CDE: Concept-Driven Exploration for Reinforcement Learning

Le Mao[1], Andrew H. Liu[2], Renos Zabounidis[3], Zachary Kingston[2], Joseph Campbell[2]

***Abstract*— Intelligent exploration remains a critical challenge in reinforcement learning (RL), especially in visual control tasks. Unlike low-dimensional state-based RL, visual RL must extract task-relevant structure from raw pixels, making exploration inefficient. We propose Concept-Driven Exploration (CDE), which leverages a pre-trained vision–language model (VLM) to generate object-centric visual concepts from textual task descriptions as weak, potentially noisy supervisory signals. Rather than directly conditioning on these noisy signals, CDE trains a policy to reconstruct the concepts via an auxiliary objective, using reconstruction accuracy as an intrinsic reward to guide exploration toward task-relevant objects. Because the policy internalizes these concepts, VLM queries are only needed during training, reducing dependence on external models during deployment. Across five challenging simulated visual manipulation tasks, CDE achieves efficient, targeted exploration and remains robust to noisy VLM predictions. Finally, we demonstrate real-world transfer by deploying CDE on a Franka Research 3 arm, attaining an 80% success rate in a real-world manipulation task. Code and videos are available at: https://sites.google.com/view/concept-learn/home.**

I. Introduction

Reinforcement learning (RL) has shown impressive performance across a variety of robotic tasks, including manipulation [1–3], navigation [4–6] and task planning [7]. Yet exploration remains challenging, especially under sparse or delayed rewards where random exploration leads to wasteful environment interactions. This difficulty is amplified in visual control: policies must first learn to extract task-relevant objects and relations from high-dimensional images in order to effectively ground credit assignment. Recent works have explored leveraging pre-trained VLMs to automatically generate dense reward signals which incorporate task-related domain knowledge [8–13], greatly simplifying the learning process. However, in practice VLMs produce noisy outputs which often lead to incorrect reward signals, and directly optimizing over these can reduce training effectiveness.

In this work, we propose a novel method for improving sample efficiency with VLM guidance. Our method, Concept-Driven Exploration (CDE), takes a representation-first approach: a VLM is used to shape a policy's learned features, and those features in turn guide exploration. From a natural language task description, a VLM proposes relevant visual concepts—concrete, task-level cues such as the segmentation mask of a target object (see Fig. 1). Rather than directly conditioning the policy on these concepts [14–16], which are only as accurate as the VLM itself, CDE instead assumes the visual concepts are inherently noisy and treats them as weakly supervised learning targets [17, 18]. The policy is trained to predict these concepts via an auxiliary reconstruction loss, and uses the reconstruction error as an intrinsic reward to guide exploration.

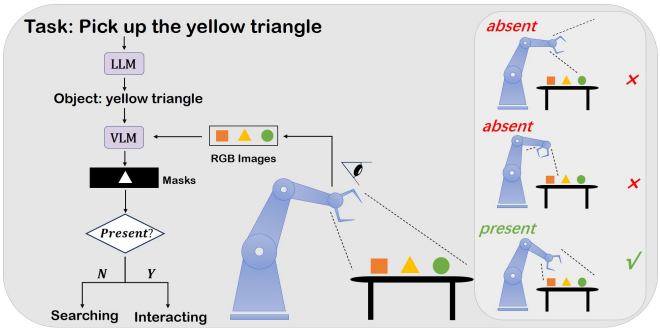


Fig. 1: Concept-Driven Exploration overview. Task-relevant objects are identified from a task description and then used by a VLM to generate segmentation masks for each object. These segmentation masks shape policy representation learning and guide exploration.

This yields four benefits: (i) the agent is incentivized to look at and attend to the target objects; (ii) as predictions improve, the agent moves progressively closer to the target objects so as to encounter novel states with high reconstruction error, resulting in a targeted breadth-first search-like pattern; (iii) by treating incorrect concepts as supervised label noise, the impact of VLM errors is mitigated during training instead of directly misguiding exploration [19]; and (iv) because the policy has learned to predict concepts, we no longer need to query the VLM at test-time. Intuitively, the VLM-generated concepts serve as "hints" which help the policy learn to recognize task-relevant objects, while the intrinsic reward guides exploration toward them.

Further, prior works often rely on global or multi-camera views. However, at deployment time a global camera view may not always be available, therefore, we assume access to only wrist-mounted camera observations for a more generalizable framework. Compared to a fixed global camera that offers relatively stable visual observations, a wrist-mounted camera returns frames with drastic visual changes, and the target object is not always visible in the camera view, making policy learning more challenging. To address this, we use Concept Embedding Models (CEMs) to learn dual object representations: one for when the object is visible, and one for when it is not. Since policy behavior is visibility-dependent—interact with the object if it is visible, search for it otherwise—the two representations capture complementary

[1] LM is with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA mao214@purdue.edu

[2] AL, ZK and JC are with the Department of Computer Science, Purdue University, West Lafayette, IN 47906, USA {liu3458, zkingston, joecamp}@purdue.edu

[3] RZ is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA renosz@andrew.cmu.edu

features, resulting in improved learning efficiency.

We summarize our contributions as follows:

- We propose a concept-driven exploration method that utilizes VLMs to generate visual concepts in a zero-shot manner, with no manual annotations.
- We treat these visual concepts as weakly supervised targets to learn task-relevant representations and generate intrinsic rewards for exploration.
- We integrate CEMs with the policy network to represent both the presence and absence of task-relevant objects, resulting in representations compatible with wrist-mounted cameras where objects may not be visible.
- We empirically show that CDE out-performs baselines on five visual manipulation tasks and evaluate transfer to real-world settings.

