SEMANTIC ROLE–STRUCTURED WORLD MODEL FOR PHYSICAL INTERACTION PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

World models for robotic manipulation must capture how actions transform a scene in ways that are meaningful for task execution. Existing object-centric approaches preserve the transient identity of entities, while leaving the evolving task structure that gives those dynamics meaning largely implicit. We introduce SEMROLE, a world model that represents scene evolution through semantic roles: a task-conditioned structure explicitly encoding the functional status of scene entities, their task-relevant relations, and the current phase of execution. Conditioned on language, visual observations, and proprioception, SEMROLE uses this structured state to govern action generation and action-conditioned dynamics, predicting interaction events, relation satisfaction, and phase progression. At inference time, these predictions guide phase-aware action selection and local repair of invalid rollouts. We instantiate SEMROLE on both lightweight LEWM and pretrained COSMOS backbones, improving closed-loop success from 45.5% to 83.0% on LIBERO, 42.9% to 55.2% on RoboCasa, and 55.0% to 81.0% on real-robot Galbot transfer. Across backbones and domains, explicitly modeling task-semantic structure yields substantially stronger world-model-based manipulation.

1 INTRODUCTION

World models enable agents to anticipate the consequences of actions and evaluate possible futures before acting (Hafner et al., 2025; Zhou et al., 2024; Maes et al., 2026). Beyond changes in appearance, geometry, or motion, an action changes how entities participate in an instructed interaction: which task function each entity is bound to, which relations must hold among them, and the execution phase in which those relations are required. These are not independent labels. The same physical effect—opening the gripper—violates a required relation during transport and completes a valid release only after support is established at the goal; a planner cannot decide whether to keep a prefix or repair a suffix from any one of these signals alone. We call this instruction-dependent organization of participation, relations, and phase semantic-role structure, meaning how entities take part in the commanded interaction rather than linguistic predicate roles. A world model for physical interaction planning should therefore predict physical changes in this joint state, making explicit whether they establish, preserve, or violate the conditions needed for task progress.

Object-centric world models provide a useful starting point: they represent scene entities and their interactions, supporting structured prediction and control (Kipf et al., 2020; Ferraro et al., 2025; Spieler et al., 2026). However, an entity-centered state does not by itself specify which interactions an instruction requires at each stage. Figure 1 makes this distinction concrete. Opening a gripper ends a grasp in both cases, but during transport it violates a required relation, whereas after support is established at the goal it completes a valid release. Predicting this physical effect must therefore be coupled with determining whether it establishes or preserves the conditions needed for task progress.

Task objectives can turn such predictions into decisions, as demonstrated by reward-based imagination (Hafner et al., 2025) and visual-goal planning (Zhou et al., 2024). However, a return or goal-distance score alone does not identify *which interaction condition fails* or *where an invalid continuation begins*. Without this information, rejecting a candidate does not specify how to retain its useful prefix and correct the remainder. Selection also cannot recover a suitable action absent

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.

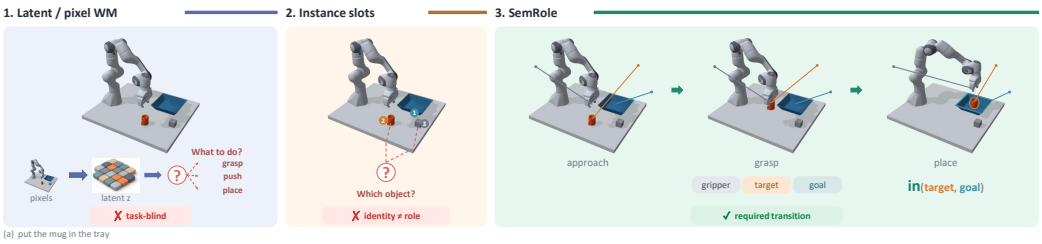


Figure 1: **Why semantic roles matter for world-model planning.** For the instruction "put the mug in the tray" **(a)**, three state organizations imply different decisions. **Left:** a latent/pixel world model compresses the observation into z but remains *task-blind*, so the next action (grasp/push/place) is under-specified. **Middle:** instance slots recover object identities, yet *identity* $\neq$ *role*: which entity is Target vs. Goal is still unresolved. **Right:** SEMROLE binds entities to typed roles (gripper / target / goal), tracks the required phase transitions (approach→grasp→place), and evaluates relations such as $\mathrm{in}(\mathrm{target}, \mathrm{goal})$. The schematic compares state organization, not measured model outputs.

from the proposal set. This motivates a task representation that shapes proposed actions and exposes intermediate interaction conditions for their evaluation and repair.

We introduce SEMROLE, a *semantic role–structured world model for physical interaction planning* (Figure e 1(c)). SEMROLE makes semantic-role structure the organizing principle of a shared predictive *role state*. Concretely, we bind visual entities to their task functions, couple them with relational and phase context, and learn action-conditioned transitions in this joint space. The role state predicts whether an action establishes, preserves, or violates these conditions—grasp and contact, fixture state, and phase transitions—rather than scoring only a return or a goal distance. The same state also conditions action generation, so task structure shapes candidates before their predicted consequences are evaluated. The same state also conditions action generation, so task structure shapes candidates before their predicted consequences are evaluated.

To ground task participation in observations, an instruction-conditioned binder combines visual entities with language and proprioception, while an interaction-history encoder supplies phase context. To improve the proposals available to the selector, we condition a flow-matching action generator on the same role state. To test whether candidates respect changing requirements, a semantic rollout module predicts intermediate relations and phase transitions for phase-aware selection. When these predictions locate the first violation, a prefix-conditioned repair module preserves the preceding action sequence and regenerates only its suffix. New observations update the role state after each executed prefix, grounding the next decision in observed execution.

We instantiate SEMROLE with a compact LEWM backbone on LIBERO and pretrained COSMOS on RoboCasa and Galbot. On LIBERO, role conditioning improves single-candidate success from 45.5% to 62.8%, with selection and repair increasing it to 83.0%. Within the COSMOS setting, success rises from 42.9% to 55.2% across 17 RoboCasa tasks and from 55.0% to 81.0% on Galbot.

Our contributions are threefold: **(i)** we introduce SEMROLE, a world model whose shared *role state* binds visual entities to instruction-conditioned task functions, relations, and phase, and uses that same state both to generate actions and to predict whether they establish, preserve, or violate the conditions required for progress; **(ii)** we instantiate this state with instruction-conditioned binding, phase-aware semantic rollout that localizes the first violated condition, and prefix-conditioned suffix repair that keeps a valid prefix and regenerates only the remainder; **(iii)** we evaluate SEMROLE with a compact LEWM backbone and pretrained COSMOS on LIBERO, RoboCasa, and Galbot, and isolate the gains from role-conditioned generation, selection, and repair.

2 RELATED WORK

World Models for Planning. Dreamer learns policies through imagined trajectories (Hafner et al., 2025). DINO-WM predicts pretrained visual features for goal-conditioned planning (Zhou et al., 2024), while LEWM learns an end-to-end joint-embedding world model from pixels (Maes et al., 2026). COSMOS provides pretrained world models at foundation scale (NVIDIA, 2025). RC-

aux aligns latent prediction with planning through multi-horizon prediction and budget-conditioned reachability (Li et al., 2026). Semantic World Models formulate prediction as questions about task-relevant future semantics (Berg et al., 2025). For interaction planning, evaluating a future outcome also requires knowing which conditions must hold along the way and when those requirements change. SEMROLE makes these phase-dependent requirements explicit in predicted transitions, exposing which relations an action should establish or preserve and where a continuation first becomes invalid.

Object-Centric and Structured World Representations. Object-centric representations make scene entities and their interactions explicit. C-SWM learns object-factored relational dynamics (Kipf et al., 2020), and Slot Attention learns exchangeable object representations (Locatello et al., 2020). FOCUS, VIOLA, and Slot-MPC use object structure for exploration, visuomotor control, and planning (Ferraro et al., 2025; Zhu et al., 2022; Spieler et al., 2026). OA-WAM makes objects addressable to an action decoder (Liu et al., 2026); WorldDP combines object-centric subgoal planning with diffusion policies (Goswami et al., 2026); MaskWAM unifies mask prompting and prediction (Yu et al., 2026). Object-centric Gaussian world models predict action-conditioned scene dynamics (Kreber et al., 2026), while Grounding DINO and SAM-family models can supply proposals for visual grounding (Liu et al., 2024; Kirillov et al., 2023; Ravi et al., 2025). Entity structure alone, however, does not specify each participant's instruction-dependent function or which relations must hold at the current stage. Task and motion planning addresses task structure by connecting discrete task specifications to continuous geometric constraints (Garrett et al., 2021). SEMROLE brings entity participation and task conditions into a shared predictive representation: instruction-dependent participant bindings, task-relevant relations, and execution phase jointly form the role state. This state is estimated from observations and rolled forward under candidate actions, allowing the controller to distinguish a relation that must be preserved from one that may validly end as the task progresses.

