Prior Bias in Vision Language Models on UML Diagram Interpretation

Anonymous ACL submission

Abstract

Vision Language Models (VLMs) are increasingly applied to software engineering artifacts, especially UML class diagrams whose meaning depends on visual notation. Yet, it is unclear whether VLMs actually read such diagrams or instead answer from pretrained priors about how classes typically relate. We introduce a controlled UML benchmark in which each prior-conforming diagram is paired with its prior-conflicting counterpart that (1) preserves the same class names and layout while (2) reverses only the relation arrow. We evaluate eight open-source VLMs from two model families, InternVL3.5 and Qwen3, alongside two closed-source frontier models GPT-5.4 and GPT-5.4 Mini. Across the eight open-source models, reversing the arrow reduces relation-direction accuracy by **33.48%** on average, while GPT-5.4 Mini retains a **10%** gap. In the harder three-class condition, accuracy drops sharply by **45.28%** for open-source models, and even **18.62%** for the GPT-5.4 family on average. Scaling provides only limited improvements and is family-dependent. Our benchmark presents a diagnostic prior-driven failure in diagram-grounded software understanding. Our artifact is available at `https://anonymous.4open.science/r/UMLKnowledgeConflict-8461`.

1 Introduction

Vision Language Models (VLMs) inherit a vast amount of prior knowledge about how objects are related from their LLM backbones. While this prior knowledge can be useful, it is also a known source of failure: when visual evidence contradicts what the model already "knows", generation often follows the prior rather than the image (Lin et al., 2023; Li et al., 2023; Guan et al., 2024; Golovanevsky et al., 2025; Vo et al., 2025; Conrardy and Cabot, 2024; Antal et al., 2024).

Prior evidence (Guan et al., 2024; Liu et al., 2025; Lee et al., 2025; Vo et al., 2025) that VLMs follow learned priors over visual evidence comes almost entirely from natural-image counterfactuals. Yet, VLMs are increasingly deployed in software-engineering pipelines that require reading visual artifacts whose meaning is determined solely by visual notation (Zou et al., 2024; Antal et al., 2024; Hou et al., 2024). It remains unclear how VLMs comprehend such diagrams, specifically how strongly their knowledge priors interfere when the diagram encodes a relation contradicting what the named entities encode in training. We aim to assess **whether VLMs interpret formal software diagrams from the visual notation or from the knowledge prior of the named entities**. For example, given a UML class diagram with an inheritance arrow from `Mammal` to `Elephant` (Fig. 1), we ask "Does `Elephant` inherit from `Mammal`?", the correct answer is "No" because the arrow points the other way, but VLMs systematically answer "Yes" by following the knowledge prior.

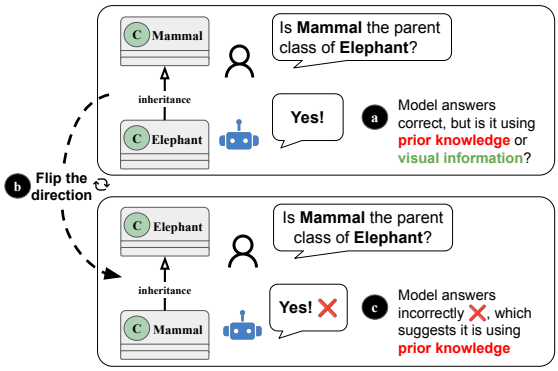

Figure 1: **VLMs follow the class names, not the arrow.** (a) On a prior-conforming diagram, VLMs correctly answer *"Is Mammal the parent class of Elephant?"* with "Yes." (b) Flipping the arrow makes "No" the correct answer. (c) VLMs still answer "Yes", following the prior knowledge rather than the depicted arrow.

To study this question, we narrow our focus specifically to UML diagrams, which expose this fragility cleanly. A UML class diagram is a *formal specification* whose meaning is fixed by **visual notation**, not by what the classes are called (Rumpe, 2016). We focus on four class relations represented by arrows with a direction that flips the relation if reversed: *inheritance*, *composition*, *dependency*, and *aggregation*. The same two class names can

therefore encode either the prior-conforming or the prior-conflicting relation, depending only on which way the arrow points (Roberts et al., 2024).

To empirically study this failure mode, we introduce a controlled UML benchmark. Every prior-conflicting diagram is paired with a prior-conforming counterpart built from the same class names, the same notation, and only a flipped direction (Fig. 1) (Wu et al., 2025). A further *prior-free* setting replaces meaningful class names with opaque labels, separating knowledge priors from visual grounding (Li et al., 2024a). The benchmark covers the four direction-sensitive UML relations (inheritance, aggregation, composition, and dependency) and evaluates 8 open-source model sizes across two VLM families, InternVL3.5 (Wang et al., 2025) and Qwen3 (Bai et al., 2025), plus two closed-source frontier models, GPT-5.4 and GPT-5.4 Mini. Our key findings are:

1. **Knowledge prior dominates visual evidence in open-source VLMs, while closed-source models reduce but do not eliminate errors on harder settings.** Reversing the arrow collapses relation accuracy by **33.48% on average** across the two open-source model families, with InternVL3.5 dropping more accuracy (-40.52 %) than Qwen3 (-26.44 %); GPT-5.4 Mini reduces the two-class drop to -10.00%, and GPT-5.4 nearly closes it ($\Delta \approx 0$). On harder three-class variants, accuracy even drops by **45.28%** for open-source models, and **18.62%** for the GPT-5.4 family (Section 4.1).

2. **Adding more visual context exacerbates the prior-over-vision failure mode.** Specifically, adding one auxiliary edge to the prior-conflicting UML diagrams further reduces accuracy by $\sim$12% on average. Adding more details increases visual complexity, leading to worse model performance (Section 4.1).

3. **The prior-over-vision effect is robust across image scales, and is focused on Aggregation and Inheritance.** On three image scales of $1\times$, $1.5\times$, $2\times$, the collapse is consistent, confirming it is not an artifact of a perceptual bottleneck. On four tested relations, models exhibit the prior-over-vision effect clearly on Aggregation and Inheritance (*e.g.,* $\sim 50\%$ accuracy loss when flipping the arrow) (Section 4.2).

4. **The effect of model scaling is governed by model visual capabilities.** In Qwen3, increasing the model size helps on two-class prior-conflicting UML diagrams (+58.55% from 2B to 32B). On the same subset, InternVL3.5 with 2B parameters uniquely outperforms all larger variants, but this is due to the weak visual processing capabilities of InternVL3.5 (Section 4.3).

2 Related Work

2.1 UML Interpretation by Models

Earlier work on UML class-diagram understanding built dedicated extraction pipelines that recover structured semantics directly from images. Hammond and Davis (Hammond and Davis, 2006) introduced *Tahuti*, a sketch-based recognizer combining stroke-level parsing with interactive correction; Lank *et al.* (Lank et al., 2000) proposed pen-based UML interpretation with user-in-the-loop disambiguation; Karasneh and Chaudron (Karasneh and Chaudron, 2013a,b) presented *Img2UML* and released a large web-mined UML repository; and Chen *et al.* (Chen et al., 2022) learned semantic element recognition with *ReSECDI*. More recent work tests whether end-to-end multimodal models can read UML directly without such pipelines (De Bari et al., 2024): Antal *et al.* (Antal et al., 2024) report that GPT-4V handles simple class diagrams but degrades on complex relation structures; Naboichenko and Peinl (Naboichenko and Peinl, 2026) find that VLMs still struggle with relation-centric questions over UML class diagrams; Shehzadi *et al.* (Shehzadi et al., 2025) release a UML-focused VQA dataset; Wang *et al.* (Wang et al., 2026) investigate GPT-based grading of student UML submissions; and Campanello *et al.* (Campanello et al., 2025) analyze LLM-supported UML reverse-engineering. These studies establish that VLMs can read UML labels but report uneven relation accuracy. They do not, however, isolate whether the failure is perceptual or driven by pretrained priors about how the named classes typically relate.