II. Related Work

A. Vision-based Reinforcement Learning

Existing works have explored both model-based and model-free RL approaches for visuomotor control. For model-based RL, PlaNet [20] learns environment dynamics from only image observations. Visual Foresight [21] predicts future frames conditioned on past actions and current image observation. Without reversible access to MDP dynamics [22], model-free approaches are typically less efficient than model-based approaches. Previous works have tried different methods to improve sample efficiency of model-free RL in image space. CURL [23] leverages contrastive learning to learn rich representations and thus accelerate policy learning. DrQ [24] and DrQv2 [25] apply image augmentation to the training process and achieve state-of-the-art performance on several DeepMindControl tasks. SEAR [26] learns agent and background representations through reconstruction. DEAR [27] extends SEAR by maximizing distance between two representations but without background reconstruction. However, the above methods fail to focus on task-relevant objects, leading to inefficient exploratory actions. In contrast, CDE learns task-relevant visual concepts of target objects, leading to efficient and object-centric exploration.

B. Concept Learning in Reinforcement Learning

Previous works have explored reconstructing pre-defined concepts from intermediate embeddings and using these for downstream policy learning. Concept Policy Models [28] apply Concept Bottleneck Models (CBM) [29] to multi-agent RL. SCoBots [30] extracts object-related symbolic concepts from raw observations and refines them into human-understandable relational concepts, with concept pruning and reward shaping performed by human experts. LICORICE [31] focuses on reducing human labor in labeling concepts and querying a VLM for labels. However, the above works require human-defined concepts which may vary across environments. In contrast, CDE only uses segmentation masks as concepts, which improves generalization ability. Moreover, VLM outputs are often noisy and directly conditioning on them may hurt policy learning. In contrast, CDE absorbs such errors as label noise during training.

C. Intelligent Exploration

Intelligent and efficient exploration remains a critical challenge in RL. For example, frequency-based methods encourage novelty-seeking behavior by giving higher rewards to states with lower visitation counts [32]. Curiosity-based methods incentivize the policy to visit unseen states through intrinsic motivation, by predicting future states and using the prediction error as an additional reward. ICM [33] predicts environment dynamics with inverse and forward models. RND [34] forces the prediction network to approximate a randomly initialized network. VIME [35] leverages information gained from the agent's belief. However, the above approaches focus only on exploring novel states which is inefficient since most exploration does not involve interacting with the target object.

III. Preliminaries

A. Reinforcement Learning

Reinforcement learning is modeled as a Markov decision process (MDP) described by the tuple $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$. $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space. $P(s_{t+1}|s_t, a_t) \in [0, 1]$ denotes the transition probability from state s_t to s_{t+1} given the action a_t. $R(s_t, a_t) \in \mathbb{R}$ denotes the reward function. $\gamma \in (0, 1)$ is the discount factor. At each time step t, the agent takes an action a_t from policy $\pi(a_t|s_t)$ given the state s_t. The objective is to maximize the expected return $\mathbb{E}_{s,a}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

B. Concept Embedding Models

Concept Embedding Models (CEMs) [36] were originally proposed as interpretable image classification models. CEMs map the input image $\mathbf{x}$ into a set of human-defined concepts $C = \{c_1, c_2, \cdots, c_n\}$ through an intermediate layer $f(\cdot)$. A concept c_i is represented by two embeddings $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$:

$$\hat{\mathbf{c}}_i^+, \hat{\mathbf{c}}_i^- = f(\mathbf{x}). \tag{1}$$

The positive embedding $\hat{\mathbf{c}}_i^+$ represents that the concept c_i is active, while the negative embedding $\hat{\mathbf{c}}_i^-$ represents that the concept is inactive. A probability $\hat{p}_i \in [0, 1]$ is predicted from the two embeddings through a probability generator $P_i(\cdot)$ to indicate whether the concept is active. The final concept embedding $\hat{\mathbf{c}}_i$ is the weighted mixture of $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$ which is formulated as:

$$\hat{\mathbf{c}}_i = \hat{p}_i \hat{\mathbf{c}}_i^+ + (1 - \hat{p}_i)\hat{\mathbf{c}}_i^- \quad \text{where} \quad \hat{p}_i = P(\hat{\mathbf{c}}_i^+, \hat{\mathbf{c}}_i^-). \tag{2}$$

IV. Concept-Driven Exploration

We propose CDE, a concept-driven exploration method for improving RL sample efficiency in vision-based manipulation tasks (See Fig. 2). Before each episode, an LLM extracts relevant target objects from a task description. During policy rollout, a VLM provides per-step (potentially noisy) segmentation masks of each object. The policy encodes each image into a concept embedding used by downstream policy layers. We train this embedding jointly with the RL objective and a mask-reconstruction loss, which is computed by decoding the target mask from the embedding and comparing it to the VLM-generated mask. The reconstruction loss serves

