How Far Can Unsupervised RLVR Scale LLM Training?

Bingxiang He | THUNLP | Advisor: Prof. Zhiyuan Liu

Homepage: https://hbx-hbx.github.io/

2026.04.18

How Far Can Unsupervised RLVR Scale LLM Training?

Bingxiang He[*1], **Yuxin Zuo**[*†1,2], **Zeyuan Liu**[*1], **Shangziqi Zhao**[*3], **Zixuan Fu**[1], **Junlin Yang**[1], **Cheng Qian**[4], **Kaiyan Zhang**[1,5], **Yuchen Fan**[6], **Ganqu Cui**[2], **Xiusi Chen**[4], **Youbang Sun**[1], **Xingtai Lv**[1], **Xuekai Zhu**[6], **Li Sheng**[1], **Ran Li**[1], **Huan-ang Gao**[1], **Yuchen Zhang**[7], **Bowen Zhou**[‡1,2], **Zhiyuan Liu**[‡1], **Ning Ding**[‡1,2]

[1]Tsinghua University [2]Shanghai AI Lab [3]Xi'an Jiaotong University [4]University of Illinois Urbana-Champaign
[5]Frontis.AI [6]Shanghai Jiao Tong University [7]Peking University

[*]**Equal Contribution. Orders are determined randomly.** [†]**Project Lead.** [‡]**Corresponding Authors.**

https://github.com/PRIME-RL/TTRL.

hebx24@mails.tsinghua.edu.cn, dingning@mail.tsinghua.edu.cn

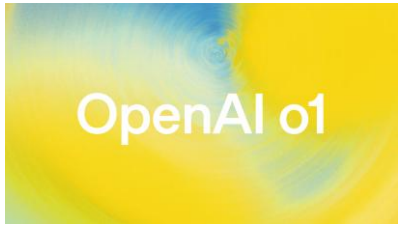
Full Paper **Code** **Homepage**

Outline

- **Background**
- **Taxonomy of Unsupervised RLVR**
- **The Sharpening Mechanism**
- **Experiments**

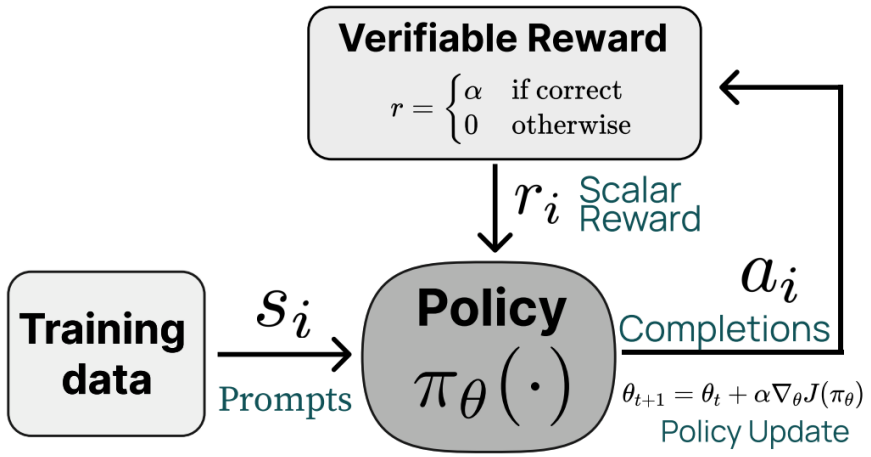
Background

- **2025 is a year of reasoning models trained via large scale RL**
  - Extended thinking through long CoT
  - SoTA performance on challenging reasoning benchmarks



Background

- **Reward is the core component of RL**
  - **Verifiable domain**: coding & math & physics, rule-based verifier
  - **Open domain**: creative writing & open QA, model-based verifier



2.2.2. Reward Modeling

The reward is the source of the training signal, which decides the optimization direction of RL. To train DeepSeek-R1-Zero, we adopt a rule-based reward system that mainly consists of two types of rewards:

- **Accuracy rewards**: The accuracy reward model evaluates whether the response is correct. For example, in the case of math problems with deterministic results, the model is required to provide the final answer in a specified format (e.g., within a box), enabling reliable rule-based verification of correctness. Similarly, for LeetCode problems, a compiler can be used to generate feedback based on predefined test cases.
- **Format rewards**: In addition to the accuracy reward model, we employ a format reward model that enforces the model to put its thinking process between '<think>' and '</think>' tags.

We do not apply the outcome or process neural reward model in developing DeepSeek-R1-Zero, because we find that the neural reward model may suffer from reward hacking in the large-scale reinforcement learning process, and retraining the reward model needs additional training resources and it complicates the whole training pipeline.

Tülu 3: Pushing Frontiers in Open Language Model Post-Training.
DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.

Background

- **Reward is the core component of RL**
  - **Verifiable domain**: coding & math & physics, rule-based verifier
  - **Open domain**: creative writing & open QA, model-based verifier

```
[System]
Please act as an impartial judge and evaluate the quality of the responses provided by two
AI assistants to the user question displayed below. You should choose the assistant that
follows the user's instructions and answers the user's question better. Your evaluation
should consider factors such as the helpfulness, relevance, accuracy, depth, creativity,
and level of detail of their responses. Begin your evaluation by comparing the two
responses and provide a short explanation. Avoid any position biases and ensure that the
order in which the responses were presented does not influence your decision. Do not allow
the length of the responses to influence your evaluation. Do not favor certain names of
the assistants. Be as objective as possible. After providing your explanation, output your
final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]"
if assistant B is better, and "[[C]]" for a tie.

[User Question]
{question}

[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

Figure 5: The default prompt for pairwise comparison.

