OXFORD JOURNAL
OF STUDENT
SCHOLARSHIP

When Multimodality Hurts: The Effect of Language Ambiguity on Tabletop Manipulation Policies

Ethan Tian, Ivy He
ethanyangtian@gmail.com, xiao_he@brown.edu

ABSTRACT

We study how language ambiguity, scene complexity, input modality, instruction type, and object-set size affect target-slot prediction in simulated tabletop manipulation. A modality-configurable behavioral cloning classifier receives wrist-camera images, table-camera images, instruction text, or their combinations. Accuracy measures agreement with the recorded target slot, not successful grasp execution or physical manipulation. When averaged within each object-set size, full multimodal input is not the best configuration. Scene complexity has a significant but non-monotonic effect, while the planned tests do not support a consistent ambiguity effect or ambiguity–complexity interaction. Accuracy decreases from 61.11% with 3 objects to 49.77% with 5 and 2.78% with 9, but sparse per-slot coverage confounds the 9-object result. Failure analysis identifies similar-object confusions, spatial-relation errors, and instruction-valid alternatives penalized by exact-target scoring.

Keywords: *language ambiguity, multimodal learning, robot manipulation, behavioral cloning, vision language-action policies*

1. INTRODUCTION

Natural-language interfaces let users direct robots by describing a goal rather than programming a controller. Outside carefully scripted settings, however, instructions are often underspecified: “pick up the box” may identify one object in a simple scene but several among similar packages. The intended action therefore depends on both what the words permit and what the robot can distinguish, even when the policy itself produces a confident prediction.

Referring-expression and planning systems can detect uncertainty and request clarification [1, 2, 3]. By contrast, systems including CLIPort, RT-2, and OpenVLA learn robot actions from paired visual and linguistic inputs [4, 5, 6]. These vision-language-action (VLA) policies are nevertheless often assessed using unambiguous commands or aggregate success metrics. Neither approach isolates whether an error arose from multiple valid targets, visual similarity, the required reasoning, or multimodal fusion. It therefore remains unclear when another camera view or language input helps and when it adds misleading cues.

We investigate this question in a controlled simulated target-selection task. Given $N \in \{3, 5, 9\}$ objects and an instruction x, the policy predicts the recorded target slot. Referential ambiguity is the candidate set size $|C(x, S_N)|$: an instruction permits one, two, or three objects. We vary this factor alongside scene complexity—fully distinct objects, category overlap, or highly similar objects—and compare referential instructions with reasoning-based instructions that require scene-level comparisons. The design tests how target-slot accuracy changes with ambiguity, visual context, instruction type, input modality, and object-set size.

We use a modality-configurable behavioral cloning classifier whose inputs change while its prediction task remains fixed. It receives a wrist image, a fixed table image, instruction text, or evaluated combinations of these streams. Frozen ResNet-18 encoders represent images, a bag-of-words encoder represents text, and a fusion multilayer perceptron predicts one of N slots. This compact model is intended for controlled input ablations, not for competitive VLA performance.

Our factorial design combines three object-set sizes with three levels each of scene complexity and ambiguity and with two instruction types, yielding 54 conditions with 18 demonstrations each. Six modality configurations and five episode-level splits produce 1,620 runs; these splits reuse one collected dataset rather than repeat data collection. We use exact-match target-slot accuracy and matched statistical comparisons. Because this metric accepts only the recorded slot, it may reject another instruction-valid choice; we therefore pair accuracy with failure analysis.

Mean validation accuracy falls from 61.11% with 3 objects to 49.77% with 5 and 2.78% with 9, although sparse per-slot coverage confounds the 9-object collapse. Scene complexity significantly affects every object-set size, but category-overlap and symmetric scenes outperform distinct scenes for 3 and 5 objects. The full multimodal model is not the best configuration when averaged within each object-set size, and the planned tests do not support a consistent ambiguity effect or ambiguity–complexity interaction after Holm correction. Failures include similar-box confusions, spatial-relation errors, and plausible responses penalized by single-target scoring. Robustness therefore cannot be inferred from modality count alone; ambiguity must be evaluated alongside scene design, label semantics, and data coverage.

1.1 Contributions

This paper makes three contributions:

- We introduce a controlled formulation and simulated benchmark that vary candidate-set ambiguity, scene complexity, instruction type, input modality, and object-set size while keeping the target selection task consistent.
- We develop a modality-configurable behavioral cloning baseline and a matched evaluation protocol for comparing wrist vision, table vision, and language across 1,620 training runs.
- We provide an empirical and statistical account of when multimodal input does and does not help, together with a failure audit that exposes visual confusions, spatial-reasoning errors, strict-scoring artifacts, and data-coverage limits in the expanded evaluation.

1.2 Paper Organization

Section 3 defines the target-selection problem, ambiguity levels, scene-complexity levels, instruction types, and hypotheses. The next section describes the model, data-generation process, preprocessing, and evaluation protocol. The evaluation section reports the original pilot, the expanded factorial study, modality ablations, matched hypothesis tests, and failure analysis. The final sections summarize the conclusions, limitations, and directions for a larger and more tightly controlled follow-up study.

2 RELATED WORK

2.1 Ambiguity in Language

Ambiguity is intrinsic to natural language: information-theoretic accounts argue that efficient communication systems are necessarily ambiguous, since shared context lets speakers reuse short, frequent forms [7]. Modern language models still struggle to recognize ambiguity and recover plausible interpretations [8], while recent taxonomy work organizes the different ambiguity types that NLP systems must handle [9]. For embodied agents, the most consequential form is referential ambiguity—expressions such as “pick up the box” mapping to multiple candidate entities—which can only be resolved by integrating language with perceptual context.

2.2 Ambiguity in Language-Conditioned Robotics

Referring-expression grounding has long been a core HRI problem: a robot identifies the intended object among visually similar candidates and may query the user for clarification [1, 2]. More recent systems use VLMs or LLMs to detect task-level ambiguity and actively request clarification before execution— for instance, AmbResVLM raises downstream success from 69.6% to 97.1% [10]. A complementary line calibrates when to ask for help: KnowNo uses conformal prediction over LLM-planner outputs to provide statistical task-completion guarantees [3]. Related approaches incorporate introspective or affordance-based signals when estimating whether a robot should seek help [11, 12]. All of these treat ambiguity as something to resolve at the planner level; none characterize how ambiguity, left unresolved, propagates through a learned VLA policy.

2.3 Vision-Language-Action Models

VLA policies condition action generation jointly on visual observations and language [13]. CLIPort [4], RT-2 [5], and OpenVLA [6] are representative systems; their training draws on large cross-embodiment collections [14], and benchmarks such as LIBERO are commonly used for evaluation [15]. Training and evaluation commonly use fixed benchmark task descriptions [15, 6]. Robustness work challenges that assumption: under linguistic or layout perturbations, a VLA can follow fixed vision–action correlations instead of grounding the command [16], while binary success obscures important variation in execution quality [17]. SAFE instead detects likely failures from internal VLA features across unseen tasks [18]. Ambiguity, uncertainty, and silent degradation are therefore closely related, but prior evaluations have not isolated language ambiguity as an experimental factor.

2.4 Multimodality and Ambiguity

Multimodality is often framed as an antidote to ambiguity, with vision disambiguating language and vice versa [2]. VLA evidence complicates this: naive fusion fails to align pixel- and word-level statistics [13], and under ambiguous or visually redundant inputs, models frequently ignore the language channel and default to vision–action priors [16]. The benefit of additional modalities is therefore conditional, not monotonic.

