Semantic World Models

Jacob Berg[1] Chuning Zhu[1] Yanda Bao[1] Ishan Durugkar[2] Abhishek Gupta[1]

[1]University of Washington [2]Sony AI

{jacob33,zchuning,yandabao,abhgupta}@cs.washington.edu
ishan.durugkar@sony.com

***Abstract*—Planning with world models offers a powerful paradigm for robotic control. Conventional approaches train a model to predict future frames conditioned on current frames and actions, which can then be used for planning. However, the objective of predicting future pixels is often at odds with the actual planning objective; strong pixel reconstruction does not always correlate with good planning decisions. This paper posits that instead of reconstructing future frames as pixels, world models only need to predict task-relevant *semantic* information about the future. For such prediction the paper poses world modeling as a visual question answering problem about semantic information in *future frames*. This perspective allows world modeling to be approached with the same tools underlying vision language models. Thus, vision-language models can be trained as "semantic" world models through a supervised finetuning process on image-action-text data, enabling planning for decision-making while inheriting many of the generalization and robustness properties from the pretrained vision-language models. In this paper, we introduce this new method and validate it on simulated robotic tasks leading to improvements over typical paradigms of reconstruction-based action-conditional world modeling.**

I. Introduction

World models are a class of learning methods capable of absorbing large amounts of data to make generative predictions about future outcomes in the world. These predictions can then be used to inform decision-making via planning [37, 15, 32, 17], helping policies acquire generalizable and robust behaviors. The practical instantiations of world models are diverse, ranging from smaller state-based dynamics models [1] to large action-conditioned video prediction models [4]. Across these instantiations, pixel-level reconstruction of future observations is commonly used as a training recipe. While these approaches are often successful at generating realistic images, as evident from high-quality video generations, they can be challenging to use for planning. Despite the visual fidelity, these predictions often miss (or misrepresent) key semantic details necessary for decision making, e.g., the details of precise dexterous contact. While there have been suggestions for modeling "task-relevant" latent representations [39, 17, 40], these methods often impose additional assumptions on the availability of rewards [18] or known factors [23], making them challenging to use in practice across a variety of world modeling problems.

If pixels are not necessary for planning, what is actually needed to make decisions about acting in the world? This paper posits that the ability to predict *semantic* information about future outcomes is sufficient. Rather than forecasting raw visual frames, world models should capture task-relevant information about objects and their interactions, e.g., "Did the arm get closer to the object?", "Did the red cube tip over?", "Was the blue moon picked up?". This work frames such information as a visual question-answering (VQA) problem about the future, leveraging the fact that any desired outcome can be expressed as a set of yes/no questions[1]. That is, *the problem of world modeling can be redefined as a VQA problem about outcomes in the future*.

[1]Other textual question-answer types may be applicable as well.

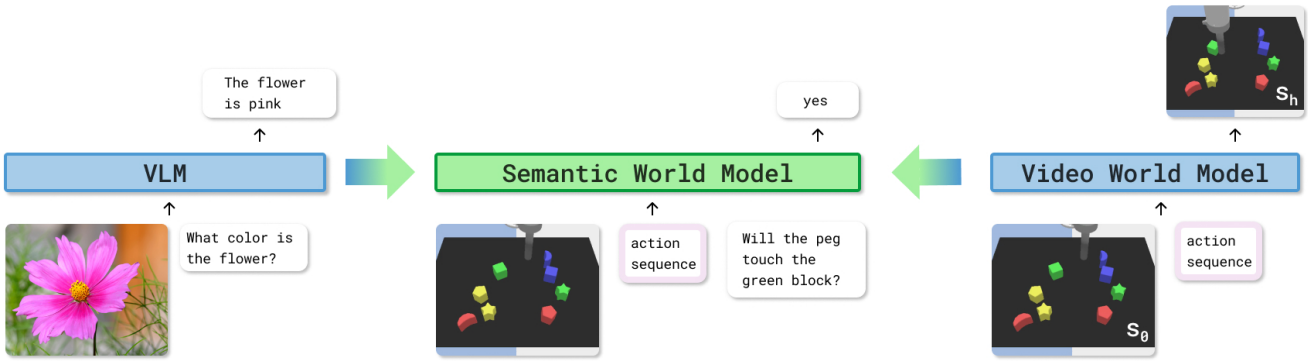

Fig. 1: Comparison between Vision-Language Models, Video World Models, and **Semantic World Models**. While Vision-Language Models answer questions about static observations and Video World Models predict future observations given actions, Semantic World Models take observations and actions as input to directly answer questions about the future outcomes of those actions.

There already exists a class of models with extensive tooling for VQA on static observations, i.e., vision-language models (VLMs). For world modeling, VLMs offer two key advantages: they provide a strong foundation for VQA through large-scale pretraining and broad generalization, and they encode prior knowledge about which tasks and semantic features are relevant in a scene. These strengths make frontier VLMs well suited to formulating task-relevant questions and producing reliable answers when given static observations. However, their lack of predictive capacity about future outcomes limits their direct utility for decision-making.

This work proposes a new paradigm for world modeling - Semantic World Models (SWM): a world model that is represented as an action-conditional vision-language model that answers questions about the semantic effects of actions in the future. Unlike traditional world models that predict future frames, a Semantic World Model *answers questions about the future* given current observations (represented as an image) and a sequence of actions. As shown in Fig. 1, the model takes as input the current observations, a proposed action sequence, and a natural language query about the future. It then generates a textual answer by understanding the consequences of taking the actions in the environment. Since SWM is fundamentally a task-agnostic world model, it can be trained on general sequential data with minimal quality assumptions, including both play and suboptimal data. The training data can be easily obtained from any (expert or non-expert) data corpus in the format of current observations, actions, questions (about the future), and expected answers.

The ability to reason about outcomes in the future with an SWM enables flexible open-world multi-task planning in action space: given a task specification in natural language, one could either leverage a pre-trained VLM [27, 5] or manually decompose the task specification into a set of questions and expected answers in text form. Given this QA set, SWM can then be used to plan actions that elicit the expected answers to these questions *in the future* with high likelihood. While a plethora of techniques can be used for this planning, this work shows compatibility with both zero-order sampling-based methods [30, 37] and first-order gradient planning methods [31, 32] that perform optimization with respect to the expected likelihood objective. It shows that these planning methods can be computationally tractable, enabling a significant test-time improvement over nominal action selection methods. Moreover, it demonstrates the extensibility of such planning methods to multi-step long-horizon problems.

SWM is empirically evaluated on a suite of multiple different tasks in two commonly used multi-task simulation domains – Language Table (LangTable) [25] and OGBench [28]. This evaluation shows that (1) SWM can accurately answer questions about future outcomes while generalizing to novel scenes, and (2) SWM can be combined with standard sampling-based planning techniques and a gradient-based improvement technique to solve diverse robotics tasks with considerable policy improvement through test-time optimization. SWM introduces a new class of world models that leverage the rich pretraining knowledge from VLMs for grounded, flexible, and scalable robotic control.

II. Related Work

a) Vision-Language Models (VLMs): broadly encompass representation learning methods and multimodal generative models trained on vision and language data. Representation learning methods jointly train a vision encoder and a text encoder by aligning their encoded representations. These representations can then be utilized in various applications, such as classification, retrieval, and control. CLIP [29] learns such representations from image-text data by utilizing a contrastive loss, contrasting positive image-text pairs with negative pairs. SigLIP [38] replaces the contrastive loss with a pairwise sigmoid loss to facilitate scalable training. Multimodal generative models, commonly known as VLMs, enable a broad range of promptable behaviors such as understanding, summarizing, and question answering [27, 13, 11, 3, 5, 36]. A VLM takes in an image and a language prompt as input and generates a natural language response. They are typically trained with a next-token prediction objective. Recently, a family of vision-language-action models (VLAs) has been introduced to bring the vision-language understanding capabilities of VLMs to embodied decision-making [7, 20, 6]. VLAs are trained on annotated robot trajectories to generate actions conditioned on image observations and language instructions. OpenVLA [20] directly predicts discrete action tokens, while Pi-0 [6] decodes actions via a diffusion action head. Unlike VLAs, an SWM takes in observations, actions, and a natural language prompt as input, and generates a natural language response about the future after taking the actions. In some sense, an SWM can be viewed as an "inverted" VLA, where the actions become the input and the language becomes the output. This approach hypothesizes that using language as the output format can better retain the pretraining knowledge of VLMs, since they were trained with next token prediction objectives.

