Extended Abstract

Motivation. Language-conditioned robot policies trained by imitation learning often exhibit language neglect: they exploit visual regularities, especially proximity and depth, while ignoring natural language instructions. Such policies may appear competent yet succeed for the wrong reasons, reducing trust in language as a robot's source of control. Standard task-success metrics cannot distinguish a policy that genuinely follows instructions from one that exploits visual shortcuts to the same end; both may achieve high performance under controlled evaluation while failing the moment the visual regularities shift (in a new environment, a different camera angle, or with altered object arrangements). This failure is especially problematic because natural language is appealing for its flexibility: it allows a user to specify intent at deployment time rather than at training time. If a policy learns to ignore language, that promise collapses. We study this in a benchmark where a distractor and a target block are randomly placed on a table, removing depth and proximity as reliable shortcuts and forcing the policy to consult the instruction to succeed.

Method. Three conditions were implemented with a frozen CLIP ViT-B/32 (Radford et al., 2021)

- A (vision-only BC) — baseline
- B (vision + language BC) — diagnose language neglect
- C (SAC fine-tune from B) — test whether RL can repair it

Condition C v2 adds compliance rewards (CLIP: $+2$ target lift, -2 distractor lift, Objpose: $+10$ target lift, -10 distractor lift), lift-aware shaping (reach, gated descent, grasp bonus, target-z lift), 50/50 oracle demo replay, BC-regularized SAC (λ_{bc}=20 with Q-normalization), and full BC warm-start. These four fixes were used to fix some of the issues of v1 and allowed for greater task alignment and lift success.

Implementation. The Franka Panda arm was implemented in robosuite (Zhu et al., 2020). Using 200 oracle demos with multi-phase reach-over when the target is far, the behavior cloning approach of Condition A and Condition B was trainged with 80 epochs (90/10 val split). Condition C using SAC was trained with 500k env steps (horizon 300, checkpoint every 50k) on a local GPU. Each evaluation runs 50 episodes with randomized layout and in-distribution colors. The metrics used are defined as follows: task alignment (correct block within 12 cm or lifted, no distractor lift), success (target lifted >4 cm), language compliance (B/C, behavioral), and distractor-lift rate as a shortcut sanity check.

Results. With CLIP-only, we saw a task alignment of 46% in Condition A, 52% in Condition B, and 64% in Condition C. Only C successfully lifts the block (10% success vs. 0% for A/B). Language compliance tracks tasks alignment (52%, 64%), and distractor lift at 0% confirms that implementation A<B<C and C>B without proximity shortcuts. Proprio (+3-d gripper position) helps the BC condtions (A 56%, B 64%) but collapses C (12% success, 30% alignment). Object pose (+9-d, ordered near-to-far) achieves a 30% success and 74% task alignment at the 200k SAC checkpoint but regresses to 6%/36% at 500k, motivating early stopping.

Discussion. BC reaches the correct block roughly half the time but never lifts under CLIP-only sensing. SAC supplies descent, grasp, and lift gradients that demonstrations alone omit. Proprio helps imitation but destabilizes the language–vision input under RL. Object pose improves lift but requires checkpoint selection (task alignment peaks 86% at 150k steps and success peaks at 200k steps).

Conclusion. Compliance-shaped SAC repairs language neglect beyond BC. The improvement from Conditions A to B to C diagnoses when BC ignores language and when RL repair helps. Observation design and checkpoint selection remain primary concerns for sparse-reward manipulation with frozen vision encoders. Future work should add causal language tests, vary the use of language (e.g. pick up the block nearest to the edge), and compound multiple language instructions together (including color of block, placement of block, orientation of arm, and environment details). Additionally, a focus on how vision and language can help the robot gripper lift up the block instead of only identifying the correct block would be a fascinating experiment.

Forced Grounding: Diagnosing and Repairing Language Neglect in Imitation-Learned Robot Policies via RL

Elana Chen
Department of Computer Science
Stanford University
elanac25@stanford.edu

Hayden Kwan
Department of Computer Science
Stanford University
haykwan@stanford.edu

William Rose
Department of Computer Science
Stanford University
wrose03@stanford.edu

Abstract

Language-conditioned imitation learners often ignore instructions and rely on visual shortcuts. We diagnose this *language neglect* with a controlled A/B/C comparison on a distractor/target block-lifting task in robosuite, then repair it via SAC fine-tuning with distractor penalties and lift-aware shaping. On frozen CLIP observations (N=50 episodes), alignment improves from 46% (vision-only BC) to 52% (language BC) to 64% (BC-regularized SAC), and only SAC lifts the target (10% success vs. 0%). Distractor-lift rates remain near zero, supporting honest grounding. With changes to the SAC approach, Condition C v2 achieved a 30% task success rate and 74% alignment. We contribute: (1) a randomized distractor/target diagnostic for language neglect; (2) a SAC v2 pipeline combining demo-mixed replay, full BC warm-start, and compliance rewards; and (3) behavioral metrics separating alignment from lift success.

