# CAN VLMS ASSEMBLE LOCAL DECISIONS INTO GLOBAL STRUCTURES?

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Many vision-language tasks require combining multiple local decisions into a single structured output. We ask whether vision-language models (VLMs) can reliably assemble such decisions into assignments, orderings, groupings, and hierarchies. We identify an *assembly deficit*: a model can answer the underlying local questions accurately yet fail to produce the corresponding global structure. On RecipeQA, we isolate this deficit while holding the model and task input fixed, finding that local step–image decisions are substantially stronger than whole-order prediction or scoring. We introduce *local+solve*, a training-free procedure that extracts local scores from a frozen VLM and assembles them with a task-specific solver under task-defined constraints. Across six controlled structural families, five tasks drawn from public benchmarks, and GUI-agent settings, *local+solve* improves over matched global prediction, including stronger inference variants where tested, and its gain operationally measures the recoverable part of the deficit. The deficit narrows from 7B to 72B but persists in most families. Post-training on *local+solve*'s outputs nearly internalizes assembly on the trained task, while multi-family *local+solve* pseudo-labels improve global prediction on two unseen families, suggesting that assembly is a distinct, learnable capability.

## 1 INTRODUCTION

Consider a vision-language model (VLM) given shuffled images of recipe steps and asked to recover their correct order. The task can be decomposed into simpler decisions, such as matching images to individual steps or deciding which of two images comes first. These decisions must then be combined into one valid ordering. Prior work similarly decomposes problems into simpler subproblems in language reasoning (Khot et al., 2023; Zhou et al., 2023) and into modular executable steps in visual reasoning (Gupta & Kembhavi, 2023; Surís et al., 2023). Structured vision-language tasks often take the same form: several local decisions must be combined into one globally consistent answer. We compare two prediction modes. *Global* asks the VLM to predict the complete structured answer directly, whereas *Local* asks for the underlying decisions separately and forms a prediction without the task-specific solver (Fig. 1). Our central question is: if a VLM can make the required local decisions, can it combine them into the correct global structure?

Existing evaluations generally do not tell us whether a model fails because it cannot make the required individual decisions or because it cannot combine those decisions into a correct overall answer. Some benchmarks, such as GUI grounding, evaluate one decision at a time and never test combination. Others require multi-image or temporally structured outputs but do not reveal, when the final answer is wrong, whether the underlying local decisions are themselves wrong or are answerable but fail to be combined correctly. We call the latter case an *assembly deficit*. We use *assembly deficit* as a behavioral notion: it does not assume that the model internally computes the same local decisions during global generation. Language models exhibit an analogous *compositionality gap*, in which solved subproblems fail to compose into the overall solution (Press et al., 2023). Here we study the corresponding vision-language setting, where multiple decisions depend on shared visual evidence and must be combined into one valid structured answer. To determine whether global failure comes from missing local information or from an *assembly deficit*, we hold the model and task input fixed while changing only how the answer is requested. On RecipeQA, the same frozen VLM receives the same ordered recipe text and shuffled step images. Complete-order prediction and scoring remain weak, whereas step–image matching is substantially stronger.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

