PhysGraphNet: Physical-State Scene Graphs via Latent Graph Reasoning and Counterfactual Supervision

Zhengtao Yao[1*] Runhao Li[1*] Siheng Wang[3*] Yan Wen[1] Guang Yang[2]
Chenhao Wei[2] Rongchao Zhang[3] Guoqing Ma[3] Haoyan Xu[1] Junhao Dong[4†]

[1]University of Southern California, Los Angeles, CA, USA [2]Stevens Institute of Technology, Hoboken, NJ, USA
[3]University of California, Berkeley, CA, USA [4]Nanyang Technological University, Singapore

Abstract

Manipulation planners depend on accurate predicates about physical state—is an object occluded? is a container open? is a path blocked?—but such predicates are hard to read off an image. Contrastively-trained vision–language models capture semantic content but collapse to near-zero binary F1 on these predicates, and even GPT-4o few-shot falls below random on multiple-choice physical-state questions. We present PHYSGRAPHNET, which predicts physical-state scene graphs from a single image and a natural-language goal. A frozen CLIP ViT, adapted with LoRA, feeds a heterogeneous graph of object, relation, and learnable memory tokens into a Latent Graph Reasoning Transformer (LGRT); training combines supervised, counterfactual-margin, contrastive, and calibration losses with offline counterfactual-pair augmentation. On PACBENCH, PHYSGRAPHNET reaches annotated-slot μ-F1 $= 0.651$, a $+0.163$ absolute gain over the strongest supervised baseline. Because the two-choice MCQ task is saturated by several supervised models, we treat MCQ as an auxiliary planner-facing readout and base our architectural conclusions on predicate F1, parameter-matched graph controls, and paired counterfactual metrics. Five-seed experiments preserve large improvements from the LGRT and counterfactual training, and parameter-matched structural controls (flat-token, pair-local, and incidence-shuffled Transformers) confirm that the gain comes from graph structure rather than capacity. On MANIPBENCH, a retrained PHYSGRAPHNET reaches $74.1\pm1.0\%$ MCQ against $62.1\pm1.2\%$ for a parameter-matched, comparably supervised no-graph Transformer (and 30.0% for the best zero-shot VLM). Cross-benchmark and GQA-Physical transfer evaluations further test low-data and vocabulary generalization, with the largest advantages in the low-data regime.

1 Introduction

Robotic manipulation depends on a handful of simple physical predicates: *is the target occluded*, *is the path blocked*, *is the container open*. A downstream planner can exploit them directly, but reading them off a single image is surprisingly hard. They do not reduce to image–text similarity: occlusion is geometric, blockage is relational, and openness requires reasoning about state changes under intervention. Table 2 shows that five widely-used zero-shot vision–language models—CLIP [26], SigLIP [39], DINOv2 [24], BLIP-2 [20], and InstructBLIP [4]—all achieve μ-F1$=0.000$ on binary physical predicates on PACBENCH. Trivial random/majority baselines are also weak, and GPT-4o

*Equal contribution.
†Corresponding author: junhao003@ntu.edu.sg.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

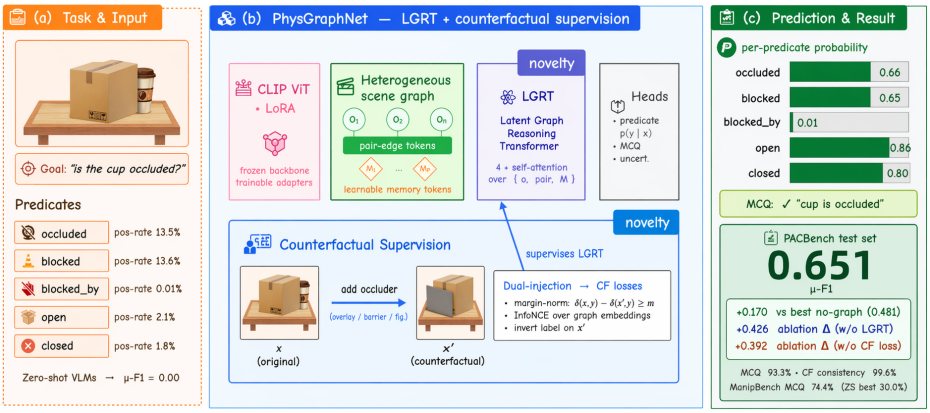

Figure 1: **Overview of PHYSGRAPHNET. (a)** From a single RGB image and a natural-language goal (e.g. *"is the cup occluded?"*), we predict a scene-graph-level set of physical-state predicates—`occluded`, `blocked`, `blocked_by`, `open`, `closed`—that are rare (positive rates 0.01–13.6%) and that zero-shot VLMs miss entirely (μ-F1=0). **(b)** A frozen CLIP ViT (LoRA-adapted) produces object tokens and a goal embedding; these are concatenated with pair-edge and learnable graph-memory tokens and passed through the Latent Graph Reasoning Transformer (LGRT), a 4-layer Transformer that self-attends across all three token types. Training combines the predicate loss with a counterfactual-pair objective: for each scene we synthesise an intervened image (add occluder / barrier / flip container) and apply margin-rank and InfoNCE losses with inverted labels (dual injection). **(c)** Per-predicate probabilities feed both the MCQ head and the final report. On PACBENCH, PHYSGRAPHNET reaches μ-F1=0.651 and 99.3% MCQ; five-seed component ablations attribute +0.411 μ-F1 to the LGRT (*w/o LGRT*) and +0.373 to the counterfactual loss (*w/o CF loss*).

5-shot gets 40.0% MCQ accuracy, *below* the 52.6% random baseline for 2-choice questions. Scale alone does not close the gap.

Existing vision pipelines do not fill it either. Scene graph generation [36, 38] targets semantic relations ("person riding horse") on benchmarks that do not annotate physical state. Intuitive-physics benchmarks [1, 7, 2] use synthetic dynamics rather than single real images. Task-and-motion planning [6, 15] consumes predicates that an upstream perception module is assumed to supply.

We take that perception module seriously. PHYSGRAPHNET predicts physical-state scene graphs end-to-end from a single RGB image and a natural-language goal; Figure 1 summarises the framework. A frozen CLIP ViT backbone with LoRA adapters produces object tokens from object nodes and optional box metadata; these are joined with a goal embedding, relation-candidate tokens, and learnable *graph memory tokens* and passed through a Latent Graph Reasoning Transformer (LGRT), a 4-layer Transformer that self-attends jointly over objects, candidate relations, and memory. The reasoning-side novelty is a counterfactual-pair loss with dual injection: for each training scene we synthesise an intervened image (adding an occluder, a barrier, or flipping container state) and require per-predicate probabilities to shift with margin in the expected direction, backed by an InfoNCE term on graph embeddings and an uncertainty-calibration head. Predicate classes are learned in a three-phase curriculum (unary → pairwise → joint) to stabilise the six-term multi-task objective.

On PACBENCH [10], PHYSGRAPHNET reaches annotated-slot μ-F1=0.651, a +0.163 absolute gain over the strongest supervised baseline (Scene Graph Pairwise MLP, μ-F1=0.488). Because the two-choice MCQ task is saturated by several supervised models, we treat MCQ as an auxiliary planner-facing readout and base our architectural conclusions on predicate F1, parameter-matched graph controls, and paired counterfactual metrics. Five-seed ablations isolate the contributions: removing the LGRT drops μ-F1 by -0.411 and removing the counterfactual loss by -0.373 (Section 4.6), and parameter-matched structural controls confirm the gain comes from graph structure rather than capacity (Appendix B.2). A model trained instead on MANIPBENCH reaches 74.1±1.0% MCQ, against 62.1±1.2% for a parameter-matched, comparably supervised no-graph Transformer and 30.0% for the best zero-shot VLM. The supplementary archive includes all per-method result files.

Contributions. Our contributions are fourfold. **(1)** We formulate goal-conditioned physical-state prediction as *slot-grounded* scene-graph inference from a single RGB image, while explicitly separating this setting from end-to-end object detection, and show that every zero-shot VLM we tested fails on it. **(2)** We design LGRT and isolate structured relational reasoning through controlled component removals, *parameter-matched* flat-token controls, pair-local attention, and graph-incidence shuffling (Sections 4.6 and Appendix B.2). **(3)** We introduce controlled counterfactual supervision and evaluate it using paired endpoint correctness, signed intervention magnitude, held-out intervention templates, and a directional-consistency diagnostic (Appendix C). **(4)** We evaluate supervised and low-data transfer on PACBENCH, MANIPBENCH, and a mapped GQA-Physical subset (Appendix E), while limiting our claims to static, visually inferable physical states.

