LET RELATIONS SHAPE THE WORLD: LEARNING INSTRUCTION-CONDITIONED WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Relational instructions can express the same task across different participants, yet demonstrations provide only scene-specific visual outcomes. We introduce **RWM**, a *relation-shaped world model* that learns language-grounded object interactions to predict desired futures. Its relational transformer, **RELATRON**, grounds language in ordered object pairs through a read–write operation, with separate branches determining which interactions contribute and what information they convey. Both branches are trained end-to-end with the future predictor using demonstrated visual changes, without explicit object-pair relation annotations. Shared pairwise operators provide an inductive bias for reuse across participants, while their spatial context augments rather than replaces dense visual features. The predictor generates short-term and terminal visual goals supervised by future observations rather than symbolic goal-state targets. These goals guide action proposals, whose predicted consequences are compared with the short-term goal in a shared visual representation space. Controlled evaluations on our **TELETRAAN I** benchmark show higher future-goal agreement under simultaneous target–reference color shifts, improved instructed-object grasping under joint target–distractor changes, and higher mean pickup success on colors excluded from task training. Ablations show contributions from both interaction selection and language-conditioned messages. On LIBERO, consequence-based selection improves mean success over single-proposal execution. These results support learning relational conditioning from demonstrations to apply familiar task requirements under changes in participant appearance and scene context.

1 INTRODUCTION

Visual world models predict action consequences and support planning toward visual goals (Zhou et al., 2024a; Assran et al., 2025). Language-guided planning also requires constructing the goal itself. An instruction such as "place the block to the left of the bowl" specifies participating objects and a requested relational change, rather than a fixed destination. Changing "left" to "right" should change the desired future for the same scene, while changing participant appearances should not require relearning the relational requirement. Demonstrations provide scene-specific outcomes, however, rather than annotations of the intermediate interactions needed to predict them. We therefore ask: *how can task-relevant relational grounding be learned from demonstrated outcomes and reused across participant changes?*

Language-guided generators learn visual plans and subgoals (Du et al., 2023; Black et al., 2024), while compositional generators and relational planners expose reusable task structure (Zhou et al., 2024b; Gkanatsios et al., 2023). We place this structure in the language-conditioning pathway: object interactions determine how the instruction shapes prediction, while dense visual features retain the information needed to realize the requested change. The relational computation is learned jointly with future prediction, without explicit object-pair relation labels or symbolic goal configurations as prediction targets.

We introduce **RWM**, a *relation-shaped world model*, built around the relational transformer **RELATRON**. Language tokens read ordered object-pair features and write scene-grounded information back to those pairs. Separate branches learn which interactions contribute and what task information they convey, allowing the same participants to receive different conditioning under different

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.

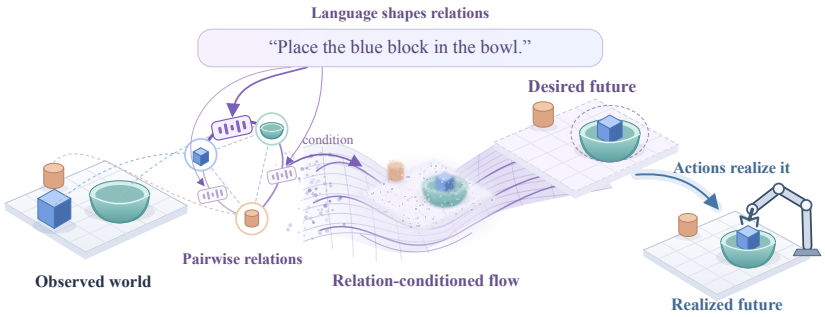

Figure 1: **From relational instructions to action.** Language-grounded object interactions shape desired visual futures that guide robot actions.

instructions. Shared pairwise operators provide an inductive bias for reuse across participants. Their spatial context augments dense visual features to predict demonstrated changes; paired-instruction contrast further encourages predictions to reflect the requested outcome.

A shared generator learns short-term and terminal feature changes from future observations in demonstrations. The terminal future provides a task-completion goal, while the short-term future supplies a target at the action-chunk horizon. An action expert receives both futures and grounded current-state features to propose candidate chunks. A separate model predicts their endpoints without receiving the instruction, and RWM compares these consequences with the short-term goal in the same visual representation space. It executes a prefix and replans, refreshing the short-term goal while retaining the terminal goal. This separates constructing the requested outcome from predicting what candidate actions will achieve.

We introduce **TELETRAAN I** to assess visual goal construction separately from execution under changes in task participants and scene context. Under controlled conditioning comparisons, RELATRON achieves higher future-goal retrieval accuracy under both target-only and joint target–reference color shifts, with joint-shift accuracy of 60.42% versus 55.21% for Object Transformer (OCWM-style adaptation; Jeong et al., 2025). RELATRON-based policies also improve instructed-object manipulation under target–distractor appearance changes and achieve 57.50% mean pickup success on colors excluded from task training, versus 39.17% for the strongest baseline in that evaluation. Qualitative visualizations show task-relevant interaction selection without relation annotations, and ablations establish contributions from both selection and language-conditioned messages. LIBERO and real-robot approach and pick-and-place experiments provide additional evaluation of goal-directed control. Our contributions are:

- **Relational conditioning for visual goal prediction.** RELATRON grounds language in ordered object interactions, separating interaction selection from message content while retaining dense visual information for scene-specific prediction.
- **A visual-goal interface connecting language to control.** RWM combines demonstration-supervised short-term and terminal futures with action proposals and instruction-free consequence prediction for closed-loop, goal-directed selection.

2 Related Work

Language-conditioned visual goals. UniPi (Du et al., 2023) generates video plans, SuSIE (Black et al., 2024) synthesizes image subgoals, and Video Language Planning (Du et al., 2024) combines language-level search with video prediction. RoboDreamer (Zhou et al., 2024b) supports compositional generation by factorizing instructions into reusable primitives. RWM likewise constructs visual futures from language, but structures conditioning through ordered object interactions in the observed scene to learn task transformations across participants.