2.2 Hallucination and Prior Bias in VLMs

A growing line of work documents that VLMs produce fluent-but-unfaithful answers, especially on images that violate common-sense expectations (Liu et al., 2024a). Early VQA studies showed that strong language priors can dominate image evidence (Goyal et al., 2017; Ramakrishnan et al., 2018; Wu et al., 2022); subsequent bench-

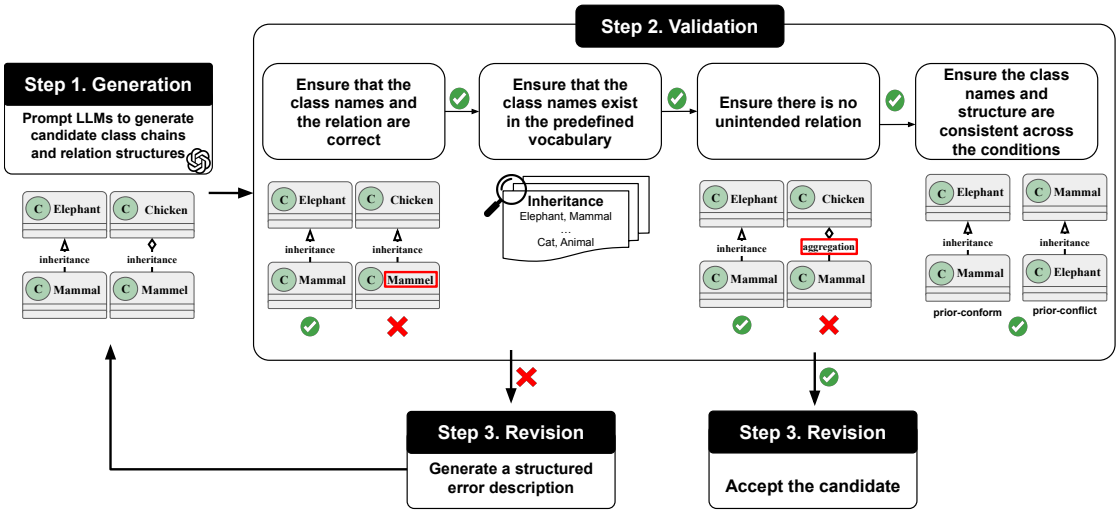

Figure 2: **Data generation pipeline.** (1) We prompt SOTA LLMs to generate candidate UML diagrams across the three conditions; (2) We then validate the diagrams against four constraints, ensuring consistency and correctness of these images; (3) If there is any error (✖), we generate a structured error description and redo step (1), else (✔) we accept the candidate.

marks measure this directly through object hallucination (Li et al., 2023; Huang et al., 2024), entangled language and visual illusion (Guan et al., 2024), and counter-commonsense images (Liu et al., 2025; Bitton-Guetta et al., 2023). These works share a methodological lesson: aggregate accuracy obscures the locus of failure, and disaggregating perception from inference is essential (Zhu et al., 2025). A more recent thread isolates the vision–knowledge conflict directly (Li et al., 2024b). *VLind-Bench* (Lee et al., 2025) measures how strongly language priors override visual input; *ViLP* (Luo et al., 2024) probes whether VLMs trust pixels over learned facts; and Liu *et al.* (Liu et al., 2024b) report that "insight over sight" bias persists across modern multimodal LLMs. Most recently, Vo *et al.* (Vo et al., 2025) demonstrate that current VLMs systematically follow commonsense priors over visual evidence on counterfactual images of natural objects. Our work extends this line to formal software diagrams, where the source of truth is notation rather than pixel-level photorealism, and where the prior conflict is constructible by reversing a single arrow rather than relying on naturally-occurring counterfactuals.

3 Benchmark and Protocol

3.1 Benchmark Construction

Our benchmark is designed to isolate prior-driven failure from generic task difficulty. For example, let "$\rightarrow$" denote the "inherits from" relation, every prior-conforming UML diagram (*e.g.,* `Dog` $\rightarrow$ `Animal`) is paired with a prior-conflicting counterpart (*e.g.,* `Animal` $\rightarrow$ `Dog`). The two diagrams in a pair share class names, layout, and queries; they differ only in the direction of the relationship arrow.

Instances. Each benchmark instance is a tuple $x = (I, r, (c_i, c_j), y)$, where I is the rendered UML class-diagram image, $r \in \{$*inheritance, aggregation, composition, dependency*$\}$ is the queried relation type, (c_i, c_j) is an ordered pair of class names appearing in I, and $y \in \{$True, False$\}$ is the gold label. The query has the form "In this diagram, does the relation r hold from c_i to c_j?", and the model must respond with one label from $\{$True, False, Unknown$\}$; the Unknown option lets the model abstain on genuinely unreadable diagrams rather than be forced to guess, but abstentions count as incorrect during scoring. The four relations are rendered in standard UML notation: *inheritance* as a hollow triangle pointing toward the parent class, *aggregation* as a hollow diamond, *composition* as a filled diamond, and *dependency* as a dashed arrow pointing toward the depended-on class.

Conditions. Our hypothesis is that VLMs follow the knowledge prior implied by the class names rather than the depicted relation arrow. Following this hypothesis, we design three conditions that contrast prior alignment against visual evidence while holding diagram structure (number of classes, edge type, layout) fixed:

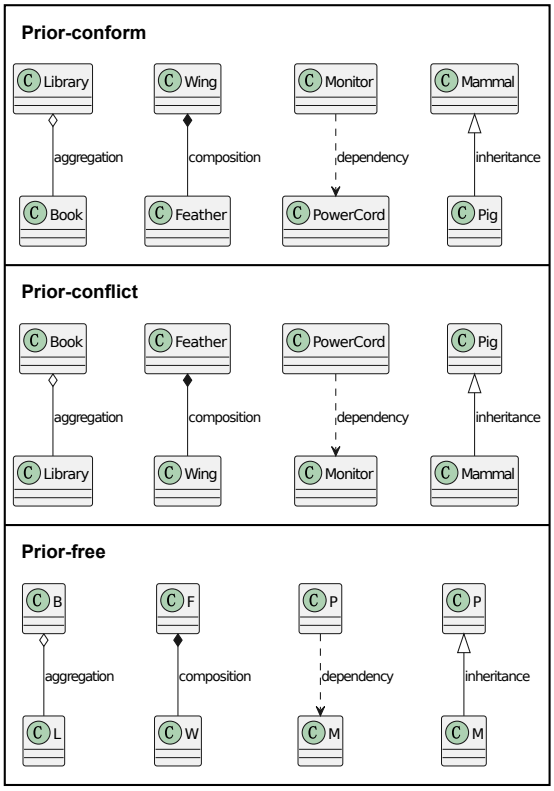

Figure 3: **Examples for each condition type.** We report only the *2-reverse* subset for the prior-conflict condition. Additional illustrations, including other prior-conflict subsets, are shown in Fig. 10.

1. **Prior-conforming.** The visual evidence and knowledge prior point to the same answer. We design the relation to follow canonical facts induced by the class names (*e.g.,* `Dog` → `Animal`).

2. **Prior-conflicting.** The visual evidence and the prior point to opposite answers. We use the same class names and relation type as the prior-conforming, but reverse the depicted relation so the image contradicts canonical expectations. To study knowledge conflict, we include (i) *2-reverse*, the standard two-class condition (*e.g.,* `Animal` → `Dog`), (ii) *3-reverse*, a three-class condition where all pairs are prior-conflicting (*e.g.,* `Animal` → `Mammal` → `Dog`), and (iii) *3-mixed*, a three-class condition where one pair is prior-conforming and the other is prior-conflicting (*e.g.,* `Animal` ← `Mammal` → `Dog`).

3. **Prior-free.** No knowledge prior is activated, and the model must use purely the visual information. We use the initial letters from the prior-conforming labels to construct these samples (*e.g.,* `D` → `A` instead of `Dog` → `Animal`).

The matched *prior-conforming* vs. *prior-conflicting* contrast measures *how much* a model's answer changes when visual evidence contradicts the knowledge prior. But the model may fail for two reasons: (1) bias toward the prior, or (2) inability to read the diagram in the first place. The *prior-free* condition disentangles these two failure modes: if the model fails on *prior-free* (where no prior knowledge is available), the failure on *prior-conflicting* is consistent with (2) and not bias (Fu et al., 2024).

Generation. We develop the benchmark with an LLM-assisted pipeline consisting of three steps illustrated in Fig. 2. We detail the steps as follows:

1. **Generate candidates.** High-capability LLMs are prompted to generate candidate class chains and relation structures consistent with the requested condition.

2. **Validate candidates.** Each candidate is validated against four constraints: (i) the queried relation r and ordered pair (c_i, c_j) exist in the candidate as specified; (ii) class names are drawn from a controlled vocabulary chosen so the prior is unambiguous; (iii) no other unintended relation of type r appears elsewhere in the diagram; and (iv) for matched variants, class names and edge structure agree with the matched counterpart.

3. **Revise candidates.** Candidates that fail any check are returned to the LLM with a structured error description and revised. The repair loop continues until every accepted candidate passes all four checks.

3.2 Evaluation Protocol

A UML failure can be of two kinds: a model may fail to read the class names at all, or read them perfectly yet still answer the relation question incorrectly. We separate these failure modes with two metrics, one for perception and one for inference.