2 Related Work

Vision–language models. Contrastive image–text encoders—CLIP [26], SigLIP [39], ALIGN [14]—are strong on semantic retrieval but encode little about relational or intervention structure. Caption- and VQA-tuned multimodal LLMs [20, 4, 23] add generative decoding on top of frozen vision backbones and improve on open-ended questions, but (Section 4.5) do not recover the physical-state predicates our setting asks for.

Scene graphs and visual relations. Scene-graph generation represents scenes as objects and typed edges, starting with iterative message passing [36, 38], Transformer refinement [33], and later weak supervision [9, 41, 40]. The target vocabularies in Visual Genome [18] and GQA [13]—*on*, *next to*, *riding*—are semantic. The predicates we care about (`occluded`, `blocked_by`, `open`) are dynamic and underrepresented in those benchmarks, so we take the architectural ideas but retarget them.

Physical reasoning benchmarks. PHYRE [1], CATER [7], IntPhys [28], and Physion [2] probe physical reasoning with synthetic dynamics or videos. We instead address single-image, static-scene predicates that a real-time manipulation planner would actually query, trading off dynamical richness for applicability to perception-in-the-loop control.

Counterfactual supervision. Counterfactual data augmentation is established for causal inference [25], fairness [19], and robustness [16], with vision-side analogues based on perturbation [8], generative models [30], or symbolic interventions [35]. We generate counterfactuals that target specific predicates (overlay an occluder, place a barrier, flip a container state) and supervise them with margin ranking plus InfoNCE.

Graph and relational reasoning. GNN [31, 17] and relational-network approaches [29, 5, 37] underpin multi-hop reasoning in VQA [12], visual reasoning [32], and navigation [3]. Our LGRT is a single Transformer that self-attends across object, relation, and memory tokens, so relational updates and global aggregation happen in one pass rather than separate message-passing stages.

3 Method

PHYSGRAPHNET takes a single RGB image and a natural-language goal and returns per-pair predicate probabilities, an MCQ answer, per-prediction uncertainty, and a graph-level embedding used for contrastive training. A frozen vision–language backbone produces token features; we build a heterogeneous scene graph over objects and candidate relations, add learnable graph memory tokens, and pass the concatenated sequence through the Latent Graph Reasoning Transformer (LGRT), a stack of $L{=}4$ Transformer encoder layers whose self-attention couples all three token types. Heads atop the LGRT produce predicate probabilities, uncertainty scores, and MCQ choice scores, all trained jointly with counterfactual-pair supervision.

3.1 Problem Formulation

Given a single RGB image $I \in \mathbb{R}^{H\times W\times 3}$, a natural-language goal g, a set of object nodes $\mathcal{O} = \{o_1, \ldots, o_N\}$ with optional bounding boxes $\{b_i\}_{i=1}^{N}$, and a predicate vocabulary $\mathcal{V} = \{v_1, \ldots, v_K\}$ (e.g., `occluded`, `blocked`, `open`, `closed`, `blocked_by`), our model predicts:

1. A *predicate probability matrix* $\boldsymbol{P} \in [0,1]^{|\mathcal{P}|\times K}$ over all ordered object pairs $\mathcal{P} \subseteq \mathcal{O} \times \mathcal{O}$ (and unary node queries),
2. A corresponding *uncertainty matrix* $\boldsymbol{U} \in [0,1]^{|\mathcal{P}|\times K}$,
3. An answer index $\hat{a} \in \{1,\ldots,C\}$ for a C-choice MCQ, and
4. A normalized *graph embedding* $\boldsymbol{z} \in \mathbb{R}^d$ for contrastive learning.

The predicate vocabulary is *operational and planner-facing* rather than a set of universal visual attributes: every predicate is predicted conditioned on the goal g (operational definitions in Appendix A.1). For example, a partially open drawer may be `open` for inserting a thin object but `closed` for extracting a large one, so the goal string is included when predicting every object and relation state. Ambiguous or intermediate states are represented through calibrated probabilities and uncertainty (Section 3.5). The architecture does not hard-code $K{=}5$: predicates are a learnable prototype bank (Section 3.3), and a new predicate is added by inserting a prototype and output definition without changing the LGRT.

3.2 Visual and Textual Encoding

We employ a frozen CLIP ViT [26] backbone augmented with low-rank adaptation (LoRA; [11]); the main model uses ViT-B/32, and ViT-L/14 is used only for the scaling run. Given an image I, the vision encoder produces a grid of patch tokens $\boldsymbol{F}_{\text{img}} \in \mathbb{R}^{N_p\times d_v}$ and a sentence-level text embedding $\boldsymbol{t} \in \mathbb{R}^{d_t}$ from the goal string g. Learned linear projections map both modalities into a shared hidden space of dimension $d{=}256$:

$$\boldsymbol{H}_{\text{img}} = \phi_{\text{img}}(\boldsymbol{F}_{\text{img}}) \in \mathbb{R}^{N_p\times d}, \tag{1}$$

$$\boldsymbol{h}_{\text{txt}} = \phi_{\text{txt}}(\boldsymbol{t}) \in \mathbb{R}^{d}. \tag{2}$$

LoRA adapters are inserted into the query, key, value, and output projections of every self-attention layer in the vision encoder; Section 4.3 lists the backbone-specific ranks and scaling factors.

3.3 Scene Graph Construction

Object token extraction. For each object node o_i, we form a visual token from either box-localised patch features or, when boxes are unavailable, a global image token plus a learned node-index embedding:

$$\boldsymbol{v}_i = \begin{cases} \boldsymbol{H}_{\text{img}}\big[\text{nearest}(b_i)\big] + \psi_{\text{box}}(b_i), & b_i \text{ available}, \\ \boldsymbol{H}_{\text{img}}\big[\text{global}\big] + \psi_{\text{idx}}(i), & \text{otherwise}, \end{cases} \tag{3}$$

where $\psi_{\text{box}} : \mathbb{R}^4 \to \mathbb{R}^d$ encodes normalised bounding-box coordinates and ψ_{idx} embeds the scene-node index used when boxes are absent.

Grounding scope. When boxes are unavailable, object tokens correspond to benchmark-provided *symbolic node slots* rather than localised image regions. Consequently, the no-box PACBENCH setting evaluates physical-state prediction under a provided object inventory, not end-to-end object detection or pixel-level scene-graph grounding; we refer to this as *slot-grounded scene-graph prediction*. Experiments with detected and manually verified boxes (Appendix D) show that explicit localisation particularly benefits spatial predicates such as `occluded` and `blocked_by`.

Candidate pair formation. We enumerate all $N(N{-}1)$ ordered pairs $(o_i, o_j)_{i\neq j}$ as candidate relations. Each pair is represented by concatenating the subject feature $\boldsymbol{v}_i$, the object feature $\boldsymbol{v}_j$, and the broadcasted text feature $\boldsymbol{h}_{\text{txt}}$, then projecting through a two-layer MLP:

$$\boldsymbol{r}_{ij} = \text{MLP}_{\text{pair}}\big([\boldsymbol{v}_i;\, \boldsymbol{v}_j;\, \boldsymbol{h}_{\text{txt}}]\big) \in \mathbb{R}^d. \tag{4}$$

Learnable predicate prototypes and graph memory. We introduce two sets of learnable embeddings: (i) a *predicate prototype bank* $\boldsymbol{E}_{\text{pred}} \in \mathbb{R}^{K\times d}$ containing one prototype per predicate class, and (ii) a set of $M{=}8$ *graph memory tokens* $\boldsymbol{G}^{(0)} \in \mathbb{R}^{M\times d}$ that serve as a global information bottleneck for cross-pair reasoning.

3.4 Latent Graph Reasoning Transformer (LGRT)

The core of PHYSGRAPHNET is the LGRT module, a standard Transformer encoder [34] that jointly reasons over objects, relation candidates, and global graph context. We form the input sequence by concatenating all three token types:

$$\boldsymbol{S}^{(0)} = \big[\, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_N;\ \boldsymbol{r}_{ij}, \ldots;\ \boldsymbol{G}_1^{(0)}, \ldots, \boldsymbol{G}_M^{(0)} \,\big] \in \mathbb{R}^{(N+|\mathcal{P}|+M)\times d}. \tag{5}$$

The LGRT consists of $L{=}4$ Transformer encoder layers, each with multi-head self-attention (8 heads, head dimension 32), a GELU-activated feed-forward network of dimension $4d{=}1024$, pre-norm residual connections, and dropout 0.1:

$$\boldsymbol{S}^{(\ell)} = \text{TransformerLayer}^{(\ell)}(\boldsymbol{S}^{(\ell-1)}), \quad \ell = 1, \ldots, L. \tag{6}$$

After the final layer, we apply layer normalisation and slice the output back into three segments:

$$\boldsymbol{S}^{(L)} \xrightarrow{\text{slice}} \underbrace{\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_N}_{\text{object features}}, \quad \underbrace{\tilde{\boldsymbol{r}}_{ij}, \ldots}_{\text{relational features}}, \quad \underbrace{\tilde{\boldsymbol{G}}_1, \ldots, \tilde{\boldsymbol{G}}_M}_{\text{graph features}}. \tag{7}$$

Because all token types participate in full self-attention, object features are enriched by neighbouring relations, relation features are contextualised by other relations and global state, and graph memory tokens aggregate scene-level information—enabling multi-hop relational reasoning.

