VLA-CoRe: Empowering Vision Language Action Model with Cognitive Reasoning

Anonymous authors
Paper under double-blind review

Abstract

Vision Language Action (VLA) models have demonstrated strong behavioral cloning performance on short-horizon manipulation tasks, yet they fundamentally lack *cognitive* awareness: the ability to estimate their own uncertainty, identify the source of potential failure, and selectively invoke specialized reasoning modules before committing to an action. We propose VLA-CoRe, a cognitive VLA framework that augments a pretrained multimodal LLM with an Uncertainty Monitor that gates three differentiable reasoning modules: (1) Perception Query, which resolves ambiguity by re-examining occluded or visually complex scene regions; (2) World Imagination, which performs counterfactual "what-if" simulations to predict the outcome of an action before physical execution; and (3) Causal Tracing, which attributes unexpected failures to specific past reasoning steps to revise the current plan, based on uncertainty estimates. The reasoning outcomes (from these three reasoning modules) are combined into a *visual plan latent* that conditions a Diffusion Transformer action decoder for robust low-level control. To train VLA-CoRe without expensive manual annotation, we introduce **CIDAS** (**C**ognitive **I**nstruction **D**ata for **A**utonomous robot rea**S**oning), a dataset of $\sim$170k conversation-style cognitive reasoning samples extracted from existing behavior cloning trajectories across seven structured task types. Training combines (i) a Supervised Fine-Tuning stage on CIDAS and (ii) Group Relative Policy Optimization (GRPO) guided by a cognitive objective. Crucially, our framework improves failure recovery rates by 21% and reduces expected calibration error by 38%, marking a significant step toward resilient & self-aware robots. Code & data will be released.

1 Introduction

A robot executing a manipulation task makes dozens of sequential decisions under perceptual uncertainty. At each step, contemporary Vision-Language-Action (VLA) models (Kim et al., 2024; Brohan et al., 2023a; Zhao et al., 2025; Huang et al., 2026) emit an action token with no accompanying estimate of how confident that decision is, a design choice inherited from behavioral cloning, where the training objective provides no incentive for self-assessment. This conflation of prediction and confidence is not a mere theoretical inelegance: it is a direct cause of compounding failure in long-horizon tasks. A model that cannot distinguish a confident grasp from a speculative one will not know to pause, re-examine the scene, or revise its plan before an error becomes unrecoverable. We formalize this as a *confidence gap*: the divergence between a model's implicit confidence in its predictions and the empirical probability of success at the corresponding timesteps. To quantify it, we instrument five state-of-the-art VLAs on LIBERO-Long (Liu et al., 2023a) and retroactively label each timestep with the uncertainty score introduced in this work (section 4). We find that **63% of episode failures** occur at timesteps assigned uncertainty $u_t > 0.6$, yet every model proceeds unconditionally, issuing an action at the same rate as at fully confident ($u_t < 0.2$) timesteps. This indicates that these models are systematically overconfident where caution is most warranted.

We argue that closing this reasoning gap requires equipping VLAs with *cognitive* capacity: the ability to monitor their own internal state, estimate epistemic uncertainty, and selectively invoke targeted reasoning modules before committing to action. This is distinct from simply adding a reasoning chain (Zawalski et al., 2024; Clark et al., 2025): an agent that reasons but ignores its own uncertainty will still over-commit at critical junctures. What is needed is a *gated* architecture where real-time

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

uncertainty estimates act as a switch, selectively invoking one of specialized reasoning modules only when the model's confidence dictates that deeper deliberation is required.

We propose VLA-CoRe (VLA with Cognitive Reasoning), a framework that formalizes this inductive bias through three tightly coupled components. First, an *Uncertainty Monitor* $\mathcal{U}_\psi$ reads the LLM hidden state $\mathbf{h}_t$ at each timestep and produces a scalar confidence estimate $c_t \in (0, 1)$ and a binary failure flag $f_t \in \{0, 1\}$. Second, based on (c_t, f_t), the monitor routes execution to one of three differentiable reasoning modules: a *Perception Query*, invoked under moderate confidence, that re-encodes a task-relevant scene crop to resolve localized visual ambiguities, a *World Imagination* module that simulates counterfactual action outcomes to anticipate and prevent potential execution errors when highly uncertain, and a *Causal Tracer* that performs cross-attention backward attribution over a step history upon failure detection to diagnose the root cause of unexpected failures for corrective replanning. Steps with very high c_t bypass all reasoning modules entirely, preserving computational efficiency. Third, reasoning outputs are processed into a *visual plan latent* $\mathbf{z}_t$ that conditions a Diffusion Transformer (DiT) action decoder (Peebles & Xie, 2023).

The central challenge in training VLA-CoRe is *lack of cognitive reasoning dataset*: standard robot demonstration datasets encode no signal about *when* to invoke reasoning or *why* a particular reasoning direction is appropriate. We address this by introducing CIDAS (**C**ognitive **I**nstruction **D**ata for **A**utonomous Robot rea**S**oning), a dataset of ∼170k conversation-style samples automatically extracted from existing HDF5 demonstration files across seven structured task types. CIDAS requires no manual annotation and no external model calls for its core generation: cognitive labels are derived entirely from epistemic uncertainty signals computable in closed form from pixel statistics and action kinematics. To optimize VLA-CoRe, we propose a two-stage training paradigm. We first perform supervised fine-tuning on CIDAS to establish reasoning behaviors, and subsequently refine the policy using Group Relative Policy Optimization (GRPO (Shao et al., 2024)). Our contributions are:

- **Automated Epistemic Data Synthesis.** To overcome the absence of reasoning labels in standard behavioral cloning data, we introduce CIDAS. By mapping physical observations (occlusion, depth complexity, and kinematic spikes) to uncertainty bounds, we automatically extract 171K complex reasoning conversations across seven task types from existing LIBERO and Open-X trajectories, entirely bypassing manual annotation (section 4).
- **A Cognitive Reasoning-based VLA.** We propose VLA-CoRe, a framework that bridges high-level deliberation and low-level control. We introduce a learned *Uncertainty Monitor* that projects LLM hidden states into calibrated epistemic confidence scores to dynamically gate three differentiable reasoning modules: Perception Query, World Imagination, and Causal Tracing. The resulting reasoning outcomes are fused into a visual plan latent to condition a DiT action decoder, optimized via a two-stage SFT→GRPO training pipeline with cognitive rewards (section 5).
- **New Cognitive Benchmarks and SoTA Performance.** We establish a rigorous evaluation paradigm for robot reasoning by formalizing Expected Calibration Error (ECE), Reasoning Efficiency, and Failure Recovery Rate as principled VLA metrics. Under this framework, VLA-CoRe sets new state-of-the-art records on LIBERO-Long, Calvin ABC-D, and MetaWorld-Hard, improving failure recovery by over 21% compared to the strongest prior reasoning baselines (section 6).

2 Related Work

Vision-Language-Action Models. VLA paradigms emerged from co-fine-tuning internet-scale VLMs (RT-2 (Brohan et al., 2023a); OpenVLA (Kim et al., 2024)) and cross-embodiment data (Open X-Embodiment (O'Neill et al., 2024)). Recent generalist transformers include Octo (Team et al., 2024), RoboVLMs (Li et al., 2024d), and the high-frequency flow-matching agents π_0 / $\pi_{0.5}$ (Black et al., 2024; Intelligence et al., 2025). Dual-stage frameworks (GR00T N1 (Bjorck et al., 2025); Helix (Figure AI, 2025)) partition slow System 2 reasoning from fast System 1 reactivity. For spatial grounding, works employ visual traces (TraceVLA (Zheng et al., 2025)), trajectories (LLARVA (Niu et al., 2024)), affordance reasoning (CoA-VLA (Li et al., 2024a)), or auxiliary datasets (LLaRA (Li et al., 2024b)). CIDAS extends this via *cognitive reasoning*, providing calibrated uncertainty and reasoning-routing labels rather than spatial descriptions.

Reasoning and RL-Based Training. VLA training increasingly embeds reasoning via plan-grounding (SayCan (Brohan et al., 2023b); Inner Monologue (Huang et al., 2022)), linguistic chain-of-thought (ECoT (Zawalski et al., 2024); RAD (Clark et al., 2025); ManipLLM (Li et al., 2024c)), or visual subgoals (CoT-VLA (Zhao et al., 2025)). RL-optimized reasoning, utilizing

Table 1: **CIDAS statistics.** All samples are auto-generated from LIBERO (50K) and Open-X Bridge v2 (30K) demonstrations. Token counts use Vicuna-7B tokenization.

Task	Cognitive Skill	Train	Val	Avg. Tokens
T1 – Uncertainty Dialogue	Confidence calibration & routing	42,500	5,000	178
T2 – Counterfactual Branch	World imagination grounding	25,500	3,000	224
T3 – Failure Attribution	Causal tracing (failed trajs.)	21,250	2,500	246
T4 – Reasoning Module Selection Rationale	Decision policy with rationale	21,250	2,500	203
T5 – Subgoal Decomposition	Long-horizon planning	17,000	2,000	312
T6 – Recovery Reasoning	Full VLA-CoRe loop	12,750	1,500	285
T7 – Imagination Verification	World model calibration	12,750	1,500	198
Total	-	**153,000**	**18,000**	221

GRPO (Shao et al., 2024) or DeepSeek-R1 (Guo et al., 2025) techniques, includes ThinkAct (action-aligned rewards (Huang et al., 2026)), VLA-R1 (verifiable rewards (Ye et al., 2026)), RoboAlign (accuracy-based refinement (Kim et al., 2026)), and VLA-RL (multi-turn conversation (Lu et al., 2025)). Unlike these unconditional approaches, VLA-CoRe *gates* deliberation via explicit uncertainty monitoring and cognitive reasoning modules.

Uncertainty Quantification and Tool-Augmentation. Reliable uncertainty estimates, such as MC-Dropout (Gal & Ghahramani, 2016) and deep ensembles (Lakshminarayanan et al., 2017), facilitate out-of-distribution detection in BC and RL (Chua et al., 2018). Conformal prediction (Vovk et al., 2005) provides safety and coverage guarantees ((Lindemann et al., 2023), Kumar et al. (2025)), while EPICSCORE (Cabezas et al., 2025) disentangles aleatoric and epistemic factors. Tool-augmented policies enable API calls (Toolformer (Schick et al., 2023)), retrieval (RAG (Lewis et al., 2020)), and failure diagnosis (AHA (Duan et al., 2024)). VLA-CoRe bridges the gap by being the first to integrate calibrated uncertainty with reasoning-gating via cognition.

3 Preliminaries

Problem Setting. We consider visuomotor policy learning as behavior cloning over an Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}c, \mathcal{O})$, where $\mathcal{S}$ is the state, $\mathcal{A}c$ is the action space of end-effector delta commands, and $\mathcal{O}$ is the space of RGB observations. At each timestep t, the model receives an observation $o_t \in \mathcal{O}$ and a language instruction ℓ, and predicts action $a_t \in \mathcal{A}c$ based on learned policy. Given expert demonstrations $\mathcal{D} = \{(o_t, \ell, a_t^*)\}$, standard behavioral cloning minimizes $\mathcal{L}_{\text{BC}} = \mathbb{E}_{\mathcal{D}}\left[\|a_t^* - \pi_\phi(o_t, \ell)\|_2^2\right]$ using learned policy π_ϕ. We extend this objective with two requirements: (i) the policy must maintain *calibrated* uncertainty estimates, and (ii) it must learn to gate deliberation by conditionally triggering one of K specialized reasoning modules $\{\mathcal{T}_k\}_{k=1}^K$ according to the model's self-assessed uncertainty.

Calibrated Uncertainty and ECE. A confidence predictor $c : \mathcal{O} \to [0, 1]$ is *perfectly calibrated* (Guo et al., 2017) if $\mathbb{P}(\text{success} \mid c(o_t, \ell) = p) = p$ for all $p \in [0, 1]$. Calibration (Naeini et al., 2015) is quantified via Expected Calibration Error: $\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{A} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|$, where B_m's are equal-width confidence bins, $A = \sum_m |B_m|$, $\text{acc}(B_m)$ is the empirical success rate, and $\text{conf}(B_m)$ is mean predicted confidence within the mth bin. Because policies are systematically overconfident under distribution shift (Ovadia et al., 2019), minimizing ECE is a prerequisite for using confidence as a reliable reasoning-gating signal.

VLM Backbone and Fine-Tuning. Following LLaVA (Liu et al., 2023b), our backbone comprises a frozen visual encoder $f_V(\cdot; \theta_V)$, trainable projection layer $f_{\text{MLP}}(\cdot; \theta_{\text{MLP}})$, and LLM $f_{\text{LLM}}(\cdot; \theta_{\text{LLM}})$. Visual tokens $\mathbf{v}_t = f_V(o_t, \theta_V)$ are projected into the LLM embedding space, concatenated with tokenized instruction embeddings, and decoded autoregressively to produce both reasoning outcomes and action tokens. We fine-tune θ_{MLP} and θ_{LLM} via LoRA (Hu et al., 2021), injecting low-rank updates to preserve pretrained world knowledge and optimizing trainable parameters.

