Pre-trained Text-to-Image Diffusion Models Are Versatile Representation Learners for Control

Gunshi Gupta*[1] and Karmesh Yadav*[2]
Yarin Gal[1] and Dhruv Batra[2] and Zsolt Kira[2] and Cong Lu[1] and Tim G. J. Rudner[3]

***Abstract*— Vision- and language-guided embodied AI requires a fine-grained understanding of the physical world through language and visual inputs. Such capabilities are difficult to learn solely from task-specific data, which has led to the emergence of pre-trained vision-language models as a tool for transferring representations learned from internet-scale data to downstream tasks and new domains. However, commonly used contrastively trained representations such as in CLIP have been shown to fail at enabling embodied agents to gain a sufficiently fine-grained scene understanding—a capability vital for control. To address this shortcoming, we consider representations from pre-trained text-to-image diffusion models, which are explicitly optimized to generate images from text prompts and as such, contain text-conditioned representations that reflect highly fine-grained visuo-spatial information. Using pre-trained text-to-image diffusion models, we construct *Stable Control Representations* which allow learning downstream control policies that generalize to complex, open-ended environments. We show that policies learned using Stable Control Representations are competitive with state-of-the-art representation learning approaches across a broad range of simulated control settings, encompassing challenging manipulation and navigation tasks. Most notably, we show that Stable Control Representations enable learning policies that exhibit state-of-the-art performance on OVMM, a difficult open-vocabulary navigation benchmark.**

I. Introduction

In this paper, we propose **Stable Control Representations (SCR)**: pre-trained vision-language representations from text-to-image diffusion models that can capture both high and low-level details of a scene [1], [2]. While diffusion representations have seen success in downstream vision-language tasks, for example, in semantic segmentation [3], [4], [5], they have, to date, not been used for control. We perform a careful empirical analysis in which we deconstruct pre-trained vision-language representations from text-to-image diffusion models to understand the effect of different design decisions.

In our empirical investigation, we find that—despite not being trained for representation learning—diffusion representations can outperform general-purpose models like CLIP [6] across a wide variety of embodied control tasks. This is the case even for purely vision-based tasks and settings that require task understanding through text prompts. A highlight of our results is the finding that diffusion model representations enable better generalization to unseen object categories in a challenging open-vocabulary navigation benchmark [7] and provide improved interpretability through attention maps [8].

*Equal Contribution
[1]University of Oxford [2]Georgia Tech [3]New York University
Corresponding email: gunshi.gupta@cs.ox.ac.uk, kyadav32@gatech.edu

Our key contributions are as follows:

1) In Section III, we introduce a multi-step approach for extracting vision-language representations for control from text-to-image diffusion models. We show that these representations are capable of capturing both the abstract high-level and fundamental low-level details of a scene, offering an alternative to models trained specifically for representation learning.
2) In Section IV, we evaluate the representation learning capabilities of diffusion models on a broad range of embodied control tasks, ranging from purely vision-based tasks to problems that require an understanding of tasks through text prompts, thereby showcasing the versatility of diffusion model representation.
3) In Appendix I-G, we systematically deconstruct the key features of diffusion model representations for control, elucidating different aspects of the representation design space, such as the input selection, the aggregation of intermediate features, and the impact of fine-tuning on enhancing performance.

We have demonstrated that diffusion models are versatile representation learners for control and can help drive progress in embodied AI. The code for our experiments can be accessed at `https://github.com/ykarmesh/stable-control-representations`

II. Background

We briefly review diffusion models and text-conditional image generation, and then describe the control setting we consider in this work.

A. Diffusion Models

Diffusion models [9], [10] are a class of generative models that learn to iteratively reverse a forward noising process and generate samples from a target data distribution $p(\boldsymbol{x}_0)$, starting from pure noise. Given $p(\boldsymbol{x}_0)$ and a set of noise levels σ_t for $t = 1, \ldots, T$, a denoising function $\epsilon_\theta(\boldsymbol{x}_t, t)$ is trained on the objective

$$\begin{aligned}\mathcal{L}_{\mathrm{DM}}(\theta) &= \mathbb{E}_{\boldsymbol{x}_0,\epsilon,t}[\|\epsilon - \epsilon_\theta\left(\boldsymbol{x}_t, t)\right)\|_2^2] \\ &= \mathbb{E}_{\boldsymbol{x}_0,\epsilon,t}[\|\epsilon - \epsilon_\theta\left(\boldsymbol{x}_0 + \sigma_t \cdot \epsilon, t)\right)\|_2^2]\,, \end{aligned} \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, 1)$, $t \sim \mathrm{Unif}(1, T)$, and $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$. To generate a sample $\boldsymbol{x}_0$ during inference, we first sample an initial noise vector $\boldsymbol{x}_T \sim \mathcal{N}(0, \sigma_T)$ and then iteratively denoise this sample for $t = T, ..., 1$ by sampling from $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, which is a function of $\epsilon_\theta(\boldsymbol{x}_t, t)$.

In some settings, we may want to generate samples with a particular property. For example, we may wish to draw

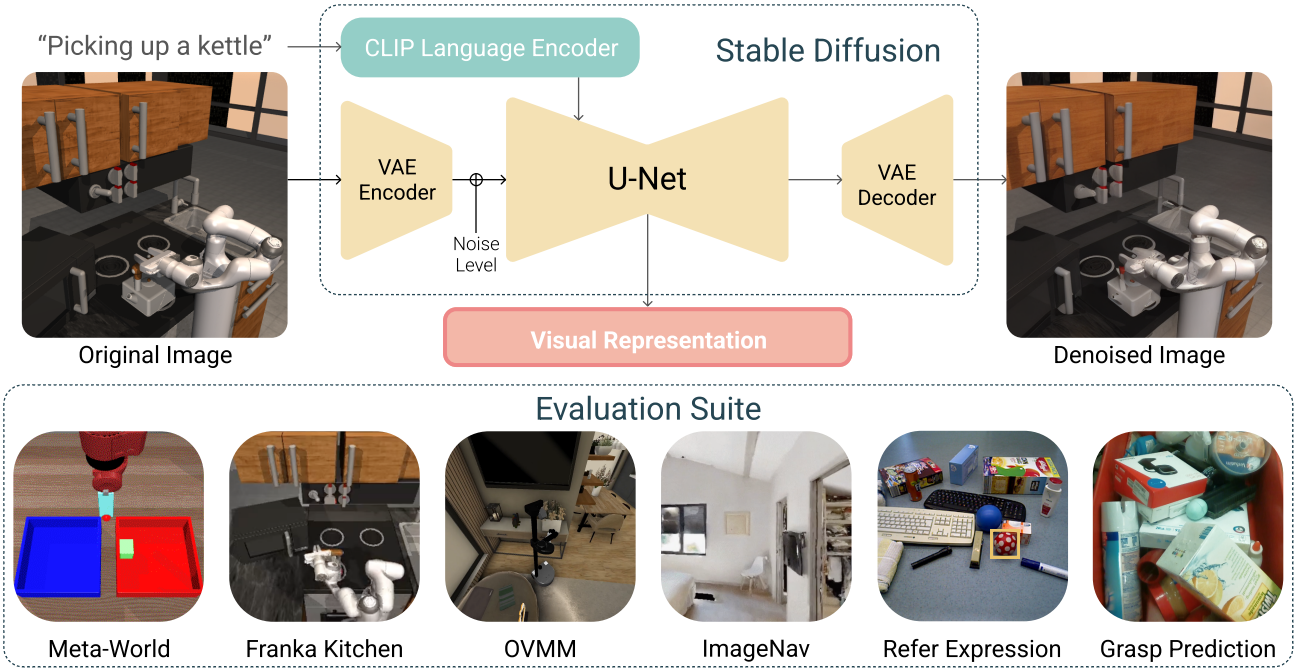

Fig. 1: **Top:** Our paper proposes Stable Control Representations, which uses pre-trained text-to-image diffusion models as a source of language-guided visual representations for downstream policy learning. **Bottom:** Stable Control Representations enable learning control policies that achieve all-round competitive performance on a wide range of embodied control tasks, including in domains that require open-vocabulary generalization. Empirical results are provided in Section IV.

samples from a conditional distribution over data points, $p(\boldsymbol{x}_0|c)$, where c captures some property of the sample, such as classification label or a text description [1], [11]. In these settings, we may additionally train with labels to obtain a conditioned denoiser $\epsilon_\theta(\boldsymbol{x}_t, t, c)$ and generate samples using classifier-free guidance [12].

