SPATIAL CONCEPT BOTTLENECK FOR INTERPRETABLE ROBOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot learning models have been proliferating with increasingly stronger results for both generalist and specialist policies. However, true deployment potential of these learned policies is still unknown. As the community closes in on translational research, it is now ever more crucial to develop mechanisms that not only enable interpretability but intervention as well, e.g., through concept bottleneck models. In this work, we present an initial investigation along the lines of an interpretable spatial concept bottleneck that acts as a meaningful intermediate representation between a perception-planning backbone and an action head. Specifically, we define a spatial concept as a geometric relationship between entities corresponding to the task and the robot, which can be computed directly from the scene's geometry, and instantiate it for both navigation and manipulation. We present a series of experiments demonstrating that such a spatial concept bottleneck maintains original performance (without the bottleneck), while serving as an interpretable interface through which humans or VLMs can intervene to correct or steer the robot. Videos and results are available at `https://spatial-cbm.github.io`.

1 INTRODUCTION

Robot learning methods for predicting actions or control commands are predominantly end-to-end, for example, Vision Action Policies, Vision Language Action (VLA) Models (Black et al., 2024; Intelligence et al., 2025; Kim et al., 2024), and World Action Models (WAMs) (Wang et al., 2026; Agarwal et al., 2025). These robot models have demonstrated performance scaling in proportion to data and model size, akin to the trends observed in Multimodal Large Language Models (MLLMs) (Anthropic, 2026; Yang et al., 2025). Unlike text generation, robot actions are spatially grounded and strongly correlated with input images that represent the physical task space. This very distinction inherently offers visual interpretability, and even intervention from humans and Vision Language Models (VLMs). This is generally achieved as a Concept Bottleneck Model (CBM) (Koh et al., 2020), where an interpretable concept layer is inserted between the model backbone and final output layer. While CBM and its variants have been used for vision and language research, it's been largely amiss in robot learning despite being more prone to physical safety failures.

In this paper, inspired by the original CBM work, we propose Spatial Concept Bottleneck as an attempt towards a general framework that leverages scene geometry information in some form to predict a (visually) interpretable intermediate representation that acts as a bottleneck between the perception backbone and the action head. We make the following key contributions:

- a general formulation of a *spatial concept* defined as geometric relationship between entities corresponding to the robot and the environment;
- a high-level architectural structure for what we refer to as a *Spatial CBM*, realized as an intermediary specifically between the perception backend and the action head; and
- a set of Spatial CBM variants and intervention routines that closely follow the original CBM to lay its foundation in robot learning literature.

To the best of our knowledge, this paper is the first to import the well-established ideas and benefits of concept bottlenecks into robot learning. Thus, we deliberately formulate Spatial CBM as closely

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

as possible to the original CBM work so that future works can further build on this, especially with regards to the specific choices in defining a task-relevant spatial concept. In this paper, using our general definition of a spatial concept, we instantiate its two variants respectively for the navigation and manipulation task, and conduct experiments across four different variants of Spatial CBM architectures. Our results show that a spatial concept bottleneck is clearly viable in robot learning, where in some cases the bottleneck improves performance over the lack of it. We further show how an oracle-triggered VLM-based intervention can improve closed-loop navigation performance.

2 Related Works

Interpretable Concept Bottlenecks. Concept bottleneck models (CBMs) (Koh et al., 2020) split a model into a concept predictor and a task predictor that sees only the concepts, so a human can read the intermediate and correct it at test time. Koh et al. (2020) compare four ways to train them: standard (no concept), multitask (the concept as an auxiliary loss), independent (the task predictor trained on true concepts) and sequential (trained on predicted concepts), and show that intervening on mispredicted concepts recovers accuracy. Later work found that soft concepts leak information beyond their meaning (Mahinpei et al., 2021), studied hard concepts and side channels (Havasi et al., 2022), showed that interventions can hurt sequential and joint models and how to choose which concepts embeddings (Shin et al., 2023), and increased concept capacity with concept embeddings (Espinosa Zarlenga et al., 2022). Post-hoc and label-free variants attach the bottleneck to a pretrained model and obtain concepts from a concept bank or a vision-language model, with a residual path for what the concepts miss (Yüksekgönül et al., 2023). These works target classification and regression on independent samples. However, a robot acting in a closed loop setting where each action changes what it sees next would suffer from forwarding errors for a wrong concept. We use concepts that are spatial, such as navigation costmap giving each pixel's distance to the next sub-goal, or the position of the target object and its destination relative to the gripper, so these can be computed directly from the scene's 3D geometry.

Interpretability in End-to-End Robot Learning. End-to-end policies remove most of the handles a human could use to understand or correct a robot's behaviour, and several works have tried to bring some structure back. Object-centric representations such as Slot Attention (Locatello et al., 2020) decompose a scene into object slots, but the slots are learned without supervision and do not necessarily map to quantities a human can name or edit. More recently, VLAs predict human-readable intermediates before acting, such as visual traces (Gu et al., 2024; Lee et al., 2025) or reasoning chains (Zawalski et al., 2024), some of which can be edited at test time (Lee et al., 2025). However, the action head still attends to the image and the instruction, so nothing forces it to follow the intermediate. In navigation, learned policies (Savinov et al., 2018; Shah et al., 2023) and world models (Bar et al., 2025) are typically conditioned on the current observation and a goal or sub-goal image, which leaves little to inspect between perception and action. Recent topological navigation methods instead condition the controller on an explicit costmap over the current view, computed at the object (Garg et al., 2025) or pixel (Garg et al., 2026) level by planning over a prior traversal. Such a costmap is interpretable, but it is computed by a fixed pipeline rather than predicted, and it has not been studied as a bottleneck that can be trained and intervened on in the sense of CBMs (Koh et al., 2020). We predict it with a learned perception backbone and treat it as such a concept bottleneck.