2.5 Gap and Motivation

Prior work either resolves ambiguity at the planner level [10, 3, 2] or evaluates VLAs on mostly clear instructions with aggregate success metrics [15, 6, 17]. We investigate how language ambiguity affects target selection in manipulation settings, how it interacts with scene complexity, and when multimodal conditioning helps. Our experimental framework varies language ambiguity and scene complexity to characterize a compact target-slot classifier under linguistic and perceptual uncertainty.

3 PROBLEM FORMULATION

We investigate how a multimodal classifier selects targets under controlled levels of language ambiguity and scene complexity in simulated tabletop scenes.

Let $S_N = \{o_1, \dots, o_N\}$ denote a scene with $N \in \{3, 5, 9\}$. For a natural-language command x, the policy must identify an object $o^* \in S_N$ that satisfies that command. The agent receives visual observations (e.g., wrist-camera and table-camera images) and optionally a language instruction. The learning target is the single slot selected in the recorded demonstration; other objects may also satisfy an ambiguous command.

Unlike prior work that assumes clear and unambiguous instructions, we focus on settings where the instruction may be ambiguous and the environment may contain objects that are difficult to distinguish. This introduces two sources of uncertainty:

- Language ambiguity: multiple objects may satisfy the instruction.
- Scene complexity: objects may be visually or categorically similar.

Our central question is how language ambiguity and scene complexity jointly affect multimodal object selection, including whether combining vision and language consistently improves performance.

3.1 Experimental Factors

We define scene complexity, language ambiguity, and instruction type explicitly so that they can be varied systematically.

The original evaluation uses three objects placed at fixed spatial locations (left, center, right); the expanded evaluation also includes 5- and 9-object layouts. Let a three-object scene be denoted as:

$S = \{o_1, o_2, o_3\}$

where each object o_i has attributes such as category (e.g., box, bottle) and identity (e.g., sugar box, cracker box).

3.1.1 Scene Complexity

We operationalize scene complexity by how readily the objects can be told apart. Specifically, we construct three levels:

• Easy (Fully Distinct): All objects belong to different categories.

S = {mustard bottle, sugar box, tomato can}

• Medium (Category Overlap): Two objects share the same category while one differs. S = {sugar box, cracker box, mustard bottle}

• Hard (Symmetric / Duplicate): All objects share the same category or are highly similar. S = {box, box, box}

The levels are intended to reduce object distinguishability as complexity increases. Because the levels also contain different object identities, the evaluation treats this ordering as a hypothesis rather than an assumed fact.

3.1.2 Language Ambiguity

We define language ambiguity using a candidate set formulation. Given an instruction x, we define the candidate set:

$C(x, S) = \{o \in S \mid o \text{ is consistent with instruction } x\}$

The level of ambiguity is determined by the size of the candidate set $|C|$:

• Unambiguous: $|C| = 1$

• Somewhat Ambiguous: $|C| = 2$

• Highly Ambiguous: $|C| = 3$

Examples:

• "Pick up the mustard bottle" → $|C| = 1$

• "Pick up the box" → $|C| = 2$

• "Pick up an object" → $|C| = 3$

3.1.3 Instruction Types

We further categorize instructions into two types based on how the candidate set is derived:

• Referential Instructions: Objects are selected using explicit attributes (e.g., category or name), such as "the box" or "the mustard bottle." These correspond to attribute-based filtering: $C = \{o \in S \mid f_{\text{attr}}(o, x) = 1\}$.

• Reasoning-based Instructions: Objects are selected using relations over the scene, such as "the odd one out" or "the object that is different from the others." These require global comparison: $C = \{o \in S \mid f_{\text{rel}}(o, S, x) = 1\}$.

While both types can be expressed under the same candidate set formulation, they differ in how $C(x, S)$ is computed: referential instructions rely on local attributes, whereas reasoning-based instructions require relational inference over the entire scene.

3.2 Hypotheses

We test five hypotheses and one exploratory question. Here, performance refers to exact-match target-slot validation accuracy.

H1—Language ambiguity. Performance decreases as language ambiguity increases. A larger candidate set, $|C(x, S)|$, gives the policy more plausible objects to choose from, so accuracy should fall.

H2—Scene complexity. Performance decreases as scenes become more complex. Objects that look alike should be harder to match to an instruction.

H3—Multimodal fusion. Multimodal input does not consistently outperform unimodal input; effectiveness depends on task conditions. Extra inputs may help when their cues agree, but may hurt when they are noisy, redundant, or difficult to fuse.

H4—Instruction type. Policies perform better on referential instructions than on reasoning-based instructions. Referential instructions use names or attributes, while reasoning-based instructions require the policy to compare objects or understand their spatial relationships.

H5—Combined effects. Language ambiguity and scene complexity have a compounding effect. We expect the largest drop when an instruction is highly ambiguous and the objects are difficult to tell apart.

E1—Object-set size (exploratory). Accuracy decreases as the number of objects increases. We compare scenes with 3, 5, and 9 objects while leaving H1–H5 unchanged.

4 METHODS

4.1 Model Architecture

Our policy is a modality-configurable behavioral-cloning classifier. From the enabled camera and text channels, it predicts one of $N \in \{3, 5, 9\}$ candidate slots. The architecture consists of input encoders, a fusion layer, and a classification head.

Image encoders. A single frozen, ImageNet-pretrained ResNet-18 [19] serves both camera branches. After removing its original classification head, we use the convolutional stack and global-average-pooling output to obtain 512 features from each 224 × 224 image. The wrist and table streams share all encoder parameters.

Text encoder. Natural language instructions are encoded using an EmbeddingBag module with a bag of-words mean-pooling strategy, mapping tokenized instruction words into a shared 32-dimensional embedding space. A vocabulary of up to 5,000 words is built dynamically during training. This lightweight encoder is intentionally simple to isolate the effect of language as a modality rather than conflating it with pretrained language model capacity.

Fusion and classification. We join the representations from every enabled channel; using all three produces a 1,056-dimensional vector (512 + 512 + 32). The classification head reduces this vector to 64 units, applies ReLU and dropout (p = 0.3), and maps the result to N logits. We optimize cross-entropy with Adam, set weight decay to 10^{-4}, and anneal the learning rate according to a cosine schedule. Training lasts 30 epochs in the original pilot and 40 epochs in the expanded evaluation.

The modular design allows any non-empty subset of the three input modalities to be enabled or disabled at training and inference time without architectural changes, enabling the controlled ablation study described in Section 5.3. Figure 1 illustrates the full architecture.