b) World Models for Control: are approximate models of the dynamics of the world, typically trained to predict future observations conditioned on current observations and actions. The ability to forecast the future without interacting with the world can greatly facilitate decision-making and control. A prominent line of work focuses on planning with world models. [10, 15, 32]. PETS [10] learns a one-step dynamics model and applies the cross-entropy method to plan for optimal actions for a given reward. PlaNet [15] learns a recurrent latent dynamics model with a reconstruction objective and applies planning in the latent space. LatCo [32] leverages collocation-based planning to enable long-horizon planning with latent dynamics models. Another line of work utilizes world models as a simulator for reinforcement learning [16, 39, 17]. Dreamer [16] and TD-MPC [17] use a latent dynamics model to generate rollouts for actor-critic policy optimization, achieving remarkable sample efficiency. [39] learns a latent representation predictive of dynamics and reward, which can then be used as an invariant representation for RL policies. Recently, world models have been used together with

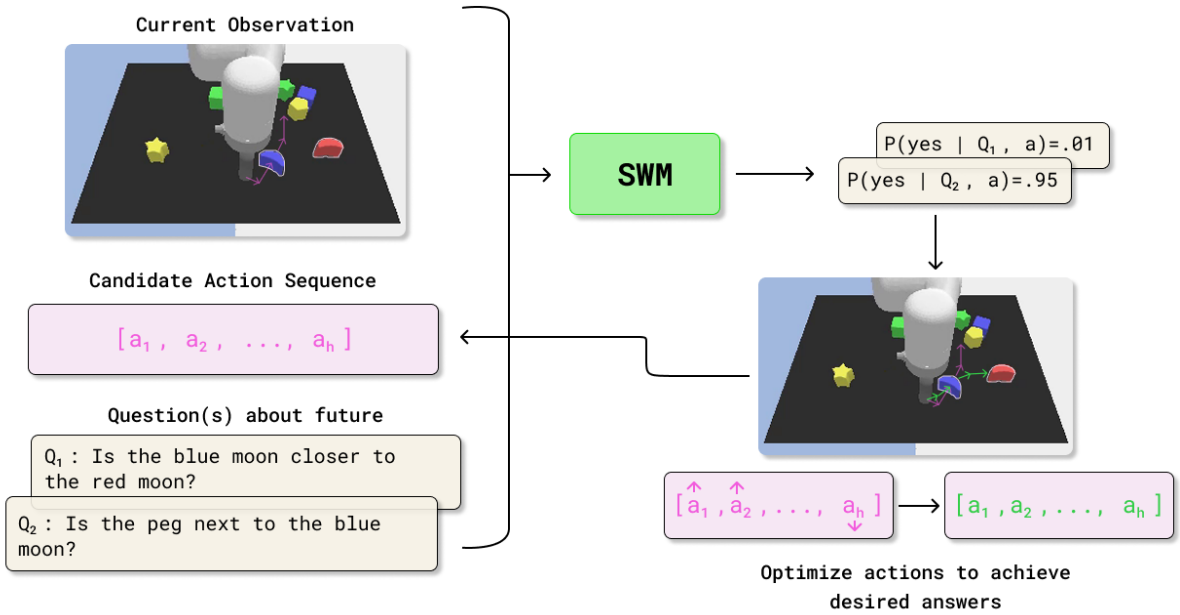

Fig. 2: **Overview of Semantic World Models.** SWM is a VLM adapted to answer questions about the future realized by the actions used to condition the model. Using a set of questions and desired answers, its predictions can be converted into a planning signal and iteratively refine the action sequence.

imitation learning methods to facilitate out-of-distribution generalization [12, 41]. UniPi [12] uses a world model as a high-level planner to condition low-level policies. UWM [41] trains a unified video-action diffusion model, incorporating video data into pretraining to improve generalization. Unlike these explicit world models, SWM understands the dynamics of the world by reasoning in language space, allowing the model to bootstrap from the Internet-scale pretraining of VLMs. SWM can then be used with planning techniques to derive versatile language-conditioned policies. Additional work also explores abstractions connected to Semantic World Models. VLWM [8] generates a high-level textual plan given the visual state of the world and a textual goal, using reasoning about the effects of each high-level step to ground the planning. Prior works on predicate learning [33, 2] learn abstract or semantic predicates to decompose long-horizon tasks into shorter subgoals, a direction that is complementary to SWM's use of future QA for planning.

III. World Modeling as VQA

This section presents details of the data generation pipeline, the SWM architecture, and the training methodology. It then touches on the sampling-based and gradient-based planning methods used for policy extraction under SWM. Fig. 2 provides an overview of the model and planning procedure.

A. Dataset Generation

To train a world model to answer questions about the future, a state-action-question-answer (SAQA) dataset is generated. It is defined as

$$\mathcal{D}_{\text{SAQA}} = \{(S_i, a_{i:j}, Q_{S_j}, A_{S_j}), \dots\} \quad \text{where } j = i + h$$

where S_i represents the current state (RGB frame in our case), h is the horizon, $a_{i:j}$ is a sequence of actions taken from state S_i, and Q_{S_j}, A_{S_j} is a question answer tuple about the future state S_j which is reached by taking actions $a_{i:j}$ from state S_i. Fig. 3 illustrates a single state paired with multiple questions and answers in the dataset.

The SAQA dataset is generated from a dataset of trajectories $\{T_1, T_2, \dots\}$, where each trajectory is given by a sequence of state-action tuples $\{(S_0, a_0), (S_1, a_1), \dots\}$. Here, each state comprises an image observation and privileged information, such as object positions, which are used for programmatic question generation. For each state S_i in the trajectory, multiple different action horizons h are sampled. As shown in Fig. 3, for each sampled horizon h, the oracle information from future state S_{i+h} is used to create a set of questions and answers, which gives the final dataset to train the model.

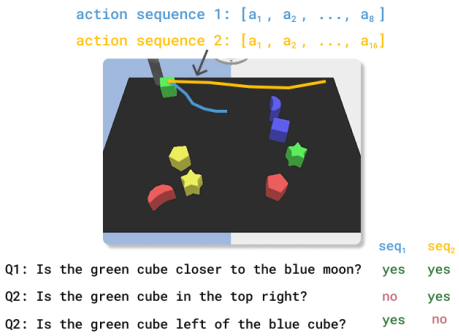

Fig. 3: Example state entry in the SAQA dataset with two action horizons and six QA pairs.

For each type of question generation, multiple phrasings are included in the training dataset. Examples of training question types and rewards for each task are provided in the Appendix VI-C2. While this generation procedure currently uses state information to construct SAQA tuples, we provide preliminary evidence that off-the-shelf frontier VLMs can supply this supervision directly from images, achieving 87–92% accuracy against ground truth across LangTable, OGBench, and DROID. This suggests a path toward removing the oracle dependence and scaling SAQA generation to real-world data; full results are reported in Appendix VI-E4.

B. Semantic World Models Architecture

This section presents a model capable of answering questions about future events conditioned on actions. A model with such capability is fundamentally a visual question-answering model with action conditioning. Therefore, it is natural to bootstrap from large pretrained VLMs to transfer their generalization capabilities to robotics tasks. This SWM architecture is based on an open-source VLM, PaliGemma [5].

The model contains three core pretrained components: a transformer-based autoregressive language model with a token embedding size d_{tok}, a vision encoder v_ϕ with a feature size d_{img}, and a projection matrix $W \in \mathbb{R}^{d_{\text{tok}} \times d_{\text{img}}}$. The PaliGemma architecture is built on top of two individually trained components: the Gemma LLM [14] and the SigLIP image encoder V_{sc} [38]. W is used to project from Z_{sc} to Z_{LLM}, where Z_{sc} is the feature space of v_ϕ, and Z_{LLM} is the input token embedding space of the LLM. This paper uses the 3B parameter checkpoint from PaliGemma as the base model. This architecture and components are described in Appendix VI-A.

To adapt the base model to answer questions about a specific future as a result of the actions, the model needs to be conditioned on these actions. Thus, a new projection matrix $P \in \mathbb{R}^{d_{\text{tok}} \times d_{\text{act}}}$ is used which projects a single action $a \in \mathbb{R}^{d_{\text{act}}}$ into the latent space Z_{LLM} similar to the W projection matrix. Given a tuple $(S_i, a_{i:j}, Q_{S_j}, A_{S_j})$ from the dataset $\mathcal{D}_{\text{SAQA}}$, the input sequence is constructed by concatenating the image embeddings, action embeddings, and question token embeddings as $\texttt{concat}\left(W^\top V_{sc}(S_i), P^\top a_i, P^\top a_{i+1}, \ldots, P^\top a_j, Q_{S_j}\right)$. The model is then fine-tuned in an end-to-end manner to predict the target answer A_{S_j} by optimizing the standard cross-entropy loss

$$\mathcal{L} = -\log p(A_{S_j} | S_i, a_{i:j}, Q_{S_j}).$$

This training procedure enables the model to capture the dynamics of the environment in language space to answer questions about future states without explicitly generating pixel-level representations.