Spatial and 3D Representations in VLAs. Several recent VLAs add 3D spatial information to image-based vision-language backbones to improve precision. SpatialVLA (Qu et al., 2025) injects the depth-lifted 3D position of each patch into the visual tokens and tokenizes actions on adaptive spatial grids. Spatial Forcing (Li et al., 2026a) aligns intermediate visual tokens with the features of a 3D foundation model during training. Depth and 3D tokens are also used as auxiliary prediction targets (Li et al., 2026b), or as part of a predicted future state (Zhang et al., 2025). In CBM terminology, these methods are closest to the multitask variant: the 3D information shapes the features, but it is never an input that a layperson can understand or intervene.

3 Proposed Approach

3.1 Concept Bottleneck Model

We provide a brief background for the original Concept Bottleneck Model (CBM) (Koh et al., 2020). The general idea is to split the black-box predictive structure of a neural network into two parts: *i)* a concept predictor g maps the inputs x to concept $\hat{c} = g(x)$, and *ii)* a task predictor f maps the concept $\hat{c}$ to $\hat{y} = f(\hat{c})$ in the output space, where $y/\hat{y}$ and $c/\hat{c}$ denote true/predicted output and concept values. The concepts naturally form a bottleneck between the input and the output. An ideal concept design ensures that the concepts are not only interpretable by humans (ideally even by a layperson), but also preserve sufficient expressiveness to adequately complete the task. Koh et al. (2020) defined four different variants for end-to-end learning with regards to the bottleneck: `standard`, `multitask`, `independent`, and `sequential`, as also summarized in Table 1. Both standard and multitask methods are concept bottleneck free and predict $z = g(x)$ directly with a task prediction loss $\mathcal{L}_Y(f(z), y)$. Multitask has an additional auxiliary loss $\mathcal{L}_C(h(z), c)$ coming from the concept head attached on standard's backbone. Note that multitask generally improves the standard case with concept loss as a regularizer but cannot use the concept directly to predict the task outputs. The sequential and independent models both have the concept bottleneck and differ only in the way the concept is used by the task predictor. Sequential's task predictor is trained on the predicted $\hat{c}$ while independent swaps out the training for the true concept c.

Table 1: **CBM variants** (Koh et al., 2020) and our names. $\hat{c} = g(x)$ are predicted and c true concepts, z features (no bottleneck); "then" marks two stages, "and" two separate trainings, and multitask's $\mathcal{L}_C$ acts on an extra head $h(z)$. GT is our reference for comparisons.

Ours	**CBM**	f **on** train	f **on** test	**Loss**
GT	–	c	c	$\mathcal{L}_Y$
Spatial CBM	Sequential	$\hat{c}$	$\hat{c}$	$\mathcal{L}_C$, then $\mathcal{L}_Y$
Spatial CBM*	Independent	c	$\hat{c}$	$\mathcal{L}_C$ and $\mathcal{L}_Y$
Direct Action	Standard	z	z	$\mathcal{L}_Y$
Spatial CAM	Multitask	z	z	$\mathcal{L}_Y + \lambda\,\mathcal{L}_C$

3.2 Spatial Concept Bottleneck

CBM was originally designed for classic machine learning tasks where the network's decisions do not directly change the state of the machine or the environment. In contrast, robot learning methods predict actions which have an immediate consequence in the physical world, and pose a much higher requirement for safety constraints. This motivates the need for interpretability and intervention research in robot learning akin to CBMs. At the same time, physical world offers more avenues for interpretability as robot's actions mimic that of a humans' which inherently carry a spatial-semantic understanding of the world. This can be in the form of *task-relevant spatial affordances* tagged to whatever is being observed in the environment, e.g., navigation affordance of a door vs a couch depending on the task of exiting vs relaxing, or manipulation affordance of an object depending on whether it is to be picked or placed on. Such spatial understanding can naturally form a concept bottleneck. Secondly, the trajectory a robot takes is not independently and identically distributed. Each action changes either the robot location or the scene or both, which impacts the way the next timestamp of the world looks, which further changes its action decision. Essentially, errors as well as interventions may both compound in the closed-loop scenario. While open-loop evaluations are helpful for validating results, they are no longer sufficient for embodied tasks, and we must rely on closed-loop evaluations.

Formally, we define the inputs, concepts, and outputs for each time step t. Input $x_t = (o_t, H)$ contains the live observation o_t and the task context H, which may be a token sequence, other images, and other task-relevant items. Given the multimodal nature of most robot learning pipelines, we propose a spatial concept bottleneck as an intermediate representation that precedes an action head and is in itself a function of a multimodal perception (and planning) backbone. Concretely, the perception backend g predicts the spatial concept $\hat{c}$ and the actor/controller head maps the concept and optionally the proprioceptive state ρ_t to action $\hat{y}_t = f(\hat{c}_t, \rho_t)$. The robot executes $\hat{y}_t$, observes o_{t+1}, and the loop repeats. We formulate the ground truth concept c_t as a function ψ defined as an interaction between two spatial entities stemming respectively from the task and the robot (embodiment/observation). The task mandates an *exocentric entity* $\{a_t^1, \ldots, a_t^K\} \subset \mathbb{R}^3$, which serves as the task-relevant reference like navigation sub-goals or destination points for the gripper. The robot provides an *egocentric entity* r through its embodiment or observation of the task space.

$$c_t^k = \psi\big(a_t^k, r\big), \qquad r \in R_t, \tag{1}$$

Thus, a spatial concept c can potentially better adapt to both task-relevant understanding and sensory/state inputs, since the subsequent action prediction is conditioned on this bottleneck. The task

context H defines the anchors but we restrict its information in the spatial concept bottleneck style via ψ. R_t and ψ are left up to us as the spatial concept bottleneck design choices.

