useful meta-cognition strategies are more likely to be retrieved and prepended to the policy prompt.

## 5. Domain-Specific Hints

In domain-specific settings, hint-based RL converts domain-native signals into training-time guidance. We organize representative work along three domain families. In technical domains, hints are drawn from executable or formally checkable feedback. Kevin Baronio et al. (2025) leverages compilation and runtime feedback for CUDA kernel refinement, SGS Bailey et al. (2026) synthesizes related theorems around unsolved targets with Lean providing verification, and C2F-Thinker Luo et al. (2026) steers multimodal sentiment reasoning through hierarchical coarse-to-fine cues. In interactive agent domains, hints are drawn from environment states and action-validity signals. COS-PLAY Wu et al. (2026) abstracts game interaction segments into skill protocols, WebGen-Agent Lu et al. (2025a) turns execution, visual, and GUI-agent feedback into step-level guidance, and UI-S1 Lu et al. (2025b) patches failed rollouts with expert GUI actions. In business and personal-service domains, hints are drawn from behavioral norms, workflow rules, and user-specific preferences. TaoSR-AGRL Yang et al. (2026) treats e-commerce relevance labels as guided replay signals, CRMWeaver Lai et al. (2025) retrieves business workflow guidelines, VeriRole Wang et al. (2026) derives verifiable role-awareness signals, and PEARL Li et al. (2026) maintains preference strategies for time-management decisions.

## 6. Discussion

### *6.1. Scope and Boundaries*

Several lines of work are formally similar to hint-based RL but fall outside our scope. ***Trajectory augmentation methods*** expand sampling at predefined branching points via random perturbation or tree search to enhance exploration Ji et al. (2026); Dong et al. (2026), raising the pass@k of the current policy without introducing information beyond its own distribution. ***Input rewriting methods*** modify the query itself, for instance by adjusting problem difficulty Muhtar et al. (2026); Guo et al. (2026); Zhang et al. (2025) or converting open-ended questions into multiple-choice ones Chen et al. (2026), which also mitigates zero-advantage failure. Such methods reshape the training query distribution, whereas hints inject external information while preserving the original query. Finally, some methods introduce hint-like mechanisms such as reference trajectory segments Nourzad and Joe-Wong (2026), critics Xu et al. (2026), and finer evaluation criteria Sheng et al. (2026); others use teacher Gu et al. (2026) or self-generated Yang et al. (2026) text for on-policy distillation Song and Zheng (2026). Their shared purpose is to assign more accurate rewards to existing rollouts rather than to expand the reachable trajectory set, which places them under ***fine-grained reward methods***.

**Table 1.** Count-based cross-analysis of hint construction sources and utilization mechanisms. Abbreviations on the left correspond to the four utilization types.

| | Offline | | | Online | | | |
|---|---|---|---|---|---|---|---|
| | **Human** | **Base** | **Teacher** | **Old** | **Current** | **Teacher** | **Total** |
| **Inj.** | 3 | 1 | 16 | 6 | 10 | 2 | 38 |
| **Cont.** | 1 | 1 | 8 | 1 | 1 | 0 | 12 |
| **Repl.** | 1 | 1 | 9 | 1 | 9 | 2 | 23 |
| **Opt.** | 3 | 2 | 3 | 3 | 3 | 6 | 20 |
| **Total** | 8 | 5 | 36 | 11 | 23 | 10 | 93 |

 doi:10.20944/preprints202606.1050.v1



### 6.2. Cross-Level Analysis of Hints

Sections 3 and 4 organize methods by content refinement and experience base type, respectively. Here we provide a cross-analysis of construction sources and utilization mechanisms.

Task-level hints extend sample-specific hints into cross-sample experience bases, with their unique contribution in base construction and maintenance. On the utilization side, however, task-level methods inherit the same four mechanisms. Table A4 shows task-level methods distributing across the same four rows, with injection predominant and continuation least used. Task-level methods also show notably lower density in replacement, as abstracted experience entries are better suited for injection than trajectory substitution. The two levels thus share a utilization framework but diverge in construction: sample-level methods focus on per-hint generation quality, task-level methods on cross-sample accumulation and management.

Task-level hints extend sample-specific hints into cross-sample experience bases, with their unique contribution in base construction and maintenance. On the utilization side, however, task-level methods inherit the same four mechanisms. Table A4 shows task-level methods distributing across the same four rows, with injection predominant and continuation least used. Task-level methods also show notably lower density in replacement, likely because experience entries are typically abstracted and better suited as contextual conditions than as trajectory substitutes. The two levels thus share a utilization framework but diverge in construction: sample-level methods focus on per-hint generation quality, task-level methods on cross-sample accumulation and management.

This divergence entails different trade-offs. Sample-level hints offer the most direct intervention for zero-advantage groups but are single-use, leaving construction costs unamortized. Task-level hints amortize costs through reuse and support continual adaptation and self-evolution, yet risk reinforcing erroneous or stale entries. Static bases are stable but cannot adapt to policy changes; evolving bases are flexible but harder to control. Further category-level comparisons are in Appendix A.3.

### 6.3. Challenges and Future Directions

Despite the promise of hint-based RL in mitigating zero-advantage failure, several open problems and promising directions remain. (1) Regarding the quality of hints, controllable construction of high-quality hints remains a core challenge. An ideal hint should elicit responses that remain close to the policy's distribution while introducing novel knowledge or reasoning patterns that the policy can effectively absorb. Meanwhile, the train-inference mismatch caused by hints being available only during training has yet to be systematically studied, since policies may degrade once the hint is removed. We provide an extended discussion of this issue in Appendix A.4. (2) Regarding the use of hints, hint-based RL offers a viable path toward model self-evolution, and debiasing mechanisms for task-level experience bases are crucial for long-term training stability, since uncurated entries bias future retrieval. Learnable hinters also stand out as a promising direction, where multiple models may serve as hinters for one another and be jointly optimized during training. (3) Finally, when hints are biased or experience bases are contaminated with inadvertent errors or deliberately injected malicious entries, the policy may drift unpredictably or even violate safety constraints. Trustworthy hint-based RL incorporating trustworthiness estimation and safety mechanisms thus stands as an important direction for future research.

## 7. Conclusions

This survey consolidates recent hint-based RL methods that address the zero-advantage failure through explicit textual guidance. We organize them into sample-level hints, intervening within a single sample via trajectory-based or scaffold-based guidance, and task-level hints, maintaining reusable experience bases across samples. We further outline open problems and future directions.

## Limitations

While we have aimed to provide a systematic review of hint-based RL, several limitations remain. First, the field is moving quickly, and we may have missed recent contributions, especially preprints

 doi:10.20944/preprints202606.1050.v1



and other unpublished work. Second, since experimental setups differ substantially across hint-based RL methods and uniform reproduction is impractical, a fair empirical comparison across paradigms lies beyond the scope of this survey. Third, the open problems and future directions we highlight reflect our current understanding and may become outdated as the field continues to evolve.

## Appendix A

### *Appendix A.1. Taxonomy and Representative Methods*

Figure A1 provides an expanded view of the taxonomy defined in Section 2 and lists representative methods under each branch. Sample-level methods are divided into trajectory-based and scaffold-based hints according to whether the explicit textual signal is a response trajectory or a semantic scaffold. Task-level methods are divided into static and evolving experience bases according to whether the shared hint repository is updated during RL training. Domain-specific methods are shown as a separate application-oriented branch because domain-native signals are especially important when specialized knowledge, executable feedback, environment states, or user preferences must be converted into training-time guidance. These methods can still be analyzed through the sample-level or task-level granularity, and the separate branch highlights important deployment settings without treating them as a disjoint mechanism family.

