香港中文大學
The Chinese University of Hong Kong

ViSurf: Visual Supervised and Reinforcement Fine-Tuning

Yuqi Liu, Liangyu Chen, Jiazhen Liu, Mingkang Zhu,
Zhisheng Zhong, Bei Yu, Jiaya Jia
CUHK, RUC, HKUST

May 8, 2026

Outline

Introduction

Post-training Large Vision-and-Language Models (LVLMs) typically involves Supervised Fine-Tuning (SFT) for knowledge injection or Reinforcement Learning with Verifiable Rewards (RLVR) for performance enhancement.

- SFT often leads to sub-optimal performance.
- RLVR remains constrained by the model's internal knowledge base.
- Sequential SFT $\rightarrow$ RLVR pipeline introduces significant computational overhead and suffers from catastrophic forgetting.

Our evaluation across diverse vision-language benchmarks, summarized in Figure 1, confirms this phenomenon: SFT excels in out-of-distribution knowledge acquisition, whereas RLVR thrives on tasks aligning with pre-existing capabilities.

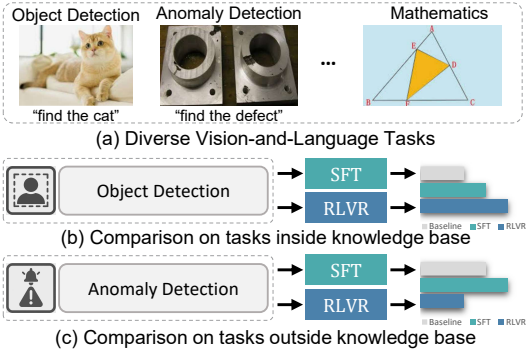


Figure: (a) Examples of vision-ang-language tasks. (b) For tasks within LVLMs' knowledge base, RLVR performs better than SFT. (c) For tasks that exceed LVLMs' knowledge, SFT performs better, whereas RLVR performs worse than baseline.

To overcome these challenges, we propose ViSurf (**Vi**sual **Su**pervised and **R**einforcement **F**ine-Tuning), a unified, single-stage paradigm that integrates the complementary advantages of SFT and RLVR.

- We provide a rigorous analysis of the underlying objectives and gradients for both methods, theoretically demonstrating that their shared gradient patterns allow for integration into a singular ViSurf objective.
- ViSurf offers a theoretically grounded, unified perspective. The gradient of ViSurf objective can be interpreted as a composite of the gradients from both SFT and RLVR.
- We introduce three novel reward control strategies for ground-truth labels: (i) preference alignment with policy rollouts, (ii) exclusion of "thinking" rewards for static labels, and (iii) reward smoothing to prevent optimization spikes.

Method

Let π_θ denote a large vision-and-language model (LVLM), parameterized by θ. Common post-training paradigms for optimizing π_θ include Supervised Fine-Tuning (SFT) and Reinforcement Learning with Verifiable Rewards (RLVR). Both SFT and RLVR utilize the same input dataset, $\mathcal{D}_{\text{input}} = \{(v_i, t_i)\}_{i=1}^{N}$, where v_i is a visual input, t_i is a textual input, and N is the dataset size.

Supervised Fine-Tuning (SFT) optimizes π_θ against a set of ground-truth labels, $\mathcal{D}_{\text{label}} = \{y_i\}_{i=1}^{N}$. The objective is to minimize the negative log-likelihood of the labels:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\text{input}} \\ y\sim\mathcal{D}_{\text{label}}}} \left[\log \pi_\theta(y \mid v, t)\right], \tag{1}$$

where y corresponds to (v, t). A more precise notation would be $(v, t, y) \sim \text{zip}(\mathcal{D}_{\text{input}}, \mathcal{D}_{\text{label}})$. Nevertheless, we retain the current notation, $(v, t) \sim \mathcal{D}_{\text{input}}, y \sim \mathcal{D}_{\text{label}}$, for clarity and ease of comparison in the subsequent discussion.