3.2.1 NAVIGATION

In this section, we describe how we use spatial concept bottleneck for the navigation task, which requires a thorough spatial understanding via inherent components relating to robust perception, localization and path planning. We particularly test for learning-based navigation via subgoal-conditioned action prediction (Shah et al., 2023; Savinov et al., 2018; Garg et al., 2025). The robot's live view contains the spatial information needed to predict actions, while the task context is available in the form of subgoal images/patches. During inference, this is executed iteratively to reach a long-horizon goal, which often leads to compounding errors in a closed-loop setting, potentially derailing the robot whenever an action decision goes wrong. We define a spatial concept bottleneck for the aforementioned navigation task as a visual representation of navigation affordances in robot's current view. This is inspired by the existing works (Garg et al., 2025; 2026), including broadly similar ideas (Apicella et al., 2025; Yoo et al., 2024; Sun et al., 2025), where object- (Garg et al., 2025) or pixel-level (Garg et al., 2026) path planning costs are used to condition a learned action head.

Concept formalization. We formally define costmaps for navigation under the spatial concept bottleneck framework, based on Eq. 1, as the following.

$$C_t(u) = \psi\big(a_t^1, X_t(u)\big) = \big\|a_t^1 - X_t(u)\big\|_2 \tag{2}$$

Here, we have the sub-goal pixel as the only anchor ($K = 1$), and compute $\psi(a, r) = \|a - r\|_2$ and we compute the egocentric entity r_t as the 3D point for each pixel of the current observation u using the perception backend for the current view X_t. We average this over the patches of a 16×16 grid on the current view and min-max normalize per image, as in ObjectReact (Garg et al., 2025). The part till a sub-goal is short enough for Euclidean approximation and the part beyond cancels out for a particular view. The Euclidean approximation simplifies our requirements since we don't need the navigation mesh for sub-goal traversals.

3.2.2 MANIPULATION

We extend our study to manipulation using pick and place tasks to establish an initial groundwork for robust spatial concept abstraction. In navigation, we keep the world frozen while the robot moves around, and in manipulation, we keep the robot's location frozen while its arms and objects move.

Concept formalization. For manipulation, we define two exocentric spatial entities ($K = 2$): source and destination, corresponding respectively to the pick and place objects. The source object defines where to act as a_t^1 and the destination defines where to place as a_t^2. We use a single egocentric spatial entity that represents the end effector position r_t. Following Eq. 1, we define concept as the source (destination) position a_t^k relative to the gripper position r_t, where β represents a discretization function to suit the input requirements of the action head.

$$c_t^k = \psi(a_t^k, r_t) = \beta\left(a_t^k - r_t\right), \qquad k \in \{1, 2\}. \tag{3}$$

3.3 INTERVENTION

The key benefit of the bottleneck structure of CBMs is that they allow intervention of human-interpretable concepts at test time. They provide insight into the model's intent which it *has* to use to generate outputs. An added benefit is that intervention by VLMs, human experts, or even layperson allow the deployed system to learn the concept predictor g better based on feedback, in a style similar to DAgger (Ross et al., 2011). In simulation, the true exocentric entities, and hence the true concept c_t, come from the simulator's state. Since `independent` is trained on true concepts and `sequential` on predicted ones, we expect the former to gain more when an intervener provides true values.

Formalization. The intervener is provided all information of x_t and the predicted spatial concept $\hat{c}_t$. The oracle has additional access to c_t. Based on the design of the spatial concept bottleneck,

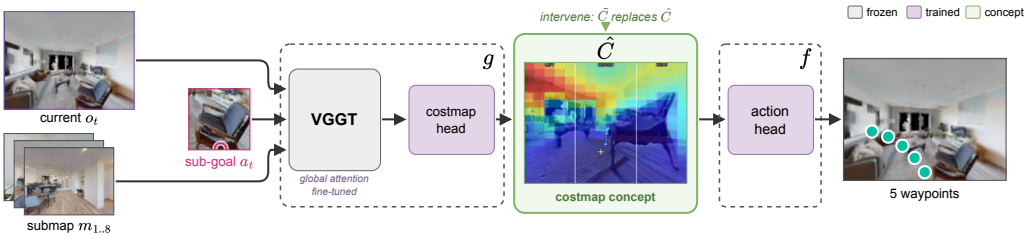

Figure 1: Spatial concept bottleneck for navigation. The concept predictor g (VGGT (Wang et al., 2025) with fine-tuned global attention and a costmap head) reads the current view o_t, the eight localized traversal frames $m_{1..8}$ and the sub-goal a_t, and predicts the costmap $\hat{C}$. The task predictor f (the action head) reads only the costmap and outputs five waypoints.

the intervener provides exocentric entities $\tilde{a}_t^k$ for a subset $S_t \subseteq \{1, \ldots, K\}$. The robot's own ego entities R_t are unchanged. We are thus able to re-generate the entire concept on intervention without changing all the exocentric entities.

$$\tilde{c}_t^k = \begin{cases} \psi(\tilde{a}_t^k, r), & k \in S_t, \\ \hat{c}_t^k(r), & k \notin S_t, \end{cases} \qquad r \in R_t, \qquad \tilde{y}_t = f(\tilde{c}_t, \rho_t). \tag{4}$$

4 Experiments and Results

4.1 Navigation

Experimental Setup. We follow the inputs and outputs of learned topological navigation in the Habitat simulator (Savva et al., 2019). The robot gets a video of a previous traversal and has the freedom to pre-process it as it likes. It is then required to complete the following tasks, which set different goals, following ObjectReact (Garg et al., 2025). *Imitate*, reach the end of the traversal. *Alt-goal*, reach an object seen in the video but not at the end frame. *Shortcut*, reach the end faster by avoiding the detour in the video. *Reverse*, reach the start by starting at the end.

We use the input structure of ObjectReact (Garg et al., 2025) and MASt3R-Nav (Garg et al., 2026), where the robot localizes against the traversal frames using eight nearest to it, and the global planner assigns the length of path to the goal for each pixel. The walkable pixel with the lowest path length becomes the sub-goal. The perception backend, equivalent to the concept predictor g of the CBM, reads the live camera input along with the eight frames of context and the sub-goal, to generate a costmap. Figure 1 shows the full pipeline.

We follow the output formatting of GNM (Shah et al., 2023) based controllers, similar to their usage in previous works. The controller, equivalent to the task predictor f of the CBM, outputs five future waypoints which are turned into linear and angular velocities. The controller is trained with imitation regression loss.