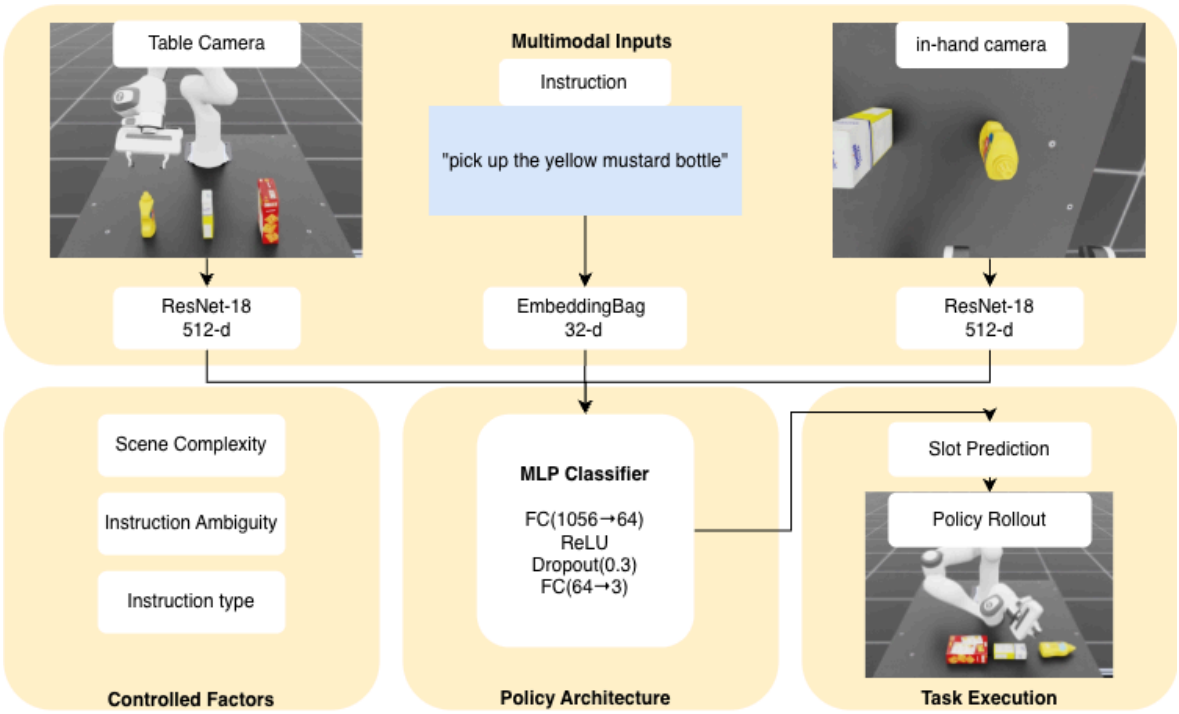


Figure 1: Target-slot classification architecture, illustrated for 3 objects. A shared ResNet-18 produces 512 features per camera image, and an EmbeddingBag produces 32 text features. A fusion MLP predicts one of $N \in \{3, 5, 9\}$ target slots; the depicted output layer uses $N = 3$. Evaluation ends at slot prediction. The downstream rollout and task-execution illustration does not represent a measured manipulation-success outcome.

4.2 Data Generation

For the original pilot, we collected 105 teleoperated demonstrations in NVIDIA Isaac Lab with Isaac Sim 5.1 [20] using a custom tabletop grasping task. The objects come from the YCB benchmark [21], which provides 3D models of household items such as mustard bottles, sugar boxes, and cracker boxes. Their visual similarities support the scene-complexity conditions defined in Section 3.

Pilot demonstrations are teleoperated. After reading a natural-language instruction, the operator moves the arm to the intended object and completes the grasp. The robot executes a structured action sequence consisting of three LOOK steps (scanning each of the three object positions) followed by a GRASP step (selecting the target slot). This yields six discrete actions: LOOK(0), LOOK(1), LOOK(2), GRASP(0), GRASP(1), GRASP(2), where the index denotes the left, center, or right spatial position respectively.

The 105 pilot demonstrations span 7 scenario groups × 3 target slots × 5 trajectories. The original ablation and ambiguity analyses use a 54-episode pilot subset (18 episodes per ambiguity level), as reported with their results. The expanded evaluation is separate: it crosses 3 object-set sizes, 3 scene-complexity levels, 3 ambiguity levels, and 2 instruction types, yielding 54 conditions with 18 episodes each. Every demonstration is paired with an instruction reflecting its target ambiguity level.

Each demonstration is stored as a .csv row containing the action sequence and instruction, together with .npy camera frames. Each episode saves one table_cam_initial.npy before any action. It also saves wrist frames during LOOK actions, when LOOK actions are present.

4.3 Data Preprocessing

All preprocessing is applied consistently across modalities prior to model training.

Visual preprocessing. We load each .npy camera array as a PIL image and use bilinear resampling to obtain a 224 × 224 image. Pixel intensities are normalized to [0, 1] before conversion to CHW tensors. When multiple wrist frames are available, they are averaged at the pixel level. If no wrist frames are available, the loader substitutes a zero tensor.

Language preprocessing. Instruction strings are lowercased and whitespace-tokenized. Token indices come from a dynamically built vocabulary of up to 5,000 words, with index 0 reserved for padding. The EmbeddingBag module mean-pools the word embeddings to produce a fixed 32-dimensional representation regardless of instruction length.

Label encoding. The prediction target for each episode is the final slot, i.e., the index $\{0, \ldots, N-1\}$ of the GRASP action taken by the demonstrator. Episodes in which the final slot is undefined (e.g., incomplete demonstrations) are excluded from the dataset.

Train/validation split. The full 105-demonstration pilot is randomly partitioned into 84 training and 21 validation episodes. The 54-example ablation uses its own 43/11 split. In the expanded evaluation, each

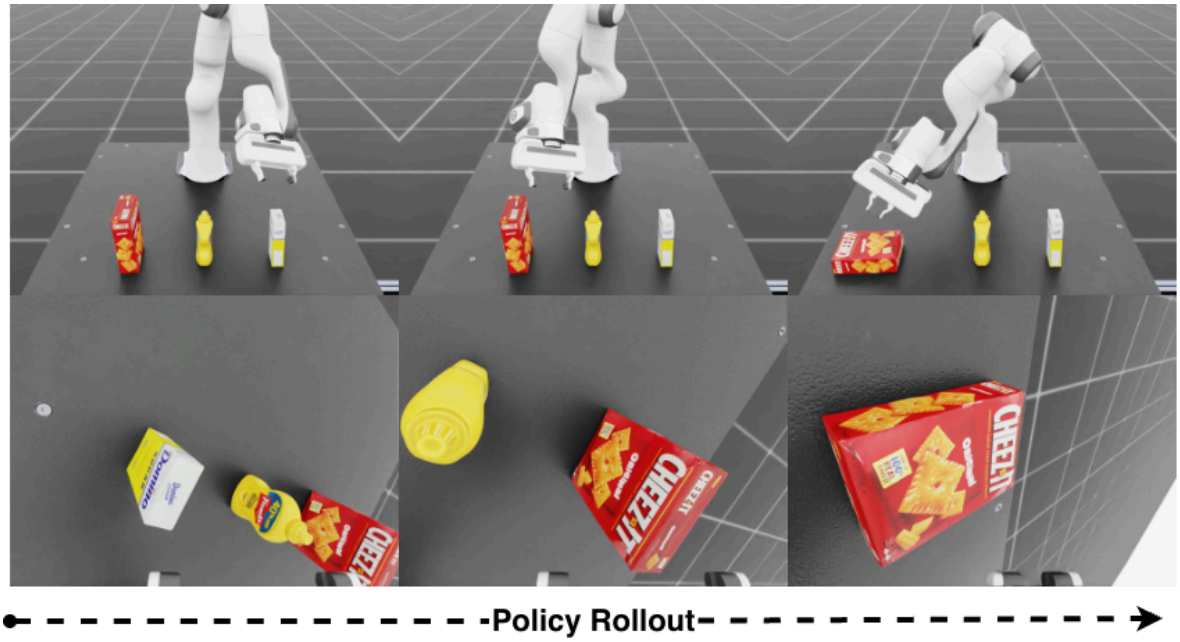


Figure 2: Camera views from an example recorded demonstration: fixed table camera (top) and wrist camera (bottom). The sequence illustrates the data source, not an evaluation of learned-policy grasp success.

