PRISM: PROBING PHYSICAL CONSEQUENCE SIGNALS IN WORLD MODEL PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned world models predict the consequence of each candidate action, yet standard planners reduce each prediction to a single feature distance to a goal image. This scalar readout couples task progress with obstacle safety in one number, and the progress signal dominates: consequence information that the model faithfully predicts is lost in the aggregation. We expose this readout bottleneck through a controlled comparison. A frozen text-image projection, aligned once on robot data without collision labels, reads named object relations from the same predictions. Where feature-distance planning fails on obstacle-dense pick-and-place sequences, this relational readout achieves 71.7%, revealing that the learned representations already encode physical consequences. Three isolating interventions confirm the information is per-candidate and prediction-dependent: shuffling action-score pairings eliminates every gain, prompting from current observations rather than prospective consequences yields only partial benefit, and goal-image contrast rules out directional matching. This is a representation-level finding: the consequence information exists regardless of how downstream planning uses it. PRISM exploits this by decoupling direction from selection: geometry handles progress toward the goal, and two sequential consequence gates read VLM-written relations from each candidate's predicted scene, filtering harmful candidates before geometry picks among survivors. On 450 transport trials, consequence gating raises completion from 62.2% to 86.7% and cuts obstacle displacement by 63%, lifting safe completion from 33.3% to 55.6%. The mechanism transfers to pick-and-place sequences and to a real Franka arm without retraining.

1 INTRODUCTION

Can a frozen projection that has never seen a collision label tell which robot actions will topple a nearby obstacle? We find that it can, by reading named object relations from an action-conditioned world model's next-frame predictions. Neither the world model nor the projection received safety-specific supervision, yet their composition cuts obstacle displacement by 63%. This result exposes a readout bottleneck in world-model planning: feature distance to a goal image integrates over all predicted dimensions, letting the large reaching signal drown out the small consequence signal from a local obstacle interaction. On pick-and-place sequences, goal-image readout yields zero success while relational readout on the same predictions achieves 71.7%. The consequence information is present in the learned representation; the standard readout discards it. Three controlled substitutions verify this: shuffling action-score pairings eliminates every gain, ruling out global bias; current-observation prompting—where the VLM describes the current scene rather than prospective outcomes—reaches 71.1%, above geometry alone but well below the 86.7% from prospective prompts, confirming that both static spatial reasoning and forward dynamics carry independent value; and goal-image contrast rules out directional matching. This is a representation-level finding about what action-conditioned world models learn, not an artifact of a particular planning setup. Because the projection needs only feature-space alignment, the readout extends wherever action-conditioned prediction meets text-image features.

Action-conditioned world models (Zhou et al., 2024; Assran et al., 2025) plan by matching predicted features to a goal image. This couples two objectives in a single score, reaching the target and keeping nearby objects undisturbed. When the two conflict, the reaching signal dominates: in high-dimensional feature space, the held object and the target account for most of the representation

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.

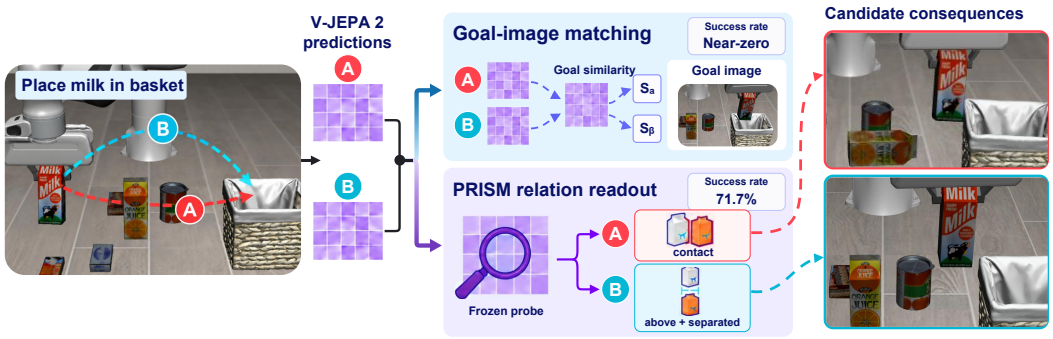


Figure 1: **The readout bottleneck.** The same V-JEPA 2 world model predicts the scene after each candidate action. *Upper:* feature-distance readout scores candidates by proximity to a goal image; the reaching signal dominates and obstacle consequences are lost. *Lower:* a frozen text-image projection reads named relations (contact versus clearance) from the same predictions, recovering consequence information and achieving 71.7%. Both readouts operate on identical predictions; the difference is purely in how information is read.

variance, while whether a peripheral obstacle remains upright or topples is a low-variance perturbation that feature distance effectively ignores. A single distant goal image specifies where the object should end up, not what it passes on the way. Classical collision-aware planners (Sundaralingam et al., 2023; Heins & Schoellig, 2023) query explicit obstacle geometry, but such models may not exist for a single camera, and even where depth sensing provides spatial proximity, it cannot predict dynamic consequences: two trajectories with identical clearance can produce opposite outcomes when one clips an edge that the other clears. Comparing candidates through text-based relations bypasses both limitations: it queries a specific semantic axis rather than an aggregate distance, and it reads from predicted scenes rather than static geometry.

Consider placing a carton past a juice box. Both a direct and a lifting motion reduce the distance to the basket and score alike, yet only the lift clears the obstacle. A goal image fails because both end positions look alike and obstacle stability is secondary in feature distance.