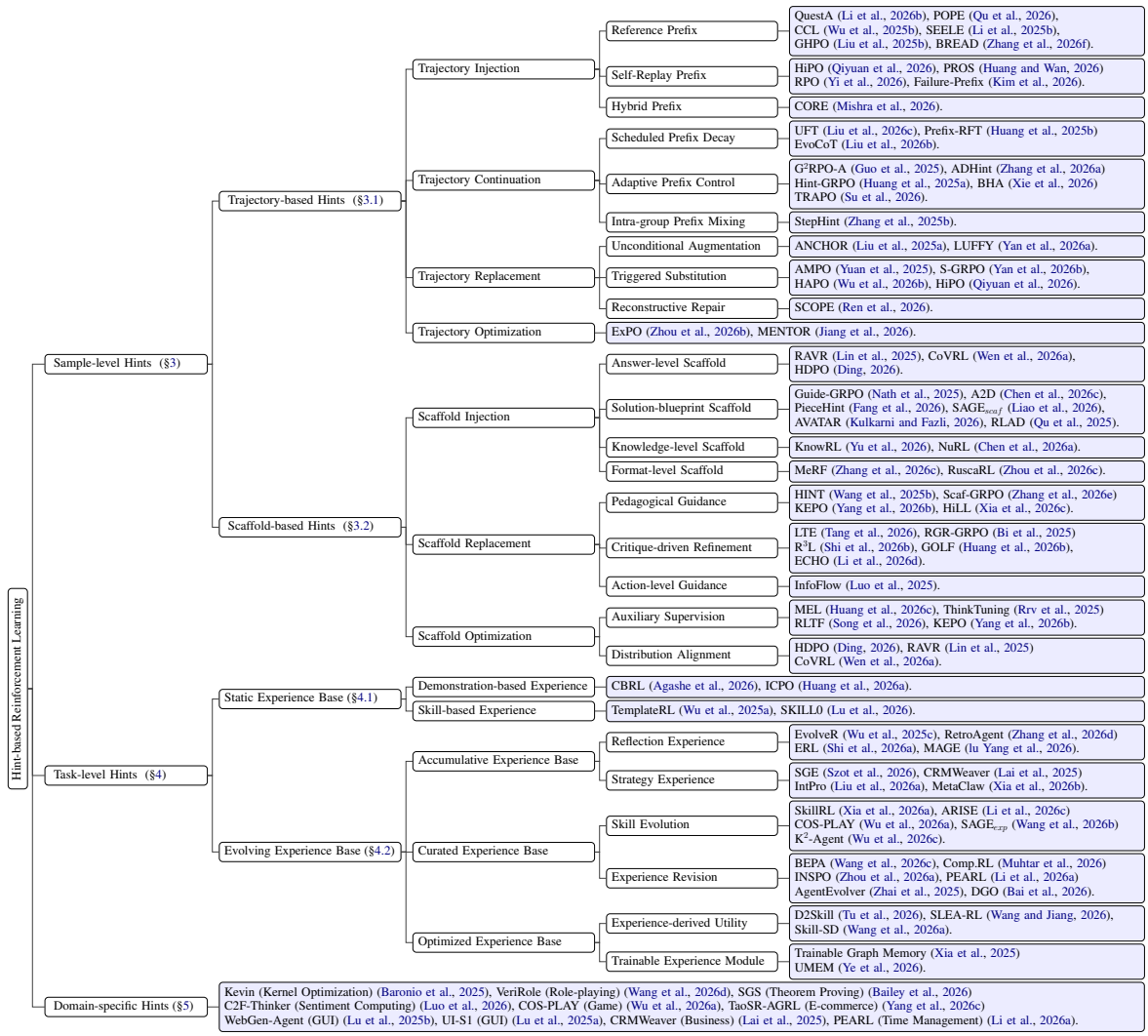


**Figure A1.** Taxonomy of hint-based reinforcement learning methods.

 doi:10.20944/preprints202606.1050.v1



*Appendix A.2. Comparison Tables and Field Definitions*

Tables A1, A2, and A3 summarize the reporting fields used for method-level comparison. The fields are defined as follows.

- *Hint Content*. This field names the explicit textual signal used by a method, such as a reference prefix, repaired trajectory, critique, workflow, skill, memory entry, executable feedback, or preference signal. It corresponds to the hint object that directly affects rollout generation, rollout repair, or the RL update.
- *Source*. This field records where the hint comes from. *On-policy* indicates that the hint is generated by the trained model itself. *Off-policy* indicates that the hint is produced with an external model, teacher model, external annotated trajectory, or external reference solution. *Hybrid* corresponds to *Off&On-policy*, where hint construction explicitly depends on both the trained model and an external model or teacher. External verifiers, rule-based scorers, and Math-Verify style correctness checkers are treated as reward or validation mechanisms. They do not change the hint source by themselves.
- *Trigger*. This field records when the hint intervenes in RL training. We use *Always*, *Stage-wise Curriculum*, and *Specific Rules* to distinguish persistent intervention, curriculum-dependent intervention, and rule-triggered intervention.
- *Inference*. This field records whether hints are still used after training. It supports the deployment discussion in Appendix A.4, where training-time hints may be removed, retained, or partially retained at inference.
- *Objective*. This field records the concrete optimization objective or training recipe reported by each method. It may include GRPO-family objectives, DAPO, REINFORCE, or method-specific variants, as well as auxiliary terms such as SFT, Kullback-Leibler divergence (KL), negative log-likelihood (NLL), Jensen-Shannon divergence (JSD), or self-distillation losses (SDL). Note that objectives labeled as GRPO in the table may also refer to GRPO variants adapted to specific methods.
- *Retrieval / Use*. This field appears in the task-level table and records how stored experience is selected or applied, such as random sampling, similarity retrieval, category retrieval, top-$K$ retrieval, or population sampling.
- *Experience Base Operation*. This field in the task-level table records whether the experience repository is fixed or updated through operations including addition, update, deletion, merging, pruning, or reconstruction.

**Table A1.** Detailed comparison of sample-level hint methods in hint-based reinforcement learning.

| Method | Hint Content | Source | Trigger | Inference | Objective |
|---|---|---|---|---|---|
| | | *LLM* | | | |
| StepHint Zhang et al. (2025) | step-level reference prefixes | Off-policy | Always | No | GRPO |
| ANCHOR Liu et al. (2025) | verified reference trajectory anchor | Off-policy | Always | No | GRPO |
| LUFFY Yan et al. (2026) | off-policy expert reasoning trace | Off-policy | Always | No | GRPO |
| RAVR Lin et al. (2025) | ground-truth answer | Off-policy | Always | No | GRPO+KL |
| CoVRL Wen et al. (2026) | ground-truth answer | Off-policy | Always | No | GRPO+KL+NLL |
| MeRF Zhang et al. (2026) | natural-language reward specification | Off-policy | Always | No | GRPO |
| RAVR Lin et al. (2025) | answer-guided reference distribution | Off-policy | Always | No | GRPO+KL |
| CoVRL Wen et al. (2026) | answer-guided mixed trajectory distribution | Off-policy | Always | No | GRPO+KL+NLL |
| POPE Qu et al. (2026) | short useful reference prefix | Off-policy | Rules | No | GRPO |
| SEELE Li et al. (2025) | reference prefix with learned length control | Off-policy | Rules | No | GRPO+SFT |
| BREAD Zhang et al. (2026) | episode-level expert prefix | Off-policy | Rules | No | GRPO |
| $C^2$RPO-A Guo et al. (2025) | trajectory prefix adjusted by reward trend | Off-policy | Rules | No | GRPO |
| TRAPO Su et al. (2026) | expert prefix selected within the rollout group | Off-policy | Rules | No | GRPO+SFT |
| AMPO Yuan et al. (2025) | selected teacher reference trajectories | Off-policy | Rules | No | GRPO |
| HAPO Wu et al. (2026) | reference trajectory replacing a low-reward rollout | Off-policy | Rules | No | GRPO+SFT |
| HDPO Ding (2026) | ground-truth answer | Off-policy | Rules | No | GRPO+JSD |
| Guide-GRPO Nath et al. (2025) | teacher-generated high-level guidance | Off-policy | Rules | No | GRPO |
| KnowRL Yu et al. (2026) | minimal sufficient knowledge points | Off-policy | Rules | Opt. | GRPO |
| HINT Wang et al. (2025) | teacher-generated heuristic hints | Off-policy | Rules | No | GRPO |
| RGR-GRPO Bi et al. (2025) | unmet rubric criteria | Off-policy | Rules | No | GRPO |
| HDPO Ding (2026) | answer-conditioned teacher distribution | Off-policy | Rules | No | GRPO+JSD |
| UFT Liu et al. (2026a) | reference prefix with scheduled decay | Off-policy | Curriculum | No | GRPO+KL |
| PieceHint Fang et al. (2026) | high-value reasoning pieces | Off-policy | Curriculum | No | GRPO |
| RuscaRL Zhou et al. (2026) | checklist-style rubric criteria | Off-policy | Curriculum | No | GRPO |
| ThinkTuning Rrv et al. (2025) | teacher reflection tokens | Off-policy | Curriculum | No | GRPO |
| QuestA Li et al. (2026) | partial reference solution prefix | Off-policy | Rules+Curriculum | No | GRPO |