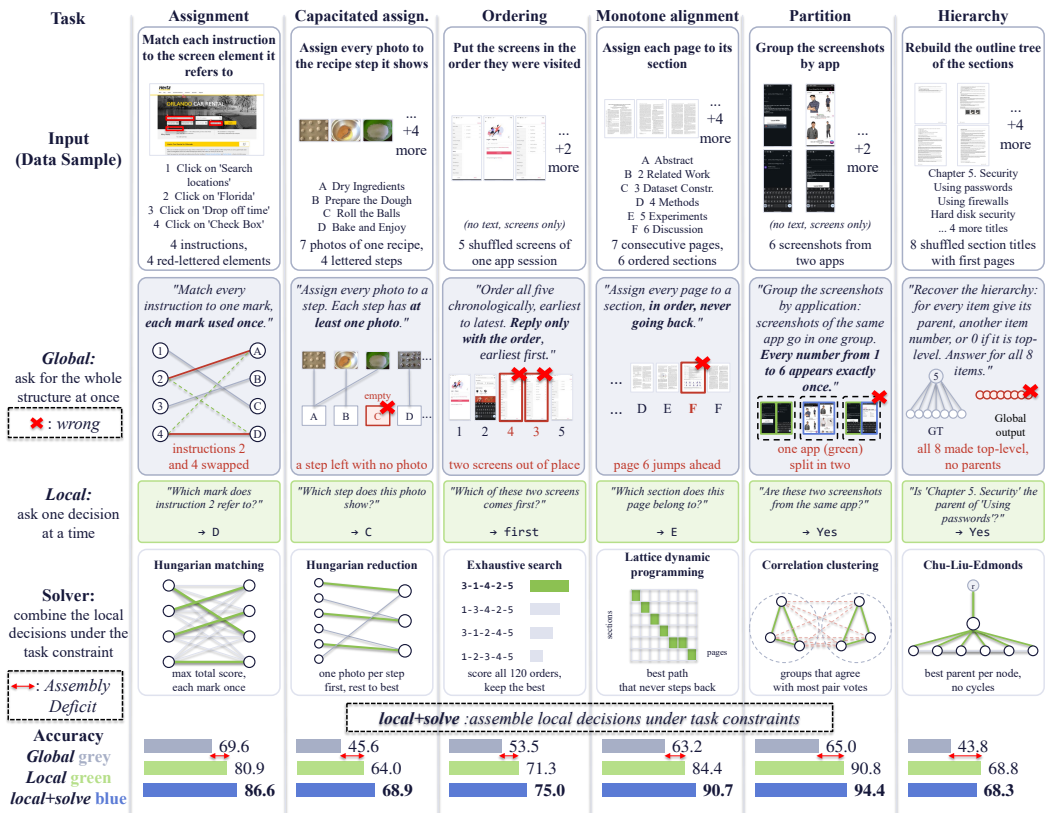



Figure 1: **The *assembly deficit* across six structural families.** Each column shows one task instance evaluated with the same frozen Qwen2.5-VL-7B and no additional instance-specific evidence: the input, the *Global* output, a representative local query and answer, the task-specific solver, and the performance of *Global*, *Local*, and *local+solve* under the family-specific metrics of Tab. 1.

We test this diagnosis with *local+solve*. The frozen VLM provides scores for the underlying local decisions, and a task-specific solver combines them under the structural constraints already defined by the task. Because *local+solve* adds local queries, computation, and task-defined constraints but no new instance-specific evidence, the improvement from *Global* to *local+solve* provides an operational measure of the recoverable component of the *assembly deficit*. Across the main six-family evaluation, all five public tasks, and several GUI-agent settings, *local+solve* outperforms *Global*, including strengthened variants where tested (constrained decoding, self-consistency, and procedure prompting). This procedure thus measures how much of the failure the model's own local decisions can already recover, and combines those decisions into a valid global structure. We further examine when the deficit persists. From 7B to 72B, the deficit narrows but persists in most families. It largely recurs with InternVL3-8B and GLM-4.1V-9B-Thinking as backbones. Post-training on *local+solve*'s outputs nearly closes the gap on the trained task, while training on *local+solve* pseudo-labels spanning three structural families improves global prediction on two unseen structural families. The benefit of external assembly also depends on the quality of the local scores, which an eight-instance AUC probe can check in advance.

Our contributions are threefold:

- We identify the *assembly deficit*: strong local decisions often fail to yield a correct global structure, a failure that accuracy on the complete structure alone cannot reveal.
- We behaviorally isolate this deficit under matched evidence on RecipeQA and measure its recoverable component with *local+solve*, a training-free procedure, across six structural families, five public tasks, and GUI-agent settings.
- We show that assembly can be learned and partly transferred to unseen families, and chart how the deficit varies with scale, local-score informativeness, and backbone.