Results. Table 2 ablates the Spatial CBM layer as defined in Section 3.2. GT-Geodesics clearly establishes that action prediction using true spatial concept values for both training and testing yields nearly perfect results. This implies that having the spatial concept as an intermediary is in itself not a performance bottleneck, rather it is the lack of the ability to predict perfect concepts. We also report results for GT-ED which mainly indicate that geodesic distances provide a better signal than Euclidean distances, and that the perception backend, VGGT, which is trained to learn Euclidean geometry, does require learning geodesic geometry via finetuning to be performant on the navigation tasks. In general across all four navigation tasks, we observe that Spatial CBM is on par or better than predicting Direct Actions. On the Imitate task, which closely resembles the teach-and-repeat style of training data of ObjectReact benchmark, direct action prediction works better than having the spatial bottleneck. This is in line with the observations from the original CBM work. However, a surprising trend is the improved performance of Spatial CBM on the more challenging navigation tasks, perhaps somewhat similar to the findings of Concept Embedding Models (CEMs) (Espinosa Zarlenga et al., 2022). This can be attributed to the *explicit* induction of spatial understanding via the bot-

Table 2: **Spatial CBM ablation** on four diverse navigation tasks. We report SPL and SSPL for different variants of Spatial CBM. Best non-oracle scores are in **bold**.

Method	Imitate		Alt Goal		Shortcut		Reverse		*Average*	
	SPL	SSPL	SPL	SSPL	SPL	SSPL	SPL	SSPL	SPL	SSPL
GT-Geodesics	$96.43_{\pm 2.34}$	$98.78_{\pm 1.72}$	$86.53_{\pm 4.07}$	$91.16_{\pm 3.20}$	$88.42_{\pm 0.20}$	$91.67_{\pm 1.19}$	$82.99_{\pm 0.32}$	$91.97_{\pm 0.39}$	$88.59_{\pm 1.63}$	$93.40_{\pm 0.83}$
GT-ED	$72.16_{\pm 2.00}$	$74.58_{\pm 1.53}$	$70.29_{\pm 4.89}$	$78.49_{\pm 5.94}$	$67.76_{\pm 2.28}$	$70.72_{\pm 3.19}$	$68.41_{\pm 2.51}$	$77.53_{\pm 1.12}$	$69.65_{\pm 1.78}$	$75.33_{\pm 1.35}$
Direct Action	$82.61_{\pm 5.77}$	$86.92_{\pm 3.07}$	$59.99_{\pm 13.07}$	$67.84_{\pm 12.07}$	$61.15_{\pm 5.91}$	$71.86_{\pm 4.21}$	$18.41_{\pm 3.28}$	$\mathbf{34.47}_{\pm 2.82}$	$55.54_{\pm 0.47}$	$65.28_{\pm 0.49}$
Spatial CAM	$\mathbf{87.25}_{\pm 0.03}$	$\mathbf{89.70}_{\pm 0.46}$	$28.18_{\pm 7.16}$	$41.45_{\pm 9.01}$	$38.13_{\pm 0.27}$	$42.01_{\pm 0.70}$	$7.86_{\pm 2.25}$	$15.71_{\pm 0.78}$	$40.36_{\pm 1.17}$	$47.22_{\pm 2.00}$
Spatial CBM*	$81.14_{\pm 5.83}$	$86.44_{\pm 2.54}$	$57.81_{\pm 2.94}$	$68.08_{\pm 3.56}$	$66.06_{\pm 3.92}$	$74.20_{\pm 2.97}$	$12.55_{\pm 0.08}$	$27.33_{\pm 2.27}$	$54.39_{\pm 3.19}$	$64.01_{\pm 1.70}$
Spatial CBM	$80.80_{\pm 2.08}$	$86.12_{\pm 1.82}$	$\mathbf{68.27}_{\pm 6.43}$	$\mathbf{73.89}_{\pm 5.33}$	$\mathbf{73.06}_{\pm 2.69}$	$\mathbf{80.29}_{\pm 1.80}$	$\mathbf{19.38}_{\pm 10.22}$	$33.73_{\pm 9.12}$	$\mathbf{60.38}_{\pm 5.36}$	$\mathbf{68.51}_{\pm 4.52}$

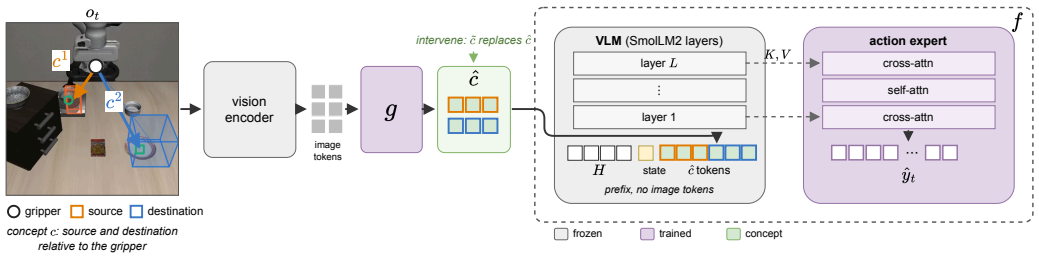

Figure 2: Spatial concept bottleneck on SmolVLA. Left: the concept c is where the source (c^1) and the destination (c^2) are relative to the gripper. The frozen vision encoder feeds only the concept predictor g, whose prediction $\hat{c}$ enters the task predictor f as concept tokens, after the instruction H and the state; the image tokens are removed. f is SmolVLA's frozen language layers and its trained action expert. An intervention replaces $\hat{c}$ with $\tilde{c}$.

tleneck, which tends to be more invariant to out-of-distribution (OOD) samples encountered more often in tasks such as Shortcut and Reverse. A similar trend is observed for the Spatial CAM where predicting spatial concepts as an auxiliary objective only improves performance on the Imitate task but does not generalize to other challenging tasks - this indicates that the network tends to overfit to direct action prediction more quickly alongside the spatial auxiliary objective. Finally, the comparison between Spatial CBM* and Spatial CBM shows that test-time distribution gap between the true and predicted spatial concepts does not strongly affect concept-condition action prediction, as the average performance drop is primarily due to the most challenging task of navigating in reverse.