C. Planning with Semantic World Models

Planning with world models requires evaluating the value of action sequences. For each task, a set of questions (e.g., "is the gripper touching the block") and desired answers (e.g., "yes") can be defined. For our main results, each task has a predefined set of questions and weights. We also provide ablations on the importance of these weights and on automatic question decomposition in Appendix. VI-E. A scalar score is then derived by combining the likelihood of the model generating the desired answer for each question, weighted by some heuristic weights. Specifically, each task is defined as a set of questions, answers, and weights $\mathcal{T} := \{(Q_i, A_i^*, W_i)\}_{i=1}^k$. Given an observation S and a sequence of actions $a_{1:n}$, its value under the task is calculated as:

$$V^{\mathcal{T}}(S, a_{1:n}) = \sum_{i=1}^{k} W_i \cdot p_{\text{wm}}(A_i^* | S, a_{1:n}, Q_i) \tag{1}$$

Empirical evaluation shows that rewarding the model for achieving the desired outcome earlier in the action sequence leads to better performance. This early reward is provided by breaking each full action sequence down to sub-chunks of length c, and then querying the model on action sequences with increasing numbers of concatenated sub-chunks:

$$V^{\mathcal{T},c}(S, a_{1:n}) = \sum_{i=1}^{k} \sum_{\substack{j=c \\ j+=c}}^{n} W_i \cdot p_{\text{wm}}(A_i^* | S, a_{1:j}, Q_i) \tag{2}$$

Setting $c = 1$ is equivalent to evaluating the model once for every single action in the sequence, and setting $c = h$ is equivalent to the vanilla formulation in Eqn. 1. Various planning techniques can be used to extract optimal actions by using the model with a well-defined value function.

1) Sampling-Based Planning: Sampling-based planning provides a straightforward approach to planning with the model. An example is Model Predictive Path Integral (MPPI) control algorithm [37], which maintains a Gaussian distribution of action parameters and iteratively refines it by querying the model. The action distribution is initialized as $\mathbf{a}^{(0)} \sim \text{Unif}(a_{\min}, a_{\max})$. At each iteration, a set of K control sequences $\{\mathbf{a}^{(k)}\}_{k=1}^K$ is sampled from the current action distribution. The value of each of these sampled trajectories V_k is computed using our SWM. The distribution for the next iteration is $\mathbf{a}_{t+1} \sim \mathcal{N}\left(\mu_t, \sigma_t^2\right)$ where

$$\mu_t = \sum_{k=1}^{K} \frac{\exp\left(\frac{V_k}{\lambda}\right)}{\sum_{j=1}^{K} \exp\left(\frac{V_j}{\lambda}\right)} \mathbf{a}_t^{(k)}, \sigma_t^2 = \sum_{k=1}^{K} \omega_k \left(\mathbf{a}_t^{(k)} - \mu_t\right)^2$$

and λ is a temperature parameter that controls exploration. For our rollouts, we execute the mean sequence of the last iteration.

2) Gradient-Based Planning: For more complicated tasks, sampling-based planning methods typically require a large number of samples and optimization iterations, which become increasingly hard to scale for a large model like SWM. To reduce the number of samples and model forward passes, we propose a gradient-based optimization procedure together with a base proposal policy. The gradient provides directed information for optimizing the model, thus converging faster than sampling-based techniques. The base proposal policy can effectively trim down the planning search space. Given a base policy π_b, a control sequence $\mathbf{a} \sim \pi_b(S)$, and the semantic

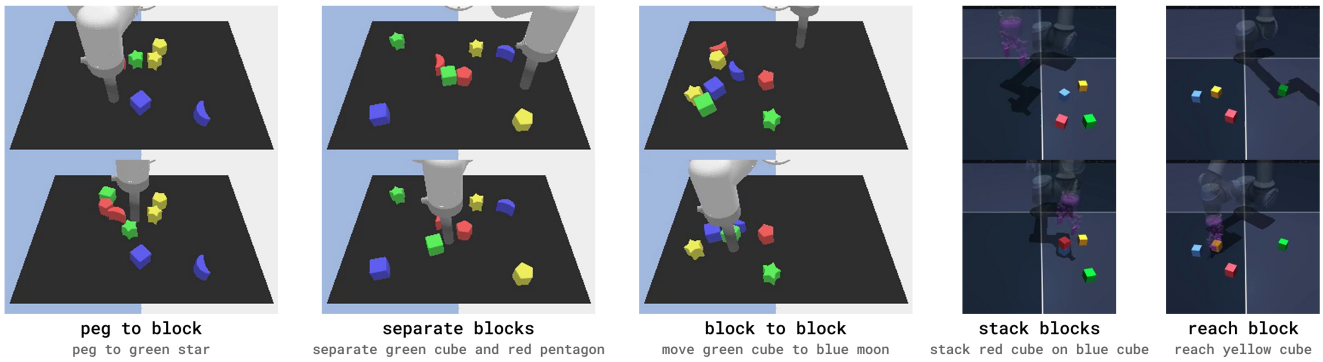

Fig. 4: Examples of each evaluation task. The top frame represents the initialization, and the bottom frame represents task completion. The first three tasks are for LangTable and the last two are for OGBench.

world model p_{wm}, gradient ascent is used to optimize the following objective:

$$J^{\mathcal{T}}(\mathbf{a}) = V^{\mathcal{T},c}(S, \mathbf{a}) \tag{3}$$

Where $\mathbf{a}$ is the control sequence being optimized, $\mathcal{T} = \{(Q_i, A_i^*, W_i)\}_{i=1}^{k}$ is the list of questions, desired answers, and weights, c is the reward subchunk size, and S is our state. To improve stability during learning, gradient norm clipping is used before each step. Refer to Appendix VI-E7 for a visualization of this optimization process and Appendix VI-E8 to compare the computational speed of planning times for each method.

D. Multistep Tasks

To solve long-horizon tasks, the aforementioned planning procedure can be extended to a multi-step formulation. The capabilities of SWM can be used to track task progress and transition between subgoals without requiring any additional components. A series of sequential subgoals $g_1, g_2, \ldots, g_T$ is defined where each subgoal g_t is associated with a question and a desired answer corresponding to when the subgoal was completed. Each subgoal is executed sequentially and its completion is verified using SWM. This verification is feasible at no additional cost because zero-horizon examples are included in the training dataset. For example, in the block picking task, the following sub-goals are used: ["Is the block grasped?", "Is the block stacked on top of the other block?"], with the desired answers ["yes", "yes"] in order to accomplish a two-stage task. This method is used to extend planning to multi-step LangTable tasks.

IV. Experiments and Results

A. Experimental Setup

SWM is evaluated in two simulation environments, LangTable [25] and OGBench [28], capturing combinatorial generalization and dexterous manipulation. Fig. 4 shows examples of tasks in each domain. This section provides an overview of the experiment setup and details are provided in Sec. VI-B.

a) LangTable: [25] SWM is evaluated on *reaching*, *separating blocks*, and *pushing* in the LangTable environment, using both sampling-based planning and gradient-based improvement over a base policy. SWM is trained on a mixture of expert data collected with a scripted policy and suboptimal data collected with a random policy. To evaluate in out-of-distribution conditions, the block color combinations are changed during evaluation to test compositional generalization. For example, our training data only includes the red pentagon, and evaluation is performed with a green pentagon and a novel purple pentagon.

b) OGBench: [28] In OGBench SWM is evaluated on *cube reaching* and a custom *cube stacking* task. It is trained on a mixture of optimal and suboptimal data, collected using the provided noisy expert data and play data from OGBench, respectively. Background color is changed during evaluation to measure generalization.

For both environments, a per-task Diffusion Policy [9] is trained on 300 expert trajectories for 100 epochs as the base policy. The expert trajectories were collected using the same experts as in the offline dataset.

During training, the dataset was balanced in both the number of each possible question type and the answer distribution for each respective question. For example, for each state in the LangTable environment, there are $\binom{8}{2}$ possible questions about whether two blocks are touching, but 8 questions about whether the end effector is touching a given block. Similarly, most blocks are separated in the initial states of the LangTable environment, leading to far more 'yes' answers than 'no' answers. The imbalance is addressed during training by oversampling tuples such that there is a balanced amount of question types and answer distributions.

B. Baselines

Semantic World Models is compared to the baselines below. Details about each baseline and hyperparameters are described in Appendix VI-B.