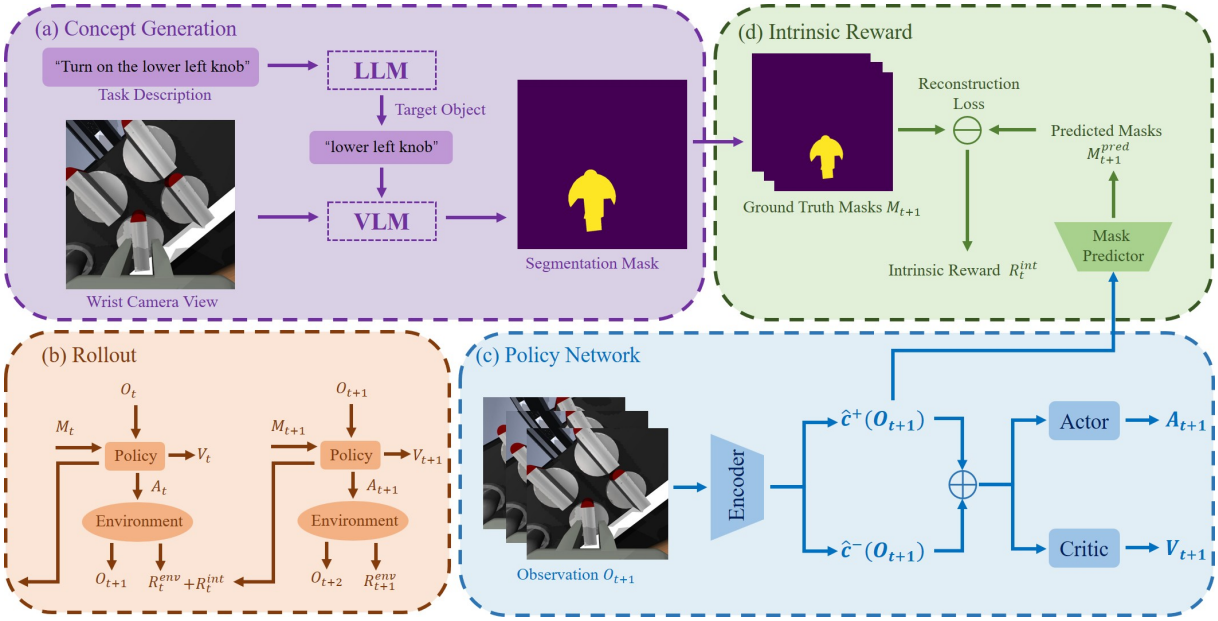


Fig. 2: Architecture. (a) The LLM parses the task description to extract the target object. The VLM segments masks on input RGB images. (b) During training, the policy takes segmentation masks as additional input and generates intrinsic reward signals for the last timestep. (c) At each timestep $t+1$, the policy network receives environment observation O_{t+1}, and encodes O_{t+1} into a positive embedding $\hat{\mathbf{c}}^+(O_{t+1})$ and a negative embedding $\hat{\mathbf{c}}^-(O_{t+1})$, the final concept embedding is a weighted sum of the two embeddings. (d) The segmentation masks M_{t+1} are used to supervise the mask reconstruction from the positive embedding $\hat{\mathbf{c}}^+(O_{t+1})$ and generate intrinsic reward signal R_t^{int}.

two roles: (a) an intrinsic reward that guides object-centric exploration, and (b) an auxiliary optimization objective that shapes the representation toward object-relevant features (See Alg. 1). Rather than providing segmentation masks directly as observations, we treat masks as supervision for the policy's internal representations.

A. Concept Generation

Prior works [28, 30, 31] often rely on human-interpretable concepts (e.g., relative position, orientation), however, in visual control tasks such concepts are difficult to define purely from RGB images and task descriptions. To solve this problem, we propose to use the *segmentation mask* of the target object as a concept. We first use an LLM to generate a list of objects that the policy should interact with. Below is an example for generating a list of objects for the *Kitchen-Microwave* task using OpenAI GPT-4 [37]:

> **Prompt:** Task description: "Open the microwave door". Give me a list of objects to interact with in order to solve the task. Please return a Python list. Do not output anything else.
> **Output:** ["microwave door handle"]

Given the resulting task-related objects, we query a VLM to generate segmentation masks for each object in the list from visual observations.

B. Concept Learning with Concept Embedding Models

Because VLM-generated segmentation masks are imperfect, providing them directly as policy inputs yields poor performance. We instead use the masks as auxiliary targets, guiding the policy to learn object-centric representations. However, unlike previous works [26, 27] that use a global camera, our wrist-mounted camera frequently omits the target from the frame. In such frames, embeddings that encode only object-specific features may not aid learning (See Sec. V-E). To address this, we use CEMs to learn two complementary representations: one for when the object is present and one for when it is absent.

The CEM represents a concept with two embeddings $\hat{\mathbf{c}}_i^+$ and $\hat{\mathbf{c}}_i^-$. For visual manipulation tasks, the positive embedding $\hat{\mathbf{c}}_i^+$ corresponds to the case when the target object is present in the observation and the negative embedding $\hat{\mathbf{c}}_i^-$ when the target object is absent. Given the two embeddings and a probability p_i that the object is present, we compute a weighted sum of the concept embeddings for the downstream policy (Eq. (2)). Unlike in the original CEM setting, we do not predict the probability from the embeddings, since we know whether the object is present from the segmentation mask. Namely, we treat p_i as a binary value such that

$$\hat{\mathbf{c}}_i = p_i\hat{\mathbf{c}}_i^+ + (1-p_i)\hat{\mathbf{c}}_i^- \quad \text{where} \quad p_i = \begin{cases} 1 & \text{if px} \geq \epsilon \\ 0 & \text{if px} < \epsilon, \end{cases} \tag{3}$$

where px is the number of active pixels in the segmentation mask and ϵ is a small threshold, e.g., 20. During training this is computed from the VLM-generated segmentation mask, while at deployment we use the predicted mask. We additionally introduce an auxiliary reconstruction loss $\mathcal{L}_{\text{recons}}$. Specifically, we use the positive embedding to reconstruct the segmentation mask of the target object. This encourages the positive embedding to encode object-related visual informa-