Direction and selection are separate problems. Geometry solves the first, using anchors on the held object and the container rim lifted from RGB-D to measure progress. The second requires knowing the consequences of each candidate, which geometry alone cannot provide. These consequences can be stated as relations between named objects, such as "the carton is above the juice box and separated from it", and they describe a single frame after the action—exactly what an action-conditioned world model predicts for each candidate.

PRISM (Probing Representations In Scene Models) makes this decomposition explicit. A VLM writes one relation to seek and one to avoid. For each candidate, the world model predicts the next scene and a frozen text-image projection reads how well it matches each relation. These scores drive two sequential consequence gates. The first removes candidates whose predictions resemble the undesired relation, the second keeps those matching the desired one, and geometry picks the survivor (Figure 2). Each gate compares candidates on its own scale, sidestepping cross-score calibration. A projection head, trained once for feature-space alignment and frozen during planning, enables the gates to read consequences already encoded by the world model, adding only lightweight text-image comparison on top of existing predictions. The underlying finding—that self-supervised prediction objectives incidentally encode consequence information accessible through relational queries—generalizes beyond this specific planning setup.

We evaluate on a factorial benchmark for the isolating interventions above: 450 transport trials (15 layouts, three planner seeds, 10 repetitions), sharing candidates and metrics. PRISM cuts displacement by 63% and raises completion from 62.2% to 86.7%. Safe completion, defined as task completion with no asset event, rises from 33.3% to 55.6%, characterizing the ceiling of single-step

consequence probing. The mechanism generalizes: 71.7% success across 180 pick-and-place trials, and transfers to a real Franka arm without retraining the world model or planner.

In summary, our contributions are:

- A readout bottleneck in world-model planning: the same predictions that fail under feature-distance readout yield 71.7% under relational readout, revealing that physical consequence information is encoded in the learned representation but discarded by the standard planning objective.
- Controlled identification: a frozen text-image projection, aligned once on robot data without collision labels, extracts consequence information from next-frame predictions. Three isolating interventions—score shuffling, current-observation prompting, and goal-image comparison—verify the information is per-candidate and prediction-dependent across 450 transport and 180 pick-and-place trials.
- Consequence gating: geometry handles direction, and two sequential consequence gates read VLM-written object relations from each candidate's predicted scene, removing harmful candidates without requiring geometric and semantic scores to share a common scale. On 450 transport trials, gating raises completion from 62.2% to 86.7% and cuts displacement by 63%; it transfers to pick-and-place and to a real Franka arm without retraining.

2 Related Work

Planning with predictive world models. Learned simulators and world models predict the outcome of actions from visual input (Yang et al., 2024). DINO-WM (Zhou et al., 2024) and V-JEPA 2-AC (Assran et al., 2025) predict in the feature space of pretrained visual encoders and plan by optimizing actions so that the predicted features approach those of a goal observation. Dreamer (Hafner et al., 2023) and TD-MPC2 (Hansen et al., 2024) learn latent dynamics for model-based control and evaluate actions with learned value or reward predictions rather than by matching a goal image. Goal-distance planning evaluates candidates through feature distance to a goal, coupling task completion and safety in a single scalar; value-based planning couples them in a single learned scalar instead. PRISM uses the same kind of action-conditioned prediction but reads consequence signals that this coupling discards, and it uses predictions to filter candidates rather than as the sole planning objective.

Language and VLMs for manipulation. Methods such as CLIPort (Shridhar et al., 2021), VoxPoser (Huang et al., 2023), ReKep (Huang et al., 2024), MOKA (Liu et al., 2024), and SayCan (Ahn et al., 2022) ground language in the current observation. Our goal-image contrast (Section 4.2) shows that the same predictions that fail under scalar distance readout yield 71.7% under relational readout.

Collision-aware planning and safe transport. Classical planners such as cuRobo (Sundaralingam et al., 2023) avoid collisions using explicit obstacle geometry, and model-predictive control can keep transported objects upright while avoiding obstacles (Heins & Schoellig, 2023). These methods query known meshes or dynamics models; however, clearance is not consequence. Two trajectories that pass the same distance from an obstacle can produce opposite outcomes if one clips an unstable edge that the other clears. A simpler alternative—adding a depth-based clearance penalty to the geometric score—does not distinguish trajectories that approach an obstacle at the same distance but with different dynamic outcomes (e.g., clipping versus clearing an edge). PRISM asks a different question: do learned predictions already encode consequence information? It reads signals from a learned predictor that simulates forward in feature space, requiring neither obstacle meshes nor dynamics models. PRISM is complementary to explicit collision avoidance: it could be combined with mesh-based checking where models are available, and it provides consequence information where they are not.

Representation probing and feature readout. Probing frozen representations to discover what they encode is standard in NLP (Belinkov, 2022) and increasingly common in vision (Caron et al., 2021). A hallmark of the probing paradigm is that the probe is kept simple and frozen so that a positive result reflects structure already present in the representation, not capacity added by the readout (Alain & Bengio, 2016). The standard evidence is controlled comparison: the same readout applied to different information sources isolates what each source encodes. PRISM applies

this methodology to action-conditioned world models: a frozen text-image projection trained once for feature-space alignment serves as the probe, and three controlled substitutions (shuffled scores, current-observation prompts, and goal-image features) isolate the contribution of per-candidate relational readout. The finding is a representation-level result: consequence signals are readable without collision-specific training, independent of the downstream planning mechanism. Keeping the probe simple is essential: a complex readout could discover signals through its own capacity rather than the representation, compromising the attribution.

