Communicating Cooperative Affordances: A Shared Physical Effect Space for Semantic Goal Understanding in Cooperative Manipulation

Josse Van de Kerckhove, Victor Tuytte, Pieter Van Molle, Bart Dhoedt, Pieter Simoens
IDLab, Department of Information Technology, Ghent University - imec
Belgium
josse.vandekerckhove@ugent.be

***Abstract*—When one robot arm holds a part while another acts on it, the support arm must understand *what its partner is trying to do*, not merely where the contacts are. We present CoCoA, an ongoing effort to make the goal of a manipulation primitive an explicit, grounded, and reusable semantic representation for cooperative affordance prediction. A short natural language prompt from the acting arm (e.g. "stabilize the desktop while I unscrew the leg") is the *only* channel between the arms: there is no kinematic coupling. CoCoA turns this prompt into a structured, neurosymbolic description of the intended physical effect through heads that predict operation, motion, axis, and expected failure mode, and then grounds that description in a *shared physical effect space* where the language intent, a memory of past interaction attempts, and a latent prediction of the object's motion are aligned by task and by physical consequence. This shared space lets language retrieve goal relevant experience and lets experience ground language. It also gives a concrete mechanism for later transfer to related instructions. We further add metacognitive corrective prompt refinement after observed failures and a language outcome reward that keeps the semantic representation tied to physics. We describe the representation, the three phase training that builds it, and diagnostics probing goal disambiguation, semantic memory retrieval, history use, and alignment structure. This is an ongoing paper. We present the representation and training design and summarize diagnostic behavior of the shared space.**

I. Introduction

Cooperative manipulation makes the difference between acting and *understanding* unusually concrete. Consider two arms working on a piece of furniture: one drives a bolt while the other holds the desktop steady. The holding arm cannot succeed from geometry alone, because the same leg socket must be supported very differently depending on what the other arm is about to do. Screwing and unscrewing apply opposite torques. Pulling a leg out demands the part stay put. Lifting demands it move, parallel and upward. The contact geometry is identical across all four. Only the *goal* differs. To choose a good support, the holding arm must understand what its partner's action *means* for the object, including the disturbance it will create, and act to counter it. This is a goal understanding problem before it is a control problem.

Recent collaborative affordance methods learn point level actionability for multi arm interaction directly from geometry. DualAfford [1] decomposes dual gripper manipulation into connected affordance subtasks, BiAssemble [2] learns collaborative affordances for bimanual geometric assembly, and A3D [3] adds interaction feedback for adaptive support during furniture assembly. These works show that dense 3D affordances are an effective substrate for coordination, but the *intent* behind the interaction stays implicit: it is buried in contact points, object state, or task specific training heads. When tasks share geometry but differ in force direction, motion type, or failure mode, an implicit, geometry only signal struggles to tell them apart.

We argue that the goal should instead be an explicit, *semantic* representation that the support arm reasons over. CoCoA (Communicating Cooperative Affordances) takes the acting arm's intent as a short natural language prompt and treats that prompt as a physical signal rather than a task label. Crucially, language is the *only* channel between the arms. They share no kinematic state (Fig. 1). This makes the representation of the goal, not the mechanical coupling, the thing that has to carry the coordinating information, and it is what makes the setting a clean testbed for semantic goal understanding.

Our central idea is that language intent, interaction experience, and physical outcome are three descriptions of *the same thing*: an effect on an object. "Unscrew the leg," a point cloud recording of a past unscrewing attempt, and the measured SE(3) displacement of the part all encode the same counterclockwise disturbance. CoCoA maps all three into a shared physical effect space, so that semantic structure, rather than raw geometry, organizes the support decision.