Algorithm 1 Concept-Driven Exploration

Require: LLM, VLM, Encoder E_θ, Mask Predictor MP_ϕ, task description d

Object $C \leftarrow \text{Prompt}(\text{LLM}, d)$
for $t = 1$ to T **do**
  Collect transition $\left(o_t, m_t^{\text{gt}}, a_t, r_t^{\text{env}}, o_{t+1}\right)$
  $m_{t+1}^{\text{gt}} \leftarrow \text{VLM}\left(o_{t+1}, C\right)$
  $\hat{c}_{t+1}^{+}, \hat{c}_{t+1}^{-} \leftarrow E_\theta\left(o_{t+1}\right)$
  $m_{t+1}^{\text{pred}} \leftarrow MP_\phi\left(\hat{c}_{t+1}^{+}\right)$ ▷ Mask Prediction
  $\mathcal{L}_{\text{recons}} = \mathcal{L}_{\text{BCE}}\left(m_{t+1}^{\text{pred}}, m_{t+1}^{\text{gt}}\right)$
  $r_t^{\text{int}} = \gamma \text{clip}\left(\mathcal{L}_{\text{recons}}, 0, 1\right)$ ▷ Intrinsic Reward
  $\mathcal{D} \leftarrow \mathcal{D} \cup \left(o_t, m_t^{\text{gt}}, a_t, r_t^{\text{env}} + r_t^{\text{int}}, o_{t+1}\right)$
  Update $(\mathcal{D})$
end for
function Update $(\mathcal{D})$
  $\left(o_t, m_t^{\text{gt}}, a_t, r_{t:t+n-1}, o_{t+n}\right) \sim \mathcal{D}$
  Sample data augmentation A
  $\hat{c}_t^{+}, \hat{c}_t^{-} \leftarrow E_\theta\left(A\left(o_t\right)\right)$
  $m_t^{\text{pred}} \leftarrow MP_\phi\left(\hat{c}_t^{+}\right)$
  $\mathcal{L}_{\text{recons}} = \mathcal{L}_{\text{BCE}}\left(m_t^{\text{pred}}, A\left(m_t^{\text{gt}}\right)\right)$
  Compute $\mathcal{L}_{\text{critic}}$ ▷ See [25]
  $\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{critic}} + \beta \mathcal{L}_{\text{recons}}$
  Update Critic, E_θ and MP_ϕ using $\mathcal{L}_{\text{total}}$
  Update Actor using RL
end function

tion. The final objective is:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{critic}} + \beta \mathcal{L}_{\text{recons}}. \tag{4}$$

C. Intrinsic Reward

In addition to learning object-related representations, we find that segmentation mask reconstruction can also be used to generate intrinsic reward signals to incentivize exploration.

$$r_t^{\text{total}} = r_t^{\text{env}} + \gamma \text{clip}\left(\mathcal{L}_{\text{recons}}, 0, 1\right). \tag{5}$$

Since the model is supervised to reconstruct segmentation masks from the positive embedding $\hat{\mathbf{c}}_i^{+}$ through a mask predictor MP_ϕ, the reconstruction loss is expected to be smaller for previously visited states than for unseen states. Therefore, the policy is encouraged to visit novel states where the target object is present to maximize the reconstruction loss.

V. Experiments

A. Environment

We conduct experiments to evaluate CDE's performance on two robot manipulation benchmarks (see Fig. 3).

Franka Kitchen [38]: is a challenging RL benchmark for visual control. The agent needs to interact with various objects in the kitchen given only sparse reward signals. We choose 4 tasks: *Microwave*, *Knob*, *Switch* and *Cabinet*.

Robosuite [39]: contains several challenging robotic tabletop manipulation tasks. We evaluate on the *Lift* task with sparse rewards.

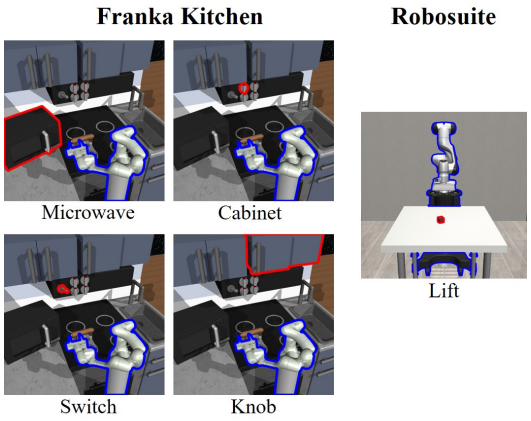


Fig. 3: Environment setup. The agent is required to interact with the outlined target object to accomplish the task.