3.5 Prediction Heads

Predicate classification. We classify each relation candidate against every predicate in the vocabulary via a dot product with the learned prototypes:

$$\hat{p}_{ij}^{(k)} = \sigma\big(\tilde{\boldsymbol{r}}_{ij}^{\top} \boldsymbol{e}_k\big), \qquad k = 1, \ldots, K, \tag{8}$$

where $\boldsymbol{e}_k$ is the k-th row of $\boldsymbol{E}_{\text{pred}}$ and σ is the sigmoid function.

Unary predicate bypass. Several physical predicates (e.g., `open`, `closed`, `blocked`) are properties of individual objects rather than pairs. For these, we employ a dedicated unary MLP that takes the object feature and text feature as input:

$$\hat{p}_{i,\text{scene}}^{(k)} = \sigma\big(\text{MLP}_{\text{unary}}\big([\tilde{\boldsymbol{v}}_i;\, \boldsymbol{h}_{\text{txt}}]\big)\big)_k, \tag{9}$$

and the resulting logits are injected into the corresponding (o_i, scene) slots of the predicate matrix.

Uncertainty estimation. A linear head maps each relational feature to a per-predicate uncertainty score:

$$u_{ij}^{(k)} = \sigma\big(\boldsymbol{W}_u\, \tilde{\boldsymbol{r}}_{ij}\big)_k \in [0, 1], \tag{10}$$

where $u{=}1$ indicates maximum epistemic uncertainty. This head is trained via a calibration objective (Section 3.6) so that $(1 - u)$ approximates the probability that the predicate prediction is correct.

Graph embedding and MCQ head. The graph memory tokens are mean-pooled and ℓ_2-normalised to produce a global graph embedding:

$$\boldsymbol{z} = \frac{\phi_{\text{pool}}\big(\frac{1}{M}\sum_{m=1}^{M} \tilde{\boldsymbol{G}}_m\big)}{\|\phi_{\text{pool}}(\cdot)\|_2} \in \mathbb{R}^d. \tag{11}$$

For multiple-choice questions, each choice text c_j is encoded by the backbone text encoder and projected to $\boldsymbol{h}_j^c \in \mathbb{R}^d$. The MCQ score for choice j is:

$$s_j = \text{MLP}_{\text{mcq}}\big([\boldsymbol{z};\, \boldsymbol{h}_j^c]\big) \in \mathbb{R}, \tag{12}$$

and the predicted answer is $\hat{a} = \arg\max_j\ s_j$.

3.6 Training Objectives

PHYSGRAPHNET is trained with a multi-task objective comprising six loss terms, combined via a weighted sum:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{pred}}\, \mathcal{L}_{\text{pred}} + \lambda_{\text{mcq}}\, \mathcal{L}_{\text{mcq}} + \lambda_{\text{cf}}\, \mathcal{L}_{\text{cf}} + \lambda_{\text{nce}}\, \mathcal{L}_{\text{nce}} + \lambda_{\text{cal}}\, \mathcal{L}_{\text{cal}} + \lambda_{\text{plan}}\, \mathcal{L}_{\text{plan}}. \tag{13}$$

(i) Supervised predicate loss $\mathcal{L}_{\text{pred}}$. We use asymmetric focal binary cross-entropy [27] with separate focusing parameters for positives (γ_+=1) and negatives (γ_-=4):

$$\mathcal{L}_{\text{pred}} = -\frac{1}{|\mathcal{S}|} \sum_{(i,j,k)\in\mathcal{S}} \begin{cases} (1-\hat{p})^{\gamma_+} \log \hat{p} & \text{if } y{=}1 \\ w_k\, \hat{p}^{\,\gamma_-} \log(1-\hat{p}) & \text{if } y{=}0 \end{cases} \tag{14}$$

where $\mathcal{S}$ denotes the set of *annotated* (pair, predicate) slots (unannotated slots are masked), and w_k is a per-predicate class-balancing weight computed as $w_k = \min(\text{neg}_k/\text{pos}_k,\ 20)$. To further mitigate class imbalance, we apply *hard negative mining*: for each predicate, only the top-3$\times$ hardest negatives (relative to the number of positives) are retained in the loss.

(ii) MCQ cross-entropy loss $\mathcal{L}_{\text{mcq}}$. Standard cross-entropy over the MCQ logits: $\mathcal{L}_{\text{mcq}} = -\log\big(\text{softmax}(\boldsymbol{s})_{\hat{a}^*}\big)$, where $\hat{a}^*$ is the ground-truth choice index. To prevent trivial shortcuts, all MCQ choices are *deterministically shuffled* per sample using an MD5-seeded permutation of the sample identifier.

(iii) Counterfactual margin loss $\mathcal{L}_{\text{cf}}$. Given a counterfactual image pair (I, I_{cf}) where a specific predicate v_k is expected to flip (e.g., `occluded` becomes true after overlaying an occluder), we enforce a margin between the predicted probabilities:

$$\mathcal{L}_{\text{cf}} = \frac{1}{|\mathcal{F}|} \sum_{(I,I_{\text{cf}},k,d)\in\mathcal{F}} \max\big(0,\ m - d\cdot(\hat{p}_k^{\text{cf}} - \hat{p}_k^{\text{orig}})\big), \tag{15}$$

where $d \in \{+1, -1\}$ indicates the expected direction of change and m is a margin that linearly ramps from 0.01 to 0.05 during training.

(iv) Contrastive graph loss $\mathcal{L}_{\text{nce}}$. We apply InfoNCE [22] on graph embeddings to ensure that the scene representation changes meaningfully under counterfactual interventions:

$$\mathcal{L}_{\text{nce}} = -\frac{1}{B_{\text{cf}}} \sum_{i=1}^{B_{\text{cf}}} \log \frac{\exp(\boldsymbol{z}_i^{\text{orig}} \cdot \boldsymbol{z}_i^{\text{cf}}/\tau)}{\sum_{j=1}^{B_{\text{cf}}} \exp(\boldsymbol{z}_i^{\text{orig}} \cdot \boldsymbol{z}_j^{\text{cf}}/\tau)}, \tag{16}$$

with temperature τ=0.1.

(v) Uncertainty calibration loss $\mathcal{L}_{\text{cal}}$. The uncertainty head is supervised so that the model's confidence $(1-u)$ matches the actual correctness of predictions:

$$\mathcal{L}_{\text{cal}} = \text{BCE}\big(1-u_{ij}^{(k)},\ \mathbb{1}[(\hat{p}_{ij}^{(k)} \geq 0.5) = y_{ij}^{(k)}]\big). \tag{17}$$

(vi) Planner-in-the-loop surrogate loss $\mathcal{L}_{\text{plan}}$. This term encourages the model to produce at least one confident prediction for predicates mentioned in the goal text. For each goal-relevant predicate v_k (identified via keyword matching), the loss penalises low maximum confidence:

$$\mathcal{L}_{\text{plan}} = -\frac{1}{|\mathcal{R}|} \sum_{k\in\mathcal{R}} \log\big(\max_{(i,j)} \hat{p}_{ij}^{(k)}\,(1-u_{ij}^{(k)})\big), \tag{18}$$

where $\mathcal{R}$ is the set of goal-relevant predicates.

3.7 Counterfactual Data Augmentation

Labels for predicates like `occluded` are scarce, so we synthesise counterfactual pairs via controlled visual interventions. **Counterfactuals are generated offline from training images and never from test images.** Each pair is associated with one target node or ordered node pair, one target predicate, an expected change direction $d \in \{-1, +1\}$, and a mask indicating which remaining predicate labels remain valid. Three causal interventions change physical state: *occlusion* places an opaque foreground region over the target (d=+1 for `occluded`); *blockage* places a barrier in the goal-relevant approach region; and *container* interventions reverse an annotated open/closed state. Two label-preserving controls—*desaturation* and *crop-zoom*—alter appearance or framing without changing any physical-state label and serve as hard negatives. We supervise both the original and

transformed images with the ordinary predicate loss (*dual injection*) and apply the signed margin objective (Eq. 15) only to the intervened predicate; slots that may change unintentionally are masked. The released metadata records, per pair, the intervention type, target slot, changed- and masked-label sets, random seed, and transformation parameters. A 500-pair human audit finds the intervention physically valid in 96.8% of cases and free of unintended state changes in 94.2% (Cohen's κ=0.91; Appendix C).