```
[System]
Please act as an impartial judge and evaluate the quality of the response provided by an
AI assistant to the user question displayed below. Your evaluation should consider factors
such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of
the response. Begin your evaluation by providing a short explanation. Be as objective as
possible. After providing your explanation, please rate the response on a scale of 1 to 10
by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]
```

Figure 6: The default prompt for single answer grading.

Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena

Background

- **Scaling supervision requires prohibitively high human costs**

Benchmark	Nemotron-3-Nano 30B-A3B	Nemotron-3-Super 120B-A12B	Qwen3.5 35B-A3B	Nemotron-Cascade-2 30B-A3B
Math				
IMO 2025	–	–	–	🥇 **35 pts**
IMO AnswerBench	70.4‡	77.2‡	74.8‡	79.3
IMO ProofBench	–	–	–	72.9
AIME 2025	89.1	90.2	91.9‡	92.4 (98.6)†
AIME 2026	89.9‡	89.8‡	91.1‡	90.9 (95.0)†
HMMT Feb25	84.6‡	93.7	89.0	94.6
Code Reasoning				
IOI 2025	–	–	348.6‡	🥇 **439.28**
ICPC World Finals 2025	–	–	–	🥇 **10/12**
LiveCodeBench v6 (2408-2505)	68.3	78.7	74.6	87.2 (88.4)†
LiveCodeBenchPro 25Q2 (Easy)	54.5‡	81.7‡	81.1‡	87.0 (89.3)†
LiveCodeBenchPro 25Q2 (Med)	3.50‡	23.2‡	17.8‡	27.6 (36.8)†
SciCode	33.3	42.1	38.0	36.4

Background

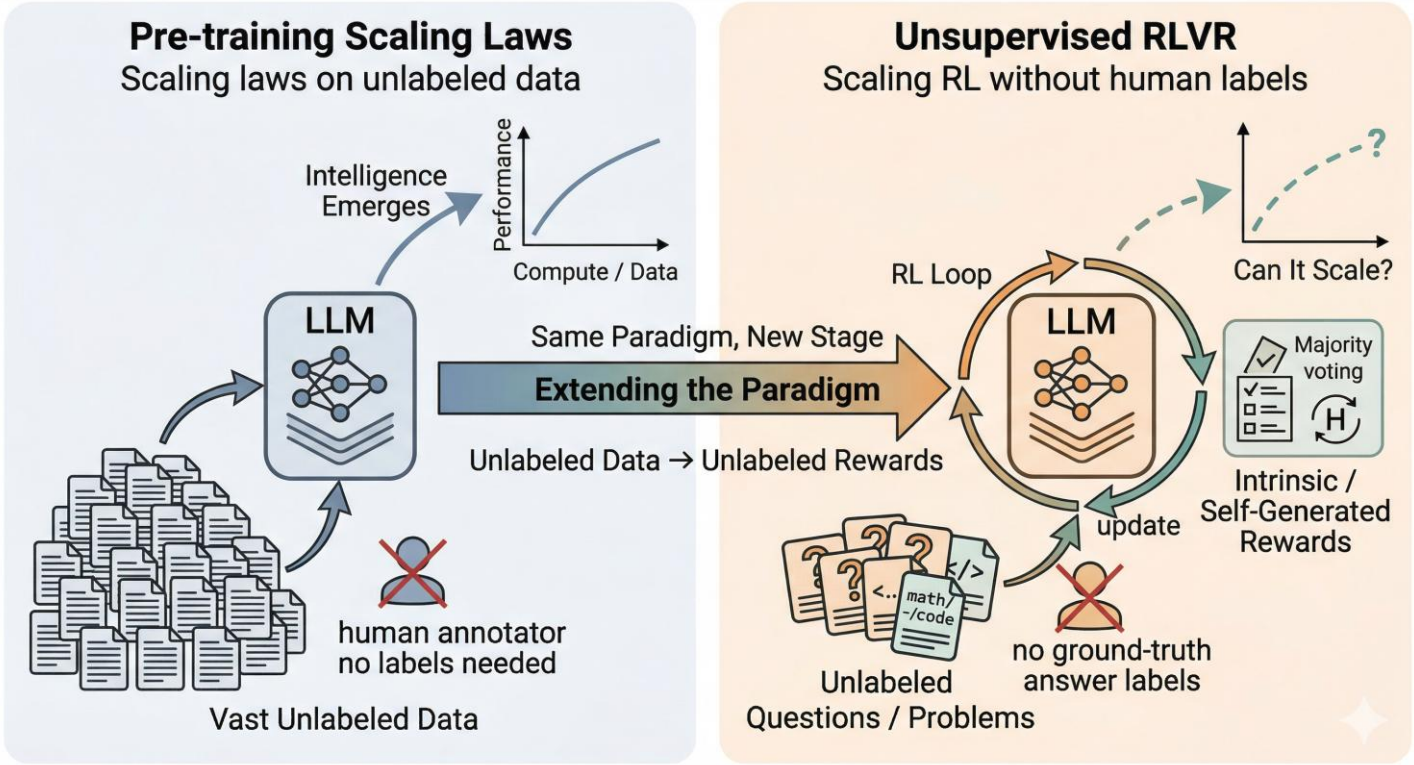
- **Scaling supervision requires prohibitively high human costs**

Curating Cold Start RL Data We constructed our initial training data through the following process:

1. We crawled problems from Art of Problem Solving (AoPS) contests [2], prioritizing math olympiads, team selection tests, and post-2010 problems explicitly requiring proofs, totaling 17,503 problems. This problem set is denoted as $\mathcal{D}_p$.
2. We generated candidate proofs using a variant of DeepSeek-V3.2-Exp-Thinking. As this model was not optimized for theorem proving and tended to produce concise but error-prone outputs, we prompted it to iteratively refine its proofs over multiple rounds to improve comprehensiveness and rigor.
3. We randomly sampled proofs across diverse problem types (e.g., algebra and number theory) and had mathematical experts score each proof according to the evaluation rubrics described above.

Background

- **Unsupervised RLVR (URLVR)**
  - TTRL, EMPO, RENT, Intuitor, etc
  - Derives rewards without ground truth labels