1 Introduction

Robots instructed in natural language should select actions consistent with those instructions. In practice, policies trained by behavior cloning (BC) on paired image–language–action data often *ignore* language and exploit simpler visual cues, a failure mode we call *language neglect*. This is especially problematic when depth or proximity correlates with the correct action. A policy that always reaches for the nearest object may appear competent while never using the instruction. As language interfaces become standard for general-purpose robots, diagnosing and repairing such neglect is a prerequisite for trustworthy deployment.

Prior work maps instructions to rewards rather than policies directly (Fu et al., 2019; Bahdanau et al., 2018), and CLIP-based rewards have been applied to manipulation (Mahmoudieh et al., 2022). However, less is understood about whether explicit compliance penalties during RL fine-tuning can repair grounding failures inherited from BC, especially when perception is frozen and high-dimensional. We address this gap with

Table 1: Experimental conditions (CLIP-only primary run).

Cond.	Observation	Training	Role
A	512-d z_{vis}	BC	Vision-only baseline
B	1024-d $[z_{\text{vis}}.z_{\text{lang}}]$	BC	Language-conditioned BC
C	1024-d (same as B)	SAC v2 from B	RL repair + compliance

a controlled experimental setup" vision-only BC, language-conditioned BC, and SAC fine-tuning from the language BC checkpoint, all evaluated on identical rollouts with randomized spatial layout.

We design a distractor/target benchmark to make visual shortcuts fail. A Franka Panda arm must pick up a target block identified by an instruction (e.g., "pick up the red block") while a distractor sits closer to the robot along the table's depth axis. Each episode randomizes whether the target occupies the near or far slot (50/50), so a fixed depth heuristic cannot succeed across episodes. When the target is far, the distractor physically blocks a straight-line reach, forcing a reach-over trajectory. **Condition A** (vision-only BC) provides a benchamrk; **Condition B** (vision + language BC) tests whether demonstrations alone induce grounding; **Condition C** (SAC fine-tuned from B) tests whether RL with explicit compliance rewards can *force* grounding and enable lifting.

We investigate two questions: (1) Does a compliance-shaped RL reward improve grounding over BC alone? (2) Does Condition C outperform Condition B on task success and language compliance? We answer (1) and (2) affirmatively on CLIP-only observations.

Table 1 summarizes the experimental ladder. All policies share the same MLP architecture and training demos. They differ in observation space and learning objective. Evaluating all three under identical randomized layouts ensures that measured differences reflect conditioning and training rather than distribution shift.

Our CLIP-only experiments confirm A $<$ B $<$ C on alignment and C $>$ B on success. Extended observation ablations (proprioception, object pose) are reported in Section 5.3. Code, configs, and evaluation scripts are organized as a reproducible pipeline: environment verification, demo collection, BC training, SAC v2 fine-tuning, and rollout-based comparison of all three conditions.

2 Related Work

RL fine-tuning with explicit repairs grounding failures inherent in imitation learning is supported by Fu et al. (2019). Instead of grounding natural language commands as direct policies, we have them as reward functions. Leveraging inverse RL to convert commands into rewards can help the agent to plan and understand to help generalization. This theoretically supports our approach that RL fine-tuning will outperform the baseline BC models and connects the fact that policy-learning approach to instruction following fals to generalize because it relies on zero-shot transfer aligns with the investigation of language neglect in BC policies.

Bahdanau et al. (2018) also explores this framework for mapping language to rewards to handle underspecified instructions. They have instruction-conditional reward models trained from expert examples of completed tasks to avoid manual rewarding for each task. However, this is limited to symbolic grid-world environments and formal languages leaving a gap for real-world visuomotor control which we will delve into. Moving this into high-dimensional space for continuous robot manipulation and instead of expert exmaples to train a reward model, using direct -1 penalty for distractor interactions, creates a similar environment for the agent to learn.