3 METHOD

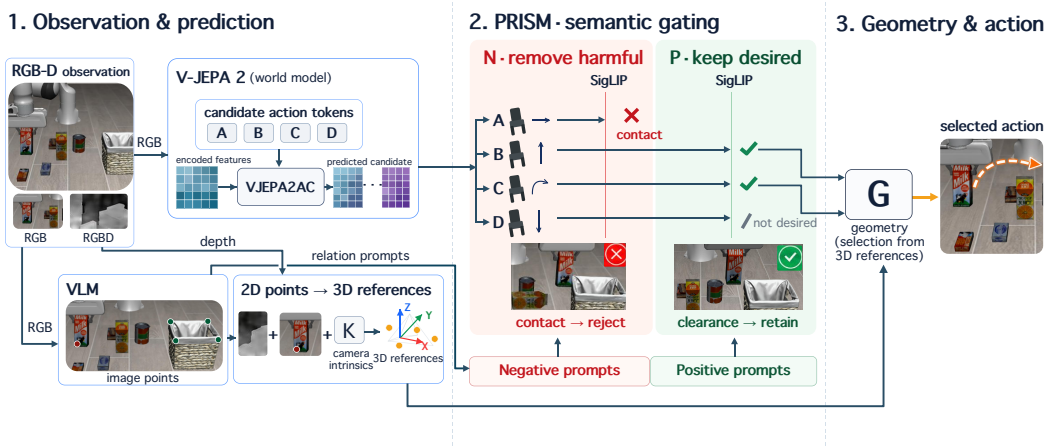


Figure 2: **The PRISM pipeline.** (1) An RGB observation is encoded by V-JEPA 2, which predicts the next scene for each candidate action. (2) A VLM writes relation prompts; a frozen SigLIP projection scores each prediction against negative (remove harmful) and positive (keep desired) relations, and two sequential gates N and P filter candidates. (3) RGB-D image points are lifted to 3D references via camera intrinsics; geometry G selects the best surviving candidate for execution.

The testbed is a held-object placement task: the robot must place a held object in a container without disturbing nearby obstacles (Figure 2).

3.1 PROBLEM SETTING

We study the transport and place stages of a tabletop placing task: at the start of an episode the gripper already holds the target object. At decision step t, the planner receives an RGB observation o_t, proprioceptive state q_t, a language task τ (e.g., "put the milk in the basket"), and the names of the target, the receiving container, and the obstacles declared for the scene. It samples a set of K candidate end-effector actions $\mathcal{A}_t = \{a^{(k)}\}_{k=1}^{K}$, selects one action $a_t^\star \in \mathcal{A}_t$, and executes it for a single control frame before observing again. The planner never receives an image of the desired final state. An episode ends when the held object comes within $\delta = 10\,\text{mm}$ of the placement reference, at which point the gripper opens, or when a budget of $T_{\max} = 72$ decisions is exhausted.

Many candidates shorten the distance similarly, yet some drag the held object across a neighbouring box. The outcomes to avoid are easy to state as named relations but contribute a small fraction of feature distance in a goal image, making this a clean testbed for probing whether predictions encode consequence information beyond geometry. Geometry determines where to go; the probe tests whether language-specified relations can identify which ways of getting there are admissible.

3.2 GEOMETRIC PROGRESS FROM RGB-D ANCHORS

The geometric score uses metric reference points extracted from the current observation (Figure 2). For the held object we take pixel anchors at material points identified by the VLM; for the container

we compute a surface-centre reference from visible rim points lifted via RGB-D. Each anchor is lifted to the robot frame as $\boldsymbol{x} = T(q_t)\,\pi^{-1}(\boldsymbol{u}, d)$, where π is the camera projection, $T(q_t)$ the camera-to-robot transform computed from the current joint state q_t via forward kinematics, and d the median of the valid depth values in a small window around $\boldsymbol{u}$. The median keeps the lift stable when depth is missing at object boundaries. Let $\boldsymbol{r}$ denote the lifted object anchor, $\boldsymbol{g}$ the lifted rim centre, and $\Delta\boldsymbol{x}(a)$ the displacement of the anchor under candidate a, which we take to be the commanded end-effector translation because the object moves rigidly with the gripper. The geometric score is

$$G(a) = -\left\|\boldsymbol{r} + \Delta\boldsymbol{x}(a) - \boldsymbol{g}\right\|_2 . \tag{1}$$

G rewards candidates whose endpoint lies closer to the container. It is blind to what the held object touches on the way there, and this is the failure that the relation scores are meant to catch.

3.3 Relation scores on predicted representations

The core question is whether per-candidate predictions encode consequence information beyond what the current observation provides. To test this, we compare each candidate's predicted scene to language-specified relations through a frozen projection head.

The current observation already provides static scene semantics; geometric scoring already captures kinematic displacement. Section 4.2 shows that neither suffices. The VLM does not observe predicted candidate scenes; it receives the current RGB observation, task description, and object names, and produces relation prompts describing expected or undesired outcomes. Candidate-specific representations are generated by the action-conditioned world model and scored by the frozen projection head. The projection head is trained on robot interaction data, minimizing the mean squared error between the projected V-JEPA 2 feature and the frozen SigLIP image embedding of the same frame. It never receives collision, safety, or task-success labels. Our claim is therefore label-free with respect to collision-specific supervision, not fully unsupervised. Relation knowledge enters through the SigLIP encoder's pretraining, not through the projection head's training data. Section 4.2 shows that prospective relations yield larger gains than current-observation relations, consistent with consequence information accessible through the predictions.