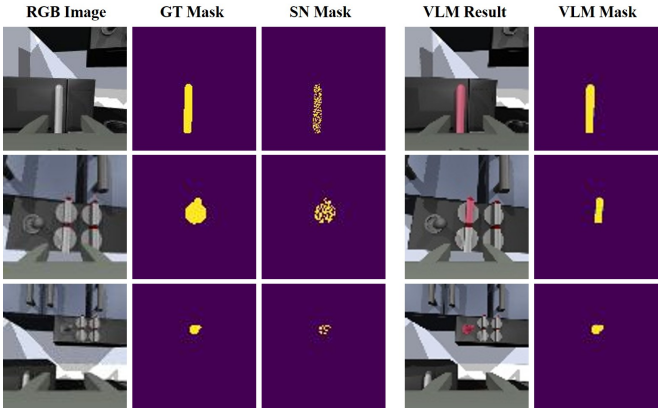


Fig. 4: Examples masks under each noise setting. The resolution of both images and masks is 84 × 84, for VLM mask, we segment the mask on 320 × 320 RGB images and downsample to 84 × 84.

B. Baselines

We compare CDE with the following baselines, representing standard methods of incorporating segmentation masks into policy learning.

- **DrQv2-RGB** [25]: state-of-the-art model-free visual RL using only RGB observations.
- **DrQv2-RGBM**: DrQv2 with the VLM-generated segmentation mask concatenated as a fourth image channel.
- **DrQv2-ME**: DrQv2 with a dedicated segmentation mask encoder that produces a mask embedding which is then fused with the RGB embedding.

C. Experimental Setup

We evaluate CDE under three noise settings (See Fig. 4), including: ground truth (GT) masks, synthetic noise (SN) masks, and VLM-generated masks. GT masks are obtained through MuJoCo's [40] rendering API. SN masks are derived from the GT mask by randomly inverting pixels according to a binomial distribution. VLM masks are generated with Grounded-SAM2 [41], which integrates the open-vocabulary object detection model Grounding-DINO [42] with Segment Anything Model 2 (SAM2) [43].

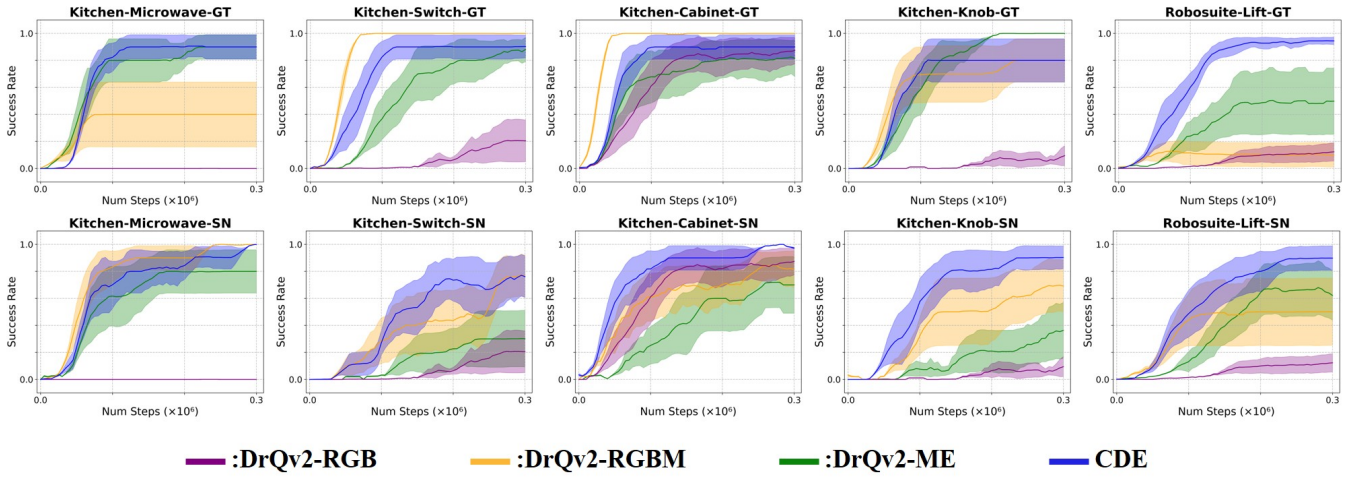


Fig. 5: Simulated task results. All tasks are run across 10 random seeds and we report average success rate with standard error. (Top row) Learning with ground truth masks. (Bottom row) Learning with masks with synthetic noise. CDE shows better stability and robustness to noisy mask inputs than baselines.

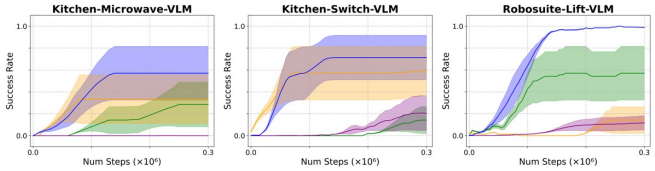


Fig. 6: Simulated task results with VLM-generated masks. Due to computational constraints, we run a subset of the tasks across 7 random seeds and report average success rate with standard error. For DrQv2-RGB, we use the results from previous experiments for comparison.

For DrQv2-RGB, we use the task reward only. For DrQv2-RGBM and DrQv2-ME, the segmentation mask is provided as input, so no reconstruction is available to supply an intrinsic reward. Instead, we provide a shaping reward proportional to the number of pixels in the segmentation mask. For all environments, the resolution of both RGB images and masks is 84×84, and a frame stack of 3 is used. We run all the experiments for 0.3M timesteps on an NVIDIA A100 GPU (40GB); each run takes around 4 hours with GT or SN mask, and around 8 hours with the VLM mask.