IDQL [19]: IDQL is an offline RL baseline which uses IQL [21] to reweight a behavior diffusion policy. For each task, the offline dataset used for Semantic World Model is combined with the per-task expert dataset used for the base policy. This

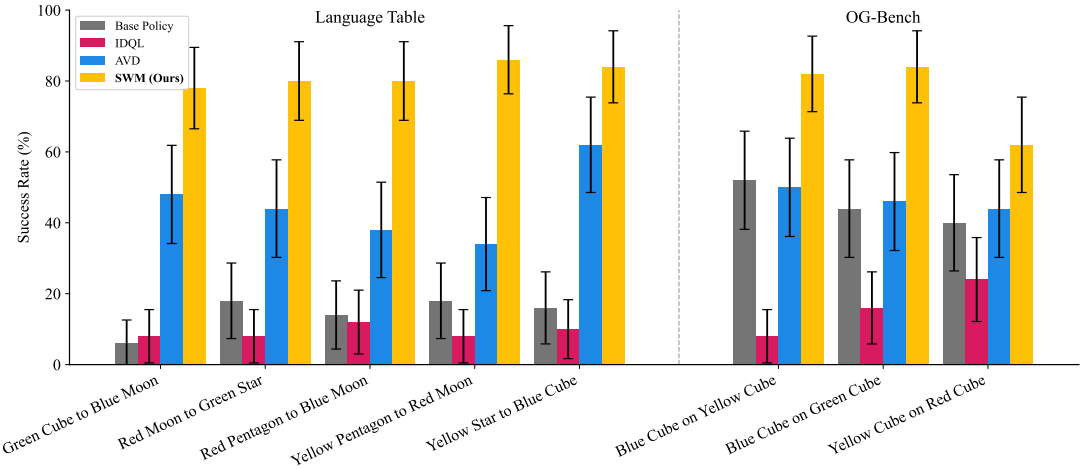

Fig. 5: **Policy Improvement** across LangTable and OGBench across multiple tasks. The average success rates of the base policies (14.4% on LangTable and 45.33% on OGBench) increase to 81.6% and 76.0%, respectively. SWM further outperforms the IDQL and AVD baselines across all evaluated tasks and environments. Reported success rates over $n = 50$ seeds with 95% confidence intervals (normal approximation).

combined dataset is labeled with binary rewards and used to train the IDQL policy. The architecture and hyperparameters of the diffusion policy used as the IDQL behavior policy are the same as for the base policies, except with a horizon of 8.

Action Conditioned Video Diffusion (AVD): To isolate the benefits of modeling only task-relevant semantics, we introduce a pixel-based world model baseline. We train a diffusion transformer model to predict future observations conditioned on actions. In order to obtain learning signal from these raw pixel predictions, this baseline first generates the future observation and then extracts the necessary semantic information using VQA. The resulting outputs are used as rewards for MPPI planning. This baseline therefore represents a conventional pixel-centric approach: extracting task-relevant information from the generated visual observations, rather than modeling semantic information directly. The initial trajectory candidate samples are generated through the same base diffusion policy used in SWM. To ensure a fair comparison, we prioritize AVD's generation quality (using 100 diffusion steps) over inference speed.

Latent World Models: We additionally evaluated two latent world modeling approaches, Dreamer [16] and TDMPC2 [18], online model based RL approaches, which we modify to use our offline dataset and sparse rewards. We found that neither was able to successfully solve the LangTable tasks under our setting, so we omit their performance from the main results. Additional details are provided in Appendix VI-B1.

C. Results

The evaluation aims to address the following questions: (1) Is SWM an effective world model for decision making? (2) Does suboptimal data improve modeling performance? (3) Does SWM preserve the generalization capabilities from the base VLM?

a) Is SWM an effective world model for decision making? : The planning capabilities of SWM is evaluated first by applying a sampling-based planning method, MPPI, to a SWM model on LangTable and OGBench tasks. As shown in Tab. I, it is possible to directly plan on top of the semantic world model using sampling-based planning methods, achieving close to perfect success rates on reaching and block separation tasks across both environments.

Task	SWM
LT Reach Block	100%
LT Separate Blocks	100%
OG Reach Cube	97%

TABLE I: **Planning Results** MPPI planning success rates over 100 seeds.

However, the computational cost of the sampling-based planning method with large models makes it infeasible to run MPPI on more challenging tasks requiring a higher number of samples. Therefore, for more complicated tasks, consider a scenario in which a base policy generates a candidate trajectory that is refined using SWM and gradient-based optimization (described in Sec. III-C2). As shown in Fig. 5, the method is able to refine candidate trajectories and show substantial improvement over the base policies. SWM demonstrates an average performance increase over the base

Task	Base Policy	AVD	SWM (Ours)
MS1	6% ± 6.6	8% ± 7.5	**50%** ± 13.9
MS2	4% ± 5.4	2% ± 3.9	**66%** ± 13.1
MS3	4% ± 5.4	2% ± 3.9	**54%** ± 13.8
MS4	2% ± 3.9	4% ± 5.4	**54%** ± 13.8

TABLE II: **Multi-Step Results.** SWM model improvement results on four different multi-step compositional tasks. The tasks are as follows: MS1 - red pentagon to blue moon, yellow pentagon to red moon. MS2 - yellow star to blue cube, yellow pentagon to red moon. MS3 - yellow star to blue cube, red pentagon to blue moon. MS4 - green cube to blue moon, yellow pentagon to red moon. Reported success rates over $n = 50$ seeds with 95% confidence intervals (normal approximation).

Dataset Type	*LangTable* Expert Data	*LangTable* Expert Data OOD	*OGBench* Expert Data	*OGBench* Expert Data OOD
Sub Optimal	85.98 ± 0.33	81.99 ± 1.46	90.83 ± 0.39	85.56 ± 1.10
Expert	91.27 ± 0.79	86.49 ± 0.39	96.53 ± 0.13	87.33 ± 2.13
Combined	**92.92** ± 0.34	**88.32** ± 2.10	**96.86** ± 0.13	**88.16** ± 1.54

TABLE III: **Future QA Performance.** Accuracy of answers on future QA evaluated on expert SAQA datasets generated by experts on test time seeds in both in-domain and out-of-domain block combinations. Reported standard deviation across 3 model training seeds.

policies from 14.4% to 81.6% on average for LangTable and 45.33% to 76% on average for OGBench. SWM also outperforms both the AVD and IDQL baselines across all tasks, demonstrating the effectiveness of SWM for planning.

SWM also demonstrates the capability for longer horizon tasks by both selecting subgoals and then planning using that specific subgoal. SWM demonstrates an average policy improvement of 52.0% as shown in Tab. II on multistep tasks, outperforming the AVD baseline. For both AVD and SWM, subgoal completion was determined using the SWM model without action conditioning.

b) Does suboptimal data improve modeling performance?: One of the key aspects of a world model is its ability to learn from suboptimal data. To measure the effects of suboptimal demonstrations, a test set of future QA data collected from expert demonstrations in both the in-distribution and out-of-distribution environments is created. The models are then trained on three different seeds and fixed hyperparameters to convergence with suboptimal data, optimal data, or a 50/50 mixed dataset. As seen in Table III, mixing in the suboptimal data improves accuracy over training on just expert data. SWM is also able to achieve moderate levels of performance by training only on suboptimal data, demonstrating how effective suboptimal data can be for training our world model.

c) Does training preserve the generalization capabilities from the base VLM?: To measure the effects of VLM pretraining on generalization, SWM is evaluated on compositional and scene out-of-distribution environments, depicted in Fig. 8. Since the offline dataset was misaligned with these evaluation tasks, the IDQL baseline is not evaluated.

To measure semantic compositional generalization, a new colored block is introduced and the existing block color-shape pairs are modified in the LangTable environment. Tab. IV shows an average of 20.0% improvement over the base policies under these conditions. This performance indicates that SWM is able to retain some of the pretraining knowledge, resulting in compositional generalization.

To test robustness to background changes, OGBench's background color is changed to a novel combination. SWM is again able to demonstrate a 20% boost in performance compared to the base policy and is able to generalize to these conditions, while the AVD method is unable to.

d) Does the model's internal representations attend to the task-relevant information?: To understand the learned representations of the model, the attention maps from the language tokens to the image patches are visualized from an intermediate layer of the model. As shown in Fig. 6, the model correctly attends to the task-relevant location in the image depending on the language prompt. For example, when asked "Is the red moon touching the blue cube?", the attention score is higher on the image patches corresponding to the objects. Although never finetuned on questions with more than two objects, the model was found to correctly attend to three objects when asked to. This shows that the model inherits generalization from the pretrained VLM. In Appendix VI-E6 more visualizations of individual layers as well as entire trajectories are provided.

V. Conclusions

This paper presents semantic world models, a novel world modeling approach that explicitly models future outcomes through QA about the consequences without needing to reconstruct or use pixel-level information as a training objective. It shows that this approach can be used with both sampling-based planning and gradient-based policy improvement. Empirical evaluation demonstrates considerable gains over pixel and latent based world modeling and offline RL methods, suggesting SWM could be the basis of a new framework for world modeling. These benchmarks were selected to highlight complementary aspects of the approach, covering long-horizon and compositional tasks (LangTable), and tabletop manipulation (OGBench), providing meaningful evidence for the viability of the SWM paradigm.

