A. Pseudocodes

We present the GTR-Turbo pseudocodes, both for the SFT and KL thought guidance variants.

Algorithm 1 Training Procedure of GTR-Turbo (SFT)

Input: Environment `env`, agent model π_{θ_0}, Replay buffer size B, update epoch K
$\mathcal{C} \leftarrow [\pi_{\theta_0}]$ ▷ Checkpoint buffer
$\mathcal{D} \leftarrow \varnothing$ ▷ Thought dataset
for $k = 0$ to $K - 1$ **do**
  $\mathcal{B} \leftarrow \varnothing$ ▷ On-policy RL data buffer
  Obtain $\pi^{(k)}_{\text{merged}}$ by merging all checkpoints in $\mathcal{C}$ ▷ Eqn. 3
  $o_t = \texttt{env.reset()}$
  while $|\mathcal{B}| < B$ **do**
    Generate (th_t, a_t) using π_{θ_k} given o_t
    Generate $(\hat{th}_t, \hat{a}_t)$ using $\pi^{(k)}_{\text{merged}}$ given o_t ▷ Reference thought
    $r_t, o_{t+1} = \texttt{env.step}(a_t)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup (o_t, a_t, r_t, o_{t+1})$
    $\mathcal{D} \leftarrow \mathcal{D} \cup (o_t, \hat{th}_t)$
  Sample mini-batch b from $\mathcal{B}$, d from $\mathcal{D}$
  Compute $\mathcal{L}_{\text{PPO}}$ with b
  Compute $\mathcal{L}_{\text{SFT}}$ with d ▷ Eqn. 2
  $\theta_{k+1} = \arg\min_\theta(\mathcal{L}_{\text{PPO}} + \mathcal{L}_{\text{SFT}})$ ▷ Eqn. 5
  $\mathcal{C} \leftarrow \mathcal{C} \cup \pi_{\theta_{k+1}}$
Output: π_{θ_K}

Algorithm 2 Training Procedure of GTR-Turbo (KL)

Input: Environment `env`, agent model π_{θ_0}, Replay buffer size B, update epoch K
$\mathcal{C} \leftarrow [\pi_{\theta_0}]$ ▷ Checkpoint buffer
for $k = 0$ to $K - 1$ **do**
  $\mathcal{B} \leftarrow \varnothing$ ▷ On-policy RL data buffer
  Obtain $\pi^{(k)}_{\text{merged}}$ by merging all checkpoints in $\mathcal{C}$ ▷ Eqn.3
  $o_t = \texttt{env.reset()}$
  while $|\mathcal{B}| < B$ **do**
    Generate (th_t, a_t) using π_{θ_k} given o_t
    Calculate RevKL $\left(\pi_{\theta_k}, \pi^{(k)}_{\text{merged}}; th_t\right)$ ▷ Eqn. 6
    $r_t, o_{t+1} = \texttt{env.step}(a_t)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \left(o_t, a_t, r_t - \beta \cdot \text{RevKL}\left(\pi_{\theta_k}, \pi^{(k)}_{\text{merged}}; th_t\right), o_{t+1}\right)$
  Sample mini-batch b from $\mathcal{B}$
  Compute $\mathcal{L}_{\text{PPO}}$ with b
  $\theta_{k+1} = \arg\min_\theta \mathcal{L}_{\text{PPO}}$
  $\mathcal{C} \leftarrow \mathcal{C} \cup \pi_{\theta_{k+1}}$
Output: π_{θ_K}

In GTR-Turbo (KL), β controls the contribution of the reserve KL term within the reward. Throughout this paper, we use the default setting of $\beta = 1$.

B. Additional Details on Training

B.1. Training Setting

Drawing inspiration from the common practice in RL post-training frameworks, [3, 5, 7], we perform one epoch of supervised fine-tuning on the base Qwen2.5-VL [1] model before RL training, so that the agent possesses a basic instruction-following capability. The datasets are sourced from the RL4VLM paper [7], with labels for the Points24 provided by a task solver and labels for the ALFWorld environment generated by GPT-4V.

B.2. Hyperparameters

We provide the hyperparameters used for GTR-Turbo training in Table 1, which are primarily derived from previous work [5, 7]. We employ LoRA [2] to fine-tune the entire VLM model.