B. Latent Diffusion Models

Latent diffusion models [1] reduce the computational cost of applying diffusion models to high-dimensional data by instead diffusing low-dimensional representations of high-dimensional data. Given an encoder $\mathcal{E}(\cdot)$ and decoder $\mathcal{D}(\cdot)$, Equation (1) is modified to operate on latent representations, $\boldsymbol{z}_0 \doteq \mathcal{E}(\boldsymbol{x}_0)$, yielding

$$\mathcal{L}_{\text{LDM}}(\theta) = \mathbb{E}_{\boldsymbol{x}_0, c, \epsilon, t}[\|\epsilon - \epsilon_\theta\left(\mathcal{E}(\boldsymbol{x}_0) + \sigma_t \cdot \epsilon, t, c\right)\|_2^2], \quad (2)$$

where $\epsilon \sim \mathcal{N}(0,1)$, $t \sim \text{Unif}(1,T)$, $\boldsymbol{x}_0, c \sim p(\boldsymbol{x}_0, c)$. After generating a denoised latent representation $\mathbf{z}_0$, it can be decoded as $\boldsymbol{x}_0 = \mathcal{D}(\boldsymbol{z}_0)$.

A popular instantiation of a conditioned latent diffusion model is the text-to-image Stable Diffusion model [1]. The SD model is trained on the LAION-2B dataset [13] and operates in the latent space of a pre-trained VQ-VAE image encoder [14]. The model architecture is shown at the top of Figure 1 and is based on a U-Net [15], with the corresponding conditioning text prompts encoded using CLIP's [6] language encoder.

C. Policy Learning for Control

We model our environments as Markov Decision Processes (MDP, [16]), defined as a tuple $M = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces respectively, $P(s'|s,a)$ the transition dynamics, $R(s,a)$ the reward function, and $\gamma \in (0,1)$ the discount factor. Our goal is to optimize a policy $\pi(a|s)$ that maximizes the expected discounted return $\mathbb{E}_{\pi,P}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

In this paper, we consider visual control tasks that may be language-conditioned, that is, states are given by $s = [s_{\text{image}}, s_{\text{text}}]$, where s_{text} specifies the task. We are interested in pre-trained vision-language representations capable of encoding the state s as $f_\phi(s_{\text{image}}, s_{\text{text}})$. This encoded state is then supplied to a downstream, task-specific policy network, which is trained to predict the action a_t. Our evaluation encompasses both supervised learning and reinforcement learning regimes for training the downstream policies. We train agents through behavior cloning on a small set of demonstrations for the few-shot manipulation tasks we study in appendix I-B.2. For the indoor navigation tasks we study in Secs. IV-A and IV-B, we use a version of the Proximal Policy Optimisation [17] algorithm for reinforcement learning.

III. Stable Control Representations

In this paper, we consider extracting language-guided visual representations from the open-source Stable Diffusion model. We follow a similar protocol as [5], [18], and [19]: Given an image-text prompt, $s = \{s_{\text{image}}, s_{\text{text}}\}$, associated with a particular task, we use the SD VQ-VAE model as the encoder $\mathcal{E}(\cdot)$ and partially noise the latents $\boldsymbol{z}_0 \doteq \mathcal{E}(s_{\text{image}})$ to some diffusion timestep t. We then extract representations from the intermediate outputs of the denoiser $\epsilon_\theta(\boldsymbol{z}_t, t, s_{\text{text}})$. This process is illustrated in Figure 2. We refer to the extracted representations as **Stable Control Representations (SCR)**. We will describe the design space for extracting SCR in the remainder of this section.

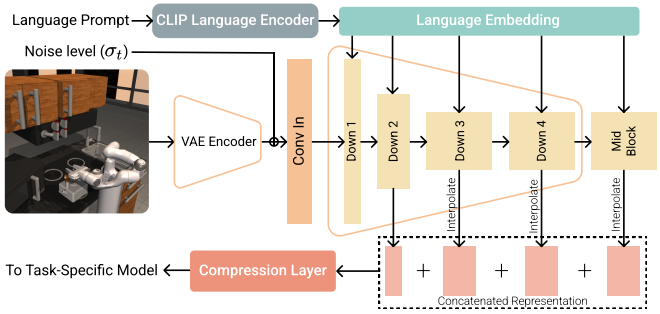

Fig. 2: Extraction of Stable Control Representations from Stable Diffusion. Given an image-text prompt, $s = \{s_{\text{image}}, s_{\text{text}}\}$, we encode and noise the image and feed it into the U-Net together with the language prompt. We may then aggregate features from multiple levels of the downsampling process, as described in Section III.

A. Layer Selection and Aggregation

We are interested in evaluating the internal representations from the denoiser network, that is, the U-Net $\epsilon_\theta(\cdot)$. The first design choice we consider is which layers of ϵ_θ to aggregate intermediate outputs from. The U-Net does not have a representational bottleneck, and different layers potentially encode different levels of detail. Trading off size with fidelity, we concatenate the feature maps output from the mid and down-sampling blocks to construct the representation. This results in a representation size comparable to that of the other pre-trained models we study in Section IV. This is shown at the bottom of Figure 2 and we ablate this choice in Appendix I-G.1. Since outputs from different layers may have different spatial dimensions, we bilinearly interpolate them so that they are of a common spatial dimension and can be stacked together. We then pass them through a learnable convolutional layer to reduce the channel dimension before feeding them to downstream policies. The method used to spatially aggregate pre-trained representations can significantly affect their efficacy in downstream tasks, as we will discuss in Appendix I-F. We use the best-performing spatial aggregation method for all the baselines that we re-train in Section IV.

B. Diffusion Timestep Selection

Next, we consider the choice of extraction timestep t for the denoising network (shown on the left of Figure 2). Recall that the images we observe in control tasks are un-noised (i.e., corresponding to $\boldsymbol{x}_0$), whereas the SD U-Net expects noised latents, corresponding to $\boldsymbol{z}_t$ for $t \in [0, 1000]$. The choice of timestep t influences the fidelity of the encoded latents since a higher value means more noising of the inputs. [19] have observed that there are task-dependent optimal timesteps and proposed adaptive selection of t during training, while [20] have used $t = 0$ to extract representations from un-noised inputs to do open-vocabulary segmentation. We hypothesize that control tasks that require a detailed spatial scene understanding benefit from fewer diffusion timesteps, corresponding to a later stage in the denoising process. We provide evidence consistent with this hypothesis in Appendix I-G.2. To illustrate the effect of the timestep, we display final denoised images for various t values in different domains in Figure 7.

C. Prompt Specification

Since text-to-image diffusion models allow conditioning on text, we investigate if we can influence the representations to be more task-specific via this conditioning mechanism. For tasks that come with a text specifier, for example, the sentence "go to object X", we simply encode this string and pass it to the U-Net. However, some tasks are purely vision-based and in these settings, we explore whether constructing reasonable text prompts affects downstream policy learning when using the U-Net's language-guided visual representations. We present this analysis in Appendix I-G.3.

D. Intermediate Attention Map Selection

Recent studies [5], [8] demonstrate that the Stable Diffusion model generates localized attention maps aligned with text during the combined processing of vision and language modalities. [5] leveraged these word-level attention maps to perform open-domain semantic segmentation. We hypothesize that these maps can also help downstream control policies to generalize to an open vocabulary of object categories by providing helpful intermediate outputs that are category-agnostic. Following [8], we extract the cross-attention maps between the visual features and the CLIP text embeddings within the U-Net. An example of the word-level attention maps is visualized in Figure 8. We test our hypothesis on an open-domain navigation task in Section IV-B, where we fuse the cross-attention maps with the extracted feature maps from the U-Net. We refer to this variant as **SCR-ATTN**.

E. Fine-Tuning on General Robotics Datasets

Finally, we consider fine-tuning strategies to better align the base Stable Diffusion model towards generating representations for control. This serves to bridge the domain gap between the diffusion model's training data (e.g., LAION images) and robotics datasets' visual inputs (e.g., egocentric tabletop views in manipulation tasks or indoor settings for navigation). Crucially, we do not use any task-specific data for fine-tuning. Instead, we use a small subset of the collection of datasets used by prior works on representation learning for embodied AI [21], [22]: we use subsets of the EpicKitchens [23], Something-Something-v2 [24], and the Bridge-v2 [25] datasets.

We adopt the same text-conditioned generation objective as that of the base model for the fine-tuning phase. As is standard, we fine-tune the denoiser U-Net ϵ_θ but not the VAE encoder or decoder. Image-text pairs are uniformly sampled from the video-text pairs present in these datasets. A possible limitation of this strategy is that text-video aligned pairs (a sequence of frames in a control task that correspond to a single language instruction) may define a many-to-one relation for image-text pairs. However, as we see in experiments in which we compare to the base Stable Diffusion model in Section IV, this simple approach to robotics alignment is useful in most cases. Further details related to fine-tuning

TABLE I: Average Success Rate and standard error evaluated across different representations.