Language-conditioned relational computation. Relation Networks (Santoro et al., 2017) use shared pairwise computations, LCGN (Hu et al., 2019) conditions graph connectivity and mes-

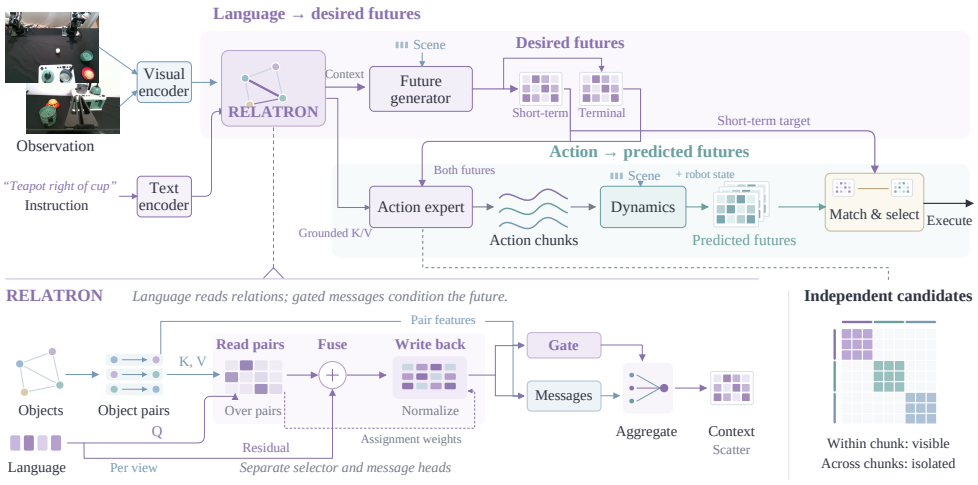

Figure 2: **RWM architecture.** Language-grounded object interactions condition desired-future generation. The generated goals guide action proposals, and predicted chunk endpoints provide the consequences used for goal-directed selection.

sage passing on language, and C-SWM (Kipf et al., 2020) learns object-centric graph dynamics. RWM adapts these building blocks to instruction-specified future prediction: RELATRON performs language-grounded read–write operations over ordered pair features, separating interaction selection from message content. Its spatial context conditions future-feature generation, with demonstrations across participants supervising shared pairwise parameters.

Relational goal generation. Compositional energy models generate and edit images from relational descriptions (Liu et al., 2021), while EBMPlanner (Gkanatsios et al., 2023) optimizes goal arrangements using spatial-predicate energies. StructDiffusion (Liu et al., 2023) generates language-conditioned 6-DoF goal poses, and Diffusion-CCSP (Yang et al., 2023) composes constraint-specific diffusion models over factor graphs. Unlike pose- and constraint-space planners, RWM uses relational features to condition visual future prediction, supervised by subsequent observations rather than explicit object configurations or constraint assignments.

Predictive models for goal realization. DINO-WM (Zhou et al., 2024a) and V-JEPA 2-AC (Assran et al., 2025) predict action consequences in visual feature space to plan toward image goals. GR-1 (Wu et al., 2024) and CauVA (Li et al., 2026) couple visual prediction with action generation, while role-structured world models evaluate actions through semantic transition prediction (Cheng et al., 2026). RWM focuses on relational goal construction: a relational predictor generates the desired future, while a separate action-conditioned predictor estimates candidate outcomes without receiving the instruction. Comparing these predictions in a shared visual representation space guides action selection.

3 METHOD

RWM first constructs the future requested by language, then selects actions whose predicted consequences approach that goal (Figure 2). A frozen V-JEPA 2 encoder maps two camera images to visual features Z_t, and a SigLIP text encoder maps the instruction to text tokens L. The desired futures comprise a short-term goal H_t for an action chunk $U_t = (a_t, \ldots, a_{t+h-1})$ and a terminal goal T for task completion. Throughout, feature vectors occupy matrix rows; MSE and MAE average over all tensor entries, and LN acts on each token's feature dimension.

3.1 RELATIONAL CONDITIONING WITH RELATRON

Mask pooling and learned projections produce node features X. Shared pair encoders combine ordered object features and geometry into pair matrices P^{sel} and P^{msg} for the selector and message branches. Each branch operates on the valid pair set $\mathcal{E}$ within a view; parameters are shared across views. All row-wise networks share their parameters across objects or pairs.

Language read–write. For either branch, write P for its pair features and L for its projected language tokens. With attention head r, language reads the pairs and writes grounded content back through the same assignments:

$$\begin{aligned} A_r &= \mathrm{softmax}_{\mathcal{E}}\left(\frac{(LW_{Q,r})(PW_{K,r})^\top}{\sqrt{d_r}}\right), \\ U &= L + \left[\Big\|_r A_r P W_{V,r}\right] W_O, \qquad G = D_A^{-1}\bar{A}^\top U. \end{aligned} \tag{1}$$

Here $\bar{A}$ averages heads, $\|$ concatenates features, and $D_A = \mathrm{Diag}(\max(\bar{A}^\top \mathbf{1}, \epsilon))$ normalizes incoming assignments. The two branches independently form G^{sel} and G^{msg}.

Gated messages and spatial context. Let X_1, X_2 gather the node features at each ordered pair's endpoints, and let Γ collect pair geometry. With gate budget B and temperature τ_g, selection and messages are

$$\begin{aligned} \boldsymbol{\alpha} &= \min(B, |\mathcal{E}|)\ \mathrm{softmax}\big(f_{\mathrm{gate}}(P^{\mathrm{sel}} + G^{\mathrm{sel}})/\tau_g\big), \\ R &= f_{\mathrm{rel}}([P^{\mathrm{msg}} \| G^{\mathrm{msg}}]), \qquad M = f_{\mathrm{msg}}([X_1 \| X_2 \| R \| \Gamma]). \end{aligned} \tag{2}$$

The incidence matrix Q assigns each message to its first endpoint; S contains object-to-patch mask weights. Aggregation and spatial scatter are

$$\widetilde{X} = \mathrm{LN}\Big(X + D_N^{-1/2} Q\, \mathrm{Diag}(\boldsymbol{\alpha}) M\Big), \qquad C_t = D_S^{-1} S^\top \widetilde{X} W_C, \tag{3}$$

where $D_N = \mathrm{Diag}(\max(Q\mathbf{1}, 1))$ and $D_S = \mathrm{Diag}(\max(S^\top \mathbf{1}, \epsilon))$. The per-view fields are concatenated for future prediction. RELATRON's node updates are equivariant to object re-indexing, and its spatial field is invariant to that indexing (Appendix A.2).