4 The CIDAS Dataset

CIDAS is a dataset of 171K cognitive robot instruction conversations (statistics in Table 1) automatically extracted from existing behavioral cloning demonstrations without manual annotation. The source corpora are LIBERO (Liu et al., 2023a) (50K demos) and Open-X v2 (O'Neill et al., 2024) (30K demos).

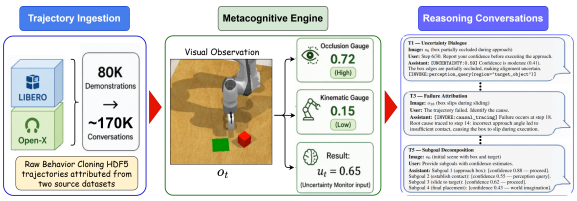

Figure 1: **CIDAS generation pipeline.** Each HDF5 trajectory is passed through seven task generators, each applying a guidance for one of cognitive conversation.

Data format. Every sample in CIDAS is a single-turn conversation consisting of an image o_t, a *user* turn (a question about the robot's epistemic

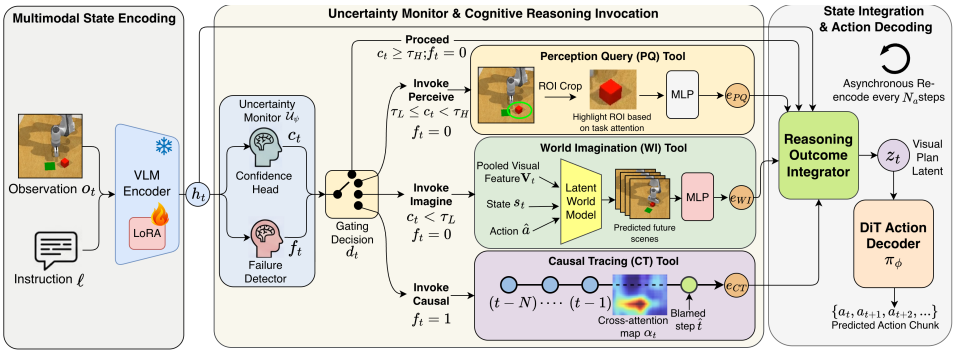

Figure 2: VLA-CoRe Overview. At each timestep t, observation o_t and instruction ℓ are encoded by a VLM to produce hidden state $\mathbf{h}_t$. An Uncertainty Monitor $\mathcal{U}_\psi$ predicts confidence c_t and failure flag f_t, routing execution to either proceed or invoke reasoning modules: Perception Query (visual disambiguation), World Imagination (counterfactual prediction), or Causal Tracing (failure attribution). Reasoning module outputs are fused into a visual plan latent $\mathbf{z}_t$, which conditions a Diffusion Transformer policy to generate action sequences.

state or action plan), and an *assistant* turn (the supervised target). The assistant turn contains at least one structured token from the vocabulary $\{$`[UNCERTAINTY:·]`, `[INVOKE:·]`, `[PROCEED]`$\}$, which VLA-CoRe is trained to emit and which the action decoder conditions on. A representative example from Task 1 (T1) is:

> **Image:** o_{14} (gripper approaching partially occluded mug)
> **User:** Step 14/28. Task: pick up the red mug. Report your confidence.
> **Assistant:** `[UNCERTAINTY:0.58]` Confidence is moderate (0.42). The mug handle is partially occluded by the bowl rim, making the grasp axis uncertain. I will invoke a perception query before committing to the approach. `[INVOKE:perception_query(region='target_object')]`

Task taxonomy. Seven task generators each target a distinct cognitive skill (Table 9). Generators differ only in *which aspect of the trajectory they analyze* and *which structured token(s) the assistant must emit*; all share the same uncertainty signal and routing rule from Equation 3. T1 teaches per-step confidence estimation and routing; T2 teaches counterfactual evaluation of alternative actions via world imagination; T3 teaches failure attribution and is extracted only from failed trajectories; T4 teaches the priority ordering among reasoning modules; T5 teaches long-horizon subgoal decomposition; T6 trains the full recovery loop (causal tracing followed by revised imagination); and T7 teaches the model to assess its own world model after an outcome, detailed in **Appendix-A**.

5 VLA-CoRe: Method

As seen in Figure 2, VLA-CoRe couples a VLM encoder with three components trained: an Uncertainty Monitor $\mathcal{U}_\psi$ that estimates per-step confidence and failure from the LLM's hidden state, a set of three differentiable reasoning modules $\{\mathcal{T}_k\}_{k=1}^3$ selectively invoked based on that estimate, and a Diffusion Transformer action decoder π_ϕ conditioned on the fused module output.

5.1 Architecture Design

VLM Encoder. We initialize the encoder from CLIP ViT-L/14@336 as the vision backbone and Vicuna-7B-v1.5 as the language model, connected by a two-layer MLP projection. The backbone produces $N_{\text{patch}} = (336/14)^2 = 576$ patch embeddings; we apply adaptive average pooling along the sequence dimension to project these to $N_v \times d_h$-dim visual tokens:

$$\mathbf{V}_t = \text{MLP}(\text{AdaptiveAvgPool}(\text{ViT}(o_t)_{1:},\ N_v)) \in \mathbb{R}^{N_v \times d_h}, \tag{1}$$

where the subscript $1{:}$ drops the `[CLS]` token. The LLM backbone is frozen in training stages; only LoRA adapter weights (Hu et al., 2021) inserted into all attention projection layers are updated.

Uncertainty Monitor. At each timestep t, the Monitor $\mathcal{U}_\psi$ reads the LLM hidden state $\mathbf{h}_t \in \mathbb{R}^{d_h}$ at the position of the final input token and produces two outputs. A *ConfidenceHead*, a three-layer MLP with GELU activations and a sigmoid output gate, yields a scalar confidence estimate:

$$c_t = \sigma\Big(\mathbf{W}_c^{(3)}\,\text{GELU}\Big(\mathbf{W}_c^{(2)}\,\text{GELU}\Big(\mathbf{W}_c^{(1)}\mathbf{h}_t + \mathbf{b}_c^{(1)}\Big) + \mathbf{b}_c^{(2)}\Big) + \mathbf{b}_c^{(3)}\Big) \in (0,1). \tag{2}$$

$\mathbf{W}_c^{(i)}; \mathbf{b}_c^{(i)}$ indicate trainable weights and bias vectors for the ith MLP layer. A *FailureDetector* monitors the attention entropy of the previous hidden states and raises a binary flag $f_t \in \{0, 1\}$ when the entropy falls outside a learned anomaly band. Together, (c_t, f_t) determine four routing decision:

$$d_t = \begin{cases} \texttt{CAUSAL} & f_t = 1; \\ \texttt{PROCEED} & c_t \geq \tau_H \text{ and } f_t = 0; \end{cases} \qquad \begin{cases} \texttt{PERCEIVE} & \tau_L \leq c_t < \tau_H \text{ and } f_t = 0 \\ \texttt{IMAGINE} & c_t < \tau_L \text{ and } f_t = 0 \end{cases} \tag{3}$$

where the failure flag f_t takes precedence, followed by calibrated thresholds τ_H and τ_L.

Reasoning Modules. Each module consumes the current observation and hidden state and emits a latent embedding $\mathbf{e}_k$ that is appended to the LLM context before the action token is generated.

Perception Query (PQ) is invoked when the model is moderately uncertain ($\tau_L \leq c_t < \tau_H$). To overcome the loss of fine-grained details in global image embeddings, PQ resolves localized visual ambiguity by isolating task-critical features. It localizes the relevant scene region by computing a cross-attention map $\mathcal{A}_\ell$ over the language tokens, crops the corresponding region of o_t, and re-encodes it at a higher effective resolution:

$$\mathbf{e}_{\text{PQ}} = \text{MLP}(\text{CLIP}(\text{RoICrop}(o_t, \mathcal{A}_\ell))) \,. \tag{4}$$

World Imagination (WI) is invoked when confidence is critically low ($c_t < \tau_L$). Because acting blindly under high uncertainty risks unrecoverable errors, WI enables prospective "what-if" reasoning to verify feasibility before physical execution. A compact latent world model, comprising a Transformer-based StateEncoder, a two-layer LatentTransition MLP, and a lightweight UNet decoder, predicts the visual embedding that would result from executing a candidate action $\hat{a}$:

$$\mathbf{e}_{\text{WI}} = \text{MLP}(\text{LatentTransition}(\text{StateEncoder}(\mathbf{V}_t, \mathbf{s}_t), \hat{a})), \tag{5}$$

where $\mathbf{s}_t$ is the proprioceptive state. The world model is pretrained offline on (state, action, next-state) tuples before integration.

Causal Tracing (CT) is invoked when the *FailureDetector* fires ($f_t = 1$). Since failures are often the delayed consequence of earlier compounding errors, CT is designed to provide diagnostic backward attribution rather than purely reactive correction. It performs cross-attention over a history buffer $\mathcal{H} = \{(\mathbf{h}_{t'}, a_{t'})\}_{t'=t-N}^{t-1}$ to locate the specific past step most responsible for the current failure:

$$\alpha_{t'} = \text{softmax}\left(\frac{\mathbf{h}_t \mathbf{W}_Q \, (\mathbf{h}_{t'} \mathbf{W}_K)^\top}{\sqrt{d_k}}\right), \qquad \mathbf{e}_{\text{CT}} = \sum_{t'} \alpha_{t'} \, \mathbf{h}_{t'} \mathbf{W}_V, \tag{6}$$

with the blamed step $\hat{t} = \arg\max_{t'} \alpha_{t'}$ surfaced as a structured token in the model's output to condition subsequent replanning. Here $\mathbf{W}_{\{Q,K,V\}}$ represent query, key, value metrics, respectively.

Reasoning Outcome Integrator. Embeddings from the invoked modules are aggregated into a single visual plan latent via cross-attention over the current hidden state:

$$\mathbf{z}_t = \text{CrossAttn}\Big(\mathbf{h}_t, \text{concat}\big[\mathbf{e}_k\big]_{k \in \mathcal{K}_t}\Big) + \mathbf{h}_t \in \mathbb{R}^{d_h}, \tag{7}$$

$\mathcal{K}_t \subseteq \{1, 2, 3\}$ is reasoning module at step t (empty when $d_t = \texttt{PROCEED}$, where $\mathbf{z}_t = \mathbf{h}_t$).

Diffusion Transformer Action Decoder. The visual plan latent $\mathbf{z}_t$ conditions a 6-layer Diffusion Transformer (DiT) (Peebles & Xie, 2023) with adaLN-Zero blocks that predict a chunk of $N_a = 8$ future actions per reasoning step. Training minimizes the standard denoising objective with $K = 1000$ steps (Ho et al., 2020):

$$\mathcal{L}_{\text{action}} = \mathbb{E}_{a_t, \, \epsilon \sim \mathcal{N}(0,I), \, k}\Big[\big\|\epsilon - \epsilon_\phi\big(\sqrt{\bar{\alpha}_k}\, a_t + \sqrt{1 - \bar{\alpha}_k}\, \epsilon, \; \mathbf{z}_t, \; k\big)\big\|^2\Big] \,. \tag{8}$$

At inference, actions are decoded in 10 DDIM steps (Song et al., 2021). As reasoning MLLM reasons every N_a steps while the decoder executes at full control frequency.

5.2 Training

Stage 1: Supervised Fine-Tuning on CIDAS. We jointly fine-tune the MLP projection, LoRA adapters, and monitor parameters using the standard next-token prediction loss, computed only over assistant tokens via prompt-masking:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(o_t, \, \ell, \, \mathbf{w}) \sim \mathcal{D}}\Big(\frac{1}{|\mathcal{T}_{\text{asst}}|} \sum_{i \in \mathcal{T}_{\text{asst}}} \log p_\theta(w_i \mid w_{<i}, \, o_t, \, \ell)\Big), \tag{9}$$

where $\mathbf{w} = (w_1, \ldots, w_L)$ is the full token sequence of a training sample, $\mathcal{T}_{\text{asst}} \subseteq \{1, \ldots, L\}$ is the index set of assistant tokens (prompt tokens are masked from the loss). For T1 and T4 samples, which carry explicit uncertainty labels u_t, we add an auxiliary binary cross-entropy loss on the ConfidenceHead's scalar output, providing direct calibration supervision beyond what can be inferred from the token sequence alone.