Mahmoudieh et al. (2022) explores using the CLIP embedding dot product as a task reward signal. They realized that vanilla dot products between CLIP language and image features fails to distinguish spatial relationships often. This issue is solved when they factor "what" and "where" by spatially grounding noun

phrases. While CLIP is frequently used for high-level goal representation, there is a gap in understanding how RL fine-tuning specifically corrects for the language-neglect bias often found in frozen CLIP-based imitation learners. Our project will directly look into reach-over tasks with a distractor, testing the paper's identified "spatial neglect". CLIP was argued to be suboptimal for robotics by Nguyen et al. (2024) because it is trained on static image-text pairs, so utilizing Centered Kernel Alignment (CKA) to track embedding changes, showing if the RL phase is fine-tuning the perception-to-action mapping that CLIP doesn't possess.

More recently, the synergy between Large Language Models (LLMs) and RL has been used to unlock offline learning for generalizable policies Pouplin et al. (2024), while residual learning frameworks provide efficient methods for finetuning behavior cloning policies Ankile et al. (2025). Further efficiency in RL finetuning has been achieved through posterior behavioral cloning Wagenmaker et al. (2025). This prompts our strategy of using SAC.

3 Method

3.1 Task and Environment

We simulate a Franka Panda with operational-space pose control in robosuite (Zhu et al., 2020) at 20 Hz. Two 3 cm colored cubes (red, green, blue, yellow) rest on a table at $z{=}0.8$ m. One cube is the target, one the distractor, and colors are randomized unless fixed for evaluation splits. The natural-language instruction is "`pick up the {target_color} block`", set at episode reset.

Spatial layout uses two depths, a region ($y \in [-0.15, -0.10]$) and a far region ($y \in [-0.35, -0.30]$). With probability 0.5, near/far assignments are swapped so the target is not always behind the distractor. When the target is far, the distractor blocks a direct reach and the oracle executes a multi-phase reach-over (approach distractor, clear above it, approach target, grasp, lift). When the target is near, the oracle reaches directly. Task success requires lifting the target center more than 4 cm above the table surface.

3.2 Observations and Policy Architecture

Perception uses a frozen CLIP ViT-B/32 encoder (Radford et al., 2021). The `agentview` RGB camera ($256{\times}256$, preprocessed to $224{\times}224$) yields $z_{\text{vis}} \in \mathbb{R}^{512}$. The instruction yields $z_{\text{lang}} \in \mathbb{R}^{512}$. Condition A observes $o_A = z_{\text{vis}}$. Conditions B and C observe $o_{B,C} = [z_{\text{vis}}; z_{\text{lang}}] \in \mathbb{R}^{1024}$.

A shared MLP policy maps observations to actions: $512 \to 256 \to 7$ with ReLU activations, outputting six end-effector pose deltas plus one gripper command. Behavior cloning minimizes Huber loss ($\delta{=}1$) with Adam (lr$=10^{-4}$, batch 256) for 80 epochs on 200 oracle demonstrations with a 90/10 train/validation split, with the best validation checkpoint retained for each condition.

3.3 Expert Data and Oracle

Two hundred successful oracle episodes are collected with randomized layout. Each episode stores RGB frames, actions, the instruction string, a success flag, and `target_is_far` boolean. The oracle is a scripted finite-state controller with position gains tuned for reliable grasping. It provides high-quality but narrow coverage of the state space, motivating RL fine-tuning for lift completion under the compliance reward.

Color splits. Training and evaluation use in-distribution color pairs sampled from {red, green, blue, yellow}. At evaluation time, target and distractor colors are randomized independently subject to being distinct.