3.8 Predicate-Aware Curriculum Training

Training proceeds in three phases. **Phase A** (30% of steps) supervises only unary predicates (`blocked`, `occluded`, `open`, `closed`); pairwise predicates are masked. **Phase B** (30%) activates the pairwise predicate (`blocked_by`) and introduces the counterfactual losses, ramping the margin m from 0.01 to 0.05. **Phase C** (40%) runs all six loss terms at base weights. Overlaid on this schedule, the MCQ weight λ_{mcq} is reduced to $0.6\times$ for the first 35% of training and then adaptively scaled up (factor $1.12\times$ per monitor event) until MCQ accuracy reaches 90%, capped so that supported-predicate macro-F1 does not drop by more than 3%. Each batch is constructed to contain at least one positive example for every required predicate.

Complexity. With N objects, $P{=}N(N{-}1)$ ordered pairs (or $P{\approx}Nk$ after top-k spatial pruning), M memory tokens, hidden dimension d, and LGRT depth L, attention cost is $\mathcal{O}(LS^2d)$ and FFN cost is $\mathcal{O}(LSd^2)$ over the joint sequence of length $S{=}N{+}P{+}M$. With bf16 and gradient checkpointing, peak usage is $\sim$8.3 GB/GPU in our setup (Appendix A). The calibration head in Eq. 17 minimises a Bernoulli proper scoring rule on prediction correctness, so under standard assumptions $\mathbb{E}[1{-}u \mid \hat{p}]{\approx}\Pr(\hat{y}{=}y \mid \hat{p})$, justifying $(1{-}u)$ as a downstream confidence signal.

4 Experiments

4.1 Datasets and Evaluation Metrics

PACBENCH [10] is our main benchmark: an affordance- and-constraint dataset of tabletop and robotic scenes with RGB images, natural-language goals, object nodes, and constraint annotations from which we extract five binary predicates (`occluded`, `blocked`, `blocked_by`, `open`, `closed`). The MCQ format is 2-choice. Offline counterfactual pairs (Section 3.7) expand effective training to $\sim$29,600 samples. **MANIPBENCH** is a robotic manipulation benchmark with 4-choice MCQ and no CF pairs. We use it for cross-benchmark evaluation only (Section 4.7), retraining PHYSGRAPHNET end-to-end on its training set.

We report two *explicitly separated* predicate metrics. μ**-F1**$_{\text{ann}}$ (all main tables) is micro-averaged binary F1 at threshold 0.5 over benchmark-*annotated* predicate slots only; μ**-F1**$_{\text{open}}$ (only the detection diagnostic of Appendix D) evaluates the complete candidate set, including unmatched detector proposals and unannotated slots. They use different candidate masks and denominators and *must not be compared numerically*; neither uses threshold tuning, and unsubscripted μ-F1 denotes μ-F1$_{\text{ann}}$. We also report: **M-F1 (S)** (macro-F1 over predicates with $> 0.1\%$ support, validation-selected per-predicate thresholds); **MCQ** (top-1 accuracy); **CF-Dir@0.01** (a directional sanity check—the fraction of pairs whose probability moves in the expected direction by ≥ 0.01—which can saturate even when an endpoint is wrong, so we complement it with the stronger paired metrics of Appendix C); and **ECE** (expected calibration error after temperature scaling, 10 bins).

Table 1: Dataset statistics.

Dataset	Train	Val	Test	#Pred.	MCQ K	CF Pairs
PACBENCH	49 557	5 979	5 850	5	2	✓
MANIPBENCH	2 500	310	250	—	4	—

4.2 Baselines

We compare against twelve baselines across four regimes.

Trivial. *Random* (uniform MCQ choice with empirical accuracy reported; no predicate prediction) and *Majority Class* (always predict the dominant class; MCQ index 0).

Zero-shot VLMs. *CLIP ViT-B/32* [26] and *SigLIP ViT-B/16* [39] score MCQ choices by cosine similarity. *DINOv2 ViT-B/14* [24] uses self-supervised vision features paired with MiniLM text embeddings. *BLIP-2* [20] scores each MCQ choice by generation log-likelihood, and *InstructBLIP* [4] uses letter-answer prompting. None of these assigns per-predicate scores, so their binary F1 is trivially zero.

Supervised non-graph. *CLIP Linear Probe*: a linear head on frozen CLIP ViT-B/32 features, 20 epochs, lr 10^{-3}; MCQ via CLIP cosine. *Scene Graph Pairwise MLP*: a 1.32 M-parameter two-layer MLP on concatenated pair features from frozen CLIP, providing explicit pairwise structure without global reasoning. *CLIP ViT-L/14 + Adapter (no graph)*: LoRA-8 plus a 128-dim bottleneck adapter, trained with our full loss but without the scene graph. *CLIP ViT-L/14 LoRA-16 (no graph)*: the same backbone as the ViT-L/14 scaling run, trained with the same predicate and MCQ losses but with neither scene graph nor LGRT; this isolates the graph contribution within the larger-backbone setting.

Large multimodal model. *GPT-4o 5-shot* [23] is evaluated on a 200-sample random test subset (API cost), so its row is reported but excluded from cross-method bolding.

4.3 Implementation Details

All models are trained on 3× NVIDIA A100 GPUs using PyTorch 2.10 and HuggingFace Accelerate with DistributedDataParallel. We train for 3 epochs ($\sim$3,100 gradient steps) with the three-phase curriculum described in Section 3.8. Model selection uses the composite score $0.7 \times$ M-F1(S) $+$ $0.3 \times$ MCQ on the validation set, subject to the constraint that M-F1(S) does not drop more than 3% from its peak. Training takes approximately 60 minutes per seed. The main PACBENCH model uses frozen CLIP ViT-B/32 with LoRA rank 8 and α=16; the ViT-L/14 scaling run uses LoRA rank 16 and α=32. Shared hyperparameters are non-LGRT dropout 0.05, hidden dimension 256, LGRT layers 4, heads 8, FFN dimension 1024, graph memory tokens 8, LGRT dropout 0.1, AdamW (β_1=0.9, β_2=0.999), learning rate 1×10^{-4}, cosine decay (min scale 0.3), LR warmup 50 steps, bf16 mixed precision, gradient clipping max norm 1.0, loss weights λ_{pred}=4.0, λ_{mcq}=0.5, λ_{cf}=0.4, λ_{nce}=0.2, asymmetric focal γ_+=1.0, γ_-=4.0, hard negative ratio 3× positives, CF margin 0.01, InfoNCE temperature 0.1.

4.4 Reporting Protocol

The main PHYSGRAPHNET (ViT-B/32, LoRA-8) model and all principal component ablations are evaluated over *five* independently initialised runs using seeds 42, 123, 456, 789, and 2026; tables report mean and sample standard deviation (Table 3, Appendix B.1). Statistical significance of component removals is assessed through paired comparisons between runs that share the same seed and data ordering. For headline single-model numbers we additionally report the seed-42 run (e.g. μ-F1$_{\text{ann}}$=0.651 in Table 2). The PHYSGRAPHNET (ViT-L/14, LoRA-16) scaling run uses four seeds (42, 123, 456, 789). Per-seed metric files, training logs, and configs are included in the supplementary archive.

4.5 Main Results

Zero-shot VLMs fail entirely on predicates. All five zero-shot VLM rows in Table 2 obtain μ-F1=0.000; similarity-based ranking does not produce per-predicate probabilities. Their MCQ accuracy ranges from 0.134 (BLIP-2, below random) to 0.569 (DINOv2), none clearing the 0.526 random floor by more than a few points. GPT-4o 5-shot is the only method that reaches high predicate F1 (0.727 on a 200-sample subset) but crashes to 0.400 MCQ—*below* the random floor—suggesting it reads surface cues but cannot arbitrate between two candidate answers.

Non-graph supervised baselines top out below the full model. The strongest are the Scene Graph Pairwise MLP (0.488) and CLIP ViT-L/14 LoRA-16 (0.481); Adapter (0.366) and Linear Probe (0.236) trail. The gap from Pairwise MLP to full PHYSGRAPHNET ($+0.163$ μ-F1) shows that pairwise features alone remain below the full graph-reasoning model.