3.2 GENERATING DESIRED FUTURES

A shared future Transformer processes both views and uses a horizon embedding to distinguish H_t from T. Noisy future tokens attend to keys/values and the current scene with relational context:

$$K_t = Z_t W_K + \gamma_K C_t U_K, \qquad V_t = Z_t W_V + \gamma_V C_t U_V, \tag{4}$$

where projections are block-specific and γ_K, γ_V are learned scalars. This interface grounds denoising in the current visual state and the language-specified interactions.

Clean-change prediction. Following JiT (Li & He, 2025), the network predicts clean visual change and is trained with a velocity-space loss. For horizon $\eta \in \{H, T\}$, let $\Delta_\eta = \mathrm{Norm}_\eta(Z_\eta^{\mathrm{gt}} - Z_t)$. Gaussian noise ξ defines $y_\tau = (1-\tau)\xi + \tau\Delta_\eta$; then

$$\begin{aligned} \widehat{\Delta}_\eta &= D_\theta(y_\tau, \tau, \eta; Z_t, C_t), \qquad \widehat{v}_\theta = \frac{\widehat{\Delta}_\eta - y_\tau}{\rho(\tau)}, \\ v^\star &= \frac{\Delta_\eta - y_\tau}{\rho(\tau)}, \qquad \mathcal{L}_{\mathrm{flow}}^\eta = \mathbb{E}\ \mathrm{MSE}(\widehat{v}_\theta, v^\star). \end{aligned} \tag{5}$$

With $\rho(\tau) = \max(1-\tau, \epsilon_\tau)$, the full objective is $\mathcal{L}_\eta = \mathcal{L}_{\mathrm{flow}}^\eta + \lambda_c \mathcal{L}_{\mathrm{change}}^\eta + \mathbf{1}_{\{\eta=T\}} \lambda_i \mathcal{L}_{\mathrm{contrast}}$. The additional terms weight changing patches and, for examples with paired instructions, contrast correct versus alternative instructions under the same scene, noise, and target. Sampling integrates the velocity from noise at $\tau = 0$ to $\tau = 1$; denormalizing the sampled change and adding Z_t yields the desired future. Loss and sampling details appear in Appendix A.3.

3.3 Future-Conditioned Action Generation

A $\pi_{0.5}$-initialized action expert receives H_t, T, and grounded current K/V from the first short-term generator block. Adapters with camera and temporal-role embeddings form conditioning $\mathcal{B}_t$. For a normalized, padded demonstration chunk A and Gaussian noise ε, conditional flow matching uses

$$\begin{aligned} A_\sigma &= (1-\sigma)A + \sigma\varepsilon, \\ \mathcal{L}_{\text{action}} &= \mathbb{E}\ \text{MSE}(v_\phi(A_\sigma, \sigma \mid \mathcal{B}_t), \varepsilon - A)\,. \end{aligned} \tag{6}$$

At inference, integrating from $\sigma = 1$ to 0 under shared conditioning and independent noise produces denormalized candidates $\{U_t^{(k)}\}_{k=1}^K$. Each chunk has full internal attention. Appendix A.3 describes future conditioning and consequence-based selection.

3.4 Consequence Prediction and Goal Realization

We adapt the V-JEPA 2-AC predictor (Assran et al., 2025) to predict chunk endpoints. Its action projection encodes the entire temporally concatenated chunk, and the shared predictor processes each view independently, conditioned on current robot state s_t. Write $\widehat{Z}_{t+h}^{(k)}$ for candidate k's concatenated predicted views:

$$\begin{aligned} \widehat{Z}_{t+h}^{(v)} &= F_\psi\Big(\text{LN}(Z_t^{(v)}), \text{vec}(U_t), s_t\Big)\,, \\ \mathcal{L}_{\text{AC}} &= \mathbb{E}\,\frac{1}{2}\sum_{v=1}^{2}\text{MAE}\Big(\text{LN}(\widehat{Z}_{t+h}^{(v)}), \text{LN}(Z_{t+h}^{(v)})\Big)\,, \\ k^\star &= \arg\min_k d(\widehat{Z}_{t+h}^{(k)}, H_t). \end{aligned} \tag{7}$$

Here vec concatenates controls in temporal order, and the frozen encoder supplies observed endpoint targets. The matching cost d averages squared differences between normalized features equally across views. RWM executes a prefix of the selected chunk and replans with a refreshed short-term goal; T is sampled once and retained throughout the episode.

3.5 Training Curriculum

Future training uses $\mathcal{L}_{\text{future}}^{(k)} = \mathcal{L}_T + \lambda_H(k)\mathcal{L}_H$. Training first optimizes T alone; subsequently, λ_H increases from zero to one while terminal supervision continues. Action training starts with ground-truth futures and a fixed future model, then progressively introduces generated goals and jointly optimizes $\mathcal{L}_{\text{action}} + \mathcal{L}_H + \mathcal{L}_T$. Visual and text encoders remain frozen. The consequence predictor is trained separately with $\mathcal{L}_{\text{AC}}$ and held fixed during selection.

4 Experiments

We evaluate RWM along the path from an instruction to a relationally specified future and its realization through action. The experiments address three questions: can the resulting goals support task execution, how does language reshape the predicted future and its participating relations, and how well do the predicted goals transfer across scene and instruction changes? We report task success, examine generated futures and relation selection, and evaluate future-goal agreement and its conditioning components.