Method

We illustrate RLVR using the on-policy Group Relative Policy Optimization (GRPO) algorithm. GRPO optimizes the policy π_θ using a verifiable reward function. For a given input $(v_i, t_i) \in \mathcal{D}_{\text{input}}$, the old policy $\pi_{\theta_{old}}$ (from a previous optimization step) generates a group of G rollouts $\{o_j\}_{j=1}^{G}$ by sampling with different random seeds. Each rollout o_j is then evaluated by a reward function $r(\cdot)$, resulting in a set of rewards $\{r(o_j)\}_{j=1}^{G}$. The objective of RLVR is to minimize the equation:

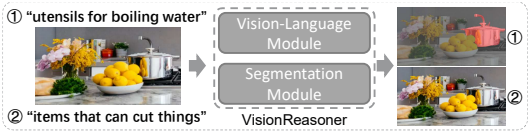
$$\hat{A}_j = \frac{\mathrm{r}(o_j) - \mathrm{mean}\left(\{\mathrm{r}(o_j)\}_{j=1}^{G}\right)}{\mathrm{std}\left(\{\mathrm{r}(o_j)\}_{j=1}^{G}\right)}, \tag{2}$$

$$\mathcal{L}_{\mathrm{RLVR}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\text{input}} \\ \{o_j\}_{j=1}^{G}\sim\pi_{\theta_{\text{old}}}}} \left[\frac{1}{G}\sum_{j=1}^{G} \min\left\{ \frac{\pi_\theta(o_j \mid v, t)}{\pi_{\theta_{\text{old}}}(o_j \mid v, t)}\,\hat{A}_j, \right.\right. \\ \left.\left. \mathrm{clip}\left(\frac{\pi_\theta(o_j \mid v, t)}{\pi_{\theta_{\text{old}}}(o_j \mid v, t)},\ 1-\epsilon,\ 1+\epsilon \right)\hat{A}_j \right\} \right], \tag{3}$$

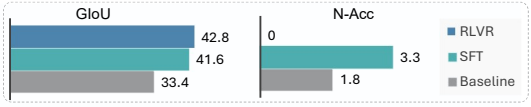
where the ϵ is a constant that controls the clipping boundary.

Method

We conduct a case study on non-object referring expression segmentation, where instructions include both valid expressions and "incorrect" ones.



(a) Non-Object Segmentation and VisionReasoner



(b) Performance comparison on gRefCOCO

(c) Visualization of model trained by pure RLVR

Figure: (a) Illustration on Non-Object Segmentation and VisionReasoner. (b) Performance comparison of SFT and RLVR on gRefCOCO. While RLVR achieves higher overall GIoU, it fails on non-object instructions. (c) Specifically, the RLVR model consistently outputs a mask even when no relevant object is detected.

Our analysis reveals:

- Model trained by SFT achieves suboptimal gIoU performance but tends to learn to correctly identify the absence of objects.
- Model trained with RLVR attain higher overall gIoU scores, yet they often generate object masks even when no relevant object is present. This limitation arises because the RLVR model, relying solely on self-rollouts, lacks the corrective mechanism necessary to produce a correct "no object" output.

In essence, RLVR drives performance gains through self-exploration, whereas SFT provides the external grounding necessary when exploration fails. This dichotomy motivates our central research question: *how can we efficiently synthesize the strengths of both training paradigms within a unified training stage?*

Gradient Analysis of SFT and RLVR.
The gradient of SFT can be derived from Equation (1) as:

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\mathrm{input}} \\ y\sim\mathcal{D}_{\mathrm{label}}}} \left[\nabla_\theta \log \pi_\theta(y \mid v, t)\right]. \tag{4}$$

The gradient of RLVR can be derived from Equation (3) using approximation $\pi_\theta \approx \pi_{\theta_{old}}$ and log-derivative trick:

$$\nabla_\theta \mathcal{L}_{\mathrm{RLVR}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\mathrm{input}} \\ \{o_j\}_{j=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}}} \left[\frac{1}{G}\sum_{j=1}^{G} \hat{A}_j \nabla_\theta \log \pi_\theta(o_j \mid v, t)\right]_{\theta\approx\theta_{\mathrm{old}}}. \tag{5}$$

We observe that the gradients of the SFT and RLVR losses, $\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta)$ and $\nabla_\theta \mathcal{L}_{\mathrm{RLVR}}(\theta)$, share a similar form. The difference between them is the guidance signal (y vs. $\{o_j\}_{j=1}^{G}$) and coefficient (1 vs. $\hat{A}_j$).