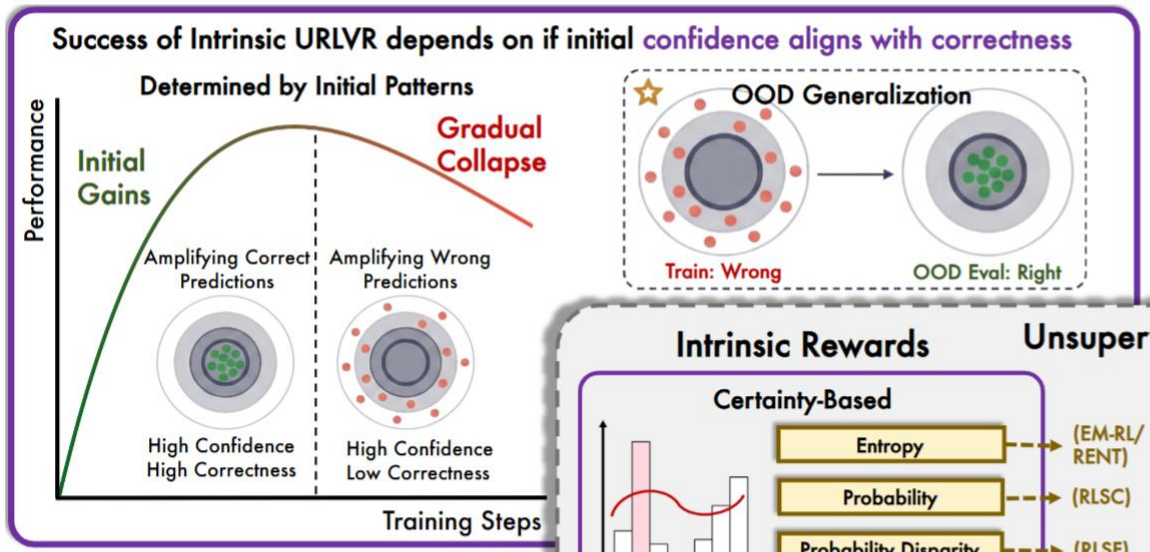


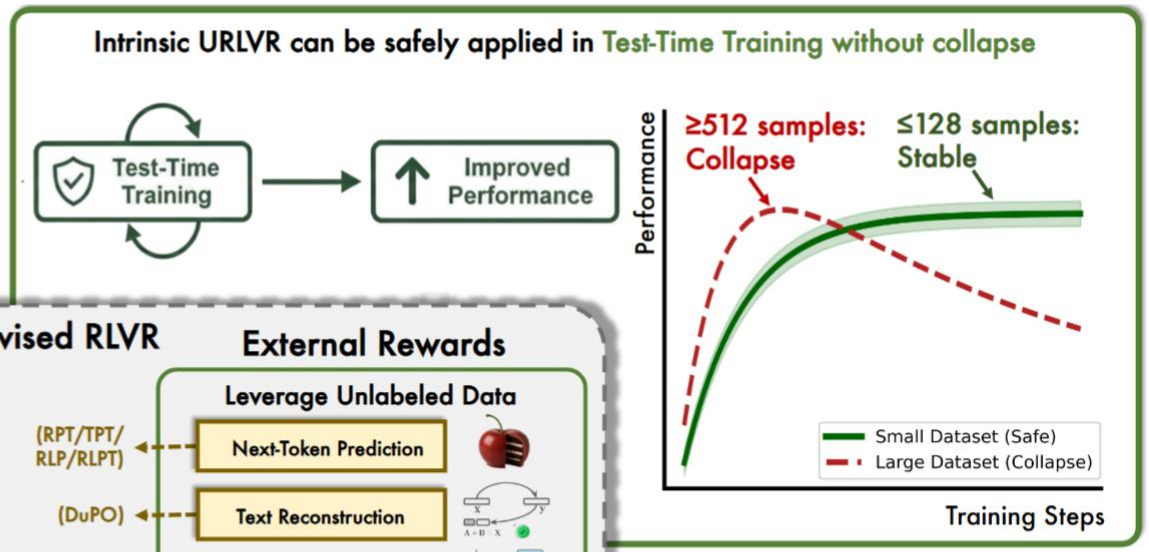
Background

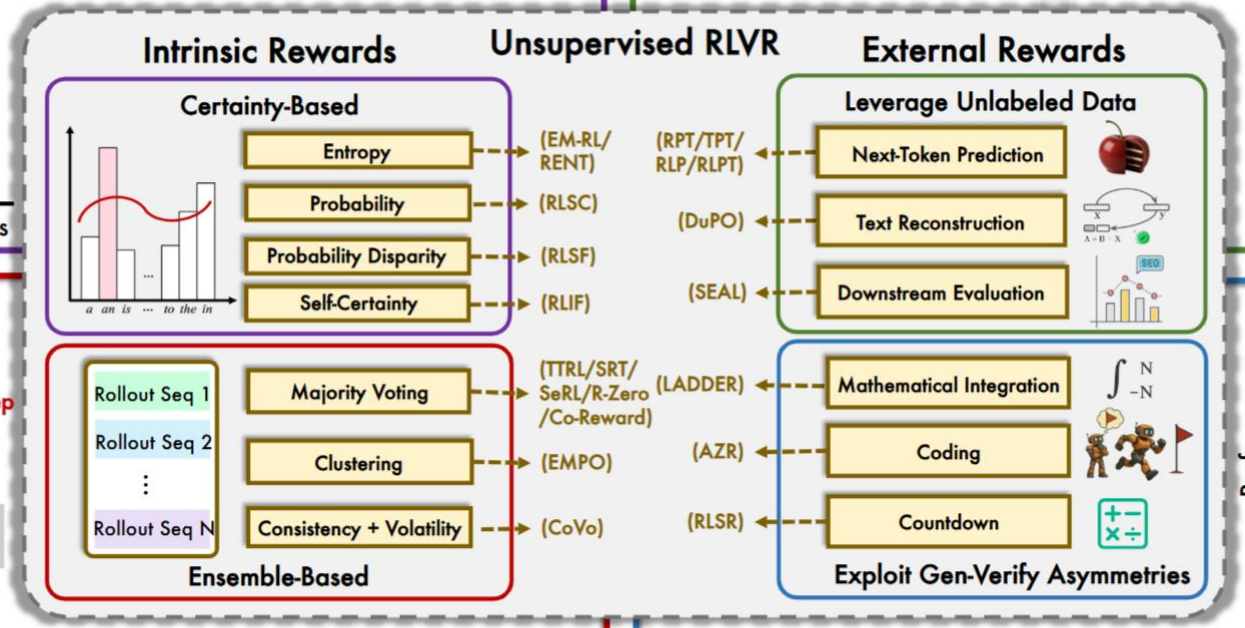
How Far Can Unsupervised RLVR Scale LLM Training?

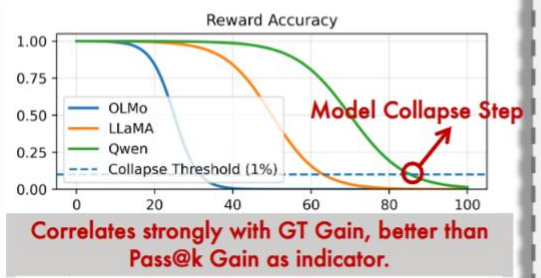
Background

How Far Can Unsupervised RLVR Scale LLM Training?

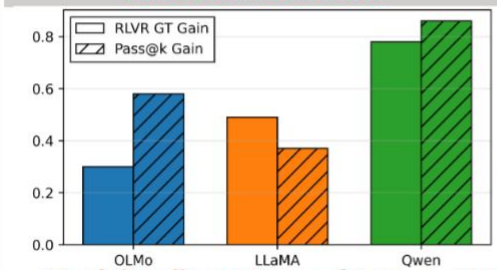


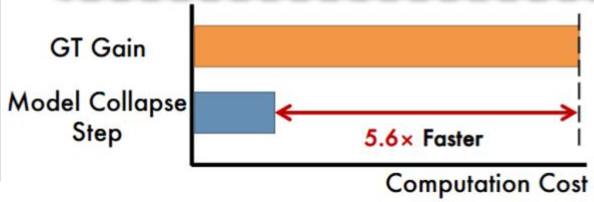






Correlates strongly with GT Gain, better than Pass@k Gain as indicator.





Model Collapse Step of Intrinsic URLVR can serve as indicator for RL trainability

Hard to generate

3 + 4 x 5 = ?

(3 + 4) x 5 = ?

3 x 4 + 5 = ?

Asymmetry of Verification

Easy to verify

3 + 4 x 5 = 23

(3 + 4) x 5 = 35

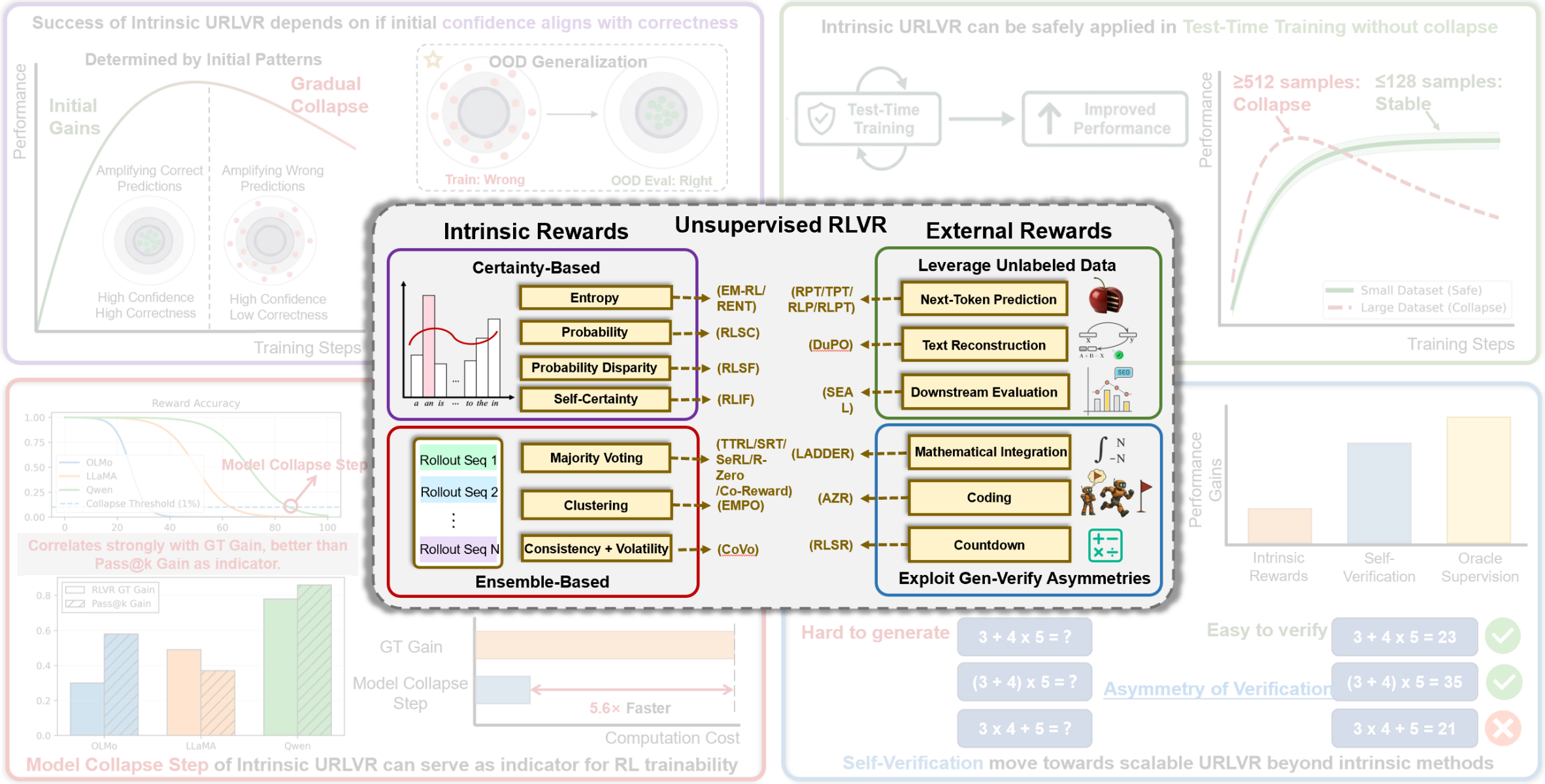
3 x 4 + 5 = 21

Self-Verification move towards scalable URLVR beyond intrinsic methods

Outline

- Background
- **Taxonomy of Unsupervised RLVR**
- The Sharpening Mechanism
- Experiments

Taxonomy of Unsupervised RLVR



Taxonomy of Unsupervised RLVR

Unsupervised RLVR

Problem Setting: We investigate reinforcement learning for verifiable tasks where ground-truth labels are difficult to obtain. In **Unsupervised RLVR**, models must learn from proxy reward signals derived without relying on human efforts.

- **Unsupervised RLVR** = Unsupervised RL + Verifiable Rewards (VR)
  - To precisely define the domain of tasks we are studying
  - To distinguish from general-domain self-rewarding methods

Intrinsic Reward Methods

- **Intrinsic rewards:** generated solely by the model itself
  - **Certainty-Based Rewards**: Derive a reward from policy’ s confidence (e.g., logits) along a trajectory, encouraging low-entropy, high-confidence predictions
  - **Ensemble-Based Rewards**: Derive a reward from agreement across multiple rollouts (e.g., majority voting), assuming that cross-sample consistency correlates with correctness

Intrinsic Reward Methods

- **Certainty-Based Rewards**: They are different mathematical formalizations for rewarding and reinforcing high-confidence predictions

Method	Estimator	Formula
RLIF	**Self-Certainty**	$r(x,y) = \frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} D_{\mathrm{KL}}(U \Vert \pi_\theta(\cdot\vert x, y_{<t}))$
EM-RL	**Trajectory-Level Entropy**	$r(x,y) = \frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} \log \pi_\theta(y_t\vert x, y_{<t})$
EM-RL, RENT	**Token-Level Entropy**	$r(x,y) = -\frac{1}{\lvert y\rvert}\sum_{t=1}^{\lvert y\rvert} H(\pi_\theta(\cdot\vert x, y_{<t}))$