4.2 Manipulation

Experimental Setup. We build our manipulation experiments on SmolVLA (Shukor et al., 2025), starting from its released checkpoint fine-tuned on LIBERO (`HuggingFaceVLA/smolvla_libero`). We train on the 500 demonstrations of LIBERO-Spatial (Liu et al., 2023), which consist of ten pick and place tasks with 50 demonstrations each. Since the simulator state is available for every demonstration frame, we re-render each frame at 256×256 for the agent-view and wrist cameras and obtain the ground truth concepts from the same state. The exocentric entities for the spatial concept are the positions of the source bowl a_t^1 and the destination plate a_t^2; the egocentric entity is the end effector position r_t; and β discretizes each axis of the difference in positions into 13 bins on a logarithmic scale (Eq. 3). The concept therefore consists of six values, the x, y and z offsets of the source and of the destination from the gripper. The bins are narrow near the gripper and wide far from it (Figure 3), giving centimetre precision close to the object and a coarse direction elsewhere.

Figure 2 shows how we use the spatial bottleneck inside SmolVLA. The vision-language backbone is kept frozen, and we remove the image tokens from the prefix that the action expert attends to. The prefix therefore contains only the instruction H, the state ρ_t, and six concept tokens, where each concept bin selects a row of a learned embedding table.

Following the navigation experiments, we report the variants of Table 1. Direct Action is the original SmolVLA with its image tokens and without any concept. GT trains and tests the action expert on the true concepts, while Spatial CBM* is trained on the true concepts but tested on the predicted

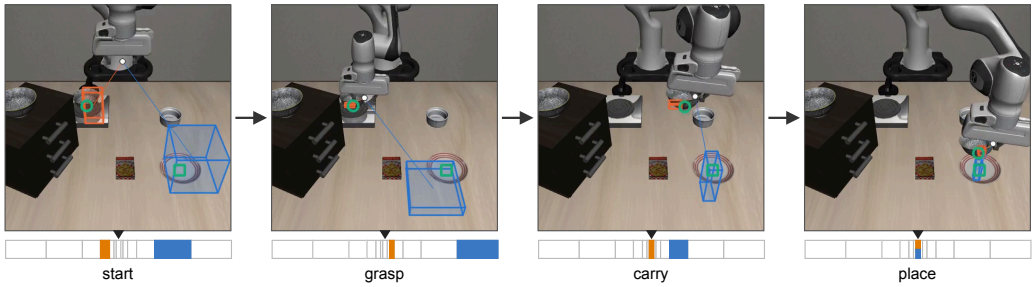

Figure 3: The concept along one rollout (LIBERO-Spatial, true concepts). Each box is the bin β puts the source (orange) or the destination (blue) in, drawn around the gripper. Green represents the object's true position. Below each frame, the 13 bins of one axis to scale, with the source's and the destination's bin. The triangle is the gripper.

Table 3: **Spatial CBM on manipulation** (SmolVLA, LIBERO-Spatial). Success rate (%) for 10 tasks. Best non-oracle scores are in **bold**.

Method	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	*Average*
GT	$100.0_{\pm 0.0}$	$97.5_{\pm 3.5}$	$100.0_{\pm 0.0}$	$95.0_{\pm 0.0}$	$75.0_{\pm 7.1}$	$25.0_{\pm 0.0}$	$97.5_{\pm 3.5}$	$100.0_{\pm 0.0}$	$87.5_{\pm 17.7}$	$70.0_{\pm 7.1}$	$84.8_{\pm 2.5}$
Direct Action	$\mathbf{97.5}_{\pm 3.5}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{97.5}_{\pm 3.5}$	$\mathbf{97.5}_{\pm 3.5}$	$\mathbf{70.0}_{\pm 7.1}$	$\mathbf{27.5}_{\pm 3.5}$	$92.5_{\pm 10.6}$	$\mathbf{95.0}_{\pm 0.0}$	$\mathbf{82.5}_{\pm 3.5}$	$\mathbf{77.5}_{\pm 10.6}$	$\mathbf{83.8}_{\pm 1.8}$
Spatial CBM*	$95.0_{\pm 7.1}$	$95.0_{\pm 0.0}$	$92.5_{\pm 10.6}$	$90.0_{\pm 7.1}$	$65.0_{\pm 0.0}$	$15.0_{\pm 0.0}$	$\mathbf{95.0}_{\pm 7.1}$	$92.5_{\pm 3.5}$	$75.0_{\pm 7.1}$	$57.5_{\pm 3.5}$	$77.2_{\pm 2.5}$
Spatial CBM	$90.0_{\pm 0.0}$	$85.0_{\pm 0.0}$	$95.0_{\pm 0.0}$	$90.0_{\pm 7.1}$	$62.5_{\pm 3.5}$	$12.5_{\pm 3.5}$	$\mathbf{95.0}_{\pm 0.0}$	$87.5_{\pm 10.6}$	$75.0_{\pm 7.1}$	$57.5_{\pm 3.5}$	$75.0_{\pm 0.7}$
Intervention											
Spatial CBM, intervened	$95.0_{\pm 0.0}$	$87.5_{\pm 3.5}$	$97.5_{\pm 3.5}$	$95.0_{\pm 0.0}$	$67.5_{\pm 3.5}$	$17.5_{\pm 3.5}$	$97.5_{\pm 3.5}$	$95.0_{\pm 0.0}$	$82.5_{\pm 10.6}$	$62.5_{\pm 3.5}$	$79.8_{\pm 2.5}$
Ablations											
No concept	$90.0_{\pm 7.1}$	$80.0_{\pm 7.1}$	$87.5_{\pm 3.5}$	$85.0_{\pm 0.0}$	$62.5_{\pm 3.5}$	$15.0_{\pm 0.0}$	$90.0_{\pm 7.1}$	$72.5_{\pm 3.5}$	$62.5_{\pm 3.5}$	$37.5_{\pm 3.5}$	$68.2_{\pm 2.5}$
GT, swapped†	40.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5

ones. Spatial CBM is trained on out-of-fold predicted concepts from 5-fold cross-validation over the demonstrations, so that its training concepts carry the same kind of errors as at test time, where it uses the same concept predictor as Spatial CBM*. We also report two ablations: No concept, which removes the concept tokens altogether, and GT, swapped, which exchanges the source and destination concepts at test time. Each variant is evaluated on 200 episodes per training seed (LIBERO's 20 fixed initial states for each of the ten tasks), with at most 220 steps per episode, executing 10 actions from each predicted chunk.