To combine SFT and RLVR into a single stage, we design an objective function that naturally yields a gradient combining both $\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta)$ and $\nabla_\theta \mathcal{L}_{\mathrm{RLVR}}(\theta)$.
We construct an augmented rollout set $y \cup \{o_j\}_{j=1}^{G}$. Then the corresponding rewards are $\mathsf{r}(y) \cup \{\mathsf{r}(o_j)\}_{j=1}^{G}$. This formulation modifies the advantage calculation of rollouts in Equation (2) as follows:

$$\hat{A}_j = \frac{\mathsf{r}(o_j) - \mathsf{mean}\left(\mathsf{r}(y) \cup \{\mathsf{r}(o_j)\}_{j=1}^{G}\right)}{\mathsf{std}\left(\{\mathsf{r}(y) \cup \{\mathsf{r}(o_j)\}_{j=1}^{G}\}\right)}, \tag{6}$$

and the advantage of ground-truth y is calculated as:

$$\hat{A}_y = \frac{\mathsf{r}(y) - \mathsf{mean}\left(\mathsf{r}(y) \cup \{\mathsf{r}(o_j)\}_{j=1}^{G}\right)}{\mathsf{std}\left(\{\mathsf{r}(y) \cup \{\mathsf{r}(o_j)\}_{j=1}^{G}\}\right)}. \tag{7}$$

The objective of ViSurf is to minimize the equation:

$$\mathcal{L}_{\text{ViSurf}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\text{input}} \\ \{o_j\}_{j=1}^{G}\sim\pi_{\theta_{\text{old}}} \\ y\sim\mathcal{D}_{\text{label}}}} \left[\frac{1}{G+1} \left(\sum_{j=1}^{G} \min \left\{ \frac{\pi_\theta(o_j \mid v,t)}{\pi_{\theta_{\text{old}}}(o_j \mid v,t)} \hat{A}_j,\ \text{clip}\left(\frac{\pi_\theta(o_j \mid v,t)}{\pi_{\theta_{\text{old}}}(o_j \mid v,t)},\ 1-\epsilon,\ 1+\epsilon \right) \hat{A}_j \right\} \right. \right. \tag{8}$$

$$\left. \left. + \min \left\{ \frac{\pi_\theta(y \mid v,t)}{\pi_{\theta_{\text{old}}}(y \mid v,t)} \hat{A}_y,\ \text{clip}\left(\frac{\pi_\theta(y \mid v,t)}{\pi_{\theta_{\text{old}}}(y \mid v,t)},\ 1-\epsilon,\ 1+\epsilon \right) \hat{A}_y \right\} \right) \right].$$

Algorithm 1: ViSurf Optimization Step

Input: policy model π_θ; reward function $\mathrm{r}(\cdot)$; input data $\mathcal{D}_{\text{input}}$; label data $\mathcal{D}_{\text{label}}$
for *step* $= 1, \ldots, M$ **do**
  Sample a mini-batch $\mathcal{B}_{\text{input}}$ and corresponding $\mathcal{B}_{\text{label}}$
  Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
  Sample G outputs $\{o_j\}_{j=1}^{G} \sim \pi_{\theta_{\text{old}}}()$ for each $(v, t) \in \mathcal{B}_{\text{input}}$
  Compute rewards $\{\mathrm{r}(o_j)\}_{j=1}^{G}$ for each sampled output o_j
  Compute rewards $\mathrm{r}(y)$ for label $y \in \mathcal{B}_{\text{label}}$
  Compute $\hat{A}_j$ and $\hat{A}_y$ through relative advantage estimation
  Update the policy model π_θ using Equation (8)
end
Output: π_θ

The gradient of Equation (8) can be derived using approximation $\pi_\theta \approx \pi_{\theta_{old}}$ and log-derivative trick:

$$\nabla_\theta \mathcal{L}_{\mathrm{ViSurf}}(\theta) = -\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\mathrm{input}} \\ \{o_j\}_{j=1}^{G}\sim\pi_{\theta_{\mathrm{old}}} \\ y\sim\mathcal{D}_{\mathrm{label}}}} \left[\frac{1}{G+1} \left(\sum_{j=1}^{G} \hat{A}_j \nabla_\theta \log \pi_\theta(o_j \mid v, t) + \hat{A}_y \nabla_\theta \log \pi_\theta(y \mid v, t) \right) \right]_{\theta\approx\theta_{\mathrm{old}}} . \tag{9}$$

The advantage $\hat{A}y$ for the ground-truth label y is always positive till now, as correct labels inherently receive higher rewards. However, this static setup is often sub-optimal; it can lead to reward hacking and suppresses the relative advantage $\hat{A}_j$ of self-generated rollouts, even when the policy has already produced correct trace and answers.