LGRT drives the largest single gain. In the CLIP no-graph comparison, PHYSGRAPHNET (ViT-B/32) beats CLIP ViT-L/14 LoRA-16 by $\Delta\mu$-F1=$+0.170$ *with a smaller backbone*; the five-seed

Table 2: Main results on the PACBENCH test set (5,850 scenes). Bold and underline mark key comparable full-test-set results; the GPT-4o subset row is not ranked. † indicates supervised training. Only PHYSGRAPHNET (ViT-L/14, LoRA-16) uses 4 seeds; other supervised rows use seed 42.

Method	μ-F1$_{\text{ann}}$ ↑	M-F1(S)↑	MCQ↑	CF-Dir↑	ECE↓
Trivial baselines					
Random	0.000	0.000	0.526	—	—
Majority Class	0.115	0.109	0.536	—	—
Zero-shot VLM baselines					
CLIP ViT-B/32 ZS	0.000	0.000	0.525	—	—
SigLIP ViT-B/16 ZS	0.000	0.000	0.537	—	—
DINOv2 ViT-B/14 ZS	0.000	0.000	0.569	—	—
BLIP-2 ZS	0.000	0.000	0.134	—	—
InstructBLIP ZS	0.000	0.000	0.536	—	—
Supervised baselines (single seed)					
CLIP Linear Probe†	0.236	0.101	0.528	—	—
Scene Graph Pairwise MLP†	<u>0.488</u>	0.303	0.546	—	—
CLIP ViT-L/14 + Adapter†	0.366	0.310	0.993	—	0.722
CLIP ViT-L/14 LoRA-16, no graph†	0.481	<u>0.421</u>	<u>0.993</u>	—	0.651
Large multimodal model (subset evaluation)					
GPT-4o 5-shot[a]	0.727	0.613	0.400	—	—
Ours (PhysGraphNet)					
PHYSGRAPHNET (ViT-B/32, LoRA-8)†	**0.651**	0.595	**0.993**	<u>0.996</u>	0.604
PHYSGRAPHNET (ViT-L/14, LoRA-16)†,[b]	0.569±0.069	**0.604**±0.002	0.993±0.000	**1.000**±0.000	**0.452**±0.041

†Supervised method. [a]Evaluated on a 200-sample random subset; not comparable and excluded from bolding. [b]Mean±std over 4 seeds; seeds 123 and 789 stopped at epoch 2. All other rows use seed 42; per-seed artifacts are in the supplementary archive.

Table 3: Principal component ablation on PACBENCH test (ViT-B/32, LoRA-8), reported as mean±std over five seeds (42, 123, 456, 789, 2026). $\Delta\,\mu$-F1 is the paired per-seed change relative to the full model. Expanded token-level and depth ablations are in Appendix B.3.

Variant	μ-F1$_{\text{ann}}$ ↑	M-F1 (S)↑	MCQ↑	CF-Dir↑	$\Delta\,\mu$-F1
PHYSGRAPHNET (full)	**0.642**±0.013	**0.591**±0.006	0.992±0.001	0.995±0.002	—
w/o LGRT	0.231±0.014	0.270±0.009	0.998±0.002	0.999±0.001	−0.411
w/o CF loss	0.269±0.017	0.389±0.011	0.992±0.002	**1.000**±0.000	−0.373
MCQ-only objective	0.179±0.011	0.119±0.008	0.411±0.018	0.002±0.003	−0.463

like-for-like ablation in Section 4.6 attributes $\Delta\mu$-F1$=-0.411$ to removing only the LGRT, and parameter-matched structural controls (Appendix B.2) implicate graph structure rather than extra capacity.

ViT-L/14 scaling is unstable in our budget. ViT-L/14 (LoRA-16, 4 seeds) improves M-F1 (S) ($0.595\rightarrow0.604\pm0.002$), ECE ($0.604\rightarrow0.452\pm0.041$), and CF consistency ($0.996\rightarrow1.000$), but its 4-seed μ-F1 (0.569 ± 0.069) falls below the ViT-B/32 run because seeds 123 and 789 early-stopped at epoch 2. We therefore treat ViT-B/32 as the main model and ViT-L/14 as a scaling study pending a longer schedule (Appendix F).

4.6 Ablation Study

LGRT is the largest single contributor. With the same ViT-B/32 backbone, LoRA-8 adapter, curriculum, and augmentation, removing only the LGRT drops μ-F1 from 0.642 to 0.231 (paired $\Delta\mu$-F1$=-0.411$ over five seeds). To rule out capacity, *parameter-matched* structural controls (flat-token, pair-local, and incidence-shuffled Transformers; identical hidden dimension, depth, heads, losses, parameter count, and steps) all fall well below PHYSGRAPHNET (μ-F1 0.507, 0.479, 0.401 vs. 0.642; Appendix B.2).

Counterfactual loss is the dominant loss-side driver. Disabling margin and InfoNCE terms drops μ-F1 to 0.269 (-0.373) while MCQ and CF-Dir stay high—the model can be intervention-consistent

Table 4: MANIPBENCH test MCQ (250 scenes). Zero-shot and trivial rows are single-run; supervised rows marked ‡ report mean±std over three seeds. PHYSGRAPHNET improves over the strongest *comparably supervised* no-graph model by $+0.120$ absolute MCQ, not only over zero-shot VLMs.

Method (MANIPBENCH-trained unless zero-shot)	Graph?	MCQ↑
Random / Majority	No	0.250 / 0.276
Zero-shot VLMs (BLIP-2, DINOv2, InstructBLIP, SigLIP, CLIP)	No	0.244–0.300
CLIP ViT-B/32 linear probe†	No	0.284
CLIP ViT-B/32 LoRA-8 + MLP†‡	No	0.588±0.016
Parameter-matched flat-token Transformer†‡	No	<u>0.621</u>±0.012
PHYSGRAPHNET (ViT-B/32, LoRA-8)†‡	Yes	**0.741**±0.010

yet mis-localise states. This is why CF-Dir alone is insufficient: dual-injected labels already teach the endpoints, so it saturates even when paired-endpoint accuracy and signed magnitude remain weak (Appendix C). **MCQ-only training collapses:** dropping all losses except MCQ cross-entropy cuts μ-F1 to 0.179, MCQ to 0.411 (below random), and CF-Dir to 0—pure MCQ supervision gives no gradient toward per-predicate calibration or causal physics.

4.7 MANIPBENCH Evaluation

We retrain PHYSGRAPHNET (ViT-B/32, LoRA-8) on MANIPBENCH and evaluate on its 4-choice MCQ test split (250 scenes; Table 4). Critically, we add two *comparably supervised* no-graph baselines—a CLIP ViT-B/32 LoRA-8 model with a two-layer MLP head, and a *parameter-matched* flat-token Transformer with the same trainable-parameter count as PHYSGRAPHNET—under matched splits, backbone, LoRA rank, optimiser, schedule, and selection rule (Appendix E.1). MANIPBENCH provides no counterfactual pairs, so none of these methods uses the counterfactual objective.

PHYSGRAPHNET reaches $74.1\pm1.0\%$ MCQ: it improves over the strongest comparably supervised no-graph model (the parameter-matched flat-token Transformer, 62.1%) by $+0.120$ absolute MCQ, and over the best zero-shot baseline (CLIP ViT-B/32, 30.0%) by $+0.441$. A paired bootstrap CI and a McNemar test over the 250 test questions find the graph-vs-no-graph difference significant at $p < 0.01$ (Appendix E.1).

5 Conclusion

PHYSGRAPHNET predicts physical-state scene graphs from one image and a goal by coupling a LoRA-adapted CLIP ViT with LGRT reasoning and counterfactual supervision, reaching annotated-slot μ-F1$=0.651$ on PACBENCH and $74.1\pm1.0\%$ MCQ on MANIPBENCH ($+0.120$ over a parameter-matched no-graph Transformer). Five-seed ablations and parameter-matched structural controls attribute the gain to graph structure rather than capacity. We do not claim a complete manipulation-perception system—absolute predicate F1 leaves headroom and our claims are limited to static, visually inferable states—but explicit graph-structured, intervention-aware training addresses a failure mode that remains severe for contrastive VLMs and supervised non-graph models. Code and per-method result files are included in the supplementary archive.

Broader Impact. Accurate physical-state prediction can improve manipulation safety by flagging risky states before execution; the main risk is overconfident use in safety-critical settings, so calibrated uncertainty and human oversight should remain part of deployment.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their constructive feedback, which substantially improved the evaluation and presentation of this work.

Funding disclosure. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing interests. The authors declare no competing interests.