 doi:10.20944/preprints202606.1050.v1



**Table A1.** *Cont.*

| Method | Hint Content | Source | Trigger | Inference | Objective |
|---|---|---|---|---|---|
| | | *LLM* | | | |
| CCL Wu et al. (2025) | threshold-calibrated reference prefix | Off-policy | Rules+Curriculum | No | GRPO |
| GHPO Liu et al. (2025) | stagewise reference prefix | Off-policy | Rules+Curriculum | No | GRPO |
| Prefix-RFT Huang et al. (2025) | offline demonstration prefix | Off-policy | Rules+Curriculum | No | Dr.GRPO |
| Scaf-GRPO Zhang et al. (2026) | hierarchical knowledge, planning, and solution hints | Off-policy | Rules+Curriculum | No | GRPO |
| RPO Yi et al. (2026) | truncated historical response prefix | On-policy | Always | No | GRPO/DAPO |
| HiPO Qiyuan et al. (2026) | successful self-prefix from current batch | On-policy | Rules | No | GRPO |
| PROS Huang and Wan (2026) | uncertainty-truncated historical correct prefix | On-policy | Rules | No | GRPO |
| HiPO Qiyuan et al. (2026) | successful self-hinted trajectory group | On-policy | Rules | No | GRPO |
| LTE Tang et al. (2026) | wrong-answer negative hints | On-policy | Rules | No | GRPO |
| Failure-Prefix Kim et al. (2026) | rare failure prefix from saturated samples | On-policy | Rules+Curriculum | No | GRPO |
| ExPO Zhou et al. (2026) | answer-conditioned self-explanation trajectory | Hybrid | Always | No | GRPO+SFT |
| RLAD Qu et al. (2025) | reasoning abstraction | Hybrid | Always | Yes | DAPO+RFT+SFT |
| ECHO Li et al. (2026) | diagnostic feedback from a trainable critic | Hybrid | Always | No | GRPO |
| RLTF Song et al. (2026) | natural-language critique feedback | Hybrid | Always | Opt. | GRPO+CE+AWR |
| CORE Mishra et al. (2026) | answer-stripped prefix from a successful peer policy | Hybrid | Rules | No | GRPO |
| SCOPE Ren et al. (2026) | teacher-corrected suffix after a valid prefix | Hybrid | Rules | No | GRPO+NLL |
| A2D Chen et al. (2026) | decomposer-generated sub-questions | Hybrid | Rules | No | GRPO+NLL |
| $SAGE_{scaf}$ Liao et al. (2026) | self-generated plan or decomposition | Hybrid | Rules | No | GRPO |
| HiLL Xia et al. (2026) | hinter-generated pedagogical hints | Hybrid | Rules | No | GRPO |
| GOLF Huang et al. (2026) | group-level feedback from critiques | Hybrid | Rules | Opt. | GRPO |
| MEL Huang et al. (2026) | verified meta-experience target | Hybrid | Rules | No | GRPO+NLL |
| EvoCoT Liu et al. (2026b) | CoT prefix constructed from final answer | Hybrid | Curriculum | No | GRPO |
| BHA Xie et al. (2026) | distribution-aligned teacher prefix | Hybrid | Rules+Curriculum | No | DAPO |
| MENTOR Jiang et al. (2026) | mixed-policy rollout with expert token intervention | Hybrid | Rules+Curriculum | No | GRPO |
| NuRL Chen et al. (2026) | self-generated abstract knowledge cue | Hybrid | Rules+Curriculum | No | GRPO |
| | | *VLLM* | | | |
| ADHint Zhang et al. (2026) | reference prefix selected by query difficulty | Off-policy | Rules | No | GRPO |
| Hint-GRPO Huang et al. (2025) | stepwise reference prefix | Off-policy | Rules | No | GRPO |
| S-GRPO Yan et al. (2026) | single verified reference trajectory | Off-policy | Rules | No | GRPO |
| AVATAR Kulkarni and Fazli (2026) | precomputed strategy hints | Off-policy | Rules | No | GRPO+SFT |
| KEPO Yang et al. (2026) | answer-conditioned teacher reasoning hints | Off-policy | Rules | No | GRPO+KL+SFT |
| | | *Agent* | | | |
| InfoFlow Luo et al. (2025) | pathfinding hint used as a search query | Off-policy | Rules | No | SFT+GRPO+KL |
| $R^3L$ Shi et al. (2026) | reflection diagnosis with corrective guidance | Hybrid | Rules | No | GRPO |

**Table A2.** Detailed comparison of task-level hint methods in hint-based reinforcement learning.

| Method | Hint Content | Source | Retrieval / Use | Experience Base Operation | Trigger | Inference | Objective |
|---|---|---|---|---|---|---|---|
| | | | *LLM* | | | | |
| CBRL Agashe et al. (2026) | Cross-sample solved demonstrations | Off-policy | Label sampling | Fixed | Curriculum | No | GRPO |
| TemplateRL Wu et al. (2025a) | Problem-solving templates | Hybrid | PCC matching | Fixed | Always | Yes | Dr.GRPO |
| ICPO Huang et al. (2026) | Expert problem-solution demonstrations | Hybrid | Random sampling | Fixed | Rules | No | GRPO |
| DGO Bai et al. (2026) | Correct and wrong reasoning experience | Hybrid | Experience retrieval | Add/Update/Delete | Curriculum | Yes | GRPO+SFT |
| | | | *Agent* | | | | |
| SkillRL Xia et al. (2026) | Task-level skills | Off-policy | Category retrieval | Add/Update | Always | Yes | GRPO+SFT |
| CRMWeaver Lai et al. (2025) | Workflow guidelines | Off-policy | Top-1 similarity | Add/Update | Rules | Yes | DAPO+SFT |
| SKILL0 Lu et al. (2026) | Procedural skills | Off-policy | Skill selection | Fixed | Rules+Curriculum | No | GRPO |
| INSPO Zhou et al. (2026) | Evolved instructions | Off-policy | Population sampling | Add/Delete/Reconstruct | Rules+Curriculum | Yes | GRPO |
| COS-PLAY Wu et al. (2026) | Long-horizon game skills | On-policy | Skill retrieval | Add/Update/Merge/Delete | Always | Yes | GRPO+SFT |
| $SAGE_{exp}$ Wang et al. (2026) | Executable function skills | On-policy | Skill retrieval | Add/Update | Always | Yes | GRPO+SFT |
| UMEM Ye et al. (2026) | Editable memory-style experience entries | On-policy | Top-K retrieval | Add/Update | Rules | Yes | GRPO |
| EvolveR Wu et al. (2025b) | Guiding and cautionary reflections | On-policy | Explicit search | Add/Update/Merge/Prune | Rules | Yes | GRPO+SFT |
| RetroAgent Zhang et al. (2026) | Retrospective lessons | On-policy | SimUtil-UCB | Add/Update | Rules | Yes | GRPO |
| ERL Shi et al. (2026) | Failure-correction reflections | On-policy | Threshold retry | Add | Rules | No | GRPO+SFT |
| SGE Szot et al. (2026) | Successful and failed strategies | On-policy | FIFO sampling | Add/Delete | Rules | No | GRPO |
| SLEA-RL Wang and Jiang (2026) | Step-level strategies and warnings | On-policy | Cluster retrieval | Add/Delete | Rules | Yes | GRPO |
| MAGE lu Yang et al. (2026) | Interaction history and reflections | On-policy | Meta-context | Temporary aggregation | Curriculum | Yes | GiGPO |
| PEARL Li et al. (2026) | Time-management strategies | On-policy | StrategyHub | Add/Update/Delete | Rules+Curriculum | Yes | GRPO |
| Comp.RL Muhtar et al. (2026) | Reusable trajectory-derived experience | Hybrid | Search-and-ask | Add/Update/Merge/Delete | Always | Yes | Split-GRPO |
| D2Skill Tu et al. (2026) | Task and step skills | Hybrid | Utility comparison | Add/Update/Prune | Always | Yes | GRPO |
| Skill-SD Wang et al. (2026) | Teacher-side task skills | Hybrid | Teacher scoring | Add/Update | Always | No | GRPO+SDL |
| IntPro Liu et al. (2026) | User intent patterns with explanations | Hybrid | Tool retrieval | Add/Update | Rules | Yes | GRPO+SFT |
| MetaClaw Xia et al. (2026) | Failure-driven skill instructions | Hybrid | Embedding search | Add | Rules | Yes | GRPO |
| BEPA Wang et al. (2026) | Per-task successful trajectories | Hybrid | Cache lookup | Add/Update | Rules | No | GRPO |
| AgentEvolver Zhai et al. (2025) | Natural-language experience entries | Hybrid | Vector retrieval | Add/Update | Rules | Yes | GRPO |
| Trainable G.Mem. Xia et al. (2025) | Graph-structured meta-cognition | Hybrid | Graph retrieval | Add/Merge/Update | Rules | Yes | REINFORCE |
| $K^2$-Agent Wu et al. (2026) | Mobile-control rules and demonstrations | Hybrid | Type retrieval | Add/Update/Merge | Curriculum | Yes | C-GRPO |
| ARISE Li et al. (2026) | Seed and generated skills | Hybrid | Manager selection | Add/Update/Delete | Rules+Curriculum | Yes | GRPO |