To mitigate these issues, we propose three reward control strategies:

- Aligning Ground-truth Labels with Rollouts Preference. This alignment minimizes the distribution shift between π_θ and $\pi_{\theta_{old}}$, thereby upholding the core assumptions $\pi_\theta \approx \pi_{\theta_{old}}$.
- Eliminating Thinking Reward for Ground-truth Labels. This ensures that the model learns to generate its own reasoning traces through self-rollouts.
- Smoothing the Reward for Ground-truth Labels. This smoothing ensures that the advantage for the ground-truth, $\hat{A}_y$, becomes zero (as per Equation (7)), eliminating the external supervision signal.

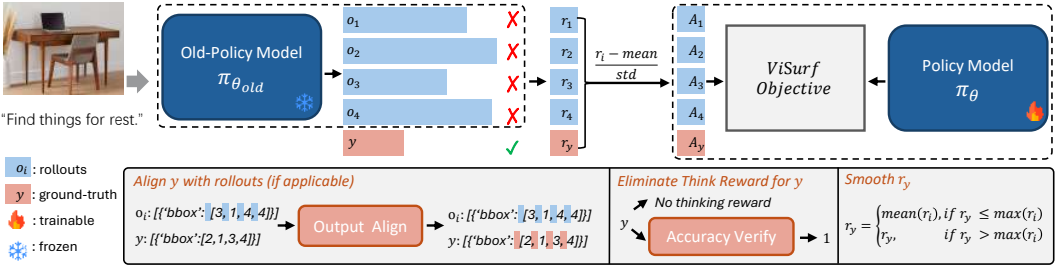


Figure: ViSurf Framework. Upper: The integration of external guidance y with internal guidance o_i, which is critical when self-rollouts are unsuccessful. Bottom: Three reward control strategies designed to regulate y, thereby preventing entropy collapse.

To better analysis the optimization step, we reformulate the gradient Equation (9) as following:

$$\nabla_\theta \mathcal{L}_{\text{ViSurf}}(\theta) =$$

$$\underbrace{-\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\text{input}} \\ \{o_j\}_{j=1}^{G}\sim\pi_{\theta_{\text{old}}}}}\left[\frac{1}{G+1}\sum_{j=1}^{G}\hat{A}_j\,\nabla_\theta \log \pi_\theta(o_j \mid v,t)\right]_{\theta\approx\theta_{\text{old}}}}_{\text{RLVR Term}}$$

$$\underbrace{-\mathbb{E}_{\substack{(v,t)\sim\mathcal{D}_{\text{input}} \\ y\sim\mathcal{D}_{\text{label}}}}\left[\frac{1}{G+1}\hat{A}_y\,\nabla_\theta \log \pi_\theta(y \mid v,t)\right]_{\theta\approx\theta_{\text{old}}}}_{\text{SFT Term}}. \tag{10}$$

Equation (10) integrates both the external guidance from SFT and the internal guidance from RLVR.

- The RLVR term in Equation (10) is structurally identical to the standard RLVR gradient in Equation (5), differing only in its scaling coefficient ($\frac{1}{G+1}\hat{A}_j$ vs. $\frac{1}{G}\hat{A}_j$).
- The SFT term in Equation (10) resembles the SFT gradient from Equation (4), with two key distinctions: (i) the coefficient is weighted by $\frac{1}{G+1}\hat{A}_y$ instead of 1, and (ii) the use of the approximation $\pi_\theta \approx \pi_{\theta_{old}}$.

Experiments

- We verify ViSurf on benchmarks across several domains, including Non-Object Segmentation (e.g., gRefCOCO), Reasoning Segmentation (e.g., ReasonSeg), GUI Grounding (e.g., OmniACT), Industrial Anomaly Detection (e.g., RealIAD), Medical Imaging (e.g., ISIC2018), and Mathematical Reasoning (e.g., MathVista).
- We instantialize ViSurf algorithm with Qwen2.5VL-7B and adopt SAM2 if needed.