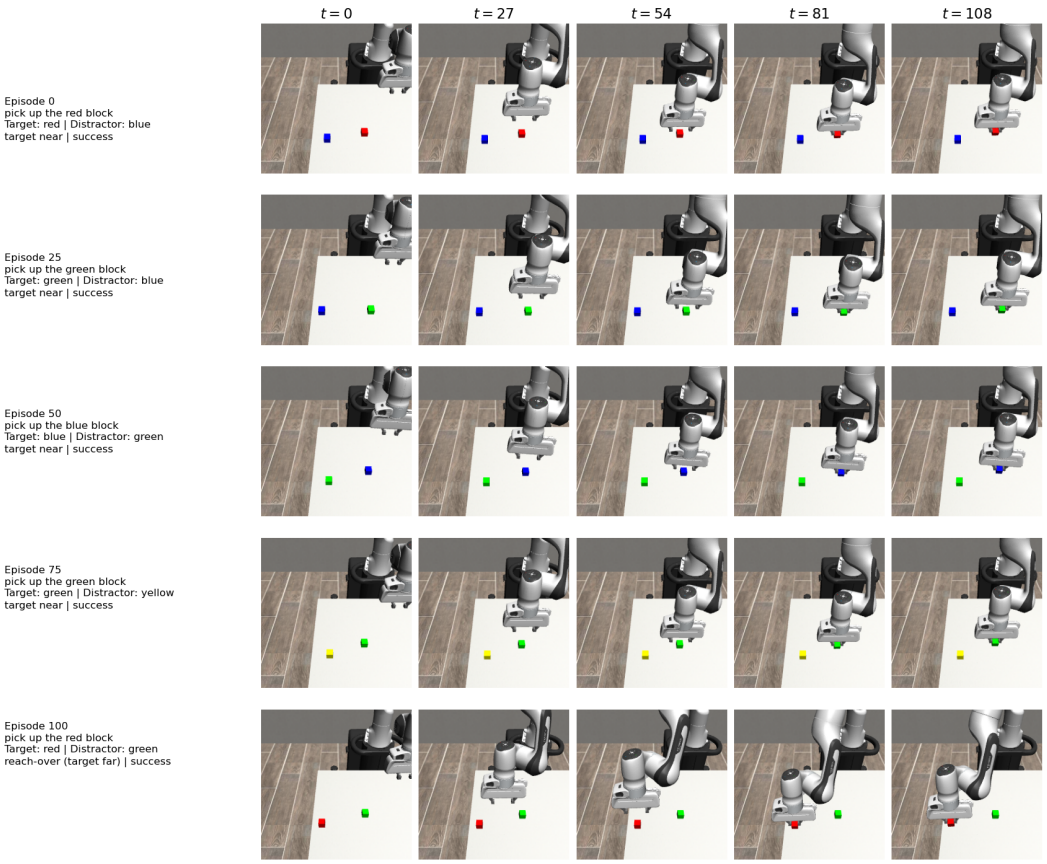

Figure 1: Reach-over task. The robot must identify the instructed block (here, red) and reach past the nearer distractor when the target occupies the far slot.

3.4 Condition C: From SAC v1 Failure to SAC v2

Our first SAC implementation (v1) initialized only part of the actor from BC, used a per-step penalty for gripper proximity to the distractor, and relied on sparse ± 1 terminal rewards and XY distance to distractor/target blocks. After 500k steps, v1 policies rarely hovered near the correct block but almost never grasped or lifted (0% eval success). Three failure modes were found: (i) the distractor proximity penalty accumulated over a 1000-step horizon and dominated the return, encouraging equidistant hovering, (ii) XY-only shaping provided no gradient toward the lift event, and (iii) no critic initialization caused poor reward feedback.

Condition C v2 addresses each issue:

1. **Full BC warm-start**: actor mean, log-standard-deviation, and critic observation trunk initialized from Condition B.
2. **BC-regularized SAC**: the actor loss adds $\lambda_{\text{bc}} \| \tanh(\mu(s)) - a_{\text{demo}} \|^2$ with λ_{bc}=20, plus Q-normalization so the BC term is not overwhelmed by critic magnitude.

3. **Demo-mixed replay**: each gradient step draws 50% of the minibatch from a frozen oracle replay buffer (100 successful episodes collected under the v2 reward wrapper).
4. **Lift-aware reward shaping** (no distractor proximity penalty): per-step terms include one-sided XY reach toward the target, gripper-z descent gated when within 6 cm XY of the target, one-sided target-z lift shaping above 2 cm, a one-time grasp bonus when closing within 4 cm, terminal ± 2 for correct/incorrect lift, and a small time penalty (0.001/step).

Training uses 5000 steps of supervised actor pretraining on oracle demos, 1000 actor-warmup gradient steps (critic-only updates), then 500k online steps with horizon 300, fixed entropy coefficient 0.05, and checkpoints every 50k.

Per-step reward combines reach, descent, lift, and grasp terms:

$$r_t = r_{\text{reach}} + r_{\text{descent}} + r_{\text{lift}} + r_{\text{grasp}} - 0.001 + r_{\text{terminal}}, \tag{1}$$

where r_{reach} is one-sided XY progress toward the target, r_{descent} rewards lowering the gripper when within 6 cm XY of the target, r_{lift} rewards target height increases above 2 cm, r_{grasp} is a one-time bonus for closing near the target, and $r_{\text{terminal}} \in \{+2, -2, 0\}$ fires on correct lift, distractor lift, or neither. Unlike v1, no per-step penalty fires for approaching the distractor.