RLSC	**Probability**	$r(x,y) = \prod_{t=1}^{\lvert y\rvert} \pi_\theta(y_t\vert x, y_{<t})$
RLSF	**Probability Disparity**	$r(x,y) = \frac{1}{M}\sum_{t=1}^{\lvert a\rvert} \left[\max_{a_t} \pi_\theta(a_t\vert x, c, a_{<t}) - \max_{a_t \neq \arg\max \pi_\theta} \pi_\theta(a_t\vert x, c, a_{<t})\right]$

Table 1 | Overview of certainty-based rewards, estimators and their formulas. All variants reward high-confidence predictions through different formalizations of model certainty.

Intrinsic Reward Methods

- **Ensemble-Based Rewards**: They assume that consistency across independent samples correlates with correctness.

Method	Estimator	Formula
TTRL, SRT, ETTRL SeRL, SQLM, R-Zero	Majority Voting	$r(x,y) = \mathbb{1}\left[y = \arg\max_{y'} \sum_{i=1}^{N} \mathbb{1}[y_i = y']\right], \quad \{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\|x)$
Co-Reward	Majority Voting across Rephrased Question	$r(x,y) = \mathbb{1}[y = \arg\max_{y^*} \sum_{i=1}^{N} \mathbb{1}[y_i = y^*]], \quad \{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\|x)$ $+ \mathbb{1}[y = \arg\max_{y^*} \sum_{j=1}^{N} \mathbb{1}[y'_j = y^*]], \quad \{y'_j\}_{j=1}^{N} \sim \pi_\theta(\cdot\|\text{rephrase}(x))$
RLCCF	Self-consistency Weighted Voting	$r(x,y) = \mathbb{1}\left[y = \arg\max_{a} \sum_{n=1}^{N} \left(\max_{a'} \sum_{k=1}^{K} \mathbb{1}[o_{n,k} = a']\right) \cdot \sum_{k=1}^{K} \mathbb{1}[a = o_{n,k}]\right],$ $\{o_{n,k}\}_{k=1}^{K} \sim \pi_{\theta_n}(\cdot\|x), \quad n = 1, \ldots, N$
EMPO	Semantic Similarity	$r(x,y) = \frac{\|C(y)\|}{G}, \quad C(y) \in \texttt{SemanticCluster}(\{o_i\}_{i=1}^{G}), \quad \{o_i\}_{i=1}^{G} \sim \pi_\theta(\cdot\|x)$
CoVo	Trajectory Consistency and Volatility	$r(x,y) = \frac{1}{G}\left\|\sum_{i=1}^{G} \texttt{Con}(y_i) \cdot [\cos(\texttt{Vol}(y_i)), \sin(\texttt{Vol}(y_i))]\right\| + r_{\text{cur}},$ $\{y_i\}_{i=1}^{N} \sim \pi_\theta(\cdot\|x), \quad G = \|\{i : \text{ans}(y_i) = \text{ans}(y)\}\|$

Table 2 | Overview of ensemble-based rewards, estimators and their formulas. All variants operationalize the assumption that consistency across independent samples correlates with correctness.

External Reward Methods

- **External rewards:** generate verifiable rewards through external mechanisms
  - **Leveraging Unlabeled Data for Reward Generation**: Large-scale unlabeled corpora provide natural verification signals by converting language modeling into reward-based tasks.