Generative Actions and Closed-Loop Decisions. Diffusion Policy models multimodal action sequences with receding-horizon execution (Chi et al., 2025). Diffuser generates trajectories by denoising and expresses planning conditions through inpainting (Janner et al., 2022). Flow matching learns a continuous transformation from noise to data (Lipman et al., 2023). SEMROLE builds on conditional generation: the shared role state shapes candidate actions, while predicted semantic transitions guide selection and identify the first invalid segment for prefix-conditioned suffix repair.

3 SEMANTIC-ROLE-STRUCTURED WORLD MODELS

A task-directed planner must determine whether an interaction remains valid on the way to a goal and where to intervene when it does not. This motivates a shared role state that connects task participation to predicted physical changes: it shapes action proposals, exposes intermediate requirements, and supplies a boundary for local repair. Figure 2 follows this interface from state construction and binding (a,b), through semantic rollout (c), to selection and repair (d).

Setting and notation. At control step $t \geq 0$, the robot receives a finite RGB history o_t from V cameras, corresponding proprioception $q_{\leq t}$, and instruction ℓ. Each replan samples K action chunks $A_t = (a_t, \ldots, a_{t+H-1}) \in \mathbb{R}^{H \times d_a}$, with normalized actions $a_t \in \mathbb{R}^{d_a}$. The robot executes $1 \leq h_{\text{exec}} \leq H$ steps before observing again. Integer boundaries $0 = h_0 < h_1 < \cdots < h_J = H$ divide a chunk into J prediction segments; segment j contains relative action indices $h_{j-1}, \ldots, h_j - 1$. Parameters $\phi, \theta, \psi, \omega$ denote the state encoder, dynamics, action generator, and selector, respectively.

3.1 ROLE-STATE CONSTRUCTION AND PARTICIPANT BINDING

From observations to reusable entities. The same scene can support different instructions, so its visual description should remain reusable as task assignments change. We first aggregate backbone patch features into N soft entities $e_{t,n}$ using learned queries and spatial, view, and temporal cues. To avoid depending on mask availability, optional proposals bias attention while full patch context remains accessible; proposal dropout trains both input settings (Appendix A.2).

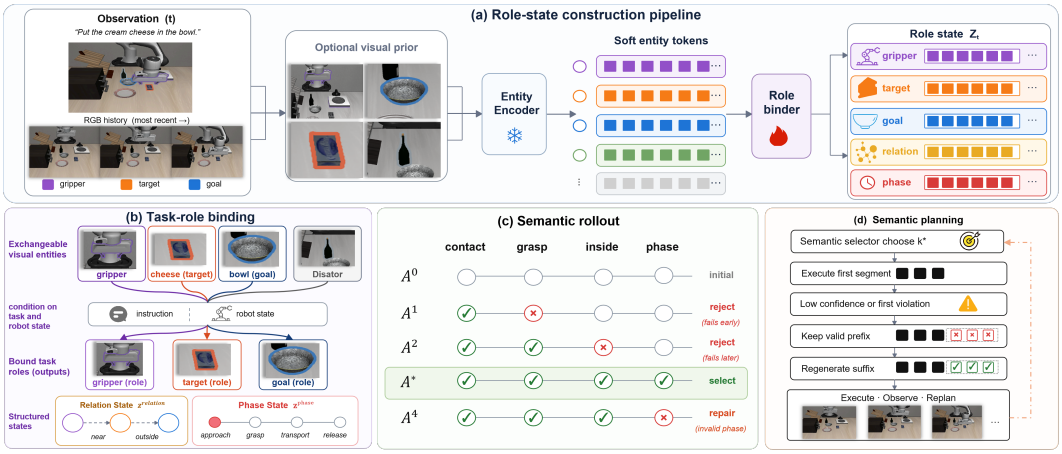


Figure 2: **SemRole connects role-state construction to semantic decisions.** **(a)** Encode observations into soft entities and construct a joint role state. **(b)** Bind task participants and encode relation/phase context. **(c)** Check role-conditioned candidates; columns summarize semantic requirements over their rollouts. **(d)** Select, optionally repair and re-score, then execute and replan from new observations. Training icons are schematic; encoder optimization depends on the backbone.

Binding entities to task participants. An exchangeable entity index does not specify which entity to manipulate or which goal state to pursue. The binder in Figure 2(b) therefore uses language to assign task participation and proprioception to condition that assignment on the robot's current configuration.

Let $\mathcal{R}_{\text{ent}} = \{\text{grip}, \text{tgt}, \text{goal}\}$ index the gripper, manipulated target, and goal components. For each r, scaled dot-product attention (Vaswani et al., 2017) uses a query $Q_r(\ell, q_t) \in \mathbb{R}^{d_k}$ and entity keys $K_e(e_{t,n}) \in \mathbb{R}^{d_k}$, with key width d_k. To make binding sensitive to spatial configuration as well as feature affinity, a geometry descriptor $g_{t,n} \in \mathbb{R}^{d_g}$ and learned vector $w_r \in \mathbb{R}^{d_g}$ add a component-specific bias. The binding weights and d-dimensional component state are:

$$
\begin{aligned}
b_{t,r,n} &= \mathrm{softmax}_n\left(\frac{Q_r(\ell, q_t)^\top K_e(e_{t,n})}{\sqrt{d_k}} + w_r^\top g_{t,n}\right), \\
z_t^r &= U_r\left(\sum_{n=1}^{N} b_{t,r,n} V_e(e_{t,n}), \ell, q_t\right), \qquad r \in \mathcal{R}_{\text{ent}}.
\end{aligned}
\tag{1}
$$

Here V_e projects values to $\mathbb{R}^d$, and U_r fuses attended evidence with language and proprioception. Soft binding accommodates tasks in which target and goal refer to the same articulated fixture. The goal component can then combine its appearance with a desired state such as *closed*, while the gripper component can use proprioception directly. These assignments remain independent of persistent segmentation IDs.

Coupling participants to interaction context. Identifying participants leaves open how they interact and which conditions the next action must preserve. We therefore couple their states to relational and phase context (Figure 2b). A relation encoder R_ϕ captures current configurations and conditions. Because their task significance depends on preceding interaction, a history encoder P_ϕ summarizes the last L observed steps, truncated at episode start. Both use language; their width-d outputs complete the joint role state:

$$
\begin{aligned}
z_t^{\text{rel}} &= R_\phi(z_t^{\text{grip}}, z_t^{\text{tgt}}, z_t^{\text{goal}}, \ell), \\
z_t^{\text{phase}} &= P_\phi\left(\{z_s^{\text{grip}}, z_s^{\text{tgt}}, z_s^{\text{goal}}, z_s^{\text{rel}}\}_{s=\max(0,t-L+1)}^{t}, \ell\right), \\
Z_t &= (z_t^{\text{grip}}, z_t^{\text{tgt}}, z_t^{\text{goal}}, z_t^{\text{rel}}, z_t^{\text{phase}}) \in \mathbb{R}^{5\times d}.
\end{aligned}
\tag{2}
$$

The three participant components and two context components jointly represent semantic-role structure. Sharing this state lets generation and prediction use the same task interpretation. Binding,

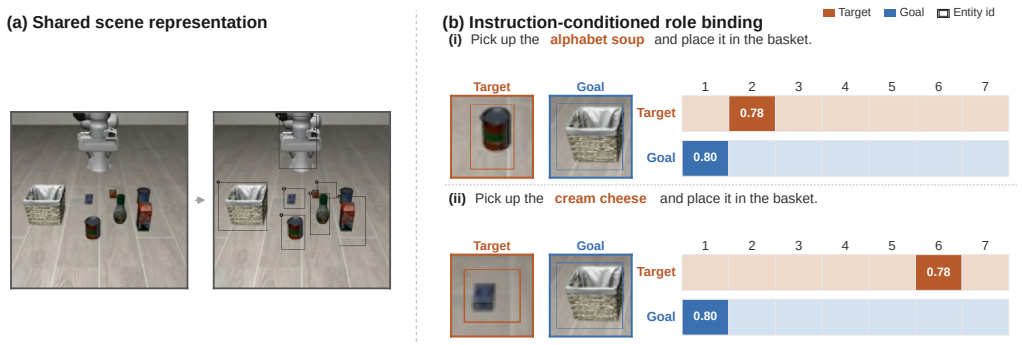


Figure 3: **Task participation depends on the instruction.** The same scene supports different target assignments while the basket remains the goal. Target and goal bindings ground the participant components of the joint role state. The attention weights illustrate soft binding; quantitative grounding results appear in Table 8.

predicate, and phase labels anchor its semantics during training; deployment uses observations, language, and proprioception. Figure 3 in Appendix A.2 illustrates instruction-dependent rebinding.