3.5 Evaluation Metrics

All metrics are computed per episode and averaged over N rollouts:

- **Task success**: the target block is lifted above the environment threshold (> 4 cm).
- **Task alignment**: the distractor is *not* lifted, the minimum gripper–target XY distance is less than the gripper–distractor distance, and either the target is lifted or the gripper comes within 12 cm of the target.
- **Language compliance** (B/C only): identical behavioral test to task alignment. The policy received language input, but this does *not* establish that the language embedding causally determined the action.
- **Distractor lift rate**: fraction of episodes where the distractor is lifted. This was a check that high alignment is not achieved by repeatedly grabbing the nearer block.

4 Experimental Setup

Training data and BC. We collect 200 successful oracle episodes with `randomize_layout=True`. BC is trained jointly for Conditions A and B from the same demo pool, differing only in observation inputs. Checkpoints `cond_a_bc.pt` and `cond_b_bc.pt` are selected by validation Huber loss.

SAC fine-tuning. Condition C v2 loads `cond_b_bc.pt`, builds or loads a cached demo replay buffer, and trains for 500k environment steps with batch size 256, replay buffer size 200k, discount γ=0.99, Polyak rate τ=0.005, and learning rate 3×10^{-4}. Training and evaluation uses local GPU for simulation and rendering CLIP encodings.

Evaluation protocol. Each condition is evaluated for 50 episodes with randomized layout and in-distribution colors. We report mean rates for task success, task alignment, language compliance (B/C), and distractor lift. Reported CLIP-only numbers come from the final v2 run documented. Observation ablations (proprio +3-d gripper position and object pose +9-d near/far block and gripper positions ordered by depth) reuse the same evaluation framework.

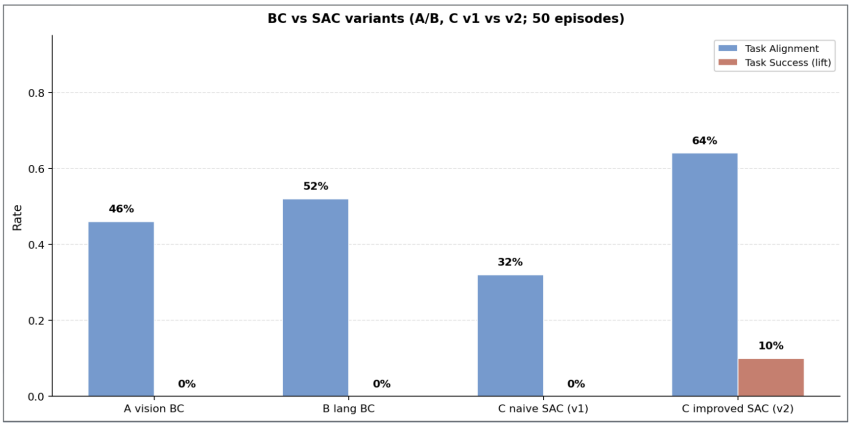

Figure 2: CLIP-only Conditions A/B/C (N=50). SAC v2 improves alignment, language compliance, and lift success over BC baselines.

5 Results

5.1 SAC v1 vs. v2

Before presenting the final A/B/C numbers, we summarize why v2 replaced v1. After 500k v1 steps, Condition C reached roughly 32% alignment and 0% success—hovering near the correct block without grasping. Root causes included a cumulative distractor proximity penalty (-0.02 per step when closer to the distractor than the target), telescoping XY shaping without lift gradient, and partial BC initialization. V2 removes the proximity penalty, adds lift/descent/grasp shaping, fully warm-starts from BC, and mixes oracle demos into every batch. The v2 run reported below achieves 64% alignment and 10% success under identical evaluation.

5.2 Main Result: CLIP-Only A/B/C

Figure 2 summarizes the primary experiment. Task alignment increases monotonically from A to B to C: 46% $\rightarrow$ 52% $\rightarrow$ 64%. Only Condition C achieves non-zero lift success (10%). Language compliance equals alignment for B (52%) and C (74%). Distractor-lift rate is 0% for all three conditions.

These results support our core hypotheses. **Language helps BC:** Condition B exceeds A on alignment (52% vs. 46%), confirming that language input improves target identification when layout is randomized. **RL repairs beyond BC:** Condition C improves alignment to 64% and achieves the only non-zero success rate (10%).

The zero distractor-lift rate across conditions is important. Improved alignment is not explained by repeatedly lifting the nearer block. Qualitatively, BC policies often reach to within a few centimeters of the correct block, sometimes hovering with an open gripper, but fail to close and lift. SAC shaping rewards descent and grasp closure, converting alignment into occasional successful lifts.