Predicting the consequence of each candidate. Let f_θ be the V-JEPA 2 video encoder, p_ϕ its action-conditioned predictor (Assran et al., 2025), and q_t the current robot proprioceptive state (joint angles, gripper width). For every candidate we predict the latent state that follows it:

$$\boldsymbol{z}_t = f_\theta(o_t), \qquad \hat{\boldsymbol{z}}_{t+1}^{(k)} = p_\phi\big(\boldsymbol{z}_t, q_t, a^{(k)}\big), \tag{2}$$

where $\hat{\boldsymbol{z}}_{t+1}^{(k)}$ is a grid of patch tokens. We average these tokens over space and map the result into the SigLIP image-embedding space (Zhai et al., 2023) with a projection head h_ψ that is trained beforehand and kept frozen during planning:

$$\boldsymbol{v}^{(k)} = h_\psi\big(\,\mathrm{MeanPool}(\hat{\boldsymbol{z}}_{t+1}^{(k)})\big). \tag{3}$$

All K candidates share the encoding $\boldsymbol{z}_t$ and are predicted in one batch. Spatial pooling aggregates the 256 patch tokens into a single scene-level vector. We treat $\boldsymbol{v}^{(k)}$ as a scene-level summary of the candidate's short-term consequence.

Relation statements from a VLM. Every $R = 3$ decisions, a VLM (Gemini 3.8 Flash (**?**)) receives the current RGB frame and the declared object names and returns four statements: a desired relation s^+, a contrast s_c^+, an undesired relation s^-, and a safe counterpart s_c^-. The prompt asks for static spatial relations verifiable in a single predicted frame (details in Appendix A). For example, the desired relation reads "the milk carton is above the orange-juice box and separated from it"; the undesired relation reads "the milk carton or the gripper touches the orange-juice box, and the box tilts or shifts".

Relation margins. We encode each statement with the SigLIP text encoder $e(\cdot)$ and score each candidate by two contrastive margins:

$$P(a) = \cos\big(\boldsymbol{v}^{(a)}, e(s^+)\big) - \cos\big(\boldsymbol{v}^{(a)}, e(s_c^+)\big), \tag{4}$$

$$N(a) = \cos\big(\boldsymbol{v}^{(a)}, e(s^-)\big) - \cos\big(\boldsymbol{v}^{(a)}, e(s_c^-)\big). \tag{5}$$

A high P means that the predicted scene is closer to the desired relation than to its contrast; a low N means that it is closer to the safe counterpart than to the undesired relation. Subtracting the contrast is intended to cancel what both statements share, such as the object names, the table, and the basket, and to keep what is specific to the relation.

3.4 SEQUENTIAL GATING

The three scores are not on a common scale: G is in metres, P and N are cosine differences. A weighted sum $G + \lambda(P - N)$ was sensitive to λ because G is in metres (typical range 0.01–0.05) while $P - N$ spans cosine differences (range up to ± 2), and the absolute scale of P and N shifts every time the VLM refreshes its statements, so PRISM uses P and N only for admissibility and leaves the final choice to G (Algorithm 1).

Motion eligibility. When the object anchor is farther than 20 mm from the reference, we discard candidates that move it by less than 5 mm, including the zero action; within 20 mm, every candidate is eligible. This keeps the gates from stalling the robot far from the goal. We call the eligible set $\mathcal{A}_t^0$.

Risk gate. We keep the candidates whose predicted scene looks less like the undesired relation than the typical candidate does: $\mathcal{A}_t^N = \{a \in \mathcal{A}_t^0 : N(a) \leq \text{median}_{\mathcal{A}_t^0} N\}$, which retains about half of the eligible set.

Relation gate. Among the survivors, we keep the candidates whose predicted scene matches the desired relation nearly as well as the best one: $\mathcal{A}_t^P = \{a \in \mathcal{A}_t^N : P(a) \geq \max_{\mathcal{A}_t^N} P - \rho \, \text{range}_{\mathcal{A}_t^N} P\}$ with $\rho = 0.60$.

Geometric selection. The executed action is $a_t^\star = \arg\max_{a \in \mathcal{A}_t^P} G(a)$.

The full procedure is listed as Algorithm 1 in Appendix A. Both thresholds are relative: the median and the ρ-fraction are invariant to affine rescaling of the margins, which shift whenever the VLM statements are refreshed. Neither gate can empty the set, so PRISM always returns an action. The gating mechanism is deliberately minimal so that the controlled comparisons in Section 4 test what the predictions encode, not what the readout can learn (design rationale in Appendix D).

4 EXPERIMENTS

All comparisons follow the standard design of representation probing: the planner, candidates, geometric selection, and prediction backbone are held fixed across conditions, and only the information available to the gates varies. As in NLP and vision probing, the goal is to isolate what a representation encodes, not to benchmark against alternative architectures; the three controlled substitutions (Section 4.2) serve this purpose by holding the readout constant and varying only the information source.

Scope of comparison. The experiments compare readout methods on the same prediction backbone rather than comparing PRISM against other planning systems such as cuRobo (Sundaralingam et al., 2023) or MPC-based transport (Heins & Schoellig, 2023). This is standard in representation probing: the goal is to isolate what a representation encodes, not to claim that one planning system outperforms another (Belinkov, 2022; Alain & Bengio, 2016). A system-level comparison would answer a different question (which planner is better for safe transport?) rather than the one we address (do predictions already encode consequence information that standard readouts discard?). The three controlled substitutions provide strong internal validity: shuffled scores on the same candidates eliminate every gain, current-observation prompts on the same predictions yield only partial benefit, and goal-image readout on the same predictions yields zero success.