This paper focuses on the representation and how it is built. First, we introduce a *neurosymbolic goal decomposition* that grounds a frozen CLIP prompt [4] into structured operation, motion, axis, and failure mode attributes, anchoring free form language to manipulation semantics. Second, we introduce a *shared physical effect space* that aligns this language goal with a semantic memory of past attempts and with a latent world model prediction of object motion, so the goal is grounded in consequences and

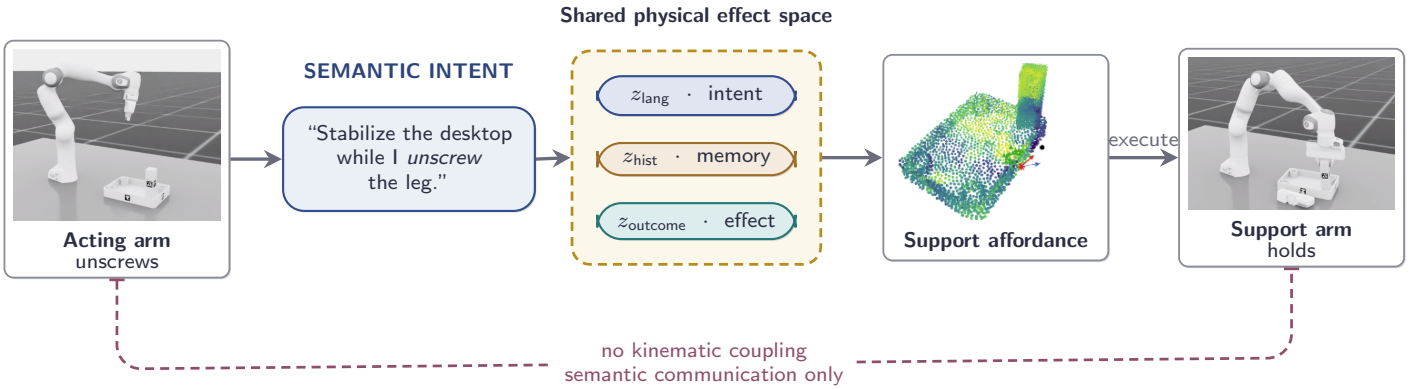

Fig. 1. **Goal understanding through a semantic channel.** The acting arm communicates its goal as language. This is the only link between the arms (no kinematic coupling). CoCoA decomposes the prompt into a structured intent representation and grounds it in a shared physical effect space, where the language goal (z_{lang}), a memory of past attempts (z_{hist}), and a predicted object motion (z_{outcome}) are aligned, from which the support arm reads out where and how to hold the object. *Left: IsaacSim view, right: predicted pointwise support affordance heatmap (warmer is better).*

can retrieve relevant experience. Third, we add two mechanisms that keep the representation honest and adaptive through metacognitive corrective prompt refinement after observed failures and a language outcome reward that ties predicted effects back to the prompt. We describe a three phase training procedure that builds the space and an evaluation protocol probing goal disambiguation, semantic alignment and retrieval, and closed loop history use. As ongoing work, we present the design and the diagnostic behavior of the shared space.

II. Related Work

a) Language as a semantic signal for manipulation: Language conditioned policies use language to select, localize, or generate actions. CLIPort [5] combines a CLIP "what" stream with a spatial "where" stream for pick and place affordances. PerAct [6] and generalist vision language action models such as RT-2 [7] predict full actions from language and vision. RoboCLIP [8] derives reward from video language similarity. These systems read language as a command. CoCoA instead treats a frozen CLIP text embedding [4] as a description of a *physical effect* and asks a narrower, complementary question: what does the goal imply about how a *second*, loosely coupled arm should support the object?

b) 3D affordances and goal grounding: Point level affordance prediction suits furniture style parts, where small local geometry changes alter which contacts help. Where2Act [9] learns actionable points for articulated objects. OpenAD [10] and the language conditioned affordance pose work of Nguyen et al. [11] push 3D affordances toward open vocabulary and language grounded settings. AffordBot [12] predicts structured instruction conditioned affordance elements such as location, motion type, and motion axis. For multi arm coordination, DualAfford [1], BiAssemble [2], and A3D [3] learn collaborative affordances but keep intent implicit. CoCoA builds on this affordance line and on the PointNet++ backbone [13], but replaces implicit intent with an explicit semantic goal representation aligned to experience and outcome.

c) Structured world models and joint embedding prediction: Predicting the latent consequence of an action, rather than reconstructing pixels, underlies joint embedding predictive architectures [14]. CoCoA adopts this view at the level of physical effect: a latent outcome predictor estimates the object's SE(3) motion in the shared space, giving the support critic a small, goal conditioned world model. We instantiate everything on FurnitureBench based parts [15].