  - **Exploiting Generation-Verification Asymmetries**: Rather than relying on human labels or on the model’ s own internal confidence, the verification procedure itself acts as an external, objective and infinitely scalable reward.

Outline

- Background
- Taxonomy of Unsupervised RLVR
- **The Sharpening Mechanism**
- Experiments

The Sharpening Mechanism

- Dynamics of one-step update: rich getting richer

$$\max_{\pi_\theta} \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r(x,y)\right] - \beta D_{\mathrm{KL}}\left[\pi_\theta(\cdot|x)\|\pi_{\mathrm{ref}}(\cdot|x)\right],$$

$$\pi_\theta^*(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right),$$

$$\pi_\theta^{*,(k+1)}(y|x) = \begin{cases} \frac{\pi_\theta^{(k)}(y|x)\cdot e^{1/\beta}}{Z_k(x)}, & \text{if } \mathrm{ans}(y) = \mathrm{maj}_k(Y_k), \\ \frac{\pi_\theta^{(k)}(y|x)}{Z_k(x)}, & \text{otherwise,} \end{cases}$$

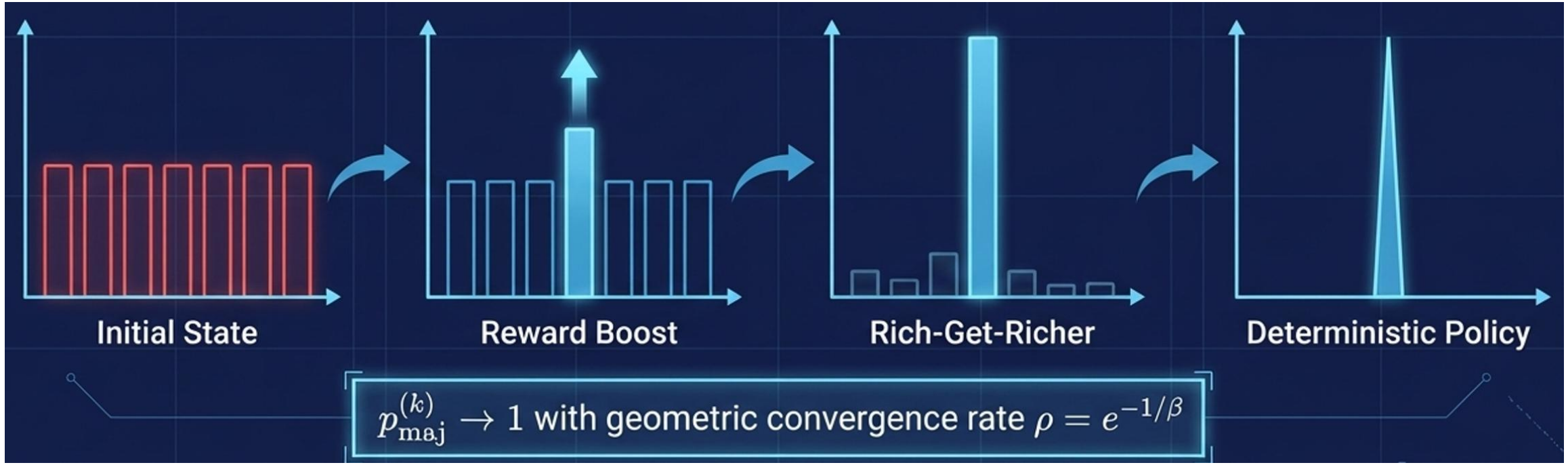
$$p_{\mathrm{maj}}^{(k)} = \sum_{y:\mathrm{ans}(y)=\mathrm{maj}_k(Y_k)} \pi_\theta^{(k)}(y|x)$$

$$p_{\mathrm{maj}}^{*,(k+1)} = \frac{p_{\mathrm{maj}}^{(k)}\cdot e^{1/\beta}}{p_{\mathrm{maj}}^{(k)}\cdot e^{1/\beta} + (1-p_{\mathrm{maj}}^{(k)})}.$$

$$p_{\mathrm{maj}}^{*,(k+1)} \geq p_{\mathrm{maj}}^{(k+1)} \geq p_{\mathrm{maj}}^{(k)}.$$

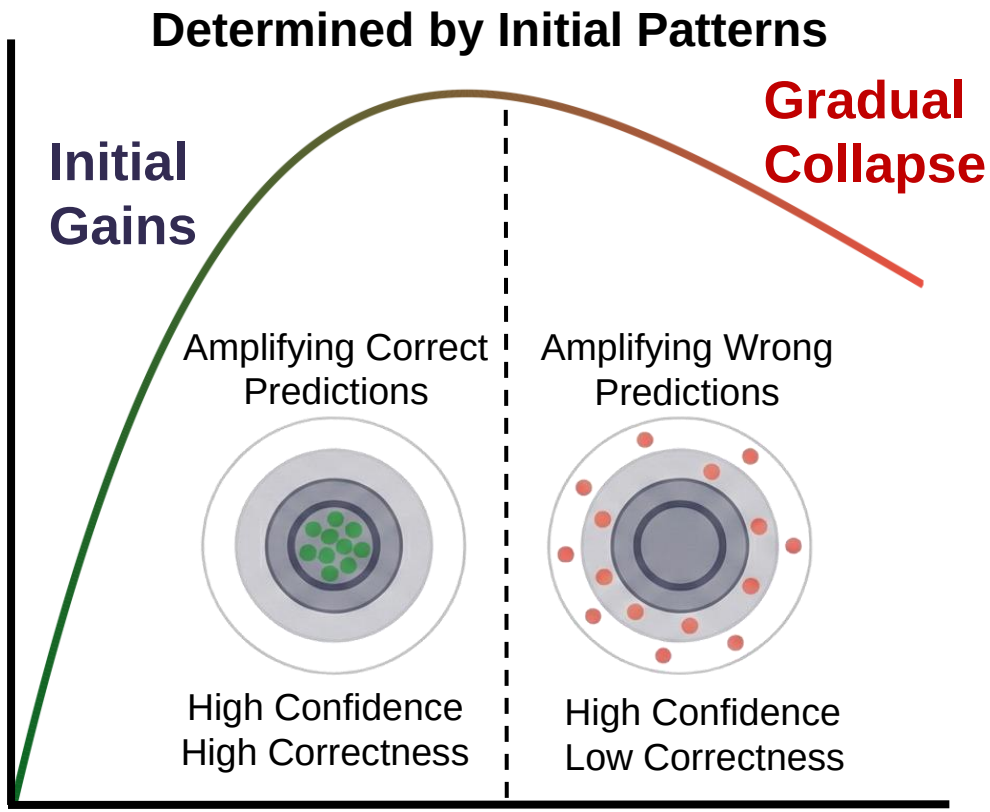
The Sharpening Mechanism

- The model optimized through intrinsic rewards converges towards sharpening its initial distribution



The Sharpening Mechanism

- Implication: depending on the model prior
  - If confidence aligns with correctness, convergence reinforces good solutions
  - If confidence is poorly aligned, the same mechanism amplifies errors



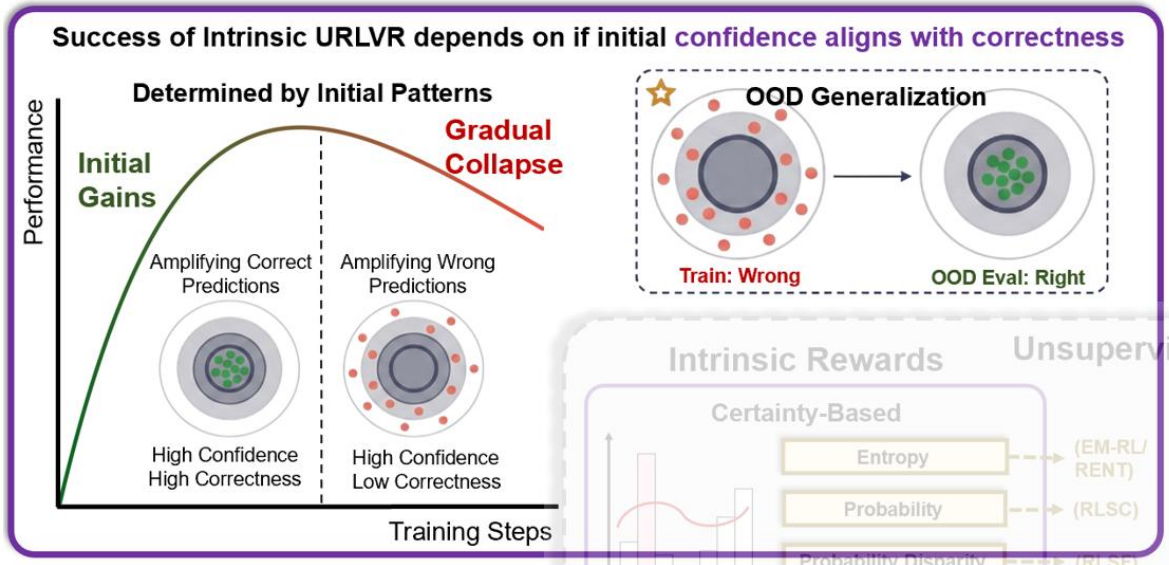
Outline

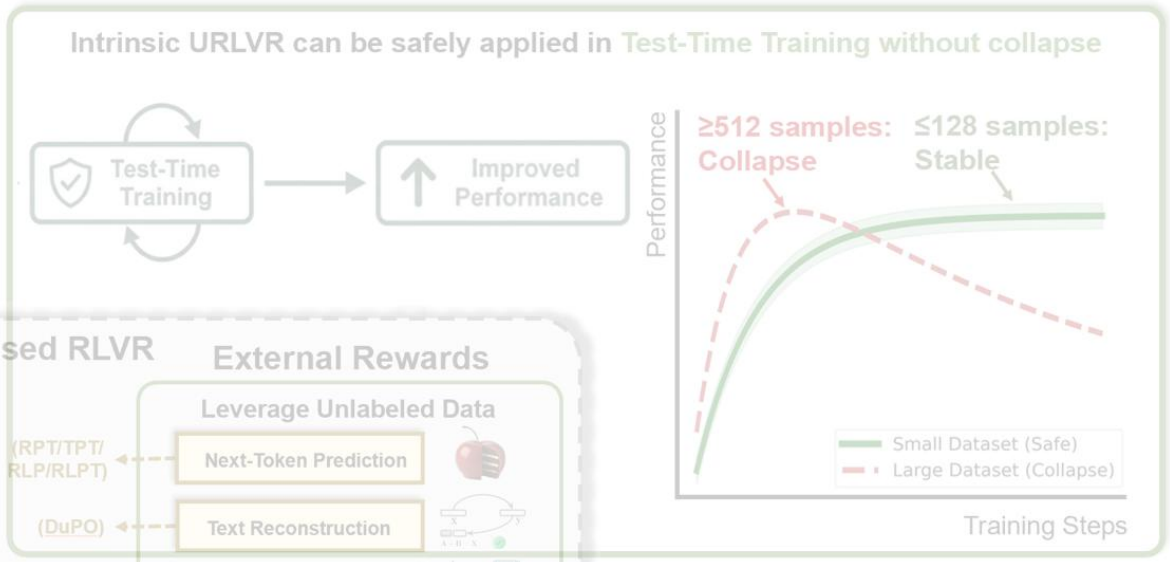
- Background
- Taxonomy of Unsupervised RLVR
- The Sharpening Mechanism
- **Experiments**

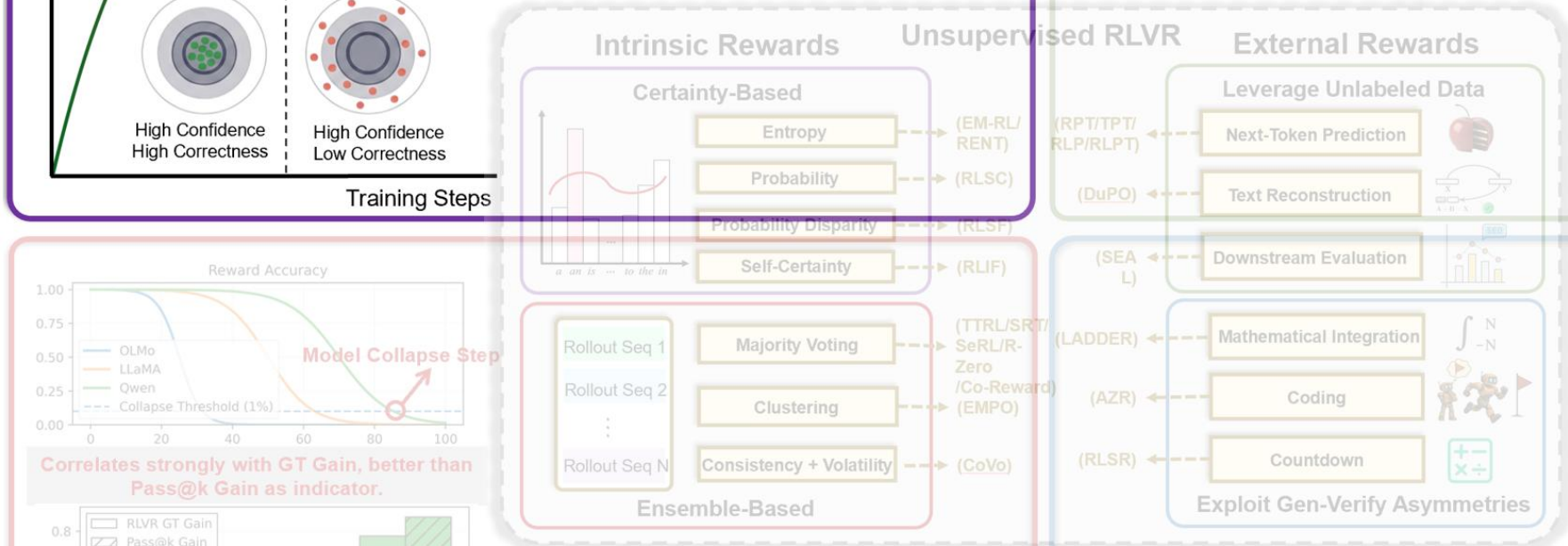
Experiments

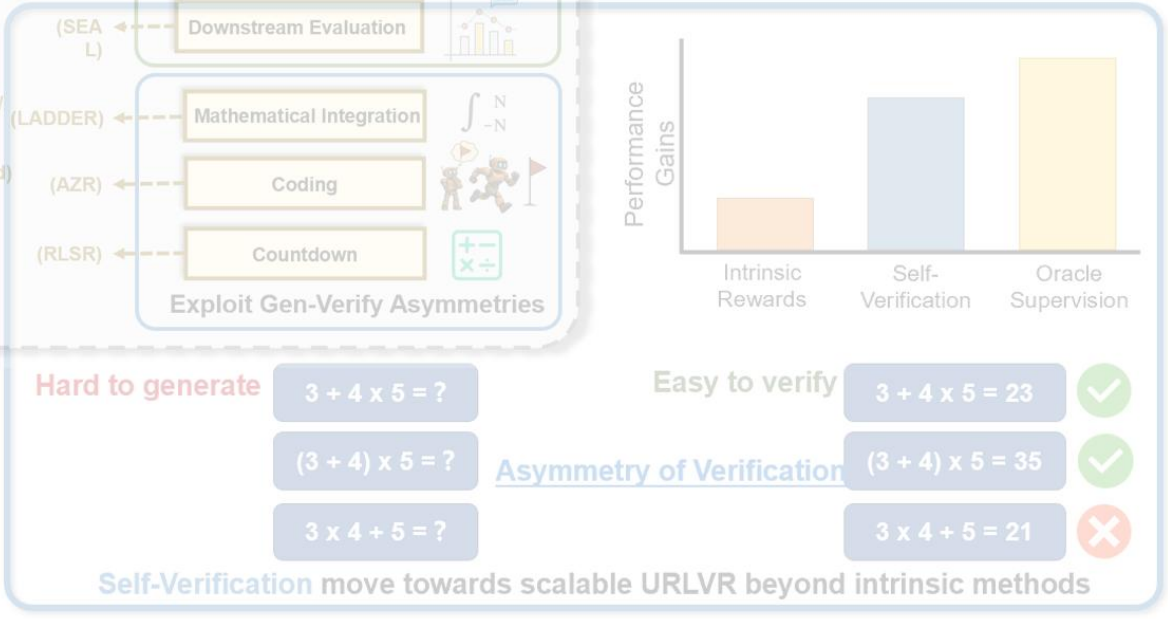
When Does Intrinsic URLVR Work?

When Does Intrinsic URLVR Work?



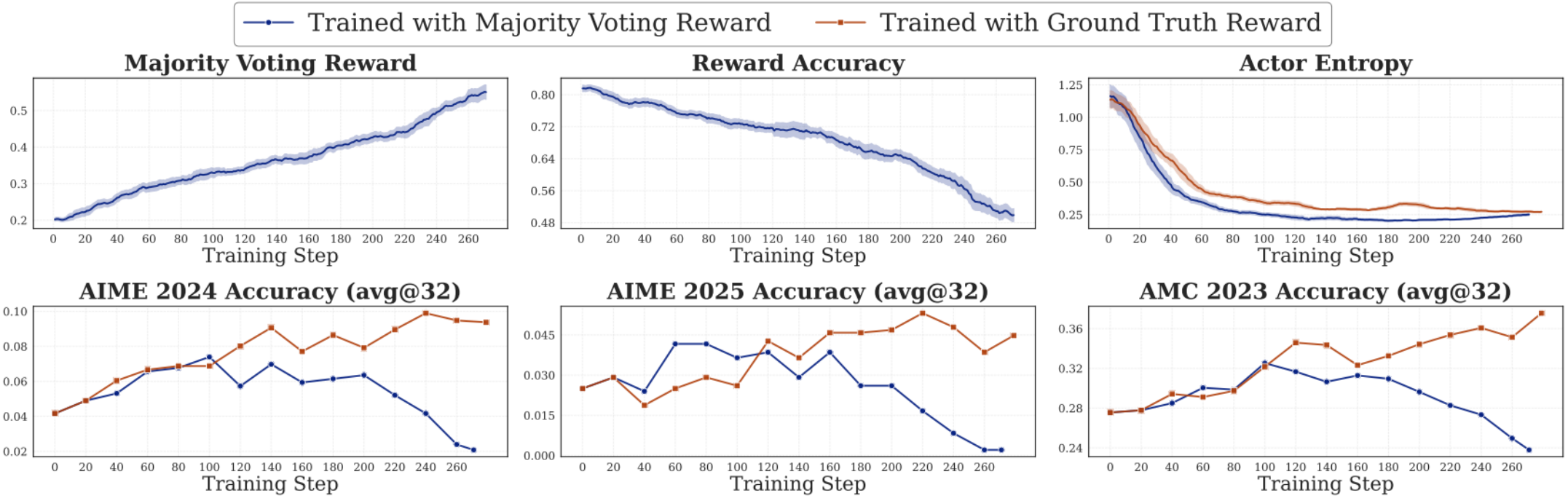






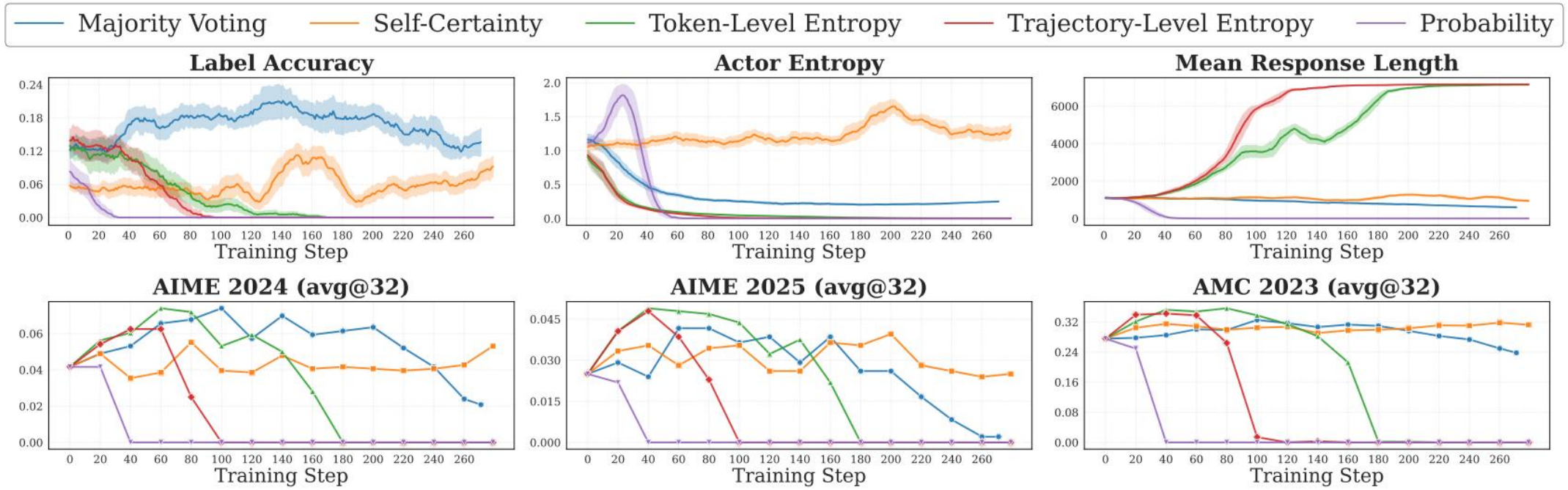
When Does Intrinsic URLVR Work?

- The rise and fall of intrinsic URLVR: Early Success, Later Collapse
  - Qwen3-1.7B-Base on DAPO-math-17k
  - Unsupervised Reward: majority voting reward



When Does Intrinsic URLVR Work?

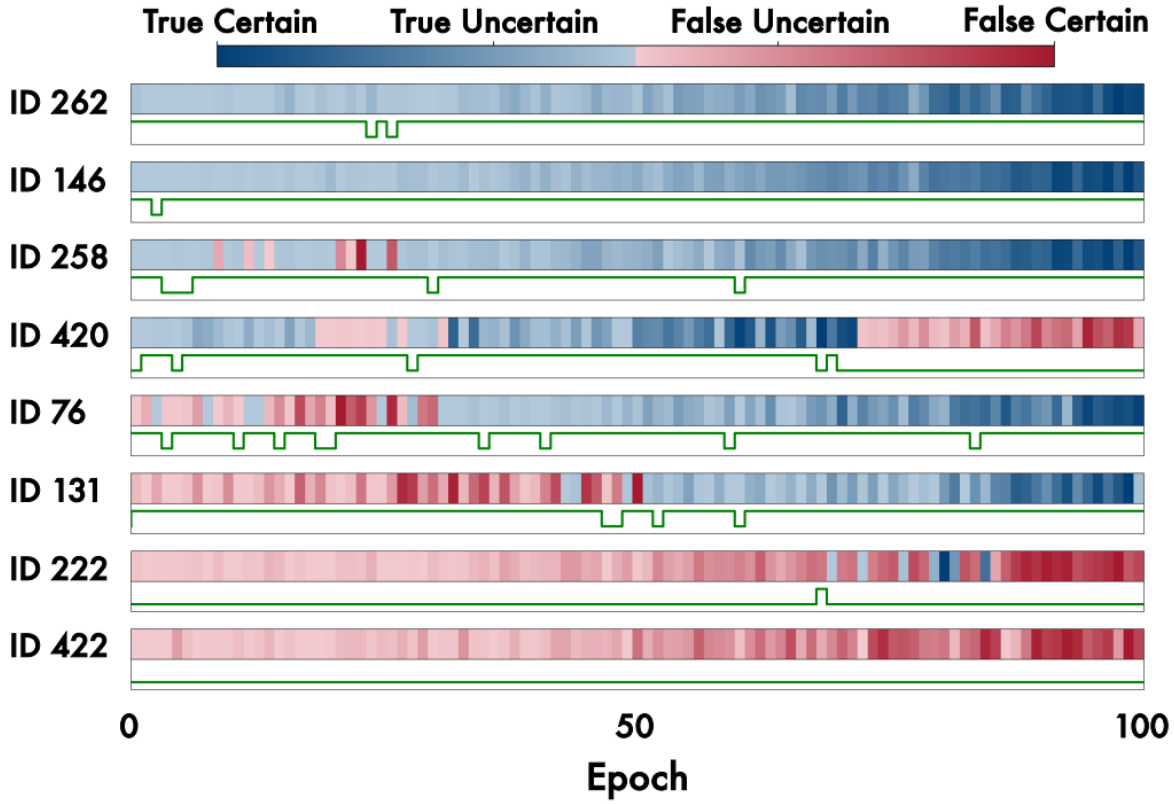