Task	Base Policy	AVD	SWM (Ours)
Push Blue Star to Red Cube	54% $\pm$ 13.8	66% $\pm$ 13.1	**86%** $\pm$ 9.6
Push Yellow Moon to Purple Cube	54% $\pm$ 13.8	56% $\pm$ 13.8	**78%** $\pm$ 11.5
Stack Red to Green OOD Background	62% $\pm$ 13.5	28% $\pm$ 12.4	**72%** $\pm$ 12.4
Stack Blue to Yellow OOD Background	50% $\pm$ 13.9	50% $\pm$ 13.9	**70%** $\pm$ 12.7

TABLE IV: **Out-of-Distribution Improvement Results.** SWM model improvement results on tasks in LangTable and OGBench on out-of-distribution scenes. SWM is able to show policy improvement and outperform AVD across both environments. Reported success rates over $n = 50$ seeds with 95% confidence intervals (normal approximation).

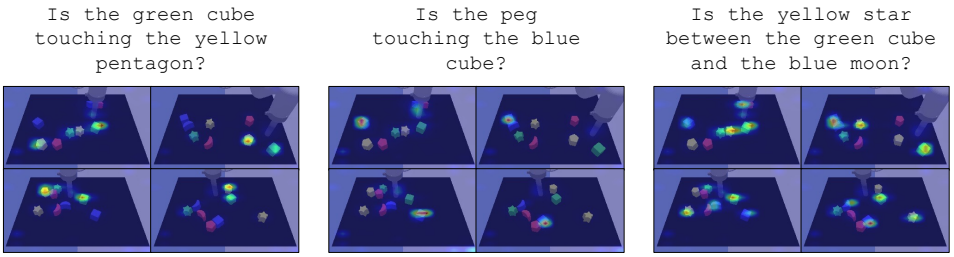

Fig. 6: Visualization of the attention map from language tokens to image patches in the 4th transformer layer. The language tokens correctly attend to the task-relevant locations in the image depending on the prompt.

Limitations and Future Work:: While Semantic World Models demonstrate strong performance on these tasks, several limitations remain that suggest directions for extending this paradigm to real-world settings. First, the high parameter count of the base VLM makes sample-based planning methods computationally prohibitive to perform on a single GPU or at a reasonable control frequency. Although gradient-based planning is substantially more efficient, it depends on a base policy to generate an initial trajectory. Recent advances in compact VLMs, such as FastVLM and SmolVLM [26, 22], suggest a promising path toward scalable sample-based planning without reliance on a base policy. Second, SWM currently uses ground truth state information to construct the SAQA dataset, which is difficult to obtain in real-world environments. Replacing oracle-generated QA pairs with those bootstrapped from a base VLM could significantly increase data diversity and enable the incorporation of real-world data into the training of Semantic World Models. We present preliminary results evaluating the accuracy and forward progress of frontier VLMs for this form of spatial reasoning supervision in Appendix VI-E4. Finally, the current formulation uses binary yes/no questions for the world model, although the framework should conceptually cover richer QA pairs; extending the QA framework to richer, open-ended responses or including reasoning could unlock more expressive semantic supervision.

REFERENCES

[1] Bo Ai, Stephen Tian, Haochen Shi, Yixuan Wang, Tobias Pfaff, Cheston Tan, Henrik I. Christensen, Hao Su, Jiajun Wu, and Yunzhu Li. A review of learning-based dynamics models for robotic manipulation. *Science Robotics*, 10(106):eadt1497, 2025. doi: 10.1126/scirobotics.adt1497. URL https://www.science.org/doi/abs/10.1126/scirobotics.adt1497.

[2] Ashay Athalye, Nishanth Kumar, Tom Silver, Yichao Liang, Jiuguang Wang, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. From pixels to predicates: Learning symbolic world models via pretrained vision-language models, 2025. URL https://arxiv.org/abs/2501.00296.

[3] J. Bai, S. Bai, S. Yang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. URL https://arxiv.org/abs/2309.16609.

[4] Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models, 2025.

[5] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, 2024. URL https://arxiv.org/abs/2407.07726.

[6] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. π_0: A vision-language-action flow model for general

robot control. arXiv preprint arXiv:2410.24164, 2024. submitted October 2024.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.025.

[8] Delong Chen, Theo Moutakanni, Willy Chung, Yejin Bang, Ziwei Ji, Allen Bolourchi, and Pascale Fung. Planning with reasoning using vision language world model, 2025. URL https://arxiv.org/abs/2509.02722.

[9] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[10] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pages 4754–4765, 2018.

[11] M. Deitke et al. Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models. *arXiv preprint arXiv:2409.17146*, 2024. URL https://arxiv.org/abs/2409.17146.

[12] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bo8q5MRcwy.

[13] Gemini Team. Gemini: A family of highly capable multimodal models, 2023. URL https://arxiv.org/abs/2312.11805.

[14] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL https://arxiv.org/abs/2403.08295.

[15] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 2555–2565, 2019.

[16] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

[17] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control, 2022. URL https://arxiv.org/abs/2203.04955.

[18] Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

[19] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies, 2023. URL https://arxiv.org/abs/2304.10573.

[20] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning (CoRL)*, volume 270 of *Proceedings of Machine Learn-*

ing Research, pages 2679–2713, November 2025.

[21] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

[22] Pavan Kumar, Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. Fastvlm: Efficient vision encoding for vision language models, June 2025.

[23] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

[25] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time, 2022. URL https://arxiv.org/abs/2210.06407.

[26] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Vaibhav Srivastav, Joshua Lochner, Hugo Larcher, Mathieu Morlon, Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Smolvlm: Redefining small and efficient multimodal models, 2025. URL https://arxiv.org/abs/2504.05299.

[27] OpenAI. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

[28] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

[30] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2004. ISBN 038721240X.

[31] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017. URL https://arxiv.org/abs/1609.04747.

[32] Oleh Rybkin, Chuning Zhu, Anusha Nagabandi, Kostas Daniilidis, Igor Mordatch, and Sergey Levine. Model-based reinforcement learning via latent-space collocation. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pages 8691–8702, 2021.

[33] Tom Silver, Rohan Chitnis, Nishanth Kumar, Willie McClinton, Tomas Lozano-Perez, Leslie Pack Kaelbling, and Joshua Tenenbaum. Predicate invention for bilevel planning, 2025. URL https://arxiv.org/abs/2203.09634.

[34] Gemini Robotics Team. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer, 2025. URL https://arxiv.org/abs/2510.03342.

[35] OpenAI Team. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

[36] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/arXiv.2302.13971. URL https://arxiv.org/abs/2302.13971.

[37] Grady Williams, Paul Drews, Brian Goldfain, James M. Rehg, and Evangelos A. Theodorou. Aggressive driving with model predictive path integral control. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1433–1440, 2016. doi: 10.1109/ICRA.2016.7487277.

[38] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, October 2023.

[39] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

[40] Chuning Zhu, Max Simchowitz, Siri Gadipudi, and Abhishek Gupta. Repo: Resilient model-based reinforcement learning by regularizing posterior predictability. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=OIJ3VXDy6s.

[41] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

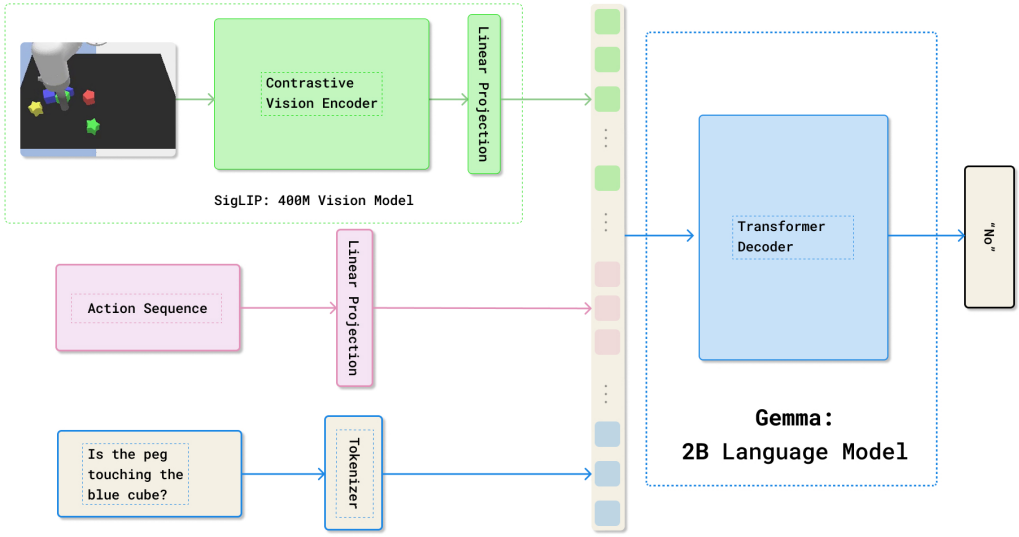

Fig. 7: Architecture of Semantic World Model

VI. Appendix

A. Model Architecture and Training Details

Fig. 7 shows the architecture of Semantic World Model. We use the Paligemma 3B checkpoint as our base model. The only new component we introduce is a linear projection matrix that is dimension `act_dim` $\times$ `2048` where 2048 is the embed dimension of the Gemma model. We perform full weight fine-tuning on all model parameters using a linear LR decay starting at $1e^{-5}$ for approximately $24,000$ gradient steps on LangTable and $64,000$ gradient steps for OGBench. We use an effective batch size of 96. Each model is trained on a node comprising 4 AMD Instinct MI250X GPUs (each equipped with 2 MI200 GPU accelerators), resulting in a total training time of approximately 24 hours.