(a) ImageNav

Model	Success
R3M	30.6
CLIP-B	52.2
VC-1	70.3
MVP	68.1
SD-VAE	46.6
SCR	**73.9**
SCR-FT	69.5

(b) OVMM

Model	Success
Oracle	77.6
Detic	36.7
CLIP	38.7 ± 1.7
VC-1	40.6 ± 2.2
SCR	38.7 ± 1.2
SCR-FT	**41.9 ± 1.0**
SCR-FT-ATTN	**43.6 ± 2.1**

are provided in Appendix I-I. We refer to the representations from this fine-tuned model as **SCR-FT**.

IV. Empirical Evaluation

In this work, we evaluate Stable Control Representations (SCR) on an extensive suite of tasks from 6 benchmarks covering few-shot imitation learning for manipulation in Appendix I-B.2, reinforcement learning-based indoor navigation in Sections IV-A and IV-B, and owing to space limitations, two tasks related to fine-grained visual prediction in Appendix I-B. Together, these tasks allow us to comprehensively evaluate whether our extracted representations can encode both high and low-level semantic understanding of a scene to aid downstream policy learning. In the following sections we will describe the individual task setups and results and defer the description of the baselines to Appendix I-B.1.

A. *Image-Goal Navigation*

We now assess SCR in more realistic visual environments, surpassing the simple table-top scenes in manipulation benchmarks. In these complex settings, the representations derived from pre-trained foundational models are particularly effective, benefiting from their large-scale training. We study Image-Goal Navigation (ImageNav), an indoor visual navigation task that evaluates an agent's ability to navigate to the viewpoint of a provided goal image [26]. The position reached by the agent must be within a 1-meter distance from the goal image's camera position. This requires the ability to differentiate between nearby or similar-looking views within a home environment. This task, along with the semantic object navigation task that we study in Section IV-B, allows for a comprehensive evaluation of a representation's ability to code both semantic and visual appearance-related features in completely novel evaluation environments.

We follow the protocol for the ImageNav task used by [21] and input the pre-trained representations to an LSTM-based policy trained with DD-PPO [27] for 500 million steps on 16 A40 GPUs (further details in Appendix I-K.3). Given the large training requirements, we only run SCR-FT and directly compare to the results provided in [21].

Results. We evaluate our agent on 4200 episodes in 14 held-out scenes from the Gibson dataset and report the success rate in Table Ia. We find that SCR outperforms MVP and CLIP (ViT-B), and is almost on par with VC-1 (69.5% vs 70.3%), the SOTA visual representation from prior work. We also see that R3M, the best model for few-shot manipulation from Table II performs very poorly (30.6%) in this domain, showing its limited transferability to navigation tasks.

B. *Open Vocabulary Mobile Manipulation*

We now shift our focus to evaluating how Stable Diffusion's web-scale training can enhance policy learning in open-ended domains. We consider the Open Vocabulary Mobile Manipulation (OVMM) benchmark [7] that requires an agent to find, pick up, and place objects in unfamiliar environments. One of the primary challenges here is locating previously unseen object categories in novel scenes (illustrated in Figure 6 (left)).

To manage this complex sparse-reward task, existing solutions [7] divide the problem into sub-tasks and design modular pipelines that use open-vocabulary object detectors such as Detic [28]. We study a modified version of the Gaze sub-task (detailed in Appendix I-K.2), which focuses on locating a specified object category for an abstracted grasping action. The task's success is measured by the agent's ability to precisely focus on the target object category. This category is provided as an input to the policy through its CLIP text encoder embedding. The evaluation environments cover both novel instances of object categories seen during policy learning, as well as entirely unseen categories. We compare to VC-1, the best model from Section IV-A and CLIP, since prior work has studied it for open-vocab navigation [29], [30]. We also incorporate a baseline that trains a policy with ground truth object masks, evaluated using either the ground truth or Detic-generated masks (labeled as Oracle/Detic).

Results. Table Ib shows SCR matches the performance of CLIP, while SCR-FT surpasses VC-1 by 1.3%, beating CLIP and SCR by 3.2%. Surprisingly, VC-1's visual representation does better than CLIP's image encoder representation, given that the downstream policy has to fuse these with the CLIP text embedding of the target object category. Compared to these baselines, we can see the benefit of providing intermediate outputs in the form of text-aligned attention maps to the downstream policy (+1.7%). These word-level cross-attention maps simultaneously improve policy performance and also aid explainability, allowing us to diagnose successes and failures. Samples of attention maps overlaid on evaluation episode images can be found in Appendix I-K.

Interestingly, the foundation model representations (CLIP, VC-1, SCR) perform better than Detic. While object detections serve as a category-agnostic output that downstream pick-and-place policies can work with, noisy detections can often lead to degraded downstream performance, as we see in this case. Nonetheless, there is still a sizeable gap to 'Oracle' which benefits from ground truth object masks.

V. Conclusion

In this paper, we proposed Stable Control Representations, a powerful method for leveraging general-purpose diffusion features for control. We showed that our extracted representations lead to strong performance across a wide variety of tasks. As such, we hope that SCR will help drive data-efficient control and enable open-vocabulary generalization in challenging domains; these capabilities will only improve as generative modeling advances. IF

REFERENCES

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[2] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, and T. Salimans, "Imagen video: High definition video generation with diffusion models," 2022.

[3] D. Baranchuk, A. Voynov, I. Rubachev, V. Khrulkov, and A. Babenko, "Label-efficient semantic segmentation with diffusion models," in *International Conference on Learning Representations*, 2022.

[4] J. Tian, L. Aggarwal, A. Colaco, Z. Kira, and M. Gonzalez-Franco, "Diffuse, attend, and segment: Unsupervised zero-shot segmentation using stable diffusion," *arXiv preprint arXiv:2308.12469*, 2023.

[5] J. Wang, X. Li, J. Zhang, Q. Xu, Q. Zhou, Q. Yu, L. Sheng, and D. Xu, "Diffusion Model is Secretly a Training-free Open Vocabulary Semantic Segmenter," *arXiv e-prints*, p. arXiv:2309.02773, Sep. 2023.

[6] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8748–8763.

[7] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023.

[8] R. Tang, L. Liu, A. Pandey, Z. Jiang, G. Yang, K. Kumar, P. Stenetorp, J. Lin, and F. Ture, "What the DAAM: Interpreting stable diffusion using cross attention," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[9] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. PMLR, 2015, pp. 2256–2265.

[10] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[11] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in Neural Information Processing Systems*, vol. 35, pp. 36 479–36 494, 2022.

[12] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[13] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, "LAION-5B: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 25 278–25 294, 2022.

[14] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 873–12 883.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*, ser. Lecture Notes in Computer Science, N. Navab, J. Hornegger, W. M. W. III, and A. F. Frangi, Eds., vol. 9351. Springer, 2015, pp. 234–241.

[16] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[17] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv e-prints*, p. arXiv:1707.06347, Jul. 2017.

[18] J. Traub, "Representation learning with diffusion models," *arXiv preprint arXiv:2210.11058*, 2022.

[19] X. Yang and X. Wang, "Diffusion model as representation learner," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 938–18 949.

[20] J. Xu, S. Liu, A. Vahdat, W. Byeon, X. Wang, and S. De Mello, "ODISE: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models," *arXiv preprint arXiv: 2303.04803*, 2023.

[21] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *arXiv preprint arXiv:2303.18240*, 2023.

[22] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv:2203.06173*, 2022.

[23] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "Scaling egocentric vision: The epic-kitchens dataset," in *European Conference on Computer Vision (ECCV)*, 2018.

[24] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5842–5850.

[25] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He *et al.*, "Bridgedata v2: A dataset for robot learning at scale," *arXiv preprint arXiv:2308.12952*, 2023.

[26] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. K. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning," *ICRA*, 2017.

[27] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames," *arXiv preprint arXiv:1911.00357*, 2019.

[28] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in *ECCV*, 2022.

[29] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[30] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "Zson: Zero-shot object-goal navigation using multimodal goal embeddings," *Advances in Neural Information Processing Systems*, vol. 35, pp. 32 340–32 352, 2022.

[31] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav," 2023.

[32] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," in *6th Annual Conference on Robot Learning*, 2022.

[33] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang, "Language-driven representation learning for robotics," in *Robotics: Science and Systems (RSS)*, 2023.

[34] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *arXiv preprint arXiv:2303.04137*, 2023.

[35] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann *et al.*, "Imitating human behaviour with diffusion models," *arXiv preprint arXiv:2301.10677*, 2023.