III. Semantic Goal Representation

A. Problem: goal understanding over a semantic channel

A primary arm performs a manipulation primitive and communicates its goal as a natural language prompt l. A support arm must choose a stabilization action $a_s = (p_s, d_s)$ with contact point $p_s \in \mathbb{R}^3$ and normalized direction $d_s \in \mathbb{R}^3$, given the current scene point cloud, the prompt l, and an optional buffer of past attempts. Quality is a continuous stability score $y \in [0, 1]$ computed per task to normalize across motion scales (screw/unscrew via SE(3) geodesic distance, pull via peak transient displacement, lift via a bounded height and stability score). The arms share no kinematic state, so *all* cross arm coordination must pass through the semantic representation of l. The technical question is therefore how to represent the goal so that it is (a) grounded in physical consequences, (b) able to retrieve relevant experience, and (c) able to transfer to unseen instructions.

B. Neurosymbolic goal decomposition

A frozen CLIP text backbone [4] maps l to $z_l \in \mathbb{R}^{512}$. CLIP embeddings are semantically rich but not tied to support physics. Inspired by AffordBot's structured language heads for affordance elements such as motion type

and axis [12], we project z_l and decompose it with four symbolic heads that name the physical content of the goal (Table I). Each head emits a softmax distribution over a small discrete vocabulary and indexes a learnable embedding table. The categorical embeddings are fused with the projection to form the language goal embedding $z_{\text{lang}} \in \mathbb{R}^d$ (d=128). This hybrid neurosymbolic step converts a free form sentence into an interpretable, physics anchored goal. For instance, the signed yaw axis class forces the representation to separate screwing from unscrewing even though they share every word but one and all of their geometry.

TABLE I
NEUROSYMBOLIC HEADS DECOMPOSING THE PROMPT INTO PHYSICAL INTENT ATTRIBUTES.

Head	Vocabulary
Operation	push, pull, rotate, lift, insert, screw
Motion	translation, rotation, lift
Axis	$\pm x, \pm y, \pm z$, roll, pitch, yaw_{cw}, yaw_{ccw}
Failure mode	slide, rotate, tip, lift, no motion

C. A shared physical effect space

The conceptual core of CoCoA is a single space with dimension d holding three embeddings that describe the same physical effect from three sources (Table II): the language goal z_{lang}, a memory embedding z_{hist} of past interaction attempts, and a latent outcome z_{outcome} predicting the object's SE(3) displacement. Aligning them makes "what the partner intends," "what happened before," and "what will happen" mutually grounded.

TABLE II
EMBEDDINGS IN THE SHARED PHYSICAL EFFECT SPACE.

Embedding	Source	Encodes
z_{lang}	language encoder	intended effect (the goal)
z_{hist}	history encoder	effects observed in past attempts
z_{outcome}	outcome predictor	predicted object motion

Alignment uses task labels across modalities. Prompts and movement histories from the same physical goal are encouraged to occupy the same region of the space, while other task labels define the rest of the batch. We therefore match language to the goal class expressed by a rollout, not to an arbitrary episode identity. A domain specific subtlety shapes the target: the static desktop geometry is almost identical across the tasks, so aligning language to the full history embedding risks learning object shape rather than physical effect. We therefore align z_{lang} to a *movement only* embedding z_{dyn} (support action and observed part motion, geometry excluded) instead of the full z_{hist}. Empty or low movement histories collapse together across tasks, and some cross task movements alias (a pull attempt and a lift attempt can both move $+z$), so the raw contrastive loss is not a direct performance metric. We judge the representation by cross modal task structure and retrieval in the shared space.

Two further losses keep the space tied to physics. A *latent outcome predictor* maps point features, the candidate action, and the fused language/history context to an embedding of the measured SE(3) displacement, trained end to end against an outcome encoder. This gives a small, goal conditioned world model in the spirit of joint embedding prediction [14]. A *counterfactual* loss requires a wrong task prompt to score lower than the correct one, forcing the critic to actually read the semantic representation rather than ignore it.