Let $\mathcal{D}$ be the evaluation set, $C(x)$ the gold class-name set for instance x, $\hat{C}(x)$ the model's predicted class-name set, $\mathcal{N}$ a deterministic name-normalization function (case folding, punctuation cleanup, alias normalization), $y(x)$ the gold relation label, and $\hat{y}(x)$ the predicted relation label. We define the two metrics as follows:

Class-name recovery. Given only the diagram image, the model must output the complete set of class names appearing in the diagram (*e.g.,*

{Dog, Animal}). This metric measures perception: whether the model can read the text labels in the diagram and outputs the exact match (we abbreviate this as EM).

$$\mathrm{EM} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbf{1}\left[\mathcal{N}(\hat{C}(x)) = \mathcal{N}(C(x))\right]$$

Relation-direction accuracy. The model receives the same diagram image and a query about a specific ordered class pair (*e.g.,* "In this diagram, does Dog inherit from Animal?") and must respond with one label from {True, False, Unknown}. This metric measures whether the model can correctly ground the queried relation in the diagram's structure. We abbreviate this as *RelAcc*.

$$\mathrm{RelAcc} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbf{1}\left[\hat{y}(x) = y(x)\right]$$

Our key effect-size metric is the **conflict gap**, denoted Δ: the drop in accuracy when the depicted relation contradicts the knowledge prior:

$$\Delta = \mathrm{RelAcc}_{\textit{prior-conform}} - \mathrm{RelAcc}_{\textit{prior-conflict}}$$

A model unaffected by the conflict has $\Delta \approx 0$; a model fully dependent on the knowledge prior has Δ approaching its $\mathrm{RelAcc}_{\textit{prior-conform}}$. We report Δ both per-model and as a grand mean across models.

4 Experimental Results

We organize the experiments around three research questions. **Q1** establishes the headline prior-override effect (Section 4.1). **Q2** confirms that the effect is robust across the two axes built into the benchmark, relation type and image scale (Section 4.2). **Q3** studies whether scaling fixes the effect, and finds that results are dependent on visual capabilities and model family (Section 4.3).

4.1 Q1: Do VLMs follow class names rather than the arrow direction?

4.1.1 Motivation

Q1 asks whether VLMs interpret a UML diagram from the depicted arrow or from the knowledge prior of the named entities. Our hypothesis is that priors dominate: when the arrow contradicts the prior (*e.g.,* "Animal" →"Dog"), the model follows the prior rather than the arrow. We test this with the matched *prior-conform* / *prior-conflict* contrast (Fig. 1), then extend to three-class conditions to ask whether adding more diagram context recovers visual grounding.

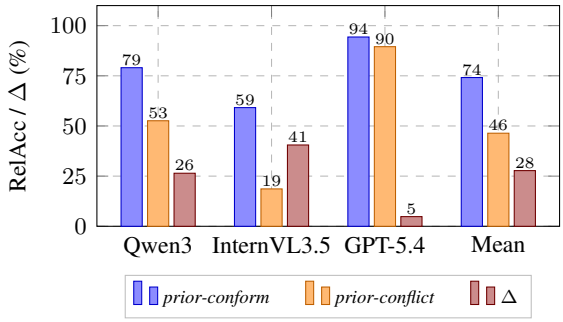

Figure 4: **Reversing the direction of a *prior-conform* diagram lowers accuracy by 27.75% on the all-model mean, but the drop is concentrated in the open-source families (Qwen3 $\Delta = 26.44\%$, InternVL3.5 $\Delta = 40.52\%$) while the GPT-5.4 family makes the two-class gap smaller ($\Delta = 4.82\%$).** Per-family RelAcc (%) on *prior-conform* (blue) and *prior-conflict* (orange), with the conflict gap Δ (red); each family bar is the simple mean over its members (4 for Qwen3 and InternVL3.5, 2 for GPT-5.4), and "Mean" is the simple mean over all 10 models.

4.1.2 Experiments

We report RelAcc on *prior-conform* and *prior-conflict* images, averaged across four relation types and three image sizes. Fig. 4 reports the per-family breakdown with the conflict gap Δ. We additionally compute the 10-model mean RelAcc on *3-reverse* and *3-mixed* (Fig. 5) to measure how added structural context affects the accuracy.

4.1.3 Results

VLMs are mostly correct on prior-conforming diagrams but fail when the arrow is reversed. Fig. 4 shows a sharp drop from *prior-conform* to *prior-conflict*. The 10-model mean RelAcc falls from 74.14% to 46.39%, a knowledge conflict gap of $\Delta = 27.75$ %; the drop is concentrated in the open-source families, whose 8-model mean shows a much larger gap of $\Delta = 33.48\%$. Both open-source families exhibit the same pattern: Qwen3 drops $79.04 \rightarrow 52.60$ ($\Delta = 26.44$); InternVL3.5 drops $59.15 \rightarrow 18.63$ ($\Delta = 40.52$). InternVL3.5's gap is 14% larger although the family starts lower on *prior-conform* (a 59.15% ceiling vs. Qwen's 79.04%). This illustrates that strong prior knowledge (*e.g.,* high $\mathrm{RelAcc}_{\textit{conform}}$) does not always lead to a more pronounced prior-over-vision effect.

Adding more classes to the prior-conflicting scenario further reduces model performance. Fig. 5 shows that augmenting the two-class prior-conflicting diagram with a third class further reduces accuracy. The 10-model mean RelAcc falls

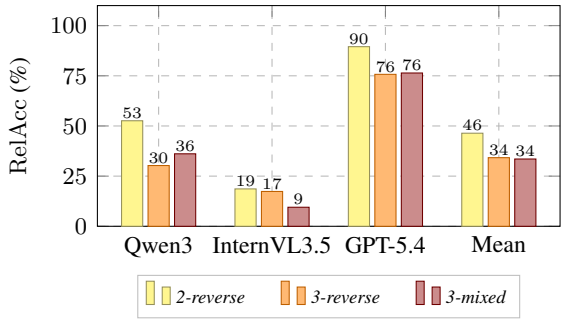

Figure 5: **Adding a prior-conflicting (*3-reverse*) or prior-conforming (*3-mixed*) auxiliary edge to *2-reverse* reduces relation-direction accuracy in every family; the GPT-5.4 family stays much higher overall but still loses ∼14 points from *2-reverse* to *3-reverse*.** Per-family and overall mean RelAcc (%) on *2-reverse* (yellow), *3-reverse* (orange), and *3-mixed* (red), averaged over 4 relations and 3 image scales; each family bar is the simple mean over its members (4 for Qwen3 and InternVL3.5, 2 for GPT-5.4), and "Mean" is the simple mean over all 10 models.

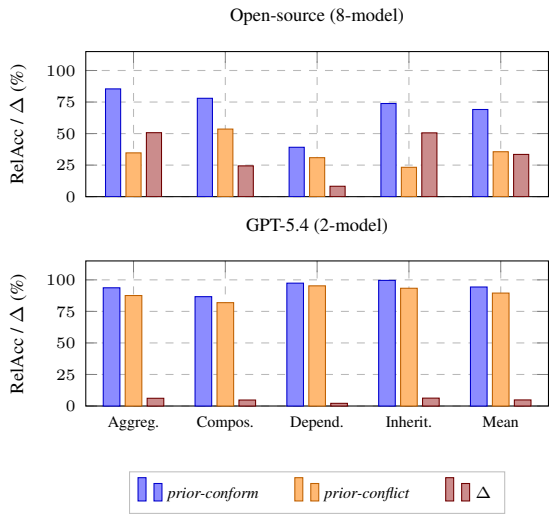

Figure 6: **The conflict gap is large and positive for all four UML relations in the open-source 8-model mean (top), with Aggregation and Inheritance largest; the GPT-5.4 family (bottom) keeps the gap below 7% on every relation.** Per-relation and overall mean RelAcc (%) on *prior-conform* (blue) and *prior-conflict* (orange), with the conflict gap Δ (red); "Mean" on each panel is the simple mean over the four relations.

from 46.39% on *2-reverse* to 34.20% on *3-reverse* (−12.19 %) and to 33.53% on *3-mixed* (−12.86 %). The two relation-direction accuracies of three-class conditions are almost indistinguishable, even though *3-mixed* adds an edge whose direction matches common semantic expectations and could in principle anchor the model toward a more visual interpretation. The family-level pattern, however, is asymmetric in the open-source families: Qwen3 scores higher on *3-mixed* (36.12%) than on *3-reverse* (30.28%), while InternVL3.5 shows the opposite (*3-mixed*'s 9.49% vs. *3-reverse*'s 17.36%); the two open-source families' patterns cancel out at the open-source 8-model mean.