## 2 Related Work

**Local decisions in existing VLM benchmarks.** Many VLM benchmarks evaluate either individual visual decisions or outputs that depend on several such decisions. GUI-grounding models and agent benchmarks often focus on individual instruction–target or action decisions (Cheng et al., 2024; Wu et al., 2025; Gou et al., 2025; Qin et al., 2025; Lin et al., 2025; Deng et al., 2023; Li et al., 2024), while compositional benchmarks test attributes, relations, and correspondences in image–text understanding (Thrush et al., 2022; Yuksekgonul et al., 2023; Ma et al., 2023b; Hsieh et al., 2023; Zeng et al., 2024). Ordering tasks on procedural and story datasets instead require complete sequences built from multiple underlying decisions (Yagcioglu et al., 2018; Tang et al., 2019; Huang et al., 2016; Agrawal et al., 2016). These benchmarks therefore measure either individual decisions or complete structured outputs, but their results alone do not reveal whether a wrong complete output arises because the underlying decisions are themselves wrong or because the model fails to combine otherwise available decisions correctly.

**Failures in multi-image and multi-step reasoning.** Recent studies show that VLM and language-model performance can degrade when models must jointly reason over several related observations or steps, with performance sensitive to input order, query or output format, and sequence or set size (Wang et al., 2025b; Imam et al., 2025; Ahn et al., 2026; Sethuraman et al., 2026; Lyu et al., 2025; Zhang et al., 2025b; Zhao et al., 2024; Alonso et al., 2025; Anika & Miah, 2025). Related benchmarks evaluate reasoning across multiple images or temporally linked observations, including MMIU (Meng et al., 2025), MuirBench (Wang et al., 2025a), and Vinoground (Zhang et al., 2024b). Together, these studies show that reasoning across multiple observations can be difficult. Our question is complementary: when the answer is wrong, does the model fail on the local decisions themselves, or can it make those decisions but fail to combine them into a global structure?

**Decomposition and structured inference.** Prior work improves complex reasoning by decomposing difficult problems into simpler subproblems (Khot et al., 2023; Zhou et al., 2023), using additional test-time search or aggregation (Wang et al., 2023; Snell et al., 2025; Wang et al., 2025c), or composing models with visual programs, planners, and symbolic solvers (Gupta & Kembhavi, 2023; Surís et al., 2023; Zhang et al., 2024a; Gao et al., 2023; Liu et al., 2023; Pan et al., 2023). Most closely related, Press et al. (2023) study a *compositionality gap* in language models, where the model can answer the individual subquestions correctly but still fail on the overall compositional question. Our setting asks an analogous question for vision-language tasks in which multiple local decisions must be combined into a valid global structure. Beyond using decomposition to improve reasoning, we use it to test whether the required local decisions remain available when global prediction fails. The VLM remains frozen, and *local+solve* combines scores for those decisions using only the structural constraints already defined by the task.

**Structural constraints and post-training.** Prior work incorporates structured output spaces through differentiable relaxations during training (Mena et al., 2018), constrained decoding (Geng et al., 2023), and solver-backed symbolic inference (Orvalho et al., 2026). Recent multimodal post-training methods also improve visual reasoning through verifiable rewards or self-supervised ordering objectives (Liu et al., 2025; Wu et al., 2026). Our primary diagnosis instead keeps the VLM frozen and lets the solver operate only on scores obtained from local queries. We then use post-training in §4.4 to test whether the model can learn to produce the global structure itself and whether this behavior transfers to structural families not seen during training.

## 3 Method

Our framework separates the local decisions that support an answer from the step that combines them into a valid global structure. For a task with a structured output, we query a frozen VLM about task-specific *local decisions*: the individual choices or relations that make up the complete answer. Each query provides a *local score* from the model's answer-token distribution.

We compare three prediction modes. *Global* predicts the complete structure directly. *Local* converts the local scores into a prediction without applying the task-specific solver. *local+solve* applies the solver to the same scores under the task-defined feasible set. Thus, *Local* and *local+solve* differ

only in whether the task-specific structured inference step is applied. We instantiate this framework across six *structural families*: assignment, capacitated assignment, ordering, monotone alignment, partition, and hierarchy. Each family defines its local decisions, feasible set, and task-specific solver.