Results. Table 3 shows that with true concepts, the spatial bottleneck performs on par with Direct Action, which means that the six concept tokens carry enough information for the action expert to complete these tasks. The ablations further show that the action expert actually relies on this information. Removing the concept tokens reduces success by 16.6% points compared to GT, and swapping the source and destination concepts brings the GT policy down to 4.5%, i.e., the robot follows the concept rather than the instruction. With predicted concepts, Spatial CBM* and Spatial CBM drop by 6.6% and 8.8% respectively. Similar to navigation, this indicates that the gap comes from predicting the concept rather than from the bottleneck itself, and intervening on Spatial CBM recovers it by 4.8%. We also note that Spatial CBM does not improve over Spatial CBM* (-2.2 ± 4.2, paired over both seeds); this is in line with the original CBM, demonstrating that there is an accuracy-interpretability trade-off although not drastic.

4.3 Intervention: Navigation

Experimental Setup. We split the spatial concept bottleneck of navigation, the costmap, into three sectors: left, center, and right, which map the three most common commands an intervener would likely give. Each sector is a column which spans five, six, and five pixels respectively of a 16×16 costmap. For the purposes of our experiments, we bound the number of interventions to 16 as we test Spatial CBM and Spatial CBM* approaches.

In line with the original CBM work, we set baseline comparisons with respect to the simulator's ground truth state to extract the exocentric entity a_t^1, which generates the ground truth costmap c_t.

To stress-test our proposed spatial concept in deployment scenarios, we run inference on Claude Opus 5 (Anthropic, 2026) as our choice of the VLM intervener. The relevant task context H is provided via a fixed system prompt, briefly describing its inputs, embodiment and the costmap, and on the visual end, we provide it the last six camera frames taken five steps apart, eight frames from the earlier traversal, the current view, the sub-goal-containing view, and the final destination image.

Most importantly, Claude is provided only the predicted costmap $\hat{c}_t$ (and never the true costmap c_t) to propose an intervention. It is required to output "none" if no change is required, or the proposed sector along with a brief reasoning. The trigger for VLM intervention is tested in two cases, with and without the oracle gate. In the former, we use the oracle to detect differences between the predicted and true costmaps and trigger the intervention. In the latter, we let VLM call the shots on every single decision.

Results. Without edits, Spatial CBM succeeds in 84 of the 126 episodes. The oracle's intervention fixes six episodes and breaks one. When Claude is triggered on oracle disagreement, it fixes ten episodes and breaks three. When Claude is ungated, it still fixes 10 episodes but ends up breaking 16 with eight of them on Alt-goal. Spatial CBM* starts lower at 73 successes, and the oracle fixes 10 episodes but breaks 5, bringing it to 78.

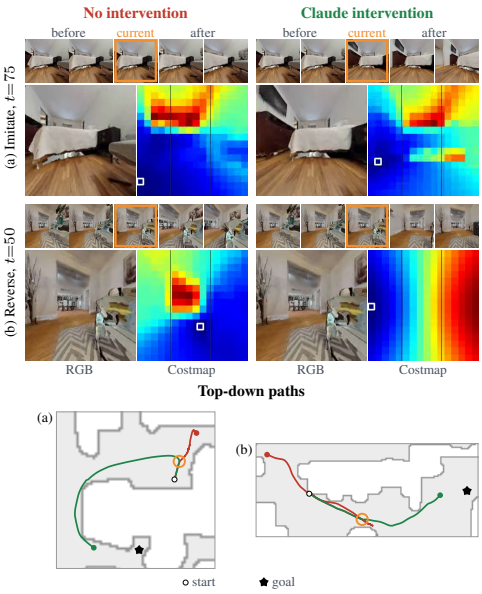

Figure 4: VLM's interventions at a decisive step t, without (left) and with (right) intervention: context frames 10 and 5 steps before and after t (orange: step t), the RGB view and the costmap (blue is low cost; white square: least-cost cell), and the top-down paths of both runs (orange rings: positions at t). (a) Early on in the episode Claude realizes it is looking at the bed, and says, "The bed blocks the right side". Without intervention the robot is never able to leave the room. (b) Claude turns left even though the oracle's sector was right, and the robot reaches the goal.

Figure 4 shows two episode samples where the VLM intervention converts a failed episode into a success by editing only the spatial concept. In the first episode, both runs are initially stuck in front of a bed. Claude reasons (akin to how a human would) that the head of the bed would block the right side of the room and it should turn left, allowing it to complete the rest of the trajectory with no further interventions. In the second episode, Claude turns left towards the open room even though the true concept points to the right.

The underlying theme across these interventions is that spatial concept edits at decisive times can change episode success. It is also evident from the reasoning traces that a VLM is able to "reason" somewhat like a human, and natively understands walls, doorways, and layout of bedrooms. At the very least, it is able to pull out the robot from trivially problematic cases like heading into an obstacle. Note that intervening directly to predict waypoints would require the VLM to directly solve for geometric outputs, whereas intervention based on the visual spatial concept is more human-like and it decouples semantic action planning from geometric waypoint prediction (note that a corrected costmap is performant at predicting correct waypoints).