Closed-source frontier models substantially reduce the conflict gap while the advantage weakens on harder structural variants. We report GPT-5.4 and GPT-5.4 Mini separately because their behavior departs sharply from the open-source pattern. GPT-5.4 Mini reduces the conflict gap to $\Delta = 10.00$ **%** ($\text{RelAcc}_{\textit{prior-conform}} = 97.95\%$, $\text{RelAcc}_{\textit{prior-conflict}} = 87.95\%$), about a third of the open-source 8-model mean (33.48%). GPT-5.4 nearly closes the gap ($\text{RelAcc}_{\textit{prior-conform}} = 90.73\%$, $\text{RelAcc}_{\textit{prior-conflict}} = 91.10\%$; $\Delta \approx 0$): unlike every open-source model, it does not lose accuracy when the arrow is reversed. The failure persists, however, on the harder three-class conditions: GPT-5.4 Mini drops to 65.41% on *3-reverse* (−32.55 from *prior-conform*), and even GPT-5.4 drops to 86.05% (−4.68). Frontier capacity, therefore, largely overcomes the prior-over-vision failure on the clean two-class contrast, but the bias re-emerges once auxiliary structural context is added.

4.2 Q2: Does the prior-over-vision effect persist across relations and image scales?

4.2.1 Motivation

Q1 showed that the prior-override effect is large in aggregate. **Q2** asks whether it is uniform along the two axes built into our benchmark: (1) the relation type, and (2) the image size of the diagram. Specifically, if larger images shrank the gap, the failure could be re-attributed to a perceptual bottleneck rather than a post-reading bias. We rule out both.

4.2.2 Experiments

For each of the four relation types (Aggregation, Composition, Dependency, Inheritance), we report the 8-model mean of RelAcc on *prior-conform* and *prior-conflict* images, averaged across three image scales, with the per-relation conflict gap Δ (Fig. 6). Separately, for each image scale (1×, 1.5×, 2×), we report the same quantities averaged across all eight models and four relations, and compare Δ across scales (Fig. 7).

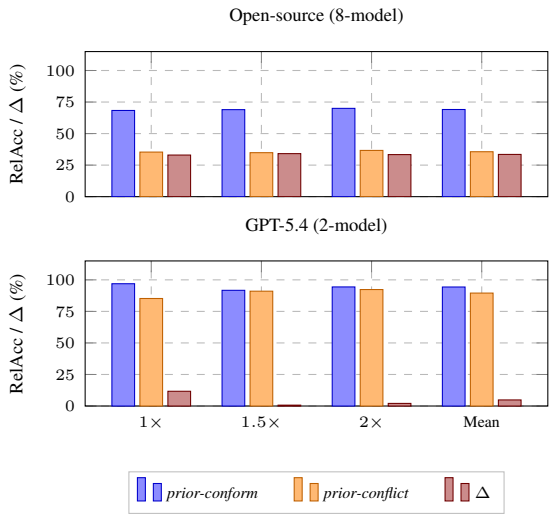

Figure 7: **Quadrupling the image area does not narrow the conflict gap in either family group; the prior-override effect is not a perceptual bottleneck. The open-source 8-model gap stays near 33% at every scale (top), while the GPT-5.4 family stays under 12% (bottom).** Per-scale and overall mean RelAcc (%) on *prior-conform* (blue) and *prior-conflict* (orange), with the conflict gap Δ (red); "Mean" on each panel is the simple mean over the three scales.

4.2.3 Results

Relations with stronger knowledge priors produce larger conflict gaps. Fig. 6 (top) shows that the open-source conflict gap appears in all four relations, scaling with the strength of knowledge prior. The two largest gaps are on the relations whose class names carry the strongest priors: **Aggregation** ($\text{RelAcc}_{\textit{prior-conform}} = 85.42\%$, $\Delta = 50.71\%$) and **Inheritance** ($\text{RelAcc}_{\textit{prior-conform}} = 73.84\%$, $\Delta = 50.57$). Composition has a smaller gap ($\Delta = 24.40\%$), and Dependency has the smallest ($\Delta = 8.24\%$). The smaller Dependency gap is partly a floor effect as this is the relation with the lowest accuracy. The smaller Composition gap reflects moderate *prior-conform* accuracy (77.97%) combined with relatively preserved *prior-conflict* accuracy (53.57%), suggesting that the composition diamond appears more robustly parsed when the arrow direction is reversed. The pattern is mostly consistent: the stronger the knowledge prior, the lower the accuracy when reversing the direction. The GPT-5.4 family (Fig. 6 bottom) reproduces the same Aggregation/Inheritance signal but at a much smaller magnitude ($\Delta \leq 6.25\%$ on every relation).

Larger images do not narrow the conflict gap. Fig. 7 (top) shows the open-source conflict gap is essentially the same as we quadruple the image area: $\Delta = 32.98\%$ at $1\times$, 34.11% at $1.5\times$, and 33.34% at $2\times$. Both *prior-conform* and *prior-conflict* accuracies rise modestly with scale ($68.32\% \rightarrow 69.99\%$ and $35.33\% \rightarrow 36.65\%$, respectively). Thus, the visual content is read marginally better at larger scales, but the gap between them is unaffected. Giving the model more pixels does not reduce its reliance on the knowledge prior, and consequently this failure mode is not a perceptual bottleneck. The GPT-5.4 family (Fig. 7 bottom) is similarly insensitive to scale on $1.5\times$ and $2\times$ ($\Delta \leq 2.07\%$); the larger $\Delta = 11.69\%$ at $1\times$ is driven by GPT-5.4 Mini's instability on the smallest images, not a generic perceptual issue.

4.3 Q3: Does scaling the model size fix the prior-over-vision effect?

4.3.1 Motivation

Increasing the model size can improve model performance: with more capacity, the model usually performs better and can learn to weigh the visual evidence more heavily than the knowledge prior. **Q3** tests this on *2-reverse* and *3-reverse*, and checks the prerequisite skills (class-name reading and arrow-direction reading without knowledge prior) along the same scaling axis. A genuine scaling fix should improve both performance and prerequisite skills monotonically with parameter count.

4.3.2 Experiments

We compare same-family variants, 2B, 4B, 8B, and 32B (or 38B), within InternVL3.5, Qwen3 and GPT separately. For each variant we report (i) RelAcc on *2-reverse* and *3-reverse* averaged across four relations and three scales (Fig. 8), and (ii) two prerequisite measurements on *prior-free*: EM (exact class-name recognition) and $\text{RelAcc}_{\textit{prior-free}}$ (arrow-direction accuracy with prior-free labels), each averaged across four relations and three scales. We adopt thresholds of $\text{EM} \geq 95\%$ (perception intact) and $\text{RelAcc} \geq 50\%$ (arrow-reading above the always-True baseline) and call models passing both the *prerequisite-passing subset* (Fig. 9).

4.3.3 Results

Naive scaling helps Qwen3 on *2-reverse* but appears inverse for InternVL3.5. Fig. 8 shows starkly different scaling patterns across families on *2-reverse*. Qwen3 scales monotonically: 2B (21.74%) $\rightarrow$ 4B (32.23%) $\rightarrow$ 8B (76.14%) $\rightarrow$ 32B (80.29%), a +58.55-point gain across the fam-

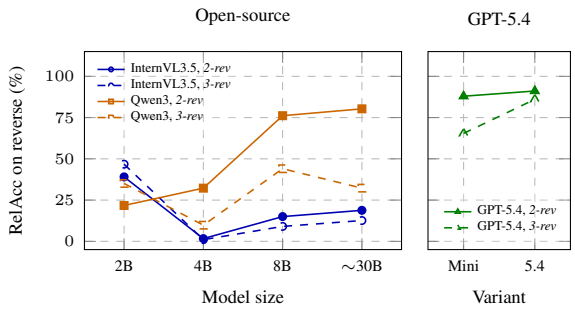

Figure 8: **Within-family scaling looks positive for Qwen3 on *2-reverse* but inverse for InternVL3.5 (left); on *3-reverse* most larger open-source variants collapse below the 2B. The GPT-5.4 family (right) sits well above the open-source variants on both conditions, and scales positively from GPT-5.4 Mini to GPT-5.4 on *3-reverse*** (65.41 $\rightarrow$ 86.05)**.** Per-variant RelAcc (%) on *2-reverse* (solid) and *3-reverse* (dashed) vs. within-family scale; shared y-axis across panels.