D. Results and Analysis

From the results shown in Fig. 5, in the GT setting we observe that CDE achieves the highest average success rate on the *Microwave* and *Lift* tasks, but lower success rate on the *Knob*, *Switch* and *Cabinet* tasks. The latter three tasks are simpler and so the exploration benefits resulting from our intrinsic reward are less pronounced. In the SN setting, we observe that CDE outperforms all baselines on the *Knob*, *Switch*, *Cabinet* and *Lift* tasks, and achieves the same success rate with DrQv2-RGBM on the *Microwave* task. The baselines directly operate over masks and propagate mask errors into both perception and exploration, whereas CDE reconstructs masks from learned embeddings, reducing the impact of noise.

An interesting exception is the *Knob* task, where performance in the SN setting exceeds GT. We conjecture that multiple non-target knobs act as distractors under GT: when reconstructing the top-left knob, the presence of other knobs inflates the reconstruction loss and weakens the exploration signal. Our synthetic noise setting perturbs only the target object's mask, incidentally focusing the policy on the target and yielding a more informative intrinsic reward.

We show results with VLM-generated masks in Fig. 6. CDE outperforms all baselines on all tasks and achieves performance similar to that under GT mask and SN mask settings, while the performance of the baselines is even worse than the SN setting. These results suggest that CDE learns useful object representations and produces informative exploration rewards, even in the presence of strong noise.

E. Ablation Studies

We conduct ablation studies to investigate the contribution of key components of CDE (See Tbl. I). First, using both positive and negative embeddings (CDE, Model 2) outperforms variants with only positive embeddings (Model 4, Model 3). This confirms that the CEM helps the policy in cases where the object is not visible, improving learning.

Second, we compare the impact of pixel reward (PR) and reconstruction reward (RR). Although variants with PR (Model 2) and PR+RR (Model 1) show strong performance on Franka Kitchen tasks, their performance degrades significantly on the *Lift* task, whereas our RR-based method

	Component				Success Rate (%)					
	PE	NE	RR	PR	Microwave	Knob	Switch	Cabinet	Lift	Average
1	✓	✓	✓	✓	**90** ±09	80 ±16	**90** ±09	**100** ±00	40 ±23	80
2	✓	✓	×	✓	**90** ±09	**90** ±09	**90** ±09	**100** ±00	48 ±24	84
3	✓	×	×	✓	80 ±16	**90** ±09	81 ±13	**100** ±00	40 ±24	78
4	✓	×	✓	×	64 ±20	60 ±24	80 ±16	87 ±09	93 ±03	77
CDE	✓	✓	✓	×	**90** ±09	77 ±16	**90** ±09	90 ±09	**95** ±02	**88**

TABLE I: Ablation studies. PE stands for positive embedding, NE stands for negative embedding, RR stands for reconstruction reward and PR stands for pixel reward. We report the average success rate with standard error, the highest success rate for each task is highlighted in yellow.

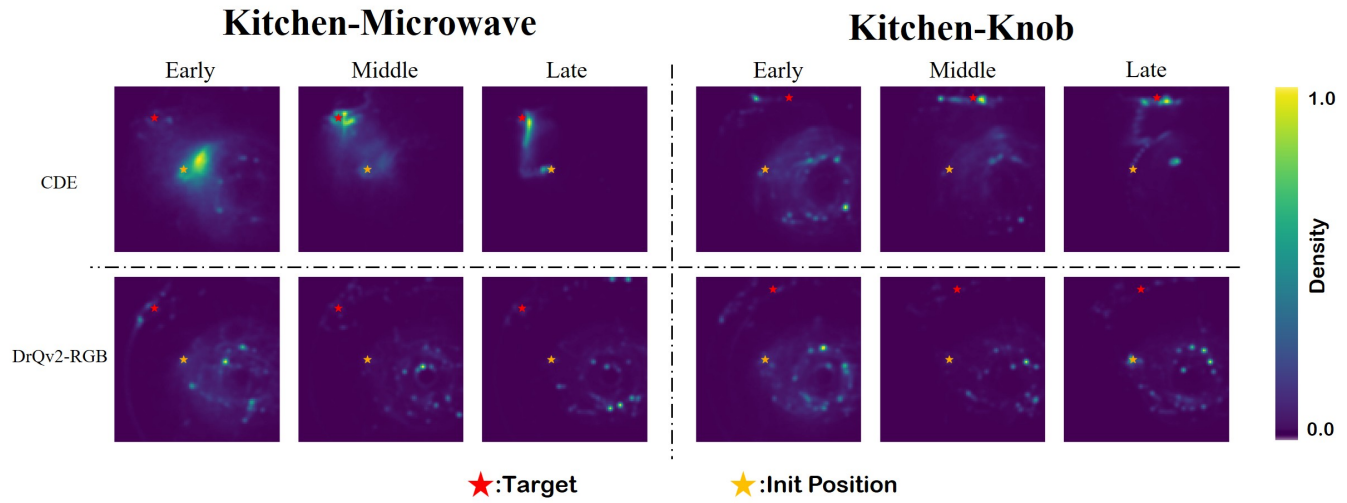
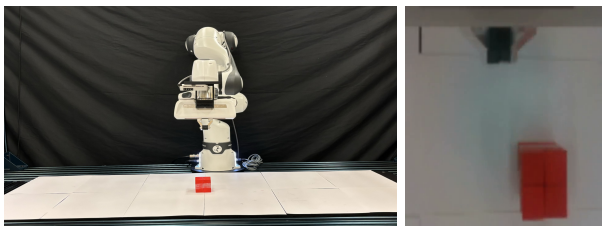


Fig. 7: Heatmap showing visited states during three different stages of training: Early (33%), Middle (66%), Late (100%). CDE explores states near the target object, while DrQv2-RGB explores largely randomly.