18-episode condition is split into 14 training and 4 validation episodes under five split seeds. All splits are applied at the episode level to prevent frame-level leakage.

4.4 Evaluation Protocol

All reported accuracies measure exact-match target-slot prediction. A prediction is correct only when its slot matches the single recorded target slot; validation accuracy is measured at the final training epoch. An alternative object permitted by an ambiguous instruction is still scored as incorrect if it differs from that target. The evaluation does not measure grasp execution, object lifting, or successful physical manipulation in simulation or on a real robot. Recorded demonstrations provide observations and target labels, rather than evidence of learned-policy execution success. Figure 2 illustrates the recorded camera views.

5 EVALUATION

We evaluate how language ambiguity, scene complexity, instruction type, input modality, and object-set size affect target-slot accuracy. We first summarize the original three-object pilot and then report the expanded evaluation. The expanded study reuses one collected dataset and trains each condition five times with different data splits. These are training repetitions, not five independent rounds of data collection. We use the exact-match target-slot metric defined in Section 4.4. Its single-target scoring limitation is examined in the failure analysis (Section 5.9).

5.1 Original Three-Object Evaluation

Before the expanded evaluation, we ran a smaller three-object experiment. We include its original tables for comparison, but we do not combine its values with the larger experiment because the datasets and evaluation procedures are different. These results are best treated as early descriptive evidence for the updated hypotheses.

Table 1: Original pilot modality ablation on a 54-episode subset, split into 43 training and 11 validation episodes. Accuracies are exact-match target-slot percentages. A separate model is trained for each input combination; bold marks the highest validation accuracy.

Inputs used	Training loss	Validation loss	Training accuracy	Validation accuracy
Wrist image	0.887	1.221	58.14%	54.55%
Table image	0.501	1.141	81.40%	**72.73%**
Instruction text	0.110	1.632	100.00%	45.45%
Wrist image + instruction	0.477	1.425	86.05%	54.55%
Table image + instruction	0.707	1.133	69.77%	27.27%
Wrist image + table image	0.708	0.756	58.14%	45.45%
Wrist image + table image + instruction	0.661	0.724	67.44%	63.64%

Table 2: Original 3-object evaluation matrix. Exact-match target-slot validation accuracy (%) is shown for each instruction type, ambiguity level, and scene-complexity level.

Condition	Easy	Medium	Hard
Referential, $\|C\| = 1$	85.00%	65.00%	50.00%
Referential, $\|C\| = 2$	55.00%	90.00%	25.00%
Referential, $\|C\| = 3$	20.00%	70.00%	75.00%
Reasoning-based, $\|C\| = 1$	35.00%	80.00%	75.00%
Reasoning-based, $\|C\| = 2$	25.00%	65.00%	75.00%
Reasoning-based, $\|C\| = 3$	30.00%	60.00%	65.00%

The original ablation table gives descriptive motivation for H3. Table-only input has the highest validation accuracy (72.73%), while adding instruction text lowers it to 27.27%. The full multimodal model recovers to 63.64% but still does not beat table-only input. Thus, more modalities do not always help. The perfect training accuracy and 45.45% validation accuracy of the instruction-only model also suggest memorization rather than reliable language generalization.

Table 2 gives a more mixed picture for the other hypotheses. Accuracy does not always decrease as ambiguity grows, so the table does not give consistent support for H1. Hard scenes are not always worse than easy scenes, so the pilot does not consistently support H2. Referential instructions outperform reasoning-based instructions in some cells but not others, so H4 also depends on the scene and ambiguity level. Finally, the worst results do not occur only when both ambiguity and complexity are high, offering no clear descriptive support for H5. The matched tests below provide the stronger evidence used for the final hypothesis decisions.

5.1.1 Original Ambiguity and Data-Scale Trends

Figure 3 summarizes validation accuracy in the original ambiguity-specific pilot. Medium ambiguity reaches 75%, whereas the low- and high-ambiguity subsets reach 24% and 25%; the pooled mixed model reaches 63.64%. This non-monotonic result is descriptive and does not by itself support H1. It is retained to document the pilot behavior that motivated the larger matched evaluation.

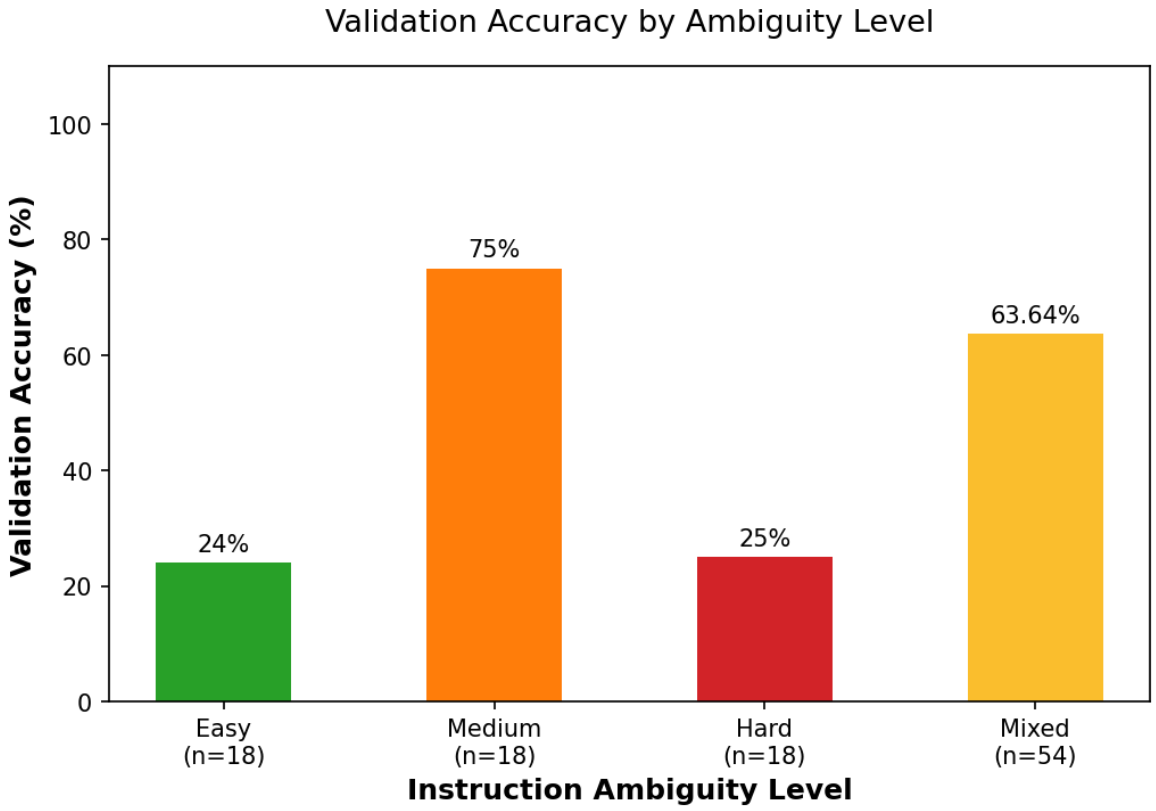


Figure 3: Original pilot exact-match target-slot validation accuracy (%) by instruction-ambiguity subset. The original plot labels the low-, medium-, and high-ambiguity models as Easy, Medium, and Hard. Each uses 18 examples; Mixed uses all 54. These values are not combined with the expanded evaluation.