4.4 Intervention: Manipulation

Experimental Setup. In manipulation, the intervener provides the source or destination point $\tilde{a}_t^k$, and the concept is regenerated using Eq. 4. We use the oracle as the intervener and test three uses of the same operation. For correction, the oracle provides the true source and destination points at every step, for both Spatial CBM and Spatial CBM*, on the episodes of Table 3. For steering, we train the action expert with a single sentence shared across all ten tasks in place of the instruction, so that the concept is the only input that tells the robot which bowl to pick, and the oracle provides

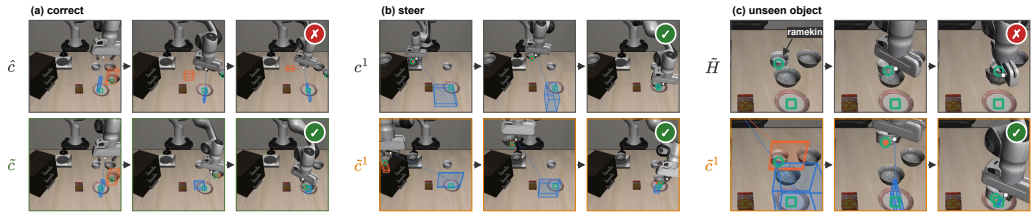

Figure 5: Intervention on the manipulation concept, with two runs from the same start in each panel. (a) Correction: the robot fails with the predicted concept $\hat{c}$ and succeeds with the corrected $\tilde{c}$. (b) Steering: the intervener provides the other, identical bowl as the source $\tilde{a}_t^1$, and the robot places that bowl instead. (c) Unseen object: Direct Action does not place the ramekin when its instruction $\tilde{H}$ asks for it, whereas providing the ramekin as the source $\tilde{a}_t^1$ makes the robot place it.

the position of the other, identical bowl as the source. For an unseen object, the oracle provides the position of the ramekin, which is never moved in any demonstration, and we compare against Direct Action whose instruction asks for the ramekin. Finally, we test whether MolmoAct (Lee et al., 2025), which predicts a visual trace before acting, acts as a concept bottleneck, by replacing its trace with that of another task.

Results. Correction improves Spatial CBM* from 77.2% to 84.8% and Spatial CBM from 75.0% to 79.8% (Table 3, Figure 5a). For steering, the single shared sentence does not hurt performance with true concepts (85.8%), and the robot places the other bowl in 93.5% of the episodes over two seeds (Figure 5b). For the unseen object, the robot places the ramekin in 67.0% of the episodes, against 7.0% for Direct Action (Figure 5c). Replacing MolmoAct's trace with that of another task barely changes its success (85.0% against 89.5%), so its trace does not act as a concept bottleneck.

5 Limitations

We see this work as a first step, and several directions remain open. We evaluate in simulation, since it gives us the true concept both for supervision and for the oracle, which is what makes the CBM variants and their interventions measurable. The concept itself only requires 3D points, which can equally come from depth and pose estimates on a real robot. We study *what* to intervene with and use an oracle to decide *when*, and learning when to intervene, e.g., from the concept predictor's own uncertainty, is a natural next step. Our concepts use one or two anchor points, which suffice for the navigation and pick and place tasks studied here, and richer concepts with more objects or orientations can be expressed with the same formulation. We also fix the resolution of each concept by hand, a 16×16 costmap and 13 bins per axis, and learning this resolution from data may help the concept predictor. Similarly, we instantiate each domain with a single backbone, VGGT for navigation and SmolVLA for manipulation, and the formulation applies unchanged to other perception backbones and VLAs. Finally, the corrections collected through interventions could be used to train the concept predictor further, closing the loop between deployment and learning.

6 Conclusion

In this paper, we brought concept bottleneck models to robot learning through the spatial concept bottleneck, where the concept is a geometric relation between task-defined points and the robot, and its ground truth therefore comes from the scene's geometry. We instantiated it as a costmap over the current view for navigation, and as the positions of the source and the destination relative to the gripper for manipulation, and evaluated it with the four CBM variants of Koh et al. (2020). In both domains, the bottleneck with true concepts performs on par with direct action prediction, so the spatial concept itself does not limit performance. In navigation, Spatial CBM is on par with or better than Direct Action on average, with the largest gains on the tasks furthest from the training data. Intervention corrects and steers the manipulation policy, and in navigation, a VLM that never sees the true concept improves closed-loop success when it is asked at oracle-triggered checks.

AI USE STATEMENT

In this work, we used generative AI tools mainly for the VLM-based intervention experiments as a measure of real-world deployability of the navigation systems. We have not used generative AI tools for any of the other experiments, or to implement methods, or to refine hypotheses, or to provide support in critical ingredients of our mathematical claims, and the rest of the required disclosure tasks like generating synthetic datasets are not applicable to this work. Additionally, we used generative AI tools for writing code at the function level, to propose restructuring of code, to monitor code running on GPU to surface any issues during code execution (this was manually reviewed by two authors), and for grammatical checks in a few places to polish writing. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Wherever we noticed high inter-seed variance, results for multiple seeds are provided with mean and standard deviation details to ensure robustness of the numbers. The main tables reporting Spatial CBMs evaluations for both navigation and manipulation has been reported with multiple seeds to ensure that the main line of claims this paper builds on is on concrete foundation. We will make all source code and experiments public upon acceptance of the paper.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yi-Fan Ding, Daniel Dworakowski, Jiaojiao Fan, Michele Fenzi, Francesco Ferroni, Sanja Fidler, Dieter Fox, Songwei Ge, Yunhao Ge, Jinwei Gu, Siddharth Gururani, Ethan He, Jiahui Huang, Jacob Samuel Huffman, Pooya Jannaty, Jingyi Jin, Seung Wook Kim, G. Klár, Grace Tin Yan Lam, Shiyi Lan, Laura Leal-Taixé, Anqi Li, Zhaoshuo Li, Chen-Hsuan Lin, Tsung-Yi Lin, Huan Ling, Ming-Yu Liu, Xian Liu, Alice Luo, Qianli Ma, Hanzi Mao, Kaichun Mo, Arsalan Mousavian, Seungjun Nah, Sriharsha Niverty, David Page, Despoina Paschalidou, Zeeshan Patel, Lindsey Pavao, Morteza Ramezanali, Fitsum A. Reda, Xiao-Shuai Ren, Vasanth Rao Naik Sabavat, Edward Schmerling, Stella Shi, Bartosz Stefaniak, Shitao Tang, Lyne P. Tchapmi, Przemek Tredak, Wei-Cheng Tseng, Jibin Rajan Varghese, Hao Wang, Hao-Xiang Wang, Heng Wang, Tingwei Wang, Fangyin Wei, Xinyue Wei, Jay Zhangjie Wu, Jiashu Xu, Wei Yang, Yen-Chen Lin, Xiaohui Zeng, Yuan Zeng, Jing Zhang, Qinsheng Zhang, Yuxuan Zhang, Qingqing Zhao, and Artur Zolkowski. Cosmos world foundation model platform for physical ai. *ArXiv*, abs/2501.03575, 2025. URL https://api.semanticscholar.org/CorpusID:275342594.