Hyperparameter		**Value**
General Setup - Training		
Learning rate		`CosineAnnealingLR`
Initial learning rate		$1e-5$
Final learning rate		$1e-9$
Maximum learning rate step		25
Discount factor γ		0.9
GAE λ		0.95
PPO entropy coefficient		0.01
PPO value loss coefficient		0.5
PPO clip parameter c		0.1
PPO epoch		4
Gradient accumulation steps		128
LoRA r		128
LoRA α		256
LoRA `dropout`		0.05
KL loss coefficient β	(for KL guidance)	1
General Setup - Models		
Generation max text length		256
Generation temperature		0.2
Generation repetition penalty		1.2
Model Merging Method		TIES
TIES Density		0.8
Teacher Generation base temperature	(for SFT guidance)	0.2
Teacher Generation max temperature	(for SFT guidance)	0.9
Teacher Generation temperature retry coefficient	(for SFT guidance)	1.1
For Points24 task		
Environmental steps		30000
Thought probability coefficient		0.5
For ALFWorld task		
Environmental steps		20000
Thought probability coefficient		0.2

Table 1. **Hyperparameters of GTR-Turbo**

C. Additional Experiment Results

C.1. Results on stronger and more recent models

We also evaluate the efficacy of GTR-Turbo using the newly released Qwen3-VL-8B-Instruct model. We evaluate on ALFWorld using the KL variant of GTR-Turbo. The results show that GTR-Turbo remains compatible with the latest model family, and the stronger base capability of Qwen3-VL leads to improved performance, even surpassing the success rate of Qwen2.5-VL-32B, a model that is four times larger in scale.

Moreover, we observe that, in general-knowledge reasoning tasks such as ALFWorld, Qwen3-VL can perform RL directly without any SFT initialization. This suggests that as foundation models continue to evolve, GTR-Turbo may become even simpler to use and more broadly applicable.

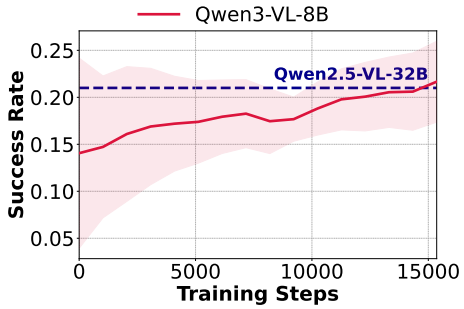

Figure 1. **Result of Qwen3-VL-8B on ALFWorld.**

C.2. Additional Experiments on other open-ended tasks

We conduct an experiment on the challenging GUI benchmark Android-in-the-Wild (AitW) [4] using Qwen3-VL-8B-Instruct, as shown in Table 2. GTR-Turbo still outperforms strong DigiRL and PPO baselines without heavy hyperparameter tuning and reward shaping. Moreover, we also compare the quality of reasoning traces between PPO and GTR-Turbo using GPT-5.2 with a simple LLM-as-a-judge method. These results demonstrate the efficacy of GTR-Turbo across diverse visual environments.

Method	Success Rate	Reasoning Score
DigiRL	71.9%	-
PPO	75.0%	3.26
GTR-Turbo	**80.2%**	**3.93**

Table 2. **Experiment results on Android-in-the-Wild.**

C.3. Pass@k Comparison

Following prior work [6], we use $pass@k$ success rate with increasing k until convergence ($k = 32$ achieves the same performance as $k = 16$) to assess the upper bound of the base model's capability. Figure 2 shows that the agent trained by GTR-Turbo can easily surpass the ceiling, indicating that GTR-Turbo enables the model to acquire capabilities beyond its original distribution.

C.4. Ablation study regarding merging frequency

In Figure 3, we ablate the merging frequency. GTR-Turbo continues to yield appealing results up to a merging interval of 10, demonstrating its robustness to this hyperparameter.

C.5. Reasoning Score Evaluation

To further verify whether GTR-Turbo can mitigate "thought collapse", we employ an LLM-as-a-judge approach using GPT-5.2 to evaluate the quality of agent-generated reasoning traces during RL in terms of factual correctness, logical rigor, and coherence. Figure 4 shows that the RL-trained agent without guidance from the merged teacher exhibits a clear "thought

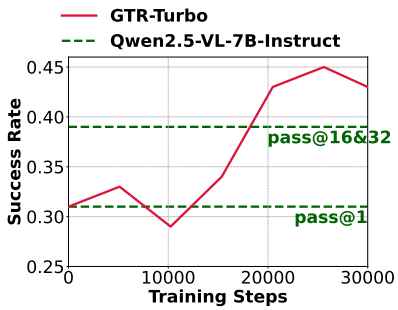

Figure 2. **Comparison of GTR-Turbo training curve with the pass@k results of the base model.**