[36] P. Hansen-Estruch, I. Kostrikov, M. Janner, J. G. Kuba, and S. Levine, "Idql: Implicit q-learning as an actor-critic method with diffusion policies," *arXiv preprint arXiv:2304.10573*, 2023.

[37] I. Kapelyukh, V. Vosylius, and E. Johns, "Dall-e-bot: Introducing web-scale diffusion models to robotics," *IEEE Robotics and Automation Letters*, 2023.

[38] T. Yu, T. Xiao, A. Stone, J. Tompson, A. Brohan, S. Wang, J. Singh, C. Tan, J. Peralta, B. Ichter *et al.*, "Scaling robot learning with semantically imagined experience," *arXiv preprint arXiv:2302.11550*, 2023.

[39] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar, "Cacti: A framework for scalable multi-task multi-scene visual imitation learning," *arXiv preprint arXiv:2212.05711*, 2022.

[40] C. Lu, P. J. Ball, Y. W. Teh, and J. Parker-Holder, "Synthetic experience replay," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[41] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[42] A. Ajay, S. Han, Y. Du, S. Li, A. Gupta, T. Jaakkola, J. Tenenbaum, L. Kaelbling, A. Srivastava, and P. Agrawal, "Compositional foundation

models for hierarchical planning," *arXiv preprint arXiv:2309.08587*, 2023.

[43] Y. Du, M. Yang, B. Dai, H. Dai, O. Nachum, J. B. Tenenbaum, D. Schuurmans, and P. Abbeel, "Learning universal policies via text-guided video generation," *URL https://arxiv. org/abs/2302.00111*, 2023.

[44] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019.

[45] V. Kumar, R. Shah, G. Zhou, V. Moens, V. Caggiano, J. Vakil, A. Gupta, and A. Rajeswaran, "Robohive – a unified framework for robot learning," in *NeurIPS: Conference on Neural Information Processing Systems*, 2023.

[46] K.-J. Wang, Y.-H. Liu, H.-T. Su, J.-W. Wang, Y.-S. Wang, W. Hsu, and W.-C. Chen, "OCID-ref: A 3D robotic dataset with embodied language for clutter scene grounding," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 5333–5338.

[47] Y. J. Ma, W. Liang, V. Som, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "LIV: Language-Image Representations and Rewards for Robotic Control," *arXiv e-prints*, p. arXiv:2306.00958, Jun. 2023.

[48] J. Zhao, H. Zheng, C. Wang, L. Lan, and W. Yang, "Magicfusion: Boosting text-to-image generation performance by fusing diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 22 592–22 602.

[49] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[50] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva, "Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation," *arXiv preprint*, 2023.

[51] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[52] L. Mezghani, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. Alahari, "Memory-augmented reinforcement learning for image-goal navigation," *arXiv preprint arXiv:2101.05181*, 2021.

[53] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9068–9079.

APPENDIX I

A. Related Work

In this section, we review prior work on representation learning and orthogonal work on diffusion models for control.

Representation Learning with Diffusion Models. Diffusion models have received a lot of recent attention as flexible representation learners for computer vision tasks of varying granularity—ranging from key point detection and segmentation [4], [5] to image classification [19], [18]. [5] has shown that intermediate layers of a text-to-image diffusion model encode semantics and depth maps that are recoverable by training probes. These approaches extract representations in a similar manner to us by considering a moderately noised input, and find that the choice of timestep can vary based on the granularity of prediction required for the task. [19] train a policy to select an optimal diffusion timestep, we simply used a fixed timestep per class of task. Several works [4], [5], [8] observed that the cross-attention layers that attend over the text and image embeddings encode a lot of the spatial layout associated with an image and therefore focus their method around tuning, post-processing, or extracting information embedded within these layers.

Visual Representation Learning for Control. Over the past decade, pretrained representation learning approaches have been scaled for visual discrimination tasks first, and control tasks more recently. Contrastively pretrained CLIP [6] representations were employed for embodied navigation tasks by [29]. MAE representations have been used for control tasks by VC-1 [21], MVP [22] and OVRL-v2 [31]. R3M [32] and Voltron [33] leverage language supervision to learn visual representations. In contrast, we investigate if powerful text-to-image diffusion models trained for image generation can provide effective representations for control.

Diffusion Models for Control. Diffusion models have seen a wide range of uses in control aside from learning representations. These can broadly be categorized into three areas. First, there are works that treat diffusion models as a class of expressive models for learning action distribution for policies [34], [35], [36]; this can often help model multimodality and richer action distributions than Gaussians.

Next, off-the-shelf diffusion models have been used to augment limited robot demonstration datasets by specifying randomizations for object categories seen in the data through inpainting [37], [38], [39]. Diffusion models trained from scratch have also been shown to be an effective method for data augmentation [40].

Finally, there are works that cast planning as sequence modeling through diffusion models [41], [42], [43]. Diffuser [41] proposes to view RL as a sequence modeling problem where a diffusion model can be trained to predict trajectories of interleaved state-actions given a behavior dataset. These state-actions can then be guided towards higher value and then used in a receding horizon control setup.

B. Extended Empirical Evaluation Details

1) ***Baselines:*** We compare SCR and its variants (i.e., SCR-FT and SCR-FT-ATTN) to the following prior work in representation learning for control:

- **R3M** [32] pretrains a ResNet50 encoder on video-language pairs from the Ego4D dataset using time-contrastive video-language alignment learning.
- **MVP** [22] and **VC-1** [21] both pretrain ViT-B/L models with the masked auto-encoding (MAE) objective on egocentric data from Ego4D, Epic-Kitchens, Something-Something-v2 [24, SS-v2] and ImageNet, with VC-1 additionally pretraining on indoor navigation videos.
- **CLIP** [6] trains text and ViT-based image encoders using contrastive learning on web-scale data.
- **Voltron** [33] is a language-driven representation learning method that involves pretraining a ViT-B using MAE and video-captioning objectives on aligned text-video pairs from SS-v2.
- **SD-VAE** [1] is the base VAE encoder used by SD to encode images into latents.

To assess how well the vision-only methods would do on tasks with language specification, we concatenate their visual representations with the CLIP text embeddings of the language prompts. While we are limited by the architecture designs of the released models we are studying, to ensure a more fair comparison we try to match parameter counts as much as we can. We use the ViT-Large (307M parameters) versions of CLIP, MVP, and VC-1 since extracting SCR involves a forward pass through 500M parameters.

2) ***Few-shot Imitation Learning:*** We start by evaluating SCR on commonly studied representation learning benchmarks in few-shot imitation learning. Specifically, our investigation incorporates five commonly studied tasks from Meta-World [44] (same as CORTEXBENCH [21]), which includes bin picking, assembly, pick-place, drawer opening, and hammer usage; as well as five tasks from the Franka-Kitchen environments included in the RoboHive suite [45], which entail tasks such as turning a knob or opening a door. We adhere to the training and evaluation protocols adopted in their respective prior works to ensure our results are directly comparable (detailed further in Appendix I-K.1).

TABLE II: Meta-World & Franka-Kitchen.

Model	Meta-World	Franka-Kitchen
R3M	**96.0 ± 1.1**	**57.6 ± 3.3**
CLIP	90.1 ± 3.6	36.3 ± 3.2
VC-1	92.3 ± 2.5	47.5 ± 3.4
Voltron	72.5 ± 5.2	33.5 ± 3.2
SD-VAE	75.5 ± 5.2	43.7 ± 3.1
SCR	**94.4 ± 1.9**	45.0 ± 3.3
SCR-FT	**94.9 ± 2.0**	49.9 ± 3.4

Results. We report the best results of SCR and baselines in Table II. On Meta-World, we see that SCR outperforms most prior works, achieving 94.9% success rate. In comparison, VC-1, the visual foundation model for embodied AI and CLIP achieved 92.3 and 90.1% respectively. On Franka-Kitchen, SCR obtains 49.9% success rate, which is much higher than CLIP (36.3%) and again outperforms all other baselines except for R3M. We note that R3M's sparse representations excel in few-shot manipulation with limited demos but

struggle to transfer beyond this setting [21], [33].

We see that while the SD-VAE encoder performs competitively on Franka-Kitchen, it achieves a low success rate on Meta-World. This observation allows us to gauge the improved performance of SCR from the base performance gain we may get just from operating in the latent space of this VAE. Additionally, we see that the task-agnostic fine-tuning gives SCR-FT an advantage (4%) over SCR on Franka-Kitchen while making no difference on Meta-World. Note that the other high-performing baselines (R3M and Voltron) have been developed for downstream control usage with training objectives that take temporal information into account, while VC-1 has been trained on a diverse curation of robotics-relevant data. In this context, SCR's comparable performance shows that generative foundation models hold promise for providing useful representations for control, even with relatively minimal fine-tuning on non-task-specific data.