B. Baselines and Hyperparameters

IDQL [19] is an offline RL method that applies implicit Q-learning to reweight a behavior diffusion-based policy. We use the base diffusion policy architecture for SWM as the policy for IDQL, except with an action horizon of 8 instead of 16. For the Q and Value functions in IDQL, we only condition on the current observation.

For the AVD baseline, we train a latent action-conditioned transformer video diffusion model, based on the architecture of Unified World Models [41], without the action prediction head. Since our semantic world model is trained to answer questions with a variable action horizon, we use SWM to extract semantic information from the predicted future frame by including zero actions. Due to the computational cost of running the AVD forward and then using the generated frame for VQA, we are unable to run this baseline with a high number of samples. Since the MPPI initial samples were initialized from the base policy, we perform 10 iterations of MPPI with 16 samples to get our final action prediction. Each AVD run takes around 10 hours on a single GPU.

The hyperparameters used for the base diffusion model, the IDQL algorithm, and the AVD model are detailed in Tab. XV. The only difference across environments is the size of the input image. All models are trained with the AdamW optimizer [24].

1) Latent World Models: We evaluated a latent model-based RL method, Dreamer [16], on our LangTable tasks. Across three offline tasks, we found that it is unable to solve them, achieving an average success rate of 1.33% from the same offline dataset and sparse rewards used for our SWM, IDQL, and AVD methods. We validated our results using the online version of Dreamer with fixed initialization and dense rewards. This variant achieves 100% success, indicating Dreamer's reliance on online exploration data and dense rewards.

We found similar behavior evaluating the official implementation of TDMPC2 [18], modified to use the sparse offline dataset. It achieved on average across three language table tasks a 10% success rate and struggled to learn effectively without dense reward signals or active exploration.

C. Environments and Tasks

1) Environment Details: Fig. 4 shows an example of each type of task we used to evaluate SWM. In Fig. 8, we provide examples of out-of-distribution configurations used to evaluate the generalization capabilities of SWM. More details about each environment and task are discussed below.

LangTable The LangTable environment has a control frequency of 10 Hz. For each task, we terminate each episode after 120 environment steps. Our observation space is a single `180×320` RGB image of the table. The action space is xy delta poses, ranging from -.03 to .03. Our reach block task is

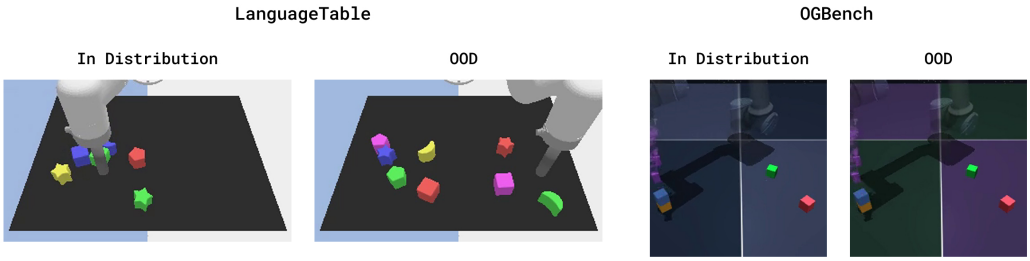

Fig. 8: Out-of-distribution configurations for the evaluation tasks. LangTable is configured to have OOD block/color combinations. OGBench is configured to have a different color background.

marked as a success if the peg made contact with the target block. The separate block task is marked as a success if the L2 distance between the target block and the blocks to separate it from is over .1 M. For pushing blocks together, the episode is marked as a success if the L2 distance between the two target blocks is less than .075. The expert and noisy demonstrations used for our offline dataset and expert diffusion dataset are collected on environment seeds 0-300, and we evaluate on seeds 6000-6050. For the SWM improvement, we use an action chunk of 8, a gradient learning rate of 0.02, 10 planning iterations, and execute 4 out of the 16 predicted actions before replanning. We use a gradient clipping of 1 before updating each action during planning.

OGBench We use the cube environment as the basis for our tasks. This environment has a control frequency of 10Hz, and we terminate each episode after 200 steps. Our observation space is a single 224×224 RGB image. The action space is 5-dimensional, comprising of delta xyz and orientation, and a gripper action. For the ReachCube task, we measure success as the gripper pads touching the cube. For our cube stacking task, we initialize all block poses randomly and then define success as the first cube being stacked on top of the second cube, with a gap between the top cube and the robotic gripper. The expert and noisy demonstrations used for our offline dataset and expert diffusion dataset are collected on environment seeds 0-300, and we evaluate on seeds 6000-6050. For the SWM improvement, we use an action chunk of 8, a gradient learning rate of 0.2, 20 planning iterations, and execute 4 out of the 16 predicted actions before replanning. We use gradient clipping of 10 before updating each action during planning.

2) Question-Answer Dataset Curation: We precompute the future QA pairs for our offline dataset. For each state, we sample four different action horizon lengths between 0 and 20, and generate a set of questions for each sampled horizon. Tab. V shows the question types and an example of each question type on both the LangTable and OGBench environments.

For each question type, we also use multiple variations in wording. For example, for *block touching* questions, given two blocks {block1} and {block2}, we use:

- Is the {block1} touching the {block2}?
- Are the {block1} and {block2} blocks in contact with each other?
- Is there contact between the {block1} block and the {block2} block?
- Does the {block1} touch the {block2}?
- Is the {block1} block in physical contact with the {block2} block?
- Are the {block1} and {block2} blocks touching each other?
- Is the {block1} and {block2} directly touching?
- Do the {block1} and {block2} blocks meet?

3) Task Specification: For each task, we use a fixed set of questions and answers to specify the goals. All of our tasks are single-subgoal tasks except the stack cube task, which has two goals. In order to create a multi-step task for LangTable, we use two subgoals of independent Block to Block tasks, and use the SWM to pick the behavior policy and the subgoal to use. The questions, answers, and weights for all tasks are in shown Tab. VI.

D. Full Improvement Results

We provide the full improvement results corresponding to Fig. 5 in the experiments section.

E. Ablations

1) Ablation on question weights: We ablate the inclusion of question weights across all of the in-distribution LangTable and OGBench tasks. We found that removing weights decreased performance by an average of 2.4% in LangTable and increased performance by 3.3% in ogbench. Full results are in Tab. VIII.

2) Ablation on planning without base policy: To evaluate whether SWM can plan effectively without relying on a base policy, we conducted an ablation study on three LangTable pushing tasks. In this setting, the initial action sequence was sampled uniformly from the environment's action space, removing any prior structure provided by a base policy. Despite the lack of a warm start, SWM was able to successfully plan under these randomly initialized trajectories, achieving success rates of 46%, 50%, and 58% on the tasks of pushing the yellow pentagon to the red moon, pushing the red moon to the green star, and pushing the yellow star to the blue cube. These results demonstrate that the gradients are reasonable even from random initializations.

TABLE V: Question types and examples for LangTable and OGBench

Type	Example
LangTable	
Block touching	Is the red star touching the blue cube?
Peg to block	Is the green cube next to the peg?
Block board position	Is the red star in the center of the board?
Peg block relative direction	Is the peg above the red cube block?
Block to block relative direction	Is the red star to the right of the blue cube?
Block move direction	Did the red cube move left?
Block move	Did the red star block move?
Peg move direction	Did the robotic peg move downward?
Block to block closer	Are the red star and blue cube closer together?
Peg to block closer	Is the robotic peg closer to the red cube?
OGBench	
Cube grasped	Is the red cube grasped by the robot?
Gripper touching block	Is the blue cube touching the robot gripper?
Block touching block	Is the green cube touching the yellow cube?
Block on top of block	Is the red cube on top of the blue cube?
Gripper closer to block	Is the gripper closer to the green cube?
Block closer to block	Is the red cube closer to the blue cube?