### 3.1 Problem Formulation

We cast each task as choosing the valid global structure best supported by the local scores:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \sum_{r \in R(y)} s_\theta(r), \tag{1}$$

where $s_\theta(r)$ is the frozen VLM's score for a task-specific local decision $r$, with higher values indicating stronger model support, $\mathcal{Y}$ is the set of globally valid structures defined by the task, and $R(y)$ is the set of local decisions that constitute structure $y$. In RecipeQA ordering, for example, $r$ may be "step 2 matches image B," $s_\theta(r)$ measures the model's support for that match, and $y$ is a one-to-one assignment of recipe steps to images, which determines the image ordering. Eq. (1) separates local scoring from *assembly*: the VLM provides scores for the local decisions, and structured inference combines those scores into a valid structure in $\mathcal{Y}$. The score $s_\theta(r)$ may aggregate multiple queries of the same local decision. For example, RecipeQA scores both directions (App. A.2).

### 3.2 Local Scoring

We obtain local scores from the frozen VLM's answer-token distributions. The form of each local query depends on whether the decision selects among candidates or evaluates a task-specific relation.

For candidate-based decisions, consider $n$ items $x_1, \ldots, x_n$ and $m$ candidates $z_1, \ldots, z_m$ defined by the task instance. Examples include selecting a visual target for an instruction, assigning a photo to a recipe step, or assigning a document page to a section. For each $x_i$, we present the candidates to the model and use its answer-token distribution to score them. We denote by $S[i, c]$ the score for assigning item $x_i$ to candidate $z_c$, where $i \in \{1, \ldots, n\}$ and $c \in \{1, \ldots, m\}$.

For relation-based decisions, the model scores whether a task-specific relation holds between items rather than selecting among candidates. These relations are pairwise in most of our tasks. Given items $x_1, \ldots, x_n$, we query whether the relation holds between $x_i$ and $x_j$, where $i, j \in \{1, \ldots, n\}$. Examples include whether one event precedes another (*ordering*), whether two observations belong to the same group (*partition*), or whether one section is the parent of another (*hierarchy*). We denote the resulting pairwise score by $s(i, j)$. For hierarchy, we additionally score whether each section is a root. Both $S[i, c]$ and $s(i, j)$ are task-specific instances of the local score $s_\theta(r)$ in Eq. (1).

*Local* converts the local scores into a prediction using a simple family-specific rule, without the task-specific solver. For candidate-based tasks, it selects the highest-scoring candidate for each item. For hierarchy, it selects the highest-scoring parent for each section. For ordering, it ranks items by their summed pairwise precedence scores (Borda count), and for partition, it thresholds pairwise scores and takes connected components. *local+solve* uses the same local scores but replaces this simple rule with the task-specific solver in Eq. (1).

Local queries introduce no additional instance-specific evidence: they use only images and text available to *Global*. Because an individual local query may present only the subset relevant to that decision, however, the amount of context can differ from that of *Global*. In §4.2, we control for this difference by presenting the same full context and changing only the requested prediction.

### 3.3 Structural Families

We instantiate Eq. (1) across six structural families, which differ in how local scores must be combined to form a valid global structure. Each family specifies a local decision type, a feasible set $\mathcal{Y}$, and a solver that enforces the corresponding structural constraint. Fig. 1 illustrates one instance of each family, and Tab. 1 compares *Global*, *Local*, and *local+solve*, with monotone alignment evaluated on both document and recipe data. Task-specific definitions, data sizes, local-score construction, and solver details are provided in App. A.1.

RecipeQA provides a concrete example of how the same local scores can support different global structures. Capacitated assignment and monotone alignment use the same photo-to-step score matrix but impose different constraints on the final assignment. Capacitated assignment requires every photo to be assigned and every step to receive at least one photo, but does not constrain the order. Monotone alignment adds an ordering requirement: along the photo sequence, the assigned recipe step never moves backward. Accordingly, the RecipeQA rows in Tab. 1 are two structural formulations of the same data, not independent datasets. For the six controlled families, the task-specific solver finds the exact optimum over the corresponding feasible set. App. A.1 describes the heuristic for long orderings in public benchmarks.