Anthropic. System card: Claude Opus 5, July 2026. URL https://www.anthropic.com/claude-opus-5-system-card. Accessed September 2026.

Tommaso Apicella, Alessio Xompero, and Andrea Cavallaro. Visual affordances: Enabling robots to understand object functionality. *CoRR*, abs/2505.05074, 2025. doi: 10.48550/ARXIV.2505.05074. URL https://doi.org/10.48550/arXiv.2505.05074.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://api.semanticscholar.org/CorpusID:274465127.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *CoRR*, abs/2410.24164, 2024. doi: 10.48550/ARXIV.2410.24164. URL https://doi.org/10.48550/arXiv.2410.24164.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller,

et al. Concept embedding models: Beyond the accuracy-explainability trade-off. *Advances in neural information processing systems*, 35:21400–21413, 2022.

Sourav Garg, Dustin Craggs, Vineeth Bhat, Lachlan Mares, Stefan Podgorski, Madhava Krishna, Feras Dayoub, and Ian Reid. Objectreact: Learning object-relative control for visual navigation. In *Conference on Robot Learning*. PMLR, 2025.

Vansh Garg, Rohit Jayanti, Krish Pandya, Sarthak Chittawar, Siddharth Tourani, Muhammad Haris Khan, Sourav Garg, and Madhava Krishna. Mast3r-nav: Waypixel navigation in relative 3d maps. In *International Conference on Robotics and Automation (ICRA)*, 2026.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. RT-trajectory: Robotic task generalization via hindsight trajectory sketches. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F1TKzG8LJO.

Marton Havasi, Sonali Parbhoo, and Finale Doshi-Velez. Addressing leakage in concept bottleneck models. *Advances in Neural Information Processing Systems*, 35:23386–23397, 2022.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Rich Walke, Anna Walling, Hao-Huan Wang, Li-Li Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *ArXiv*, abs/2504.16054, 2025. URL https://api.semanticscholar.org/CorpusID:277993634.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag 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. *arXiv preprint arXiv:2406.09246*, 2024.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. Pmlr, 2020.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. *CoRR*, abs/2508.07917, 2025. doi: 10.48550/ARXIV.2508.07917. URL https://doi.org/10.48550/arXiv.2508.07917.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, 2026a.

Yixuan Li, Yuhui Chen, Mingcai Zhou, and Haoran Li. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. In Viviana Mascardi, John Thangarajah, Chris Amato, and Louise A. Dennis (eds.), *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2026, Paphos, Cyprus, May 25-29, 2026*, pp. 2771–2780. ACM, 2026b. doi: 10.65109/LJRK3716. URL https://dl.acm.org/doi/10.65109/LJRK3716.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. Promises and pitfalls of black-box concept learning models. *CoRR*, abs/2106.13314, 2021. URL https://arxiv.org/abs/2106.13314.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. Spatialvla: Exploring spatial representations for visual-language-action model. *CoRR*, abs/2501.15830, 2025. doi: 10.48550/ARXIV.2501.15830. URL https://doi.org/10.48550/arXiv.2501.15830.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, volume 15 of *JMLR Proceedings*, pp. 627–635. JMLR.org, 2011.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations (ICLR)*, 2018.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. GNM: A General Navigation Model to Drive Any Robot. In *International Conference on Robotics and Automation (ICRA)*, 2023. URL https://arxiv.org/abs/2210.03370.

Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International conference on machine learning*, pp. 31504–31520. PMLR, 2023.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andrés Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Rémi Cadène. Smolvla: A vision-language-action model for affordable and efficient robotics. *CoRR*, abs/2506.01844, 2025. doi: 10.48550/ARXIV.2506.01844. URL https://doi.org/10.48550/arXiv.2506.01844.

Boyang Sun, Hanzhi Chen, Stefan Leutenegger, Cesar Cadena, Marc Pollefeys, and Hermann Blum. Frontiernet: Learning visual cues to explore. *IEEE Robotics and Automation Letters*, 10(7):6576–6583, 2025. doi: 10.1109/LRA.2025.3569122.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Siyin Wang, Junhao Shi, Zhao-Yang Fu, Xinzhe He, Fei Liu, Chenchen Yang, Yikang Zhou, Zhaoye Fei, Jingjing Gong, Jinlan Fu, Mike Zheng Shou, Xuanjing Huang, Xi-Peng Qiu, and Yu-Gang Jiang. World action models: The next frontier in embodied ai. *ArXiv*, abs/2605.12090, 2026. URL https://api.semanticscholar.org/CorpusID:288255685.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Hwiyeon Yoo, Yunho Choi, Jeongho Park, and Songhwai Oh. Commonsense-aware object value graph for object goal navigation. *IEEE Robotics and Automation Letters*, 2024.

Mert Yüksekgönül, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR*. OpenReview.net, 2023.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=S70MgnIA0v.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, XinQiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=PK07eretkF.