D. Semantic memory: goal conditioned retrieval

Past attempts form a memory the support arm can consult, but only the goal relevant entries should matter. Each step is encoded by parallel PointNet++ geometry, action, and movement branches and aggregated by attention *conditioned on the current goal*:

$$\alpha_j = \text{softmax}_j\Big(\frac{\phi_h(f_j)^\top \phi_l(z_{\text{lang}})}{\sqrt{d}}\Big), \qquad z_{\text{hist}} = \sum_j \alpha_j f_j, \tag{1}$$

with padded slots masked. When no interaction evidence is available yet, the memory embedding is neutral and the decision must come from the prompt and scene. As attempts accumulate, the same prompt becomes a query over physical evidence. A "pull" goal can thus attend to past pull related attempts and ignore unrelated screw steps, turning the history buffer into a queryable semantic memory rather than an undifferentiated log. A lightweight bidirectional residual fusion then lets goal and memory revise each other, producing z'_{lang} and z'_{hist} used downstream.

E. Metacognitive refinement after failure

Goal understanding should improve with evidence. After an unsuccessful attempt, the predicted failure mode (slide, rotate, tip, lift, no motion) indicates *how* the object escaped. A corrective prompt module holds a learnable correction embedding per failure category and gates a softmax weighted blend of them into z_{lang}, using the language head's *own predicted* failure distribution so it needs no ground truth label and stays active at inference. This is a small metacognitive loop: the model reflects on the observed failure and edits its own goal representation, a test time adaptation that requires no new instruction from the acting arm. A language outcome reward complements it by projecting the predicted outcome back into CLIP space and rewarding agreement with the prompt, keeping the semantic and physical descriptions of the goal consistent.