Failure modes. Rollout visualization reveals two recurring BC failure modes under random layout: (i) *hover without grasp*—the gripper centers over the correct block but grips above the block, and (ii) *far-slot under-reach*—the arm stops short when the target is in the far slot, consistent with Condition A's lower alignment. Condition C v2 reduces hover failures by rewarding gripper descent and closure.

Table 2: Full observation ablation (N=50, random layout).

Condition	Obs.	Success	Align.	Lang.	Distr.
A	CLIP	0%	46%	—	0%
B	CLIP	0%	52%	52%	0%
C	CLIP	10%	64%	64%	0%
A	CLIP+proprio	0%	56%	—	0%
B	CLIP+proprio	0%	64%	64%	0%
C	CLIP+proprio	12%	30%	30%	0%
A	CLIP+objpose	0%	50%	—	0%
B	CLIP+objpose	0%	66%	66%	0%
C	objpose @200k	30%	74%	74%	0%
C	objpose @500k	6%	36%	36%	2%

5.3 Observation Ablations

Beyond the CLIP-only primary run, we evaluate two observation extensions under the same 50-episode protocol: *proprioception* (+3-d table-relative gripper position) and *object pose* (+9-d gripper and block positions ordered near-to-far in depth, without leaking target identity). Table 2 reports all conditions. Object-pose SAC uses a lift-dominant reward with λ_{bc}=20.

Proprioception (+3-d). Appending table-relative gripper (x, y, z) raises BC alignment (A: 56%, B: 64%) but *degrades* Condition C to 12% success and 30% alignment/compliance (Figure 3). Extra proprioceptive state helps imitation of reach trajectories while interfering with language–vision fusion under online RL optimization. Despite achieving a higher raw success among C variants (12%), grounding collapses relative to CLIP-only C (64% alignment). This is believed to be due to reward hacking and over reliance on the gripper information and rewards.

Object pose (+9-d). Adding gripper and block positions (ordered near-to-far in depth, table-relative) yields the best overall numbers: 30% success and 74% alignment/compliance at the **200k-step** SAC checkpoint—roughly $3\times$ the lift rate of CLIP-only C. The 500k final checkpoint regresses to 6% success, 36% alignment, and 2% distractor lift. We evaluated every 50k checkpoint (Figure 4). Success peaks at 200k (30%)) and alignment peaks at 150k (86%). We report the 200k checkpoint as the best success–alignment trade-off. Distractor-lift rate remains at 0 at the best checkpoint, indicating improved lift reflects metric control rather than proximity shortcuts.

5.4 Qualitative Rollout Analysis

Rollout visualization reveals differences across conditions and SAC versions as well. Under Conditions A and B, the gripper consistently approaches the correct block but the arm hovers at mid-height with a closed gripper. Condition C v2 with proprioception exhibits a distinctly different failure mode: the gripper trajectory is erratic and jittery, with visible lag and no clear goal-directed motion toward either block. Condition C v2 with object positions resolves this: gripper motion is visibly smoother and more deliberate, with a clear descent phase toward the target block and gripper closure on successful episodes. Distractor contact remains absent across all conditions in rollout observation, consistent with the zero distractor-lift rates reported quantitatively.

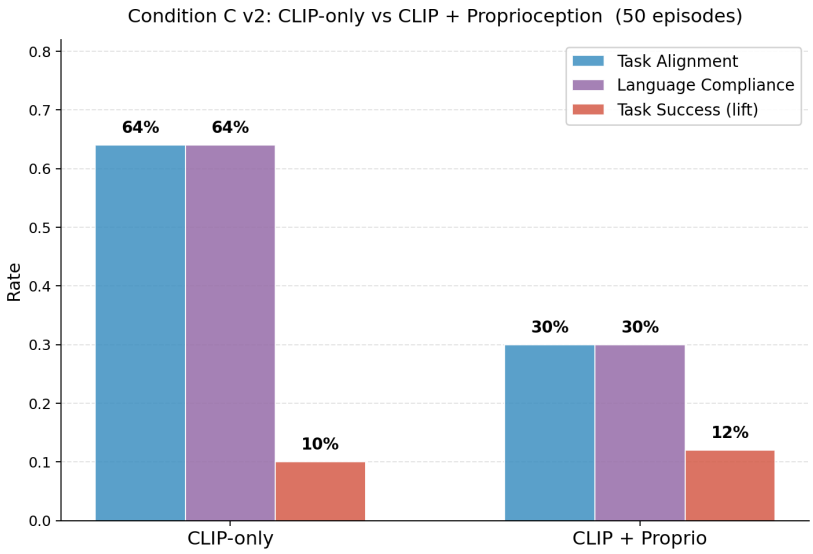

Figure 3: Condition C v2: CLIP-only vs. CLIP+proprio (N=50). Proprioception collapses alignment under RL despite higher BC alignment.