Tasks and comparisons. TELETRAAN I comprises Bowl Placement and Multi-Object Manipulation (Figure 3). Bowl Placement evaluates target–reference relations, while Multi-Object Manipulation evaluates instructed-object grasping and pickup among competing objects. LIBERO evaluates manipulation across Spatial, Object, Goal, and Long suites. Real-robot experiments evaluate Approach and Pick-and-place, alongside instruction-conditioned future visualization. We compare alternative ways of conditioning future prediction on language and objects. Text AdaLN uses global language modulation; Language Cross-Attention conditions visual tokens directly on language; Object Transformer models object interactions; Naive Relation uses language-weighted pairwise messages. RELATRON, the relational conditioner used by RWM, selects pairwise interactions and modulates their message content with language.

Object identification. The Segment Anything Model (SAM) (Kirillov et al., 2023) produces object masks from RGB observations. Pooling the visual patches within each mask forms the object representations used for relational modeling.

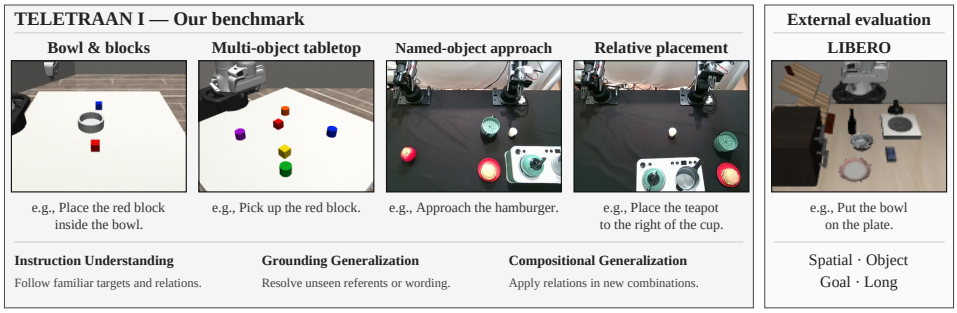

Figure 3: **Experimental setup.** Task settings and example instructions for Bowl Placement, Multi-Object Manipulation, named-object approach, relative placement, and LIBERO.

Evaluation. We measure task execution by closed-loop success and intended futures by future-goal retrieval accuracy. The latter retrieves the instructed outcome among alternative futures for the same observation; joint agreement requires the correct target, reference, and relation. Grasping and color pickup require grasping and lifting the instructed object. We visualize predicted futures by decoding the predicted visual features.

4.1 Task Success and Goal Realization

Tables 1 and 2 report closed-loop performance under changes in the instructed participants and across manipulation tasks.

Table 1: **Multi-Object Manipulation.** Grasping and zero-shot color pickup (%). (a) Success requires grasping and lifting the instructed object. Context changes the distractor; Joint changes the target and context. (b) Success requires grasping and lifting; OOD avg. is the mean over five held-out colors. ID denotes familiar scenes in (a) and seen colors in (b). Gray marks RWM; bold marks the best result in each column.

(a) Instructed-object grasping

Model / variant	ID	Target	Context	Joint
Object Transformer	54.17	41.67	60.42	46.88
Naive Relation	50.00	41.67	50.00	40.63
RWM (RELATRON)	**100.00**	**87.50**	**100.00**	**85.42**

(b) Zero-shot color pickup

Model / variant	ID	Black	Cyan	Gray	Pink	White	OOD avg.
Text AdaLN	50.00	4.17	50.00	16.67	**41.67**	16.67	25.83
Language Cross-Attention	0.00	0.00	4.17	0.00	0.00	0.00	0.83
Object Transformer	33.33	16.67	25.00	16.67	33.33	12.50	20.83
Naive Relation	29.17	16.67	33.33	16.67	29.17	16.67	22.50
LingBot-VA	41.67	**37.50**	37.50	37.50	37.50	45.83	39.17
RWM (RELATRON)	**95.83**	**37.50**	**91.67**	**50.00**	37.50	**70.83**	**57.50**

Pickup with unseen colors. Table 1(b) evaluates pickup under changes in object color and the corresponding referring expression. RWM achieves 95.83% success on seen colors and 57.50% averaged over the five held-out colors. The OOD averages are 39.17% for LingBot-VA, 25.83% for

Text AdaLN, 22.50% for Naive Relation, 20.83% for Object Transformer, and 0.83% for Language Cross-Attention. RWM exceeds the strongest reported OOD baseline by 18.33 percentage points, with the highest success on Cyan, Gray, and White and a tie on Black. Text AdaLN obtains the highest Pink success, at 41.67%.

Figure 4 summarizes the average OOD performance and the variation across individual colors.

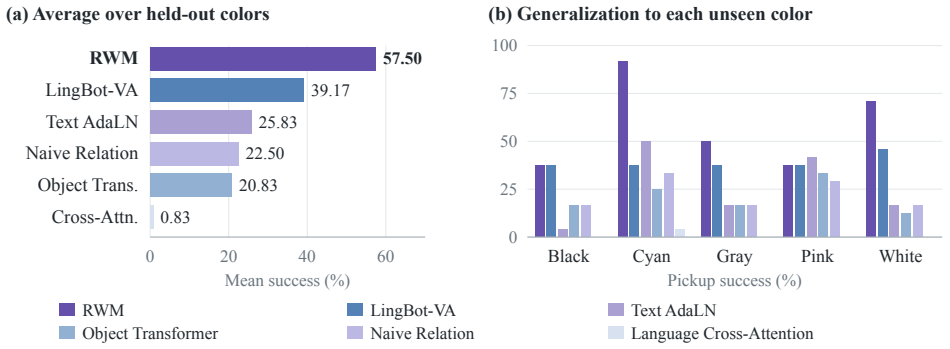

Figure 4: **Zero-shot color generalization.** (a) Average pickup success over five held-out colors. (b) Success on each unseen color. All six methods are shown.

Grasping under target and context shifts. Table 1(a) changes the target appearance, the distractor cylinder, or both. RWM obtains 100.00% on familiar scenes, 87.50% under target shift, 100.00% under context shift, and 85.42% under joint shift. Under joint shift, Object Transformer and Naive Relation obtain 46.88% and 40.63%, respectively. The comparison tests whether the requested object remains identifiable as both its appearance and the surrounding scene change.