- The rise and fall of intrinsic URLVR: Different Methods, Different Failures
  - Qwen3-1.7B-Base on DAPO-math-17k
  - Majority Voting & Self-Certainty performs the best



When Does Intrinsic URLVR Work?

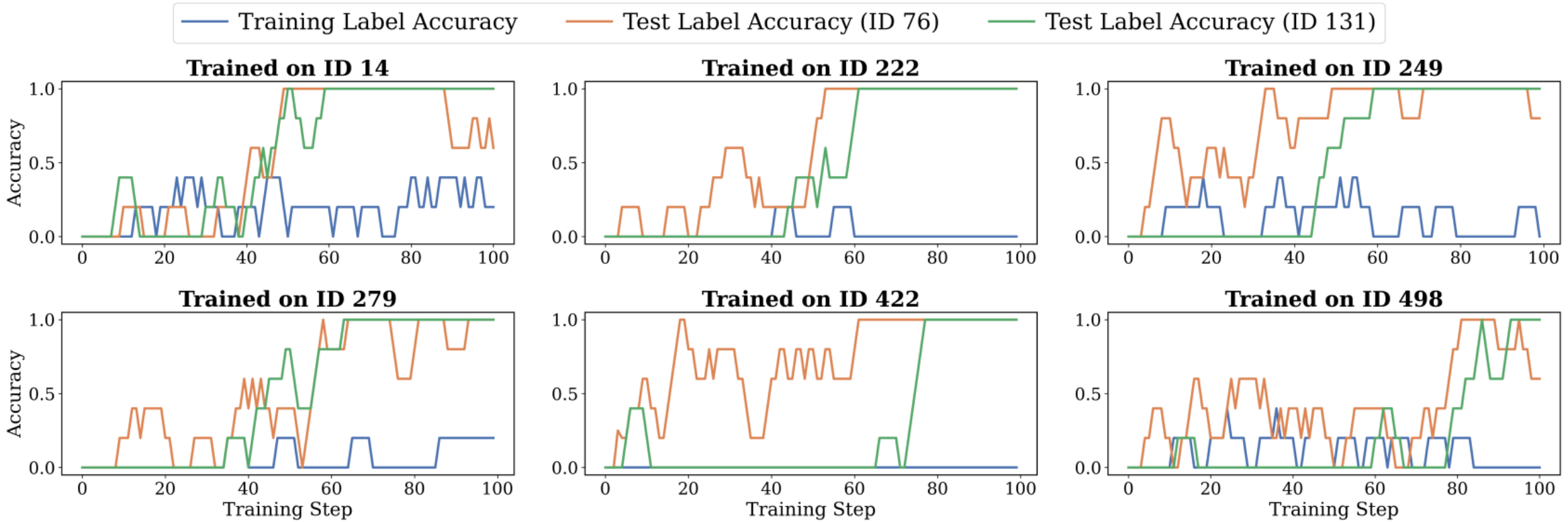
- Fine-grained per-problem analysis: In-Distribution Sharpening
  - Randomly sample problems from MATH-500, **trained separately**
  - Unsupervised Reward: trajectory-level entropy

For each problem, the heatmap shows greedy decoding correctness across epochs (**blue = correct**, **red = wrong**; **darker = higher confidence**), and the **green wave** indicates whether the highest-reward rollout is correct.



When Does Intrinsic URLVR Work?

- Fine-grained per-problem analysis: OOD Generalization
  - Randomly sample problems from MATH-500, **trained separately**
  - Unsupervised Reward: trajectory-level entropy



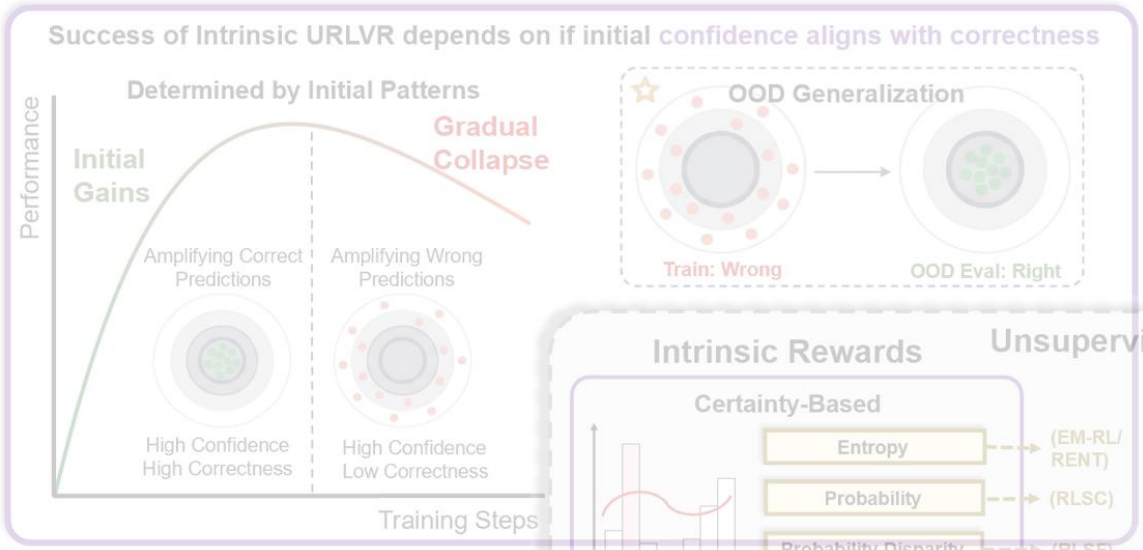
When Does Intrinsic URLVR Work?

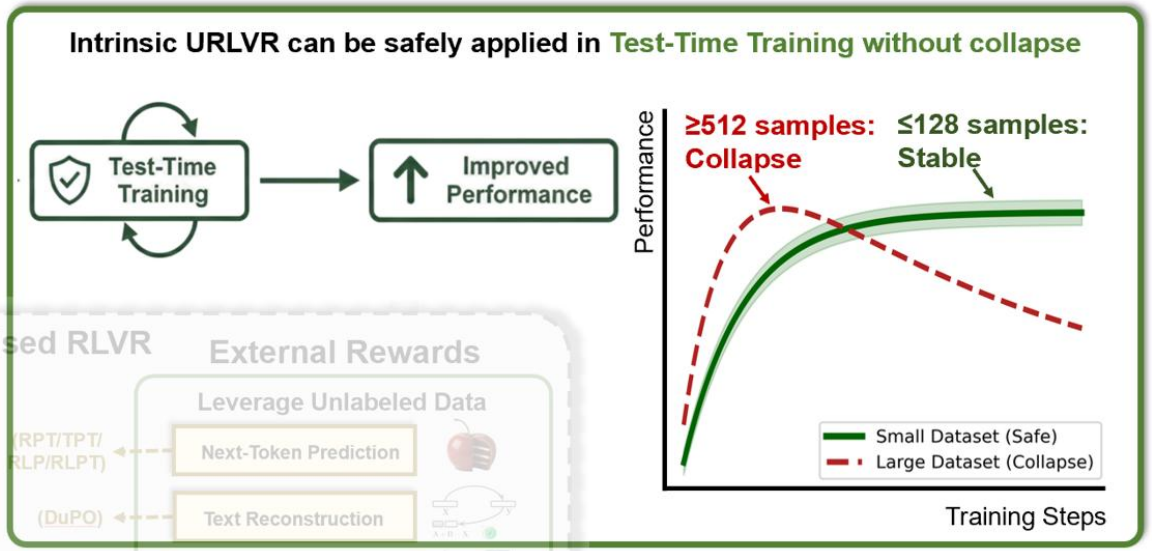
Intrinsic URLVR universally follows a rise-then-fall pattern across all methods. Early gains reflect confidence-correctness alignment in the model's prior, while eventual collapse is inevitable when this alignment breaks down.

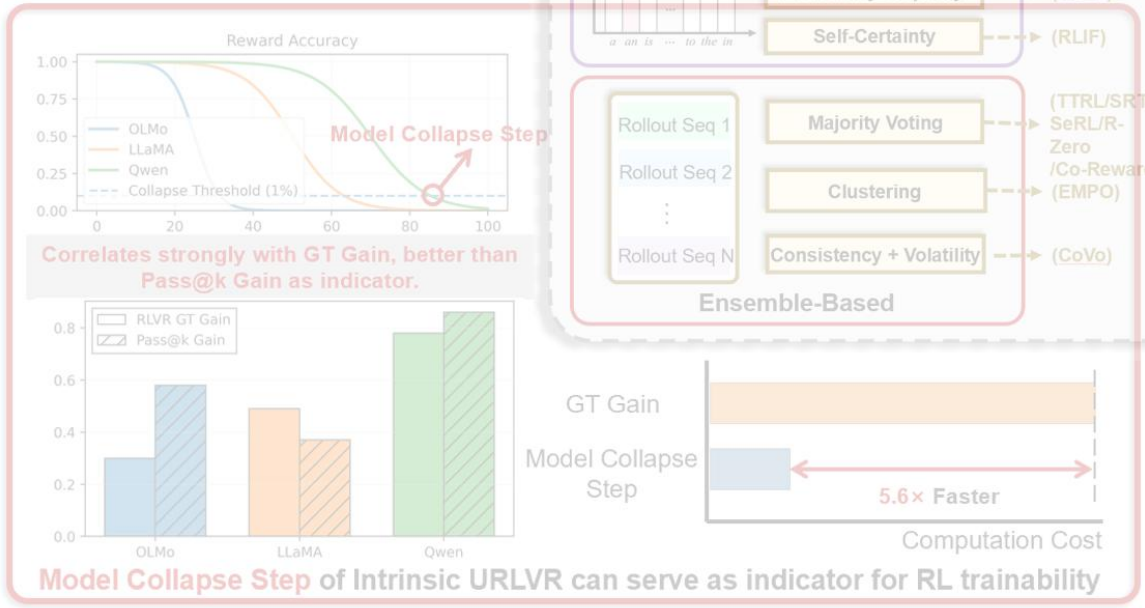
Experiments

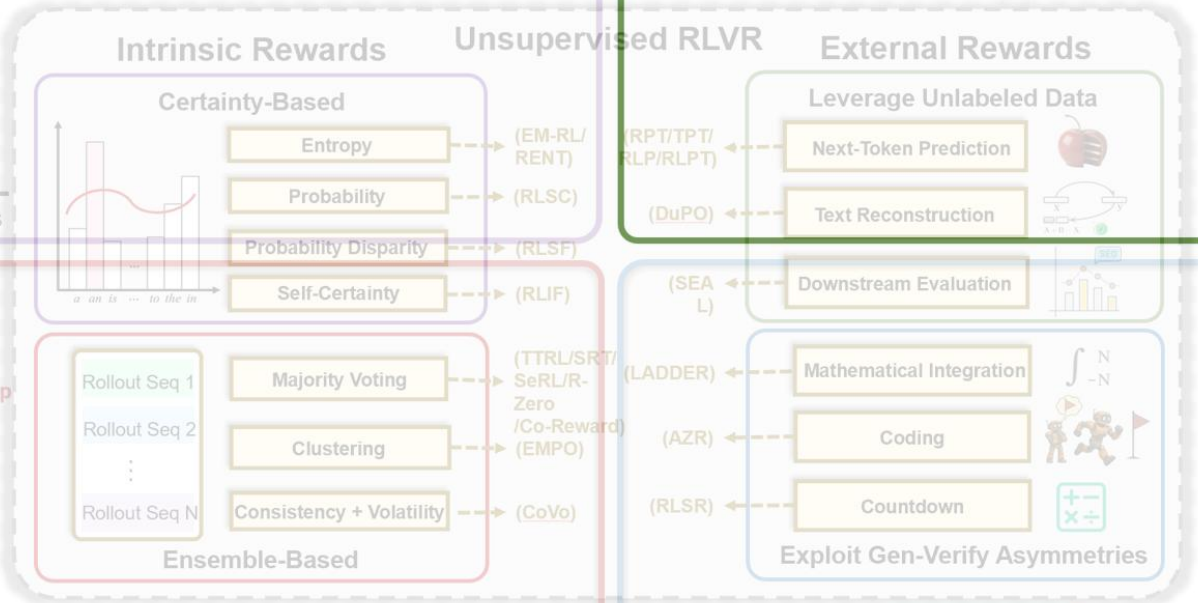
How Can Sharpening from Intrinsic URLVR Be Applied Safely?

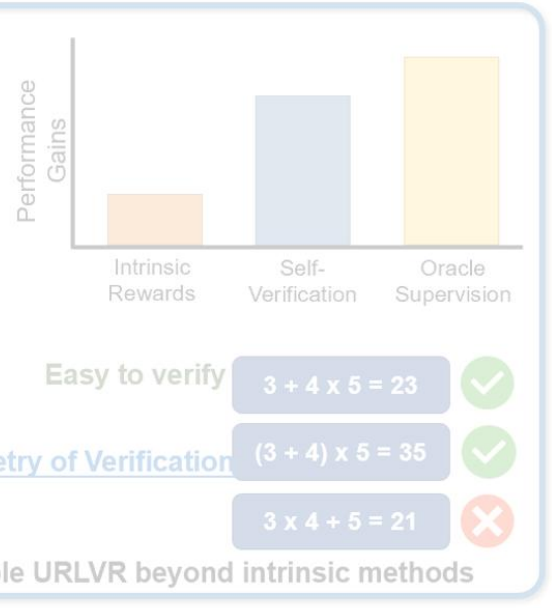
How Can Sharpening from Intrinsic URLVR Be Applied Safely?





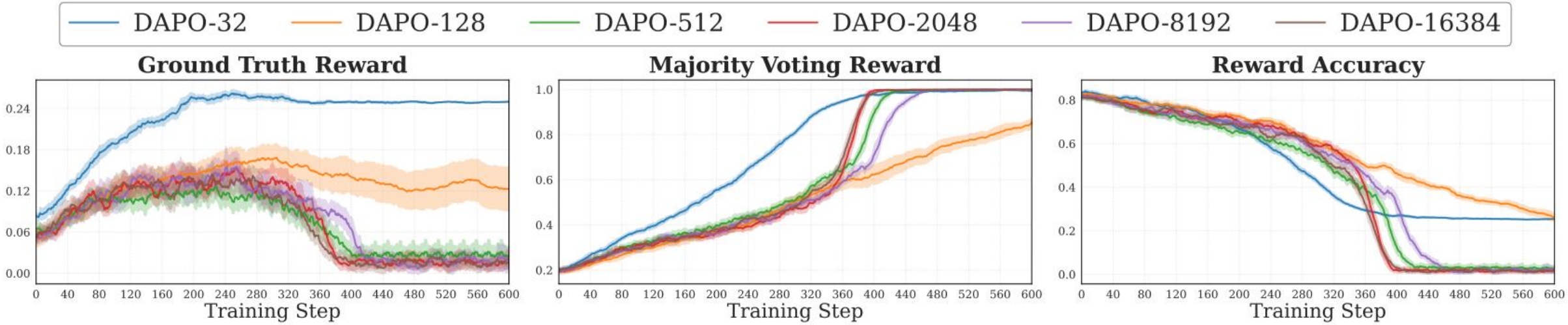






How Can Sharpening from Intrinsic URLVR Be Applied Safely?

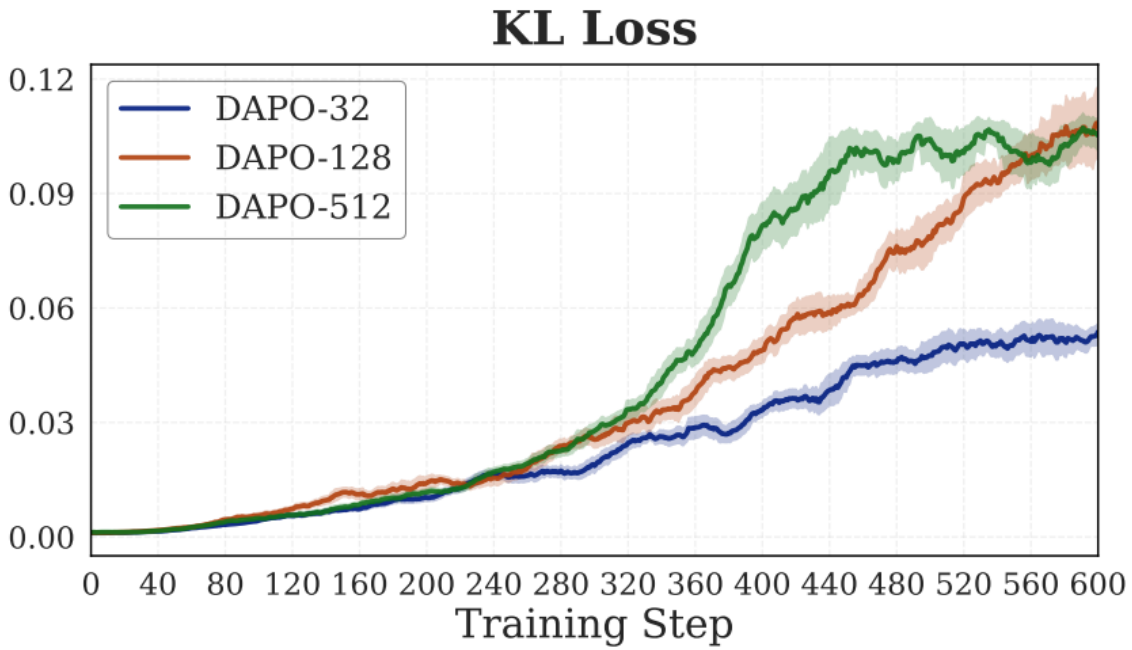
- Small Datasets Prevent Model Collapse
  - Qwen3-1.7B-Base on DAPO-math-17k (32, 128, 512, 2k, 8k, 16k samples)
  - Training with ≤ 128 samples maintains stable performance without collapse



How Can Sharpening from Intrinsic URLVR Be Applied Safely?

- Small Datasets Prevent Model Collapse
  - Qwen3-1.7B-Base on DAPO-math-17k (32, 128, 512, 2k, 8k, 16k samples)
  - Small datasets induce localized overfitting rather than systematic policy shift

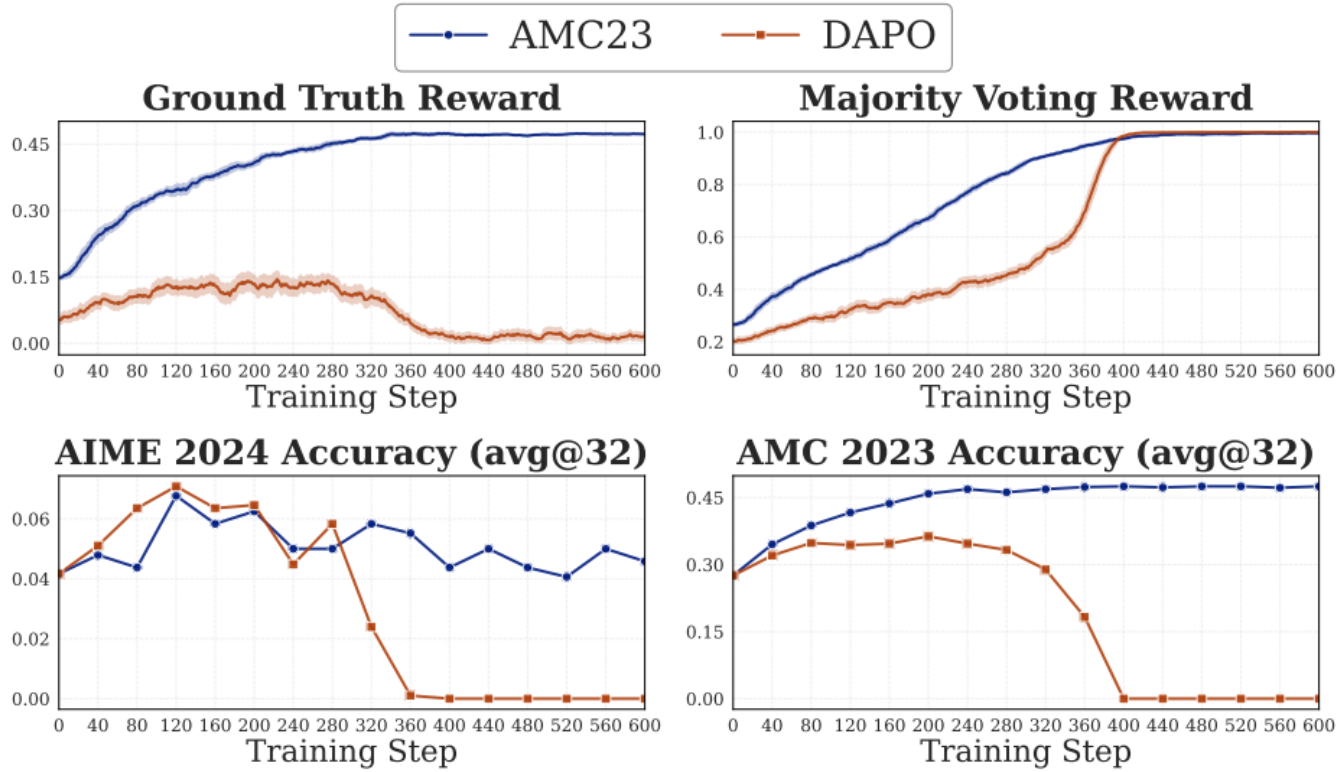
$$D_{\mathrm{KL}}^{(t)}(\pi_\theta^{(t)} \| \pi_{\mathrm{ref}}) = \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{train}}}\left[\mathbb{E}_{y\sim\pi_\theta^{(t)}(\cdot|x)}\left[\log \frac{\pi_\theta^{(t)}(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right]\right]$$