Table 1: Comparison of *Global*, *Local*, and *local+solve* across the six structural families (%, Qwen2.5-VL-7B). *Local* decodes the local scores without the task-specific solver, whereas *local+solve* applies the solver to the same scores. **Bold** marks the best result in each row. Metrics are family-specific (whole-structure accuracy: App. C.2). *Global* prompts follow App. C.4.

| Task | Data | *Global* | *Local* | *local+solve* |
|---|---|---|---|---|
| Assignment | GroundUI | 69.6 | 80.9 | **86.6** |
| Cap. assignment | RecipeQA | 45.6 | 64.0 | **68.9** |
| Ordering | GroundUI | 53.5 | 71.3 | **75.0** |
| Alignment | MMLongBench | 63.2 | 84.4 | **90.7** |
| Alignment | RecipeQA | 61.9 | 64.0 | **80.3** |
| Partition | GroundUI | 65.0 | 90.8 | **94.4** |
| Hierarchy | MMLongBench | 43.8 | **68.8** | 68.3 |

## 4 EXPERIMENTS

We address five questions about the *assembly deficit*. Does local signal remain accessible when global prediction fails (§4.2)? Does externalizing assembly recover the structure, and how much of that pipeline can be automated (§4.3)? Can assembly be internalized by training (§4.4)? Does the deficit persist under scale and stronger global inference, and where does external assembly stop helping (§4.5)? Is the pattern specific to the Qwen backbone (§4.6)?

### 4.1 EXPERIMENTAL SETUP

Unless otherwise noted, all experiments use Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) as the primary backbone. GUI-agent experiments (App. D) additionally evaluate OS-Atlas-Base-7B (Wu et al., 2025), UGround-V1-7B (Gou et al., 2025), UI-TARS-7B-DPO (Qin et al., 2025), and ShowUI-2B (Lin et al., 2025). The model-scale analysis evaluates the 3B, 32B, and 72B Qwen2.5-VL variants, while Qwen3-VL-30B-A3B-Thinking (Bai et al., 2025a) provides an extended-reasoning control. Models remain frozen except for post-training in §4.4.

**Baselines.** We compare against inference-time alternatives designed to address different explanations for poor global prediction. Constrained decoding (Geng et al., 2023) constrains *Global* decoding to valid structures, permutation self-consistency (PSC) (Tang et al., 2024) aggregates predictions across shuffled presentation orders, and self-consistency (Wang et al., 2023) aggregates multiple sampled complete answers. Setwise (Zhuang et al., 2024) constructs an ordering through iterative candidate selection. VQAScore-style yes/no scoring (Lin et al., 2024) uses the same backbone to produce alternative image–text scores, which are passed to the same task-specific solver as *local+solve*. We evaluate each baseline only on tasks for which it is applicable (App. C.3). Benchmarks retain their native metrics, and within-model comparisons use paired McNemar or sign tests.

### 4.2 LOCAL SIGNAL SURVIVES GLOBAL FAILURE

If a model can answer the individual step–image questions but fails to produce the correct overall ordering, changing only the requested output should reveal this difference. We test this with a strict shared-context comparison (Tab. 2). Each instance contains the same four recipe steps in their correct textual order, the same shuffled step images, and the same four candidate image orderings. Only how the answer is requested or scored changes across conditions.

Under *Global*, the model directly selects one of the four candidate orderings and reaches $38.4\%$ accuracy. To test whether this failure is specific to generating the global answer, we use a complete-order scoring control: instead of generating an answer, we assign a likelihood to each of the four candidate orderings and select the highest-scoring one. This still evaluates each ordering as a whole

rather than breaking the task into individual step–image matches. Accuracy remains nearly unchanged at 38.8% (McNemar $p$=.90).

In contrast, when the same context is used to ask which image matches each individual recipe step, the per-step answers reach 70.8% step–image matching accuracy. Using those local answers to select the most consistent candidate ordering raises complete-order accuracy to 73.0%. Thus, the poor *Global* performance cannot be explained solely by errors in the underlying local decisions: under the same context, those local answers combine into a much more accurate complete ordering. Complete-order accuracy can exceed the per-step rate, a different unit, because the chosen ordering need only agree with more answers than the other three candidates, so a single wrong match need not change it. Asking the per-step questions first in the same context lowers *Global* to 34.3% (App. C.1).