 doi:10.20944/preprints202606.1050.v1



**Table A3.** Detailed comparison of domain-specific hint methods in hint-based reinforcement learning.

| Method | Hint Content | Source | Trigger | Inference | Objective |
|---|---|---|---|---|---|
| | | *LLM* | | | |
| TaoSR-AGRL Yang et al. (2026) | Dimension-level business labels | Off-policy | Rules | No | GRPO |
| Kevin Baronio et al. (2025) | Execution feedback and prior kernel attempts | On-policy | Always | Yes | GRPO |
| SGS Bailey et al. (2026) | Synthetic theorem or subproblem | On-policy | Rules | No | REINFORCE |
| VeriRole Wang et al. (2026) | Role evidence snippets | Hybrid | Always | Yes | GRPO |
| | | *VLLM* | | | |
| C2F-Thinker Luo et al. (2026) | Ground-truth polarity label | Off-policy | Rules | No | GRPO+SFT |
| | | *Agent* | | | |
| WebGen-Agent Lu et al. (2025a) | Execution, visual, and GUI feedback | Off-policy | Rules | Yes | Step-GRPO+SFT |
| UI-S1 Lu et al. (2025b) | Expert GUI action patch | Off-policy | Rules | No | Semi-online GRPO+SFT |
| CRMWeaver Lai et al. (2025) | Workflow guideline | Off-policy | Rules | Yes | DAPO+SFT |
| COS-PLAY Wu et al. (2026) | Skill protocol | On-policy | Always | Yes | GRPO+SFT |
| PEARL Li et al. (2026) | Preference strategy | On-policy | Rules+Curriculum | Yes | GRPO |

*Appendix A.3. Category-Level Comparison of Hints*

Section 6.2 provides a cross-analysis of construction sources and utilization mechanisms across sample-level and task-level hints. Here we further compare the major subcategories in Figure A1 and Tables A1 and A2. For sample-level methods, *trajectory-based hints* and *scaffold-based hints* differ mainly in how explicitly they constrain the rollout. Trajectory-based hints expose concrete solution states, such as reference prefixes, completed trajectories, repaired suffixes, or mixed expert-policy rollouts. They are therefore effective for cold-start samples where the current policy rarely reaches a successful path, because they reduce the search depth or insert a reliable contrastive anchor. Their strength also brings risk. If the trajectory is too long, too off-policy, or too close to the answer, the policy may learn to imitate the hint context and fail to recover the underlying reasoning ability. This makes trajectory source, revealed amount, and trigger condition central to trajectory-based methods.

*Scaffold-based hints* provide less explicit guidance, such as abstractions, constraints, knowledge cues, rubrics, or critiques. They are useful when multiple valid reasoning paths exist or when the goal is to steer the policy's strategy without prescribing a concrete path. Compared with trajectory-based hints, scaffolds preserve more generation freedom and are less tied to one reference solution. However, their effectiveness is harder to verify. A scaffold can be too vague to change the rollout distribution or too noisy to provide reliable improvement. As a result, scaffold-based methods often control the construction or use of scaffolds through compact and non-leaking cues, verifier-guided replacement or refinement, and auxiliary supervision or distribution alignment.

Within task-level methods, the contrast between *static experience base* and *evolving experience base* concerns how persistent guidance is managed. A static experience base is constructed before target RL training and remains fixed, making it relatively stable, auditable, and easy to reproduce. This is useful when the experience source is trusted and the task distribution is stable. Its limitation is that the base cannot absorb newly discovered strategies or remove entries that become stale as the policy changes. An evolving experience base updates entries during training through addition, revision, merging, deletion, pruning, or utility estimation. This allows the experience base to track the policy's changing capability boundary and is better aligned with self-evolving agents. This flexibility also makes control more difficult. Erroneous entries, retrieval bias, duplicated strategies, or contaminated memories can be reinforced over time unless the system includes debiasing, validation, and maintenance mechanisms.

 doi:10.20944/preprints202606.1050.v1



**Table A4.** Cross-analysis of hint construction sources and utilization mechanisms across sample-level and task-level methods. Sample-level methods: trajectory-based, scaffold-based. Task-level methods: static experience base, evolving experience base. Methods spanning two construction sources appear in both columns. "–" denotes unexplored combinations. † Method spans two construction sources and appears in both corresponding columns.

| | Offline Construction | | | Online Construction | | |
|---|---|---|---|---|---|---|
| **Utilization** | **Human** | **Base Policy** | **Teacher** | **Old Policy** | **Current Policy** | **Teacher** |
| **Hint Injection** | GHPO<br>MeRF<br>CBRL† | NuRL | QuestA, POPE<br>CCL, SEELE<br>BREAD<br>Guide-GRPO<br>PieceHint, AVATAR<br>KnowRL, RuscaRL<br>RLAD†<br>SKILL0, CBRL†<br>CRMWeaver, IntPro<br>SkillRL | PROS, RPO<br>Failure-Prefix†<br>EvolveR<br>RetroAgent, SGE | CORE<br>Failure-Prefix†<br>$SAGE_{scaf}$<br>RLAD†<br>MAGE, ARISE<br>COS-PLAY<br>$SAGE_{exp}$<br>Comp.RL, PEARL | MetaClaw<br>INSPO |
| **Hint Continuation** | K2-Agent† | EvoCoT† | UFT, Prefix-RFT<br>$G^2$RPO-A, ADHint<br>Hint-GRPO, BHA<br>TRAPO, StepHint | EvoCoT† | K2-Agent† | – |
| **Hint Replacement** | ICPO | BEPA† | ANCHOR, LUFFY<br>AMPO, S-GRPO<br>HAPO<br>HINT, Scaf-GRPO<br>InfoFlow<br>RGR-GRPO† | BEPA† | HiPO, SCOPE†<br>LTE, $R^3$L<br>ECHO, HiLL<br>GOLF†<br>RGR-GRPO†<br>ERL | SCOPE†<br>GOLF† |
| **Hint Optimization** | RAVR, CoVRL<br>HDPO | ExPO<br>A2D† | MENTOR<br>DGPO<br>A2D† | DGO†<br>SLEA-RL<br>Trainable G.Mem. | MEL<br>DGO†<br>UMEM | KEPO<br>ThinkTuning<br>RLTF<br>AgentEvolver<br>D2Skill, Skill-SD |

*Appendix A.4. Hint Availability at Inference*

The *Inference* column in Tables A1, A2, and A3 records whether a method requires hints after training. This distinction is important because many hint-based RL methods use hints only during training as textual signals. In the broader coding underlying these tables, 86 out of 116 entries do not use hints at inference, while 30 do. The pattern is uneven across categories. Sample-level trajectory and scaffold methods are dominated by inference-free designs, whereas task-level experience methods more often keep retrieved skills, memories, workflows, or preferences available during deployment.