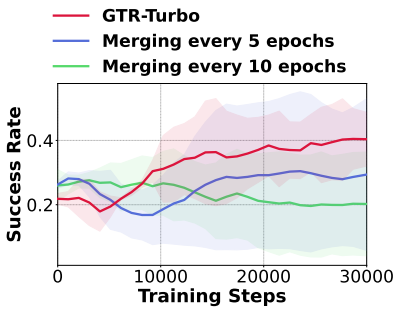

Figure 3. **Success rate results of GTR-Turbo using different merging frequencies.**

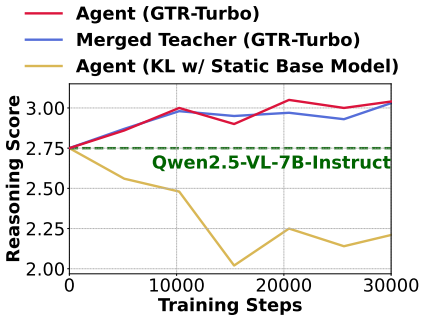

Figure 4. **Reasoning score evaluation of models in GTR-Turbo and baseline with static teacher.**

collapse" phenomenon. This indicates that the merged checkpoint is certainly a teacher with better reasoning rather than a simple regularized reference.

D. Additional Details on Environments

We provide a detailed introduction to the experimental environments used in this study.

D.1. Points24

State and action space. At each observation o_t in the Points24 task, the agent observes an image showing four poker cards and a text-based representation of the current formula. The goal is to form a formula equal to 24 using the numbers represented by the four cards and basic operators. Cards "J", "Q", "K" are all treated as number 10. The action space includes {"1", "2", . . ., "10", "+", "-", "*", "/", "(", ")", "="}, and each card can only be used once. Selecting a number not present in the image or one that has already been used is considered an illegal action. If the action is legal, the corresponding number or operator is appended to the current formula, forming the next observation o_{t+1}; if the action is illegal, the state remains unchanged $o_{t+1} = o_t$. The environment does not guarantee that the four cards in the image have a feasible solution equal to 24.

Reward function. At each step, the agent receives a reward $r = -1$ for outputting an illegal action and a reward $r = 0$ for a legal action. The episode terminates when the agent outputs "=" as an action or the step count exceeds $T = 20$. At termination, if the formula evaluates to 24, the agent receives an outcome reward $r = 10$; otherwise, it receives $r = -1$.

D.2. ALFWorld

State and action space. In the ALFWorld environment in our experiments, the agent receives an RGB observation image and a history of past actions at each observation o_t. The action space includes all possible interactions in the current scenario, typically categorized as: (1) `go to` {`recep`}, (2) `take` {`obj`} `from` {`recep`}, (3) `put` {`obj`} `in/on` {`recep`}, (4) `open` {`recep`}, (5) `close` {`recep`}, (6) `toggle` {`obj`} {`recep`}, (7) `clean` {`obj`} `with` {`recep`}, (8) `heat` {`obj`} `with` {`recep`}, (9) `cool` {`obj`} `with` {`recep`}, where {`obj`} and {`recep`} denote objects and receptacles. After an admissible action is taken, ALFWorld renders the updated scene from the agent's view as the next observation o_{t+1}. $o_{t+1} = o_t$ if the action is illegal.

Notably, the ALFWorld environment provides both an image and a text description of the observation scene at each step. As noted in GTR, the VLM agent may rely heavily on textual descriptions rather than visual observations, which contradicts the purpose of visual agentic tasks. GTR therefore modified the state by removing the text description, which we adopt in GTR-Turbo. We also align with GTR by including the action history in the input prompt to more closely simulate real-world scenarios. These adjustments increase the task's difficulty, thereby emphasizing the agent's comprehensive visual recognition and long-horizon decision-making capabilities.