Stage 2: GRPO with Cognitive Reward. Following DeepSeekMath (Shao et al., 2024), we optimize the language model using Group Relative Policy Optimization. For each prompt, $G = 8$ rollouts are sampled from the current policy and scored by a reward function, and group advantage is computed by normalizing rewards r_i (defined in Equation 12) within the group:

$$A_i = \frac{r_i - \mu_r}{\sigma_r + \epsilon}, \qquad \mu_r = \frac{1}{G}\sum_{i=1}^{G} r_i, \quad \sigma_r = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(r_i - \mu_r)^2}. \tag{10}$$

The policy is updated by the clipped surrogate objective regularized by a per-token KL penalty D_{KL} against the frozen reference policy $p_{\theta_{\text{old}}}$. With $\rho_i = p_\theta(z_i)/p_{\theta_{\text{old}}}(z_i)$:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G} \min(\rho_i A_i,\ \text{clip}(\rho_i,\, 1-\varepsilon,\, 1+\varepsilon)\, A_i) - \beta\, D_{\text{KL}}(p_\theta \| p_{\theta_{\text{old}}}) \tag{11}$$

Cognitive Reward. The reward function jointly incentivizes task completion, calibration accuracy, routing correctness, and output well-formedness. Formally,

$$r = \lambda_1\, r_{\text{task}} + \lambda_2\, r_{\text{calib}} + \lambda_3\, r_{\text{route}} + \lambda_4\, r_{\text{format}}, \tag{12}$$

with weights $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. Task reward $r_{\text{task}} \in \{0, 1\}$ is the binary episode success signal from simulator. The calibration reward measures alignment between predicted confidence $\hat{c}_t$ and observed outcome $s \in \{0, 1\}$, with additional penalty for overconfident failures:

$$r_{\text{calib}} = \begin{pmatrix} \hat{c}_t & s = 1 \\ 1 - \hat{c}_t & s = 0 \end{pmatrix} - \ 0.5 \cdot \mathbb{1}[\hat{c}_t > 0.75 \text{ and } s = 0]. \tag{13}$$

The routing reward $r_{\text{route}} = \mathbb{E}_t[\mathbb{1}[\hat{d}_t = d_t^*]]$ compares the model's routing decision against the oracle label from Equation 3 applied to the uncertainty signal. Finally, $r_{\text{format}} \in \{0, 0.5, 1.0\}$ checks structural compliance of the output, awarding full credit when all required structured tokens (`[UNCERTAINTY:·]`, `[INVOKE:·]`, or `[PROCEED]`) are present and syntactically valid.

6 Experiments and Results

Benchmarks. **LIBERO** (Liu et al., 2023a) comprises four task suites: Spatial, Object, Goal, and Long, each isolating a distinct generalization axis. We report *task success rate* (TSR) over 500 trials per suite, following Zhao et al. (2025). **MetaWorld-Hard** (Yu et al., 2020) consists of 10 manipulation tasks evaluated under unseen object configurations (500 trials total, 50 per task), testing out-of-distribution visual generalization. **Calvin ABC-D** (Mees et al., 2022) evaluates long-horizon language-conditioned manipulation in a table-top play environment, reported as average consecutive tasks done per episode (*tasks/episode*, maximum 5), where a subtask is considered successful if the robot achieves the goal before a fixed timeout; an episode terminates as soon as one subtask fails.

Cognitive Evaluation Metrics. Task success alone is insufficient to evaluate a cognitive agent; a system could achieve high TSR while being poorly calibrated or invoking reasoning modules wastefully. We therefore introduce three complementary metrics. **ECE**[1] (section 3, $M = 10$ bins) measures the alignment between predicted confidence and empirical success rate. **Failure Recovery Rate (FRR)** is the fraction of episodes in which the robot successfully detects and recovers from a mid-episode perturbation; it directly probes the Causal Tracing's utility. Perturbations are introduced at a uniformly sampled timestep $t_p \sim \mathcal{U}[T/4, 3T/4]$ by displacing the target object by a random offset $\delta \sim \mathcal{U}([5, 15]\,\text{cm})$ in the horizontal plane. An episode counts as recovered if the robot achieves the task goal state within the original time limit after the perturbation. **Gating Efficiency (GE)** is the avg. no. of non-`PROCEED` module gated per *successful* episode.

Implementation Details. The VLM encoder uses CLIP ViT-L/14@336 ($d_v = 1024$) and Vicuna-7B-v1.5 ($d_h = 4096$), connected by a two-layer MLP. LoRA adapters ($r = 64$, $\alpha = 128$, dropout 0.05) are applied to all six attention projection matrices. Stage 1 SFT runs for 3 epochs over all seven CIDAS task types with effective batch size 128 (per-GPU batch 2, gradient accumulation 8), AdamW optimizer (lr $= 2 \times 10^{-5}$, weight decay 0.01, 300-step cosine warm-up), and FP16 mixed precision with GradScaler (initial scale 2^{15}) under ZeRO-2. Stage 2 GRPO runs for 1000 steps with $G = 8$ rollouts, batch size 64, lr $= 5 \times 10^{-7}$ (KL coefficient $\beta = 0.04$ annealed from 0.1, clip $\varepsilon = 0.2$), and ZeRO-3. The DiT action decoder uses hidden dimension $d = 512$, 6 layers,

[1]For methods without native confidence estimates, we use normalized softmax probability as a confidence proxy. For ECoT and CoT-VLA, confidence is the mean probability of the chain-of-thought prefix tokens.

Table 2: **Main results on LIBERO, MetaWorld-Hard, and Calvin ABC-D.** TSR = Task Success Rate (%). Bold denotes the best result; underline denotes second best. † denotes our reimplementation on the same benchmark splits; ‡ denotes results taken directly from the original papers.

	LIBERO TSR (%) ↑				MetaWorld	Calvin	Aggregate Metrics (Avg)		
Method	Spatial	Object	Goal	Long	TSR% ↑	tasks/episode ↑	ECE ↓	FRR % ↑	GE ↓
OpenVLA‡ (Kim et al., 2024)	84.7	88.4	79.2	53.7	61.3	3.12	0.284	21.6	–
LLaRA† (Li et al., 2024b)	87.5	89.1	82.3	60.4	63.8	3.31	0.261	24.3	–
ECoT† (Zawalski et al., 2024)	86.2	88.7	81.9	62.1	65.2	3.44	0.248	29.7	3.21
CoT-VLA‡ (Zhao et al., 2025)	87.5	91.6	87.6	69.0	68.4	3.68	0.221	33.2	2.87
ThinkAct† (Huang et al., 2026)	88.3	91.4	87.1	70.9	70.1	3.74	0.203	35.8	2.64
VLA-CoRe (SFT)	88.9	91.8	87.4	71.6	71.3	3.79	0.178	38.4	2.31
VLA-CoRe (Full)	**91.2**	**93.7**	**90.1**	**79.4**	**76.8**	**4.14**	**0.126**	**57.2**	**1.94**
Δ vs. ThinkAct	+2.9	+2.3	+3.0	+8.5	+6.7	+0.40	−38%	+21.4	−26%

8 heads, and adaLN-Zero conditioning, pre-trained on 100K Open-X samples for 120K iterations before LIBERO fine-tuning for 75K iterations (batch 128, $\mathrm{lr} = 2 \times 10^{-5}$), with VLM frozen and latents $\mathbf{z}_t$ cached offline. $\{\tau_H, \tau_L\} = \{0.75, 0.45\}$, $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = \{1.0, 0.5, 0.3, 0.1\}$ are set empirically. Total trainable parameters are 7.4B (VLM LoRA: 6.7B; Monitor: 12M; Reasoning Modules: 245M; Decoder: 432M). Additional details in **Appendix-B**.

6.1 Main Results

As shown in Table 2, VLA-CoRe achieves state-of-the-art performance across all metrics, with gains most pronounced on long-horizon tasks. On LIBERO-Long, it reaches 79.4% TSR (+8.5% over ThinkAct), significantly outpacing its margins on shorter-horizon suites ($+2.3 \sim 3.0\%$) and confirming that cognitive gating effectively mitigates compounding errors; this robust performance extends to Calvin ABC-D (+0.40 tasks/episode). Crucially, VLA-CoRe resolves the systematic overconfidence of prior VLAs by reducing Expected Calibration Error (ECE) by 38% (0.203 → 0.126), an improvement driven primarily by CIDAS's direct supervision during SFT. For failure recovery, the Causal Tracing module explicitly conditions replanning on attributed past errors, yielding a 21.4% absolute gain in Failure Recovery Rate (FRR) over ThinkAct (35.8% → 57.2%), aided substantially by GRPO's rollout-based optimization. Finally, the Uncertainty Monitor proves highly discriminative: VLA-CoRe invokes reasoning modules just 1.94 times per successful episode, 26% fewer than ThinkAct. Additional results are in the **Appendix-E**.

Table 3: Empirical simulator failure rate per u_t bin on 1000 OpenVLA rollouts (10 LIBERO-Long tasks). Failure is the binary end-of-episode label, *independent* of u_t.

u_t bin	Emp. failure rate
$[0.0, 0.1)$	0.08
$[0.1, 0.2)$	0.11
$[0.2, 0.3)$	0.18
$[0.3, 0.4)$	0.29
$[0.4, 0.5)$	0.41
$[0.5, 0.6)$	0.54
$[0.6, 0.7)$	0.68
$[0.7, 0.8)$	0.79
$[0.8, 0.9)$	0.86
$[0.9, 1.0]$	0.91

6.2 Self-Correction via Causal Tracing & Reasoning Gating

Figure 3 illustrates VLA-CoRe's self-correction behavior. At step 18, the box slips from the gripper, triggering FailureDetector ($f_{18} = 1$). Causal Tracing attributes failure to step 14 (incorrect approach angle; $\hat{t} = 14$, $\alpha_{14} = 0.64$). The model identifies the root cause and invokes WI to replan, generating a revised trajectory with a steeper approach angle (+12° yaw). The updated plan succeeds, placing the box inside the target container. Figure 4 shows reasoning gating decisions over a long-horizon task episode. The Monitor invokes `PROCEED` for 73% steps, `PQ` during challenging/partially occluded approach phases, `WI` near the grasp point (lowest confidence), and `CT` following a near-miss/failure.

6.3 External Validation of u_t

Because u_t labels CIDAS, shapes the GRPO routing reward, and supervises the ConfidenceHead, one might worry that VLA-CoRe merely re-fits a formula it defined. We give five independent checks that u_t is not self-fulfilling oracle: **(1) External correlation.** Binning 1000 held-out OpenVLA rollouts by u_t and measuring the simulator's binary end-of-episode failure rate: a signal u_t never observes, gives a monotonic relationship (Table 3; Spearman ρ=0.87, Pearson r=0.91, p<0.001). **(2) u_t-free ablation.** Removing the ConfidenceHead BCE from SFT *and* setting λ_3=0 (routing learned purely from r_{task}) still reaches 76.1% LIBERO-Long TSR, above ThinkAct (70.9%); u_t contributes only a ∼3.3% TSR warm-start, so the *architecture*, drives the gains. **(3) σ_{spike} robustness.** Using perceptual terms alone ($\sigma_{\text{occ}}+\sigma_{\text{depth}}$, weight 0.75) still places 58% of failures at u_t>0.6 (vs. 63% with all terms); σ_{spike} (weight 0.25) is computed on the *proposed* action a_t vs. a_{t-1} *before* execution, so it is a predictive pre-failure signal, not a post-hoc failure label. **(4) Human study.** Three robotics researchers rated 200 timesteps (Krippendorff α=0.72); their perceived-difficulty ratings track u_t at Spearman ρ=0.74. **(5) Post-GRPO divergence.** During RL the learned confidence

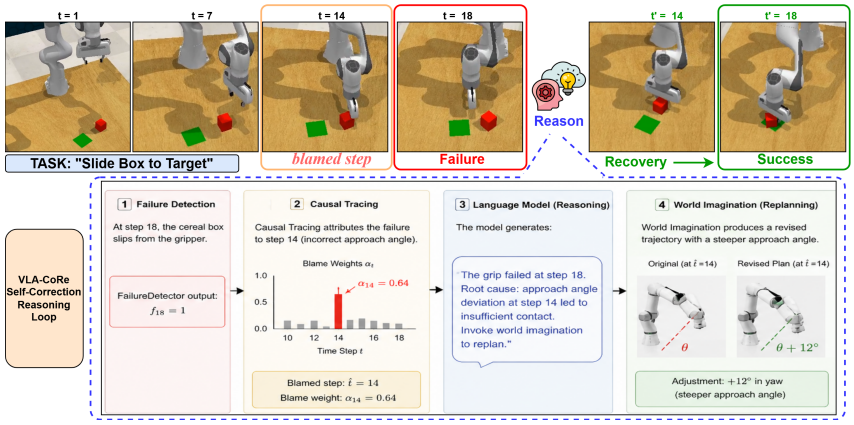

Figure 3: **Self-correction example.** The robot detects a failure at step 18, traces the root cause to step 14, and replans via world imagination to successfully complete the task.