The setting is held-object placement in LIBERO (Liu et al., 2023) with obstacles along the transport path, chosen because placement tasks separate direction (geometry suffices) from selection (consequences matter). The experiments address four questions. **(Q1)** Do action-conditioned predictions encode physical consequence signals invisible to standard planning, and can a frozen text-image probe extract them? **(Q2)** Does reading these signals through sequential gates reduce disturbance to surrounding objects without giving up task completion? **(Q3)** Does each gate contribute, and does

the benefit depend on the correct action-score pairing? **(Q4)** Does the mechanism transfer beyond controlled transport?

4.1 SETUP

Scenes. We construct 15 base layouts in which the gripper holds a target object and a basket serves as the receiving container. The layouts cover six conditions: a direct path without obstacles, a short path with the basket moved closer, an obstacle near the start, the middle, or the end of the path, and two consecutive obstacles ordered low-to-high or high-to-low. Each layout is run with three planner seeds (candidate-sampling randomness) and 10 repetitions, giving 450 transport trials in a factorial design. Each scene is frozen by a content hash and serves as its own control; only the gates vary across policies (admission criteria in Appendix B). The six conditions in Table 1 yield five pairwise comparisons against G500 and 2,250 within-scene paired evaluations. This matched factorial design provides internal validity for causal attribution: performance differences isolate the contribution of consequence signals rather than confounds introduced by mismatched baselines.

Protocol. Each policy samples 500 candidates per decision, executes one control frame, and repeats for at most 72 decisions; release triggers at 10 mm. All policies share scenes, seeds, and evaluation rules (Appendix A).

Metrics. We treat the basket and every declared obstacle as protected assets. *Completion* requires the LIBERO task predicate to hold at the end of the observation window. An *asset event* occurs when any protected asset translates by more than 10 mm or rotates by more than 15° during the episode. *Safe completion* requires completion without an asset event. *Maximum asset displacement* is the largest translation of any protected asset during the episode; we report its mean over scenes. Completion alone would credit a policy that reaches the basket by pushing it aside; the other three metrics make that cost visible.

Conditions. **G500**: motion eligibility then $\arg\max G$, no VLM calls. **P-only / N-only**: single gate plus G. **Shuffled-PN**: both gates with scores permuted across candidates (same permutation for P and N), destroying the action-consequence correspondence while preserving the joint margin structure. **Current-observation prompting**: the VLM describes relations visible in the current observation rather than generating prospective relations for candidate outcomes, while candidate-conditioned predicted features and the downstream gating procedure remain unchanged.

4.2 THE READOUT BOTTLENECK AND ITS CONSEQUENCES FOR PLANNING (Q1, Q2)

Table 1: **Readout source comparison on 450 transport trials.** All conditions share the same 500 candidates, the same world model predictions, the same geometric selection, and the same evaluation; they differ only in what information the gates read. Arrows mark the preferred direction and the best value is in bold. Displacement is the maximum translation of any protected asset in a scene, averaged over scenes.

Gates	Policy	Compl. ↑	Safe compl. ↑	Asset evt. ↓	Disp. (mm) ↓
none	G500	62.2%	33.3%	66.7%	52.9
single	N-only	66.7%	46.7%	53.3%	49.9
	P-only	80.0%	46.7%	53.3%	34.3
control	Current-obs prompting	71.1%	42.4%	–	–
both	Shuffled-PN	40.0%	13.3%	86.7%	83.3
	PRISM (ours)	**86.7%**	**55.6%**	**40.0%**	**19.7**

– Safe completion requires completion as a precondition.

Three controls isolate the source of the consequence signals, testing alternatives to the prediction-level explanation.

Score shuffling (Table 1). Randomly permuting scores across candidates at every decision drops completion from 86.7% to 40.0% and raises mean displacement from 19.7 to 83.3 mm, both worse than G500. The comparison is against the full two-gate pipeline in PRISM, so that the only change is which candidate each score belongs to, and P and N are permuted by the same permutation so their joint relation is preserved. This is inconsistent with the hypothesis that the benefit comes from the number of candidates retained or from a global bias in the VLM statements.

Current-observation prompting (Table 1). The VLM describes the current scene rather than prospective candidate outcomes; candidate predictions, projection head, and gating remain unchanged. This achieves 71.1% completion (320/450) and 42.4% safe completion—above geometry alone (62.2%/33.3%) but well below prospective prompting (86.7%/55.6%), showing that prompt content matters independently of the readout mechanism.

Goal-image contrast. We replace the consequence gates with goal-image planning: the planner selects the candidate whose predicted features are closest (patch-token ℓ_1 distance) to a goal image. On the 180 pick-and-place trials, goal-image readout succeeds in 0 of 180 trials, while text-relation readout on the same predictions succeeds in 129 of 180 (71.7%). Feature distance projects all scene information into one scalar where reaching dominates; text-relation scoring queries a specific consequence axis.

Together, the three controls reveal a gradient: current-scene prompts on candidate predictions improve over pure geometry (71.1% vs. 62.2%), prospective-consequence prompts add substantially more (86.7%), and scalar distance recovers none of it (0/180 pick-and-place trials). The benefit depends on the correct per-candidate pairing, the prompt content, and the relational readout mechanism.