6 Discussion

Answering our research questions. **Q1 (Does RL force language use?):** Under our behavioral metrics, yes—Condition C exceeds B on alignment and compliance while distractor-lift remains zero, indicating the policy reaches the instructed block rather than the nearer distractor. We are not able to claim causal use of the language embedding without counterfactual tests. **Q2 (Does C outperform BC?):** Yes on both alignment (+12 points over B) and success (+10 points).

Diagnosing language neglect. The randomized reach-over layout breaks fixed depth shortcuts. Condition A $<$ Condition B on alignment demonstrates that language input is used at least correlatively during BC. Condition C $>$ Condition B on both alignment and success shows that compliance-shaped RL adds signal beyond demonstrations alone.

Alignment vs. lift. BC policies under CLIP-only observations achieve moderate alignment but 0% success. Frozen CLIP encodes semantic "which block" information but not metric grasp geometry which is consistent with critiques of CLIP for manipulation (Nguyen et al., 2024). SAC v2's lift-aware shaping and demo replay supply gradients for the final centimeters of the skill.

Observation ablations. Proprioception helps BC but hurts RL grounding, suggesting that extra low-dimensional state can shift optimization toward reach geometry at the expense of instruction following. Object pose features improve lift substantially but require early stopping. Continuing SAC training past 200k steps degrades policy quality, a common pathology in sparse-reward fine-tuning when critics overfit to off-policy data.

Limitations. Our language compliance metric is behavioral, not causal. Shuffle-language or instruction-swapping tests would strengthen claims. We did not report novel-color transfer in this paper. CKA analysis

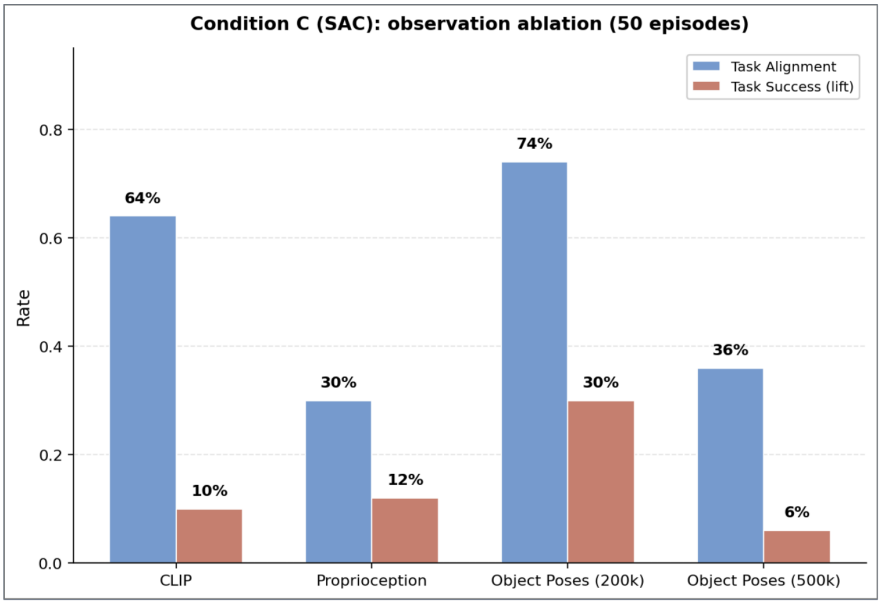

Figure 4: Object-pose Condition C success vs. SAC training steps (N=50 per checkpoint). Peak at 200k steps. Final checkpoint performs worst.

of vision–language representation fusion across SAC checkpoints (planned in our milestone) was deferred after the v2 architecture change. Sim-to-real transfer is not addressed. Success rates remain modest (10% CLIP-only, 30% with object pose at the best checkpoint), reflecting the difficulty of sparse-reward grasping from frozen CLIP features.

Practical recommendations. For practitioners facing language neglect in BC policies, we recommend: (i) randomize spatial layouts at train and test time to break depth shortcuts; (ii) pair language-conditioned BC with compliance penalties rather than relying on demonstrations alone; (iii) use demo-mixed replay and BC-regularized SAC when lift events are rare; (iv) validate checkpoint quality throughout training rather than assuming the final weights are best; and (v) augment frozen vision with metric pose features when lift success stalls.