3.2 Role-Conditioned Generation and Semantic Rollout

Role-conditioned candidates. A selector cannot recover a suitable continuation absent from its candidate set. We therefore condition generation on Z_t, so participant bindings and interaction context influence which actions are proposed. A flow-matching generator (Lipman et al., 2023) supplies the candidates in Figure 2(c); its prefix-conditioned form is reused for repair. Given a demonstrated chunk A^+, standard Gaussian noise $\epsilon \in \mathbb{R}^{H\times d_a}$, and flow time $\tau \sim \mathcal{U}[0,1]$, its vector field v_ψ is trained with:

$$A^\tau = (1-\tau)\epsilon + \tau A^+,$$
$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{A^+,Z_t,\ell,\epsilon,\tau}\frac{\|v_\psi(A^\tau,\tau \mid Z_t,\ell) - (A^+ - \epsilon)\|_F^2}{Hd_a}. \tag{3}$$

The squared Frobenius norm sums squared errors over time and action coordinates; flow time τ is distinct from robot time t. Integrating $dA^\tau/d\tau = v_\psi(A^\tau,\tau \mid Z_t,\ell)$ from $A^0 = \epsilon$ to $\tau = 1$ yields A_t^k. Independent noise samples produce K candidates with shared weights.

Intermediate role-state transitions. A plausible endpoint can conceal an earlier loss of grasp and provides no boundary for retaining a useful prefix. To expose both the violation and where repair should begin, we predict intermediate states at the J segment boundaries.

For candidate k, let $A_{t,j}^k$ denote segment j and initialize $\widehat{Z}_{t,0}^k = Z_t$. A transformer D_θ exchanges component information and attends to ordered action tokens from E_a, allowing participant changes to affect relational and phase predictions. Since a segment can affect some components while leaving others stable, it predicts a residual $\Delta z_{t,j}^{r,k} \in \mathbb{R}^d$ and a gate logit $\gamma_{t,j}^{r,k} \in \mathbb{R}^d$ for each $r \in \mathcal{R} = \mathcal{R}_{\mathrm{ent}} \cup \{\mathrm{rel}, \mathrm{phase}\}$:

$$(\Delta z_{t,j}^{r,k}, \gamma_{t,j}^{r,k}) = D_\theta^r(\widehat{Z}_{t,j-1}^k, E_a(A_{t,j}^k), \ell),$$
$$\widehat{z}_{t,j}^{r,k} = \widehat{z}_{t,j-1}^{r,k} + \sigma(\gamma_{t,j}^{r,k}) \odot \Delta z_{t,j}^{r,k}. \tag{4}$$

Here σ is the elementwise sigmoid and $\odot$ denotes elementwise multiplication. The gated update predicts $\widehat{Z}_{t,j}^k$ at time $t + h_j$.

Phase-dependent semantic checks. Valid progress can require ending a previously necessary relation. In Figure 1, transport requires a stable grasp; once support at the goal is established, release ends the grasp while preserving support. We therefore separate predicate establishment from preservation, supervising preservation only while the phase requires that condition. Heads read

consecutive predicted states and their intervening actions to predict contact and grasp status, establishment, preservation, conjunction satisfaction, fixture state, the next phase, and transition validity. Figure 2(c) summarizes satisfaction of applicable requirements; segment-level predictions retain the timing needed for repair. Appendix A.3 defines the label masks and losses.

Simulation supervision derives predicates and fixture states from task specifications and simulator state, and contact/grasp labels from contact, gripper, and relative-motion information. These are training targets, with unavailable labels masked out. Prediction at J boundaries localizes violations to segments rather than exact continuous-time events.

3.3 Selecting Actions by Semantic Validity

Semantic selection. Predicting task conditions exposes possible violations; choosing among candidates also requires judging their significance for the current phase. The selector s_ω in Figure 2(d) therefore integrates the ordered semantic predictions, using uncertainty as a confidence cue. Let $\widehat{Y}^k_{t,j}$ collect the semantic outputs at segment j, including binary-condition probabilities and the phase distribution, and u^k_t summarize predictive uncertainty. The current phase conditions their interpretation in the score:

$$S^k_t = s_\omega\Big(z^{\text{phase}}_t, (\widehat{Y}^k_{t,1}, \dots, \widehat{Y}^k_{t,J}), u^k_t\Big), \qquad k^\star = \arg\max_{1\le k\le K} S^k_t. \tag{5}$$

This contextual ranking weighs establishment, preservation, and transition validity together. The selected chunk is checked before execution. Scores are learned ranking quantities rather than calibrated success probabilities; uncertainty remains a diagnostic input.

Action-sensitive ranking. Selection needs scores that respond to different actions even when the scene and instruction stay fixed. We train this comparison directly using temporal shuffles, reversed segments, and task-matched substitutions, retaining only transformations labeled as less valid. For a valid chunk A^+ and labeled negative A^- from the same context, let S^+ and S^- be their scores. With margin $m > 0$, the objective is:

$$\mathcal{L}_{\text{rank}} = \mathbb{E}_{(A^+,A^-)}\left[\max(0, m - S^+ + S^-)\right]. \tag{6}$$

The shared context makes scene or instruction recognition alone insufficient to satisfy the ranking margin.

3.4 Conditional Suffix Repair and Replanning

Retaining useful decisions. A poor continuation can follow a useful approach and grasp; resampling the whole chunk discards those earlier decisions. When predictions locate the first invalid segment $j^\star$, SemRole therefore retains the preceding $b = h_{j^\star-1}$ steps and regenerates the remainder. The repair path in Figure 2(d) is conditional: accepted candidates proceed directly to execution. If the best score falls below a validation-set threshold η without a localized violation, use $b = 0$: low confidence alone identifies no prefix to retain.

Conditioning on a prefix does not itself keep its coordinates fixed during generation. We therefore clamp them in both initialization and flow integration. Let $M_b \in \{0,1\}^{H\times d_a}$ equal one in the first b action rows and zero elsewhere, with $\bar{M}_b = 1 - M_b$. Using the selected chunk $A^{k^\star}_t$ as the anchor gives:

$$\begin{aligned} \widetilde{A}^0 &= M_b \odot A^{k^\star}_t + \bar{M}_b \odot \epsilon, \\ \frac{d\widetilde{A}^\tau}{d\tau} &= \bar{M}_b \odot v_\psi\Big(\widetilde{A}^\tau, \tau \mid Z_t, \ell, M_b, M_b \odot A^{k^\star}_t\Big). \end{aligned} \tag{7}$$

The same v_ψ thus resamples the suffix while the masked derivative preserves every prefix coordinate. This follows trajectory inpainting (Janner et al., 2022), with a boundary determined by semantic violations. To learn continuation under this constraint, training freezes random prefixes and supervises suffix velocity (Appendix A.4); $b = 0$ reduces to full resampling.

Re-evaluation and closed-loop execution. A regenerated suffix can introduce a new violation, so every repaired candidate is rolled out and scored again. To limit accumulated prediction error, the

Table 1: **Generation, selection, and repair on LIBERO.** Success (%); Avg. equally weights Object/Spatial/Goal/Long. All rows use the LEWM line. Candidate 0 executes a fixed sample without learned selection.

Decision rule	Object	Spatial	Goal	Long	Avg.
LeWM latent: Candidate 0	58	50	46	28	45.5
SEMROLE: Candidate 0	74	67	63	47	62.8
+ semantic selector	85	79	75	62	75.3
+ full resampling	88	82	78	66	78.5
+ suffix repair	**91**	**86**	**83**	**72**	**83.0**

robot executes h_{exec} steps of an accepted chunk before using fresh observations to reconstruct Z_t and generate new candidates, closing the loop in Figure 2(d). Exhausting the repair budget without acceptance invokes the rejection fallback.