TABLE VI: QA pairs used for task rewards

Task	Question	Weight	Desired Answer
Reaching LT	Is the robotic peg touching the {target_block}?	0.8	Yes
	Is the robotic peg closer to the {target_block}?	0.2	Yes
Reaching OG	Is the robotic gripper touching the {target_block}?	0.8	Yes
	Is the robotic gripper closer to the {target_block}?	0.2	Yes
Separate Blocks	Is the robotic peg touching the {center_block}?	0.6	Yes
	Is the {avoid block} touching the {center block}?	0.4	No
Block to Block	Is the {first_block} touching the {second_block}?	0.8	Yes
	Are the {first_block} and the {second_block} closer together?	0.2	Yes
Cube Stacking	**Subgoal 1: Pick up the first cube**		
	Is the robot grasping the {first_block}?	1.0	Yes
	Subgoal 2: Stack the blocks		
	Is the {first_block} on top of the {second_block}?	0.6	Yes
	Is the robot grasping the {first_block}?	0.4	Yes

3) Ablation on question phrasings: To evaluate the robustness of SWM to different question phrasings, we conducted an ablation measuring performance under both in-distribution and out-of-distribution question phrasings. For each task, we evaluated SWM using two different phrasings seen during training and two novel OOD phrasings not present in the SAQA dataset. As shown in Table IX, SWM maintains strong performance across all phrasing variants, with only minor drops under OOD formulations. These results demonstrate SWMs robustness to new question phrasings.

4) Automatic question generation: To validate the broader applicability of SWM beyond simulators with privileged state information, we evaluate an automatic dataset generation pipeline using a VLM to provide supervision for question answers. We evaluate the accuracy of Gemini Embodied Reasoning [34] in providing question-answer supervision on both LangTable and OGBench. In addition, we have measured the accuracy on a small set of manually annotated question-answer pairs from the Droid datasets. Across both settings, Gemini achieved strong accuracy compared to ground-truth answers, indicating that frontier VLMs show a path towards generating the SAQA dataset without relying on oracle information in both sim and real. Detailed results are in Tab. XI, XII, and XIII. For each question, we issued multiple queries to the model, counted each response as a vote, and selected the answer with the highest vote count.

5) Automatic Task Decomposition: We evaluate the feasibility of using a VLM for automatic high-level decomposition. We find that for long-horizon tasks, GPT 5.1 [35] is able to break the task into subtasks and create a set of questions and desired answers to plan with. We observe that the GPT-generated questions were also more varied in phrasing. When paired with SWM planning, these planning parameters perform comparably to our oracle question-answer set planning results. Results are shown in Tab. XIV

We used structured JSON output with a full prompt text of "You are a robotic agent planning to push blocks around on a table. Break your task down into key information and brief, absolute and relative questions. Question examples we've trained on: Is the red star touching the blue cube? Is the green cube next to the peg? Is the red star in the center of the board? Is the peg above the red cube block? Is the red star to the right

TABLE VII: **Improvement Results.** SWM model improvement results on planning tasks in LangTable and OGBench on in-distribution scenes. Reported success rates over $n = 50$ seeds with 95% confidence intervals (normal approximation). The top tasks are LangTable and the bottom tasks are OGBench.

Task	Base Policy	IDQL	AVD	SWM
Push Green Cube to Blue Moon	6% ± 6.6	8% ± 7.5	48% ± 13.8	**78%** ± 11.5
Push Red Moon to Green Star	18% ± 10.6	8% ± 7.5	44% ± 13.8	**80%** ± 11.1
Push Red Pentagon to Blue Moon	14% ± 9.6	12% ± 9.0	38% ± 13.5	**80%** ± 11.1
Push Yellow Pentagon to Red Moon	18% ± 10.6	8% ± 7.5	34% ± 13.1	**86%** ± 9.6
Push Yellow Star to Blue Cube	16% ± 10.2	10% ± 8.3	62% ± 13.5	**84%** ± 10.2
Stack Blue Cube on Yellow Cube	52% ± 13.8	8% ± 7.5	50% ± 13.9	**82%** ± 10.6
Stack Blue Cube on Green Cube	44% ± 13.8	16% ± 10.2	46% ± 13.8	**84%** ± 10.2
Stack Yellow Cube on Red Cube	40% ± 13.6	24% ± 11.8	44% ± 13.8	**62%** ± 13.5

TABLE VIII: **Ablation on question weights.** Success rates for SWM with weights vs. SWM without weights on LangTable and OGBench tasks. Reported over $n = 50$ seeds with 95% confidence intervals.

Task	SWM (with weights)	SWM (no weights)
Push Green Cube to Blue Moon	78% ± 11.5	72% ± 12.4
Push Red Moon to Green Star	80% ± 11.1	78% ± 11.4
Push Red Pentagon to Blue Moon	80% ± 11.1	82% ± 10.6
Push Yellow Pentagon to Red Moon	86% ± 9.6	88% ± 9.0
Push Yellow Star to Blue Cube	84% ± 10.2	76% ± 11.8
Stack Blue Cube on Yellow Cube	82% ± 10.6	82% ± 10.6
Stack Blue Cube on Green Cube	84% ± 10.2	78% ± 11.4
Stack Yellow Cube on Red Cube	62% ± 13.5	78% ± 11.4

TABLE IX: **Ablation on Question Phrasing.** Success rates of SWM under in-distribution (ID) and out-of-distribution (OOD) task phrasings. Reported with 95% confidence intervals.

Task	Base Policy	ID 1	ID 2	OOD 1	OOD 2
Push Red Moon to Green Star	18% ± 10.6	72% ± 12.4	88% ± 9.0	84% ± 10.2	78 ± 11.5%
Push Yellow Star to Blue Cube	16% ± 10.2	78% ± 11.5	86% ± 9.6	84% ± 10.2	72% ± 12.4
Push Yellow Pentagon to Red Moon	18% ± 10.6	88% ± 9.0	88% ± 9.0	86% ± 9.6	76% ± 11.8

of the blue cube? Did the red cube move left? Did the red star block move? Did the robotic peg move downward? Are the red star and blue cube closer together? Is the robotic peg closer to the red cube?"

6) Visualization of Attention Maps: We provide additional visualizations of the attention map. In Fig. 10, we visualize the average attention scores from language tokens to image tokens on a consecutive trajectory. We find that different layers capture different semantic information. For example, layers 4 and 6 attend to the red moon and the blue block, whereas later layers also attend to the peg, likely because of the need to reason about the result of actions. In Fig. 11 we visualize the attention map in layer 4 on different trajectories, showing that the layer consistently attends to the correct objects.

7) Visualization of Gradient-Based Planning: We visualize the gradient-based planning procedure in Fig. 9. As planning iteration progresses, the candidate action sequence gradually extends to pushing the red pentagon to the blue moon, approaching the optimal trajectory over successive gradient steps.

8) Planning Efficiency: We measure the per-action-chunk planning time of MPPI and our gradient-based method in LangTable on a single GPU shown in Tab. X. MPPI is configured with eight iterations of 16 samples, and gradient-based planning uses 10 iterations on a single candidate trajectory; both methods use a reward sub-chunk size of 8 and a horizon of 16. MPPI's success rate scales with the number of samples and optimization iterations; for our main planning results in Tab. I we use 30 iterations of 192 samples (Tab. XV), which increases the per-action-chunk cost roughly proportionally.

TABLE X: Planning speed comparison between MPPI and gradient-based planning, measured on a single GPU in the LangTable environment.

Method	**Time per action chunk (s)**
MPPI	4.48
Gradient-based	1.56

TABLE XI: Gemini-ER Accuracy on LangTable

Question Type	Accuracy
block_touching	0.93
peg_to_block	0.89
block_closer	0.88
Average	**0.90**

TABLE XII: Gemini-ER Accuracy on OGBench

Question Type	Accuracy
cube_grasped	1.00
block_ontop_block	0.95
block_touching_block	0.97
block_block_closer	0.75
Average	**0.92**

TABLE XIII: Gemini-ER Accuracy on DROID

Question Type	Accuracy
claw_hold	0.89
obj	0.91
obj_relative	0.82
Average	**0.87**

TABLE XIV: Success rate (%) on long-horizon LangTable tasks with and without LLM-derived task decompositions over 50 seeds. MS1 - red pentagon to blue moon, yellow pentagon to red moon. MS2 - yellow star to blue cube, yellow pentagon to red moon. MS3 - yellow star to blue cube, red pentagon to blue moon. MS4 - green cube to blue moon, yellow pentagon to red moon.