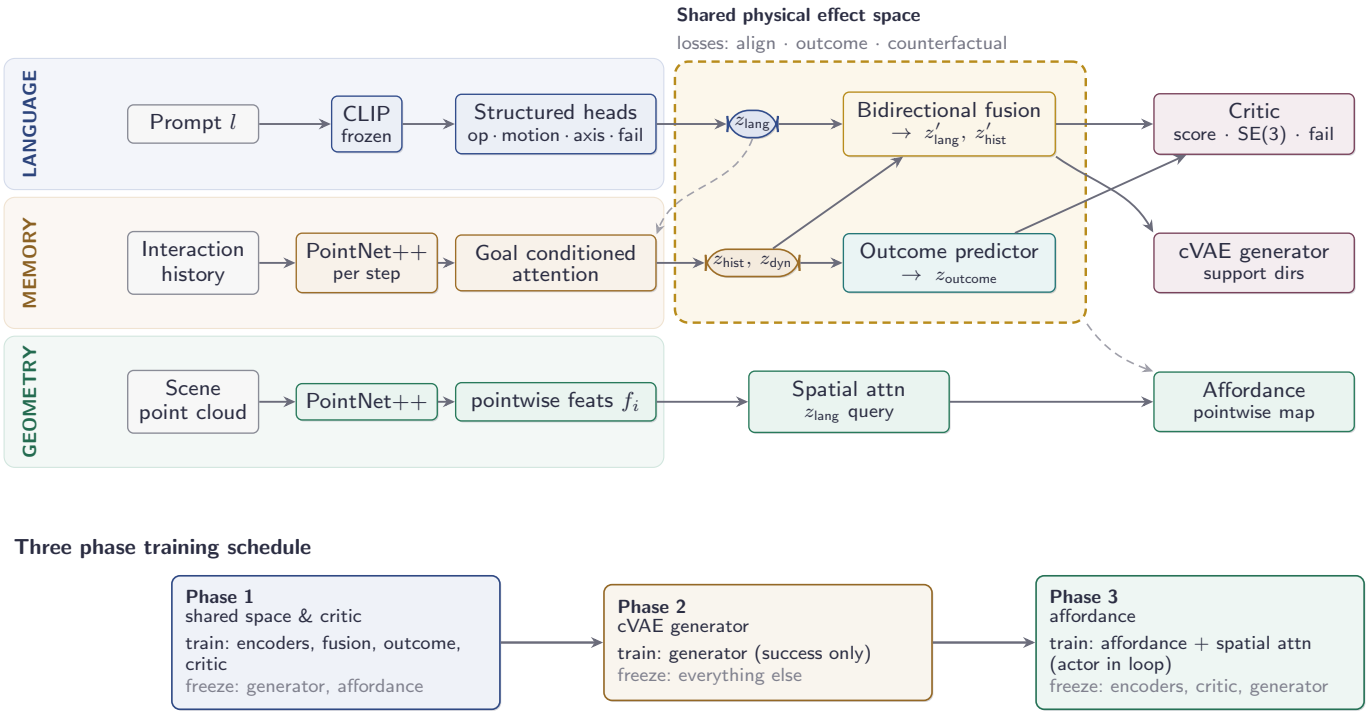

Fig. 2. **CoCoA architecture and three phase training.** Neurosymbolic heads decompose a frozen CLIP prompt into z_{lang}. Goal conditioned attention reads a semantic memory of past attempts into z_{hist} (movement only z_{dyn}). A latent outcome predictor yields z_{outcome}. The three embeddings share a physical effect space (gold) trained with alignment, outcome, counterfactual, and language outcome losses, and feed a critic, cVAE generator, and pointwise affordance head. Bottom: the three phase training schedule.

F. Reading out the support action

Downstream modules consume the grounded goal representation (Fig. 2). A pointwise affordance head scores where to hold, modulated by language guided spatial attention in which z_{lang} queries the pointwise features so that "hold the left leg" upweights the corresponding region. At inference, this affordance map first selects the top K support points. A conditional VAE then samples multiple support directions per point, yielding a small set of point direction candidates. CoCoA does not rank these candidates by affordance alone. For each candidate, the outcome predictor estimates the latent physical consequence, and the critic scores the candidate conditioned on the fused goal, memory, and predicted outcome. When the SE(3) head is available, selection further penalizes excessive predicted object motion, so the chosen support is both semantically relevant and physically conservative. The geometric pipeline mirrors prior affordance work [3, 1]. The novelty is that every module is conditioned on the *semantic* goal rather than on injected contact geometry.

IV. Building the Representation: Three Phase Training

The shared space is built before it is read out (Fig. 2, bottom). **Phase 1** trains the language encoder, the goal conditioned history encoder, the shared point encoder, the fusion module, the outcome predictor, and the critic, under the alignment, outcome, counterfactual, language outcome, and structured head losses. Multi task batches make the alignment objective compare goals that share geometry but differ in physical effect, while a curriculum ramps visible history from zero so the model first learns to act from the goal alone and only then learns to use memory. With the semantic space frozen, **Phase 2** trains the cVAE generator on success only data to propose directions, and **Phase 3** distills the frozen critic into the pointwise affordance head with language guided spatial attention. This uses actor in the loop pseudo ground truth by sampling directions with the generator, scoring them with the critic, and supervising the top K mean. Because the pseudo labels cover only a subset of points each step, the affordance loss is masked to scored points. Unscored points are treated as unknown rather than as negative examples, preventing a uniformly low heatmap. The CLIP backbone stays frozen throughout and prompt embeddings are cached.

These mechanisms make CoCoA more than prompt conditioned affordance prediction. Language is not inserted only at the input or appended only to the critic. It shapes the representation at three different timescales. Before interaction it passes through the structured prompt embedding. During interaction it supports goal conditioned retrieval from memory. After candidate generation it supports outcome aware ranking. The same semantic vector can therefore be inspected as an embedding, visualized as an affordance heatmap, and used inside the final point

direction decision.

This also clarifies the role of the ablations. Removing alignment weakens the link between prompt semantics and observed action effects. Removing outcome prediction removes the small world model used when ranking generated candidates. Removing history leaves the prompt without accumulated physical evidence, while removing the language outcome reward loosens the constraint that predicted consequences remain semantically compatible with the instruction. These mechanisms are coupled in the full model, but each one targets a different failure mode of semantic cooperative support.

V. Evaluation

We evaluate on four simulated desktop manipulation tasks with the same object family and different semantic goals: screw, unscrew, lift, and pull. For each method and task, Table III averages over 50 samples that finish successfully. Each sample contributes the number of interaction history steps required before its final successful support action. In this metric, a history step corresponds to an observed unsuccessful interaction before the final successful support action. In simulation this retry is inexpensive, but on hardware the same event could require recovery or human reset if the part is dropped, wedged, or over rotated. We therefore interpret lower history use not only as faster convergence, but also as fewer risky physical interventions before success. The table includes A3D, A3D+CLIP, full CoCoA, and CoCoA variants without alignment, alignment plus outcome prediction, corrective prompts, history, or the language outcome reward.