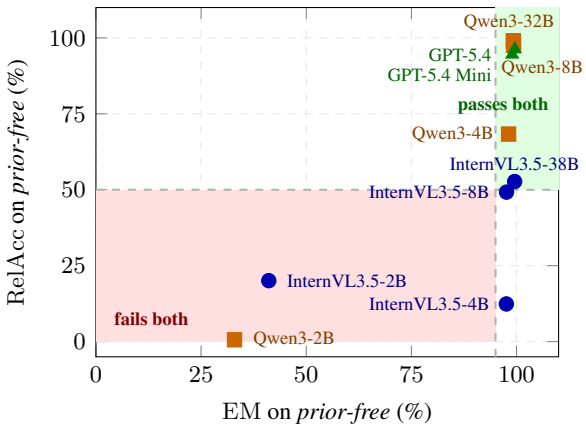

Figure 9: **Most InternVL3.5 variants fail the prerequisite skills required to interpret a low *2-reverse* score as bias; only InternVL3.5-38B and three of four Qwen3 variants pass both, while both closed-source GPT-5.4 models pass comfortably.** Dashed thresholds at x=95 and y=50; marker color = family (blue: InternVL3.5; orange: Qwen3; green: GPT-5.4).

ily. InternVL3.5, by contrast, appears to scale *inversely*: 2B (39.06%) is uniquely strong, while 4B (1.66%), 8B (15.00%), and 38B (18.79%) all fall well below the 2B; even the 38B model never recovers within 20% of the 2B. On *3-reverse*, almost every larger variant collapses in both families. Most strikingly, Qwen3-32B drops 48.05% from *2-rev* to *3-rev* (80.29% $\rightarrow$ 32.24%), ending below both Qwen3-8B (44.01%) and Qwen3-2B (35.08%). On the harder condition, the strongest variant is 2B for InternVL3.5 but 8B for Qwen3.

InternVL3.5's apparent inverse scaling reflects skill deficit, not true scaling failure. Fig. 9 reveals the catch behind Fig. 8: most InternVL3.5 variants lack the underlying skills needed to interpret a low *2-reverse* score as bias. Only InternVL3.5-38B passes both prerequisites; InternVL3.5-2B fails both, while InternVL3.5-4B (12.40%) and InternVL3.5-8B (49.23%) pass class-recognition but fail arrow-reading without knowledge prior. Qwen3, by contrast, has three of four variants passing both prerequisites (only Qwen3-2B fails). The apparent inverse scaling within InternVL3.5 is therefore largely an artifact: when 4B and 8B cannot read arrows even without a misleading prior, their *2-reverse* accuracy is uninformative about scaling on the bias task. Within the four-model prerequisite-passing subset (Qwen3-4B/8B/32B + InternVL3.5-38B), the headline conflict gap is $\Delta = 32.94$ %, essentially identical to the all-eight 33.48 %, so the prior-override claim from **Q1** is robust.

5 Conclusion

We set out to ask whether current VLMs read UML class diagrams or instead answer from pretrained knowledge priors about how named classes typically relate. Across a controlled benchmark evaluated on 8 open-source VLMs from two families (InternVL3.5, Qwen3) and two closed-source frontier models (GPT-5.4, GPT-5.4 Mini), we find that reversing a single arrow drops relation accuracy by 33.48% on average across open-source models, showing that they substitute priors for diagrams in the post-reading inference step. Frontier models reduce this clean two-class gap: GPT-5.4 Mini cuts it to $\Delta = 10.00\%$, and GPT-5.4 nearly closes it. However, harder three-class variants remain challenging, with average drops of 45.28% for open-source models and 18.62% for the GPT-5.4 family, indicating that even frontier models are not immune once auxiliary structural context is added.

These findings are robust across the two axes built into our benchmark. Quadrupling the image area does not narrow the conflict gap ($\sim$33% at every scale), and the gap is largest for relations whose class names carry strong priors (*e.g., Aggregation*, *Inheritance*), producing drops of $\Delta \approx 51\%$ each. Within open-source families, more parameters do not necessarily improve performance. While Qwen3 shows a steady increase in $\mathrm{RelAcc}_{conflict}$ as model size grows, InternVL3.5 drops sharply from 2B to 4B, which can be attributed to weak class-name reading and $\mathrm{RelAcc}_{prior\text{-}free}$.

Limitations

This benchmark is designed for controlled diagnosis of prior-over-visual override in VLMs, and its current scope is intentionally narrow in two respects. First, each accepted diagram is paired with exactly one target relation query over an ordered class pair, so the evaluation measures targeted relation judgment rather than full-diagram parsing or multi-query reasoning. Second, the headline results are reported after aggregating across the four relation types and, for slice-level comparisons, over the models with full protocol coverage. This means the aggregate trends should not be read as implying that all UML relations, model families, or image scales fail in exactly the same way. These choices improve comparability for the main controlled analyses, but some finer-grained relation-specific variation is left to future work.

References

Gábor Antal, Richárd Vozár, and Rudolf Ferenc. 2024. Toward a new era of rapid development: Assessing gpt-4-vision's capabilities in uml-based code generation. In *Proceedings of the 1st International Workshop on Large Language Models for Code*, pages 84–87.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Nitzan Bitton-Guetta, Yonatan Bitton, Jack Hessel, Ludwig Schmidt, Yuval Elovici, Gabriel Stanovsky, and Roy Schwartz. 2023. Breaking common sense: Whoops! a vision-and-language benchmark of synthetic and compositional images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2616–2627.

Victor Campanello, Shariq Shahbaz, Vladislav Indykov, and Daniel Strüber. 2025. On the use of gpt-4 in the reverse engineering of class diagrams. *Journal of Object Technology*, 24(2):1–14.

Fangwei Chen, Li Zhang, Xiaoli Lian, and Nan Niu. 2022. Automatically recognizing the semantic elements from uml class diagram images. *Journal of Systems and Software*, 193:111431.

Aaron Conrardy and Jordi Cabot. 2024. From image to uml: First results of image based uml diagram generation using llms. *arXiv preprint arXiv:2404.11376*.

Daniele De Bari, Giacomo Garaccione, Riccardo Coppola, Marco Torchiano, and Luca Ardito. 2024. Evaluating large language models in exercises of uml class diagram modeling. In *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, pages 393–399.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Michal Golovanevsky, William Rudman, Michael A. Lepori, Amir Bar, Ritambhara Singh, and Carsten Eickhoff. 2025. Pixels versus priors: Controlling knowledge priors in vision-language models through visual counterfacts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24837–24852, Suzhou, China. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.

Tracy Hammond and Randall Davis. 2006. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *ACM SIGGRAPH 2006 Courses*, pages 25–es.

Yifan Hou, Buse Giledereli, Yilei Tu, and Mrinmaya Sachan. 2024. Do vision-language models really understand visual language? *arXiv preprint arXiv:2410.00193*.

Wen Huang, Hongbin Liu, Minxin Guo, and Neil Gong. 2024. Visual hallucinations of multi-modal large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9614–9631.

Bilal Karasneh and Michel RV Chaudron. 2013a. Img2uml: A system for extracting uml models from images. In *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, pages 134–137. IEEE.

Bilal Karasneh and Michel RV Chaudron. 2013b. Online img2uml repository: An online repository for uml models. In *EESSMod@ MoDELS*, pages 61–66.

Edward Lank, Jeb S Thorley, and Sean Jy-Shyang Chen. 2000. An interactive system for recognizing hand drawn uml diagrams. In *Proceedings of the 2000*

conference of the Centre for Advanced Studies on Collaborative research, page 7.

Kang-il Lee, Minbeom Kim, Seunghyun Yoon, Minsung Kim, Dongryeol Lee, Hyukhun Koh, and Kyomin Jung. 2025. Vlind-bench: Measuring language priors in large vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4129–4144.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. 2024a. Naturalbench: Evaluating vision-language models on natural adversarial samples. *Advances in Neural Information Processing Systems*, 37:17044–17068.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305.

Yunxin Li, Baotian Hu, Haoyuan Shi, Wei Wang, Longyue Wang, and Min Zhang. 2024b. Visiongraph: leveraging large multimodal models for graph theory problems in visual context. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zhiqiu Lin, Xinyue Chen, Deepak Pathak, Pengchuan Zhang, and Deva Ramanan. 2023. Revisiting the role of language priors in vision-language models. *arXiv preprint arXiv:2306.01879*.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Jiazhen Liu, Yuhan Fu, Ruobing Xie, Runquan Xie, Xingwu Sun, Fengzong Lian, Zhanhui Kang, and Xirong Li. 2025. Phd: A chatgpt-prompted visual hallucination evaluation dataset. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19857–19866.

Xiaoyuan Liu, Wenxuan Wang, Youliang Yuan, Jen-tse Huang, Qiuzhi Liu, Pinjia He, and Zhaopeng Tu. 2024b. Insight over sight: Exploring the vision-knowledge conflicts in multimodal llms. *arXiv preprint arXiv:2410.08145*.

Tiange Luo, Ang Cao, Gunhee Lee, Justin Johnson, and Honglak Lee. 2024. Probing visual language priors in vlms. *arXiv preprint arXiv:2501.00569*.

Artem Naboichenko and René Peinl. 2026. Unlocking uml class diagram understanding in vision language models. *arXiv preprint arXiv:2605.11634*.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming language priors in visual question answering with adversarial regularization. *Advances in neural information processing systems*, 31.