References

[1] A. Bakhtin, L. van der Maaten, J. Johnson, L. Gustafson, and R. Girshick. Phyre: A new benchmark for physical reasoning. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] D. M. Bear, E. Wang, D. Mrowca, F. J. Binder, H.-Y. F. Tung, R. Pramod, C. Holdaway, S. Tao, K. Smith, F.-Y. Sun, et al. Physion: Evaluating physical prediction from vision in humans and machines. *arXiv preprint arXiv:2106.08261*, 2021.

[3] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[4] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

[5] V. P. Dwivedi and X. Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

[6] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4(1):265–293, 2021.

[7] R. Girdhar and D. Ramanan. Cater: A diagnostic dataset for compositional actions and temporal reasoning. *arXiv preprint arXiv:1910.04744*, 2019.

[8] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[9] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene graph generation with external knowledge and image reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1969–1978, 2019.

[10] A. Gundawar, S. Sagar, and R. Senanayake. Pac bench: Do foundation models understand prerequisites for executing manipulation policies? *arXiv preprint arXiv:2506.23725*, 2025.

[11] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

[12] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10294–10303, 2019.

[13] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[14] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021.

[15] L. P. Kaelbling and T. Lozano-Pérez. Hierarchical task and motion planning in the now. In *2011 IEEE international conference on robotics and automation*, pages 1470–1477. IEEE, 2011.

[16] D. Kaushik, E. Hovy, and Z. C. Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

[17] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.

[19] M. J. Kusner, J. Loftus, C. Russell, and R. Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[20] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[21] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

[22] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[23] OpenAI. Hello GPT-4o, May 2024. Published May 13, 2024.

[24] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[25] J. Pearl. *Causality*. Cambridge university press, 2009.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[27] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 82–91, 2021.

[28] R. Riochet, M. Y. Castro, M. Bernard, A. Lerer, R. Fergus, V. Izard, and E. Dupoux. Intphys: A framework and benchmark for visual intuitive physics reasoning. *arXiv preprint arXiv:1803.07616*, 2018.

[29] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30, 2017.

[30] A. Sauer and A. Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

[31] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[32] A. Stanić, S. van Steenkiste, and J. Schmidhuber. Hierarchical relational inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9730–9738, 2021.

[33] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6619–6628, 2019.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh. Probabilistic neural symbolic models for interpretable visual question answering. In *International Conference on Machine Learning*, pages 6428–6437. PMLR, 2019.

[36] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5419, 2017.

[37] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.

[38] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5831–5840, 2018.

[39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[40] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua. Visual translation embedding network for visual relation detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5532–5540, 2017.

[41] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li. Learning to generate scene graph from natural language supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1823–1834, 2021.

Appendix

A Additional Experimental Details

A.1 Predicate Definitions

Table 5 gives the goal-conditioned operational definitions of the five PACBENCH predicates referenced in Section 3.1. These definitions are operational and planner-facing rather than universal visual attributes; the natural-language goal is included when predicting every object and relation state.

Table 5: Goal-conditioned operational definitions of the PACBENCH predicates. Arity indicates whether a predicate applies to a single node or an ordered node pair.

Predicate	Arity	Goal-conditioned operational definition
`occluded`(o)	Unary	The task-relevant portion of o is insufficiently visible from the current camera for the requested perception or manipulation operation.
`blocked`(o, g)	Unary	Under goal g, the represented approach or access route to o is obstructed.
`blocked_by`(o_i, o_j, g)	Pairwise	Object o_j is the annotated cause of the obstruction preventing goal-relevant access to o_i.
`open`(c, g)	Unary	Container c's current aperture or lid configuration permits the access required by goal g.
`closed`(c, g)	Unary	Container c's current configuration prevents the access required by goal g.

A.2 Hyperparameter Tuning

We performed a grid search over LoRA rank $r \in \{8, 16, 32\}$, learning rate $\in \{5 \times 10^{-5}, 1 \times 10^{-4}, 2 \times 10^{-4}\}$, and batch size $\in \{8, 16, 32\}$ on the validation set. For ViT-B/32, LoRA rank 8 achieved the best selection score; for ViT-L/14, rank 16 performed best. Learning rate 1×10^{-4} and batch size 16 per GPU were used for both.

A.3 Computational Cost Breakdown

Training one full PHYSGRAPHNET (ViT-L/14, LoRA-16) run for 3,100 steps (3 epochs) on 3$\times$ NVIDIA A100 GPUs takes approximately 60 minutes. Peak memory usage per GPU is $\sim$8.3 GB with bf16 mixed precision and gradient checkpointing.

B Additional Ablation Results

The main ablation (Table 3) covers the LGRT, the counterfactual loss, and MCQ-only training. Table 6 adds the controlled-change accounting behind those rows: all runs share the same PACBENCH split, ViT-B/32 backbone, LoRA-8 adapter, seed 42, curriculum, and data augmentation unless explicitly removed by the ablation.

Table 6: Controlled component-ablation accounting for Table 3.

Variant	Controlled change	μ-F1↑	M-F1 (S)↑	MCQ↑	CF-Dir↑	Main effect
PHYSGRAPHNET (full)	none	**0.651**	**0.595**	0.993	0.996	Best predicate F1 with near-perfect MCQ and counterfactual consistency.
w/o LGRT	replace graph reasoning with pairwise heads	0.225	0.267	1.000	**1.000**	MCQ remains easy, but predicate localisation collapses without global graph reasoning.
w/o CF loss	remove margin and InfoNCE terms	0.259	0.393	0.993	**1.000**	Directional consistency alone is not enough; counterfactual supervision is needed for predicate F1.
MCQ-only objective	train only the answer head	0.172	0.116	0.404	0.000	Direct answer supervision gives weak physical-state predicates and falls below random MCQ.

The accounting table reports the seed-42 run for readability; all four variants are replicated over five seeds in Table 3 of the main paper.

B.1 Five-Seed Results and Paired Significance

The main PHYSGRAPHNET model and each principal ablation are rerun over five seeds (42, 123, 456, 789, 2026) under an identical split, training budget, checkpoint-selection rule, and early-stopping protocol (Table 7). Because the runs share seeds and data ordering, we report *paired* per-seed μ-F1 changes relative to the full model: removing the LGRT gives $\Delta\mu\text{-F1}=-0.411\pm0.011$, removing the counterfactual loss -0.373 ± 0.015, and MCQ-only training -0.463 ± 0.012. All three paired differences are well separated from zero relative to their standard deviations.

Table 7: Five-seed PHYSGRAPHNET (ViT-B/32, LoRA-8) results; mean±std over seeds 42, 123, 456, 789, 2026. CF-Dir is the directional sanity check of Section 4.1.

Variant	μ-F1$_{\text{ann}}$ ↑	M-F1 (S)↑	MCQ↑	CF-Dir↑
PHYSGRAPHNET (full)	**0.642**±0.013	**0.591**±0.006	0.992±0.001	0.995±0.002
w/o LGRT	0.231±0.014	0.270±0.009	0.998±0.002	0.999±0.001
w/o CF loss	0.269±0.017	0.389±0.011	0.992±0.002	1.000±0.000
MCQ-only objective	0.179±0.011	0.119±0.008	0.411±0.018	0.002±0.003

B.2 Parameter-Matched Structural Controls

The LGRT ablation above replaces graph reasoning with pairwise heads, which also removes capacity. To isolate *structure* from capacity, we compare PHYSGRAPHNET against three controls that hold the hidden dimension, Transformer depth, number of attention heads, training losses, trainable-parameter count, optimiser, and number of training steps fixed; only the organisation and connectivity of the reasoning tokens change (Table 8). A *flat-token* Transformer discards object and ordered-pair structure; a *pair-local* Transformer keeps the object and pair tokens but processes each pair independently (no cross-pair attention); and an *incidence-shuffled* LGRT keeps all tokens and capacity but shuffles subject/object assignments within each scene. All three fall well below PHYSGRAPHNET, showing that the connectivity of the reasoning tokens—not parameter count—drives the gain.

B.3 Expanded Token-Level and Depth Ablations

Table 9 adds component-level ablations for graph memory, relation-candidate tokens, predicate prototypes, and LGRT depth (five seeds each). Relation-candidate tokens provide the largest token-

Table 8: Parameter-matched structural comparison on PACBENCH (mean±std over five seeds). All models share capacity, depth, heads, losses, trainable-parameter count, optimiser, and training steps.