Learning and backbone interface. The shared state must retain predictive information while exposing task conditions usable by generation and selection. We therefore combine backbone representation learning with semantic supervision, action generation, action ranking, and masked repair. For nonnegative weights $\lambda_{\text{rep}}, \lambda_{\text{sem}}, \lambda_{\text{FM}}, \lambda_{\text{rank}}, \lambda_{\text{repair}}$, the objective is:

$$\mathcal{L} = \lambda_{\text{rep}}\mathcal{L}_{\text{rep}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} + \lambda_{\text{FM}}\mathcal{L}_{\text{FM}} + \lambda_{\text{rank}}\mathcal{L}_{\text{rank}} + \lambda_{\text{repair}}\mathcal{L}_{\text{repair}}. \tag{8}$$

Appendix A.5 specifies loss interfaces and outstanding implementation settings. We use LEWM on LIBERO and pretrained COSMOS features on RoboCasa and Galbot. Both instantiations share the role-state structure and decision procedure; comparisons remain within each backbone.

4 EXPERIMENTS

We organize the evaluation around three questions: **(Q1)** Does role structure improve the candidates before selection? **(Q2)** Can semantic prediction turn this coverage into successful decisions, and does local repair help? **(Q3)** Does the design remain useful across backbones, task families, and physical execution?

Settings and comparisons. The LEWM line evaluates LIBERO-Object, LIBERO-Spatial, LIBERO-Goal, and LIBERO-Long (also called LIBERO-10), which test different forms of manipulation generalization (Liu et al., 2023). The COSMOS line evaluates 17 atomic RoboCasa kitchen tasks (Nasiriany et al., 2024) and transfer to a Galbot robot. Baselines use the same backbone as the corresponding SEMROLE model. The baseline action prior is a flow-matching head conditioned on backbone features; it is not a claim about the unmodified backbone's published benchmark performance. The main candidate budget is $K = 4$.

The primary metric is closed-loop success. Candidate 0 denotes the first generated chunk ($k = 1$ in our equations); uniform Random@4 selects one of four chunks without learned scoring. A validation-chosen fixed index controls for candidate-index preference. Oracle@4 measures outcome-informed selection from four candidates and serves as a non-deployable diagnostic of candidate quality. Its scope is limited to the evaluated pool and continuation protocol. Appendix C records the registered protocol, including trial counts, seeds, repair budgets, and the precise oracle convention.

4.1 ROLE STRUCTURE IMPROVES BOTH PROPOSALS AND DECISIONS

Q1: Better candidates before scoring. In Table 1, changing from the LeWM latent to role conditioning raises Candidate 0 success from 45.5% to 62.8%, a 17.3-point improvement. Oracle@4 rises from 68.0% to 89.5%. These results show that the improvement begins in the action prior, before a learned selector is applied. Random selection obtains 61.8%, close to Candidate 0, while choosing a fixed candidate ID on validation reaches 64.8%. Thus the semantic selector's gain cannot be explained by consistently preferring one sample index in these evaluations.

Table 2: **Cumulative components on RoboCasa with Cosmos.** Macro-average success (%) over 17 tasks. +Role changes generation; +Score adds semantic selection; +Repair adds suffix repair. The hard subset is a descriptive grouping of 11 tasks with Base $\leq 45\%$, not an independent benchmark split. Δ is +Repair minus Base in percentage points.

Tasks	Base	+Role	+Score	+Repair	Oracle@4	Δ
All 17	42.9	45.2	51.8	**55.2**	61.4	+12.3
Hard 11	29.3	32.4	41.4	**46.1**	54.8	+16.8

Q2: Selection and repair provide distinct gains. Semantic selection raises success to 75.3%, 12.5 points above Candidate 0. Suffix repair reaches 83.0%, compared with 78.5% for full resampling. The largest remaining difficulty is the long-horizon suite, where the full model reaches 72%, versus 83–91% on the other suites. The original-pool oracle exceeds semantic selection by 14.3 points, computed before rounding suite averages, indicating scope for better decisions even before additional generation. Since repair changes the candidate pool, its performance should not be interpreted as closing a formally fixed oracle bound. The matched repair-budget specification used for this comparison is registered in Appendix C.

4.2 The Same Task Structure Helps a Foundation Backbone

Q3: Kitchen tasks with Cosmos. Table 2 evaluates cumulative additions to the COSMOS action prior. Role conditioning improves the mean from 42.9% to 45.2%; semantic scoring raises it to 51.8%; suffix repair reaches 55.2%. The 12.3-point overall gain is smaller than on LIBERO, and most of it appears after semantic scoring is added. This difference cautions against attributing one uniform mechanism of improvement to every backbone and task distribution.

The hard subset improves from 29.3% to 46.1%. At task level, the largest gains are on turning off the stove (+23 points), opening a cabinet (+22), and turning on the microwave (+22). Improvements are not universal: OpenStandMixerHead decreases by one point and PickPlaceSinkToCounter by two. The per-task breakdown in Appendix B.1 preserves these regressions. Aggregate success supports the usefulness of the decision pipeline; it does not by itself establish that insufficient candidate coverage caused each baseline failure.

Physical execution on Galbot. Simulation and physical results are reported as separate evaluation settings. With the COSMOS backbone on a Galbot robot, aggregate success increases from 55.0% for the baseline to 81.0% for SEMROLE (+26.0 points). We evaluate two physical tasks—tabletop placement ($68.0\% \rightarrow 88.0\%$) and microwave retrieval ($42.0\% \rightarrow 74.0\%$)—with **50 independent, reset-matched trials per task per method**. The reported aggregate is the unweighted mean over these two tasks. Both methods share the same backbone family, candidate budget ($K{=}4$), and trial protocol; only the role-structured decision stack differs. Models are pretrained in simulation and fine-tuned with 20 teleoperated demonstrations per task (40 total) for 2k update steps from a single adapted checkpoint (Appendix C); this comparison is therefore not claimed as zero-shot sim-to-real transfer. Where a Galbot $\pm$ is reported, it is a bootstrap standard deviation over the 50 trials, not a three-seed training std. Qualitative rollouts are shown in Figure 4; per-task rates appear in Appendix B.5.

4.3 Which Structure and Predictions Matter?

Role structure. Table 3 compares increasingly structured states on LIBERO. Adding language to object slots improves success from 55.0% to 62.0%. Explicit target and goal bindings raise it to 68.0%; adding relations reaches 73.0%; the complete role state reaches 83.0%. This pattern supports functional structure beyond instruction conditioning alone. Since the final step adds gripper and phase together, the table does not establish their individual contributions or exclude effects of supervision and model capacity.

Prediction error and decision quality differ. The full model achieves 78.2% pairwise action-ranking accuracy, versus 53.7% without shuffled negatives (Appendix B.3). Removing phase super-

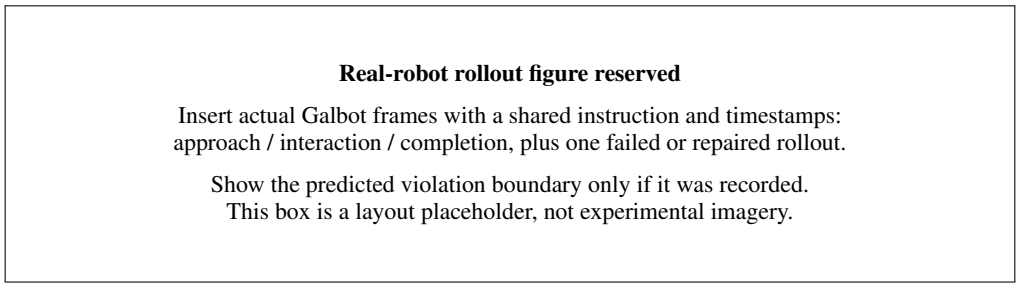


Figure 4: **Reserved qualitative evidence from physical execution.** Replace with recorded Galbot frames and a caption tied to actual task outcomes. No synthetic robot result is substituted for the missing experiment figure.

Table 3: **Task structure in the state representation.** LIBERO success (%) for progressively structured states. The $\pm$ convention remains to be specified (Appendix C). The last row jointly introduces gripper and phase, so it does not isolate their individual effects.

Representation	Success (%)
Five instance slots	55.0 ± 3.6
Object slots + instruction	62.0 ± 3.3
Target + goal roles	68.0 ± 3.1
Target + goal + relation	73.0 ± 2.9
SemRole (+ gripper, phase)	$\mathbf{83.0 \pm 2.4}$

vision slightly lowers latent prediction error while reducing ranking accuracy to 66.8%. This is a direct example of the paper's motivating distinction: a representation can predict its training targets more accurately yet be less useful for choosing actions.

Transfer and perceptual dependence. Object-suite pretraining improves adaptation on Goal and Long relative to training from scratch, but does not improve Spatial in the reported comparison (72.1% versus 72.8%). The no-mask model reaches 72.6%, compared with 75.1% with a soft segmentation prior in the separate perception diagnostic. These results support reusable task information and mask-free operation within the tested settings; they do not establish invariance to all task shifts or segmentation failures. Appendix B gives all transfer, mask, action-sensitivity, and efficiency results.