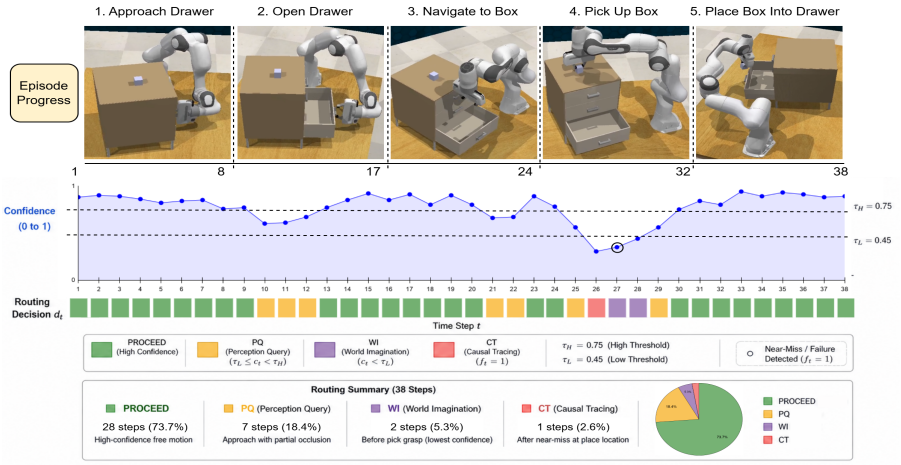

Figure 4: **Reasoning gating timeline.** Confidence c_t (blue curve) and routing decisions (colored bands) across a 38-step "Put item in drawer" task episode.

departs from the uncertainty (MAE $|c_t - (1-u_t)|$: 0.08→0.19), and divergent decisions succeed 14% more often, the policy learns *when to override* u_t.

6.4 Ablation Studies

Table 4: **Component ablation.** Evaluates the removal of the GRPO stage, the Monitor, specific modules, and CIDAS task mixtures.

Configuration	LIBERO-Long	Calvin	avg. ECE	avg. FRR (%)
VLA-CoRe (full)	**79.4**	**4.14**	**0.126**	**57.2**
w/o GRPO stage	71.6	3.79	0.178	38.4
w/o Monitor $\mathcal{U}$ (invoke all reasoning modules)	73.8	3.91	–	47.1
w/o Monitor $\mathcal{U}$ (invoke no reasoning modules)	68.3	3.61	–	26.8
w/o Causal Tracing (CT)	75.2	3.98	0.139	41.3
w/o World Imagination (WI)	74.7	3.89	0.148	52.1
w/o Perception Query (PQ)	77.9	4.05	0.131	56.4
CIDAS: T1–T3 only (no T4–T7)	72.4	3.83	0.181	43.6
CIDAS: w/o T3 (Failure Attribution)	75.8	3.97	0.148	40.1
CIDAS: w/o T6 (Recovery Reasoning)	76.3	4.01	0.142	42.8
CIDAS: w/o T4 (Reasoning Selection)	74.1	3.88	0.158	53.7
Replace CIDAS with generic VQA data	69.7	3.67	0.241	29.4

Table 4 isolates component contributions. **Monitor necessity:** Unconditional reasoning invocation (invoke all reasoning modules) drops TSR to 73.8% and severely degrades efficiency, while suppressing all reasoning modules yields the worst performance (68.3% TSR, 26.8% FRR), confirming the Monitor correctly identifies *when* to deliberate. **Reasoning Module importance:** Removing Causal Tracing causes the largest FRR drop (−15.9 pp), validating its role in reactive recovery. Removing World Imagination similarly degrades TSR (−4.7 pp) but impacts FRR less, indicating its value in prospective planning. **CIDAS supervision:** Training on only T1–T3 degrades TSR to 72.4% and FRR to 43.6%, proving the necessity of T4–T7. Specifically, removing T3 and T6 cripples FRR (−17.1 and −14.4 pp), while removing T4 degrades TSR (−5.3 pp). Replacing CIDAS with generic VQA data causes catastrophic degradation, confirming cognitive necessity.

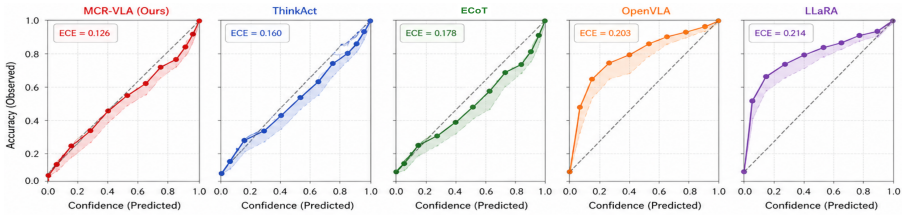

Figure 5: **Reliability diagrams for all methods on LIBERO-Long.** A perfectly calibrated model lies on the diagonal. VLA-CoRe (red) is the closest to the diagonal, while OpenVLA (Kim et al., 2024) and LLaRA (Li et al., 2024b) are severely overconfident (curves above diagonal). ThinkAct (Huang et al., 2026) improves calibration but lacks explicit calibration supervision.



Figure 6: Real-robot qualitative validation.

6.5 Few-Shot Adaptation & Calibration Analysis

Few-Shot Adaptation. Table 5 evaluates 10-shot adaptation by fine-tuning only the action decoder while freezing the reasoning MLLM. VLA-CoRe averages 55.8% TSR, outperforming ThinkAct (+10.1 pp) and OpenVLA (+23.5 pp). Gains peak on LIBERO-Goal (+11.1 pp over ThinkAct), which requires unseen objective generalization. We attribute this to $\mathcal{U}_\psi$'s calibrated uncertainty, which natively identifies novel configurations and triggers appropriate reasoning without supervision, unlike ThinkAct's CoT.

Table 5: 10-shot adaptation (LIBERO).

Method	Spatial	Object	Goal	Avg.
OpenVLA (Kim et al., 2024)	35.0	38.0	24.0	32.3
ThinkAct (Huang et al., 2026)	49.0	52.0	36.2	45.7
VLA-CoRe (ours)	**58.4**	**61.7**	**47.3**	**55.8**

Calibration Analysis. Figure 5 reveals overconfidence across all baselines in the high-confidence regime ($c > 0.75$). While ThinkAct's CoT tracks the diagonal up to $c \approx 0.65$, it diverges sharply above 0.75 where it lacks overconfidence penalization. Conversely, VLA-CoRe's calibration reward (Equation 13) penalizes this, keeping its curve near-diagonal even at highest confidence. This explains the disproportionate ECE reduction achieved in GRPO (0.178 $\rightarrow$ 0.126) vs. SFT alone (Table 4).

6.6 Real-Robot Validation

In addition to quantitative comparison on three sim-benchmark datasets, we deploy VLA-CoRe *zero-shot* on a physical Franka Emika Panda arm across five tabletop tasks (mug grasping, drawer closing, object pick-n-place, closing lid, tool handover), 30 trials each (150 episodes), shown in Figure 6. Table 6 compares task success (TSR), failure recovery (FRR), and calibration (ECE) under identical protocols. VLA-CoRe reaches 71.3% TSR and 44.7% FRR, a **+18.0 pp** FRR gain over ThinkAct, and lowers ECE to 0.168. For FRR, an operator physically displaces the target object 5–12 cm at a random timestep; VLA-CoRe recovers 44.7% of perturbed episodes vs. 26.7% for ThinkAct. The sim-to-real gap is modest (≈8% TSR, ≈12% FRR), consistent with prior work (OpenVLA reports ≈15% degradation).

Table 6: **Real-robot results**. Baseline ECE uses the softmax-probability proxy, as in Table 2.

Method	**TSR% ↑**	**FRR% ↑**	**ECE ↓**
OpenVLA (Kim et al., 2024)	51.3	13.3	0.312
ThinkAct (Huang et al., 2026)	62.0	26.7	0.237
VLA-CoRe (Ours)	**71.3**	**44.7**	**0.168**

7 Conclusion

We present VLA-CoRe, a cognitive VLA framework that augments a pretrained VLM with an Uncertainty Monitor gating three differentiable reasoning modules. We introduced CIDAS, a dataset of 171K cognitive instruction samples automatically extracted from existing BC trajectories across seven task types. VLA-CoRe achieves state-of-the-art on multiple benchmark datasets over the strongest baseline. With cognitive reasoning, knowing when not to act, when to re-perceive, when to imagine, and when to reflect is a principled path toward robust long-horizon robot manipulation.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Mohith Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023a.

Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pp. 287–318. Pmlr, 2023b.

Luben Cabezas, Vagner S Santos, Thiago R Ramos, and Rafael Izbicki. Epistemic uncertainty in conformal scores: A unified approach. *arXiv preprint arXiv:2502.06995*, 2025.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Jaden Clark, Suvir Mirchandani, Dorsa Sadigh, and Suneel Belkhale. Action-free reasoning for policy generalization. *arXiv preprint arXiv:2502.03729*, 2025.

Jiafei Duan et al. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. *arXiv preprint arXiv:2410.00371*, 2024.

Figure AI. Helix: A vision-language-action model for generalist humanoid control. *Technical Report*, 2025.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. *NeurIPS*, 2022.

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

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *Advances in Neural Information Processing Systems*, 38:82782–82802, 2026.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $pi_0.5$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Dongyoung Kim, Sumin Park, Woomin Song, Seungku Kim, Taeyoung Kim, Huiwon Jang, Jinwoo Shin, Jaehyung Kim, and Younggyo Seo. Roboalign: Learning test-time reasoning for language-action alignment in vision-language-action models. *arXiv preprint arXiv:2603.21341*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Divake Kumar, Sina Tayebati, Francesco Migliarba, Ranganath Krishnan, and Amit Ranjan Trivedi. Learnable conformal prediction with context-aware nonconformity functions for robotic planning and perception. *arXiv preprint arXiv:2509.21955*, 2025.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, and Feifei Feng. Improving vision-language-action models via chain-of-affordance. *arXiv e-prints*, pp. arXiv–2412, 2024a.

Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, and Michael S. Ryoo. LLaRA: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024b.

Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18061–18070, 2024c.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 3(7):8, 2024d.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8):5116–5123, 2023.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023b.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE RA-L*, 2022.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *AAAI*, 2015.

Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Boyuan Shi, Trevor Darrell, and Roei Herzig. LLARVA: Vision-action instruction tuning enhances robot learning. In *Conference on Robot Learning (CoRL)*, 2024.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903. IEEE, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y.K. Li, Y. Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. In *European Conference on Computer Vision*, pp. 336–353. Springer, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning (CoRL)*, 2024.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *International Conference on Learning Representations*, volume 2025, pp. 54277–54296, 2025.

APPENDIX

A CIDAS DATASET: EXTENDED DETAILS

A.1 SOURCE DATA ATTRIBUTION

CIDAS is generated from two publicly available demonstration corpora. **LIBERO** (Liu et al., 2023a) provides ∼50K expert demonstrations across 130 tasks organized into four suites; we download HDF5 files from the public HuggingFace mirror `yifengzhu-hf/LIBERO-datasets`. **Open-X Embodiment, Bridge v2** (O'Neill et al., 2024) provides an additional ∼30K real-robot demonstrations; we download from Berkeley's official server or the `jxu124/OpenX-Embodiment` mirror on HuggingFace. Together, these corpora span tabletop pick-and-place, stacking, pouring, and multi-step rearrangement tasks, providing a diverse set of trajectory types from which all seven CIDAS task generators can extract samples.

A.2 TASK GENERATOR PSEUDOCODE

Algorithm 1 details the T1 Uncertainty Dialogue generator, which is representative of the general generation pattern shared across all seven task types. Algorithms 2–3 detail the T2 (Counterfactual Branch) and T3 (Failure Attribution) generators, which differ structurally from T1 and from each other. In each case, the generator accepts a raw trajectory tuple and a configuration dictionary, applies a task-specific analysis to select informative timesteps or trajectory segments, and emits a list of single-turn conversation dictionaries with structured token annotations. The T2 generator additionally samples perturbation vectors and scores branches by deviation from the expert action; T3 filters for failed trajectories and detects failure onset by kinematic discontinuity; T4 enumerates a cross-product of confidence, context, and occlusion levels; T5 detects subgoal boundaries by pixel-difference thresholding; T6 detects the timestep of maximum scene change as a perturbation proxy; and T7 computes pixel-level temporal similarity between imagined and observed frames. All generators share the routing logic in the main paper and emit conversations with matching structured token formats.