Task	Base Policy	SWM+grad	SWM+grad + LLM planning
MS1	6	50	62
MS2	4	66	42
MS3	4	54	52
MS4	2	54	44
Average	**4**	**56**	**50**

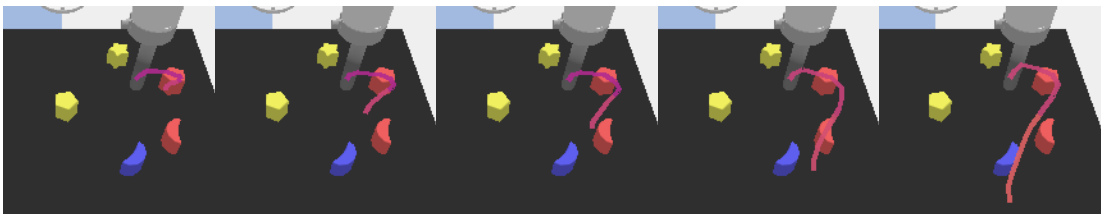

Fig. 9: Visualization of gradient-based planning on the LangTable - Red Pentagon to Blue Moon task. The initially proposed action sequence is on the left, and updates to this action sequence go progressively to the right, approaching the optimal trajectory over successive gradient steps.

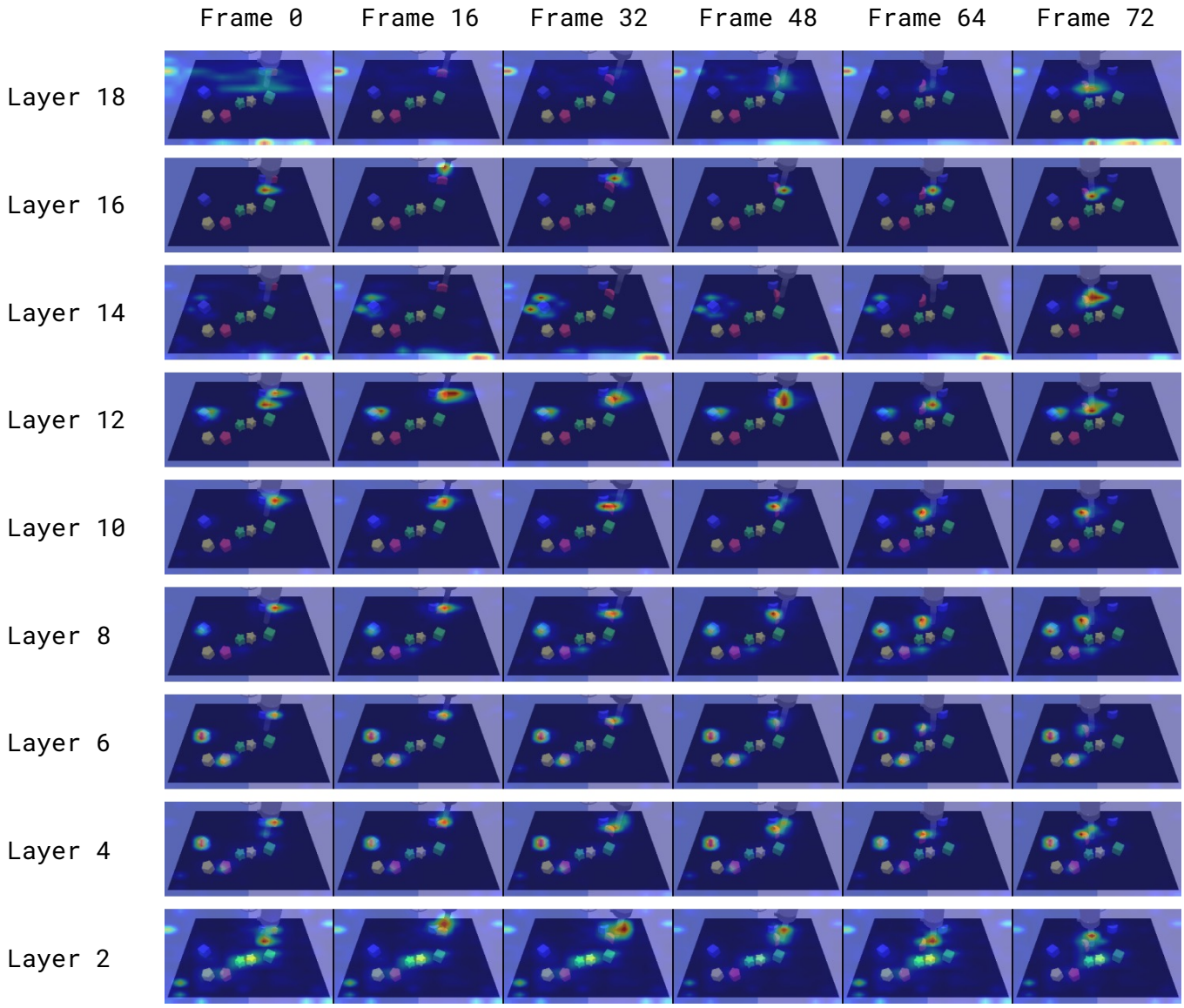

Fig. 10: Attention maps in different layers of SWM. Question: “Is the red moon touching the blue block?”

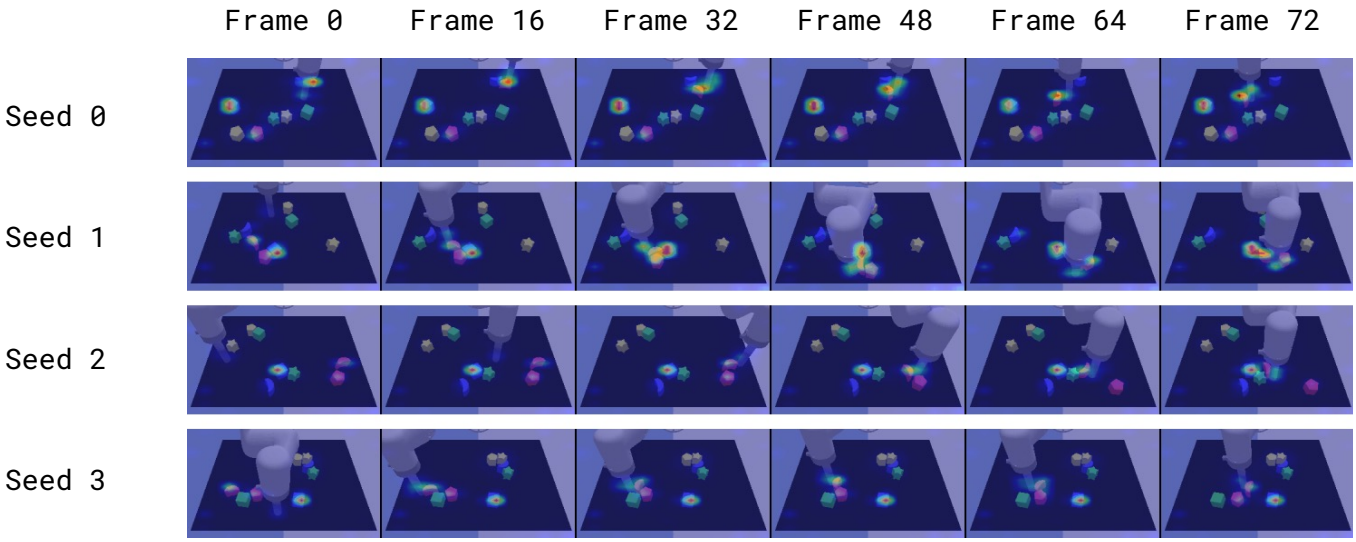

Fig. 11: Attention maps for different trajectories. Question: "Is the red moon touching the blue block?"

TABLE XV: Hyperparameters for MPPI, IDQL, Diffusion, and AVD Model

Diffusion	
Batch size	128
Epochs	100
Action horizon	16
Observation horizon	2
Diffusion iters	100
Eval diffusion iters	10
Traj end padding (steps)	12
IDQL	
Gradient steps	250,000
Batch size	128
IQL τ	0.8
Test time samples	1000
Temperature	0.5
Discount (γ)	0.99
Critic hidden dim	256
Critic learning rate	0.0003
Num layers	3
AVD Model	
Embed dim	768
Vision backbone	ViT-B/32
Timestep embed dim	512
Latent patch shape	[2,2,2]
Num Transformer Layers	12
Num heads	12
Train steps	1000
Inference steps	50
Total steps	100,000
Global batch size	288
Learning rate	1e-4
Weight decay	1e-6
MPPI	
Num Iterations	30
Samples	192
Temperature	.2