5 Conclusion

We presented SemRole, a semantic role–structured world model for physical interaction planning. It jointly represents participants' functional status, task-relevant relations, and execution phase in a shared role state. This state conditions action generation and predicts semantic transitions for phase-aware selection and suffix repair. Experiments on LIBERO, RoboCasa, and Galbot show improved closed-loop success across two backbone settings. The current role structure is tailored to a dominant target and goal; more complex interactions may require extensions. Semantic supervision costs, prediction errors, and repair overhead remain limitations, and real-time control performance requires further evaluation.

Reproducibility Statement

The role-state construction, transition model, action objective, selector, and masked repair procedure are defined in Section 3 and Appendix A. Appendix B reports the per-task and diagnostic results. Appendix C records the registered experimental configuration for the reported runs; fields still unset are marked explicitly rather than invented.

AI USE STATEMENT

Generative AI assisted with restructuring and drafting the manuscript, clarifying the mathematical specification, checking numerical summaries, reviewing related literature, interpreting reported results, and preparing the LaTeX layout and validation scripts. The reported experimental measurements were supplied by the authors; no new experimental measurements were generated during this assistance. **[To complete: Authors: verify the completed specification against the implementation, review the resulting text and claims, and record any other AI use before submission.]**

REFERENCES

Jacob Berg, Chuning Zhu, Yanda Bao, Ishan Durugkar, and Abhishek Gupta. Semantic world models. *arXiv preprint arXiv:2510.19818*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. FOCUS: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. doi: 10.1146/annurev-control-091420-084139.

Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. Unifying object-centric world models and diffusion policy: A hierarchical framework for multi-stage robotic tasks. *arXiv preprint arXiv:2606.08775*, 2026.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 9902–9915. PMLR, 2022.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.

Jens U Kreber, Lukas Mack, and Joerg Stueckler. Learning action-conditional and object-centric Gaussian splatting world models for rigid objects. *arXiv preprint arXiv:2606.01950*, 2026.

Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Predictive but not plannable: RC-aux for latent world models. *arXiv preprint arXiv:2605.07278*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

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.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. OA-WAM: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *International Conference on Learning Representations*, 2025.

Jonathan Spieler, Angel Villar-Corrales, and Sven Behnke. Slot-MPC: Goal-conditioned model predictive control with object-centric representations. *arXiv preprint arXiv:2605.14937*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. MaskWAM: Unifying mask prompting and prediction for world-action models. *arXiv preprint arXiv:2606.13515*, 2026.

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.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. VIOLA: Imitation learning for vision-based manipulation with object proposal priors. *arXiv preprint arXiv:2210.11339*, 2022.

A Mathematical Specification and Training Interfaces

This appendix makes the proposed method explicit where the source description left interfaces implicit. In particular, it specifies phase-dependent label masks and prefix-conditioned generation. The corresponding implementation choices and hyperparameters must be checked against the experimental system; fields not recoverable from the supplied experiment summaries are marked in Appendix C.

A.1 Notation and Typed State

Table 4: Symbols used in the method. Robot time, flow time, and rollout-segment indices are distinct.

Symbol	Meaning
t, τ, j	Robot control step, continuous flow time, and rollout-segment index.
H, d_a, K	Action horizon, action dimension, and initial candidate count.
J, h_j	Number of segments and end boundary of segment j.
L, V, N	Role-history length, camera count, and number of soft entities.
d, d_k, d_g	State-component width, attention-key width, and geometry-feature width.
$e_{t,n}$, $b_{t,r,n}$	Entity n and its binding weight for participant component r.
Z_t, $\widehat{Z}^k_{t,j}$	Observed role state and predicted state at $t + h_j$ for candidate k.
A^k_t, $A^k_{t,j}$	Candidate action chunk and its jth action segment.
$\widehat{Y}^k_{t,j}$, S^k_t	Predicted semantic transitions and candidate score.
$j^\star$, b, M_b	First violating segment, retained action count, and prefix mask.
h_{exec}, R	Executed prefix length and maximum repair-proposal count.

The joint role state Z_t is implemented by five components with fixed meanings. The participant components ground the end effector and grasp-relevant robot state (gripper), the object or articulated part to manipulate (target), and the destination or desired fixture state (goal). Their entity bindings can change with the instruction. The relation component combines relative geometry with task predicates; phase represents progress through applicable stages. Together they specify task participation in its interaction context. Tasks may have different stages, and a goal need not be a separate physical object.

A.2 Soft Entities and Optional Segmentation Priors

Flattening the patch tokens across views and history gives tokens x_i, with $i = 1, \ldots, P$ and P the total patch count. For entity query q^{ent}_n and patch key k^{patch}_i in a shared d_k-dimensional space, let $M_{n,i}$ be a proposal affinity aligned to query n. With nonnegative bias strength λ_{seg}, the entity attention is:

$$\alpha_{n,i} = \text{softmax}_i \left(\frac{(q^{\text{ent}}_n)^\top k^{\text{patch}}_i}{\sqrt{d_k}} + \lambda_{\text{seg}} M_{n,i} \right). \tag{9}$$

The entity vector aggregates projected patch values with these weights. Proposal masks influence attention, not the availability of image patches. Setting $\lambda_{\text{seg}} = 0$ recovers patch-only inference. Proposal-set dropout, individual-mask dropout, and bias-strength variation expose training to missing or unreliable priors. The proposal-to-query association and corruption distributions are implementation details that must be supplied; a raw segmentation ID cannot be assumed to equal a learned entity index. For the same reason, mask identity is not the definition of target or goal.

Entity permutation preserves participant binding. Consider a permutation π of the entities and apply it consistently to $e_{t,n}$ and $g_{t,n}$. Because the key, value, and geometry functions are shared across entity indices, the attention logits in Eq. 1 are permuted by π. Softmax permutes in the same way. The weighted sum is unchanged after reindexing, so z^r_t is invariant to this permutation. The argument requires that no additional entity-index embedding enters the binder. It is a property of the stated binding equation, not a claim that segmentation or temporal tracking is error-free.

A.3 Semantic Labels, Applicability, and Missing Supervision

Let s_{t+h_j} denote the simulator or annotated physical state at boundary j. For predicate p in a task's predicate set $\mathcal{P}$, let $y_{j,p} = f_p(s_{t+h_j}, \ell) \in \{0, 1\}$ be its ground-truth truth value. Examples include object containment, support, and fixture opening; f_p is the corresponding task-specific labeling rule. Let $\rho_{j,p} \in \{0, 1\}$ indicate whether predicate p is required at boundary j under the labeled phase. Establishment events and the mask for preservation supervision are:

$$y^{\text{est}}_{j,p} = (1 - y_{j-1,p})y_{j,p}, \qquad \mu_{j,p} = \rho_{j-1,p}\rho_{j,p}y_{j-1,p}. \tag{10}$$

The establishment target is $y^{\text{est}}_{j,p}$ whenever this transition is relevant and observed. Preservation predicts $y_{j,p}$ only where $\mu_{j,p} = 1$: the condition held before the segment and is required at both boundaries. Missing labels add a further availability mask. During an intentional release, the task no longer requires a grasp, so its preservation mask is zero. An accidentally lost grasp during transport still receives a negative preservation target. Requirements spanning an entire segment must be labeled consistently with intermediate events; endpoint labels alone cannot detect a brief violation followed by recovery.

For a nonempty set $\mathcal{P}_{\text{goal}}$ of required terminal predicates, the conjunction target is defined from ground-truth labels rather than assuming independence between predicted probabilities:

$$y^{\text{conj}}_j = \prod_{p \in \mathcal{P}_{\text{goal}}} y_{j,p}. \tag{11}$$

Contact and stable grasp are separate targets: proximity or momentary collision alone does not establish a stable grasp. Their labeling needs contact, aperture, relative-motion criteria, and temporal windows. A phase label $c_j \in \{1, \ldots, C\}$ identifies one of C applicable stages. Transition validity is task-dependent and must allow all legal progress within a segment, including more than one adjacent phase change when the temporal resolution permits it. Continuous fixture targets, such as joint displacement or angle, require declared units and normalization. No numerical labeling thresholds are inferred from task names.