TABLE III
History use over 50 successful samples. Each entry is the mean number of interaction history steps before the final successful support action. Lower means fewer history steps before completion.

Model	Screw	Unscrew	Lift	Pull	Mean
A3D [3]	0.80	0.82	0.96	0.28	0.72
A3D+CLIP	0.68	0.72	**0.04**	**0.00**	0.36
CoCoA full	0.34	0.48	0.10	0.04	**0.24**
w/o alignment	**0.14**	0.64	0.24	0.12	0.29
w/o alignment+outcome	0.52	**0.38**	0.10	0.22	0.30
w/o corrective prompts	0.40	0.90	0.20	0.08	0.40
w/o history	0.52	1.50	0.06	0.20	0.57
w/o language reward	0.20	0.82	0.20	0.26	0.37

a) Closed loop behavior: Full CoCoA uses little history overall (0.24 mean steps), substantially less than A3D (0.72) and A3D+CLIP (0.36). It uses little history on lift and pull (0.10 and 0.04 steps on average), while unscrewing remains the hardest semantic distinction for the full model (0.48 steps). A3D+CLIP is strongest on lift and pull, but its higher screw and unscrew values raise its mean. The comparison separates using language as an extra feature from grounding language in physical effect: A3D+CLIP helps on translational tasks, but full CoCoA is more consistent across rotational and translational goals.

b) Ablations: Within this CoCoA ablation set, removing history causes the largest increase in history use (0.57 mean steps), especially for unscrewing (1.50). Removing the language outcome reward raises the mean to 0.37, and removing corrective prompts raises it to 0.40. Removing alignment alone gives a low screw value but requires more history on the other three tasks than full CoCoA. Removing both alignment and outcome prediction improves unscrewing in isolation but still raises the overall mean. These patterns support the intended division of labor: language specifies the goal before interaction, history provides task relevant physical evidence after interaction, alignment keeps language and dynamics in a common space, and the outcome related modules affect how generated support candidates are ranked. The task wise pattern ties the mechanisms back to the tasks. Unscrewing shares geometry with screwing but reverses the torque disturbance. Its degradation without history, alignment/outcome grounding, corrective prompts, or the language reward suggests that torque direction support depends on accumulated evidence and corrective refinement.

c) Qualitative visualizations: Figure 3 shows full CoCoA heatmaps for the four desktop prompts. The selected contacts shift across prompts: screw and unscrew favor different regions of the tabletop around the threaded leg, while lift and pull concentrate support near the loaded leg corner with different selected directions. Figure 4 gives a complementary view of the representation: paired prompt and executed interaction embeddings form task structured regions for all four full CoCoA tasks, including lift. The executed interaction points use the movement only embedding z_{dyn}, so the figure is not simply clustering static object shape. We treat t-SNE as a qualitative visualization rather than a standalone clustering metric. The regions are not fully disjoint, and this overlap is meaningful: screw and unscrew share the same threaded leg geometry and many local stabilizing contacts, while lift and pull both involve upward motion of the leg/tabletop assembly. The shared space therefore organizes samples by physical effect as well as by prompt label: tasks with related mechanics partially overlap, while their paired language and dynamics embeddings remain connected within the same neighborhood. This is the intended behavior for a reusable physical effect representation, because it allows semantically different instructions to share evidence when their contact physics are similar.

VI. Discussion and Conclusion

CoCoA reframes cooperative support as a goal understanding problem: because the arms share only a language channel, the representation of intent must carry the coordinating information. Our design grounds that intent in three ways. It *decomposes* intent neurosymbolically into physical attributes. It *aligns* intent with a semantic memory of experience and a latent world model in a shared physical effect space. It also *refines* intent metacognitively