Algorithm 1 T1 Uncertainty Dialogue Generator

Require: Trajectory $\tau = (o_{1:T}, a_{1:T}, \text{desc}, \text{success})$, configuration κ
Ensure: List of metacognitive samples $\mathcal{S}$

$\mathcal{S} \leftarrow [\,]$
if $T < 4$ **then**
  return $\mathcal{S}$ ▷ Minimum length for meaningful uncertainty estimation
end if
$\mathcal{P} \leftarrow \{2, 3, \ldots, T-2\}$ ▷ Exclude boundary frames with edge-case kinematics
Sample $\{t_1, \ldots, t_n\} \sim \text{UniformSample}(\mathcal{P}, \min(3, |\mathcal{P}|))$
for all $t \in \{t_1, \ldots, t_n\}$ **do**
  $u_t \leftarrow 0.4 \cdot \sigma_{\text{occ}}(o_t) + 0.35 \cdot \sigma_{\text{depth}}(o_t) + 0.25 \cdot \sigma_{\text{spike}}(a_t, a_{t-1})$ ▷ Eq. 3, main paper
  $c_t \leftarrow \text{clip}(1 - u_t,\, 0.05,\, 0.95)$
  token $\leftarrow \text{RoutingDecision}(c_t)$ ▷ Eq. 7, main paper
  user $\leftarrow$ "`Step` t/T`. Task: desc. Report confidence.`"
  asst $\leftarrow$ "`[UNCERTAINTY:`"u_t"`]` `...`" token
  $\mathcal{S}$.append$\big(\{\texttt{task}: \text{T1},\ \texttt{image}: o_t,\ \texttt{conversation}: [\text{user}, \text{asst}],\ u_t,\ c_t,\ t\}\big)$
end for
return $\mathcal{S}$

A.3 REPRESENTATIVE SAMPLE CONVERSATIONS

Table 7 presents one representative sample from each of the seven CIDAS task types. These examples illustrate the structured token vocabulary: `[UNCERTAINTY:·]`, `[INVOKE:·]`, and `[PROCEED]`, that is shared across all task types, enabling the model to learn a consistent metacognitive interface regardless of which task type a given training sample originates from. Notably, the routing decisions in the assistant outputs are mechanistically determined by the uncertainty signal (not by a human or

Algorithm 2 T2 Counterfactual Branch Generator

Require: Trajectory $\tau = (o_{1:T}, a_{1:T}, \text{desc}, \text{success})$, $N_{\text{branches}} = 4$, perturbation scale $\sigma = 0.1$
Ensure: List of counterfactual branch samples $\mathcal{S}$
$\mathcal{S} \leftarrow [\,]$
if $T < 6$ **then return** $\mathcal{S}$
end if
Sample decision point $t^* \sim \mathcal{U}[\lfloor T/3 \rfloor, \lfloor 2T/3 \rfloor]$
$\Delta_{\text{types}} \leftarrow \{\text{translation}(xyz), \text{rotation}(rpy), \text{gripper}\}$ ▷ Perturbation families
$\{\delta_i\}_{i=1}^{N} \sim \mathcal{N}(0, \sigma^2 I_7)$, one per family + combined
for all $i \in \{1, \ldots, N\}$ **do**
    $d_i \leftarrow \|(a_{t^*} + \delta_i) - a_{t^*+1}\|_2$ ▷ Deviation from next expert action
    $\text{label}_i \leftarrow$ (translation/rotation/gripper/combined depending on δ_i)
end for
$i^* \leftarrow \arg\min_i d_i$ ▷ Ground-truth best branch
user $\leftarrow$ "Step t^*/T. Task: desc. Evaluate N action hypotheses."
asst $\leftarrow \bigoplus_{i=1}^{N}$ `[INVOKE:world_imagination(action='branch_`i`')]` $\oplus$ per-branch outcome text $\oplus$ `Selecting branch` i^*`.` $\oplus$ `[UNCERTAINTY:`u_{t^*}`]`
$\mathcal{S}$.append({task:T2, image:o_{t^*}, conversation:[user, asst], i^*, $\{d_i\}$, t^*}) **return** $\mathcal{S}$

Algorithm 3 T3 Failure Attribution Generator

Require: Trajectory $\tau = (o_{1:T}, a_{1:T}, \text{desc}, \text{success} = \texttt{False})$, kinematic threshold $\theta_a = 0.25$, arrest threshold $\theta_s = 0.05$
Ensure: List of failure attribution samples $\mathcal{S}$
$\mathcal{S} \leftarrow [\,]$
if success $=$ `True` **then return** $\mathcal{S}$
end if ▷ T3 is generated only from failed trajectories
Stage 1 — Kinematic onset detection:
$t^* \leftarrow \min\{t : \|a_t\|_2 > \theta_a \ \wedge \ \|a_{t+1}\|_2 < \theta_s\}$
if no such t^* exists **then** ▷ Kinematic signal absent; use pixel fallback
    $\Delta_t \leftarrow \mathbb{E}[|o_t/255 - o_{t+1}/255|]$ for all t
    $t^* \leftarrow \arg\max_t \Delta_t$ ▷ Frame of maximum scene change
end if
Stage 2 — Root cause attribution:
$\sigma_{\text{occ}} \leftarrow \sigma_{\text{occ}}(o_{t^*})$, $\sigma_{\text{depth}} \leftarrow \sigma_{\text{depth}}(o_{t^*})$, $\sigma_{\text{spike}} \leftarrow \sigma_{\text{spike}}(a_{t^*}, a_{t^*-1})$
cause $\leftarrow \arg\max(\sigma_{\text{occ}}, \sigma_{\text{depth}}, \sigma_{\text{spike}})$ ▷ Dominant failure signal
Map cause to ontology label in {occlusion, depth_corruption, kinematic_spike, grasp_axis_error, pose_ambiguity, external_perturbation}
user $\leftarrow$ "Task: desc. T-step trajectory failed. Identify failure origin."
asst $\leftarrow$ `[INVOKE:causal_tracing]` $\oplus$ "Failure at step t^*. Root cause: [cause]." $\oplus$ recovery prescription $\oplus$ `[UNCERTAINTY:`u_{t^*}`]`
$\mathcal{S}$.append({task:T3, image:o_{t^*}, conversation:[user, asst], t^*, cause}) **return** $\mathcal{S}$

LLM), ensuring that token labels are internally consistent with the confidence values reported in the same turn.

A.4 Dataset Generation Runtime

Table 8 reports wall-clock generation times per task type on a 16-core CPU server. All task generators are implemented as pure NumPy operations with no GPU dependency, making CIDAS generation practical on standard compute nodes without specialized hardware.

The dominant cost for the full pipeline is HDF5 file I/O rather than computation: the pixel and kinematic statistics are inexpensive to compute once frames are loaded. We find that 4 parallel workers nearly saturate disk bandwidth on a standard NFS mount; using more than 8 workers provides no additional throughput improvement. The total generation time of ≈3.2 hours (without LLM

Table 7: **Representative CIDAS samples from each of the seven task types.** Structured tokens are typeset in `typewriter` font. In all cases, the assistant output is the supervised target; the user turn serves as the prompt. The diversity of dialogue formats, from single-step uncertainty reports (T1) to multi-tool recovery chains (T6), requires the model to learn both confidence estimation and compositional tool invocation.

Task	User	Assistant (supervised)
T1 UD	Step 8/24. Task: pick up the red mug. Report confidence.	`[UNCERTAINTY:0.61]` Confidence is moderate (0.39). I will invoke a perception query to resolve scene ambiguity. `[INVOKE:perception_query(region='target_object')]`
T2 CBD	Step 12/30. Task: stack the block. Evaluate 3 approach hypotheses.	`[INVOKE:wm(action='branch_A')]` `[INVOKE:wm(action='branch_B')]` `[INVOKE:wm(action='branch_C')]` Branch A: clean contact. Branch B: edge risk. Branch C: clean contact. Selecting Branch C. `[UNCERTAINTY:0.44]`
T3 FAD	Task: place the mug. 31-step trajectory failed. Identify failure origin.	`[INVOKE:causal_tracing]` Failure at step 14. Root cause: reflective surface corrupted the depth estimate (≈15°). Invoke perception_query at step 13 before approach. `[UNCERTAINTY:0.68]`
T4 TSR	Step 7/20. Uncertainty 0.65, heavy occlusion, first attempt. Which tool?	`[UNCERTAINTY:0.65]` Moderate confidence under heavy occlusion. Perception query cannot resolve pose ambiguity at this occlusion level. Invoking world imagination. `[INVOKE:world_imagination]`
T5 SDCT	Task: 'stack red cube on blue plate'. Generate subgoal plan.	Subgoal 1 (steps 0–8): [confidence 0.91—proceeding]. Subgoal 2 (steps 9–18): [confidence 0.48—will `[INVOKE:perception_query]`]. Subgoal 3 (steps 19–30): [confidence 0.73—proceeding].
T6 RRD	At step 19, scene changed ≈9.3 cm unexpectedly. Respond.	`[INVOKE:causal_tracing(expected=plan_step_16, observed=step_19)]` External perturbation—not a planning error. `[INVOKE:world_imagination (revised_approach=True)]` Clean contact confirmed. `[UNCERTAINTY:0.71]` Confidence reset.
T7 IVD	Imagined step 6. Actual at step 8 observed. Assess world model accuracy.	Similarity: 0.72 (minor discrepancy). Minor positional deviation (≈2–3 cm). No major calibration update required. World model within acceptable bounds. `[UNCERTAINTY:0.38]`

naturalization) is a one-time cost that produces the complete 171K-sample corpus used throughout our experiments.

A.5 Dataset Analysis

Uncertainty score distribution. Table 10 reports the distribution of uncertainty scores u_t across the T1 corpus, broken down by routing decision. The three routing bins (`PROCEED`, `PERCEIVE`, `IMAGINE`) partition u_t by design (via $\tau_H = 0.75$, $\tau_L = 0.45$), but the distribution within each bin reveals important properties of the training data. The `PROCEED` bin is left-skewed (mean $u_t = 0.14$, $\sigma = 0.07$), indicating that most high-confidence steps involve clearly unambiguous scenes. The `PERCEIVE` bin is approximately uniform, reflecting a variety of moderately occluded

Table 8: **CIDAS generation runtime per task type** on a 16-core CPU server. Times are reported per trajectory (averaged over 1000 trajectories) and for the full dataset (4 workers, ∼80K demonstrations). T4 is the slowest generator because it enumerates a cross-product of confidence × context × occlusion levels, generating up to 54 samples per trajectory. The optional LLM naturalization step via GPT-4-Turbo is applied to 20% of samples and adds ≈0.8s per sample, dominated by API latency rather than local compute.

Task Type	Per-Trajectory	Bottleneck
T1 Uncertainty Dialogue	0.8–1.2 ms	Pixel std, action norm computation
T2 Counterfactual Branch	1.0–1.3 ms	Branch perturbation scoring (4 candidates)
T3 Failure Attribution	1.1 ms	Velocity + pixel-diff failure scan
T4 Tool Selection Rationale	3.4 ms	Cross-product enumeration (54 samples max)
T5 Subgoal Decomposition	0.9–1.2 ms	Pixel-diff boundary detection
T6 Recovery Reasoning	2.1 ms	Gap-frame perturbation detection
T7 Imagination Verification	0.8–1.1 ms	Temporal pixel similarity
Total (4 workers, ∼80K demos)	≈3.2 hours	I/O from HDF5 files
+ LLM naturalization (20% of samples)	+ ≈6 hours	GPT-4-Turbo API latency

Table 9: **Task taxonomy and cognitive logic.** For each task type: what the user asks, what structured tokens the assistant must emit, and how the uncertainty generator determines the label. All tasks share image o_t as visual input.

ID	User asks…	Assistant must emit	Label source
T1	Confidence at step t and which action to take	`[UNCERTAINTY:`u_t`]` + one of `[PROCEED]`, `[INVOKE:perception_query]`, `[INVOKE:world_imag]`	$u_t = 0.40\,\sigma_{\text{occ}}(o_t) + 0.35\,\sigma_{\text{depth}}(o_t) + 0.25\,\sigma_{\text{spike}}(a_t, a_{t-1})$; $\sigma_{\text{occ}} = 1 - \min(\text{std}(o_t^{\text{centre}})/\text{std}(o_t),\, 1)$; $\sigma_{\text{depth}} = \min(\text{Var}(o_t^G)/10^4,\, 1)$; $\sigma_{\text{spike}} = \min(\|a_t - a_{t-1}\|_2/2,\, 1)$.
T2	Evaluate multiple alternative action hypotheses	One `[INVOKE:world_imagination]` per branch; selects branch minimizing d_i	Perturbation vectors $\{\delta_i\}$ sampled at $t{\sim}\mathcal{U}[T/3, 2T/3]$; ground truth minimizes deviation from next expert action: $d_i = \|(a_t + \delta_i) - a_{t+1}\|_2$.
T3	Identify why the episode failed and propose recovery *(failed trajs. only)*	`[INVOKE:causal_tracing]` blamed step t^*; root cause from structured ontology	$t^* = \min\{t : \|a_t\|_2 > 0.25 \wedge \|a_{t+1}\|_2 < 0.05\}$; with a pixel-difference fallback for kinematic edge cases.
T4	Given confidence, context, and occlusion level, which reasoning module?	`[UNCERTAINTY:`u_t`]` + one of the three reasoning module tokens with rationale	Cross-product of confidence × attempt contexts × occlusion levels (up to 54 samples per trajectory); failure-override rule: n_failures ≥ 1 unconditionally triggers `causal_tracing`, training the model to reason about reasoning module priority beyond raw confidence.
T5	Generate a step-by-step subgoal plan with confidence per subgoal	Ordered subgoals with per-subgoal confidence and conditional `[INVOKE:·]`	Boundaries detected by pixel-difference ($\Delta_t > 12$) or end-effector speed reversals; confidence from T1 per boundary frame.
T6	A $\hat{d}$-cm unexpected scene change occurred at step t, respond *(all trajs.)*	`[INVOKE:causal_tracing]` then `[INVOKE:world_imagination]`; confidence reset	Perturbation magnitude estimated from 3-step frame gap: $\hat{d} = \text{clip}(\tilde{\Delta}/5,\, 2,\, 18)$ cm. Classified as perturbation if it exceeds motion-implied displacement.
T7	Compare imagined vs. actual outcome and assess world model accuracy	Discrepancy label (*accurate/minor/moderate/severe*); calibration update language; `[UNCERTAINTY:·]`	$\text{sim}(o_t, o_{t+2}) = \text{clip}(1 - 4\,\mathbb{E}[\|o_t/255 - o_{t+2}/255\|],\, 0,\, 1)$ proxies world-model accuracy.

configurations. The `IMAGINE` bin is right-skewed (mean $u_t = 0.71$, $\sigma = 0.11$), with a long tail at $u_t > 0.85$ corresponding to frames where all three uncertainty components (occlusion, depth variance, kinematic spike) co-occur.