*3) **Referring Expressions Grounding**:* In appendix I-B.2 and Secs. IV-A and IV-B, we analyzed the performance of diverse representations across a range of control tasks. We now turn our attention to two specific tasks: referring expressions grounding and grasp affordance prediction. These tasks, involving fine-grained visual prediction, have been previously examined by [33] as proxy measures to evaluate the efficacy of representations for control applications.

Here, we revisit the Referring Expressions Grounding task first introduced in appendix I-G.3 and compare to other baselines. This task requires the identification and bounding box prediction of an object in an image based on its textual description. Similar to [33], we use the OCID-Ref Dataset [46] for our experiments. All models output a frozen visual representation which is concatenated with a text embedding and passed to a shallow MLP, which predicts the bounding box coordinates. We evaluate bounding box accuracy at a 25% Intersection-over-Union (IoU) threshold across different scene clutter levels. The IoU metric characterizes the degree of overlap between labels and predictions and in this case, a minimum 25% IoU is used to decide if a prediction should be marked as correct or incorrect. We train SCR-variants along with baselines from [33] for 10 epochs with batch size 128 and lr=$1e^{-3}$. The results are presented in Tab. III.

TABLE III: Referring Expression Grounding (Accuracy at IoU 0.25).

Model	Average	Maximum clutter	Medium clutter	Minimum clutter
CLIP	68.1	60.3	76.6	67.0
CLIP	94.3	92.5	95.1	92.8
R3M	63.3	55.3	68.3	63.3
Voltron	92.5	96.9	91.8	90.2
VC-1	94.6	93.7	96.5	93.7
SD-VAE	94.3	93.2	96.3	93.4
SCR	92.9	91.1	95.9	91.8
SCR-FT	91.8	90.1	94.8	90.8

Results. We see that SCR is tied with Voltron and that VC-1 and SD-VAE perform the best with a 1.5% lead. The better performance of these vision-encoder only methods highlights that on this task, it is not a challenge for the downstream decoder to learn to associate the visual embeddings with the CLIP text encoder's output for the language specification. Since the training budget is fixed, we observed that many methods were not close to complete convergence and could potentially improve over extended training. However, we were primarily interested in this task not to compare the downstream visual prediction performance, but to use it as a testbed for exploring two questions. Specifically, we were interested in evaluating whether the performance differences between the representations we evaluated in Sec. IV stem from the absence of fine-grained spatial information encoded within the representations. We refute this claim in the following section, where we present the impact of a representation's spatial aggregation method on task performance in appendix I-F. This will explain the poor performance of CLIP as reported in [33] as well as our improved result for CLIP in gray in the table. Additionally, recall that we used this task to explore the extent to which language prompting influences the representations from SCR in appendix I-G.3.

TABLE IV: Grasp Affordance Prediction: Precision on pixels corresponding to positive graspability at varying probability threshold levels.

Model	Top99	Top95	Top90
CLIP	60.3	45.0	28.6
CLIP	72.9	55.9	36.5
Voltron	62.5	42.8	32.1
SD-VAE	55.6	41.3	33.8
SCR	72.9	55.9	54.5
SCR-FT	72.3	54.6	44.4

*4) **Grasp Affordance Prediction**:* The **Grasp Affordance Prediction** task requires segmenting areas of object in an RGB image, that would be amenable to grasping by a suction gripper. The evaluation metric adopted in prior work is the precision of predictions corresponding to positive graspability at varying confidence levels (90, 95, and 99th percentile of the predicted per-pixel probabilities, denoted as Top90, Top95, and Top99 in Tab. IV). We refer the reader to [33] for the complete task setup details. We re-ran all the methods using the evaluation repository provided with the work, and obtained slightly different results compared to the reported numbers in [33], possibly attributed to a bug we fixed related to metrics computation. The evaluation procedure for this task adopted in prior work involves a 5-fold cross-validation, and we observed a high variability in the results, with different runs of 5-fold cross-validation yielding different final test metrics.

Our findings highlight that SCR and our adaptation of CLIP both excel in this task, achieving a Top99 score of 72.9. The following section will further elaborate on our modifications to CLIP. Interestingly, we found that finetuning did not enhance performance on the visual prediction tasks explored (appendices I-B.3 and I-B.4), suggesting a potential disconnect from control task benchmarks.

C. Fine-tuning CLIP

We follow the same experimental constraints that we took into account while fine-tuning the diffusion model to get SCR-

TABLE V: Performance on Franka-Kitchen after fine-tuning CLIP.

Model	Franka-Kitchen
CLIP	36.9 ± 3.2
CLIP (FT)	34.2 ± 2.9

TABLE VI: Comparing to LIV on manipulation and navigation tasks.

Model	Franka-Kitchen	OVMM
SCR	45.0	38.7
SCR-FT	49.9	41.9
LIV	54.2	8.4

FT: we trained it on the same text-image pairs from the same datasets, and using CLIP's contrastive loss to bring the visual embedding of the middle frames of a video closer to the video caption's text embedding. Specifically, for our experiment, we use the huggingface CLIP finetuning implementation and train the model with a batch size of 384 (the maximum number of samples we were able to fit on 8 A40 GPUs) with a learning rate of 5e-5 and a weight decay of 0.001 for 5000 update steps (same as SR-FT). We present the results in Table V for Franka-Kitchen, and note the lack of improvement on the task post-fine-tuning.

D. Comparison with LIV

We include a comparison with LIV [47] on two tasks that involve manipulation and navigation. LIV is a vision-language representation learned through contrastive learning on the EpicKitchens dataset [23]. Similar to R3M results in the main paper, this representation does well on manipulation tasks but poorly on navigation tasks.

E. Overall Ranking of Representations

In Table VII, we present the consolidated scores across the four control benchmarks we study in Section IV, for all the representations we evaluate in this work. This is to give a higher-level view of the all-round performance of the different representations on the diverse set of tasks we consider. We see that VC-1, SCR, and SCR-FT emerge as the top three visual representations overall. While VC-1 is a representation-learning foundation model trained specifically for robotics tasks, SCR and SCR-FT are the diffusion model representations that we study in this paper, confirming the potential of large pre-trained foundation generative models across a wide array of downstream robotics tasks.

F. Method of Spatial Aggregation Matters

We revise the representation extraction approach used in previous studies by incorporating a convolutional layer to downsample the spatial grid of pretrained representations, to effectively preserve the spatial information. This adjustment, described as a "compression layer" by [31], aims to reduce the high channel dimension of pretrained model outputs without losing spatial details, facilitating more effective input processing by task-specific decoders like MLPs.

We show the performance gains achieved by this modification, by replacing the multi-headed attention pooling done for CLIP embeddings with a convolutional downsampling layer in appendix I-B.3. This significantly enhances performance in bounding box prediction tasks (an improvement from 68% to 94%, as reported in the grayed-out CLIP result in Tab. III). We present a similar modification and finding for the grasp affordance prediction results for CLIP in gray in Tab. IV. This finding contradicts previous claims by [33] regarding CLIP's inability for low-level spatial predictions, underscoring the importance of representation adaptation.

Although incorporating the compression layer also slightly improves CLIP's performance in control tasks (by 1-2%), it does not enable it to surpass the best-performing models. In the main paper, we used the compression layer method of aggregation for all the baselines we ran to ensure we compared to their best numbers (Tabs. Ia, Ib and II). We also present ablations over the spatial aggregation method for VC-1 and CLIP on the MuJoCo tasks in Tab. VIII, to showcase the slight improvement that using the compression layer brings across both tasks. We recommend future work to adopt this methodology where applicable to compare representations more fairly.

TABLE VII: Representation Performance Comparison: Numbers in the task columns (OVMM, ImageNav, MetaWorld, Franka Kitchen) indicate relative scores of different representations (normalized by the highest score on that task), and the average normalized score column indicates the averaged scores across the task-wise relative scores where numbers are available.

Method	OVMM	ImageNav	MetaWorld	Franka	Avg Score
VAE	-	0.629	0.786	0.759	0.725
R3M	-	0.414	1.000	1.000	0.805
VC-1	0.969	0.951	0.961	0.825	0.927
CLIP	0.924	0.706	0.939	0.630	0.800
SR	0.924	1.000	0.983	0.781	0.922
SR-FT	1.000	0.942	0.989	0.866	**0.949**

TABLE VIII: Ablating the spatial aggregation method (CLS token embedding versus using the compression layer) for CLIP and VC-1 representations on MuJoCo Tasks: Average success rate and std. error on Meta-World & Franka Kitchen.