For sample-level methods, removing the hint at inference creates a potential mismatch between training and inference. Reference prefixes, answer-conditioned scaffolds, teacher critiques, and replacement trajectories can make hard samples learnable during RL, but the deployed policy must solve the original query without these guided conditions. If the policy only learns to complete from hinted contexts, removing the hint can shift the input distribution back to the hard prompt, causing the model to skip early reasoning steps, rely on answer-conditioned shortcuts, or fail to reproduce the capability learned under guidance.

Existing methods mitigate this mismatch in several ways. One line gradually reduces hint exposure through prefix decay, backward annealing, hint dropout, random withholding, or curriculum schedules Huang et al. (2025); Lu et al. (2026); Mishra et al. (2026); Liu et al. (2026a); Xie et al. (2026); Fang et al. (2026); Zhou et al. (2026); Agashe et al. (2026). Another line controls hint strength so that the weakest useful hint is used only when the original group is uninformative, reducing long-term dependence on guided contexts Qu et al. (2026); Zhang et al. (2026,?); Qiyuan et al. (2026); Huang et al. (2025); Su et al. (2026). A third line transfers behavior from hinted generation to the unguided policy through distribution alignment, distillation, or losses computed under unscaffolded conditions Lin et al. (2025); Wen et al. (2026); Ding (2026); Zhou et al. (2026); Wang et al. (2026). Many task-level experience methods intentionally keep retrieved skills, memories, or preference strategies at inference Xia et al.

 doi:10.20944/preprints202606.1050.v1



(2026); Wu et al. (2025a); Lai et al. (2025); Zhai et al. (2025); Li et al. (2026); Tu et al. (2026). In these cases, the mismatch is reduced, but deployment depends on reliable retrieval, bounded context cost, and safeguards against stale or malicious experience. A more systematic evaluation of hint reliance, for example by comparing hinted, hint-free, and partially ablated inference at the same checkpoint, remains an important direction.

## References


Wen, X.; Liu, Z.; Zheng, S.; Ye, S.; Wu, Z.; Wang, Y.; Xu, Z.; Liang, X.; Li, J.; Miao, Z.; et al. Reinforcement Learning with Verifiable Rewards Implicitly Incentivizes Correct Reasoning in Base LLMs. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; et al. Towards Large Reasoning Models: A Survey of Reinforced Reasoning with Large Language Models, 2025, [arXiv:cs.AI/2501.09686].

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.K.; Wu, Y.; et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024, [arXiv:cs.CL/2402.03300].

Zhang, K.; Zuo, Y.; He, B.; Sun, Y.; Liu, R.; Jiang, C.; Fan, Y.; Tian, K.; Jia, G.; Li, P.; et al. A Survey of Reinforcement Learning for Large Reasoning Models, 2025, [arXiv:cs.CL/2509.08827].

Wang, J.; Zhang, Z.; He, Y.; Zhang, Z.; Song, X.; Song, Y.; Shi, T.; Li, Y.; Xu, H.; Wu, K.; et al. Enhancing Code LLMs with Reinforcement Learning in Code Generation: A Survey, 2025, [arXiv:cs.SE/2412.20367].

Zhang, G.; Geng, H.; Yu, X.; Yin, Z.; Zhang, Z.; Tan, Z.; Zhou, H.; Li, Z.Z.; Xue, X.; Li, Y.; et al. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. *Transactions on Machine Learning Research* **2026**. Survey Certification.

Liu, J.; Dhole, K.; Wang, Y.; Wen, H.; Zhang, S.; Mao, H.; Li, G.; Varshney, N.; Liu, J.; Pan, X. Toward Honest Language Models for Deductive Reasoning, 2025, [arXiv:cs.CL/2511.09222].

Qu, Y.; Setlur, A.; Smith, V.; Salakhutdinov, R.; Kumar, A. POPE: Learning to Reason on Hard Problems via Privileged On-Policy Exploration, 2026, [arXiv:cs.LG/2601.18779].

Nie, S.; Ding, S.; Zhang, W.; Yu, L.; Yang, T.; Chen, Y.; Yin, W.; Sun, Y.; Wu, H.; Liu, T. ATTNPO: Attention-Guided Process Supervision for Efficient Reasoning, 2026, [arXiv:cs.CL/2602.09953].

Ai, Z.; Shan, Z.; Ai, X.; Tang, J.; Hu, H.; Lu, P. SHAPE: Stage-aware Hierarchical Advantage via Potential Estimation for LLM Reasoning, 2026, [arXiv:cs.LG/2604.06636].

Zheng, C.; Zhu, J.; Ou, Z.; Chen, Y.; Zhang, K.; Shan, R.; Zheng, Z.; Yang, M.; Lin, J.; Yu, Y.; et al. A Survey of Process Reward Models: From Outcome Signals to Process Supervisions for Large Language Models, 2026, [arXiv:cs.CL/2510.08049].

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Dai, W.; Fan, T.; Liu, G.; Liu, L.; et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale, 2025, [arXiv:cs.LG/2503.14476].

Li, Y.; Gu, Q.; Wen, Z.; Li, Z.; Xing, T.; Guo, S.; Zheng, T.; Zhou, X.; Qu, X.; Zhou, W.; et al. TreePO: Bridging the Gap of Policy Optimization and Efficacy and Inference Efficiency with Heuristic Tree-based Modeling, 2025, [arXiv:cs.LG/2508.17445].

Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Yue, Y.; Song, S.; Huang, G. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?, 2025, [arXiv:cs.AI/2504.13837].

Yan, J.; Li, Y.; Hu, Z.; Wang, Z.; Cui, G.; Qu, X.; Cheng, Y.; Zhang, Y. Learning to Reason under Off-Policy Guidance. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026.

Huang, Z.; Cheng, T.; Qiu, Z.; Wang, Z.; Xu, Y.; Ponti, E.M.; Titov, I. Blending Supervised and Reinforcement Fine-Tuning with Prefix Sampling, 2025, [arXiv:cs.LG/2507.01679].

Zhang, X.; Wu, S.; Zhu, Y.; Tan, H.; Yu, S.; He, Z.; Jia, J. Scaf-GRPO: Scaffolded Group Relative Policy Optimization for Enhancing LLM Reasoning. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Yu, L.; Yang, T.; Ding, S.; Jin, R.; Gu, N.; Hao, X.; Nie, S.; Xiong, D.; Yin, W.; Sun, Y.; et al. KnowRL: Boosting LLM Reasoning via Reinforcement Learning with Minimal-Sufficient Knowledge Guidance, 2026, [arXiv:cs.AI/2604.12627].

Lu, Z.; Yao, Z.; Wu, J.; Han, C.; Gu, Q.; Cai, X.; Lu, W.; Xiao, J.; Zhuang, Y.; Shen, Y. SKILL0: In-Context Agentic Reinforcement Learning for Skill Internalization, 2026, [arXiv:cs.LG/2604.02268].

Xia, P.; Chen, J.; Wang, H.; Liu, J.; Zeng, K.; Wang, Y.; Han, S.; Zhou, Y.; Zhao, X.; Chen, H.; et al. SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning, 2026, [arXiv:cs.LG/2602.08234].

 doi:10.20944/preprints202606.1050.v1



Li, J.; Lin, H.; Lu, H.; Wen, K.; Yang, Z.; Gao, J.; Wu, Y.; Zhang, J. QuestA: Expanding Reasoning Capacity in LLMs via Question Augmentation. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Wu, M.; Qian, Q.; Liu, W.; Wang, X.; Huang, Z.; Liang, D.; Miao, L.; Dou, S.; Lv, C.; Wang, Z.; et al. Progressive Mastery: Customized Curriculum Learning with Guided Prompting for Mathematical Reasoning, 2025, [arXiv:cs.CL/2506.04065].