Reward function. The reward of ALFWorld consists of two components. Each observation o has a set of admissible actions $\mathcal{A}_{\text{adm}}(s)$, and illegal actions are penalized. Additionally, each task in ALFWorld has both the final goal g_{task} and sub-goals g_{sub}, and achieving these goals also provides rewards. Formally, the reward function can be written as:

$$r(s_t, a_t, s_{t+1}|g_{\text{task}}) = 50 \times \mathbf{1}(s_{t+1} = g_{\text{task}}) + \mathbf{1}(s_{t+1} = g_{\text{sub}}) - \mathbf{1}(a_t \notin \mathcal{A}_{\text{adm}}(s)). \quad (1)$$

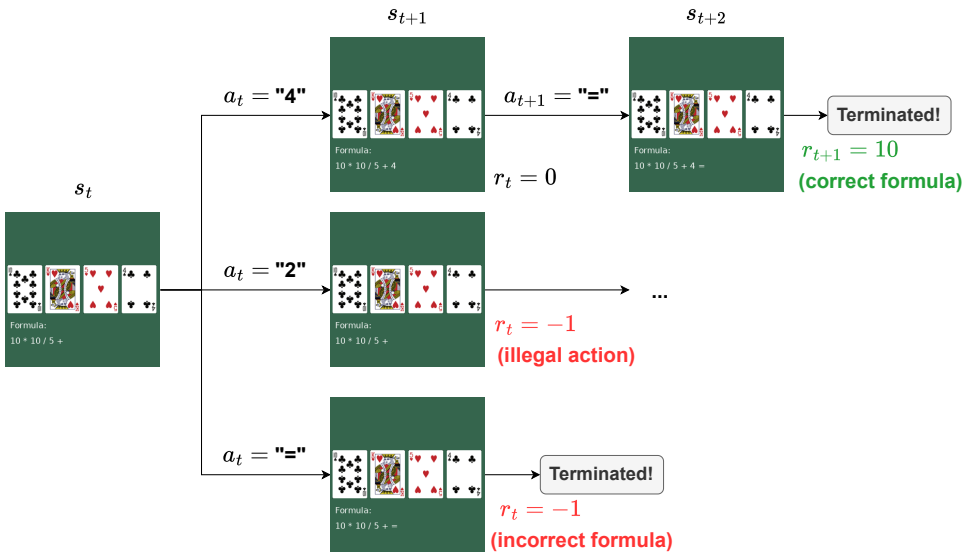

Figure 5. **The Points24 task.**

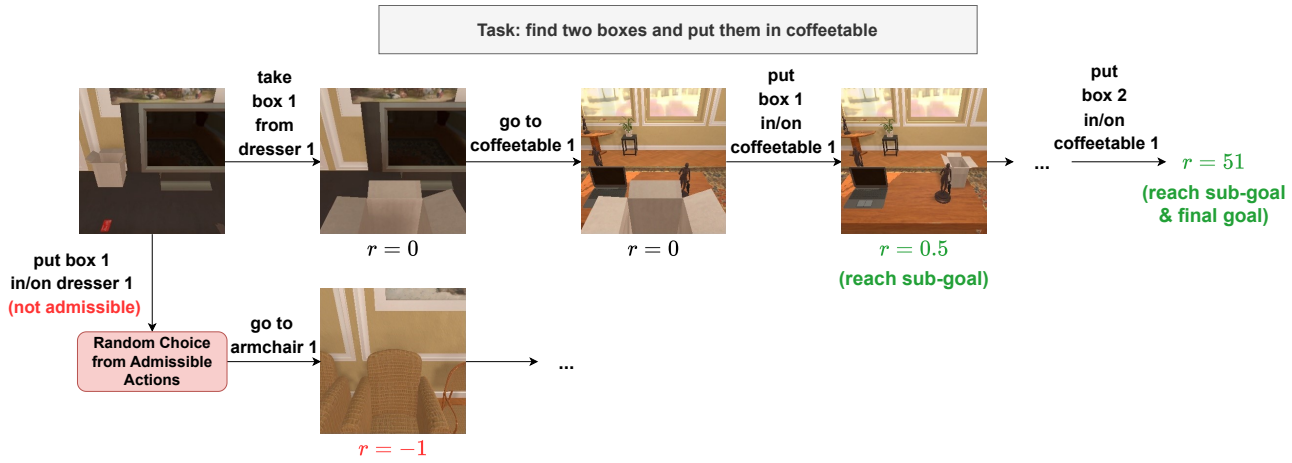

Figure 6. **The ALFWorld task.**

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 2

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 2

[3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2

[4] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Androidinthewild: A large-scale dataset for android device control. *Advances in Neural Information Processing Systems*, 36:59708–59728, 2023. 3

[5] Tong Wei, Yijun Yang, Junliang Xing, Yuanchun Shi, Zongqing Lu, and Deheng Ye. Gtr: Guided thought reinforcement prevents thought collapse in rl-based vlm agent training. *arXiv preprint arXiv:2503.08525*, 2025. 2

[6] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. 3

[7] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2025. 2