Figure 4 reports the accompanying dataset-size sweep. Validation loss decreases by orders of magnitude as the number of examples grows, although the relative ordering of ambiguity conditions changes with dataset size. The plot therefore supports a data-coverage interpretation rather than a simple claim that one ambiguity level is intrinsically easiest. Because this sweep belongs to the original pilot, it is not used for the formal hypothesis decisions in Table 5.

5.2 Expanded Evaluation

The expanded grid contains 54 experimental conditions. Each condition has 18 episodes and is evaluated with six input configurations. Table 3 shows the main averages after combining the five training repetitions.

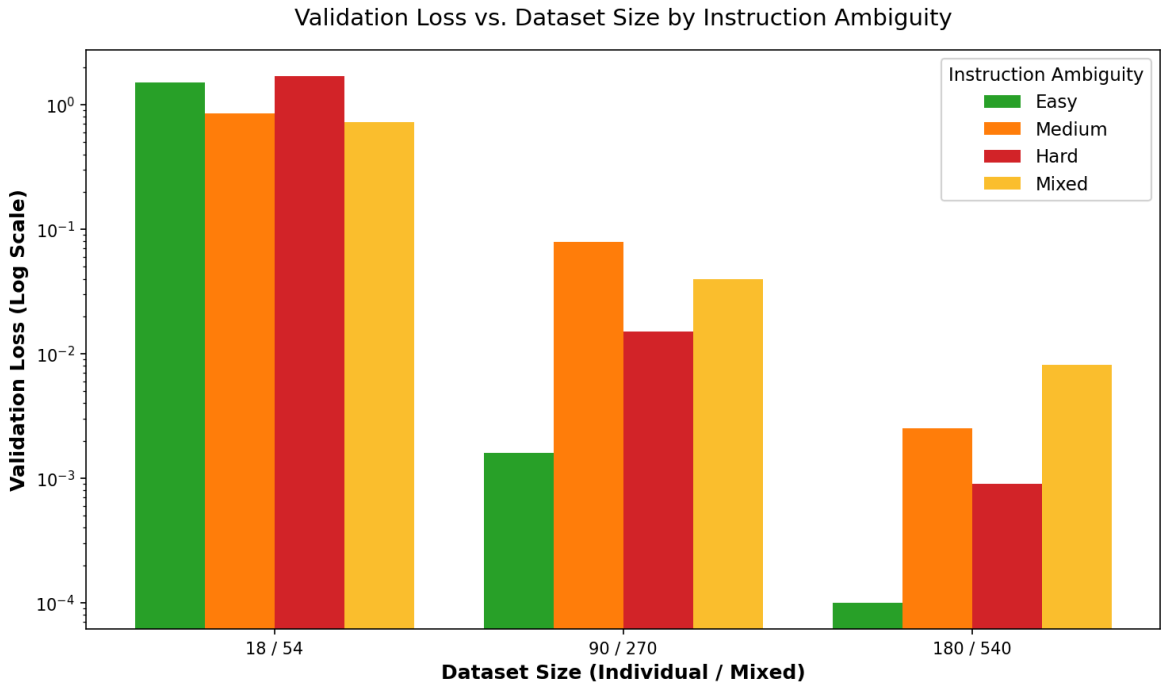


Figure 4: Original pilot target-slot classification validation loss by dataset size. Easy, Medium, and Hard denote low,

medium, and high instruction ambiguity, not scene complexity. Labels give the number of examples in each ambiguity specific dataset and the pooled Mixed dataset. The vertical axis is logarithmic.

The ambiguity, scene, and instruction columns are marginal means: each value also averages over the other experimental factors. Table 4 then separates the results by input configuration.

Across all conditions, accuracy is 61.11% with 3 objects, 49.77% with 5 objects, and 2.78% with 9 objects. The sharp 9-object drop is the largest pattern in the table, but it is not a clean measure of task difficulty. Every condition contains the same 18 episodes, so the 9-class model has only two examples per slot before splitting. Some 9-object splits also omit a slot from the training set. We return to this issue below.

Table 3: Mean exact-match target-slot validation accuracy (%) across five training repetitions. The ambiguity, scene, and instruction columns average over the six modalities and all other factors. Ref. and Reason. mean referential and reasoning-based instructions. Overall is mean ± standard deviation across the five repetition-level means.

	Ambiguity			Scene complexity			Instruction type		
Objects	$\|C\| = 1$	$\|C\| = 2$	$\|C\| = 3$	Distinct	Overlap	Symmetric	Ref.	Reason.	**Overall**
3	65.00	58.61	59.72	45.97	70.00	67.36	60.93	61.30	61.11 ± 16.70
5	60.00	51.39	37.92	31.25	42.78	75.28	65.00	34.54	49.77 ± 13.11
9	4.72	2.36	1.25	5.83	2.36	0.14	4.26	1.30	2.78 ± 1.40

Table 4: Exact-match target-slot validation accuracy (%, mean ± standard deviation across five training repetitions) by modality and object-set size. Each repetition-level mean averages over scene complexity, ambiguity, and instruction type. Bold marks the highest mean in each column. The modality differences for 9 objects are not statistically significant.

Inputs used	**3 objects**	**5 objects**	**9 objects**
Wrist	57.50 ± 14.45	52.22 ± 13.16	3.33 ± 1.86
Table	64.17 ± 16.94	45.83 ± 16.26	2.78 ± 2.95
Wrist + Table	54.17 ± 17.76	46.39 ± 11.60	1.11 ± 1.16
Wrist + Instruction	63.06 ± 19.16	$\mathbf{54.72 \pm 13.70}$	$\mathbf{5.00 \pm 3.34}$
Table + Instruction	$\mathbf{69.72 \pm 17.44}$	49.72 ± 17.16	2.50 ± 2.85
Full multimodal	58.06 ± 18.15	49.72 ± 11.84	1.94 ± 1.58

5.3 Multimodal Ablation Study

H3 asks whether multimodal input consistently outperforms simpler input conditions, or whether its value depends on the task. The expanded grid includes six modality configurations: wrist, table, wrist+table, wrist+instruction, table+instruction, and full multimodal input. It does not include an instruction-only condition, so the formal expanded test is limited to these six available configurations.

The current means are mixed. With 3 objects, table+instruction is highest at 69.72%, while the full model reaches 58.06%. With 5 objects, wrist+instruction is highest at 54.72%, while the full model reaches 49.72%. The full model is not the best configuration at any object-set size.

A Friedman test across the six available modalities finds a significant modality effect for 3 and 5 objects after Holm correction (adjusted p = .0156 for both), but not for 9 objects (adjusted p = .371). Pairwise Wilcoxon follow-ups show that table+instruction outperforms wrist+table for 3 objects and wrist+instruction outperforms wrist+table for 5 objects. H3 is therefore supported for 3 and 5 objects, while the 9-object comparison is inconclusive because all means are near zero.

5.4 Ambiguity Test

H1 predicts that accuracy falls as an instruction allows more targets. For this test, we use only configurations that receive instruction text. Camera-only models cannot observe language ambiguity and would weaken the comparison.

The clearest decline occurs with 5 objects: accuracy falls from 61.11% to 51.94% to 41.11% as $|C|$ increases from 1 to 3. The 9-object means also fall, from 5.83% to 2.78% to 0.83%, but they are close to zero. The 3-object means are not monotonic: 69.17%, 60.56%, and 61.11%. After correction, none of the three matched Friedman tests is significant. The adjusted p-values are .591, .591, and .591 for 3, 5, and 9 objects. H1 is therefore not supported. The 5-object trend is worth testing again with a larger validation set.