Table 2: **LIBERO.** Closed-loop success (%) across four suites. Avg. is the mean over four suites. The lower block holds the policy fixed and compares single-proposal execution with action-consequence (AC) selection. †LingBot-VA results from Li et al. (2026). Gray marks RWM results; bold marks the best result within each block.

Model / variant	Spatial	Object	Goal	Long	Avg.
Object Transformer	91.8	94.4	89.2	68.2	85.9
$\pi_{0.5}$	**98.6**	95.8	95.6	87.0	94.3
LingBot-VA†	98.5	**99.6**	**97.2**	**98.5**	**98.5**
RWM (single proposal)	94.4	94.8	89.2	84.0	90.6
Matched action-selection comparison					
Single proposal	92.8	93.4	80.0	51.4	79.4
AC selection	**96.2**	**96.8**	**89.2**	**85.0**	**91.8**

Manipulation across LIBERO suites. Table 2 reports RWM success of 94.4%, 94.8%, 89.2%, and 84.0% on Spatial, Object, Goal, and Long, respectively, for an average of 90.6%. Object Transformer obtains 85.9% on average.

Real-robot execution. RWM achieves 100% success on Approach and 85% strict success on Pick-and-place. These tasks evaluate physical execution of instruction-conditioned goals.

4.2 Qualitative Analysis of Generated Futures

Changing the instruction changes the desired future. Figure 5 fixes a real-robot observation and varies the instruction between "Approach the hamburger" and "Approach the teapot." The decoded futures depict different arm configurations directed toward the requested referents. This comparison shows how the instruction changes the desired outcome for a shared observed scene.



Figure 5: **Different intentions produce different desired futures.** Left: the current real-robot observation. Middle and right: decoded predicted futures conditioned on “Approach the hamburger” and “Approach the teapot,” respectively. The current observation is shared, while the requested referent changes. The predicted arm configurations differ according to the requested referent.

Table 3: **Bowl Placement.** Future-goal retrieval accuracy (%). ID denotes familiar scenes; Target/Reference changes one participant, Joint changes both, and Wording paraphrases the instruction. Gray marks the full model; bold marks the best result in each column.

Model / variant	ID	Target	Reference	Joint	Wording
Text AdaLN	98.50	60.24	0.00	15.63	8.33
Lang. Cross-Attn.	81.83	48.96	85.42	41.67	79.17
Object Transformer	**99.96**	58.88	**100.00**	55.21	**93.06**
Naive Relation	99.61	47.27	0.00	7.29	8.33
RWM (RELATRON)	**99.96**	**67.60**	**100.00**	**60.42**	90.28

Task-relevant relation selection. Figure 6 visualizes relation selection for two LIBERO instructions. Given a frame from a demo, the left pair shows “put the bowl on the plate,” and the right pair shows “put the wine bottle on the rack.” Each pair combines an object-projected selection heatmap with a scene annotated with relations. The selected interactions correspond to bowl–plate in the left task and bottle–rack in the right task. Green arrows mark selected relations associated with the requested change, red arrows mark selected but not changed relations in the original demo. With the future visualization, these panels connect the instruction, the participating interactions, and the predicted goal; the component ablations below quantify the contributions of selection and message content.

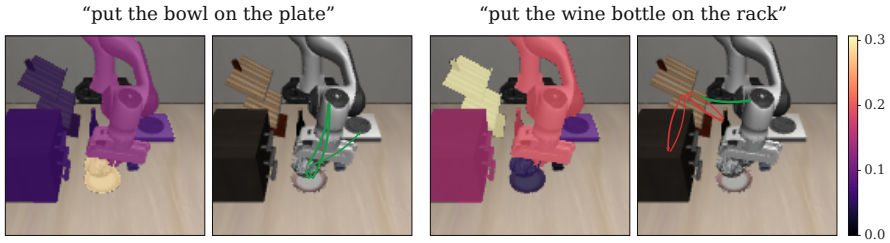

Figure 6: **Instruction-conditioned relation selection.**

4.3 Quantitative Evaluation of Generated Futures

Transferring a relation across participants. Table 3 evaluates Bowl Placement future-goal agreement. Familiar scenes use training-familiar participants and canonical instructions. Target and Reference change the respective participant colors; Joint changes both; Wording paraphrases the instruction. Color changes also change the words identifying the participants, testing transfer across visual appearance and referring expressions. RWM obtains 99.96% agreement on familiar

scenes and 100.00% under reference shift. Under joint participant shift, it achieves 60.42%, compared with 55.21% for Object Transformer and 7.29% for Naive Relation. This yields a 5.21-point advantage over the strongest baseline when both participants change. RWM achieves 67.60% under target shift, compared with 60.24% for Text AdaLN. Object Transformer has the highest wording score, at 93.06%.

4.4 Ablation Study

We examine how relation selection, language-conditioned messages, and future-based action selection contribute to RWM. The component ablations in Figure 7 isolate which interactions receive emphasis and what task information their messages carry.

Language-guided relation selection. Replacing learned selection with uniform pair contributions reduces familiar-scene agreement from 99.96% to 50.08% and reference-transfer agreement from 100.00% to 50.00%. Under joint participant shift, agreement drops from 60.42% to 46.88%. These changes show the contribution of concentrating the relational update on interactions relevant to the instruction.

Language-conditioned message content. Removing language from message content while retaining language-guided selection reduces familiar-scene agreement to 33.58%, joint-shift agreement to 22.92%, and wording agreement to 20.83% (Figure 7). Language thus contributes both to selecting relevant interactions and to specifying the changes those interactions should express.

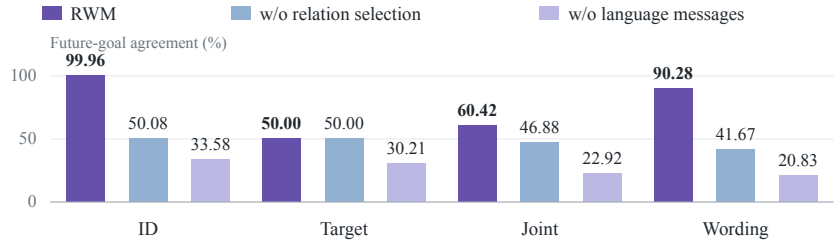

Figure 7: **Contributions of relational conditioning.** Bowl Placement future-goal agreement for RWM and its component ablations. Removing either relation selection or language messages lowers familiar and joint-shift agreement. Under joint shift, the complete model obtains 60.42%, compared with 46.88% without selection and 22.92% without language messages.