Model	Meta-World	Franka Kitchen
VC-1 (CLS)	88.8 ± 2.2	52.0 ± 3.4
VC-1 (Compression)	92.3 ± 2.5	47.5 ± 3.4
CLIP (CLS)	88.8 ± 3.9	35.3 ± 3.4
CLIP (Compression)	90.1 ± 3.6	36.3 ± 3.2

G. Deconstructing Stable Control Representations

In this section, we aim to deconstruct which design choices from Sec. III were most crucial for SCR's strong performance and assess our representation's robustness to each.

1) Layer Selection: We begin our investigation by examining how the performance of SCR is influenced by the selection of layers from which we extract feature maps. We had previously chosen to utilize outputs from the mid and downsampling layers of the U-Net (Fig. 2), because their aggregate size closely matches the representation sizes from ViT-based models such as VC-1, MVP, and CLIP. Appendix I-

TABLE IX: Ablations of the denoising timestep and layers chosen for representation extraction for SR on the Franka Kitchen benchmark. Numbers indicate mean ± standard error over 3 seeds.

Timestep	Layers	Success Rate
0	Down[1-3]	43.0 ± 3.4
0	Down[1-3] + Mid	**49.9 ± 3.4**
0	Mid	41.6 ± 3.3
0	Mid, Up[0]	42.1 ± 3.6
0	Down[1-3] + Mid	**49.9 ± 3.4**
10	Down[1-3] + Mid	**48.2 ± 3.1**
100	Down[1-3] + Mid	42.0 ± 3.7
110	Down[1-3] + Mid	42.0 ± 3.4
200	Down[1-3] + Mid	35.1 ± 3.2

J lists out the exact feature map sizes used for all the models we study.

Table IX (top) lists the success rates achieved on the Franka-Kitchen domain when we use different sets of block outputs in SCR. We see that utilizing outputs from multiple layers is instrumental to SCR's high performance. This finding underscores a broader principle applicable to the design of representations across different models: leveraging a richer set of features from multi-layer outputs should enhance performance on downstream tasks. However, it's important to acknowledge the practical challenges in applying this strategy to ViT-based models. The high dimensionality of each layer's patch-wise embeddings (16x16x1024 for ViT-L for images of size 224x224), may complicate the integration of multi-layer outputs.

2) Sensitivity to the Noising Timestep: Next, we characterize the sensitivity of task performance to the denoising step values chosen during representation extraction on the Franka-Kitchen tasks in Tab. IX (bottom). We see that the performance across nearby timesteps (0 and 10 or 100 and 110) is similar, and that there is a benefit to doing a coarse grid search up to a reasonable noising level (0 vs 100 vs 200) to get the best value for a given task.

3) How is language guiding the representations?: Recall that in our experiments on OVMM (Sec. IV-B), we concatenated the target object's CLIP text embedding to the visual representations before feeding it to the policy. For SCR and SCR-FT, we also provided the text as the prompt to the U-Net, and additionally extracted the generated cross-attention maps for SCR-FT-ATTN. In this subsection, we seek to more closely understand how the text prompts impact the generated representations in SCR.

We start with the Franka-Kitchen setup from appendix I-B.2, which includes manipulation tasks that do not originally come with a language specification. We experiment with providing variations of task relevant and irrelevant prompts during the representation extraction in SCR. Tab. X shows the downstream policy success rates for irrelevant (*"an elephant in the jungle"*) and relevant (*"a Franka robot arm opening a microwave door"*) prompts, compared to our default setting of not providing a text prompt (none). We see that providing a prompt does not help with downstream policy performance and can indeed degrade performance as the prompt gets more irrelevant to the visual context of the input.

TABLE X: Ablations of the input text prompt on Franka Kitchen.

Prompt Type	None	Relevant	Irrelevant
Success Rate	49.9 ± 3.4	49.2 ± 3.5	48.7 ± 3.3

We now move to a task that requires grounding language in vision. We consider the Referring Expressions Grounding task, which requires predicting the bounding box of an object in an image based on its textual description. Performance is measured by the accuracy of the box predictions, thresholded at a minimum overlap of 25% with the ground truth box. We show a sample image-text pair from the dataset to showcase the complexity of the task in Fig. 3 and defer a more thorough evaluation to Tab. III.

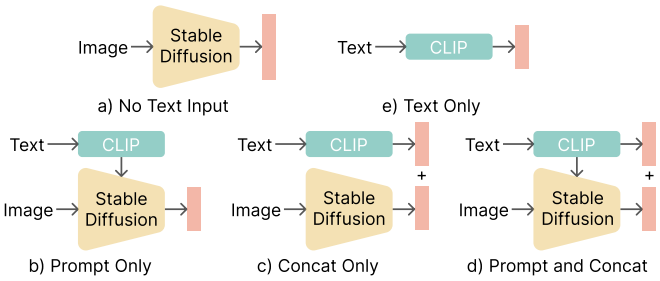

Fig. 3: Sample image-text pair from the OCID-Ref dataset [46] used for the Referring Expressions task.

We follow the same evaluation protocol of feeding in frozen representations to a trainable task-specific decoder as in Sec. IV. The decoder is a 4-layer MLP that predicts the bounding box coordinates. To study the role of the U-Net in shaping the visual representations guided by the text, we examine different text integration methods in Tab. XI and compare to the Voltron baseline. We compared the following approaches of providing the task's text specification to the task decoder (in column order):

(1) No text input: Exclude text prompt from both SCR and the task decoder by passing an empty prompt to the U-Net and using only the resulting SCR output for the decoder.

(2) Prompt Only: Pass text prompt only to the U-Net.

(3) Concat Only: Concatenate the CLIP embedding of the text prompt with the visual representation, feeding an empty prompt to the U-Net.

(4) Prompt + Concat: Combination of (2) and (3).

(5) Only text encoding: Removing visual representations completely and relying only on CLIP text embeddings.

Looking at the results of (1) and (2) in Tab. XI, it's evident that incorporating the text prompt into the U-Net significantly enhances accuracy compared to ignoring the text altogether. The transition from (2) to (3) indicates that directly providing text embeddings to the decoder improves performance, suggesting that certain crucial aspects for object localization are not fully captured by the representation alone. Going from (3) to (4) we see that with explicit text embeddings, further modulation by visual representations

TABLE XI: Ablating text input to SCR on referring expressions task.

Configuration	Score
Voltron	**92.5**
No text input	14.8
Prompt Only	82.7
Concat Only	**92.2**
Prompt + concat	**92.9**
Only text encoding	37.5

does not significantly benefit this task. Finally, (5) reveals the extent to which the task relies on text-based guesswork.

These findings align with both intuition and recent research in controllable generation through diffusion models [48], which underscores the challenges associated with using long-form text guidance. However, ongoing efforts that focus on training models with more detailed image descriptions or leveraging approaches to encode and integrate sub-phrases of lengthy texts could solve this problem.

H. Discussion

In appendix I-G, we deconstructed various components of SCR and identified where techniques used in our approach could apply to other foundational control models. Our analysis in Appendices I-F and I-G.1 revealed that using multi-layer features and appropriate spatial aggregation significantly affects performance, and overlooking these factors can lead to misleading conclusions about the capabilities of previously used representations. Next, our investigation into how language shapes diffusion model representations uncovered nuanced results. Text influence on representations does not uniformly enhance their downstream utility. This is particularly evident in tasks where text specification is not required and where training and test environments are congruent, minimizing the need for semantic generalization. Furthermore, tasks like referring expressions grounding demonstrate the necessity of direct access to text embeddings for accurate object localization, even when representations are modulated to considerable success.

In the OVMM task, we identified a scenario where multimodal alignment is essential. Here, we proposed a method to more explicitly utilize the latent knowledge of the Stable Diffusion model. While extracting similar text-aligned attention maps isn't straightforward for other multimodal models, future research could design methods to derive precise text-associated attribution maps for these models.

Finally, we contrast the simplicity of fine-tuning diffusion models with that of the contrastive learning objective required to fine-tune CLIP. While the former only requires image-text or image-only samples for the conditional or unconditional generation objectives respectively, the latter would require a sophisticated negative label sampling pipeline along with very high batch sizes to ensure that the model does not collapse to a degenerate solution [6].

I. Fine-tuning Stable Diffusion

For our experiments, we start with the `runwayml/stable-diffusion-v1-5` model weights hosted on huggingface.com and finetune them using the *diffusers* library. As mentioned in Sec. III-E, we use a subset of the frames from EpicKitchens, Something-Something-v2 and Bridge-v2 datasets. More specifically, we take the middle one-third of the video clips and sample 4 frames randomly from this chunk to increase the chances of sampling frames where the text prompt associated with the video clip is most relevant for describing the scene. This subsampling results in a paired images-language dataset of size 1.3 million. Fig. 4 shows some samples of the images from the finetuning datasets we use. Since different embodiments (human and robot) are visible in the training images, we prepend the corresponding embodiment name to the text prompt for the associated image during training.