Table: Comparison on different benchmarks in different domains under different training paradigms.

Method	Non-Object Segmentation				GUI	Anomaly	Medical:Skin	Math	Avg
	gRefCOCO		**ReasonSeg**		**OmniACT**	**RealIAD**	**ISIC2018**	**MathVista**	
	val		val	test	test	subset	test	test-mini	
	gIoU	N-Acc	gIoU	gIoU	Acc	ROC_AUC	Bbox_Acc	Acc	
Baseline	33.4	1.8	56.9	52.1	60.4	50.1	78.8	68.2	50.2
SFT	41.6	3.3	63.8	60.3	55.4	65.5	91.7	68.3	56.2
RLVR	42.8	0.0	66.0	63.2	65.5	50.0	90.3	71.2	56.1
SFT → RLVR	65.0	52.1	57.2	55.2	64.5	66.9	93.6	68.5	65.4
ViSurf	**66.6**	**57.1**	**66.5**	**65.0**	**65.6**	**69.3**	**94.7**	**71.6**	**69.6**

- Empirical evaluations demonstrate that ViSurf consistently surpasses existing methodologies across all evaluated domains, achieving a substantial average relative improvement of 38.6% over the baseline.

Table: Employ ViSurf on Qwen2VL-7B.

Method	**RealIAD** subset	**ISIC2018** test
	ROC_AUC	Bbox_Acc
Baseline	60.0	51.8
SFT	56.7	94.2
RLVR	57.1	90.5
SFT $\rightarrow$ RLVR	67.5	94.6
ViSurf	**76.0**	**95.4**

- Both RLVR and ViSurf demonstrate robustness against catastrophic forgetting.
- SFT and SFT $\rightarrow$ RLVR suffer from performance degradation, which is attributable to catastrophic forgetting.

Table: Ablation of Reward Control Strategy. The first row is the non-trained baseline. 'Align': Aligning ground-truth labels with rollouts; 'Eliminate': Eliminating thinking format reward for groud-truth labels; 'Smooth': Smoothing accuracy reward for ground-truth labels; '-': not applicable.

Align	**Eliminate**	**Smooth**	**gRefCOCO** val		**ReasonSeg** val	**MathVista** testmini
			gIoU	N-Acc	gIoU	Acc
–	–	–	33.4	1.8	56.9	68.2
✗	✓	✓	59.0	40.2	63.6	—
✓	✗	✓	**72.9**	**74.1**	58.2	67.1
✓	✓	✗	61.0	45.7	62.7	66.8
✓	✓	✓	66.6	57.1	**66.5**	**71.6**

- The reward control mechanism is necessary.

Qualitative results on various tasks are presented in Figure 4. The results demonstrate that models trained with ViSurf successfully solves multiple visual perception tasks.

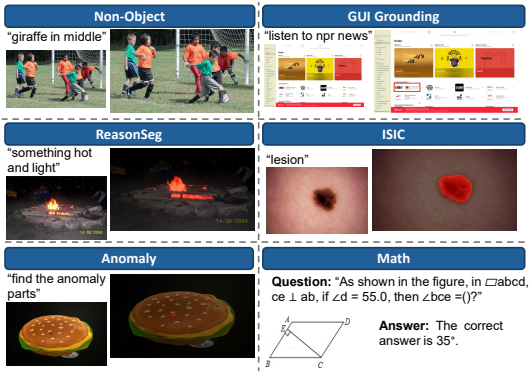


Figure: Visualization of ViSurf on various tasks.

Conclusion

- **Unified Framework**: We introduce ViSurf, a theoretically grounded, single-stage paradigm that unifies SFT and RLVR to achieve simultaneous knowledge injection and internal reinforcement.
- **Stability Optimization**: We design three novel reward control strategies to stabilize joint optimization, effectively balancing ground-truth guidance with on-policy exploration.
- **State-of-the-Art Performance**: ViSurf consistently outperforms standalone SFT, RLVR, and two-stage pipelines across diverse benchmarks, supported by in-depth analysis of its synergistic mechanics.

THANK YOU!