Li, Z.; Sun, Z.; Zhao, J.; Min, E.; Zeng, Y.; Wu, H.; Cai, H.; Wang, S.; Yin, D.; Chen, X.; et al. Staying in the Sweet Spot: Responsive Reasoning Evolution via Capability-Adaptive Hint Scaffolding, 2025, [arXiv:cs.LG/2509.06923].

Liu, Z.; Gong, C.; Fu, X.; Liu, Y.; Chen, R.; Hu, S.; Zhang, S.; Liu, R.; Zhang, Q.; Tu, D. GHPO: Adaptive Guidance for Stable and Efficient LLM Reinforcement Learning, 2025, [arXiv:cs.LG/2507.10628].

Zhang, X.; Huang, Z.; Li, Y.; Ni, C.; Chen, J.; Oymak, S. BREAD: Branched Rollouts from Expert Anchors Bridge SFT & RL for Reasoning. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026.

Qiyuan, D.; Chen, K.; Zhang, M.; Xu, Z. HiPO: Self-Hint Policy Optimization for RLVR. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Huang, B.; Wan, X. PROS: Towards Compute-Efficient RLVR via Rollout Prefix Reuse. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Yi, H.; Wang, X.; zhang, Z.; Zong, T.; Wang, Y.; Xie, J.; Yu, T.; Jin, H.; Xu, K.; Chen, F.; et al. RPO:Reinforcement Fine-Tuning with Partial Reasoning Optimization, 2026, [arXiv:cs.AI/2601.19404].

Kim, M.; Shrestha, S.; Ross, K. Training Reasoning Models on Saturated Problems via Failure-Prefix Conditioning, 2026, [arXiv:cs.LG/2601.20829].

Mishra, K.; Aubakirov, M.; Takac, M.; Lukas, N.; Lahlou, S. CORE: Collaborative Reasoning via Cross Teaching, 2026, [arXiv:cs.AI/2601.21600].

Liu, M.; Farina, G.; Ozdaglar, A.E. UFT: Unifying Supervised and Reinforcement Fine-Tuning. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026.

Liu, H.; Li, J.; Dong, Y.; Yu, C.; Chen, T.; Wang, L.; Tao, Y.; Gu, B.; Li, G. EvoCoT: Overcoming the Exploration Bottleneck in Reinforcement Learning, 2026, [arXiv:cs.LG/2508.07809].

Guo, Y.; Deng, W.; Cheng, Z.; Tang, X. $G^2$RPO-A: Guided Group Relative Policy Optimization with Adaptive Guidance, 2025, [arXiv:cs.AI/2508.13023].

Zhang, F.; Tan, Z.; Ma, X.; Dong, Z.; Leng, X.; Zhao, J.; Sun, X.; Yang, Y. ADHint: Adaptive Hints with Difficulty Priors for Reinforcement Learning, 2026, [arXiv:cs.CV/2512.13095].

Huang, Q.; Dai, W.; Liu, J.; He, W.; Jiang, H.; Song, M.; Chen, J.; Yao, C.; Song, J. Boosting MLLM Reasoning with Text-Debiased Hint-GRPO. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2025, pp. 4848–4857.

Xie, P.X.; Lin, C.Y.; Yang, C.L. Mitigating Distribution Sharpening in Math RLVR via Distribution-Aligned Hint Synthesis and Backward Hint Annealing, 2026, [arXiv:cs.AI/2604.07747].

Su, M.; Guan, J.; Gu, Y.; Huang, M.; Wang, H. Trust-Region Adaptive Policy Optimization. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Zhang, K.; Lv, A.; Li, J.; Wang, Y.; Wang, F.; Hu, H.; Yan, R. StepHint: Multi-level Stepwise Hints Enhance Reinforcement Learning to Reason, 2025, [arXiv:cs.AI/2507.02841].

Yuan, X.; Ding, Y.; Bin, Y.; Shao, W.; Cai, J.; Song, J.; Yang, Y.; Shen, H.T. More Than One Teacher: Adaptive Multi-Guidance Policy Optimization for Diverse Exploration, 2025, [arXiv:cs.CL/2510.02227].

Yan, Y.; Tang, K.; Chen, S.; Xu, K.; Hu, D.; Yu, Q.; Hu, P. S-GRPO: Unified Post-Training for Large Vision-Language Models, 2026, [arXiv:cs.LG/2604.16557].

Wu, Y.; Wang, K.; Chen, D.; Wei, K. Hindsight-Anchored Policy Optimization: Turning Failure into Feedback in Sparse Reward Settings, 2026, [arXiv:cs.LG/2603.11321].

Ren, Y.; Zhang, H.; Xiao, L.; Zhang, X.; Huang, J.; Qiu, J.; Yu, B.; Chen, Q.; Liu, L. Recycling Failures: Salvaging Exploration in RLVR via Fine-Grained Off-Policy Guidance, 2026, [arXiv:cs.AI/2602.24110].

Zhou, R.; Li, S.; Zhang, A.; Leqi, L. ExPO: Unlocking Hard Reasoning with Self-Explanation-Guided Reinforcement Learning. In Proceedings of the The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026.

Jiang, Z.; Han, J.; tingyun li.; Wang, X.; Jiang, S.; Dai, Z.; Shuguang, M.; Yu, F.; Liang, J.; Xiao, Y. Selective Expert Guidance for Effective and Diverse Exploration in Reinforcement Learning of LLMs. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

 doi:10.20944/preprints202606.1050.v1



Lin, T.; Zhao, X.; Zhang, X.; Long, R.; Xu, Y.; Jiang, Z.; Su, W.; Zheng, B. RAVR: Reference-Answer-guided Variational Reasoning for Large Language Models, 2025, [arXiv:cs.AI/2510.25206].

Wen, X.; Lou, J.; Liu, Y.; Lin, H.; He, B.; Han, X.; Sun, L.; Lu, Y.; Zhang, D. Coupled Variational Reinforcement Learning for Language Model General Reasoning, 2026, [arXiv:cs.CL/2512.12576].

Ding, K. HDPO: Hybrid Distillation Policy Optimization via Privileged Self-Distillation, 2026, [arXiv:cs.LG/2603.23871].

Nath, V.; Lau, E.; Gunjal, A.; Sharma, M.; Baharte, N.; Hendryx, S. Adaptive Guidance Accelerates Reinforcement Learning of Reasoning Models, 2025, [arXiv:cs.LG/2506.13923].

Kulkarni, Y.; Fazli, P. AVATAR: Reinforcement Learning to See, Hear, and Reason Over Video, 2026, [arXiv:cs.CV/2508.03100].

Fang, Y.; Lin, J.; Fu, X.; Qin, C.; Shi, H. Placing Puzzle Pieces Where They Matter: A Question Augmentation Framework for Reinforcement Learning, 2026, [arXiv:cs.LG/2604.15830].

Qu, Y.; Singh, A.; Lee, Y.; Setlur, A.; Salakhutdinov, R.; Finn, C.; Kumar, A. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems, 2025, [arXiv:cs.AI/2510.02263].

Chen, Z.; Qin, X.; Zhao, W.X.; Wu, Y.; Wen, J.R. Adaptive Ability Decomposing for Unlocking Large Reasoning Model Effective Reinforcement Learning, 2026, [arXiv:cs.CL/2602.00759].

Liao, B.; Dong, H.; Xu, X.; Monz, C.; Bian, J. Self-Hinting Language Models Enhance Reinforcement Learning, 2026, [arXiv:cs.LG/2602.03143].

Chen, J.; PENG, X.; Choubey, P.K.; Huang, K.H.; Zhang, J.; Bansal, M.; Wu, C.S. Nudging the Boundaries of LLM Reasoning. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Zhang, J.; Ma, G.; Liu, S.; Wang, H.; Huang, J.; Lin, T.E.; Huang, F.; Li, Y.; Tao, D. A Simple "Motivation" Can Enhance Reinforcement Finetuning of Large Reasoning Models, 2026, [arXiv:cs.CL/2506.18485].