How Can Sharpening from Intrinsic URLVR Be Applied Safely?

- Test-Time Training as a Safe Application
  - Qwen3-1.7B-Base on AMC23/DAPO-17k
  - Test-time training on AMC23 avoids collapse

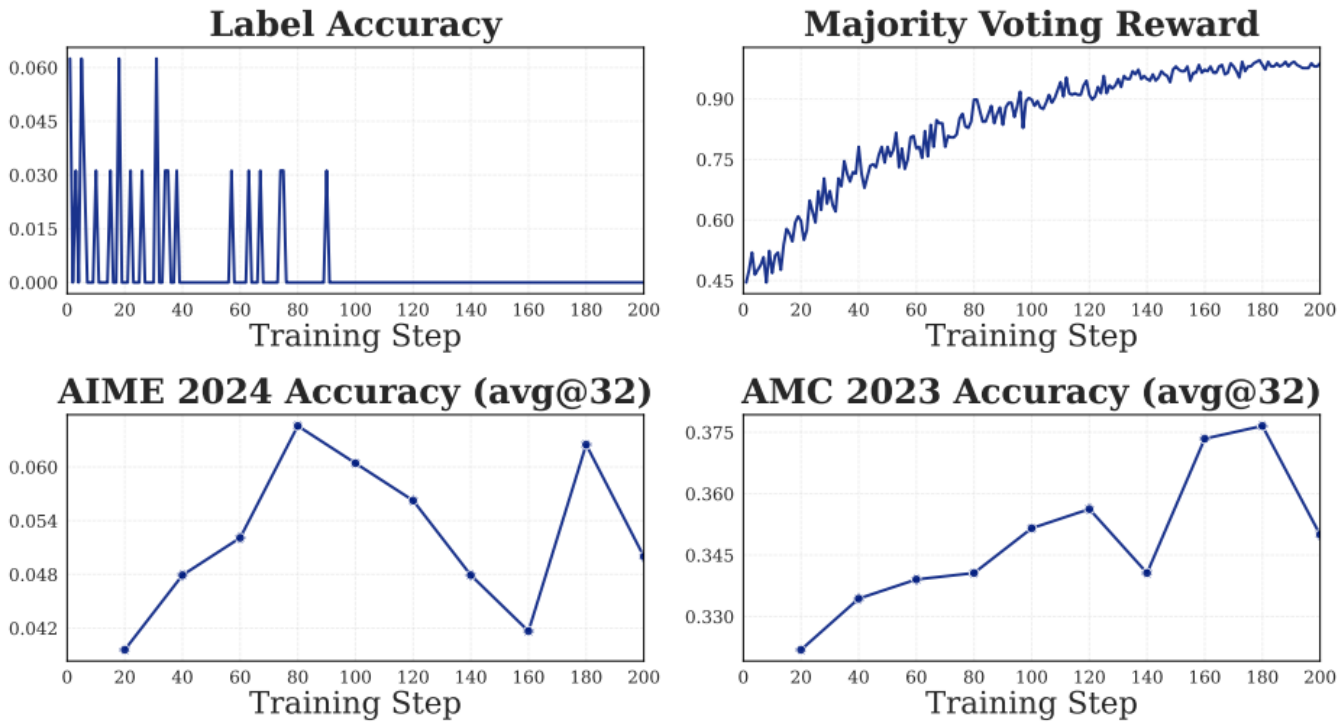
This indicates that intrinsic URLVR may be safely applied in test-time training.



How Can Sharpening from Intrinsic URLVR Be Applied Safely?

- Incorrect Majority Votes Still Improve Reasoning
  - Offline-filtering examples where the initial majority votes are incorrect
  - Train on 32 filtered samples using the same setting as DAPO-32

Even when almost all 32 samples have incorrect initial majority votes, training still produces effective learning without catastrophic collapse.



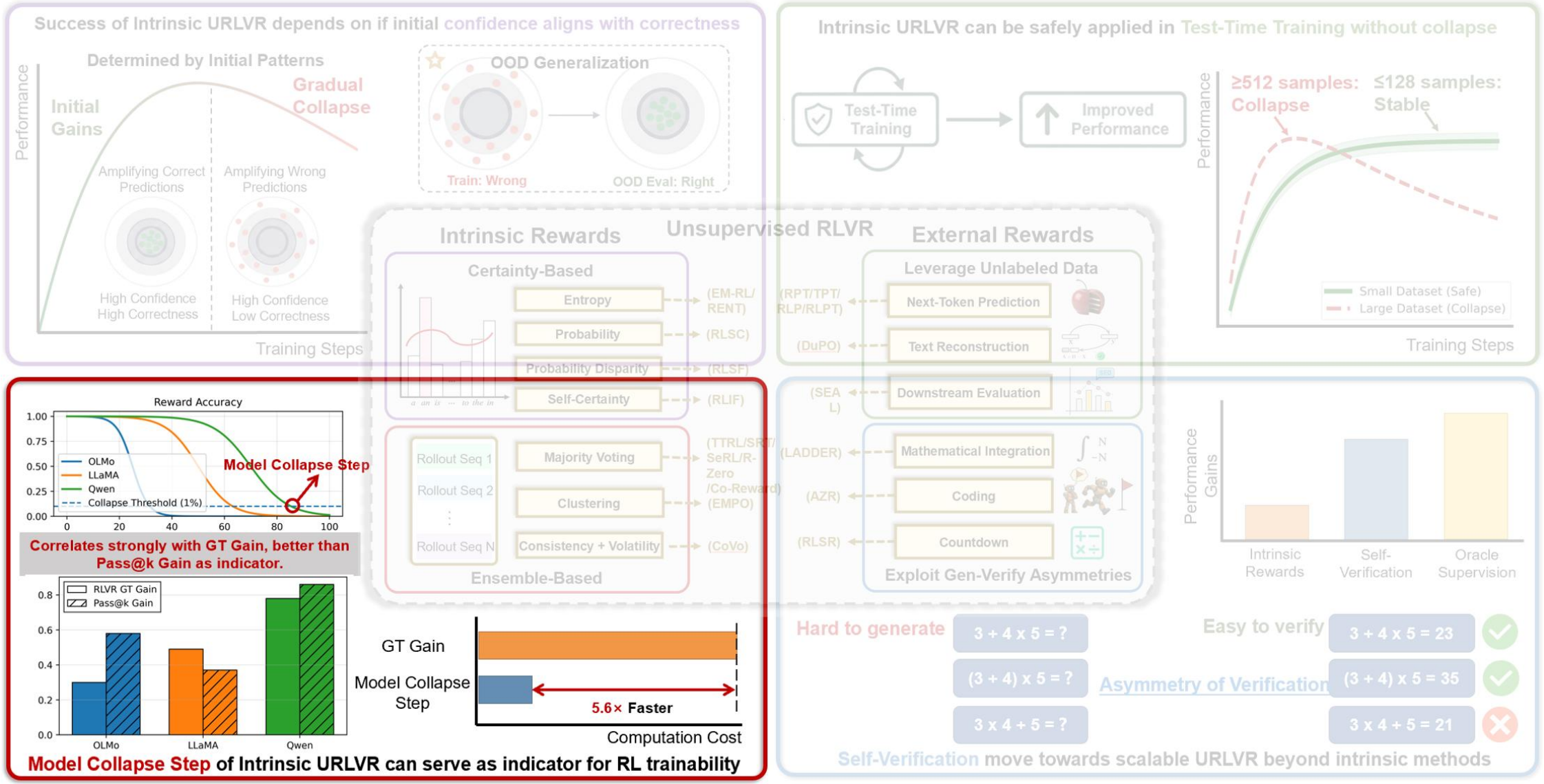
How Can Sharpening from Intrinsic URLVR Be Applied Safely?

Small datasets induce localized rather than systematic policy shift, even training on wrong problems can yield gains, making test-time training a safe and practical application.

Experiments

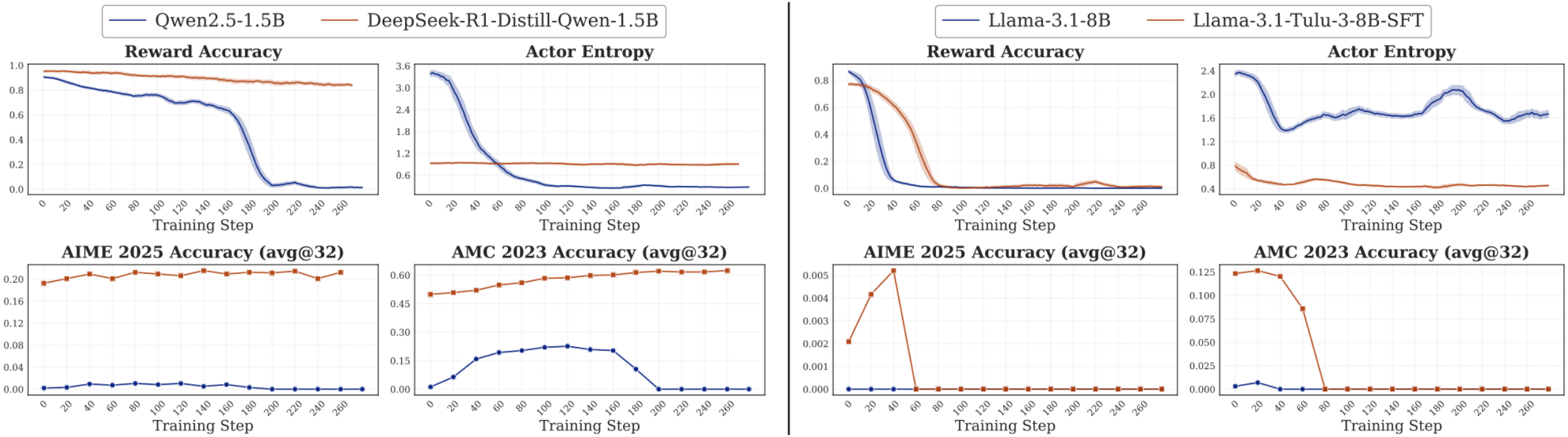
How Can We Measure Model Prior?

How Can We Measure Model Prior?



How Can We Measure Model Prior?

- Pilot Study: Different Models, Different Outcomes
  - Compare two pairs from Qwen25 / Llama31 family (base vs. SFT)
  - SFT variant in Qwen shows steady improvement, while Llama both fail



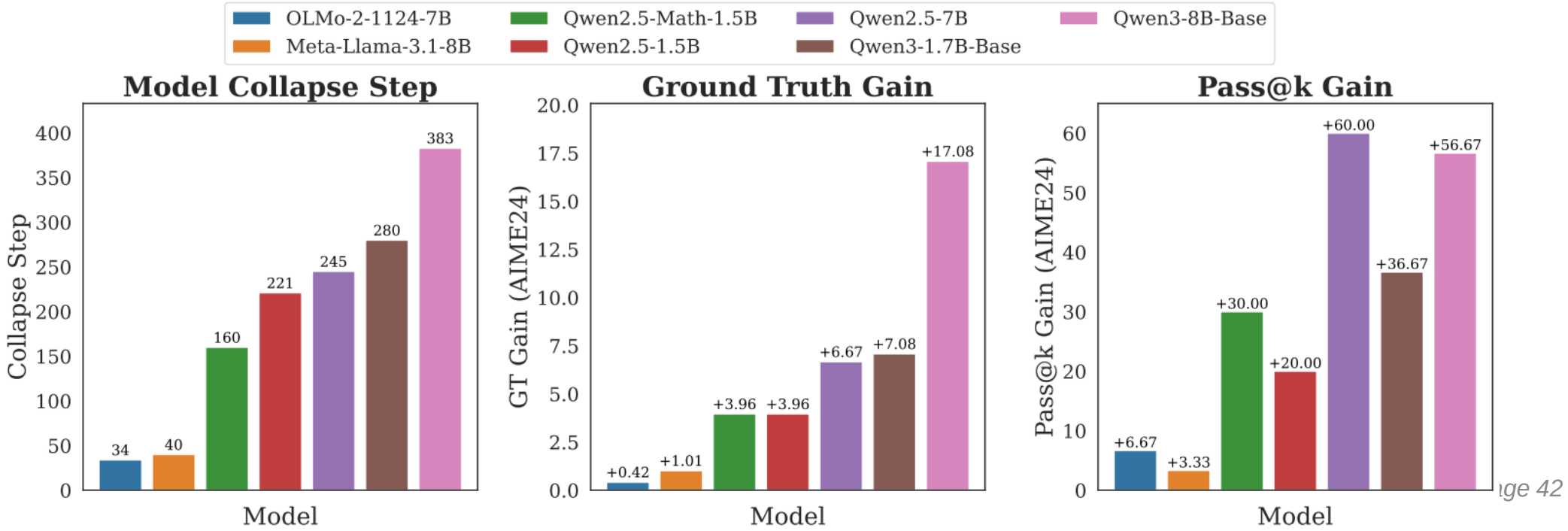
How Can We Measure Model Prior?

- What is model prior?
  - Intrinsic URLVR is effective only when the model's initial confidence is aligned with correctness
  - Can the strength of this alignment be measured?
- Why should we measure it?
  - Predict RL trainability without running full RL training to select the base model
  - Pass@k has limitations

Model Collapse Step: The training step where Reward Accuracy drops below 1%

How Can We Measure Model Prior?

- Model Collapse Step Accurately Predicts RL Gains
  - 7 models from 3 families (OLMo, LLaMA, Qwen)
  - Models that survive longer with a larger Model Collapse Step consistently yield better results in standard supervised RL training



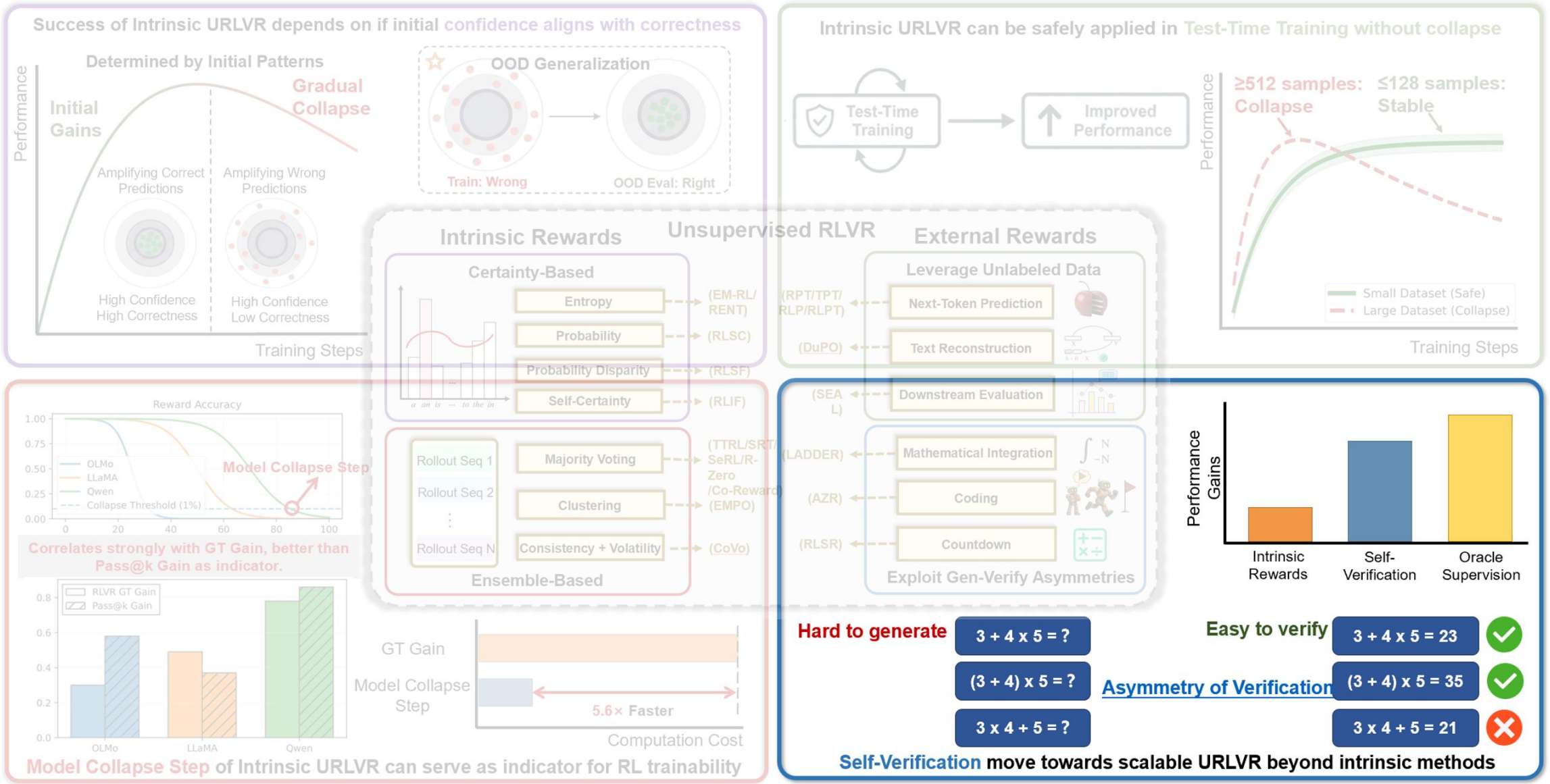
How Can We Measure Model Prior?

We propose the Model Collapse Step as a novel indicator of model priors, which measures standard RL trainability by tracking reward accuracy collapses during intrinsic URLVR. This indicator achieves accuracy in assessing trainability on par with running standard RL itself, but with higher efficiency and outperforming pass@k.