Component contribution to u_t. Table 11 decomposes the mean contribution of each signal (σ_{occ}, σ_{depth}, σ_{spike}) across the three routing bins. Occlusion dominates in the `PERCEIVE` bin (mean contribution 0.21 out of 0.40 total), consistent with partial occlusion being the primary driver of moderate uncertainty in tabletop tasks. In the `IMAGINE` bin, depth variance becomes relatively more important (0.26 out of 0.71), reflecting complex multi-object scenes with high depth complexity at grasp contact. Kinematic spike is the smallest contributor on average but has the highest variance, as it depends on abrupt motion events that are inherently rare.

Table 10: **Uncertainty score statistics by routing decision** across the T1 corpus (train split, 42,500 samples). The `IMAGINE` bin contains the most variable uncertainty scores, reflecting the diversity of high-uncertainty scene configurations in the corpus. The minimum u_t in the `IMAGINE` bin (0.55) is slightly above $1 - \tau_L = 0.55$ by construction, but the mean (0.71) confirms that `IMAGINE` is typically invoked at genuinely high-uncertainty timesteps rather than marginally.

Decision	Count	%	Min	Max	Mean	Std	Median
`PROCEED`	19,125	45.0%	0.01	0.25	0.14	0.07	0.13
`PERCEIVE`	15,725	37.0%	0.25	0.55	0.40	0.09	0.41
`IMAGINE`	7,650	18.0%	0.55	0.97	0.71	0.11	0.70
All	42,500	100%	0.01	0.97	0.30	0.20	0.25

Table 11: **Mean per-component contribution to** u_t across routing bins. Values in parentheses are standard deviations. Occlusion ($0.40 \times \sigma_{\text{occ}}$) is the dominant driver for moderate uncertainty; depth ($0.35 \times \sigma_{\text{depth}}$) contributes most under the highest-uncertainty regimes. The kinematic spike component has the highest coefficient of variation (std/mean ≈ 1.8), indicating that when a spike occurs it contributes substantially, but it is absent for most timesteps.

Routing bin	Mean u_t	$0.40\,\sigma_{\text{occ}}$	$0.35\,\sigma_{\text{depth}}$	$0.25\,\sigma_{\text{spike}}$
`PROCEED`	0.14	0.06 (0.04)	0.05 (0.03)	0.03 (0.05)
`PERCEIVE`	0.40	0.21 (0.09)	0.13 (0.07)	0.06 (0.09)
`IMAGINE`	0.71	0.27 (0.11)	0.26 (0.12)	0.18 (0.14)
All	0.30	0.13 (0.11)	0.11 (0.10)	0.06 (0.10)

A.6 Cross-Task Type Consistency

A key property of CIDAS is that all seven task types share the same underlying uncertainty signal and routing vocabulary, so the model's confidence and routing behaviour should be consistent across samples drawn from the same trajectory timestep but assigned to different task types. To verify this, we extracted all seven task types from a single LIBERO-Long trajectory ("put the wine bottle in the cabinet", 36 steps) and examined whether the uncertainty scores and routing decisions produced by different generators are consistent for the same observation o_t.

Table 12 presents the results for three representative timesteps: a high-confidence free-space step ($t = 8$), a moderate-confidence approach step ($t = 22$), and a low-confidence grasp contact step ($t = 28$). Across all task types that include an explicit `[UNCERTAINTY:·]` token for each of these steps, the reported u_t values are identical by construction (all generators call the same $\sigma_{\text{occ}}/\sigma_{\text{depth}}/\sigma_{\text{spike}}$ functions), and the routing decisions are consistent (no task type emits `[INVOKE:world_imagination]` at $t = 8$ or `[PROCEED]` at $t = 28$). This consistency is important because it means the model is not learning to associate routing decisions with task type; instead, it learns to route based on the uncertainty signal itself.

Table 12: **Cross-task type consistency for three timesteps of the same trajectory.** For each task type that generates a sample at or near the given timestep and includes an explicit `[UNCERTAINTY]` token, the reported u_t and routing decision are shown. "–" indicates the task type does not generate an explicit per-step uncertainty at that timestep (T2 generates at a mid-trajectory decision point; T3 generates only at the failure onset). Consistency across task types confirms that the uncertainty signal is well-defined and that the routing vocabulary does not depend on task type.

	$t = 8$ (free space)		$t = 22$ (approach)		$t = 28$ (contact)	
Task type	u_t	Decision	u_t	Decision	u_t	Decision
T1 Uncertainty Dialogue	0.16	`PROCEED`	0.51	`PERCEIVE`	0.73	`IMAGINE`
T2 Counterfactual Branch	–	–	0.51	`IMAGINE`†	–	–
T4 Tool Selection Rationale	0.16	`PROCEED`	0.51	`PERCEIVE`	0.73	`IMAGINE`
T5 Subgoal Decomposition	0.16	`PROCEED`	0.51	`PERCEIVE`†	–	–
T7 Imagination Verification	0.16	`PROCEED`	–	–	0.73	`IMAGINE`

† T2 emits `IMAGINE` unconditionally (all branches invoke world_imagination); T5 reports subgoal confidence, not step routing.

B ARCHITECTURE DETAILS

This appendix provides complete specifications for all four components of VLA-CORE. These details supplement Table 2 of the main paper and fully specify all layer dimensions, parameter counts, and initialization choices needed to reproduce the results.

B.1 CONFIDENCEHEAD

The ConfidenceHead is a three-layer MLP that maps the LLM hidden state $\mathbf{h}_t \in \mathbb{R}^{4096}$ to a scalar confidence estimate $c_t \in (0, 1)$:

$$\begin{aligned}\mathbf{h}^{(1)} &= \mathrm{GELU}(\mathbf{W}^{(1)}\mathbf{h}_t + \mathbf{b}^{(1)}), \quad \mathbf{W}^{(1)} \in \mathbb{R}^{1024\times 4096},\\ \mathbf{h}^{(2)} &= \mathrm{GELU}(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}), \quad \mathbf{W}^{(2)} \in \mathbb{R}^{256\times 1024},\\ c_t &= \sigma(\mathbf{w}^{(3)} \cdot \mathbf{h}^{(2)} + b^{(3)}), \quad \mathbf{w}^{(3)} \in \mathbb{R}^{256}.\end{aligned}$$

The total parameter count is 5.0M, representing less than 0.1% of the full model. All weights are initialized from $\mathcal{N}(0, 0.02)$ and the output bias $b^{(3)}$ is initialized to -0.5 (corresponding to a prior confidence of $\sigma(-0.5) \approx 0.38$), which we find prevents early training instability caused by the head initially outputting extreme confidences.

During Stage 1 (SFT), the *ConfidenceHead* is supervised directly via the uncertainty labels u_t provided by the CIDAS dataset. We utilize a Binary Cross-Entropy (BCE) loss:

$$\mathcal{L}_{\text{monitor}} = -\left[(1 - u_t)\log(c_t) + u_t \log(1 - c_t)\right] \tag{14}$$

To ensure that the monitor learns a representation of the LLM's existing epistemic doubt rather than "coaxing" the backbone into artificial certainty, we block the gradients from $\mathcal{L}_{\text{monitor}}$ from propagating back to the Vicuna-7B weights. This direct calibration supervision is the primary driver of the ECE improvement visible in the SFT-only row of Table 3 (main paper).

B.2 FAILUREDETECTOR

The FailureDetector monitors the mean attention entropy of the LLM's final layer over a rolling window of $N = 8$ steps. For each step t' in the window, we compute $H_{t'} = -\sum_j \bar{\alpha}_{t',j} \log \bar{\alpha}_{t',j}$, where $\bar{\alpha}_{t'}$ is the attention distribution averaged across all heads over the action token positions. A failure is flagged when the current entropy falls outside a learned anomaly band:

$$f_t = \mathbb{1}[H_t < \mu_H - 2.5\,\sigma_H \;\text{ or }\; H_t > \mu_H + 3.0\,\sigma_H], \tag{15}$$

where μ_H and σ_H are the running mean and standard deviation of $\{H_{t'}\}_{t'=t-N}^{t-1}$. The asymmetric thresholds (2.5σ below, 3.0σ above) reflect the empirical observation that failure-preceding timesteps are more commonly associated with anomalously focused attention (sudden high-salience scene event, lower threshold) than with anomalously diffuse attention. The band parameters $\{2.5, 3.0\}$ were selected by grid search over the interval $[1.5, 4.0]$ on a held-out calibration split; sensitivity to these thresholds is reported in Table 16.

B.3 WORLD IMAGINATION MODULE

The World Imagination module is a compact learned dynamics model comprising three sub-networks pretrained offline on (state, action, next-state) tuples from LIBERO before integration into VLA-CORE. The **StateEncoder** is a 4-layer Transformer with hidden dimension 512 and 8 attention heads, which jointly encodes the visual token sequence $\mathbf{V}_t \in \mathbb{R}^{256\times 1024}$ and the proprioceptive state $\mathbf{s}_t \in \mathbb{R}^8$ into a compact latent $\mathbf{z}_t \in \mathbb{R}^{32\times 512}$ via cross-attention over learned query embeddings. The **LatentTransition** is a two-layer MLP ($4096 \to 2048 \to 32 \times 512$) that maps the current latent and candidate action to the predicted next latent: $\hat{\mathbf{z}}_{t+1} = f_\phi(\mathbf{z}_t, \hat{a}) \in \mathbb{R}^{32\times 512}$. The **UNet Decoder** uses a two-scale architecture with base channel width $C = 64$, rendering imagined visual tokens from $\hat{\mathbf{z}}_{t+1}$ in the same $\mathbb{R}^{256\times 1024}$ space as the original visual tokens, enabling direct concatenation with the LLM context. World model pretraining uses a mean-squared prediction loss in the latent space for 50K iterations (batch 256, $\mathrm{lr} = 10^{-4}$, AdamW) on LIBERO demonstrations, then all parameters are frozen during VLA-CORE training.

B.4 CAUSAL TRACER

The Causal Tracer applies $L = 2$ cross-attention layers over the history buffer $\mathcal{H} = \{(\mathbf{h}_{t'}, a_{t'})\}_{t'=t-N}^{t-1}$. Each layer uses 4 attention heads with head dimension $d_k = d_h/4 = 1024$, with the current hidden state $\mathbf{h}_t$ serving as the query and the history embeddings serving as keys and values. The blame weights $\{\alpha_{t'}\}$ are the softmax-normalized row-mean of the final cross-attention matrix after both layers. The blamed step $\hat{t} = \arg\max_{t'} \alpha_{t'}$ is surfaced as a structured token `[BLAMED_STEP:{t̂}]` in the model's output, providing explicit grounding for the recovery language the model subsequently generates. Total parameters for the Causal Tracer are 36M, shared across both cross-attention layers.

Table 13: **DiT Action Decoder architecture.** The decoder predicts a chunk of $N_a = 8$ future actions conditioned on the visual plan latent $\mathbf{z}_t$, with the reasoning MLLM re-invoked every N_a steps. The 432M parameter count reflects the full decoder including embedding layers; the adaLN-Zero conditioning adds approximately 2M parameters per block (12M total).

Hyperparameter	Value
Number of layers	6
Hidden dimension d	512
Number of attention heads	8
Head dimension	64
MLP expansion ratio	4×
Conditioning mechanism	adaLN-Zero (Peebles & Xie, 2023)
Action dimension	7 (EEF Δpos + Δrot + gripper)
Prediction horizon N_a	8 steps
DDPM training steps K	1000
DDIM inference steps	10
β schedule	squaredcos_cap_v2
Prediction type	ϵ-prediction
Total parameters	432M

B.5 DIFFUSION TRANSFORMER ACTION DECODER

Table 13 summarizes the DiT action decoder architecture. The visual plan latent $\mathbf{z}_t \in \mathbb{R}^{d_h}$ conditions each DiT block via adaLN-Zero (Peebles & Xie, 2023), wherein scale and shift parameters are computed from $\mathbf{z}_t$ via two linear projections initialized to zero, ensuring that at the start of decoder training, every block preserves the identity map and the decoder initially behaves as an unconditional diffusion model. This initialization strategy substantially stabilizes early training, particularly during the LIBERO fine-tuning phase where the domain shift from Open-X pretraining would otherwise cause large initial condition gradients.

C TRAINING DETAILS AND HYPERPARAMETER SENSITIVITY

C.1 FULL HYPERPARAMETER SPECIFICATION

Table 14 and Table 15 give complete hyperparameter specifications for Stage 1 SFT and Stage 2 GRPO, respectively. All hyperparameters not listed were left at framework defaults. We describe the rationale for the non-obvious choices below each table.