Table 2: **Shared-context diagnosis on RecipeQA** (Qwen2.5-VL-7B). The model, images, and recipe text are fixed, and only how the answer is requested or scored changes.

| Condition | Unit | Acc. |
|---|---|---|
| *Global* | complete ordering | 38.4 |
| Order scoring | complete ordering | 38.8 |
| Step answers | step–image match | 70.8 |
| Step answers + assembly | complete ordering | 73.0 |

Table 3: Results on five public tasks from three suites (MMIU-T/V/J: temporal, visual, and spatial-jigsaw ordering). Values are percentages under native metrics. Methods below the divider share the Qwen2.5-VL-7B backbone and task instance. MATE's *local+solve* lists the objects once and binds them (no separate *Local* prediction). The italicized row is a published reference, not a matched comparison. The reference models are GPT-4o (MMIU-T), Gemini-1.5 (MMIU-V/J), and Claude 3.5 Sonnet (MATE), with sources in App. E.3. *Four-step context (App. E.3). Full official context: 36.4 (*Global*), 88.5 (*local+solve*). Dashes mark inapplicable (App. E.3) or unavailable cells.

| | *image-only* | | | *text–image* | |
|---|---|---|---|---|---|
| Method | MMIU-T 4-way | MMIU-V 4-way | MMIU-J 4-way sp. | RecipeQA* 4-way | MATE free-form |
| Uniform chance | 25.0 | 25.0 | 25.0 | 25.0 | — |
| *Benchmark-paper best VLM* | *28.0* | *43.5* | *34.5* | — | *83.3* |
| *Global* | 21.5 | 33.0 | 22.0 | 38.4 | 80.2 |
| PSC | 24.0 | 34.5 | 21.5 | 53.3 | — |
| Setwise | 32.0 | 44.5 | — | 52.5 | — |
| VQAScore-style scoring + same solver | — | — | — | 90.4 | — |
| ***local+solve*** | **51.0** | **73.5** | **42.0** | **93.3** | **89.6** |

### 4.3 Externalizing Assembly Narrows the Deficit

If the deficit lies in producing the joint structured output, externalizing assembly should recover performance. We test this across the six structural families and on MMIU, RecipeQA, and MATE. All matched comparisons use the same frozen primary backbone and task instance, and local queries may show decision-relevant subsets of it (§3.2). Because *local+solve* uses multiple local queries, we additionally compare against *Global* given a matched number of generation samples (App. C.3).

Across all five public tasks, *local+solve* improves over *Global* (Tab. 3). On RecipeQA's full official context, accuracy rises from 36.4% to 88.5% (38.4% to 93.3% with the four-step context of §4.2). Scoring in the inverse direction (each image against the steps, App. A.2) partly explains why 93.3% exceeds the 73.0% of the shared-context probe in §4.2 (Tab. 9). A VQAScore-style scorer passed to the same solver stays below our bidirectional local scoring (Tab. 3). The same qualitative pattern extends to several GUI-agent settings (App. D). The effect is not universal across model scales: on MATE, the gain holds at 3B and 7B but reverses at 72B (App. D.3). At 7B, the MATE gain grows with object count, from +1.2pp at three objects to +14.6pp at ten.

**Where does the recovery come from?** The three prediction modes separate two stages of recovery from poor global prediction. The change from *Global* to *Local* reflects querying the local decisions separately, before the solver (§3.2). The change from *Local* to *local+solve* isolates the additional effect of applying the task-specific solver to the same local scores. In six of the seven rows of Tab. 1, the *Global*-to-*Local* gain is larger than the additional *Local*-to-*local+solve* gain.

Table 4: **Effect of the task-specific solver** (Qwen2.5-VL-7B, controlled settings). Gain from applying the task-specific solver over a comparator using the same local scores. For alignment, the comparator is an order-free flow decode rather than *Local*, isolating the effect of requiring assignments to follow the sequence without moving backward.

| Family | Constraint or objective | Compared with | Gain (pp) |
|---|---|---|---|
| Assignment | one-to-one matching | row-wise argmax (*Local*) | $+5.7$ |
| Capacitated assignment | every step used | best step per photo (*Local*) | $+4.9$ |
| Ordering | exact order search | Borda ranking (*Local*) | $+3.7$ |
| Partition | correlation clustering | connected components (*Local*) | $+3.6$ |
| Recipe alignment | sequence-order constraint | order-free flow decode | $+11.5$ |
| Document alignment | sequence-order constraint | order-free flow decode | $+5.5$ |
| Hierarchy (outlines) | valid forest | best parent per section (*Local*) | $-0.5$ (ties) |

Equivalently, in most settings, more than half of the total improvement from *Global* to *local+solve* is already recovered before the task-specific solver is applied.