Selecting actions against the intended future. We compare intended-goal, uniform, and wrong-goal selection on Bowl Placement. Intended-goal selection scores predicted action consequences against the desired future; wrong-goal selection changes the future used for scoring. Success is 91.66% with the intended goal, 83.33% with uniform selection, and 79.17% with the wrong goal. The comparison supports selecting actions by their agreement with the instruction-conditioned future.

5 Conclusion

RWM connects language-grounded goal construction with action through predicted visual futures. RELATRON organizes this prediction around ordered object interactions, using separate language-conditioned branches to select interactions and determine their messages. Experiments show gains in future-goal agreement under joint participant shifts and in pickup of unseen colors, while component ablations highlight the contributions of both branches. Future-conditioned action generation supports manipulation across LIBERO suites and real-robot approach and pick-and-place tasks, and matched comparisons show that predicted consequences can improve action selection. These findings motivate relational conditioning as a shared interface between instruction interpretation, visual prediction, and control.

AI USE STATEMENT

Large Language Model tools were used to assist with code development, experiment orchestration, analysis, figure preparation, and language editing and revision of the manuscript. All AI-assisted contents were critically reviewed by the authors. The experimental results were obtained from author-conducted experiments, and the final methodological choices, analyses, interpretations, and claims were determined and verified by the authors. The authors take full responsibility for the content of this work.

REFERENCES

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pre-trained image-editing diffusion models. In *International Conference on Learning Representations*, 2024.

Juntao Cheng, Jingkai Wang, Yijun Shen, Xiansheng Chen, and Zhiwei Yu. Beyond instance slots: Semantically rich world models for physical interaction planning, 2026. URL https://arxiv.org/abs/2608.22294.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9156–9172, 2023.

Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B. Tenenbaum, Leslie Kaelbling, Andy Zeng, and Jonathan Tompson. Video language planning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/848784373188ddf641079524e89e0ac9-Abstract-Conference.html.

Nikolaos Gkanatsios, Ayush Jain, Zhou Xian, Yunchu Zhang, Christopher Atkeson, and Katerina Fragkiadaki. Energy-based models are zero-shot planners for compositional scene rearrangement. *arXiv preprint arXiv:2304.14391*, 2023. URL https://arxiv.org/abs/2304.14391.

Ronghang Hu, Anna Rohrbach, Trevor Darrell, and Kate Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. URL https://arxiv.org/abs/1905.04405.

Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation. In *ICLR 2025 Workshop on World Models*, 2025.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1911.12247.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Luyao Zhang, Mingrui Yu, Zelin Gao, Nan Xue, Boyu Zhou, Xing Zhu, Mingyu Ding, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. In *Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.016. URL https://www.roboticsproceedings.org/rss22/p016.html.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. *arXiv preprint arXiv:2511.13720*, 2025. doi: 10.48550/arXiv.2511.13720.

Nan Liu, Shuang Li, Yilun Du, Josh Tenenbaum, and Antonio Torralba. Learning to compose visual relations. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/c3008b2c6f5370b744850a98a95b73ad-Abstract.html.

Weiyu Liu, Yilun Du, Tucker Hermans, Sonia Chernova, and Chris Paxton. StructDiffusion: Language-guided creation of physically-valid structures using unseen objects. In *Robotics: Science and Systems*, 2023. URL https://www.roboticsproceedings.org/rss19/p031.html.

Adam Santoro, David Raposo, David G. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/e6acf4b0f69f6f6e60e9a815938aa1ff-Abstract.html.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.13139.

Zhutian Yang, Jiayuan Mao, Yilun Du, Jiajun Wu, Joshua B. Tenenbaum, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Compositional diffusion-based continuous constraint solvers. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3242–3265, 2023. URL https://proceedings.mlr.press/v229/yang23d.html.

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*, 2024a. URL https://arxiv.org/abs/2411.04983.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. Robodreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 2024b.

A Additional method details

A.1 Object features and ordered pairs

For a view with patch features Z_t and mask weights S_{ip}, object i is represented by

$$o_i = \frac{\sum_p S_{ip} Z_{t,p}}{\max(\sum_p S_{ip}, \epsilon)}. \tag{8}$$

Learned projections produce semantic features for pair construction and visual features h_i for message aggregation. Both use the frozen V-JEPA 2 features. Mask-derived geometry describes object position, extent and relative spatial configuration. Within each branch, an MLP maps the concatenated features of objects i and j and their pair geometry s_{ij} to p_{ij}. Only visible, distinct ordered pairs are retained. The two camera views use the same parameters, but form their pairs independently; their spatial contexts are concatenated for the future generator.

The selector and message branch use the read–write operation in Eq. equation 1 with independent parameters. Padded language tokens and invisible pairs are excluded. The gate budget is $B = 4$ and its temperature is $\tau_g = 0.5$. A single message update is applied; an empty valid-pair set contributes a zero aggregate.

A.2 Equivariance to object re-indexing

Let Π permute object indices and Ω be the induced permutation of valid ordered pairs. Re-indexing features, masks and geometry together gives $X' = \Pi X$, $S' = \Pi S$ and $P' = \Omega P$. Shared pair maps preserve this correspondence. Language features remain fixed.