Zhou, Y.; Li, S.; Liu, S.; Fang, W.; Zhang, K.; Zhao, J.; Yang, J.; Zhou, Y.; Lv, J.; Zheng, T.; et al. Breaking the Exploration Bottleneck: Rubric-Scaffolded Reinforcement Learning for General LLM Reasoning, 2026, [arXiv:cs.LG/2508.16949].

Wang, X.; Han, J.; Jiang, Z.; Li, T.; Liang, J.; Jiang, S.; Dai, Z.; Ma, S.; Yu, F.; Xiao, Y. HINT: Helping Ineffective Rollouts Navigate Towards Effectiveness, 2025, [arXiv:cs.LG/2510.09388].

Yang, F.; Meng, R.; Qi, T.D.; Ezzati, A.; Wen, Y. KEPO: Knowledge-Enhanced Preference Optimization for Reinforcement Learning with Reasoning, 2026, [arXiv:cs.AI/2602.00400].

Xia, Y.; Xu, C.; Yao, Z.; McAuley, J.; He, Y. Learning to Hint for Reinforcement Learning, 2026, [arXiv:cs.LG/2604.00698].

Tang, C.; Huang, H.Y.; Liu, W.; Bai, C.; Yang, S.; Wu, Y. Do Not Step Into the Same River Twice: Learning to Reason from Trial and Error, 2026, [arXiv:cs.LG/2510.26109].

Bi, B.; Liu, S.; Wang, Y.; Tong, S.; Mei, L.; Ge, Y.; Xu, Y.; Guo, J.; Cheng, X. Reward and Guidance through Rubrics: Promoting Exploration to Improve Multi-Domain Reasoning, 2025, [arXiv:cs.AI/2511.12344].

Huang, L.; Cheng, X.; Zhao, C.; Shen, G.; Yang, J.; Feng, X.; Gu, Y.; Yu, X.; Qin, B. Bootstrapping Exploration with Group-Level Natural Language Feedback in Reinforcement Learning, 2026, [arXiv:cs.CL/2603.04597].

Shi, W.; Chen, Y.; Li, Z.; Pan, X.; Sun, Y.; Xu, J.; Zhou, X.; Li, Y. $R^3L$: Reflect-then-Retry Reinforcement Learning with Language-Guided Exploration, Pivotal Credit, and Positive Amplification, 2026, [arXiv:cs.LG/2601.03715].

Li, Z.; Jiang, L.; Hu, Y.; Zeng, X.; Li, Y.; Zhang, X.; Chen, G.; Pan, Z.; Li, X.; Liu, Y. No More Stale Feedback: Co-Evolving Critics for Open-World Agent Learning, 2026, [arXiv:cs.AI/2601.06794].

Luo, K.; Qian, H.; Liu, Z.; Xia, Z.; Xiao, S.; Bao, S.; Zhao, J.; Liu, K. InfoFlow: Reinforcing Search Agent Via Reward Density Optimization, 2025, [arXiv:cs.CL/2510.26575].

Huang, S.; Li, Z.; Zeng, Y.; Ren, Q.; Fang, Z.; Su, Q.; Shi, K.; Chen, L.; Chen, Z.; Zhao, F. Internalizing Meta-Experience into Memory for Guided Reinforcement Learning in Large Language Models, 2026, [arXiv:cs.LG/2602.10224].

Rrv, A.; Dineen, J.; Handa, D.; Uddin, M.N.; Parmar, M.; Baral, C.; Zhou, B. ThinkTuning: Instilling Cognitive Reflections without Distillation. In Proceedings of the Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing; Christodoulopoulos, C.; Chakraborty, T.; Rose, C.; Peng, V., Eds., Suzhou, China, 2025; pp. 31248–31262. https://doi.org/10.18653/v1/2025.emnlp-main.1592.

Song, Y.; Chen, L.; Tajwar, F.; Munos, R.; Pathak, D.; Bagnell, J.A.; Singh, A.; Zanette, A. Expanding the Capabilities of Reinforcement Learning via Text Feedback, 2026, [arXiv:cs.LG/2602.02482].

Agashe, S.; Srinivasa, J.; Liu, G.; Kompella, R.; Wang, X.E. Context Bootstrapped Reinforcement Learning, 2026, [arXiv:cs.LG/2603.18953].

 doi:10.20944/preprints202606.1050.v1



Huang, H.Y.; Tang, C.; Liu, W.; Bai, C.; Yang, S.; Wu, Y. Think Outside the Policy: In-Context Steered Policy Optimization, 2026, [arXiv:cs.LG/2510.26519].

Wu, J.; Liao, C.; Feng, M.; Zhang, S.; Wen, Z.; Luo, H.; Yang, L.; Xu, H.; Tao, J. TemplateRL: Structured Template-Guided Reinforcement Learning for LLM Reasoning, 2025, [arXiv:cs.CL/2505.15692].

Wu, R.; Wang, X.; Mei, J.; Cai, P.; Fu, D.; Yang, C.; Wen, L.; Yang, X.; Shen, Y.; Wang, Y.; et al. EvolveR: Self-Evolving LLM Agents through an Experience-Driven Lifecycle, 2025, [arXiv:cs.CL/2510.16079].

Zhang, X.; Liu, Z.; Zhang, Y.; Hu, X.; Shao, W. RetroAgent: From Solving to Evolving via Retrospective Dual Intrinsic Feedback, 2026, [arXiv:cs.AI/2603.08561].

Shi, T.; Chen, S.; Jiang, B.; Song, L.; Yang, L.; Zhao, J. Experiential Reinforcement Learning, 2026, [arXiv:cs.LG/2602.13949].

lu Yang.; Xu, Z.; Xie, M.; Gao, J.; zhao shok.; Wang, Y.; Wu, Y. MAGE: Meta-Reinforcement Learning for Language Agents toward Strategic Exploration and Exploitation. In Proceedings of the ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving, 2026.

Szot, A.; Kirchhof, M.; Attia, O.; Toshev, A. Expanding LLM Agent Boundaries with Strategy-Guided Exploration, 2026, [arXiv:cs.LG/2603.02045].

Lai, Y.; Yang, Y.; Wu, J.; Mo, F.; Wang, Z.; Liang, T.; Lin, J.; Yang, K. CRMWeaver: Building Powerful Business Agent via Agentic RL and Shared Memories, 2025, [arXiv:cs.CL/2510.25333].

Liu, G.; Wu, M.; Zhang, P.; Zhang, Y.; Shu, Y.; Huang, X.; Tu, K.; Gu, N.; Zhang, L.; Wang, Q.; et al. IntPro: A Proxy Agent for Context-Aware Intent Understanding via Retrieval-conditioned Inference, 2026, [arXiv:cs.CL/2603.03325].

Xia, P.; Chen, J.; Yang, X.; Tu, H.; Liu, J.; Xiong, K.; Han, S.; Qiu, S.; Ji, H.; Zhou, Y.; et al. MetaClaw: Just Talk – An Agent That Meta-Learns and Evolves in the Wild, 2026, [arXiv:cs.LG/2603.17187].

Li, Y.; Miao, R.; Qi, Z.; Lan, T. ARISE: Agent Reasoning with Intrinsic Skill Evolution in Hierarchical Reinforcement Learning, 2026, [arXiv:cs.AI/2603.16060].

Wu, X.; Li, Z.; Shi, G.; Duffy, A.; Marques, T.; Olson, M.L.; Zhou, T.; Manocha, D. Co-Evolving LLM Decision and Skill Bank Agents for Long-Horizon Tasks, 2026, [arXiv:cs.AI/2604.20987].

Wang, J.; Yan, Q.; Wang, Y.; Tian, Y.; Mishra, S.S.; Xu, Z.; Gandhi, M.; Xu, P.; Cheong, L.L. Reinforcement Learning for Self-Improving Agent with Skill Library, 2026, [arXiv:cs.AI/2512.17102].