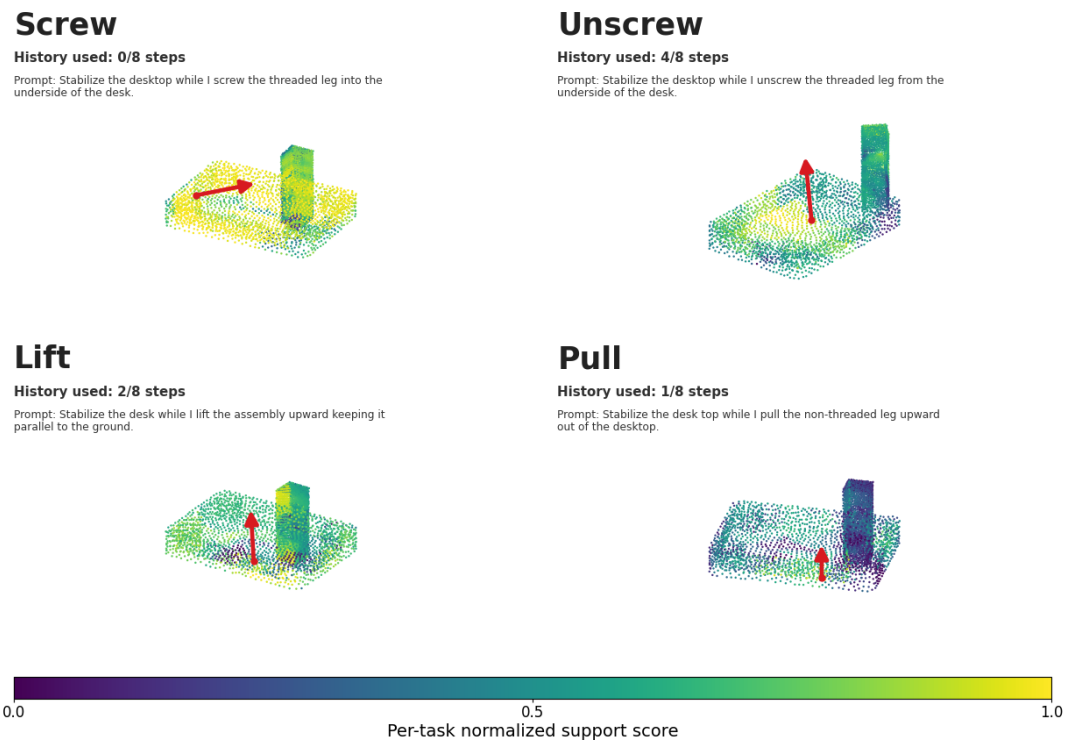

Fig. 3. **Full CoCoA support affordance heatmaps.** Colors are min max normalized within each task, so the highest scoring point in each panel maps to 1.0. Red markers and arrows show the selected support contact and direction. Panel headers show the exact prompt and number of previous interaction history steps used by the decision shown. The same object geometry yields different support choices under screw, unscrew, lift, and pull prompts.

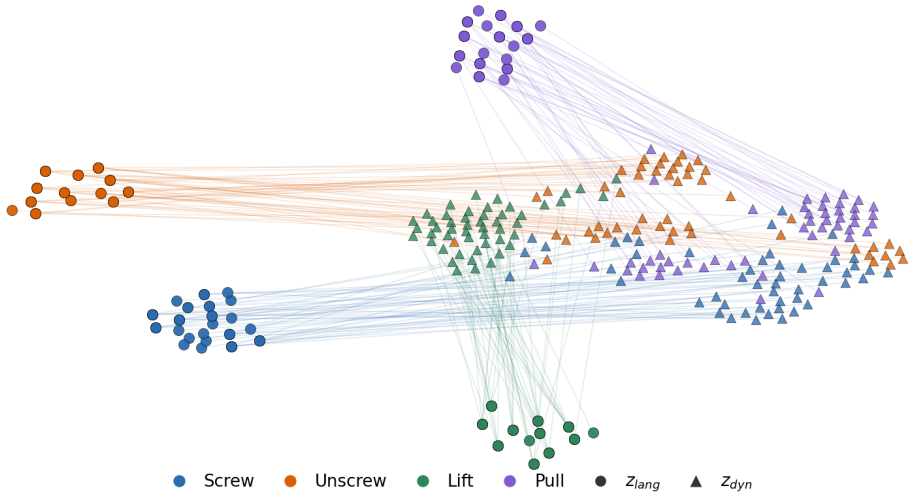

Fig. 4. **t-SNE visualization of the learned shared space.** For each task, we embed 50 paired prompt and executed interaction representations from the full CoCoA model. Circles denote z_{lang}, triangles denote z_{dyn}, and faint lines connect paired samples. This plot is used as a qualitative view of the saved alignment diagnostics, not as a standalone clustering metric.