We adopt the same text-conditioned generation objective as that of the base model for the fine-tuning phase. As is standard, we fine-tune the denoiser U-Net ϵ_θ but not the VAE encoder or decoder. Image-text pairs are uniformly sampled from the video-text pairs present in these datasets. A possible limitation of this strategy is that text-video aligned pairs (a sequence of frames in a control task that correspond to a single language instruction) may define a many-to-one relation for image-text pairs. However, as we see in experiments in which we compare to the base Stable Diffusion model in Sec. IV, this simple approach to robotics alignment is useful in most cases.

We finetune on the dataset for only a single epoch (5000 gradient steps) using 2 GPUs with a total batch size of 512 and a learning rate of $1e^{-4}$. Although the original Stable Diffusion model is trained on images of resolution 512x512, we finetune the model on images downscaled to 256x256, since it aligned with the resolution requirements of the downstream application. We show some sample generations from the diffusion model after finetuning in Fig. 5. Interestingly, we observe that the model learns to associate the prompt with not just the human or robot hand but also with the style of the background and objects of the training datasets.

J. Representation Extraction Details

Here, we describe the representation extraction details for all our baselines assuming a 224x224 input image:

- **Stable Control Representations**: The Stable Diffusion model downsamples the input images by a factor of 64. Therefore, we first resize the input image to a size of 256x256. We pass the image to the VAE, which converts it into a latent vector of size 32x32x4 and passes it to the U-Net. We use the last three downsampling blocks' and the mid block's output feature map of sizes 8x8x640, 4x4x1280, 4x4x1280, and 4x4x1280 respectively. The total size is, therefore, 102400, and we linearly interpolated them to the same spatial dimension (8x8) before concatenating them channel-wise.
- **R3M** [32]: For most of our experiments we use the original ResNet50 model, which outputs a 2048 dimensional vector. For the referring expressions and grasp affordance prediction tasks from the Voltron evaluation suite [33], a VIT-S is used, which outputs an embedding of size 14x14x384=75,264

Fig. 4: Snapshots from the datasets we use for finetuning the Stable Diffusion model.

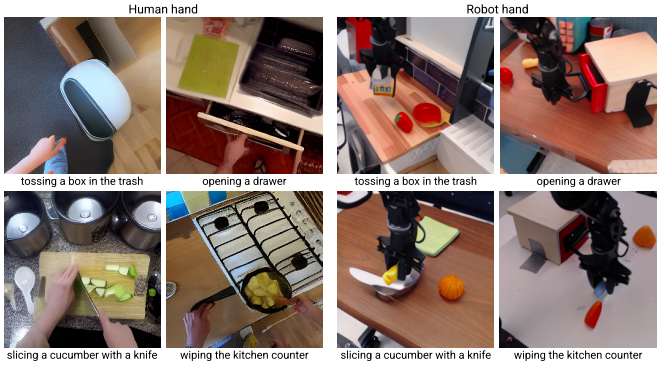

Fig. 5: Image generations from the finetuned Stable Diffusion model. We provide 4 different prompts, each prefixed with either *"Human hand"* or *"Robot hand"*.

- **MVP** [22] and **VC-1** [21]: The last layer (24^{th}) outputs an embedding of size 16x16x1024=262,144.
- **CLIP** [6]: For ViT-B, the last layer (12^{th}) outputs an embedding of size 14x14x768=150,528. For ViT-L, the last layer (24^{th}) outputs an embedding of size 16x16x1024=262,144.
- **Voltron** [33]: We use the VCond-Base model which outputs a representation of size 14x14x768=150,528.
- **SD-VAE** [1]: Outputs a latent vector of size 32x32x4=4096.

K. Task Details

1) Few-Shot Imitation Learning: For all baselines, we freeze the pretrained vision model and train a policy using imitation learning on the provided set of 25 expert demonstrations. The results are then reported as the average of the best evaluation performance for 25 evaluation runs over 3 seeds.

Meta-World. We follow [21] and use the hammer-v2, drawer-open-v2, bin-picking-v2, button-press-topdown-v2, assembly-v2 tasks from the Meta-World benchmark suite [44]. Each task provides the model with the last three 256x256 RGB images, alongside a 4-dimensional gripper pose. The model is a 3-layer MLP with a hidden dimension of 256 and is trained for 100 epochs similar to [21]. The training uses a batch size of 256 and a learning rate of 1e-3.

Franka Kitchen. The tasks involved here include Knob On, Knob Off, Microwave Door Open, Sliding Door Open, and L Door Open, each observed from three distinct camera angles. For each task, the model receives a 256x256 RGB image and a 24-dimensional vector representing the manipulator's proprioceptive state. For our experiments, we follow [45] and use a 2-layer MLP with a hidden dimension of 256 and train for 500 epochs. The batch size is set at 128, with a learning rate of 1e-4. *We additionally correct a bug in the RoboHive implementation of the VC-1 baseline, specifically on input image normalization. Adjusting the image normalization to a 0-1 range resulted in a significant improvement in its performance.*

2) OVMM: Open-Vocabulary Mobile Manipulation [7, OVMM] is a recently proposed embodied AI benchmark that evaluates an agent's ability to find and manipulate objects of novel categories in unseen indoor environments. Specifically, the task requires an agent to "*Find and pick an* `object` *on the* `start_receptacle` *and place it on the* `goal_recetacle`", where `object`, `start_receptacle` and `goal_recetacle` are the object category names. Given the long-horizon and sparse-reward nature of this task, current baselines [7] divide the problem into sub-tasks. The sub-tasks include navigation to the start receptacle, precise camera re-orientation to focus on the object (an abstracted form of grasping), navigating to the goal receptacle, and finally, object placement.

Since our aim is to investigate the open-vocabulary capabilities of pretrained representations, we choose to evaluate the models on only the precise camera re-orientation task (more commonly known as the **Gaze** task). In the original Gaze task, the agent is initialized within a distance of 1.5m and angle of 15° from the `object` which is lying on top of the `start_receptacle`. The episode is deemed successful when the agent calls the `Pick` action with the camera's center pixel occupied by the target object and the robot's gripper less than 0.8m from the object center. In our initial experiments, we found the current initialization scheme would lead the agent to learn a biased policy. This policy would call the `Pick` action after orienting towards the closest object in the field of view. Therefore, we chose to instantiate a harder version of the gaze task, where the episode starts with the agent spawned facing any random direction within 2.0m of the object.

We carry out our experiments in the Habitat simulator [49] using the episode dataset provided by [7]. This dataset uses 38 scenes for training and 12 scenes for validation, all originating from the Habitat Synthetic Scenes Dataset [50, HSSD]. These validation scenes are populated with previously unseen objects, spanning 106 seen and 22 unseen categories. The validation set consists of a total of 1199 episodes.

Our agent is designed to resemble the Stretch robot,

Fig. 6: Snapshots of a sample scene from the Habitat environments for the OVMM (left) and ImageNav (right) task.

characterized by a height of 1.41 meters and a radius of 0.3 meters. At a height of 1.31 meters from the base, a 640x480 resolution RGBD camera is mounted. This camera is equipped with motorized pan and tilt capabilities. The agent's action space is continuous, allowing it to move forward distances ranging from 5 to 25 centimeters and to turn left or right within angles ranging from 5 to 30 degrees. Additionally, the agent can adjust the head's pan and tilt by increments ranging from 0.02 to 1 radian in a single step.

In our experiments, we use a 2 layer LSTM policy and pass in the visual encoder representations after passing them through the compression layer. We initialize the LSTM weights with the LSTM weights of the Oracle model to get a slight boost in performance. We train our agents using the distributed version of PPO [27] with 152 environments spread across 4 80 GB A100 GPUs. We train for 100M environment steps while evaluating the agent every 5M steps and report the metrics based on the highest success rate observed on the validation set.

3) ImageNav: We conduct our ImageNav experiments in the Habitat simulator [51], using the episode dataset from [52]. The dataset uses 72 training and 14 validation scenes from the Gibson [53] scene dataset with evaluation conducted on a total of 4200 episodes. The agent is assumed to be in the shape of a cylinder of height 1.5m and radius 0.1m, with an RGB camera mounted at a height of 1.25m from the base. The RGB camera has a resolution of 128×128 and a 90° field-of-view.