The additional benefit of the solver varies across structural families (Tab. 4). For assignment, capacitated assignment, ordering, and partition, applying the task-specific solver adds $+3.6$ to $+5.7$pp over *Local*. For alignment, where the comparator is an order-free flow decode, requiring assignments to follow the sequence without moving backward adds $+11.5$pp on recipes and $+5.5$pp on documents. Hierarchy is the opposite case: the parent choices produced by *Local* already form valid forests on all 84 evaluated outlines, so the solver only changes how exact score ties are broken. Overall, the solver helps most when independently decoded local decisions conflict with the task's constraints.

**Solver selection can be automated within a fixed library.** Given only a paraphrased task description, Qwen2.5-VL-32B and 72B select the solver used by our formulation on $100.0\%$ and $99.3\%$ of decisions, with routing errors reducing macro-average accuracy by at most $0.1$pp (App. A.4). This automates only the solver choice, not the structural formulation or the solver library.

### 4.4 Can Assembly Be Internalized?

**Self-distillation internalizes assembly on the trained task.** We next ask whether assembly can be learned and whether what is learned transfers. Using *local+solve*'s outputs as pseudo-labels for global prediction on $2{,}000$ RecipeQA items, without ground-truth labels, raises accuracy on the $929$ RecipeQA validation questions (four-step context) from $38.4\%$ to $93.0\%$ (Tab. 5), nearly matching *local+solve* at $93.3\%$. Two controls at the same budget bracket this result: ground-truth labels reach $93.3\%$, while training on *Global*'s own answers reaches only $41.3\%$, so the gain comes from the higher-quality *local+solve* pseudo-labels rather than exposure to the task format. Other post-training regimes show no significant far-transfer gains on general benchmarks (App. F.3).

**At a fixed budget, a three-family mixture transfers better than an assignment-only one.** We compare an assignment-only adapter with one trained jointly on assignment, partition, and ordering, using $336$ *local+solve*-labeled examples in both cases ($112$ from each family for the three-family adapter). The three-family adapter improves *Global* prediction on both unseen structural families, RecipeQA alignment and MMLongBench hierarchy (Tab. 5), and outperforms the assignment-only adapter on both in every seed (App. F.2), whereas the assignment-only adapter falls below the frozen model on average. Controls trained on the same items with ground-truth or *Global* pseudo-labels transfer less. On RecipeQA alignment, *local+solve* remains nearly unchanged across the adapters (App. F.2), so the gain lies in global prediction, not in decoded local accuracy.

### 4.5 When Does the Deficit Persist?

**Scaling narrows but does not consistently eliminate the *assembly deficit*.** Across 3B, 7B, 32B, and 72B, the trend is not strictly monotonic (Fig. 2, one setting per family, with recipe alignment only in Tab. 24). From 7B to 72B, however, the gap decreases in every setting and remains significant at 72B except on MMLongBench hierarchy and recipe alignment. In each family, *Global* gains more ($+6.9$ to $+29.3$pp) than *local+solve* ($-0.4$ to $+9.7$pp, Tab. 24), so *Global* drives the narrowing.