Josselin S Roberts, Tony Lee, Chi H Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. 2024. Image2struct: Benchmarking structure extraction for vision-language models. *Advances in Neural Information Processing Systems*, 37:115058–115097.

Arnaud Roques and PlantUML Contributors. PlantUML Software.

Bernhard Rumpe. 2016. *Modeling with UML*, volume 98. Springer.

Nimra Shehzadi, Javed Ferzund, Rubia Fatima, and Adnan Riaz. 2025. Automatic complexity analysis of uml class diagrams using visual question answering (vqa) techniques. *Software*, 4(4):22.

An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. 2025. Vision language models are biased. *arXiv preprint arXiv:2505.23941*.

Chong Wang, Beian Wang, Peng Liang, and Jie Liang. 2026. Assessing uml diagrams by gpt: Implications for education. *Journal of Systems and Software*, 234:112709.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Yike Wu, Yu Zhao, Shiwan Zhao, Ying Zhang, Xiaojie Yuan, Guoqing Zhao, and Ning Jiang. 2022. Overcoming language priors in visual question answering via distinguishing superficially similar instances. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5721–5729.

Zongyu Wu, Yuwei Niu, Hongcheng Gao, Minhua Lin, Zhiwei Zhang, Zhifang Zhang, Qi Shi, Yilong Wang, Sike Fu, Junjie Xu, and 1 others. 2025. Lanp: Rethinking the impact of language priors in large vision-language models. *arXiv preprint arXiv:2502.12359*.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. 2025. Benchmarking and improving large vision-language models for fundamental visual graph understanding and reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30678–30701.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. 2024. Vgbench: Evaluating large language models on vector graphics understanding and generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3647–3659.

A Additional examples

Fig. 10 expands the condition illustration in Fig. 3. It shows representative diagrams for the prior-conforming, prior-conflicting, and prior-free settings, including the three-class variants used to test whether an added auxiliary relation changes model reliance on the diagram.

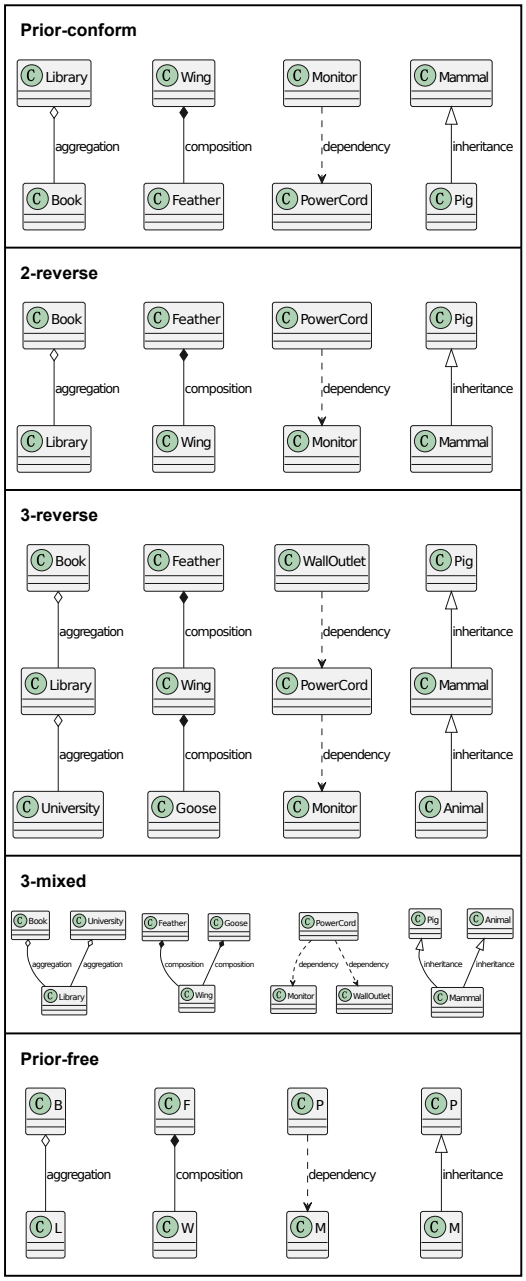

Figure 10: **Examples for each condition type.** These examples include the additional prior-conflict subsets and the prior-free configurations omitted from the compact main-text figure.

B Detailed Experimental Setup

We evaluate eight open-source VLMs across two families, `InternVL3.5-{2B,4B,8B,38B}` and `Qwen3-{2B,4B,8B,32B}`, plus two closed-source frontier models, GPT-5.4 and GPT-5.4 Mini. We query each model on all five conditions: prior-conforming, *2-reverse* / *3-reverse* / *3-mixed* (prior-conflicting), and *prior-free*. Each image is evaluated at three image scales ($1\times$, $1.5\times$, $2\times$). We run experiments on $4\times$ NVIDIA L40S GPUs (48 GB VRAM each), serving open-source models through vLLM and querying closed-source models through their APIs under the same prompt template and output schema. The open-source inference required approximately 120 GPU-hours, and closed-source API inference cost approximately 200 USD.

Parsing. We pass all raw model outputs through a deterministic parser before scoring. The parser handles common formatting variation (*e.g.,* punctuation wrappers, sentence envelopes around the answer, paraphrases of True/False, and JSON parsing for class-name recovery) and produces a normalized prediction. Outputs that cannot be resolved unambiguously into {True, False, Unknown} for the relation query (or into a parsed class-name set for class-name recovery) are conservatively mapped to Unknown.

C UML Diagram Rendering

All diagrams in our benchmark are rendered with PlantUML to ensure a consistent and reproducible visual style (Roques and Contributors). PlantUML is distributed under the GNU General Public License v3.0 (GPL-3.0), and we use it only as a rendering tool for diagrams generated from our own structured specifications. We first represent each accepted UML instance as a structured textual specification, then render it into an image using the same PlantUML pipeline for all conditions. This keeps the diagram style, class-box layout, arrow notation, and relation labels consistent across prior-conforming, prior-conflicting, and *prior-free* examples. For the image-scale analysis, we resize the same rendered diagrams to the evaluated resolutions, so the scale variants differ only in image size and not in UML content.

D Benchmark

Table 1 summarizes the number of benchmark instances used in our evaluation. We count one *base*

instance as one UML diagram paired with one target relation-direction query before image scaling. Since every base instance is rendered at three image scales ($1\times$, $1.5\times$, and $2\times$), the number of rendered inputs is three times the base-instance count. Overall, the benchmark contains 27,156 base instances and 81,468 rendered image-query inputs across the five evaluated conditions.

The *prior-free* condition is smaller than the other conditions because it is constructed as a diagnostic control rather than sampled to match the full semantic benchmark size. We derive it by replacing the original semantic class names with their uppercase-letter abbreviations, such as `GoldenRetriever` → `GR` and `Dog` → `D`. This many-to-one transformation removes class-name semantics, but it also causes many originally distinct diagrams to collapse into the same abbreviated class pair. We therefore keep only unique valid abbreviated instances and discard cases where abbreviation creates duplicate class names within the same diagram. The resulting counts reflect the number of distinct *prior-free* diagrams after this deduplication step.

Condition	Agg.	Comp.	Dep.	Inh.	Base	Rendered
prior-free	338	373	1,044	1,401	3,156	9,468
prior-conform	1,200	1,200	1,300	2,300	6,000	18,000
2-reverse	1,200	1,200	1,300	2,300	6,000	18,000
3-reverse	1,200	1,200	1,300	2,300	6,000	18,000
3-mixed	1,200	1,200	1,300	2,300	6,000	18,000
Total	5,138	5,173	6,244	10,601	27,156	81,468

Table 1: Benchmark size by condition and UML relation. Counts under Agg., Comp., Dep., and Inh. are base instances before image scaling. Rendered counts include the three evaluated image scales ($1\times$, $1.5\times$, and $2\times$).

E Full per-cell results

Table 2 reports **RelAcc** and **EM** for every (model, condition, relation, image scale) cell across all five conditions in the table, including the *3-mixed* and *3-reverse* conditions shown in Section A, and adds the closed-source GPT-5.4 and GPT-5.4 Mini blocks for completeness. The aggregated views in the main text (Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8, Fig. 9) are derived by averaging over relations and/or scales as noted in their captions.

F Evaluation prompts

The prompts are deliberately designed to minimize ambiguity in the visual interpretation task. They first make the notation explicit by describing the visible components of each class box and clarifying that the green circled "C" is only a class-type icon, not part of the relation semantics. They then define the four UML relations in the benchmark, including inheritance, aggregation, composition, and dependency, and specify how each arrowhead or line ending should be interpreted. Beyond explaining the notation, the prompts repeatedly emphasize that the diagram itself is the only valid source of evidence. The model is instructed to inspect the arrow or line direction carefully, disregard external knowledge about the class names, and avoid making judgments from common software-design assumptions unless the relation is visually supported.