Experiments

Can We Move Towards Scalable URLVR?

Can We Move Towards Scalable URLVR?



Can We Move Towards Scalable URLVR?

- Intrinsic URLVR have the ceiling
  - Faces scalability limits rooted in confidence-correctness alignment
  - Cannot consistently push the model beyond what it already knows

- External URLVR: generate verifiable through external mechanisms
  - Leverage unlabeled data structures to derive rewards from the corpus
  - Exploit generation-verification asymmetries

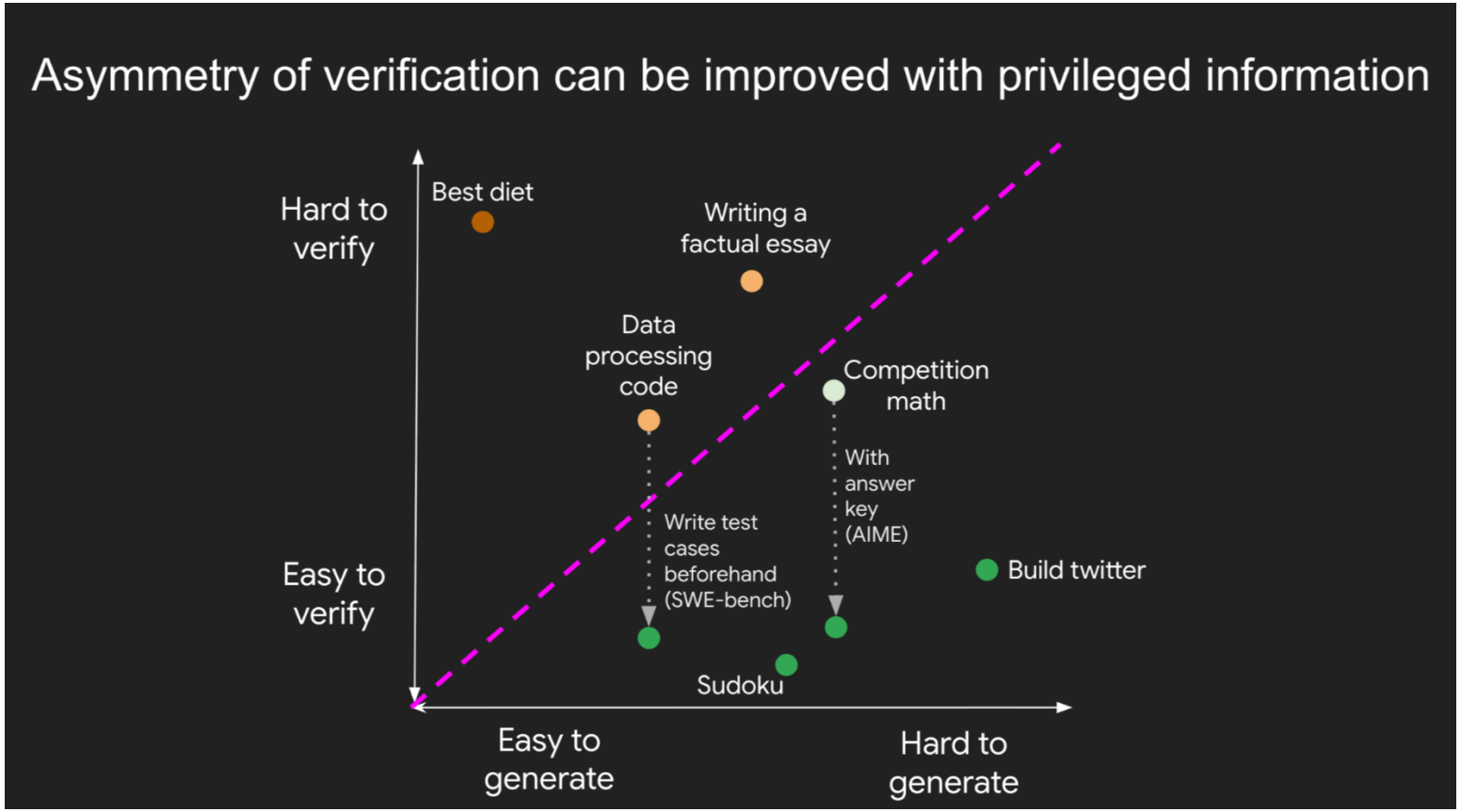
Can We Move Towards Scalable URLVR?

Hard to generate

- 3 + 4 x 5 = ?
- (3 + 4) x 5 = ?
- 3 x 4 + 5 = ?

Easy to verify

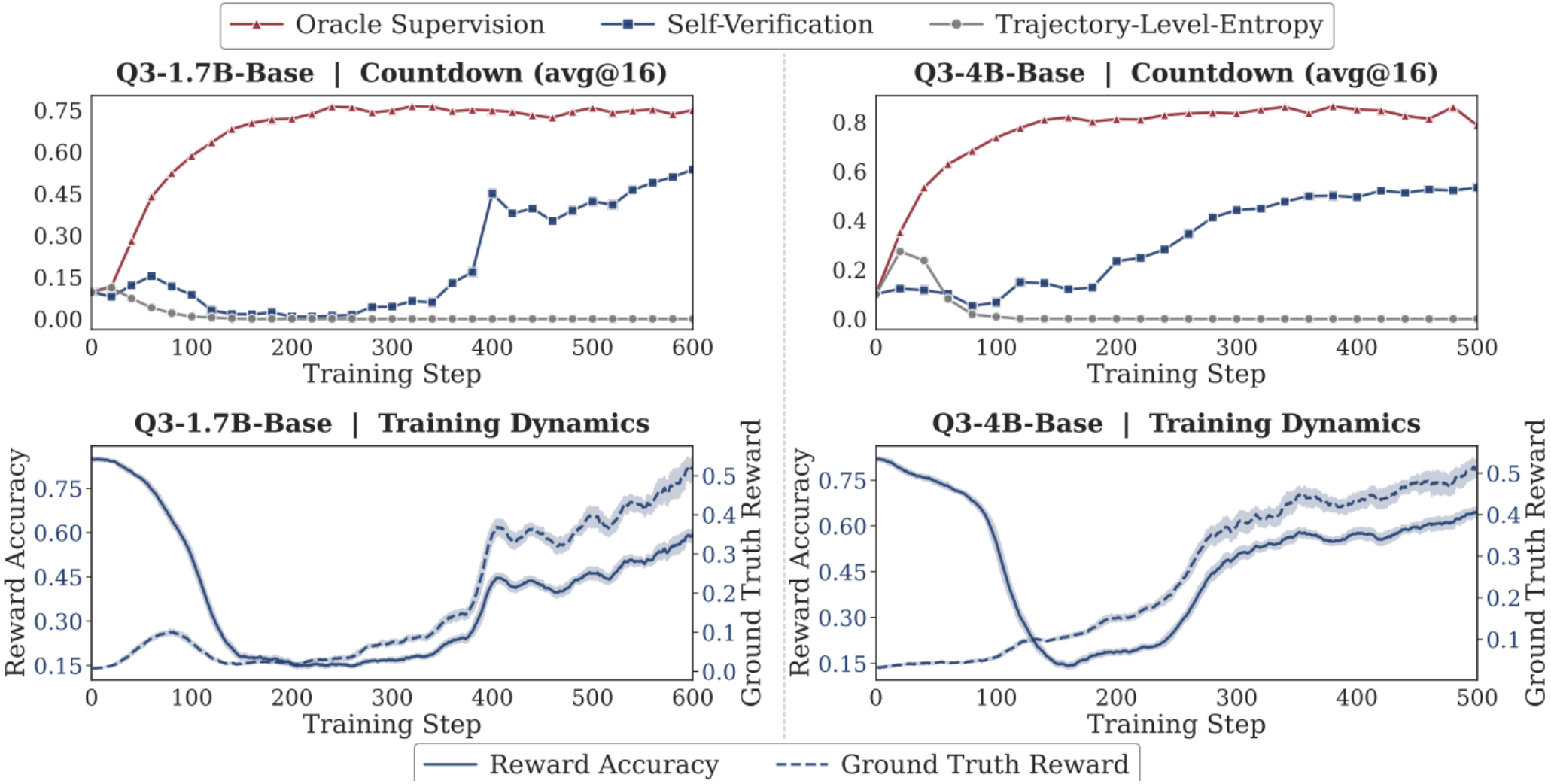
- 3 + 4 x 5 = 23 ✓
- (3 + 4) x 5 = 35 ✓
- 3 x 4 + 5 = 21 ✗



https://www.jasonwei.net/blog/asymmetry-of-verification-and-verifiers-law

Can We Move Towards Scalable URLVR?

- Self-Verification works much better than Trajectory-Level Entropy



Can We Move Towards Scalable URLVR?

Intrinsic rewards are fundamentally bounded by what the model already knows. External rewards grounded in unlabeled data or generation-verification asymmetry provide signals that scale with data and computation rather than saturating with model capacity, offering a more promising path towards scalable URLVR.

How Far Can Unsupervised RLVR Scale LLM Training?

Bingxiang He[*1], **Yuxin Zuo**[*†1,2], **Zeyuan Liu**[*1], **Shangziqi Zhao**[*3], **Zixuan Fu**[1], **Junlin Yang**[1], **Cheng Qian**[4], **Kaiyan Zhang**[1,5], **Yuchen Fan**[6], **Ganqu Cui**[2], **Xiusi Chen**[4], **Youbang Sun**[1], **Xingtai Lv**[1], **Xuekai Zhu**[6], **Li Sheng**[1], **Ran Li**[1], **Huan-ang Gao**[1], **Yuchen Zhang**[7], **Bowen Zhou**[‡1,2], **Zhiyuan Liu**[‡1], **Ning Ding**[‡1,2]

[1]Tsinghua University [2]Shanghai AI Lab [3]Xi'an Jiaotong University [4]University of Illinois Urbana-Champaign
[5]Frontis.AI [6]Shanghai Jiao Tong University [7]Peking University

[*]**Equal Contribution. Orders are determined randomly.** [†]**Project Lead.** [‡]**Corresponding Authors.**

https://github.com/PRIME-RL/TTRL.

hebx24@mails.tsinghua.edu.cn, dingning@mail.tsinghua.edu.cn

Full Paper **Code** **Homepage**

THANKS

Q & A

Bingxiang He | THUNLP | Advisor: Prof. Zhiyuan Liu

Homepage: https://hbx-hbx.github.io/

2026.04.18

We are actively working on scalable RL!