In Eq. equation 1, pair-wise softmax commutes with permutation, so $A'_r = A_r \Omega^\top$. The language readout satisfies $U' = U$, while the incoming-assignment normalization transforms as $D'_A = \Omega D_A \Omega^\top$. Hence $G' = \Omega G$ in each branch. Shared gate and message maps then give $\boldsymbol{\alpha}' = \Omega \boldsymbol{\alpha}$ and $M' = \Omega M$. The pair-to-node incidence matrix obeys $Q' = \Pi Q \Omega^\top$, yielding

$$Q' \operatorname{Diag}(\boldsymbol{\alpha}') M' = \Pi Q \operatorname{Diag}(\boldsymbol{\alpha}) M. \tag{9}$$

Since $D'_N = \Pi D_N \Pi^\top$ and layer normalization acts independently on each node, Eq. equation 3 gives $\widetilde{X}' = \Pi \widetilde{X}$. Finally, $D'_S = D_S$ and

$$C'_t = D_S^{-1} (\Pi S)^\top \Pi \widetilde{X} W_C = C_t. \tag{10}$$

Thus consistent re-indexing of objects, masks, and geometry preserves the spatial field while permuting the node updates.

A.3 Future objective and action inference

Normalization statistics for visual changes are fitted separately for each view and horizon. In Eq. equation 5, the denominator floor is $\epsilon_\tau = 0.05$. The change-weighted term scales each patch's velocity error by its true feature-change magnitude, normalized within the example and capped at four times its mean; its coefficient is $\lambda_c = 0.5$. For paired instructions, the terminal contrast term uses a hinge margin of 0.05 on change-weighted endpoint error, and its coefficient increases to 0.2. Future sampling uses 20 Euler evaluations.

Each action proposal spans eight controls, and two controls are executed before replanning. Eight independently sampled proposals share the same visual and future conditioning. Action tokens attend within each proposal; proposals do not exchange action-token information. The action and consequence objectives are given in Eqs. equation 6 and equation 7.

B Benchmarks and evaluation protocol

B.1 Reported performance

All reported rates are percentages. Attribute scores average black, cyan, gray, and pink. Bowl pair-transfer success is 7.29%; the black–gray, black–pink, cyan–gray, and cyan–pink pairs obtain 0.0%, 0.0%, 29.17%, and 0.0%, respectively.

LIBERO suite-level performance is reported in Table 2.

Physical baseline. Fine-tuned LingBot-VA obtains 0% aggregate success on the physical tasks.

B.2 Evaluation protocol

Splits and pairing. Instruction Understanding uses familiar instruction structures in new scenes. Grounding Generalization changes one referent's attributes or paraphrases a goal. Joint-attribute transfer changes both referents' attributes under a familiar relation. All appearance and instruction variants derived from one scene remain in the same data partition. Methods are compared on matched initial states and instructions.

Bowl placement. Bowl placement uses red/blue target blocks and a white bowl. Evaluation tests both target slots under left/right/inside, giving six instructions per layout. Target-only transfer uses black, cyan, gray, and pink. Reference-only transfer uses pink and gray bowls. Joint transfer crosses cyan/black targets with pink/gray references. Paired conditions preserve geometry, initial physical states, and non-target appearance. Colors and relations receive equal weight.

Multi-object tasks. The pickup training palette contains red, yellow, brown, navy, and lime targets; placement uses red/yellow blocks and blue/green/purple/orange reference cylinders. A held-out color must be absent from the corresponding task-training inventory. Pickup requires grasping and lifting the instructed target; placement measures the requested target–reference relation. Placement uses target-only, reference-only, and joint-color interventions. Joint transfer uses black/cyan targets and pink/gray references, which are absent from both roles' task-training inventories.

Paraphrases. Three alternative expressions per canonical relation are applied to every eligible target/reference pair. For placement, we replace "Place X to the left/right of Y" with "Move X to the left/right of Y," "Put X on Y's left/right side," and "Position X so it is left/right of Y." For inside, use "Put X into Y," "Move X into Y," and "Position X inside Y." Each full paraphrase is checked against training captions before evaluation. Familiar and unseen target appearances are evaluated separately.

Goal agreement and task success. For each test observation, retrieval candidates comprise reference outcomes of all supported alternative goals and an unchanged scene. Candidates match the test appearances and use the same frozen visual encoder as the predictions. Candidate records supply the target, reference, and relation labels. Joint accuracy requires all three labels; paired accuracy requires both opposite-direction goals to be correct. Retrieval calibration uses the encoded recorded future as the query. The recorded appearance-transfer retrieval evaluations use the same metric. For multi-object retrieval, endpoint calibration accuracy is 90.0% on familiar scenes and 93.33% on transfer scenes. Retrieval errors therefore reflect both prediction quality and the discriminability of the visual representation.

Execution uses the environment's fixed task assessor for every method. Bowl placement allows 240 simulator controls and requires grasp/lift, release within 2 cm of the specified goal, stability, and no disturbance of the other block. Multi-object placement uses the same task-specific assessor across policies. Assessors, time limits, and thresholds are fixed across methods before evaluation. All evaluation attempts enter the denominator.

Aggregation and uncertainty. Colors, relations, and scene families receive equal weight. Results are reported as percentage rates; error bars denote sample standard deviations where shown.

External evaluation. LIBERO uses the suite's success predicates and task horizons. Suite scores average tasks equally. Zero-shot results refer only to tasks excluded from adaptation.

B.3 Additional execution results

Table 4 breaks down Bowl Placement success by target object and spatial relation. Pickup requires grasping and lifting the instructed object; placement additionally requires release at the requested relative location and stability.

Table 4: **Execution success by instruction (%).** Each column is one prompt: "Place the *red/blue* block *to the left of/to the right of/inside* the white bowl."

	Red block			Blue block		
Method	**Left**	**Right**	**Inside**	**Left**	**Right**	**Inside**
VLA-JEPA	62.5	50.0	68.8	68.8	75.0	81.3
Direct Action	43.8	43.8	31.3	50.0	43.8	50.0
$\pi_{0.5}$	0.0	0.0	0.0	50.0	75.0	100.0
Object Transformer	25.0	25.0	25.0	100.0	75.0	100.0
LingBot-VA	0.0	0.0	0.0	0.0	0.0	0.0
RELATRON + AC	100.0	50.0	100.0	100.0	75.0	100.0

C Additional results and diagnostics

C.1 Future readouts and counterfactual instructions

An RGB probe is trained on recorded visual features and images, then frozen for decoding predicted futures. Calibration compares real RGB with the same probe applied to its encoded features. The same decoder maps generated future features to RGB.