(a) (b)

Fig. 8: (a) Real-world task setup. (b) Wrist-mounted camera view.

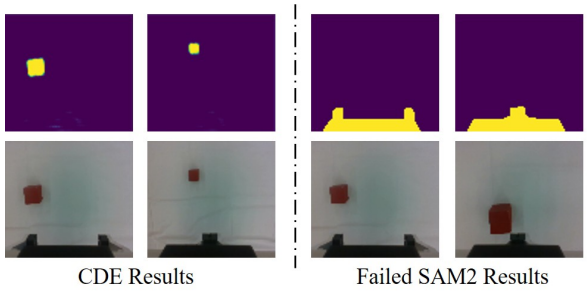


Fig. 9: Comparison between CDE mask predictor and SAM2 with on real-world images. CDE successfully segments the target object without any fine-tuning, demonstrating robust sim-to-real transfer generalization under real-world visual noise.

remains stable across all tasks and achieves the highest average success rate. This suggests that PR is sensitive to fine-grained mask accuracy and visual noise, while RR provides a more robust, task-agnostic intrinsic signal.

F. Exploration Analysis

To better understand how CDE influences exploration, we record end-effector positions at early, middle, and late stages of training and visualize the positions in a heatmap as shown in Fig. 7. We observe that in the early stage, both DrQv2-RGB and CDE explore near the initial position. By the middle stage, CDE has identified the target and begins consistent interactions with it, whereas DrQv2-RGB continues random exploration. In the late stage, CDE consistently completes the task and the heatmap exhibits a concentrated trajectory around the learned solution.

G. Real-World Experiments

We also conduct real-world experiments with a Franka Research 3 robot arm and an Intel Realsense D435i camera mounted on the wrist as shown in Fig. 8a. We train a policy on the *Lift* task with an end-effector position controller. We directly perform sim-to-real transfer without any fine-tuning for both CDE and DrQv2-ME. We execute 10 policy rollouts, resulting in 8/10 successes for CDE and 0/10 successes for DrQv2-ME. We attribute DrQv2-ME's failures to inaccurate object distance estimation and poor VLM segmentation results, which we show in Fig. 9. Grounded-SAM2 occasionally segments the irrelevant objects, rather than the target object.

VI. Conclusion

In this paper, we propose CDE, a novel concept-driven exploration method for RL, which uses a VLM to discover task-relevant visual concepts. CDE treats VLM outputs as noisy supervision for representation learning and uses reconstruction errors as intrinsic rewards, yielding generalizable object-centric exploration without relying on VLM inputs at test-time. Our approach supports wrist-mounted camera observations by learning dual object representations: one embedding when the object is visible, and one when it is not. This allows the policy to learn complementary features for each behavior mode, i.e. searching for the object vs interacting with it. Our experiments show that CDE is more stable and robust compared to baselines in both simulated and real-world manipulation tasks. CDE is simple to plug into existing RL frameworks, and we believe it opens avenues for further research in efficient object-centric exploration.

References

[1] D. Han, B. Mulyana, V. Stankovic, and S. Cheng. "A Survey on Deep Reinforcement Learning Algorithms for Robotic Manipulation". In: *Sensors* 23.7 (2023), p. 3762.

[2] H. Nguyen and H. La. "Review of Deep Reinforcement Learning for Robot Manipulation". In: *International Conference on Robotic Computing*. IEEE. 2019, pp. 590–595.

[3] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, and B. Dresp-Langley. "Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-review". In: *Robotics* 10.1 (2021), p. 22.

[4] K. Zhu and T. Zhang. "Deep Reinforcement Learning Based Mobile Robot Navigation: A Review". In: *Tsinghua Science and Technology* 26.5 (2021), pp. 674–691.

[5] F. Zeng, C. Wang, and S. S. Ge. "A Survey on Visual Navigation for Artificial Agents with Deep Reinforcement Learning". In: *IEEE Access* 8 (2020), pp. 135426–135442.

[6] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. "Target-driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning". In: *International Conference on Robotics and Automation*. IEEE. 2017, pp. 3357–3364.

[7] Y. Jiang, F. Yang, S. Zhang, and P. Stone. "Task-motion Planning with Reinforcement Learning for Adaptable Mobile Service Robots". In: *International Conference on Intelligent Robots and Systems*. IEEE. 2019, pp. 7529–7534.

[8] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson. "RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback". In: *International Conference on Machine Learning*. PMLR. 2024, pp. 51484–51501.

[9] S. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti. "RoboClip: One Demonstration is Enough to Learn Robot Policies". In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 55681–55693.

[10] P. Mahmoudieh, D. Pathak, and T. Darrell. "Zero-shot Reward Specification via Grounded Natural Language". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 14743–14752.

[11] J. Rocamonde, V. Montesinos, E. Nava, E. Perez, and D. Lindner. "Vision-Language Models are Zero-Shot Reward Models for Reinforcement Learning". In: *International Conference on Learning Representations*.