5.5 Scene Complexity Test

H2 predicts lower accuracy when objects are harder to tell apart. The data do not show one common direction. For 3 objects, accuracy is 45.97% in fully distinct scenes, 70.00% with category overlap, and 67.36% in symmetric scenes. The more complex scenes perform better, which is opposite to H2 (adjusted $p = .00243$). The same reversal appears with 5 objects: 31.25%, 42.78%, and 75.28% (adjusted $p < .001$).

Only the 9-object result follows the expected ordering, decreasing from 5.83% to 2.36% to 0.14% (adjusted $p = .000354$). However, every 9-object value is near the floor. Scene type clearly changes the result, but the experiment does not support the simple claim that visually similar scenes are always harder. The scene levels also use different object sets, so object identity may contribute to the observed differences.

5.6 Instruction Type Test

H4 predicts that referential instructions are easier than reasoning-based instructions. As in the ambiguity test, we use only models that receive the instruction. The 3-object means are close: 64.26% for referential instructions and 62.96% for reasoning-based instructions. With 5 objects, the gap is much larger: 67.59% versus 35.19%. This 32.40-point difference remains significant after correction ($p < .001$). With 9 objects, referential instructions reach 5.19% and reasoning-based instructions reach 1.11%, but the matched difference is not statistically significant. The adjusted p-values are .539, < .001, and .318 for 3, 5, and 9 objects. H4 is supported only in the 5-object setting.

5.7 Ambiguity–Complexity Interaction Test

H5 predicts the largest failure when high language ambiguity and high scene complexity occur together. Figure 5 shows a descriptive view of the full multimodal model with 3 objects.

The heatmap does not show a steady decline. Medium-complexity scenes average 71.67%, compared with 60.83% for hard scenes and 41.67% for easy scenes. In easy scenes, accuracy falls from 60.0% to 25.0% as ambiguity grows, but in hard scenes it rises from 62.5% to 70.0%. The figure's model-specific contrast is +42.5 points, opposite to the predicted compounding decline.

The planned interaction test uses only models that receive instruction text. It compares the ambiguity change in fully distinct and symmetric scenes. No object-set-specific result remains significant after correction: the adjusted p-values are .519, .519, and .665 for 3, 5, and 9 objects. H5 is not supported.

5.8 Hypothesis Testing and Statistical Analysis

We first average the five training repetitions within each exact experimental condition. We then compare conditions that match on every factor except the one being tested. H1, H2, and H3 use matched Friedman

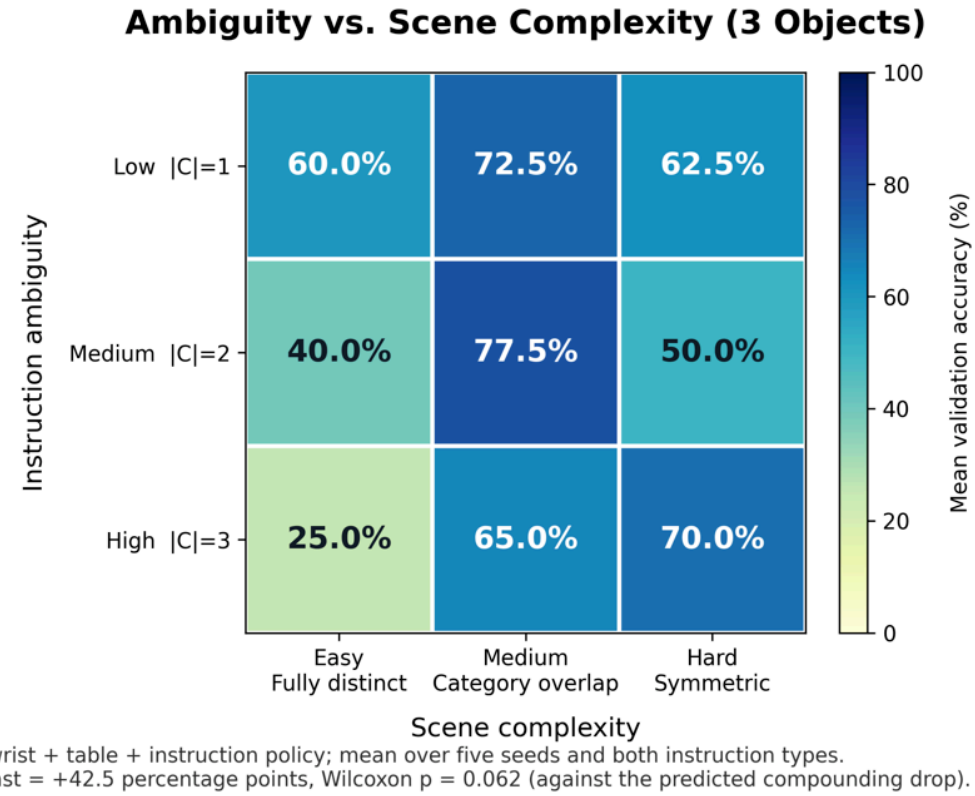


Figure 5: Language ambiguity and scene complexity in 3-object scenes. Each cell gives mean exact-match target

slot validation accuracy (%) for the full multimodal model, averaged over both instruction types and five training repetitions. The plot's annotated test concerns this full-model contrast; the planned H5 tests include all instruction aware configurations. tests; H3 also uses pairwise Wilcoxon follow-ups to identify which modalities differ. H4 uses a paired Wilcoxon test, and H5 uses a difference-in-differences comparison with a paired Wilcoxon test. H1, H4, and H5 include only models that receive instruction text. Holm correction with $\alpha = .05$ is applied within each hypothesis family across the three object-set sizes. This prevents the five training repetitions from being counted as five independent datasets and lowers the risk of false positive findings.

The object-set-size comparison in Table 3 has a large matched effect ($W = .80$, adjusted $p < .001$), and all three pairwise comparisons are significant. However, the number of examples is not held constant per class. The 3-, 5-, and 9-object conditions all use 18 episodes, so increasing the number of slots also reduces the examples available for each slot.

The 9-object mean is below its 11.11% uniform-choice baseline. An audit of the 90 condition-by-split combinations at this size found that 18 splits omit at least one target slot from training and then contain an unseen target in validation. The low result therefore shows a learning collapse in the current setup, but it does not prove that object count alone caused the collapse. A clean size comparison requires the same number of training examples per slot and a stratified test set.

Table 5: Matched tests of exact-match target-slot validation accuracy in the expanded evaluation. Each comparison holds the other experimental factors fixed. Adjusted *p*-values are listed for 3, 5, and 9 objects, respectively, where applicable. "Not supported" means that the data do not provide enough evidence for the hypothesis; it does not prove that the effect is zero.