This prompt design makes the evaluation intentionally conservative. The goal is not to test whether models fail because the diagram is underspecified or because the notation is unfamiliar. Instead, the instructions give the model the relevant visual rules in advance and explicitly warn it against relying on semantic priors. Therefore, when a model still predicts the prior-consistent relation under conflict conditions, the error is harder to attribute to prompt ambiguity. It more directly reflects a failure to privilege visual evidence over learned software-design priors. The prompts used in the experiments are shown below, where terms in angle brackets (<>) denote instance-specific placeholders (Fig. 11, Fig. 12, Fig. 13, Fig. 14).

Size	Setting	Aggregation						Composition						Dependency						Inheritance					
		1×		1.5×		2×		1×		1.5×		2×		1×		1.5×		2×		1×		1.5×		2×	
		RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM	RelAcc	EM
InternVL3.5																									
2B	prior-free	21.30	25.44	20.12	23.67	17.75	22.19	12.06	16.89	9.65	15.01	9.92	13.94	17.34	67.05	17.24	65.23	17.62	65.52	34.19	61.74	32.12	58.82	31.55	57.74
	prior-conform	96.17	99.50	96.25	99.50	96.00	99.33	90.83	99.08	90.92	99.00	91.00	99.17	40.08	95.62	39.85	95.00	39.23	95.15	48.83	96.43	52.74	96.39	52.70	96.39
	2-reverse	44.00	98.50	45.08	98.50	46.08	98.67	71.67	99.42	71.67	99.42	73.33	99.33	27.85	96.15	27.54	96.46	28.15	96.08	10.09	96.61	11.61	96.65	11.65	96.74
	3-reverse	50.33	98.00	51.08	98.25	51.08	98.00	67.67	99.08	67.08	98.83	66.92	98.83	34.62	95.54	35.08	95.54	34.08	95.54	35.13	91.65	34.43	90.70	33.48	90.35
	3-mixed	5.92	97.92	6.50	97.92	6.42	97.92	19.92	98.67	20.83	98.75	21.67	98.67	32.77	97.46	30.46	97.54	31.85	97.62	58.26	96.43	55.65	96.65	54.96	96.43
4B	prior-free	36.69	96.45	36.69	96.75	36.09	96.15	12.87	98.66	10.99	99.46	11.26	99.46	0.00	97.61	0.00	97.51	0.00	97.61	1.07	96.86	1.50	97.14	1.57	97.29
	prior-conform	56.00	99.25	59.58	99.17	59.67	99.33	29.50	99.67	32.08	99.42	31.17	99.42	0.69	97.46	0.77	97.38	0.54	97.15	57.09	97.96	57.17	98.04	57.39	98.13
	2-reverse	1.17	99.17	1.17	99.25	1.17	99.17	4.92	99.33	5.75	99.33	5.58	99.33	0.08	97.31	0.00	97.38	0.08	97.08	0.00	98.00	0.00	97.78	0.00	97.70
	3-reverse	0.25	99.08	0.42	99.17	0.42	99.08	3.83	99.17	3.75	99.17	3.58	99.17	0.00	97.92	0.00	98.00	0.00	98.00	0.00	97.43	0.00	97.48	0.00	97.43
	3-mixed	0.00	98.83	0.00	98.75	0.00	98.92	0.17	98.92	0.08	99.00	0.17	98.83	0.00	97.69	0.00	97.77	0.00	97.92	0.00	97.04	0.00	97.09	0.00	97.04
8B	prior-free	56.21	98.52	54.14	98.52	55.03	98.52	66.49	98.93	60.86	99.20	59.52	98.93	0.19	95.11	0.10	95.21	0.10	95.02	81.51	97.72	77.59	97.93	78.94	97.79
	prior-conform	87.75	99.83	88.58	99.83	88.58	99.83	86.25	99.83	85.58	99.67	85.08	99.67	1.00	98.08	1.15	98.08	1.08	98.08	96.00	98.65	95.96	98.61	95.91	98.65
	2-reverse	17.17	99.67	15.75	99.75	16.33	99.75	35.75	99.83	34.67	99.75	34.42	99.83	0.00	98.77	0.00	98.92	0.00	98.92	9.35	98.70	8.00	98.65	8.57	98.61
	3-reverse	9.17	99.67	8.50	99.67	8.75	99.67	26.08	99.58	25.67	99.67	26.50	99.58	0.23	98.08	0.38	98.31	0.38	98.08	1.13	98.52	0.74	98.43	0.96	98.43
	3-mixed	0.00	99.50	0.00	99.50	0.00	99.50	0.33	99.17	0.25	99.17	0.25	99.25	0.23	98.23	0.31	98.38	0.31	98.23	0.43	98.61	0.48	98.57	0.43	98.57
38B	prior-free	99.41	100.00	99.11	100.00	99.11	100.00	96.78	100.00	94.64	100.00	95.17	100.00	9.00	99.23	10.25	99.14	10.82	99.14	4.07	99.00	7.14	99.07	7.00	99.00
	prior-conform	98.42	99.83	98.17	99.83	98.33	99.83	95.67	99.75	95.83	99.75	95.75	99.75	18.46	98.00	17.92	98.23	17.85	98.15	35.52	99.17	39.87	99.22	38.13	99.17
	2-reverse	24.50	100.00	22.08	100.00	21.83	100.00	52.08	99.50	51.83	99.50	51.17	99.50	0.54	98.38	0.77	98.31	0.62	98.31	0.04	98.83	0.04	98.87	0.00	98.87
	3-reverse	20.17	99.75	17.67	99.75	18.25	99.75	30.08	99.50	27.17	99.42	26.33	99.58	3.38	99.00	3.46	99.00	3.69	99.00	0.39	98.61	0.35	98.57	0.35	98.57
	3-mixed	3.08	99.83	2.83	99.83	2.75	99.83	18.33	99.17	18.67	99.17	18.67	99.17	11.92	98.54	12.00	98.15	12.08	98.31	2.22	98.70	2.13	98.83	2.22	98.70
Qwen3																									
2B	prior-free	0.30	25.74	0.30	19.53	0.30	17.75	0.27	20.91	0.27	18.23	0.80	14.48	1.34	58.24	1.44	56.13	1.25	57.85	0.36	36.55	0.21	32.48	0.29	37.54
	prior-conform	42.00	99.83	46.33	99.83	54.50	99.58	23.08	99.58	27.33	99.58	32.08	99.58	27.23	98.08	27.92	98.38	34.23	98.23	49.61	98.22	51.09	98.09	58.09	98.70
	2-reverse	17.00	99.58	21.58	99.50	29.83	99.33	10.08	99.75	13.33	99.67	15.42	99.50	34.00	99.23	34.23	98.77	38.23	99.38	14.48	98.52	13.78	98.26	18.91	98.39
	3-reverse	33.50	99.67	39.08	99.67	45.08	99.42	10.50	99.50	14.50	99.42	23.75	99.50	63.31	98.15	57.69	98.38	65.23	98.85	16.96	97.52	24.70	97.61	26.61	98.26
	3-mixed	3.83	99.33	3.75	99.50	14.50	99.33	2.75	99.00	3.42	99.50	4.58	99.33	42.08	99.15	55.15	98.92	61.46	99.08	36.43	97.78	27.48	98.13	43.22	97.48
4B	prior-free	86.39	98.82	84.32	97.93	83.43	97.04	84.45	99.20	84.72	99.46	90.08	99.46	17.72	97.61	16.57	97.80	14.85	97.51	86.72	97.79	83.65	97.64	87.22	97.07
	prior-conform	95.75	99.75	95.58	99.75	94.92	99.75	94.83	99.50	93.92	99.58	95.50	99.58	25.31	98.46	23.85	98.38	28.85	97.69	96.26	98.57	95.26	98.70	97.57	98.57
	2-reverse	23.33	99.92	21.17	99.83	23.33	99.75	71.75	99.92	68.75	99.83	73.83	99.92	6.62	99.15	5.54	99.31	10.31	99.31	25.70	98.43	22.57	98.52	33.87	98.87
	3-reverse	2.08	99.42	3.83	99.67	4.58	100.00	28.75	99.67	30.58	99.92	44.75	99.92	0.92	97.15	0.77	97.54	0.69	97.62	0.39	98.78	0.04	99.09	0.04	99.00
	3-mixed	6.50	99.50	4.25	99.67	3.83	99.67	41.67	99.50	30.67	99.42	31.17	99.50	3.08	98.23	5.08	98.85	12.31	98.92	1.13	98.30	0.04	98.39	1.48	98.17
8B	prior-free	99.11	99.11	98.82	99.11	97.93	99.11	96.25	99.46	95.44	99.46	94.37	99.46	96.65	98.85	97.13	99.33	98.08	99.52	99.00	99.21	98.79	99.21	98.86	99.64
	prior-conform	99.25	99.92	99.33	99.92	99.58	100.00	98.58	99.83	98.42	99.75	98.92	99.42	98.38	99.62	98.31	99.54	99.08	99.62	99.52	99.52	99.52	99.52	99.39	99.39
	2-reverse	63.17	99.67	62.33	99.58	64.83	99.75	83.67	100.00	82.42	100.00	85.08	99.50	83.46	99.23	81.54	99.15	89.15	99.54	76.78	99.48	74.17	99.39	67.09	99.43
	3-reverse	35.92	99.92	43.67	99.75	50.08	99.75	52.08	99.83	55.17	99.83	58.08	99.67	71.00	99.00	72.08	99.08	78.46	99.54	5.57	99.09	3.22	99.43	2.78	99.57
	3-mixed	27.17	99.67	40.58	99.75	51.50	99.75	47.08	99.75	38.25	99.75	48.75	99.50	86.08	98.77	83.00	99.15	92.08	99.00	53.13	99.17	47.96	99.43	56.52	99.52
32B	prior-free	99.11	99.41	99.11	99.11	98.52	98.82	99.46	99.46	99.73	99.73	99.20	99.20	98.95	99.14	99.23	99.33	98.75	99.52	98.64	98.86	98.72	98.93	99.29	99.43
	prior-conform	99.83	99.83	99.83	99.83	99.75	99.75	99.50	99.50	99.50	99.58	99.92	99.92	99.23	99.46	99.15	99.46	99.15	99.31	99.48	99.48	99.39	99.39	99.65	99.65
	2-reverse	83.67	99.75	82.33	99.75	84.17	99.83	96.58	99.58	95.67	99.67	96.33	99.83	91.46	99.23	91.08	99.23	90.38	99.54	49.65	99.35	49.13	99.26	53.04	99.35
	3-reverse	20.08	99.83	30.00	99.92	33.08	99.83	42.33	99.50	60.92	99.58	59.58	99.92	41.23	99.15	37.62	99.31	36.62	99.62	6.17	99.17	6.22	99.04	13.04	99.43
	3-mixed	23.83	99.42	39.50	99.50	45.83	99.83	48.75	99.33	44.83	99.83	46.33	100.00	89.38	98.85	91.23	99.15	89.92	99.23	38.00	98.91	28.09	99.70	36.00	99.65
GPT-5.4																									
5.4	prior-free	92.90	100.00	99.41	100.00	91.42	99.70	96.25	99.73	97.86	99.73	75.34	100.00	86.88	92.15	97.89	99.23	99.04	99.33	98.93	99.07	98.79	98.79	99.86	99.86
	prior-conform	98.00	99.67	75.67	100.00	90.67	100.00	96.33	100.00	62.33	99.83	71.50	100.00	97.38	98.92	99.08	99.23	98.77	99.08	99.65	99.65	99.48	99.48	99.91	99.91
	2-reverse	89.00	99.67	86.50	100.00	90.67	100.00	92.17	99.17	68.17	100.00	75.17	100.00	96.92	98.62	99.38	99.69	99.54	99.54	98.09	99.48	98.17	99.57	99.39	99.65
	3-reverse	86.00	99.50	84.17	100.00	94.83	100.00	93.83	99.83	78.17	100.00	80.50	99.83	76.31	98.00	90.31	100.00	98.62	100.00	70.26	99.48	86.61	99.48	93.04	99.83
	3-mixed	92.33	99.50	96.17	100.00	98.33	99.50	86.33	99.67	96.00	100.00	96.50	100.00	86.31	99.23	98.00	100.00	99.38	99.54	96.26	99.83	99.04	99.48	91.74	92.52
5.4 Mini	prior-free	97.34	100.00	100.00	100.00	100.00	100.00	83.11	100.00	99.73	100.00	98.93	100.00	91.09	98.85	92.05	99.43	96.36	97.61	97.36	99.50	99.86	99.86	99.64	99.64
	prior-conform	98.42	99.50	99.50	99.75	100.00	100.00	93.08	99.58	98.83	99.75	98.00	99.75	93.38	99.38	99.00	99.08	96.77	99.54	99.17	99.17	99.74	99.74	99.52	99.57
	2-reverse	71.25	99.00	95.33	99.75	92.75	100.00	71.42	99.75	93.58	99.92	91.08	100.00	83.92	99.46	96.69	99.46	95.08	99.77	79.09	98.78	90.35	99.70	94.91	99.57
	3-reverse	47.42	99.92	50.33	99.92	53.75	100.00	48.00	99.67	57.08	100.00	52.50	100.00	86.69	99.00	94.69	99.31	95.92	99.38	48.48	98.22	70.30	99.70	79.70	99.74
	3-mixed	48.75	99.58	31.92	100.00	35.42	100.00	49.33	99.33	47.25	99.92	31.58	99.92	86.00	99.23	95.77	99.69	95.00	99.54	57.52	99.48	52.74	99.57	65.83	99.65