Normalized supervision. Let $\mathcal{I}_{\text{bin}}$ be the set of valid binary targets across examples, segments, and binary heads. For target y_i and predicted probability $\widehat{p}_i$, binary cross-entropy (BCE) is averaged over valid labels:

$$\mathcal{L}_{\text{bin}} = \frac{1}{\max(1, |\mathcal{I}_{\text{bin}}|)} \sum_{i \in \mathcal{I}_{\text{bin}}} \left[-y_i \log \widehat{p}_i - (1 - y_i) \log(1 - \widehat{p}_i)\right]. \tag{12}$$

An empty sum contributes zero, and the denominator prevents division by zero when all labels are missing. Probabilities are numerically clipped away from zero and one when evaluating the logarithms. Binding uses cross-entropy against a soft assignment over entities derived from the annotated regions; phase uses categorical cross-entropy; continuous normalized fixture states use regression loss. With nonnegative weights $\beta_{\text{bind}}, \beta_{\text{bin}}, \beta_{\text{phase}}, \beta_{\text{fixture}}$, these form:

$$\mathcal{L}_{\text{sem}} = \beta_{\text{bind}}\mathcal{L}_{\text{bind}} + \beta_{\text{bin}}\mathcal{L}_{\text{bin}} + \beta_{\text{phase}}\mathcal{L}_{\text{phase}} + \beta_{\text{fixture}}\mathcal{L}_{\text{fixture}}. \tag{13}$$

Losses average only over applicable, observed targets. Contact, grasp, establishment, preservation, conjunction, and transition-validity heads are included in $\mathcal{L}_{\text{bin}}$; they need not share a label mask. Binding targets for broad destinations may be distributions rather than one-hot object identities. The annotation-to-entity matching rule must respect the encoder's current entity ordering; a raw object ID is not an entity-slot index.

A.4 Prefix-Conditioned Flow Matching

For repair training, choose a demonstrated action chunk A^+ and a prefix length $b \in \{0, \ldots, H-1\}$. The mask M_b fixes all action coordinates before b. Its complement $\bar{M}_b$ contains exactly $(H - b)d_a$ coordinates. The conditional path A^τ_b and repair objective are:

$$\begin{aligned} A^\tau_b &= M_b \odot A^+ + \bar{M}_b \odot [(1 - \tau)\epsilon + \tau A^+], \\ \mathcal{L}_{\text{repair}} &= \mathbb{E}\frac{\left\|\bar{M}_b \odot [v_\psi(A^\tau_b, \tau \mid Z_t, \ell, M_b, M_b \odot A^+) - (A^+ - \epsilon)]\right\|^2_F}{(H - b)d_a}. \end{aligned} \tag{14}$$

The expectation covers training contexts, demonstrated chunks, the prefix-length sampling distribution, Gaussian noise, and uniform flow time. A completely fixed chunk ($b = H$) is excluded because it has no suffix to learn. Ordinary generation is the case $b = 0$, with a zero mask and empty anchor. Thus the same conditional interface supports both full generation and repair.

Boundary and derivative checks. At $\tau = 0$, the path contains the exact demonstrated prefix and a noisy suffix. At $\tau = 1$, it equals A^+ in every coordinate. Its derivative is $\bar{M}_b \odot (A^+ - \epsilon)$, yielding Eq. 14. In the inference ODE, multiplying both sides of Eq. 7 by M_b gives a zero derivative because $M_b \odot \bar{M}_b = 0$. The prefix therefore stays exactly equal to the anchor throughout integration. These statements concern the masking algebra; they do not guarantee that a sampled suffix will be successful or that an imagined-valid prefix is executable.

A.5 Joint Training and Decision Procedure

Representation learning. The representation term $\mathcal{L}_{\text{rep}}$ combines the retained backbone objective with consistency between predicted role states and those encoded from observed future frames. For sg denoting stop-gradient, a role-state consistency term can be expressed as:

$$\mathcal{L}_Z = \mathbb{E} \frac{1}{5dJ} \sum_{j=1}^{J} \left\| \widehat{Z}_{t,j} - \text{sg}(Z_{t+h_j}) \right\|_F^2 . \tag{15}$$

Writing the retained backbone objective as $\mathcal{L}_{\text{backbone}}$ and its role-consistency weight as $\lambda_Z \geq 0$ gives $\mathcal{L}_{\text{rep}} = \mathcal{L}_{\text{backbone}} + \lambda_Z \mathcal{L}_Z$. The backbone objective supplies representation regularization; role consistency alone is not an anti-collapse guarantee. For LeWM, its published objective combines prediction with a Gaussian embedding regularizer (Maes et al., 2026). The Cosmos initialization, feature extraction point, training objective, and frozen/trainable partition are registered in Appendix C. The stop-gradient convention above is part of this explicit specification and requires confirmation against the trained system.

Uncertainty interface. The selector's u_t^k summarizes uncertainty of the semantic predictions. A fully specified choice is the mean normalized entropy over applicable categorical heads (binary heads count as two-category distributions) and predicted boundaries. If $\mathcal{D}$ indexes these distributions, with probabilities $\widehat{p}_{i,c}$ over $C_i \geq 2$ categories, the entropy statistic is:

$$u_t^k = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \frac{-\sum_{c=1}^{C_i} \widehat{p}_{i,c} \log \widehat{p}_{i,c}}{\log C_i}. \tag{16}$$

Here $0 \log 0$ is defined as zero; continuous fixture outputs are excluded. This gives $u_t^k \in [0, 1]$ for nonempty $\mathcal{D}$. Entropy measures ambiguity in predicted labels, not epistemic uncertainty or a calibrated failure probability. The original experiment description does not identify its estimator, so Eq. 16 is a specification choice to confirm, not an additional experimental finding.

Data and optimization. The training description uses expert trajectories, semantic transition labels, and hard-negative action pairs; later training can incorporate deviation and repair examples. Episode splits must precede window extraction so neighboring frames do not cross training and evaluation partitions. A negative chunk requires a validity label, and an observation-compatible prefix must accompany a repair target. Neither a loss weighting schedule nor a fixed number of training stages follows from the mathematical formulation. Appendix C registers the settings that connect this specification to the reported runs.

B Additional Experimental Results

The following tables retain the supplied experimental measurements. Differences between nearby numbers in separately supplied tables are not silently reconciled: for example, the monolithic selector is 50.5% in Table 1 and 50.2% in the perception diagnostic; the soft-prior diagnostic is 75.1%, versus 75.3% for the main semantic selector. Appendix C registers the shared configuration; remaining nearby-number drifts may still reflect distinct repetitions. Reported $\pm$ values follow the three-seed (success) and five-run (efficiency) definitions in Appendix C.

Table 5: **Receding-horizon semantic planning.** R bounds additional repair proposals, and h_{exec} is the execution horizon. The rejection fallback and controller checks must be specified for each platform.

Step	Procedure at control time t
1	Encode the observation history, proprioception, and instruction into Z_t.
2	Draw K independent noises and integrate the action flow to obtain $A_t^1, \ldots, A_t^K$.
3	Predict J intermediate role states and semantic transitions for every candidate. Compute S_t^k and identify applicable violations.
4	Select the highest-scoring candidate. If repair is triggered, retain the prefix before its first violating segment; use an empty prefix if confidence is low without a localized boundary.
5	Generate up to R repaired proposals with the retained prefix clamped. Predict and score each new proposal again.
6	Execute the first h_{exec} actions of an accepted proposal, shortened if necessary to remain within the accepted prefix. If no proposal is accepted, invoke the platform's rejection fallback.
7	Acquire new observations and repeat. Never substitute predicted observations for the next real observation.

B.1 RoboCasa Per-Task Results

Table 6: **RoboCasa task breakdown with Cosmos.** Success (%) sorted by baseline success. The first 11 rows form the descriptive hard subset. Δ is +Repair minus Base in percentage points.

Task	Base	+Role	+Score	+Repair	Oracle@4	Δ
CloseBlenderLid	3	5	12	16	22	+13
PickPlaceDrawerToCounter	7	9	17	21	28	+14
TurnOffStove	18	22	34	41	52	+23
OpenCabinet	24	28	40	46	56	+22
TurnOnMicrowave	26	30	42	48	58	+22
CloseToasterOvenDoor	37	40	50	55	64	+18
CoffeeSetupMug	37	40	48	53	62	+16
TurnOnSinkFaucet	37	40	48	52	60	+15
PickPlaceCounterToCabinet	43	46	54	58	66	+15
PickPlaceCounterToStove	45	48	55	59	68	+14
TurnOnElectricKettle	45	48	55	58	67	+13
CloseFridge	61	63	67	70	73	+9
SlideDishwasherRack	61	63	68	70	72	+9
OpenStandMixerHead	69	69	69	68	68	-1
OpenDrawer	70	72	76	78	82	+8
PickPlaceToasterToCounter	72	72	73	73	73	+1
PickPlaceSinkToCounter	74	74	73	72	72	-2

The 17 rows in Table 6 are the source of the macro-averages in Table 2. All tasks receive equal weight. The 11-task subset is formed using the reported baseline rates at or below 45%. It is useful for describing the distribution of gains, but selecting tasks by baseline performance does not establish a separate generalization result. Suffix repair improves on Base for 15 tasks and decreases performance for two; no claim of per-task statistical significance is made without rollout counts and repeated-run information.