[12] A. Adeniji, A. Xie, C. Sferrazza, Y. Seo, S. James, and P. Abbeel. "Language Reward Modulation for Pretraining Reinforcement Learning". In: *Workshop on Training Agents with Foundation Models at RLC*.

[13] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar. "Eureka: Human-Level Reward Design via Coding Large Language Models". In: *International Conference on Learning Representations*.

[14] H. Huang, X. Chen, Y. Chen, H. Li, X. Han, Z. Wang, T. Wang, J. Pang, and Z. Zhao. "RoboGround: Robotic Manipulation with Grounded Vision-Language Priors". In: *Computer Vision and Pattern Recognition Conference*. 2025, pp. 22540–22550.

[15] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, S. Kirmani, B. Zitkovich, F. Xia, et al. "Open-world Object Manipulation Using Pre-trained Vision-language Models". In: *arXiv preprint arXiv:2303.00905* (2023).

[16] F. Liu, K. Fang, P. Abbeel, and S. Levine. "MOKA: Open-world Robotic Manipulation through Mark-based Visual Prompting". In: *arXiv preprint arXiv:2403.03174* (2024).

[17] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. "Reinforcement Learning with Unsupervised Auxiliary Tasks". In: *International Conference on Learning Representations*. 2017.

[18] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, et al. "Learning to Navigate in Complex Environments". In: *International Conference on Learning Representations*. 2017.

[19] D. Rolnick, A. Veit, S. Belongie, and N. Shavit. "Deep Learning is Robust to Massive Label Noise". In: *arXiv preprint arXiv:1705.10694* (2017).

[20] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. "Learning Latent Dynamics for Planning from Pixels". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2555–2565.

[21] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine. "Visual Foresight: Model-based Deep Reinforcement Learning for Vision-based Robotic Control". In: *arXiv preprint arXiv:1812.00568* (2018).

[22] T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker, et al. "Model-based Reinforcement Learning: A Survey". In: *Foundations and Trends® in Machine Learning* 16.1 (2023), pp. 1–118.

[23] M. Laskin, A. Srinivas, and P. Abbeel. "CURL: Contrastive Unsupervised Representations for Reinforcement Learning". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5639–5650.

[24] D. Yarats, I. Kostrikov, and R. Fergus. "Image Augmentation is All You Need: Regularizing Deep Reinforcement Learning from Pixels". In: *International Conference on Learning Representations*.

[25] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. "Mastering Visual Continuous Control: Improved Data-Augmented Reinforcement Learning". In: *International Conference on Learning Representations*.

[26] K. Gmelin, S. Bahl, R. Mendonca, and D. Pathak. "Efficient RL via Disentangled Environment and Agent Representations". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 11525–11545.

[27] A. Pore, R. Muradore, and D. Dall'Alba. "DEAR: Disentangled Environment and Agent Representations for Reinforcement Learning without Reconstruction". In: *International Conference on Intelligent Robots and Systems*. IEEE. 2024, pp. 650–655.

[28] R. Zabounidis, J. Campbell, S. Stepputtis, D. Hughes, and K. P. Sycara. "Concept Learning for Interpretable Multi-agent Reinforcement Learning". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1828–1837.

[29] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. "Concept Bottleneck Models". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 5338–5348.

[30] Q. Delfosse, S. Sztwiertnia, M. Rothermel, W. Stammer, and K. Kersting. "Interpretable Concept Bottlenecks to Align Reinforcement Learning Agents". In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 66826–66855.

[31] Z. Ye, S. Milani, G. J. Gordon, and F. Fang. "LICORICE: Label-Efficient Concept-Based Interpretable Reinforcement Learning". In: *International Conference on Learning Representations*. 2025.

[32] P. Ladosz, L. Weng, M. Kim, and H. Oh. "Exploration in Deep Reinforcement Learning: A Survey". In: *Information Fusion* 85 (2022), pp. 1–22.

[33] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. "Curiosity-driven Exploration by Self-supervised Prediction". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2778–2787.

[34] Y. Burda, H. Edwards, A. Storkey, and O. Klimov. "Exploration by Random Network Distillation". In: *International Conference on Learning Representations*.

[35] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. "VIME: Variational Information Maximizing Exploration". In: *Advances in Neural Information Processing Systems* 29 (2016).

[36] M. Espinosa Zarlenga, P. Barbiero, G. Ciravegna, G. Marra, F. Giannini, M. Diligenti, Z. Shams, F. Precioso, S. Melacci, A. Weller, et al. "Concept Embedding Models: Beyond the Accuracy-explainability Trade-off". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 21400–21413.

[37] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL].

[38] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. "Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning". In: *Conference on Robot Learning*. PMLR. 2020, pp. 1025–1037.

[39] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu. *robosuite: A Modular Simulation Framework and Benchmark for Robot Learning*. 2025. arXiv: 2009.12293 [cs.RO].

[40] E. Todorov, T. Erez, and Y. Tassa. "MuJoCo: A Physics Engine for Model-based Control". In: *International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 5026–5033.

[41] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. *Grounded SAM: Assembling Open-world Models for Diverse Visual Tasks*. 2024. arXiv: 2401.14159 [cs.CV].

[42] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. "Grounding DINO: Marrying DINO with Grounded Pre-training for Open-set Object Detection". In: *European Conference on Computer Vision*. Springer. 2024, pp. 38–55.

[43] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. “SAM 2: Segment Anything in Images and Videos”. In: *International Conference on Learning Representations*.