At the start of each training episode, an agent is randomly initialized in a scene and is tasked to find the position from where the goal image was taken within 1000 simulation steps. At each step, the agent receives a new observation and is allowed to take one of the four discrete actions including MOVE_FORWARD (25 cm), TURN_LEFT (30°), TURN_RIGHT (30°) and STOP. The episode is a success if the agent calls the STOP action within 1m of the goal viewpoint. Similar to [31], [21] we train our agents using a distributed version of DD-PPO [27] with 320 environments for 500M timesteps (25k updates). Each environment accumulates experience across up to 64 frames, succeeded by two epochs of Proximal Policy Optimization (PPO) using two mini-batches. While the pretrained model is frozen, the policy is trained using the AdamW optimizer, with a learning rate of 2.5×10^{-4} and weight decay of 10^{-6}. Performance is assessed every 25M training steps, with reporting metrics based on the highest success rate observed on the validation set.

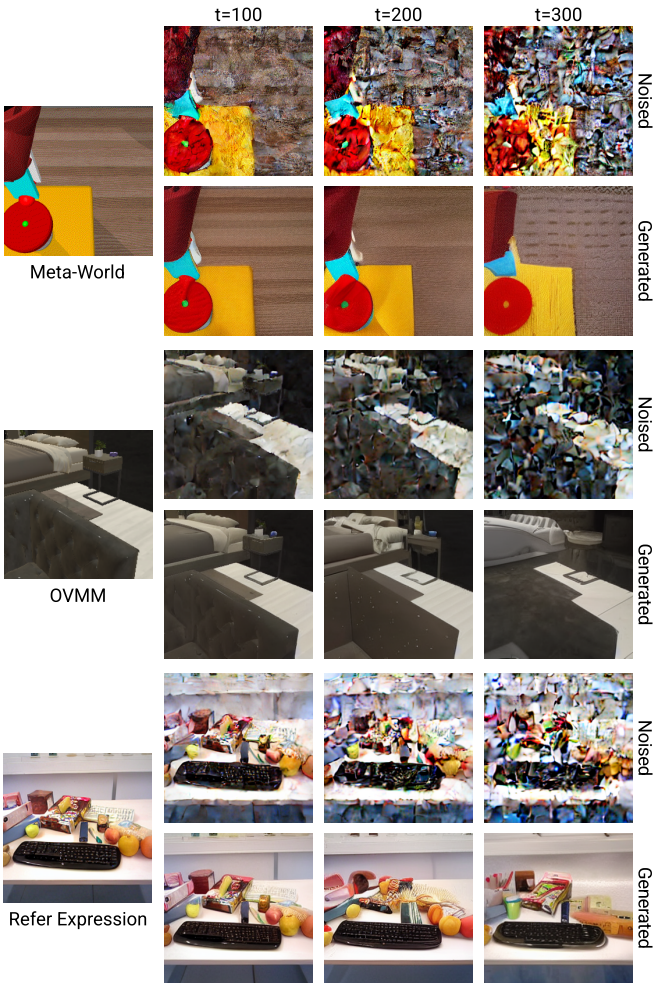

Fig. 7: Noising and denoising plots for images from 3 of our tasks using the finetuned Stable Diffusion model. For each image, we first add noise up to timestep t, where $t \in \{100, 200, 300\}$, and then denoise the image back to timestep 0. We observe that different tasks have different optimal timesteps based on the amount of information the images contain. On Meta-World, SD is able to reconstruct the image correctly even at t=200, while for refer expression, noising leads to information loss even at t=100.

L. Hyperparameters

We provide the hyperparameters used in Sec. IV for Stable Control Representations in Tab. XII.

TABLE XII: Hyperparameters and configuration settings used across tasks and methods.

Benchmark	Timestep	Prompt	Attn	Layers	Post Compression Dim
Meta-World	200	No	No	Mid + Down [1-3]	3072
Franka Kitchen	0	No	No	Mid + Down [1-3]	2048
ImageNav	0	No	No	Mid + Down [1-3]	2048
OVMM	100	Yes	Yes	Mid + Down [1-3]	2048
Referring Expression	0	Yes	No	Mid + Down [1-3]	8192
Grasp Prediction	0	No	No	Mid + Down [1-3]	8192

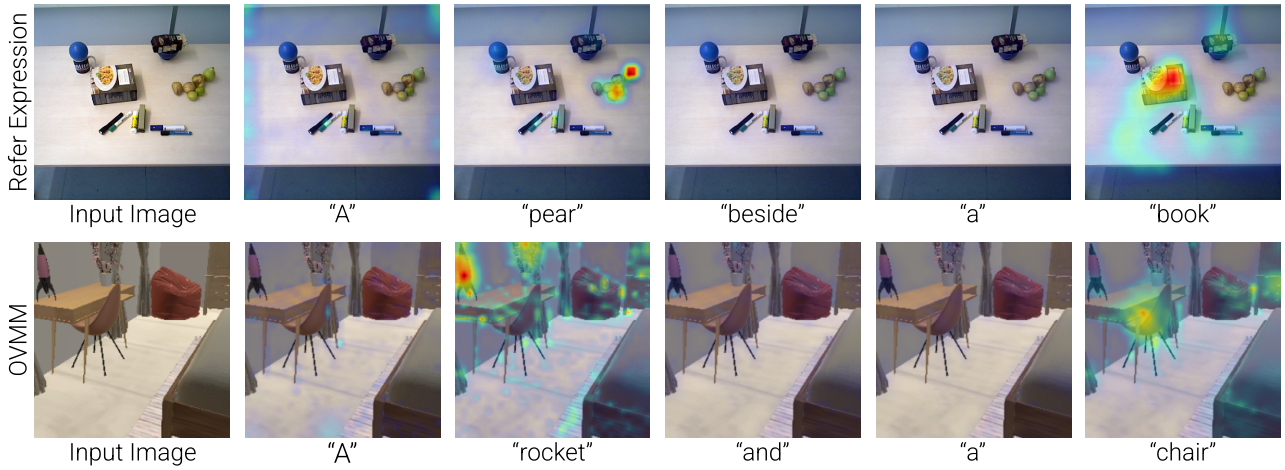

Fig. 8: The Stable Diffusion model allows us to extract word-level cross-attention maps for any given text prompt. We visualize these maps in a robotic manipulation environment and observe that they are accurate at localizing objects in a scene. Since these maps are category agnostic, downstream policies should become robust to unseen objects at test time.

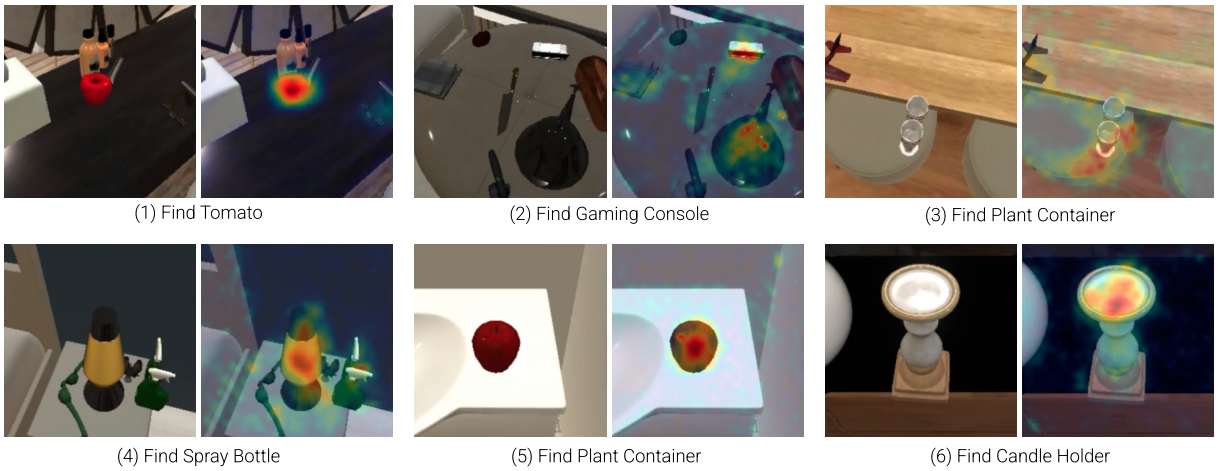

Fig. 9: Images from OVMM benchmark with their corresponding attention maps obtained from the finetuned Stable Diffusion (SD) model. The first 5 pairs of images correspond to failed episodes, with the bottom right pair corresponding to a successful episode. The attention maps help us interpret the cause of failure: (1) Tomato - SD wrongly attends strongly to an apple. (2) Gaming Console - visible in the top of the image; however, SD attends to multiple objects due to low visual quality. (3) Plant Container - SD instead focuses on the two glasses it sees in the image. (4) Spray Bottle - SD completely misses the spray bottles in the image and attends to the lava lamp. (5) Plant Container - SD wrongly attends to the apple. (6) Candle Holder - SD correctly attends.