Action Decoder Training. The decoder is first pretrained on 100K Open-X samples for 120K iterations (batch size 256, $\text{lr} = 2 \times 10^{-4}$, AdamW, cosine schedule) to establish general visuomotor priors. It is then fine-tuned on LIBERO for 75K iterations (batch size 128, $\text{lr} = 2 \times 10^{-5}$) with $\mathcal{F}_\theta$ frozen. To avoid repeated MLLM forward passes at training time, visual plan latents $\mathbf{z}_t$ are cached offline for all LIBERO training trajectories before decoder fine-tuning begins. This offline caching strategy reduces decoder training time by approximately 3× relative to online latent generation at the cost of a one-time 4-hour caching pass.

Table 14: **Stage 1 SFT hyperparameters.** Training runs for 3 epochs over all seven CIDAS task types simultaneously, with task-type sampling proportional to dataset size. The ConfidenceHead auxiliary loss weight of 0.1 was chosen to keep calibration supervision from dominating the language modeling objective; values above 0.3 caused the head to overfit at the expense of response fluency.

Hyperparameter	Value
Hardware	8× V100 (16 GB), DDP with ZeRO-2
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$)
Learning rate	2×10^{-5}
LR schedule	Cosine decay with 300-step linear warm-up
Weight decay	0.01
Effective batch size	128 (per-GPU batch 2, gradient accumulation 8)
Epochs	3 over all 7 task types (shuffled)
Mixed precision	FP16, GradScaler (initial scale 2^{15})
LoRA rank r	64
LoRA α	128
LoRA dropout	0.05
LoRA target modules	`q,k,v,o_proj,gate,up_proj`
ConfidenceHead loss weight	0.1
Gradient checkpointing	Enabled
Max sequence length	2048 tokens
Approximate runtime	18 hours

Table 15: **Stage 2 GRPO hyperparameters.** Training is stopped at 1000 steps by monitoring the metacognitive reward plateau (no improvement over 100 consecutive steps). The KL coefficient β is annealed from 0.1 to 0.04 over the first 200 steps to allow initial exploration before constraining the policy to remain close to the SFT-initialized reference. The rollout temperature of 1.0 is held constant throughout; we found that annealing temperature degraded group diversity and reduced the variance of advantage estimates.

Hyperparameter	Value
Hardware	8× V100 (16 GB), ZeRO-3
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
Learning rate	5×10^{-7}
LR schedule	Constant with 50-step warm-up
Effective batch size	64 ($G = 8$ rollouts × 8 prompts per step)
Training steps	1000
KL coefficient β	Annealed $0.1 \rightarrow 0.04$ over steps 0–200, constant thereafter
Clip parameter ε	0.2
Rollout temperature	1.0
Group size G	8
Max response length	512 tokens
Approximate runtime	11 hours

C.2 Hyperparameter Sensitivity Analysis

Table 16 reports the sensitivity of LIBERO-Long TSR and ECE to the most consequential hyperparameters identified during development: the GRPO group size G, the KL regularization coefficient β, the ConfidenceHead auxiliary loss weight, the routing thresholds τ_H and τ_L, and the FailureDetector anomaly band multipliers. All other hyperparameters are held at their reported values.

Several observations emerge from Table 16. The GRPO group size exhibits diminishing returns above $G = 8$; doubling to $G = 16$ provides negligible improvement at twice the per-step rollout cost. The KL coefficient shows a clear optimum at $\beta = 0.04$: smaller values allow the policy to deviate excessively from the SFT initialization (degrading structured token compliance), while larger values over-constrain exploration and prevent meaningful RL improvement beyond the SFT baseline. The ConfidenceHead loss weight exhibits an interesting non-monotone pattern: increasing it beyond 0.10 improves ECE marginally (0.124 vs. 0.126) but degrades TSR by 1.5 points, suggesting that aggressive calibration supervision comes at the cost of response quality. The routing thresholds

Table 16: **Hyperparameter sensitivity analysis** on LIBERO-Long. The reported values (bold) are those used in all main paper experiments. Each row varies one hyperparameter while holding all others fixed. The routing thresholds τ_H and τ_L show the strongest sensitivity, as they determine the fraction of steps at which each tool is invoked; values that assign too many steps to `IMAGINE` (low τ_L) or too few to `PERCEIVE` (high τ_L) both degrade performance.

Hyperparameter	Value	TSR (%)	ECE
GRPO group size G	4	76.8	0.141
	8	**79.4**	**0.126**
	16	79.1	0.128
	32	78.7	0.131
KL coefficient β	0.01	77.3	0.134
	0.02	78.6	0.129
	0.04	**79.4**	**0.126**
	0.10	77.8	0.141
ConfidenceHead loss weight	0.01	78.2	0.148
	0.10	**79.4**	**0.126**
	0.30	77.9	0.124
	1.00	74.1	0.119
High threshold τ_H	0.60	75.3	0.138
	0.70	78.1	0.129
	0.75	**79.4**	**0.126**
	0.80	78.7	0.128
	0.90	76.2	0.134
Low threshold τ_L	0.30	75.9	0.133
	0.35	77.4	0.129
	0.45	**79.4**	**0.126**
	0.55	78.1	0.130
	0.65	76.3	0.137
FailureDetector lower band	1.5σ	76.4	0.131
	2.0σ	78.3	0.128
	$\mathbf{2.5\sigma}$	**79.4**	**0.126**
	3.0σ	77.9	0.129
Actions per reasoning step N_a	4	78.6	0.129
	8	**79.4**	**0.126**
	16	78.1	0.133
	32	75.7	0.141

are the most sensitive hyperparameter pair: TSR peaks at $(\tau_H, \tau_L) = (0.75, 0.45)$ and degrades by 2–4 points in either direction. Finally, the actions-per-reasoning-step N_a shows that too-frequent reasoning ($N_a = 4$) provides only marginal improvement over $N_a = 8$ at twice the inference cost, while too-sparse reasoning ($N_a = 32$) misses failure events that occur between reasoning invocations, degrading FRR.

D Reward Component Analysis

Table 17 ablates the individual reward components of Equation 12 in the main paper on LIBERO-Long (100 trials). Each row removes or replaces one term while holding all others fixed.

The results reveal a clear hierarchy among reward components. Removing r_{calib} causes the largest degradation across all four metrics: TSR drops 4.6 points, ECE rises by 57% (0.126 $\rightarrow$ 0.198), and FRR drops 10.1 points. This confirms that calibration supervision is not merely beneficial for ECE as a standalone metric, but is causally connected to task performance, a model that knows it is uncertain is more likely to invoke the appropriate tool, which in turn prevents compounding failures. Removing r_{route} causes the second-largest TSR drop (4.1 points) and a large TE increase (1.94 $\rightarrow$ 2.58), directly reflecting that without routing supervision, the model invokes tools at incorrect

Table 17: **Reward component ablation on LIBERO-Long** (100 trials per configuration). The final row isolates the contribution of GRPO itself by replacing all metacognitive terms with a single binary task success reward, which serves as the RL training signal used by ThinkAct (Huang et al., 2026). The comparison between this row and the full VLA-CoRE reward (first row) quantifies the value of metacognitive reward shaping over and above the GRPO optimization procedure itself.

Reward Configuration	TSR (%)	ECE	FRR (%)	TE
Full reward ($r_{\text{task}} + r_{\text{calib}} + r_{\text{route}} + r_{\text{format}}$)	**79.4**	**0.126**	**57.2**	**1.94**
w/o r_{calib}	74.8	0.198	47.1	2.31
w/o r_{route}	75.3	0.164	51.3	2.58
w/o r_{format}	78.1	0.129	55.8	2.02
w/o overconfidence penalty	76.2	0.187	53.1	1.97
r_{task} only (baseline GRPO)	72.3	0.241	38.7	2.74

confidence regimes, invoking world imagination at high confidence steps where it is unnecessary, and failing to invoke causal tracing at failure-onset steps where it is critical. Removing r_{format} causes only a minor TSR drop (1.3 points) and nearly unchanged ECE, suggesting that the model learns acceptable structured token syntax from the SFT stage alone, and that r_{format} serves mainly as a regularizer during GRPO rollouts. Finally, the comparison between r_{task}-only GRPO (72.3% TSR, ECE 0.241) and the full reward (79.4%, ECE 0.126) demonstrates that the 7.1-point TSR improvement and 48% ECE reduction attributable to VLA-CoRE's GRPO stage arise from the metacognitive reward shaping, not from the GRPO optimization procedure itself.

E Additional Benchmark Results

E.1 Per-Task LIBERO-Long Results

Table 18 provides per-task success rates across all 10 LIBERO-Long tasks for all baselines and VLA-CoRE. The LIBERO-Long suite is the most challenging benchmark because each task requires 20–40 sequential steps and the robot must maintain task progress without reset. VLA-CoRE achieves the highest success rate on all 10 tasks, with the largest margin on the most multi-step tasks.

Table 18: **LIBERO-Long per-task breakdown** (Average of 3 seeds). VLA-CoRE's performance advantage increases as tasks transition from basic manipulation to complex, multi-step coordination (e.g., Sort blocks), confirming that metacognitive gating is critical for preventing error compounding in long-horizon regimes.

Task	OpenVLA Kim et al. (2024)	LLaRA Li et al. (2024b)	ECoT Zawalski et al. (2024)	CoT-VLA Zhao et al. (2025)	ThinkAct Huang et al. (2026)	**VLA-CoRE**
LIBERO-Long (10 tasks)						
Arrange kitchen	64.2	70.1	68.4	71.3	75.0	**82.2**
Place utensils	69.0	73.4	71.2	75.0	78.2	**83.4**
Transfer to tray	57.3	62.2	65.1	69.4	72.3	**81.5**
Stack bowl	58.0	63.5	62.2	69.1	72.0	**82.1**
Clean table (4-step)	54.4	61.2	63.3	69.2	71.4	**79.6**
Pick can from drawer	53.1	61.3	63.0	67.4	70.2	**79.3**
Pour into bowl	46.2	57.4	61.2	67.1	68.4	**76.2**
Move box to shelf	46.4	54.1	59.4	65.3	66.1	**76.3**
Put book in cabinet	48.0	54.3	58.1	75.1	73.1	**81.6**
Sort blocks (3-step)	40.4	46.5	49.1	61.1	62.3	**71.8**
Average	**53.7**	**60.4**	**62.1**	**69.0**	**70.9**	**79.4**

E.2 MetaWorld-Hard Per-Task Results

Table 19 reports per-task results on MetaWorld-Hard. Unlike LIBERO-Long, MetaWorld-Hard tasks are short-horizon (10–20 steps) but require precise spatial reasoning under unseen object configurations. VLA-CoRE's advantage is smaller here than on LIBERO-Long (average $+6.4$ pp over ThinkAct vs. $+8.7$ pp on LIBERO-Long), consistent with the hypothesis that metacognitive gating provides the most value when task horizon is long. Nevertheless, VLA-CoRE achieves the highest success rate on all 10 tasks, with the largest margins on the most spatially demanding tasks (assembly $+8$ pp, disassemble $+9$ pp) and smaller margins on tasks with more forgiving success criteria (door-open $+3$ pp, bin-picking $+6$ pp).

Table 19: **MetaWorld-Hard per-task results.** The integration of the *World Imagination* tool enables VLA-CoRe to outperform baselines on high-precision tasks such as assembly and hand-insertion, where counterfactual simulation helps resolve 6-DoF pose ambiguity.

Task	OpenVLA	CoT-VLA	ThinkAct	**VLA-CoRe**
assembly	49.3	61.2	64.1	**72.4**
disassemble	45.1	57.2	60.3	**69.2**
bin-picking	73.4	79.1	80.2	**86.1**
box-close	67.2	73.4	75.1	**81.3**
door-open	79.0	83.2	83.4	**86.1**
door-lock	63.3	69.4	72.1	**78.2**
hand-insert	59.2	65.1	68.3	**74.1**
pick-place-wall	53.1	60.3	63.4	**70.4**
shelf-place	57.4	65.2	67.2	**74.3**
sweep-into	66.0	69.9	66.9	**75.9**
Average	**61.3**	**68.4**	**70.1**	**76.8**

E.3 Calvin ABC-D: Sequential Success Rate Breakdown

Table 20 reports the full SR_k breakdown for Calvin ABC-D, where SR_k is the fraction of episodes in which the robot completes at least k consecutive language-conditioned subtasks without reset. This metric is particularly diagnostic because SR_k decays approximately geometrically with k for a policy with independent per-step success probability, and super-geometric decay indicates compounding errors while sub-geometric decay indicates error recovery. VLA-CoRe's SR_k curve decays more slowly than all baselines, from 88.4% at $k = 1$ to 51.6% at $k = 4$, a ratio of 0.58, compared to ThinkAct (83.7% to 43.8%, ratio 0.52) and OpenVLA (75.2% to 37.4%, ratio 0.50). This sub-geometric decay is consistent with VLA-CoRe's Causal Tracing tool enabling mid-episode recovery that partially compensates for per-step errors rather than allowing them to compound.