7 Conclusion

We presented a controlled reach-over benchmark and A/B/C diagnostic for language neglect in CLIP-based imitation learners. Compliance-shaped SAC fine-tuning repairs grounding and enables lifting where BC fails, with distractor-lift rates near zero supporting honest instruction following. Observation ablations show that proprioception can help BC yet hurt RL grounding, while object pose enables higher success at the cost of careful checkpoint selection. Beyond block identification, our results suggest that vision and language must jointly inform the full manipulation sequence, from approaching the object, how the gripper descends, closes, to lifting. We also shows that BC alone cannot supply the gradients needed for this final stage. Closing this gap required explicit lift-aware reward shaping, demo-mixed replay, and BC regularization during RL fine-tuning, pointing to a broader principle: imitation and reinforcement learning play complementary roles, with BC establishing language-grounded reaching and RL completing the physical skill. Future work should add

causal language interventions, evaluate novel attribute combinations, and analyze multimodal representation changes during RL repair, and investigate whether richer language instructions can further sharpen grounding and generalization..

8 Team Contributions

- **Elana Chen:** Designed initial project direction, augmented oracle and setup, implemented and trained updated condition A and B, contributed to analysis of results and presentation.
- **Hayden Kwan:** Environment setup in Robosuite, implementation of conditions A and B. Developed script to create oracle demos to train BC and fill SAC buffer. Designed initial version of Condition C (SAC).
- **William Rose:** Implementation of condition C and condition C v2. Training / evaluation / tuning of SAC. Trained on local GPU and did ablation testing.

References

Lars Ankile, Zhenyu Jiang, Rocky Duan, Guanya Shi, Pieter Abbeel, and Anusha Nagabandi. 2025. Residual Off-Policy RL for Finetuning Behavior Cloning Policies. *arXiv* (2025). doi:10.48550/arxiv.2509.19301

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. 2018. Learning to Understand Goal Specifications by Modelling Reward. *arXiv* (2018). doi:10.48550/arxiv.1806.01946

Justin Fu, Anoop Korattikara, Sergey Levine, and Sergio Guadarrama. 2019. From Language to Goals: Inverse Reinforcement Learning for Vision-Based Instruction Following. *arXiv* (2019). doi:10.48550/arxiv.1902.07742

Parsa Mahmoudieh, Deepak Pathak, and Trevor Darrell. 2022. Zero-Shot Reward Specification via Grounded Natural Language. In *Proceedings of the 39th International Conference on Machine Learning (PMLR, Vol. 162)*. 14743–14752.

Nghia Nguyen, Minh Nhat Vu, Tung D. Ta, Baoru Huang, Thieu Vo, Ngan Le, and Anh Nguyen. 2024. Robotic-CLIP: Fine-tuning CLIP on Action Data for Robotic Applications. *arXiv* (2024). https://arxiv.org/abs/2410.13126

Thomas Pouplin, Katarzyna Kobalczyk, Hao Sun, and Mihaela van der Schaar. 2024. The Synergy of LLMs RL Unlocks Offline Learning of Generalizable Language-Conditioned Policies with Low-fidelity Data. *arXiv* (2024). doi:10.48550/arxiv.2412.06877

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. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning*.

Andrew Wagenmaker et al. 2025. Posterior Behavioral Cloning: Pretraining BC Policies for Efficient RL Finetuning. In *International Conference on Machine Learning*.

Yuke Zhu et al. 2020. robosuite: A Modular Simulation Framework and Benchmark for Robot Learning. In *arXiv preprint arXiv:2009.12293*.

A Implementation Details

BC. Huber loss ($\delta = 1$), Adam lr 10^{-4}, batch 256, 80 epochs, 90/10 train/val split.

SAC v2 (CLIP). Total timesteps 500k; buffer 200k; batch 256; $\gamma = 0.99$; $\tau = 0.005$; lr $3{\times}10^{-4}$; $\lambda_{\mathrm{bc}} = 20$ (constant); demo fraction 0.5; actor warmup 1000 gradient steps; actor BC pretrain 5000 steps; `ent_coef` 0.05; horizon 300; checkpoint every 50k.

Reward (v2, CLIP run). `success_bonus`$=2$, `wrong_object_penalty`$=-2$, `reach_scale`$=1$, `lift_scale`$=4$, `descent_scale`$=2$, `descent_gate_xy`$=0.06$ m, `grasp_bonus`$=2$, `grasp_gate_3d`$=0.04$ m, `time_penalty`$=0.001$; `closer_to_distractor_penalty`$=0$.

Object-pose observation (9-d). Table-relative gripper (x, y, z) plus near-block and far-block (x, y, z), ordered by depth (near→far) so target identity is not leaked by slot ordering.