ID	Statistical test	Controlled comparison	Result and interpretation
H1	Friedman matched-samples test.	Compare $\|C\| = 1, 2, 3$ among instruction-aware models, matched by scene, instruction type, and modality.	Not supported after correction: adjusted $p = .591$, $.591$, and $.591$ for 3, 5, and 9 objects.
H2	Friedman matched-samples test.	Compare scene levels within each object-set size, matched by ambiguity, instruction type, and modality.	Significant for all sizes, but opposite to the prediction for 3 and 5 objects; adjusted $p = .00243$, $< .001$, and $.000354$.
H3	Friedman test across modalities with pairwise Wilcoxon follow-ups.	Compare wrist, table, wrist+table, wrist+instruction, table+instruction, and full multimodal input under matched tasks.	Supported for 3 and 5 objects: modality has a significant effect (adjusted $p = .0156$ for both). Not supported for 9 objects (adjusted $p = .371$). Full multimodal is not the best object-size marginal mean.
H4	Paired Wilcoxon signed-rank test.	Match instruction types by scene, ambiguity, and instruction-aware modality.	Supported only for 5 objects: 67.59% referential versus 35.19% reasoning; adjusted $p = .539$, $< .001$, and $.318$.
H5	Paired difference-in-differences with Wilcoxon test.	Compare low/high ambiguity in easy/hard scenes among instruction-aware models.	Not supported: adjusted $p = .519$, $.519$, and $.665$.
E1	Friedman test with paired Wilcoxon follow-ups.	Compare object-set sizes matched by scene, ambiguity, instruction type, and modality.	Large descriptive effect ($W = .80$, adjusted $p < .001$), but confounded by per-class data coverage.

5.9 Failure Analysis

Average accuracy does not tell us why a prediction fails. The model may confuse similar-looking objects, receive an instruction that allows several answers, misread a spatial relationship, or combine input types poorly. We use confusion counts and annotated examples to describe these four kinds of error. These results are exploratory.

5.9.1 Object-Level Confusions

Figure 6 shows that errors are concentrated on particular objects. The cracker and sugar boxes account for 18 confusions in total across both directions, or 52.9% of the 34 errors in these 120 predictions. The pattern is uneven: cracker is predicted as sugar 14 times, while sugar is predicted as cracker only four times. Accuracy is 54.76% for cracker, 71.43% for sugar, and 91.67% for the more visually distinct mustard bottle. This suggests both visual similarity and a learned bias away from the cracker label.

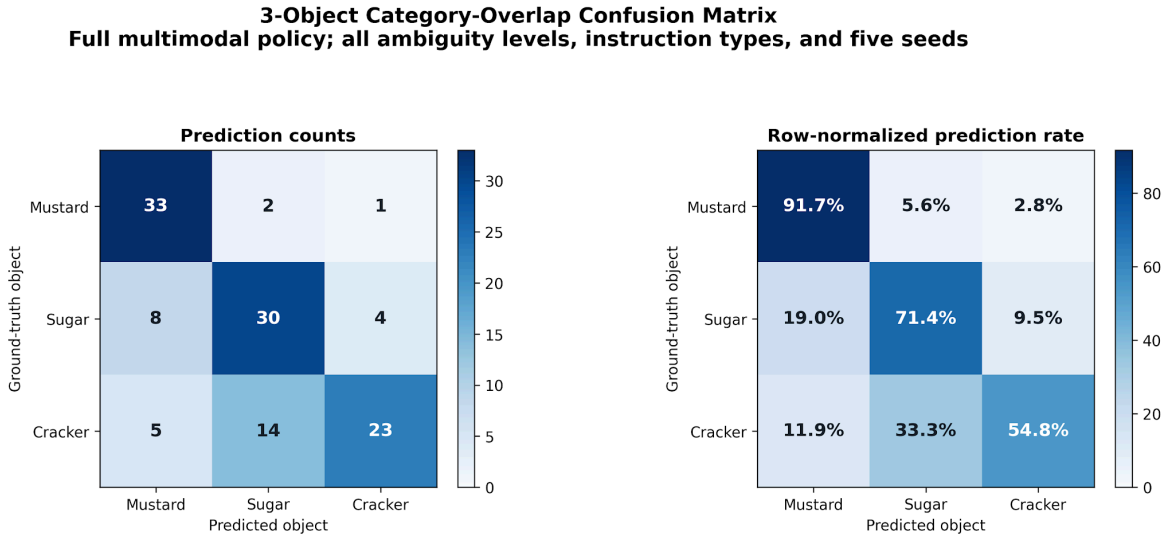


Figure 6: Target-slot predictions mapped to object identities for the full multimodal model in 3-object, category overlap scenes. Counts combine all ambiguity levels, instruction types, and five training repetitions; episodes may recur across validation splits. The left matrix reports raw counts, whereas the right matrix reports row-normalized percentages.

5.9.2 Ambiguity and Instruction Type

For the full multimodal model with 3 objects, accuracy drops by 9.17 points from unambiguous to somewhat ambiguous instructions, then by another 2.50 points at high ambiguity. This trend alone does not support H1 because the formal instruction-aware matched test is not significant and the direction changes across scenes.

Some broad instructions allow more than one valid object. A prediction may follow the instruction but still be marked wrong because it differs from the single recorded target. These cases should be separated from selections that clearly violate the instruction. In the same 3-object subset, reasoning-based accuracy is 56.67%, compared with 59.44% for referential instructions. Across all six modalities, including camera only models, the order reverses slightly and the difference is not significant. The examples show a more useful distinction: referential errors often involve object identity or vague wording, while reasoning errors often involve left/right or adjacency relationships.

Cases F1–F3 all predict sugar when cracker is the recorded target. F1 explicitly names the cracker box, so its error points to visual or slot confusion. F2 and F3 allow either box: the sugar prediction can satisfy the instruction despite disagreeing with the recorded target. F4 selects outside the named pair, whereas F5–F7 illustrate spatial-grounding errors. These examples suggest possible causes; they do not measure their frequencies in the full dataset.

Table 6: Representative predictions scored as incorrect under exact-match target-slot evaluation. Examples include similar-object confusions, instruction-valid alternatives to the recorded target, choices outside an allowed set, and spatial-selection errors.

A. Conditions and target-slot predictions

ID	Scene	Ambiguity	Instruction type	Recorded target	Prediction
F1	Medium	$\|C\| = 1$	Referential	Cracker box	Sugar box
F2	Medium	$\|C\| = 2$	Referential	Cracker box	Sugar box
F3	Medium	$\|C\| = 2$	Reasoning-based	Cracker box	Sugar box
F4	Easy	$\|C\| = 2$	Referential	Mustard bottle	Tomato can
F5	Easy	$\|C\| = 1$	Reasoning-based	Sugar box	Mustard bottle
F6	Hard	$\|C\| = 1$, symmetric	Referential	Box B	Box C
F7	Hard	$\|C\| = 1$, symmetric	Reasoning-based	Box A	Box C

B. Instructions and interpretations

ID	Instruction	Interpretation
F1	Pick up the cracker box	Two visually similar boxes; left/center slot confusion.
F2	Pick up a box	Either box satisfies the instruction; the prediction differs from the recorded target.
F3	Pick one of the two objects in the category represented twice	The prediction may satisfy the broad instruction but differs from the recorded target.
F4	Pick either the mustard bottle or the sugar box	The choice is outside the named pair; bottle/can shape similarity may contribute.
F5	Pick the object immediately right of the object at the left position	Relational grounding failure.
F6	Pick the cracker box at the center position	Center-position grounding error.
F7	Pick the object immediately left of Box B	Spatial-relation failure.

Table 7: Illustrative prediction patterns by ambiguity level for the full multimodal 3-object subset. Exact-match target-slot validation accuracy (%) is averaged over scene complexity, instruction type, and five training repetitions. The qualitative examples do not estimate error-type frequencies.

Ambiguity level	Instruction-valid selection	Scored-error interpretation	Accuracy
Unambiguous ($\|C\| = 1$)	Clear names or position words identify distinct targets; the mustard bottle performs especially well.	Similar-object confusion, slot bias, and failed relational grounding.	65.00%.
Somewhat ambiguous ($\|C\| = 2$)	A choice of either allowed object can satisfy the instruction in category-overlap scenes.	The model sometimes chooses an object outside the named pair.	55.83%.
Highly ambiguous ($\|C\| = 3$)	Any of the three objects can satisfy the broad instruction.	The prediction differs from the hidden target because the wording does not identify one object.	53.33%; some scored errors may be acceptable choices.