B.2 Cross-Suite Transfer

On the supplied matched target-data and update-budget comparisons, Object pretraining improves adapted performance over scratch by 8.3 points on Goal and 8.1 points on Long. On Spatial, the adapted model is 0.7 points below scratch. Joint pretraining improves adaptation on the reported held-out tasks by 8.6 points; the exact joint-training and held-out-task lists remain necessary to

Table 7: **Cross-suite transfer on LIBERO.** Selected top-1 success (%). Adapted and scratch use the same target-suite data and update budget in the supplied comparison. The exact splits and meaning of ± require final protocol annotation.

Train → Test	**Zero-shot**	**Adapted**	**Scratch**
Object → Spatial	58.4±3.5	72.1±3.2	72.8±3.7
Object → Goal	52.7±4.1	74.6±2.9	66.3±3.6
Object → Long	27.9±3.0	58.7±4.0	50.6±4.4
Joint → held-out tasks	65.2±3.3	78.4±2.6	69.8±3.5

establish the scope of that comparison. These outcomes support selective transfer benefits rather than a uniform adaptation advantage.

B.3 Perception and Action-Sensitivity Diagnostics

Table 8: **Role and segmentation diagnostics.** Metrics are on a 0–100 scale; success uses the same selector and candidate budget within this comparison. Bind.: binding accuracy; Rel.: relation AUROC; Phase: phase-boundary F1; SR: rollout success.

Representation	**Bind. Acc.**	**Rel. AUROC**	**Phase F1**	**Rollout SR**
Monolithic latent	60.5	69.4	54.0	50.2
Instance slots	74.6	76.5	64.1	55.8
SemRole (no masks)	88.3	89.1	80.8	72.6
SemRole + soft prior	91.8	90.7	84.1	75.1
SemRole + corrupt prior	86.2	83.9	77.4	70.8
SemRole + oracle masks	95.4	94.6	87.2	78.4

The soft-prior model improves binding accuracy and downstream success relative to the no-mask model. The latter remains executable without inference-time masks, while corruption reduces success below the no-mask result. Oracle masks provide a reference for this particular perception experiment, not a theoretical bound on other models or protocols. Binding accuracy, relation area under the receiver operating characteristic curve (AUROC), and phase F1 are reported on a 0–100 scale.

Table 9: **Action-sensitivity diagnostics.** Role error is the supplied latent-prediction error metric (normalization pending); gap is the mean positive-minus-negative score; rank accuracy is in percent. All entries retain their reported variation.

Variant	Role error ↓	Gap ↑	Rank acc. ↑
No semantic heads	0.176±0.011	0.061±0.019	60.8±2.9
No shuffled negatives	0.149±0.009	0.018±0.011	53.7±3.3
No preservation loss	0.158±0.010	0.108±0.023	69.4±2.6
No phase loss	0.141±0.008	0.087±0.020	66.8±2.8
SemRole	0.144±0.007	0.157±0.025	78.2±2.1

The action-sensitivity experiment evaluates a positive action against a labeled negative from the same context. Its score gap depends on score scale, so pairwise ranking accuracy is the more directly comparable ordering diagnostic. Removing phase loss produces the smallest reported role error but substantially worse ranking than the full model. Removing hard negatives reduces ranking to 53.7%. These comparisons support the decision relevance of the supervised distinctions; they do not imply that every loss is necessary for every task.

Table 10: **Complete-system size and component costs.** Parameters in millions; peak training memory in GB; generation and scoring latency in ms per replan at $K = 4$. Source protocol: A100-40GB, BF16, training batch 32, inference batch one, five runs after 100 warm-up replans. †Published backbone size. Timings do not establish end-to-end control latency.

Model	Total	Trainable	Memory	Gen.	Score
LeWM	15.0M†	15.0M	4.3±0.2	n/a	4.4±0.3
Latent + FM	35.8M	35.8M	8.1±0.3	28.7±1.1	4.8±0.3
Instance slots + FM	39.1M	39.1M	8.9±0.4	31.0±1.4	7.6±0.6
SemRole (world only)	24.6M	24.6M	6.6±0.3	n/a	7.1±0.5
SemRole + FM	45.4M	45.4M	9.2±0.4	30.2±1.2	7.3±0.5

B.4 Parameter and Component-Cost Accounting

The supplied model inventory reports 45.4M parameters for the complete SemRole system with its action head. The action head adds 20.8M to the 24.6M world-model stack. Thus the complete system should not be described as a 15M model merely because it uses the LeWM architecture. Generation and scoring timings exclude any unreported overhead, repeated repair, sensing, and actuation; adding the two component means does not establish a closed-loop robot control frequency. No Cosmos efficiency measurements are supplied.

B.5 Galbot Physical Evaluation

Table 11: **Galbot physical evaluation with the Cosmos backbone.** Each cell is closed-loop success (%) over N=50 independent, reset-matched trials. Aggregate is the unweighted mean of the two tasks. Oracle-style selectors are not used on the robot.

Method	Tabletop placement	Microwave retrieval	Aggregate
Cosmos-conditioned baseline	68.0	42.0	55.0
Cosmos + SemRole	**88.0**	**74.0**	**81.0**

Protocol. Two physical tasks; N=50 trials per task per method with independent episode resets. Success is binary closed-loop task completion; timed-out, operator-intervened, or e-stopped trials count as unsuccessful (timeout 60 s). Matched factors: same robot, observation interface, K=4, and trial count; the decision stack is the comparison variable. Adaptation: 20 teleoperated demonstrations per task and 2k fine-tune updates after simulation pretraining (Appendix C). Table 11 supports a within-backbone physical comparison under this registered budget and is not evidence of zero-shot transfer. Any reported Galbot ± is a bootstrap standard deviation over the N=50 trials of the single adapted checkpoint (Appendix C).

Failure analysis to accompany the figure. Occlusion, premature release, and phase disagreement remain relevant hypotheses for role grounding and constraint evaluation. Episode-level failure annotations are not supplied in the current summaries, so no frequency ranking is asserted here. A completed analysis should identify the task, show observed frames and the predicted transition, and distinguish perception error from candidate scoring or controller failure.

C Experimental Protocol

This appendix records the author-confirmed configuration used for the reported runs. Simulation (LIBERO, RoboCasa) and physical (Galbot) evaluations are separate settings. Where a field is still unset, it is marked with **[To complete: ·]** rather than invented (cf. the claim-boundary documentation style of registering only confirmed comparisons).

C.1 BACKBONES

LEWM line (LIBERO). We initialize from the official LEWM encoder–predictor of Maes et al. (Maes et al., 2026), totaling 15.0M parameters (ViT-Tiny 5.0M + predictor 10.0M), pretrained on the same LIBERO RGB streams. The feature tap is the last ViT block's patch tokens (width 192, patch 16×16). Inputs are 128×128 RGB from two views (agentview and eye-in-hand), with observation history $T_{\text{obs}} = 3$ frames at a 10 Hz subsample of the 20 Hz control stream. On LIBERO the entire 15M trunk is trainable, together with the role binder, relation/phase encoders, semantic heads, selector, and flow-matching (FM) head; no trunk layers are frozen. Parameter counts for the reported systems are 24.6M for the world-model stack without the FM head and 45.4M with the FM head.

COSMOS line (RoboCasa / Galbot). The role, FM, and selector hyperparameters match the LEWM line; only the visual trunk changes. We use a Cosmos-Predict1 video world-model trunk (NVIDIA, 2025). The feature tap is the frozen spatial tokenizer plus the last trainable temporal block (token width 256 in the reported runs). Frozen modules: the Cosmos tokenizer and 80% of the temporal blocks. Trainable modules: the last two temporal blocks and all SEMROLE heads (same role width d as the LEWM line). Images are 256×256 RGB. RoboCasa uses 2–3 kitchen views; Galbot uses head and wrist cameras. Exact Cosmos checkpoint identifier: **[To complete: Cosmos-Predict1 checkpoint name / revision]**.