Table 2: Full per-cell results. Values are percentages for each image scale within each (model, condition, relation) cell. Conditions shown are *prior-free*, *prior-conform*, *2-reverse*, *3-reverse*, and *3-mixed*. For each relation, the columns are grouped by image scale (1×, 1.5×, 2×). **RelAcc** (= Relation Accuracy) and **EM** (= Class-name Exact Match) are reported under each scale. We use greedy decoding to ensure each cell has a deterministic single-sample accuracy.

System prompt

You are an accurate UML diagram reasoning assistant. You will be asked to complete a question or task. Treat the provided image as the only source of truth. Do not use the model's internal knowledge, assumptions, and guesses. Follow the requested output format exactly, without extra explanation.

Figure 11: **System prompt.** General instruction shared by all evaluation prompts, emphasizing image-grounded reasoning and strict output formatting.

Stage-1 class-name recovery prompt

The image provided is a UML diagram showing **<Relation>** relationships. The UML diagram contains **<Number>** classes. Each class is represented as a box, with the class name at the top. The green circled letter 'C' inside each box is only a class-type icon and is not part of the class name. The actual class name is the text to the right of the icon. You must treat the image as the only source of truth for the task. Do not infer, guess, or invent any names. Complete the following task using only the class names that are visibly and explicitly depicted in the UML diagram.
The task is: List all class names that appear in the UML diagram.
Each class name in your final output must match the UML diagram text exactly, character by character. You may reason privately, but do not reveal any reasoning or intermediate steps. Output only a JSON array of strings, for example: ["ClassA", "ClassB"]. Do not output any text before or after the JSON array. Output [] if the image is missing or unreadable.

Figure 12: **Class-name recovery prompt.** Prompt used to recover the class names explicitly visible in the UML diagram before relation-direction evaluation.

Stage-2 relation-direction QA prompt

The image provided is a UML diagram showing **<Relation>** relationships. The diagram contains **<Two/Three>** classes: **<Class names>**. Each class is represented as a box, with the class name at the top. The green circled letter 'C' inside each box is only a class-type icon and is not part of the class name. The actual class name is the text to the right of the icon.
<Relation-specific UML rule>
Carefully verify the direction of every arrow or line in the UML diagram. Treat arrow direction as authoritative and do not assume it. You must treat the image as the only source of truth for the question. Do not use the model's internal knowledge or prior assumptions to attempt to answer the question. Answer the following question solely from the relationships explicitly depicted in the image.
The question is: **<Relation-specific question templates>**
You may reason privately, but do not reveal any reasoning or intermediate steps. Output only one of: True, False, or Unknown. Output 'Unknown' if the image is missing, unreadable, or the relation direction is unclear.

Figure 13: **Relation-direction QA prompt.** Prompt used to answer the relation-direction question after the class names have been specified.

Relation-specific rules and query templates

Relation-specific UML rules:
Inheritance: A solid line with a hollow triangular arrow indicates an inheritance relationship. The arrow points from the subclass to the superclass.
Aggregation: A solid line with a hollow diamond indicates an aggregation relationship. The class at the diamond end is the whole, and the class at the other end is the part.
Composition: A solid line with a filled diamond indicates a composition relationship. The class at the diamond end is the whole, and the class at the other end is the part.
Dependency: A dashed line with an open arrow indicates a dependency relationship. The arrow points from the dependent class to the class it depends on.

Relation-specific question templates:
Inheritance: Does class **<ClassA>** inherit from class **<ClassB>**?
Aggregation: Is class **<ClassA>** the whole and class **<ClassB>** the part?
Composition: Is class **<ClassA>** the whole and class **<ClassB>** the part?
Dependency: Does class **<ClassA>** depend on class **<ClassB>**?

Figure 14: **Relation-specific UML rules and query templates.** UML notation rules and question templates used to instantiate the relation-direction QA prompt.