Figure 5 shows decoded futures for two instructions applied to the same real-robot observation.

The instruction-following diagnostic separates target-object errors from relation errors (Table 5).

Table 5: **Instruction following in predicted futures.** Retrieval accuracy (%). Paired accuracy requires both opposing instructions to be correct.

Conditioner	**Object**	**Relation**	**Joint**	**Paired**
Text AdaLN	51.3	95.0	51.3	46.3
Object Transformer	84.8	96.8	84.8	79.8
Naive Relation	77.7	96.7	77.7	69.3
RELATRON	72.7	96.3	72.7	63.0

C.2 Data efficiency and appearance transfer

Table 6(a) gives the exact values underlying the learning curves in Figure 8; Table 6(b) reports per-color accuracy of the data-efficiency sweep models shown in that figure. The color mean weights colors equally.

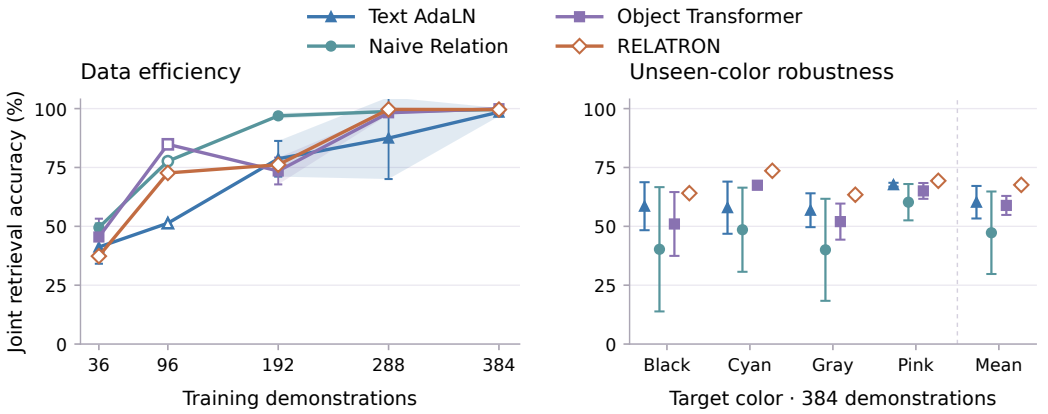

Figure 8: **Data efficiency and color transfer.** Joint retrieval accuracy. Left: training-set size. Right: unseen colors; Mean averages the four colors. Bands and error bars show sample standard deviations. Available RELATRON results are shown as points.

Table 6: **Data efficiency and color transfer.** Joint retrieval accuracy (%) for the data-efficiency sweep models. Entries with error bars report mean ± sample standard deviation.

Condition	Text AdaLN	Object Transformer	Naive Relation	RELATRON
(a) Training-set size (demonstrations)				
36	41.19 ± 7.08	45.47 ± 7.77	49.50 ± 1.73	37.33
96	51.33	84.83	77.67	72.67
192	78.69 ± 7.58	73.50 ± 5.70	96.92 ± 1.23	76.17
288	87.47 ± 17.39	98.22 ± 0.79	98.78 ± 0.59	99.67
384	98.50 ± 1.77	99.96 ± 0.06	99.61 ± 0.54	99.58
(b) Unseen target color				
Black	58.54 ± 10.19	51.00 ± 13.55	40.25 ± 26.39	64.08
Cyan	57.92 ± 11.08	67.46 ± 1.71	48.56 ± 17.87	73.58
Gray	56.83 ± 7.19	52.00 ± 7.66	40.03 ± 21.67	63.42
Pink	67.67 ± 0.82	65.04 ± 3.36	60.25 ± 7.71	69.33
Mean	60.24 ± 6.91	58.88 ± 4.04	47.27 ± 17.52	67.60

C.3 Grasp instruction transfer

The RELATRON color-grasping policy achieves 100.0% seen-color pickup success for both "Lift" and "Grasp ... and lift." Black-target success rates are 75.0% and 50.0%, respectively. The task requires grasping and lifting the instructed object.

C.4 Planning quality and computation

Single-proposal and AC evaluations obtain 58.3% and 87.5% bowl-placement success, respectively, on their evaluation cohorts. In a separate timing-study cohort, the matched eight-proposal comparison shows that uniform and AC selection each obtain 83.3%, while shuffled-goal AC obtains 79.2%. AC succeeds on two cases where uniform selection fails and fails on two where uniform succeeds. The three conditions share initial scenes, action model, sampling noise, and initial candidates. Shuffled-goal AC changes only the instruction used for scoring.

Figure 9 compares closed-loop success with first-decision latency. For the same proposal budget, AC evaluates predicted consequences before selecting a candidate, increasing decision latency relative to uniform selection.

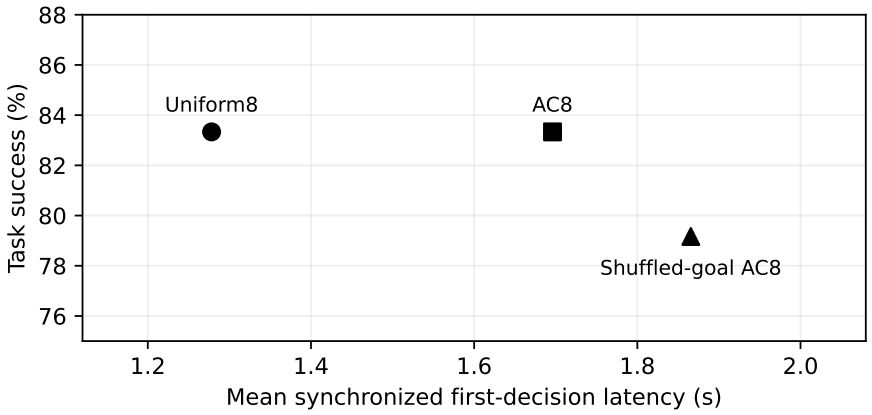

Figure 9: Success and first-decision latency for matched eight-proposal selection on the timing-study cohort. All methods share the actor, scenes, and initial candidate pools.