Wu, Z.; Mo, D.; Lu, H.; Xing, J.; Liu, J.; Jing, Y.; Li, K.; Shao, K.; HAO, J.; Shi, Y. K²-Agent: Co-Evolving Know-What and Know-How for Hierarchical Mobile Device Control. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Wang, Z.; Zhang, Z.; Zhang, X.; Qian, Z.; Lu, Y. From Off-Policy to On-Policy: Enhancing GUI Agents via Bi-level Expert-to-Policy Assimilation, 2026, [arXiv:cs.AI/2601.05787].

Bai, F.; Chen, Z.; Hao, C.; Yang, M.; Tao, R.; Dai, B.; Zhao, W.X.; Yang, J.; Xu, H. Towards Effective Experiential Learning: Dual Guidance for Utilization and Internalization, 2026, [arXiv:cs.LG/2603.24093].

Muhtar, D.; Liu, J.; Gao, W.; Wang, W.; Xiong, S.; Huang, J.; Yang, S.; Su, W.; Wang, J.; Pan, L.; et al. Complementary Reinforcement Learning, 2026, [arXiv:cs.LG/2603.17621].

Zhou, H.; Wan, X.; Vulić, I.; Korhonen, A. Agentic Policy Optimization via Instruction-Policy Co-Evolution, 2026, [arXiv:cs.LG/2512.01945].

Zhai, Y.; Tao, S.; Chen, C.; Zou, A.; Chen, Z.; Fu, Q.; Mai, S.; Yu, L.; Deng, J.; Cao, Z.; et al. AgentEvolver: Towards Efficient Self-Evolving Agent System, 2025, [arXiv:cs.LG/2511.10395].

Li, B.; Kim, J.; Qian, C.; Chen, X.; Anzenberg, E.; Kundapur, N.; Ji, H. PEARL: Self-Evolving Assistant for Time Management with Reinforcement Learning, 2026, [arXiv:cs.CL/2601.11957].

Tu, S.; Xu, C.; Zhang, Q.; Zhang, Y.; Lan, X.; Li, L.; Zhao, D. Dynamic Dual-Granularity Skill Bank for Agentic RL, 2026, [arXiv:cs.AI/2603.28716].

Wang, P.Z.; Jiang, S. SLEA-RL: Step-Level Experience Augmented Reinforcement Learning for Multi-Turn Agentic Training, 2026, [arXiv:cs.LG/2603.18079].

Wang, H.; Wang, G.; Xiao, H.; Zhou, Y.; Pan, Y.; Wang, J.; Xu, K.; Wen, Y.; Ruan, X.; Chen, X.; et al. Skill-SD: Skill-Conditioned Self-Distillation for Multi-turn LLM Agents, 2026, [arXiv:cs.LG/2604.10674].

Ye, Y.; Jiang, H.; Jiang, F.; Lan, T.; Du, Y.; Fu, B.; Shi, X.; Jia, Q.; Wang, L.; Luo, W. UMEM: Unified Memory Extraction and Management Framework for Generalizable Memory, 2026, [arXiv:cs.CL/2602.10652].

Xia, S.; Xu, Z.; Chai, J.; Fan, W.; Song, Y.; Wang, X.; Yin, G.; Lin, W.; Zhang, H.; Wang, J. From Experience to Strategy: Empowering LLM Agents with Trainable Graph Memory, 2025, [arXiv:cs.CL/2511.07800].

Baronio, C.; Marsella, P.; Pan, B.; Guo, S.; Alberti, S. Kevin: Multi-Turn RL for Generating CUDA Kernels, 2025, [arXiv:cs.LG/2507.11948].

 doi:10.20944/preprints202606.1050.v1



Bailey, L.; Wen, K.; Dong, K.; Hashimoto, T.; Ma, T. Scaling Self-Play with Self-Guidance, 2026, [arXiv:cs.LG/2604.20209].

Luo, M.; Yang, Z.; Long, J.; Sun, J.; Liu, Y.; Mai, S. C2F-Thinker: Coarse-to-Fine Reasoning with Hint-Guided Reinforcement Learning for Multimodal Sentiment Analysis, 2026, [arXiv:cs.CL/2604.00013].

Lu, Z.; Ren, H.; Yang, Y.; Wang, K.; Zong, Z.; Pan, J.; Zhan, M.; Li, H. WebGen-Agent: Enhancing Interactive Website Generation with Multi-Level Feedback and Step-Level Reinforcement Learning, 2025, [arXiv:cs.CL/2509.22644].

Lu, Z.; Ye, J.; Tang, F.; Shen, Y.; Xu, H.; Zheng, Z.; Lu, W.; Yan, M.; Huang, F.; Xiao, J.; et al. UI-S1: Advancing GUI Automation via Semi-online Reinforcement Learning, 2025, [arXiv:cs.LG/2509.11543].

Yang, J.; Jin, Y.; Jiao, P.; Dong, C.; Huang, Z.; Yao, S.; Zhou, X.; Ou, D.; Tang, H. TaoSR-AGRL: Adaptive Guided Reinforcement Learning Framework for E-commerce Search Relevance. In Proceedings of the Proceedings of the ACM Web Conference 2026, New York, NY, USA, 2026; WWW '26, p. 7955–7966. https://doi.org/10.1145/3774904.3792799.

Wang, Z.; Sun, K.; Wu, B.; qun yu.; Li, Y.; Chen, X.; Wang, B. VeriRole: Verifiable Role-Awareness through Hint-Guided Reinforcement Learning. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Ji, Y.; Ma, Z.; Wang, Y.; Chen, G.; Chu, X.; Wu, L. Tree Search for LLM Agent Reinforcement Learning. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Dong, G.; Mao, H.; Ma, K.; Bao, L.; Chen, Y.; Wang, Z.; Chen, Z.; Du, J.; Wang, H.; Zhang, F.; et al. Agentic Reinforced Policy Optimization. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Guo, Y.; Hu, T.; Sun, Z.; Lin, Y. Less Noise, More Voice: Reinforcement Learning for Reasoning via Instruction Purification, 2026, [arXiv:cs.LG/2601.21244].

Zhang, K.; Yao, Q.; Liu, S.; Zhang, W.; Cen, M.; Zhou, Y.; Fang, W.; Zhao, Y.; Lai, B.; Song, M. Replay Failures as Successes: Sample-Efficient Reinforcement Learning for Instruction Following, 2025, [arXiv:cs.AI/2512.23457].

Chen, J.C.Y.; Prasad, A.; Khan, Z.; Singh, J.; Tian, R.; Stengel-Eskin, E.; Bansal, M. Cog-DRIFT: Exploration on Adaptively Reformulated Instances Enables Learning from Hard Reasoning Problems, 2026, [arXiv:cs.LG/2604.04767].

Nourzad, N.; Joe-Wong, C. MIRA: Memory-Integrated Reinforcement Learning Agent with Limited LLM Guidance. In Proceedings of the The Fourteenth International Conference on Learning Representations, 2026.

Xu, Y.; Potje, G.; Shandilya, S.; Yuan, T.; de Oliveira Nunes, L.; Agarwal, R.; Asgari, S.; Atkinson, A.; Kıcıman, E.; Lu, S.; et al. SibylSense: Adaptive Rubric Learning via Memory Tuning and Adversarial Probing, 2026, [arXiv:cs.CL/2602.20751].

Sheng, L.; Ma, W.; Hong, R.; Wang, X.; Zhang, A.; Chua, T.S. Reinforcing Chain-of-Thought Reasoning with Self-Evolving Rubrics, 2026, [arXiv:cs.AI/2602.10885].

Gu, N.; Yang, C.; Si, Q.; Qin, C.; Yao, D.; Fu, P.; Lin, Z.; Wang, W.; Duan, N.; Wang, J. Co-Evolving Policy Distillation, 2026, [arXiv:cs.LG/2604.27083].

Yang, C.; Qin, C.; Si, Q.; Chen, M.; Gu, N.; Yao, D.; Lin, Z.; Wang, W.; Wang, J.; Duan, N. Self-Distilled RLVR, 2026, [arXiv:cs.LG/2604.03128].

Song, M.; Zheng, M. A Survey of On-Policy Distillation for Large Language Models, 2026, [arXiv:cs.LG/2604.00626].