This gradient disambiguates three sources of improvement. Current-observation prompting isolates the VLM's spatial vocabulary: even describing static relations in the current scene improves candidate selection over pure geometry. The additional 15.6-point gain from prospective prompting isolates the world model's forward prediction: describing what *will happen* rather than what *is happening* accesses information that no current-frame readout can provide. The collapse to zero under scalar distance isolates the readout mechanism: the consequence signal exists in the prediction but occupies a low-variance subspace invisible to feature distance, requiring targeted semantic queries to recover.

Table 1 quantifies the planning benefit. PRISM completes 390 of 450 trials (86.7%) versus 280 of 450 (62.2%) for G500, and achieves safe completion in 250 of 450 (55.6%) versus 150 of 450 (33.3%), while lowering displacement from 52.9 to 19.7 mm (63% reduction). Episode-level confidence intervals are reported in the appendix. Both completion and safety improve together because candidates that disturb obstacles often also spoil placement. The 55.6% safe completion characterizes the ceiling of single-step probing.

4.3 Both gates matter, and only with the right pairing (Q3)

Each gate helps on its own (Table 1); combining both gives the best value on all four metrics, improving over P-only by 6.7 points in completion and 43% in displacement. Breaking the action-score pairing reverses every gain: shuffled scores complete only 40.0% and raise displacement to 83.3 mm, worse than G500. Detailed single-gate analysis is in Appendix E.

4.4 Pick-and-place sequences and real-robot transfer (Q4)

To test whether the consequence signals generalize beyond held-object transport, we evaluate on complete pick-and-place sequences starting from an ungrasped, settled state. Six frozen scenes cover the same six obstacle conditions. Pick uses pure geometric selection (G) with VLM-guided anchor consensus and a 10 mm closure trigger (details in Appendix A). Place uses the full N-to-P-to-G consequence gating with the same 500 candidates, horizon 1, and 10 mm release trigger as in the transport experiments. Across 180 trials (6 scenes, 3 planner seeds, 10 repetitions), PRISM achieves 71.7% end-to-end success. G500 shares the same pick stage and prediction backbone but uses only geometric selection during place; it achieves 63.9% (115/180). Pick attachment succeeds in over 90% of trials for both methods, so the gap arises specifically in the place stage where consequence

gating operates. The goal-image contrast reported in Section 4.2 uses these same 180 trials and the same V-JEPA 2 backbone, providing the most direct evidence for the readout bottleneck.

Table 2: **Pick-and-place and real-robot results.** Success requires end-to-end task completion (pick, transport, and place). All simulation methods share the same pick stage, prediction backbone, and scenes.

Setting	Method	Success
Simulation: full pick-and-place	Goal-image planning	0/180
	G500	63.9%
	PRISM	**71.7%**
Real Franka	PRISM	demonstrated



Figure 3: **Real-robot deployment.** PRISM transfers to a physical Franka Emika arm without retraining the world model or planner; the projection head is separately trained on DROID data. The consequence signals readable in simulation are also readable through a different camera, kinematic chain, and independently trained projection head.

Real-robot deployment. To test whether the readout mechanism is viable under embodiment change, we deploy an identically architected but separately trained projection head, aligned on DROID data (Khazatsky et al., 2024) and frozen, on a physical Franka arm (Figure 3). The world model and planner are not retrained. The system succeeds in several trials; failures trace to depth-sensor noise, VLM naming of unfamiliar objects, and the single-step prediction horizon.

A diagnostic probe (Appendix F) isolates a single decision near the basket: the G-maximizing candidate advances 45.5 mm but shifts the basket by 7.5 mm, while PRISM advances 14.3 mm with no basket displacement.

5 Limitations and Conclusion

Probing boundaries. Four boundaries of single-step consequence probing each account for roughly a quarter of remaining non-completions: (1) single-step horizon misses multi-step chain reactions; (2) scene-level pooling loses centimetre-scale contact; (3) VLM naming errors misdirect the gates; (4) the risk gate can remove all goal-reaching candidates. The evaluation uses one camera, one container type, and preliminary real-robot trials. Each boundary suggests a clear extension; the present work establishes that single-step probing already recovers substantial consequence information from predictions trained without collision-specific supervision.

Conclusion. Action-conditioned world models encode physical consequences in their predictions, but standard goal-image planning discards those signals. Three controlled substitutions confirm that the benefit depends on the correct per-candidate pairing, the prompt content, and the relational readout mechanism. PRISM exploits this readout bottleneck, raising completion from 62.2% to 86.7%, cutting displacement by 63%, and lifting safe completion from 33.3% to 55.6% on 450 transport trials; the mechanism transfers to pick-and-place and to a real Franka arm without retraining. The factorial design and controlled substitutions provide a reusable protocol for probing other action-conditioned models: freeze the readout, vary the information source, and test whether the effect survives. We hope that reading consequence signals from world model predictions will extend to other tasks where the goal is easier to describe by what should not happen.

AI STATEMENT

We used generative AI to assist author-led work, not to conduct autonomous research, with a commitment to academic integrity and research ethics. Its uses included identifying relevant literature, drafting and polishing portions of the manuscript, checking clarity and consistency, and organizing existing experimental results into tables. AI models also serve as components of the reported method: a vision-language model generates the four relation statements consumed by the consequence gates at each refresh cycle, as described in Section 3.3 and Appendix A. All AI-assisted material underwent detailed human review before inclusion. The authors checked literature references and factual claims against original sources, verified tabulated values against the underlying experimental records, and reviewed generated text for accuracy, attribution, and fidelity to the intended meaning, correcting or excluding unsupported, misleading, or fabricated content. AI output was not treated as evidence without verification. The authors retained responsibility for all scientific judgments, experimental design, data collection, analysis, and conclusions, and take full responsibility for the accuracy, originality, and integrity of the final manuscript, figures, and reported results.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 specify the planner, and Section 4.1 specifies the scenes, protocol, and metrics; each scene is frozen by a content hash of its initial state.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *ICLR Workshop*, 2016.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. URL https://arxiv.org/abs/2506.09985.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 2022.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *ICLR*, 2024.