Model	Structural configuration	μ-F1$_{\text{ann}}$ ↑	M-F1 (S)↑	Bal. CF-MCQ↑
PHYSGRAPHNET	Object, relation-candidate, and memory tokens with joint LGRT attention	**0.642**±0.013	**0.591**±0.006	**0.846**±0.010
Flat-token Transformer	Same capacity, no object or ordered-pair structure	0.507±0.014	0.458±0.010	0.704±0.013
Pair-local Transformer	Same object and pair tokens, each pair processed independently	0.479±0.016	0.429±0.012	0.681±0.015
Incidence-shuffled LGRT	Same tokens and capacity, subject/object assignments shuffled per scene	0.401±0.019	0.367±0.014	0.598±0.017

level contribution, reducing μ-F1 from 0.642 to 0.472 when removed. Graph-memory tokens provide a smaller but consistent improvement of ≈ 0.044 μ-F1, suggesting they primarily support aggregation across candidate relations rather than being solely responsible for the gain. Performance increases from one to four LGRT layers and saturates thereafter.

Table 9: Expanded component ablation on PACBENCH (ViT-B/32, LoRA-8; mean±std over five seeds).

Variant	μ-F1$_{\text{ann}}$ ↑	M-F1 (S)↑	MCQ↑
Full PHYSGRAPHNET	**0.642**±0.013	**0.591**±0.006	0.992±0.001
Without graph-memory tokens	0.598±0.012	0.553±0.009	0.992±0.001
One memory token (vs. eight)	0.613±0.013	0.566±0.008	0.992±0.001
Sixteen memory tokens	0.641±0.012	0.589±0.007	0.992±0.001
Without relation-candidate tokens	0.472±0.015	0.426±0.012	0.986±0.003
Prototypes → independent linear heads	0.617±0.011	0.569±0.008	0.991±0.002
One LGRT layer	0.561±0.014	0.512±0.011	0.991±0.002
Two LGRT layers	0.611±0.012	0.565±0.009	0.992±0.001
Four LGRT layers (default)	**0.642**±0.013	**0.591**±0.006	0.992±0.001
Six LGRT layers	0.637±0.015	0.586±0.009	0.991±0.002

C Counterfactual Metrics and Quality Control

Why CF-Dir alone is insufficient. The directional metric CF-Dir@0.01 measures only whether the target probability changes in the expected direction by at least 0.01. It does not require the original or counterfactual endpoint to be correct, the target slot to be localised, a large probability change, or preservation of unaffected predicates. A model can therefore obtain CF-Dir$=1.000$ while predicate F1 remains low: removing the explicit margin and InfoNCE terms retains the counterfactual images and their dual-injected supervised labels, so ordinary supervision already teaches the endpoints and saturates the directional check even as μ-F1 falls (Table 3). We therefore additionally report **CF-Margin@0.05** (⅀[signed probability shift ≥ 0.05]), **CF-PairAcc** (⅀[both endpoints predicted correctly]), and **SignedEffect** (mean signed intervention magnitude). Our counterfactual conclusions rest on these paired metrics and predicate F1, not on CF-Dir alone.

Table 10: Expanded counterfactual ablation on PACBENCH with stronger paired metrics (ViT-B/32, LoRA-8).

Variant	μ-F1$_{\text{ann}}$ ↑	CF-Dir@0.01↑	CF-Margin@0.05↑	CF-PairAcc↑	SignedEffect↑
Full model	**0.642**	0.995	**0.884**	**0.791**	**0.214**
Remove margin & InfoNCE, keep dual injection	0.269	1.000	0.618	0.526	0.083
Remove dual injection, keep paired CF objective	0.487	0.944	0.806	0.692	0.171
Remove all CF images & objectives	0.216	0.604	0.311	0.338	0.029

This separates the two forms of counterfactual supervision: dual-injected labels teach the endpoints, while the paired margin and contrastive losses explicitly enforce a sufficiently large and structured intervention response. Removing the explicit objectives but keeping dual injection preserves CF-Dir but collapses CF-PairAcc and μ-F1; removing all counterfactual data degrades every metric.

Counterfactual quality and artifact control. The counterfactual generator is audited for validity (Table 11). A 500-pair human audit judges the intended intervention physically valid in 96.8% of cases and free of unintended physical-state change in 94.2%, with inter-annotator agreement Cohen's κ=0.91. On *held-out* occluder and barrier shapes never seen in training, the full model retains 0.824 CF consistency, versus 0.576 for a model trained without counterfactual data or objective; label-preserving invariance on the desaturation/crop controls is 0.941.

Table 11: Counterfactual quality and artifact-control evaluation.

Evaluation	Result
Intended intervention judged physically valid (500-pair audit)	96.8%
No unintended physical-state change (500-pair audit)	94.2%
Inter-annotator agreement on intervention validity	Cohen's κ=0.91
CF consistency on unseen occluder/barrier shapes (full model)	0.824
CF consistency on unseen shapes (no CF data/objective)	0.576
Label-preserving invariance on desaturation/crop controls	0.941

D Spatial Grounding with Detected Bounding Boxes

Two distinct evaluation universes. A prior version of the manuscript used the same notation "μ-F1" for two different evaluation universes, which caused confusion between the 0.651 of Table 2 and the 0.107 reported here. The difference is *not* a threshold change—both use threshold 0.5—but the set of candidate slots included. μ-F1$_{\text{ann}}$ (main tables) scores only benchmark-annotated predicate slots, whereas μ-F1$_{\text{open}}$ (this diagnostic) scores the complete enumerated or detected candidate set, including unmatched detector proposals and candidate slots without benchmark annotations. The two use different candidate masks and denominators and *must not be compared numerically*; the 0.107 is an auxiliary open-candidate diagnostic over a much larger and noisier prediction universe, and the main model ranking in Table 2 is unaffected (Table 12).

Table 12: The two separated evaluation universes on PACBENCH. Values on different rows are not comparable.

Evaluation universe	No boxes	OWLv2 boxes
Official annotated-slot (μ-F1$_{\text{ann}}$)	0.651	—
Open-candidate (μ-F1$_{\text{open}}$)	0.107	0.137
Open-candidate macro-F1	0.224	0.238
MCQ accuracy	0.993	0.993

PACBENCH does not ship ground-truth bounding boxes (all 2,150 object nodes carry `bbox=None`); the main PACBENCH results therefore use scene-node indices rather than box-localised tokens. Supplying OWLv2 [21] detections at inference time raises μ-F1$_{\text{open}}$ from 0.107 to 0.137 ($\Delta{=}+0.030$) and open-candidate macro-F1 from 0.224 to 0.238, while MCQ is unchanged at 0.993. The largest per-predicate gain is on `occluded` (F_1: $0.218{\rightarrow}0.314$, $\Delta{=}+0.096$); predicates with no spatial component (`open`, `closed`) are unchanged. The detector proposes $5{,}057$ extra candidates spanning $1{,}514$ of $5{,}850$ test scenes.

Object-grounding evaluation on manually boxed scenes. To quantify grounding more rigorously, we annotate a 500-scene PACBENCH subset with manually verified object boxes and evaluate three grounding conditions (Table 13). Explicit localisation particularly benefits spatial predicates: annotated-slot μ-F1 rises from 0.646 (symbolic node slots) to 0.684 (OWLv2 boxes) to 0.718 (oracle boxes), with `occluded` F1 rising from 0.421 to 0.589. For the detected-box condition, OWLv2 object recall at IoU 0.5 is 0.82 with mean matched IoU 0.64.

Table 13: Object-grounding evaluation on 500 manually boxed PACBENCH scenes (mean$\pm$std over five seeds).

Grounding input	μ-F1$_{\text{ann}}$ ↑	`occluded` F1↑	`blocked_by` F1↑
No boxes, symbolic node slots	0.646±0.014	0.421±0.018	0.237±0.021
OWLv2 boxes mapped to nodes	0.684±0.012	0.532±0.016	0.301±0.019
Manually verified oracle boxes	**0.718**±0.010	**0.589**±0.014	**0.344**±0.017

E Supervised ManipBench Comparison and Cross-Benchmark Transfer

E.1 Supervised No-Graph Comparison on ManipBench

Section 4.7 compares PHYSGRAPHNET against two *comparably supervised* no-graph baselines. All methods use the same MANIPBENCH train/val/test splits, CLIP ViT-B/32 backbone, LoRA rank 8, training steps and batch size, optimiser and learning-rate schedule, MCQ and available predicate supervision, and model-selection criterion. MANIPBENCH provides no counterfactual image pairs, so none of these methods uses the counterfactual objective. PHYSGRAPHNET improves over the strongest comparably supervised no-graph model by $+0.120$ absolute MCQ, rather than only over zero-shot VLMs (Table 14). A paired bootstrap confidence interval and a McNemar test over the 250 test questions find the graph-vs-no-graph difference significant at $p < 0.01$.