after failure. A key implementation consequence is that the semantic representation is not only visualized as an affordance map. It is used inside action selection, where generated support candidates are evaluated through a predicted outcome before the final point and direction are chosen. The simulated desktop evaluation provides an early but concrete diagnostic of this claim. Full CoCoA requires the fewest history steps overall under the 50 sample history use metric, whereas the history only A3D baseline needs substantially more interaction before completion. The ablations make the same point from the other side: removing history, alignment/outcome grounding, correc-

tive prompts, or the language reward increases the number of history steps needed before completion. The main limitation is scope. The current results are simulation only, single checkpoint diagnostics on one desktop object family. The hardest prompt is still unscrewing, where opposite torque semantics must be inferred from language but the geometry is nearly identical to screwing. These gaps make the next steps clear: broaden the object families, add richer real robot diagnostics, and test whether the same semantic channel survives real world perception and contact noise.

References

[1] Yan Zhao, Ruihai Wu, Zhehuan Chen, Yourong Zhang, Qingnan Fan, Kaichun Mo, and Hao Dong. DualAfford: Learning collaborative visual affordance for dual-gripper manipulation. In *International Conference on Learning Representations*, 2023.

[2] Yan Shen, Ruihai Wu, Yubin Ke, Xinyuan Song, Zeyi Li, Xiaoqi Li, Hongwei Fan, Haoran Lu, and Hao Dong. BiAssemble: Learning collaborative affordance for bimanual geometric assembly. In *International Conference on Machine Learning*, 2025.

[3] Jiaqi Liang, Yue Chen, Qize Yu, Yan Shen, Haipeng Zhang, Hao Dong, and Ruihai Wu. A3D: Adaptive affordance assembly with dual-arm manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18425–18433, 2026. doi: 10.1609/aaai.v40i22.38907.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[5] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, 2022.

[6] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, 2023.

[7] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, Quan Vuong, Vincent Vanhouche, Huong Tran, Radu Soricut, Anikait Singh, Jaspiar Singh, Pierre Sermanet, Pannag R. Sanketi, Grecia Salazar, Michael S. Ryoo, Krista Reymann, Kanishka Rao, Karl Pertsch, Igor Mordatch, Henryk Michalewski, Yao Lu, Sergey Levine, Lisa Lee, Tsang-Wei Edward Lee, Isabel Leal, Yuheng Kuang, Dmitry Kalashnikov, Ryan Julian, Nikhil J. Joshi, Alex Irpan, Brian Ichter, Jasmine Hsu, Alexander Herzog, Karol Hausman, Keerthana Gopalakrishnan, Chuyuan Fu, Pete Florence, Chelsea Finn, Kumar Avinava Dubey, Danny Driess, Tianli Ding, Krzysztof Marcin Choromanski, Xi Chen, Yevgen Chebotar, Justice Carbajal, Noah Brown, Anthony Brohan, Montserrat Gonzalez Arenas, and Kehang Han. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2165–2183. PMLR, 2023.

[8] Sumedh A. Sontakke, Jesse Zhang, S'ebastien M. R. Arnold, Karl Pertsch, Erdem Biyik, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. RoboCLIP: One demonstration is enough to learn robot policies. In *Advances in Neural Information Processing Systems*, 2023.

[9] Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3d objects. In *International Conference on Computer Vision*, 2021.

[10] Toan Nguyen, Minh Nhat Vu, An Vuong, Dzung Nguyen, Thieu Vo, Ngan Le, and Anh Nguyen. Open-vocabulary affordance detection in 3d point clouds. *arXiv preprint arXiv:2303.02401*, 2023.

[11] Toan Nguyen, Minh Nhat Vu, Baoru Huang, Tuan Van Vo, Vy Truong, Ngan Le, Thieu Vo, Bac Le, and Anh Nguyen. Language-conditioned affordance-pose detection in 3d point clouds. In *IEEE International Conference on Robotics and Automation*, 2024.

[12] Xinyi Wang, Xun Yang, Yanlong Xu, Yuchen Wu, Zhen Li, and Na Zhao. AffordBot: 3D fine-grained embodied reasoning via multimodal large language models. In *Advances in Neural Information Processing Systems*, 2025.

[13] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, 2017.

[14] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[15] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023.