Adam Heins and Angela P. Schoellig. Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator. *arXiv preprint arXiv:2305.17484*, 2023. URL https://arxiv.org/abs/2305.17484.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. URL https://arxiv.org/abs/2307.05973.

Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. URL https://arxiv.org/abs/2409.01652.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

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

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. MOKA: Open-world robotic manipulation through mark-based visual prompting. *arXiv preprint arXiv:2403.03174*, 2024. URL https://arxiv.org/abs/2403.03174.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. *arXiv preprint arXiv:2109.12098*, 2021. URL https://arxiv.org/abs/2109.12098.

Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, et al. cuRobo: Parallelized collision-free robot motion generation. *arXiv preprint arXiv:2310.17274*, 2023. URL https://arxiv.org/abs/2310.17274.

Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023. URL https://arxiv.org/abs/2303.15343.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024. URL https://arxiv.org/abs/2411.04983.

A Implementation Details

Algorithm 1 One decision of PRISM

Require: observation o_t; statements (s^+, s_c^+, s^-, s_c^-), refreshed every R decisions

1: Sample K candidates $\mathcal{A}_t$; compute anchors $\boldsymbol{r}, \boldsymbol{g}$ from o_t and $G(a)$ by Eq. equation 1

2: $\boldsymbol{z}_t \leftarrow f_\theta(o_t)$; for all $a \in \mathcal{A}_t$: $\boldsymbol{v}^{(a)} \leftarrow h_\psi(\mathrm{MeanPool}(p_\phi(\boldsymbol{z}_t, a)))$; compute $P(a), N(a)$ by Eqs. equation 4–equation 5

3: $\mathcal{A}_t^0 \leftarrow \{a : \|\Delta \boldsymbol{x}(a)\| \geq 5\,\mathrm{mm}\}$ if $\|\boldsymbol{r} - \boldsymbol{g}\| > 20\,\mathrm{mm}$, else $\mathcal{A}_t$ ▷ motion eligibility

4: $\mathcal{A}_t^N \leftarrow \{a \in \mathcal{A}_t^0 : N(a) \leq \mathrm{median}_{\mathcal{A}_t^0} N\}$ ▷ risk gate

5: $\mathcal{A}_t^P \leftarrow \{a \in \mathcal{A}_t^N : P(a) \geq \max_{\mathcal{A}_t^N} P - \rho\ \mathrm{range}_{\mathcal{A}_t^N} P\}$ ▷ relation gate

6: **return** $a_t^\star = \arg\max_{a \in \mathcal{A}_t^P} G(a)$; execute for one control frame

Projection head. The projection head h_ψ maps V-JEPA 2 spatial features (256 tokens of 1408 dimensions each) into the 768-dimensional SigLIP image-embedding space through a two-layer MLP with GELU activation and ℓ_2 normalization. It is trained on robot interaction data: LIBERO demonstrations for simulation experiments, DROID data for real-robot deployment. Training minimizes the normalized-vector mean squared error between the mean-pooled V-JEPA 2 feature and the frozen SigLIP image embedding of the same frame. The head learns to project V-JEPA 2 features into the SigLIP image-embedding space; relation knowledge enters through SigLIP's pre-training, not through the projection head's training data. The head sees no collision or cost signal at any point. The head is frozen after training and shared across all tasks and scenes. It is not trained on collision labels, obstacle identities, or safety annotations. The head architecture comprises Linear(1408, 2048), GELU activation, LayerNorm(2048), Linear(2048, 768), and ℓ_2 normalization. The training objective minimizes the normalized-vector mean squared error between the projected V-JEPA 2 feature and the frozen SigLIP image embedding of the same frame: $\mathcal{L} = \| \mathrm{norm}(h_\psi(\mathrm{mean}(z^{\mathrm{VJEPA}}))) - y^{\mathrm{SigLIP}} \|_2^2$. Optimization uses AdamW with learning rate 10^{-3}, weight decay 10^{-4}, batch size 256, and 50 epochs. For the simulation experiments, training data is drawn from LIBERO-Goal demonstrations. For real-robot deployment, a separate head with the identical architecture is trained on DROID interaction data. The two heads are independent; neither is transferred to the other domain.

VLM prompt contract. The VLM (Gemini 3.8 Flash (**?**)) receives the current RGB frame, the task identity, and grounding hints (names of the target, container, and visible obstacles). It returns four statements in JSON: a desired relation (positive_prompt), its contrast (positive_contrast_prompt), an undesired relation (negative_prompt), and the corresponding safe relation (negative_contrast_prompt). The prompt excludes motion verbs (lift, move, advance, collide, push, knock) and requires static spatial relations verifiable in a single predicted frame. Protected assets include the basket and all non-target objects; the target object is not protected. Each statement is limited to 30 words. The response follows the JSON schema: {``P'': ``<desired static relation>'', ``P_contrast'': ``<contrast>'', ``N'': ``<adverse relation>'', ``N_contrast'': ``<safe counterpart>''}. Preferred spatial vocabulary includes: *above, below, left of, right of, in front of, behind, inside, outside, near, far, centered over, offset from, touching, separated from, upright, tilted, stationary, displaced*. The prompt prohibits motion and control verbs: *lift, raise, ascend, move, advance, approach, translate, clear, collide, push, knock*. If the VLM returns malformed JSON, one repair request is issued; continued failure terminates the VLM call for that refresh cycle.