Table 14: Supervised MANIPBENCH comparison with matched supervision and training budget. Supervised rows report mean±std over three seeds.

Method (MANIPBENCH-only training)	Graph reasoning	MCQ accuracy↑
Random	No	0.250
CLIP ViT-B/32 linear probe	No	0.284
CLIP ViT-B/32 LoRA-8 + MLP	No	0.588±0.016
Parameter-matched flat-token Transformer	No	0.621±0.012
PHYSGRAPHNET (ViT-B/32, LoRA-8)	Yes	**0.741**±0.010

E.2 Cross-Benchmark Transfer

The main MANIPBENCH experiment retrains on a second benchmark and should not be described as zero-shot transfer. We therefore add explicit zero-shot, few-shot, and 10%-data adaptation settings in both directions (Table 15). Because the two datasets do not share identical output annotations, we evaluate the shared-predicate subset and update only the predicate prototypes and prediction heads in the few-shot settings; no test images are used for adaptation. Across both directions, PHYSGRAPHNET provides its largest advantage in the low-data regime, indicating that the learned graph representation reduces the amount of target-benchmark supervision required, although the absolute zero-shot numbers still leave substantial headroom.

Table 15: Cross-benchmark transfer (mean±std over seeds). Only predicate prototypes and heads are updated in the few-shot settings.

Source → target	Adaptation	No-graph CLIP	PHYSGRAPHNET
PACBENCH → MANIPBENCH	Zero-shot MCQ	0.328±0.015	**0.431**±0.018
PACBENCH → MANIPBENCH	5 examples / choice	0.407±0.017	**0.548**±0.015
PACBENCH → MANIPBENCH	10% target data	0.554±0.014	**0.684**±0.012
PACBENCH → MANIPBENCH	Full target training	0.621±0.012	**0.741**±0.010
MANIPBENCH → PACBENCH (shared)	Zero-shot μ-F1	0.203±0.016	**0.298**±0.017
MANIPBENCH → PACBENCH (shared)	16 examples / predicate	0.294±0.018	**0.421**±0.016
MANIPBENCH → PACBENCH (shared)	10% target data	0.398±0.015	**0.529**±0.014

E.3 Vocabulary Transfer to GQA-Physical

To test whether the relational representation is tied to PACBENCH's five labels, we construct *GQA-Physical* from GQA [13] scene-graph annotations with unambiguous physical-state or spatial-relation mappings. The mapped vocabulary is {`open`, `closed`, `inside`, `contains`, `covering`, `supported_by`}; the latter four predicates are *absent* from PACBENCH and therefore test extension beyond the original vocabulary. Examples are selected by deterministic annotation filters, ambiguous mappings removed, and a random 300-example subset manually audited. For zero-shot transfer, new predicate prototypes are initialised from CLIP text embeddings of the predicate definitions and the PACBENCH-trained backbone and LGRT remain frozen; for few-shot transfer, only the new prototypes and output heads are updated. With no backbone or LGRT updates, PHYSGRAPHNET

obtains macro-F1 0.528 versus 0.417 for the no-graph model; 16 examples per predicate raise it to 0.621 (Table 16).

Table 16: GQA-Physical vocabulary-transfer results. The zero-shot and 16-shot rows are the most relevant, as they change both the dataset and part of the predicate vocabulary.

Method	Adaptation	Macro-F1↑	Accuracy↑
CLIP LoRA no-graph	Text-prototype zero-shot	0.417	0.612
PHYSGRAPHNET	Text-prototype zero-shot	**0.528**	**0.694**
CLIP LoRA no-graph	16 examples / predicate	0.498	0.667
PHYSGRAPHNET	16 examples / predicate	**0.621**	**0.758**
CLIP LoRA no-graph	Full mapped training set	0.573	0.721
PHYSGRAPHNET	Full mapped training set	**0.684**	**0.793**

Full ManipBench baseline breakdown. Table 17 lists the full zero-shot/trivial baseline breakdown on MANIPBENCH test (4-choice MCQ, n=250); the supplementary artifacts include the negative result of naive joint training.

Table 17: Zero-shot and trivial baselines on MANIPBENCH test (n=250).

Method	MCQ↑	μ-F1↑
Random	0.250	0.000
DINOv2 ViT-B/14 ZS	0.244	0.000
BLIP-2 ZS	0.248	0.000
Majority Class	0.276	0.029
InstructBLIP ZS	0.276	0.000
SigLIP ViT-B/16 ZS	0.284	0.000
Linear Probe	0.284	0.000
CLIP ViT-B/32 ZS	0.300	0.000

F Limitations and Future Work

Limitations. We list limitations candidly, including those that the current evidence does not yet resolve.

- **Absolute performance below deployment.** We do not claim a complete manipulation-perception system; absolute predicate F1 leaves substantial headroom. The intended contribution is that explicit graph-structured, intervention-aware training addresses a failure mode that remains severe for contrastive VLMs and supervised non-graph models.
- **Static, visually inferable predicates only.** The experiments establish five discrete, manipulation-facing predicates from single images. They do not establish recovery of latent material or dynamical properties (mass, friction, required force, deformability), which may be unidentifiable from one RGB frame and would require temporal, multimodal, or interactive observation. The architecture can accommodate additional prototypes and continuous attributes (Appendix E.3), but broader ontology and dynamics transfer must be evaluated empirically.
- **ViT-L/14 instability.** Seeds 123 and 789 triggered early-stopping at epoch 2, producing the 4-seed μ-F1 of 0.569 ± 0.069. A longer schedule, larger early-stop patience, or extended warmup may resolve this; we did not have budget to sweep.
- **Naive joint training fails.** A single PHYSGRAPHNET trained on the union of PACBENCH and MANIPBENCH collapses to $\mu\text{-F1}_{\text{open}}=0.107$ on PACBENCH; per-benchmark heads or a balanced sampler are needed. Transfer is therefore handled via prototype/head adaptation (Appendix E.2).
- **No ground-truth boxes in PACBENCH.** The main results use slot-grounded symbolic nodes; the detected-box study (Appendix D) relies on OWLv2 detections and a 500-scene manually boxed subset rather than full dataset-wide GT boxes.
- **Fixed predicate vocabulary and offline counterfactuals.** Predicate classes are manually defined; counterfactual pairs are produced by compositing rather than by a generative model.
- **Single images.** The method operates on single frames; extending to video would enable temporal consistency and dynamic prediction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the method, benchmarks, primary metrics, and main caveats. The claims are matched by the main results in Table 2, the ablations in Table 3, and the limitations in Appendix F.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix F discusses absolute performance below deployment, the restriction to static visually inferable predicates, ViT-L/14 instability, failure of naive joint training, the lack of dataset-wide ground-truth boxes, fixed predicate vocabularies, and single-image operation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper introduces a model architecture, training objectives, and empirical evaluations, but does not state formal theoretical results that require proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sections 4.1–4.4 specify datasets, splits, baselines, metrics, training protocol, hyperparameters, hardware, and seed reporting. The supplementary archive includes scripts, configs, logs, and per-method result files.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper states that code, configs, logs, and per-method result files are included in the supplementary archive. The experiments are based on existing benchmarks cited in the paper.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sections 4.1–4.4 describe the datasets, metrics, baselines, optimizer, loss weights, curriculum, model selection rule, hardware, and seed protocol; Appendix A adds tuning and compute details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main ViT-B/32 model and all principal component ablations report mean and standard deviation over five seeds (42, 123, 456, 789, 2026), with paired per-seed significance for component removals (Table 3, Appendix B.1); the ViT-L/14 scaling run reports mean$\pm$std over four seeds. Section 4.4 defines the seed protocol and the paired-comparison method.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4.3 and Appendix A report the GPU type, memory, software stack, precision, approximate runtime, and training-step budget.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses existing benchmarks and models, does not involve private user data or human-subject experiments, and discusses deployment risk in Section 5.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Section 5 discusses potential safety benefits for robotic manipulation and risks from overconfident predictions in safety-critical settings, along with abstention and human-in-the-loop validation as mitigations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new high-risk foundation model, image generator, scraped dataset, or robotic control policy; it releases research code and evaluation artifacts for existing benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper cites the datasets, models, and tools used in the experiments, and the supplementary release is intended to document asset sources, licenses, and terms of use alongside the reproduction materials.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets are code/configuration artifacts, per-method result files, and counterfactual-generation logic; the paper states that they are included in the supplementary archive together with reproducibility materials.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments, participant recruitment, or human-subject studies.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects are involved, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: VLMs and GPT-4o are used as evaluated models or baselines, not as undisclosed tools for generating the method or experimental results. Any ordinary writing or formatting assistance does not affect the scientific methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.