C.2 STATE AND ACTIONS

Shared across both backbone lines:

Symbol	Value
Soft entities N	8
Role width d	256
Binding key d_k	64
Geometry bias d_g	32
Phase history L	8
Action dim. d_a	7 (Δpos 3 + Δrot 3 axis-angle + gripper 1)
Action normalization	train-set per-dim mean/std; gripper mapped linearly to $[-1, 1]$
Chunk horizon H	16
Segments J	4
Boundaries h_j	$4, 8, 12, 16$
Execution horizon h_{exec}	8 (execute half a chunk, then replan)
Control frequency	20 Hz on LIBERO / RoboCasa; Galbot commands at 10 Hz, interpolated to 20 Hz
Candidate budget K	4 (primary); sweep $K \in \{1, 2, 4, 8\}$ on LIBERO

C.3 DATA

LIBERO. We use the four official suites (Object, Spatial, Goal, Long), each with 10 tasks. Each task provides 50 expert demonstrations. Episodes are split 80/10/10 into train/val/test *before* window extraction, so neighboring frames never cross partitions. Evaluation uses 50 closed-loop trials per task and three independent training seeds $\{0, 1, 2\}$, i.e. $10 \times 50 \times 3$ trials per suite. Success follows the official LIBERO completion predicate. Timeouts use the official maxima (600 steps on Object/Spatial/Goal; 800 on Long). No human intervention is allowed in simulation.

RoboCasa. We use 17 atomic kitchen tasks with 50 successful demonstrations per task (RoboCasa human/scripted mixture; exact source mix: **[To complete: human vs. scripted counts]**). The same 80/10/10 episode split is applied. Evaluation uses 50 trials per task $\times$ 3 training seeds. Timeout is 500 environment steps. The Hard-11 subset is reported post hoc only and is not used for model selection.

Galbot. We evaluate two physical tasks: tabletop placement and microwave retrieval. Models are pretrained in simulation with the COSMOS+SEMROLE stack, then fine-tuned on the robot with 20 teleoperated demonstrations per task for 2k update steps (adaptation budget as registered here; not

claimed as zero-shot transfer). Each method is evaluated with 50 independent, reset-matched trials per task. Timeout is 60 s. Any human takeover or e-stop counts as failure.

C.4 TRAINING

Shared settings unless noted:

Item	Value
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.01
Learning rate	3×10^{-4}, cosine decay, 500-step warmup
Batch size	32 chunks
Updates	LIBERO 80k; RoboCasa 60k; Galbot fine-tune 2k
AMP	BF16
Negatives	1 negative per positive chunk: segment shuffle / reverse / trajectory swap / release perturbation (25% each)
Prefix b (training)	uniform over $\{0, 4, 8, 12\}$; $b{=}H$ excluded

Loss weights (identical on both backbone lines):

$$\lambda_{\text{rep}} = 1.0, \quad \lambda_Z = 0.5, \quad \lambda_{\text{sem}} = 1.0,$$
$$\lambda_{\text{FM}} = 1.0, \quad \lambda_{\text{rank}} = 0.5, \quad \lambda_{\text{repair}} = 0.5,$$
$$\beta_{\text{bind}} = 1.0, \quad \beta_{\text{bin}} = 1.0, \quad \beta_{\text{phase}} = 0.5, \quad \beta_{\text{fixture}} = 0.5, \quad \beta_{\text{dyn}} = 0.5.$$

Ranking margin $m = 0.2$.

C.5 SEMANTICS

Phase vocabulary. Six ordered phases:

$$\text{approach} \rightarrow \text{grasp} \rightarrow \text{transport} \rightarrow \text{align} \rightarrow \text{release} \rightarrow \text{complete}.$$

Legal transitions stay in the current phase or advance to the next phase; a segment may advance by at most one step. Jumps such as $\text{approach} \rightarrow \text{release}$ are forbidden.

Contact and grasp (simulation). Contact is true when gripper–target contact force exceeds $0.5\,\text{N}$ or a geom contact flag is set, sustained for $\geq 100\,\text{ms}$. A stable grasp requires contact, gripper aperture $< 0.6\times$ the object's characteristic width, and relative velocity $\|v_{\text{rel}}\| < 0.02\,\text{m/s}$ for a continuous $200\,\text{ms}$ window. During intentional release, the preservation mask is zero. An accidental drop during transport retains a negative preservation target.

Predicates. LIBERO / RoboCasa use simulator-native predicates (ON, IN, OPEN, CLOSE, TURNON, ...) with environment-official thresholds (e.g., door angle $> 40^\circ$ counts as open; stove-top uses discrete on/off). Establishment and preservation follow Eq. 10; conjunction is the logical AND of required terminal predicates (Eq. 11).

Binding. Soft labels are the IoU-normalized distribution between annotated regions and entity attention; entities with $\text{IoU} < 0.1$ receive mass zero.

C.6 Generation, Selection, and Repair

Item	Value
Flow network	4-layer DiT-style MLP/attention, hidden width 256, AdaLN on Z_t
ODE	Euler, $N_\tau = 10$
Noise	$\epsilon \sim \mathcal{N}(0, I)$; evaluation noise seeds are decoupled from environment seeds via noise_seed $= 1000 +$ trial_id
Threshold η	0.40 on all three lines (validation-chosen once for $\approx 70\%$ accept rate; scores uncalibrated; shared to avoid platform-specific thresholds)
Repair budget R	at most 2 suffix repairs
Low-confidence boundary	b=0 full resample, counted against the same R budget
Rejection fallback	if budget exhausted, execute the first h_{exec} steps of Candidate 0; further failure stops the episode and counts as unsuccessful

Uncertainty u_t^k is the mean normalized entropy over applicable categorical heads (Eq. 16).

C.7 Evaluation and Reported ±

Item	Value
Primary metric	closed-loop success; LIBERO suites equally weighted; RoboCasa unweighted mean over 17 tasks; Galbot unweighted mean over 2 tasks
± (LIBERO / RoboCasa success, ranking, role error)	mean ± std over 3 training seeds $\{0, 1, 2\}$ (not trial-level SE); each seed evaluated with 50 trials/task
± (Galbot success)	single adapted checkpoint; 50 reset-matched trials/task/method; ± = bootstrap std over the 50 trials (not a 3-seed training std)
± (latency / memory)	mean ± std over 5 timing runs; A100-40GB; BF16; batch 32 (train) / 1 (infer); 100 warm-up replans
Oracle@4	at each replan, select k by true H-step chunk success under simulator rollout of that chunk; not whole-episode hindsight; excludes newly repaired candidates
Fairness	full resampling and suffix repair each allowed at most $1+R = 3$ flow integrations per decision

C.8 Configuration Summary

Table 12: **Registered experimental configuration** for the reported SemRole runs. Fields marked **[To complete: ·]** remain author-editable identifiers and do not affect the numerical rates already reported.

Item	Registered fill
Backbones	LeWM 15M full fine-tune (tap width 192); Cosmos tokenizer frozen + last 2 temporal blocks trainable (tap width 256); checkpoint **[To complete: Cosmos-Predict1 id]**.
Data	50 demos/task; 80/10/10 episode split; eval 50 × 3 seeds (sim); Galbot: 20 teleop demos/task (40 total), 2k fine-tune steps, single ckpt, N=50 trials/task/method.
State / actions	N=8, d=256, d_k=64, d_g=32, L=8, H=16, J=4, h_j=4j, h_{exec}=8; 20 Hz (Galbot 10 Hz → 20 Hz); 7-D z-scored actions.
Training	AdamW 3×10^{-4}; 80k / 60k / 2k updates; batch 32; loss weights as above.
Semantics	6-phase chain; contact 0.5 N / 100 ms; grasp + aperture + v_{rel}; predicates per environment; binding by IoU soft labels.
Generation	Euler-10; K=4; R=2; η=0.40 (shared).
Evaluation	Official success predicates; timeouts 600/800 (LIBERO), 500 (RoboCasa), 60 s (Galbot); intervention / e-stop = failure.
Statistics	Sim success ± = 3-seed std; Galbot ± = bootstrap over 50 trials; efficiency ± = 5-run std.
Oracle@4	Per-replan chunk outcome on the original candidate pool only.

Oracle scope. An oracle is an upper bound only over the choices and continuation protocol it actually evaluates. Choosing the best of four complete policy rollouts differs from using ground-truth outcomes to choose one chunk at every replan. Repair adds new candidates and therefore changes the set over which an oracle would need to be computed. The manuscript's original-pool Oracle@4 numbers follow the per-replan chunk-outcome definition above and are not extended to repaired pools.

Table-to-run notes. Nearby table discrepancies retained from source summaries (e.g., instance-slot 55.8% vs. 55.0%; selector 50.5% vs. 50.2%; soft-prior 75.1% vs. main semantic 75.3%) are not silently reconciled; they may reflect distinct configurations or repetitions. The full-model 83.0% mean appears in both the main LIBERO table and the representation ablation under the same registered protocol unless a future run map states otherwise.