Candidate sampling. Each decision samples 500 candidates from a zero-mean Gaussian with per-axis standard deviation 0.085, clipped to 0.05 per axis and to an ℓ_1 norm of 0.085. Candidate 0 is the zero action. Candidate 1 is a direct proposal toward the geometric target, clipped to the same constraints. Rotation increments are zero; the gripper delta maintains the current grip state. Predictions are batched in chunks of 8 to limit GPU memory.

Anchor initialization. For the held object, the VLM proposes the lowest visible material point; RGB-D lifts it to a 3D anchor with a local depth patch. The container rim centre is computed from four visible boundary points proposed by the VLM and lifted via RGB-D, with a consistency check rejecting proposals that drift more than 30 mm from the accepted reference. Both anchors are frozen after initialization.

Evaluation thresholds. Safe completion requires zero asset events: no protected asset translates more than 10 mm or rotates more than 15°. The release threshold is 10 mm from the held-object anchor to the rim centre. Post-release observation lasts 108 frames.

Seeds and reproducibility. Each scene is identified by a scene_seed (layout and obstacles), a planner_seed (candidate sampling), and a repeat_id. All policies within a comparison share the scene hash, start state, anchors, and planner seed. Results are not selected by outcome.

B Scene Conditions and Layout

The 15 base layouts are drawn from six obstacle conditions. Direct path (3 layouts) places no obstacle between the held object and the basket. Short path (2 layouts) moves the basket closer to reduce the initial distance. Start obstacle (3 layouts) places an obstacle in the first third of the path. Mid obstacle (3 layouts) places an obstacle near the midpoint. End obstacle (2 layouts) places an obstacle close to the basket. Consecutive obstacles (2 layouts) place two obstacles in sequence, one low-to-high and one high-to-low. Target objects include milk, butter, cream cheese, chocolate pudding, and tomato sauce, and the receiving container is a basket in all scenes. A scene is admitted only if, after 120 settling frames, the target, the basket, and every declared obstacle are visible in the planner's single camera view, both anchors have valid depth, the held object touches no other object, and a three-dimensional swept-volume check confirms that the obstacle lies on the transport path. Each base layout is run with three planner seeds (candidate-sampling randomness) and 10 repetitions, giving 30 trials per layout and 450 trials in total. For layouts with obstacles, the swept-volume precheck rejects any layout where the straight-line path from the held-object bottom to the basket rim centre does not intersect the obstacle bounding box, ensuring that the obstacle poses a genuine collision risk. Direct-path layouts are exempt from this check because they contain no declared obstacle.

C Failure Classification

We classify failures in the 450 transport trials into four categories. (1) VLM naming error, where the VLM refers to the wrong object and the gates filter on an incorrect relation. (2) Over-strict risk gate, where the risk gate removes all candidates that reach the basket within the budget, causing the robot to stall. (3) Geometric convergence, where the surviving candidates make insufficient progress and the episode runs out of decisions. (4) Asset contact, where a surviving candidate still disturbs a protected asset despite passing both gates. Each accounts for roughly a quarter of remaining non-completions.

D Gating Design Rationale

The risk gate comes first because it removes candidates, whereas the relation gate ranks against the best; filtering for risk before ranking prevents a candidate with a strong desired relation from surviving on that strength alone. Both thresholds are relative to the current candidate set: the median is a nonparametric order statistic whose breakdown point is 50%, and the ρ-fraction is invariant to affine rescaling of P. Relative thresholds are necessities: VLM statements refresh every three decisions, shifting the absolute margin scale. Neither gate can empty the set: the median keeps at least one candidate and the relation gate keeps the maximizer of P.

E Detailed Ablation (Q3)

The single-gate rows of Table 1 isolate the two gates. Both single gates raise safe completion from 33.3% to 46.7% and lower asset events from 66.7% to 53.3%. P-only also completes more scenes (80.0% vs. 62.2%) and lowers mean displacement to 34.3 mm. N-only adds modest completion (66.7% vs. 62.2%) with displacement close to G500 (49.9 vs. 52.9 mm). Against P-only, PRISM raises completion from 80.0% to 86.7%, safe completion from 46.7% to 55.6%, and lowers displacement by 43%. Shuffled scores (same gates, same thresholds, permuted margins) complete only 40.0% and raise displacement to 83.3 mm, confirming that the scores carry per-candidate information.

F Diagnostic Probe and Failure Classification

We isolate a single decision near the basket. The candidate maximizing G advances 45.5 mm but shifts the basket by 7.5 mm; PRISM advances 14.3 mm and leaves the basket in place. Four boundary conditions emerge, each accounting for roughly a quarter of remaining non-completions: (1) Horizon: surviving candidates sometimes make insufficient progress. (2) Resolution: scene-level pooling misses centimetre-scale contact. (3) Naming: VLM errors misdirect the gates. (4) Coverage: the risk gate removes all candidates that reach the container. VLM statements refresh every three decisions; across up to 24 refreshes per episode ($T_{\max}/R = 72/3$), the undesired-relation structure was stable.

G Confidence Intervals

Treating trials as independent (normal approximation, $n = 450$ per condition): completion $[83.5, 89.8]$ for PRISM vs. $[57.7, 66.7]$ for G500; safe completion $[51.0, 60.1]$ vs. $[29.0, 37.7]$. The intervals do not overlap on either metric.