Table 8: Illustrative prediction patterns by instruction type for the full multimodal 3-object subset. Exact-match target-slot validation accuracy (%) is averaged over scene complexity, ambiguity, and five training repetitions. Spatial selection errors are qualitative examples, not measured error-type rates.

Instruction type	Correct-selection example	Scored-error example	Observed challenge
Referential	A clear object name can identify the intended target.	Similar boxes are confused, and vague descriptions can match several objects.	Position words may be ignored; 59.44% accuracy.
Reasoning-based	Position clues can separate objects with similar appearance.	The model chooses the wrong left, center, or right object.	Spatial relationships are difficult; 56.67% accuracy.

5.9.3 Modality Interactions

The ablations in Section 5.3 suggest possible interference between inputs, but the accuracy comparisons do not establish a fusion failure mechanism. Prediction-level comparisons would be needed to determine which errors arise when an input is added. The near-zero 9-object results further limit interpretation of modality differences.

5.9.4 Scene and Object-Set Size

Scene difficulty depends on more than how similar the objects look. In the 3-object results, medium scenes may offer a useful contrast between one distinct object and a repeated category. Easy scenes still expose failures with spatial words and broad instructions. In symmetric scenes, broad instructions can make several identical-looking objects acceptable, which may explain why accuracy does not always fall as ambiguity increases.

The split audit in Section 5.8 identifies poor per-slot coverage as a confound in the 9-object results. Slot bias or grounding errors are possible additional explanations, but the below-chance accuracy alone cannot distinguish these mechanisms from data-coverage effects.

6 CONCLUSION

This study examines target-slot prediction in simulated tabletop scenes. Scene complexity affects accuracy without a consistent monotonic ordering, and modality choice matters for 3- and 5-object scenes, where the full multimodal model is not the strongest configuration. The corrected tests do not support a consistent ambiguity effect or ambiguity–complexity interaction; referential instructions significantly outperform reasoning-based instructions only in the 5-object setting.

The 9-object decline is confounded by sparse per-slot training coverage. Similar-object confusions, spatial selection errors, and instruction-valid alternatives rejected by exact-match scoring further limit interpretation. These findings concern agreement with recorded target slots in the present classifier and dataset; they do not establish successful physical manipulation. Broader claims require better-controlled data coverage and separate evaluation of instruction-valid selection and executed manipulation.

7 LIMITATIONS & FUTURE WORK

Target selection is only one component of manipulation. The present evaluation does not test whether a selected object can be grasped, lifted, or moved successfully. Future work should report closed-loop manipulation success separately from exact-match target-slot accuracy and candidate-set-valid selection, which credits any object permitted by the instruction.

The main limitations are the small validation sets and unequal examples per class. Each condition has only four validation episodes per repetition, so a single prediction changes one cell's accuracy by 25 percentage points. The fixed total of 18 episodes also makes the 9-object task much less supervised than the 3-object task. In addition, scene complexity changes together with object identity, making it difficult to separate visual similarity from the particular objects used.

Future experiments should collect more independent scenes, keep examples per slot constant across object set sizes, and reserve a larger untouched test set. They should also test new object arrangements and use repeated data collection, rather than relying only on different training splits. These changes would make the results easier to interpret and more useful for studying real instruction-following systems.

Future work could also evaluate more advanced models. A pretrained language encoder could represent meaning and spatial relationships better than the current bag-of-words encoder. Stronger visual encoders could improve object recognition, while attention-based fusion could help the model decide which camera view or instruction words are most useful for each prediction. Larger vision-language-action models could then be tested with the same evaluation framework to determine whether the observed ambiguity and scene complexity patterns also appear in more capable systems.

REFERENCES

[1] Tom Williams and Matthias Scheutz. Resolution of referential ambiguity in human–robot dialogue using Dempster–Shafer theoretic pragmatics. In Proceedings of Robotics: Science and Systems, 2017.

[2] Mohit Shridhar, Dixant Mittal, and David Hsu. INGRESS: Interactive visual grounding of referring expressions. The International Journal of Robotics Research, 39(2–3):217–232, 2020.

[3] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners. In Proceedings of the 7th Conference on Robot Learning, volume 229 of Proceedings of Machine Learning Research, pages 661–682. PMLR, 2023.

[4] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. CLIPort: What and where pathways for robotic manipulation. In Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 894–906. PMLR, 2022.

[5] Anthony Brohan et al. 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.

[6] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P. Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In Proceedings of the 8th Conference on Robot Learning, volume 270 of Proceedings of Machine Learning Research, pages 2679–2713. PMLR, 2025.

[7] Steven T. Piantadosi, Harry Tily, and Edward Gibson. The communicative function of ambiguity in language. Cognition, 122(3):280–291, 2012.

[8] Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. We're afraid language models aren't modeling ambiguity. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 790–807, Singapore, 2023. Association for Computational Linguistics.

[9] Margaret Y. Li, Alisa Liu, Zhaofeng Wu, and Noah A. Smith. A taxonomy of ambiguity types for NLP. arXiv preprint arXiv:2403.14072, 2024. Presented at the UnImplicit workshop at EACL 2024.

[10] Eugenio Chisari, Jan Ole von Hartz, Fabien Despinoy, and Abhinav Valada. Robotic task ambiguity resolution via natural language interaction. arXiv preprint arXiv:2504.17748, 2025.

[11] Kaiqu Liang, Zixu Zhang, and Jaime Fernández Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. In Advances in Neural Information Processing Systems, volume 37, 2024.

[12] James F. Mullen, Jr. and Dinesh Manocha. Towards robots that know when they need help: Affordance-based uncertainty for large language model planners. arXiv preprint arXiv:2403.13198v1, 2024. Version 1; later versions use the title "LBAP: Improved Uncertainty Alignment of LLM Planners using Bayesian Inference".

[13] Muhayy Ud Din, Waseem Akram, Lyes Saad Saoud, Jan Rosell, and Irfan Hussain. Vision language action models in robotic manipulation: A systematic review. arXiv preprint arXiv:2507.10672, 2025.

[14] Open X-Embodiment Collaboration. Open X-embodiment: Robotic learning datasets and RT-X models. In 2024 IEEE International Conference on Robotics and Automation, pages 6892–6903. IEEE, 2024.

[15] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In Advances in Neural Information Processing Systems, volume 36, pages 44776–44791, 2023.

[16] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. arXiv preprint arXiv:2510.13626, 2025.

[17] Pablo Valle, Chengjie Lu, Shaukat Ali, and Aitor Arrieta. Evaluating uncertainty and quality of visual language action-enabled robots. arXiv preprint arXiv:2507.17049, 2025.

[18] Qiao Gu, Yuanliang Ju, Shengxiang Sun, Igor Gilitschenski, Haruki Nishimura, Masha Itkina, and Florian Shkurti. SAFE: Multitask failure detection for vision-language-action models. In Advances in Neural Information Processing Systems, volume 38, 2025. Also available as arXiv:2506.09937.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[20] Mayank Mittal et al. Isaac Lab: A gpu-accelerated simulation framework for multi-modal robot learning. arXiv preprint arXiv:2511.04831, 2025.

[21] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In 2015 International Conference on Advanced Robotics, pages 510–517. IEEE, 2015.