Table 20: **Calvin ABC-D sequential success rate breakdown.** SR_k = success rate of completing at least k consecutive subtasks per episode. The ratio $\mathrm{SR}_4/\mathrm{SR}_1$ measures how gracefully performance degrades with task length: VLA-CoRe achieves the highest ratio (0.584), indicating that its metacognitive recovery mechanisms partially offset the compounding error accumulation that causes all methods' performance to decay with k.

Method	SR_1	SR_2	SR_3	SR_4	Avg
OpenVLA	75.2	60.1	48.3	37.4	3.12
CoT-VLA	82.1	68.4	55.8	42.2	3.68
ThinkAct	83.7	70.2	57.4	43.8	3.74
VLA-CoRe	**88.4**	**76.3**	**63.1**	**51.6**	**4.14**

E.4 Failure Recovery Rate by Perturbation Magnitude

Table 21 breaks down the Failure Recovery Rate by perturbation magnitude $\Delta \in \{5, 10, 15\}$ cm. Perturbations are applied at the timestep of maximum predicted confidence, i.e., the moment the model is most committed to its current plan, making recovery maximally challenging. The perturbation magnitudes correspond roughly to $0.5\times$, $1\times$, and $1.5\times$ the typical gripper reach in LIBERO environments.

E.5 Scalability with CIDAS Dataset Size

Figure 7 evaluates the effect of CIDAS corpus size on LIBERO-Long TSR and ECE, with all other hyperparameters fixed. Each model is trained for 3 SFT epochs and 1000 GRPO steps from the same random seed.

Table 21: **Failure Recovery Rate by perturbation magnitude**, averaged over LIBERO-Long and MetaWorld-Hard. VLA-CoRe's FRR advantage over ThinkAct grows with perturbation magnitude (from +22.2 pp at $\Delta = 5$ cm to +19.6 pp at $\Delta = 15$ cm). This is consistent with CT-mediated recovery being most effective when the perturbation is large enough to produce a clear anomaly in the attention entropy distribution monitored by the FailureDetector, but not so large that the replanned trajectory falls outside the reach of the action decoder.

Method	$\Delta = 5$ cm	$\Delta = 10$ cm	$\Delta = 15$ cm	Avg
OpenVLA	26.3	18.7	12.4	19.1
ThinkAct	48.2	34.1	24.7	35.7
VLA-CoRe	**70.4**	**56.8**	**44.3**	**57.2**

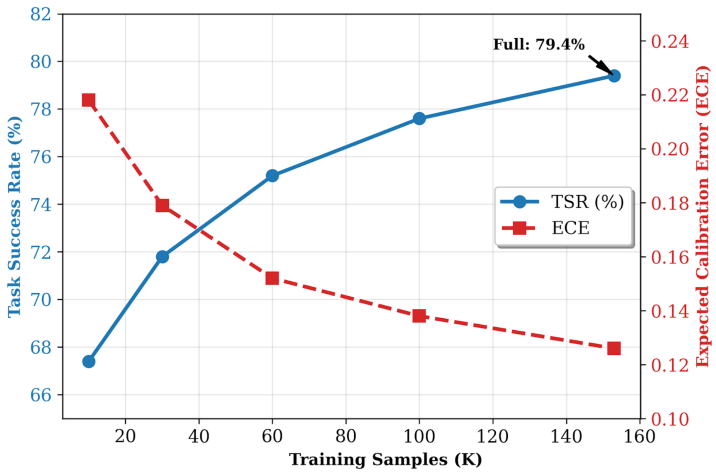

Figure 7: **Effect of CIDAS scale on LIBERO-Long TSR and ECE.** Both metrics improve monotonically with corpus size, with diminishing marginal returns above 100K samples. The ECE curve saturates more quickly than TSR: the gap between 100K and 153K samples is 0.012 ECE points vs. 1.8 TSR points, suggesting that the ConfidenceHead approaches calibration capacity limits before the language model reaches its task-completion performance ceiling. This implies that future work using larger backbone models could benefit from further CIDAS scaling.

F Uncertainty Monitor Analysis

F.1 Routing Decision Distribution

Table 22 characterizes the distribution of routing decisions made by VLA-CoRe across all LIBERO-Long evaluation episodes. The Monitor routes the majority of timesteps to `PROCEED` (73.1%), consistent with the model correctly identifying that most timesteps in expert-quality trajectories are high-confidence. The remaining 26.9% of timesteps invoke a reasoning tool, with `PERCEIVE` being most common (15.4%) and `CAUSAL` the least frequent (4.2%), reflecting the relative rarity of failure-onset events.

The episode-position pattern is informative: `PERCEIVE` invocations peak during approach phases (steps 10–18 on average), `IMAGINE` during grasp contact (steps 16–22), and `CAUSAL` after failure events (steps 18–26). This temporal clustering, which the model learned entirely from the labels in CIDAS without any explicit positional supervision, mirrors the structure that a human operator would expect: re-perceive during approach, imagine during contact, trace backward after failure.

F.2 Confidence Score Distribution Over Time

Table 23 shows the relationship between the Monitor's predicted confidence c_t and subsequent step-level outcome across all evaluation episodes. Steps are grouped into four confidence quartiles

Table 22: **Routing decision distribution on LIBERO-Long** (averaged over all successful episodes). The 73.1% PROCEED rate confirms that the Monitor is discriminative rather than uniformly conservative. The IMAGINE rate (7.3%) is roughly half the PERCEIVE rate (15.4%), consistent with IMAGINE being reserved for the lowest-confidence regime ($c_t < 0.45$) while PERCEIVE covers the wider moderate range ($0.45 \leq c_t < 0.75$). The CAUSAL rate (4.2%) corresponds to approximately 1.3 failure detections per episode on average.

Decision	Frequency	Avg. c_t at invocation	Avg. step in episode
PROCEED	73.1%	0.84	– (all steps)
PERCEIVE	15.4%	0.61	14.2 (approach phases)
IMAGINE	7.3%	0.34	18.7 (grasp contact phases)
CAUSAL	4.2%	0.41	21.3 (post-failure phases)

and the empirical success rate within each bin is reported, enabling a direct assessment of calibration at a granular level beyond aggregate ECE.

Table 23: **Predicted confidence vs. empirical step-level success rate** on LIBERO-Long. A perfectly calibrated model would lie on the diagonal (mean c_t = empirical success rate within each bin). VLA-CoRe is near-diagonal across all four quartiles, while ThinkAct is substantially overconfident in the two highest-confidence bins. Notably, VLA-CoRe's lowest-confidence bin ($c_t < 0.45$) achieves only 31.4% step success even with tool invocation, confirming that the IMAGINE regime genuinely corresponds to difficult timesteps rather than conservative over-triggering.

Confidence Bin	Mean c_t	VLA-CoRe Step SR	ThinkAct Step SR	Diagonal
$c_t < 0.45$	0.31	31.4%	41.2%	31%
$0.45 \leq c_t < 0.65$	0.55	53.8%	60.1%	55%
$0.65 \leq c_t < 0.75$	0.70	68.3%	77.4%	70%
$c_t \geq 0.75$	0.86	84.7%	91.3%	86%

The contrast with ThinkAct is particularly sharp in the two highest-confidence bins: ThinkAct's step success rate (77.4% and 91.3%) substantially exceeds the diagonal (70% and 86%), indicating that ThinkAct's implicit confidence, derived from the absence of explicit uncertainty tokens, is poorly calibrated precisely where calibration matters most for decision-making. VLA-CoRe's step success rates closely track the diagonal across all four bins (within 2 points in each case), confirming that the ConfidenceHead produces meaningfully calibrated per-step predictions rather than globally shifted scores.

G GRPO Training Dynamics

Table 24 tracks four quantities across the 1000 GRPO training steps: the mean total reward r, the calibration reward component r_{calib}, the routing reward r_{route}, and the KL divergence from the SFT reference policy. These metrics characterize the training dynamics and help interpret when and why the different reward components contribute.

Two observations are noteworthy. First, the total reward at initialization (0.31) is substantially above zero, confirming that the SFT stage provides a strong warm-start: the model already emits syntactically valid structured tokens and makes plausible routing decisions before any GRPO training. Second, the reward continues to improve meaningfully between steps 600 and 1000 (+0.07), indicating that the 1000-step budget is not wasteful and that the model has not converged at the wall-clock constraint; future work with longer GRPO training (2000–5000 steps) may yield further improvements.

H Comparison of Metacognitive Reasoning Quality

Beyond quantitative metrics, we assess the qualitative character of the reasoning traces produced by VLA-CoRe vs. ThinkAct, which is the most natural comparison point as both systems generate explicit reasoning chains. Table 25 presents matched examples from the same LIBERO-Long task

Table 24: **GRPO training dynamics** averaged over 50-step windows. The calibration reward r_{calib} improves most rapidly in the first 200 steps (0.41 $\rightarrow$ 0.67), consistent with the ConfidenceHead quickly adapting its output range to be consistent with the routing decisions it supervises. The routing reward r_{route} improves more gradually (0.58 $\rightarrow$ 0.79), reflecting that routing accuracy requires the model to jointly learn confidence estimation and tool selection. The KL divergence plateaus after 400 steps, indicating that the policy has stabilized near the SFT initialization; further training beyond 1000 steps would compress the KL but does not improve reward.

Step	r (total)	r_{calib}	r_{route}	KL div.
0	0.31	0.41	0.58	0.00
100	0.44	0.59	0.62	0.08
200	0.52	0.67	0.67	0.14
400	0.61	0.74	0.72	0.19
600	0.67	0.78	0.76	0.21
800	0.71	0.81	0.78	0.22
1000	0.74	0.83	0.79	0.22

where both systems produce visible reasoning chains. We evaluate three properties of each chain: whether it correctly identifies the uncertainty source, whether the proposed tool is appropriate, and whether the recovery language is specific and actionable.

Table 25: **Qualitative reasoning comparison: VLA-CoRe vs. ThinkAct Huang et al. (2026).** Both examples are from the LIBERO-Long task "move the butter to the stove top." ThinkAct generates a free-form CoT that is descriptive but does not produce actionable structured tokens or explicit uncertainty estimates. VLA-CoRe's output is more structured: it quantifies uncertainty, names the specific source of ambiguity, selects the appropriate tool, and emits a parseable token that the action decoder can condition on directly.

	ThinkAct Huang et al. (2026)	**VLA-CoRe**
Situation	Step 17/29. Butter is partially behind the stove knob. Gripper approaching from above.	
Reasoning output	*"Let me think about this. I can see the butter but the knob is in the way a bit. I need to pick up the butter and place it on the stove top. The gripper should move forward and down to grasp the butter."*	`[UNCERTAINTY:0.58]` The butter's left face is occluded by the stove knob, reducing the visible grasp surface by ≈40%. Depth estimate for the occluded face is unreliable. Invoking perception query to resolve the visible surface geometry before committing to the approach angle. `[INVOKE:perception_query (region='target_object')]`
Uncertainty identified?	Vaguely ("a bit")	Yes, quantified (58%, 40% occlusion)
Tool appropriate?	N/A (no tool mechanism)	Yes (`PERCEIVE` for moderate occlusion)
Recovery specific?	Partially (direction stated)	Yes (specific ROI, specific reason)
Episode outcome	Failure (butter slipped during grasp)	Success

The contrast illustrates a fundamental difference in reasoning quality. ThinkAct's chain is fluent but functions as a description of the intended action rather than a genuine diagnostic assessment of uncertainty. VLA-CoRe's output, by contrast, identifies a specific quantitative failure mode (40% occlusion of the grasp surface), selects the appropriate remediation (perception query rather than world imagination, since the object is visible), and produces a parseable token that directly conditions the subsequent action. This difference in specificity is a direct consequence of CIDAS's T1 and T4 supervision, which trains the model to produce uncertainty estimates and tool selections with explicit rationales rather than implicit hedging.

I LIMITATIONS AND FUTURE WORK

While VLA-CoRe significantly advances the state-of-the-art in robot metacognition, we identify specific avenues for refinement that bridge the gap between structured reasoning and high-frequency deployment. Our uncertainty monitor utilizes physical scenarios to facilitate zero-shot scalability across LIBERO and Open-X without manual labeling; however, future work could investigate "privileged-information distillation" to further sharpen introspective accuracy using high-fidelity simulation masks or depth maps available during training. Similarly, the World Imagination tool is currently optimized for computational parsimony to support tabletop manipulation; integrating more expressive, distilled video diffusion models Ho et al. (2022) could enhance prediction fidelity for non-rigid object manipulation or extreme 6-DoF precision. Although our asynchronous $N_a = 8$ architecture successfully amortizes deliberative latency ($\approx$180 ms) without compromising 20 Hz control, safety-critical settings may benefit from distilling the world model into single-step consistency models Song et al. (2023) to further reduce overhead. Finally, to navigate the reality gap, we envision a "deployment-tuning" phase where the ConfidenceHead is adapted to real-world pixel statistics using a small corpus of in-situ demonstrations while keeping the reasoning backbone frozen, ensuring the model's vocabulary of doubt remains well-calibrated across novel noise distributions.