Table 5: **Global prediction after self-distillation on *local+solve* outputs** (Qwen2.5-VL-7B). Task-specific adapters train on the same 2,000 RecipeQA ordering items with ground-truth labels, *Global*'s own answers (43.5% correct), or *local+solve*'s outputs (94.4% correct). The third block is a separate run with the *Global* pseudo-label control. Transfer adapters use 336 examples from one family (assignment) or three (assignment, partition, ordering), and the three-family controls reuse these items (3 to 6 replaced for *Global* pseudo-labels). Transfer cells are mean ± sd over five seeds (three for controls). *Four-step context. –: not run.

| | Trained task | Unseen families | |
|---|---|---|---|
| Adapter (training data) | RecipeQA ordering* | RecipeQA align. | MMLongBench hier. |
| Frozen (no adaptation) | 38.4 | 61.9 | 43.8 |
| Task-specific, ground-truth labels | 93.3 | – | – |
| Task-specific, *local+solve* pseudo-labels | **93.0** | – | – |
| Frozen, separate run | 38.5 | – | – |
| Task-specific, *local+solve* pseudo-labels | 92.0 | – | – |
| Task-specific, *Global* pseudo-labels | 41.3 | – | – |
| One family, *local+solve* pseudo-labels | – | 48.5 ± 7.1 | 40.8 ± 3.4 |
| Three families, ground-truth labels | – | 60.9 ± 9.3 | 49.7 ± 3.7 |
| Three families, *Global* pseudo-labels | – | 59.3 ± 3.7 | 45.0 ± 1.8 |
| Three families, *local+solve* pseudo-labels | – | **69.0** ± 2.9 | **54.3** ± 4.1 |

Table 6: **Stronger global prediction** (Qwen2.5-VL-7B, %). Compute-matched: one sample per *local+solve* query, whole-structure vote. The *Global* row and its variants use the prompts of Tab. 1, and the procedure row uses no in-prompt example. *Four-step context (App. E.3).

| | Assign. | Ordering | Partition | Align. rec. | Align. doc. | RecipeQA* |
|---|---|---|---|---|---|---|
| *Global* | 69.6 | 53.5 | 65.0 | 61.9 | 63.2 | 38.4 |
| + constrained decoding | 69.8 | 53.5 | 65.3 | 68.0 | 51.2 | 38.4 |
| + self-consistency (5 samples) | 68.2 | 52.9 | 67.8 | 66.6 | 62.8 | 36.5 |
| + self-consistency (compute-matched) | 67.3 | 52.8 | 68.6 | 61.2 | 66.4 | 36.9 |
| + procedure in the prompt | 45.7 | 50.2 | 69.1 | 64.5 | 37.0 | 60.4 |
| *local+solve* | **86.6** | **75.0** | **94.4** | **80.3** | **90.7** | **93.3** |

At 32B, *Global* is significantly ahead on hierarchy and recipe alignment per item, but not per window or per recipe, so neither reversal is robust (App. E).

**Stronger global inference does not close the gap.** We test whether the deficit can be explained by a weak *Global* inference procedure rather than by assembly itself. Self-consistency, compute-matched repeated generation, constrained decoding, and describing the full procedure in the prompt close at most 40% of the gap, and the strongest variant in every setting remains significantly below *local+solve* (Tab. 6, App. C.3). Even the largest improvement, on RecipeQA, raises *Global* from 38.4% to 60.4%, still well below *local+solve* at 93.3%. With or without the prompt's format example, no gap changes sign (App. C.4).

**The benefit of external assembly depends on local-score informativeness.** The task-specific solver in *local+solve* can only work with the local scores supplied by the VLM. We use the area under the ROC curve (AUC) as a task-agnostic measure of local-score informativeness because it measures how often a correct local choice scores above an incorrect one, without requiring a decision threshold on the scores or a calibrated score scale (an AUC of 0.5 is chance).

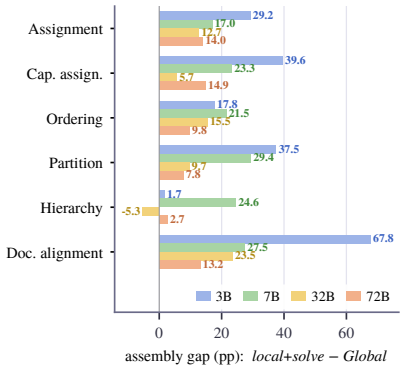



Figure 2: ***Assembly deficit* across model scale.** *local+solve*−*Global* gap for Qwen2.5-VL 3B–72B, one setting per family. From 7B to 72B, the gap narrows in every setting and remains significant in five.

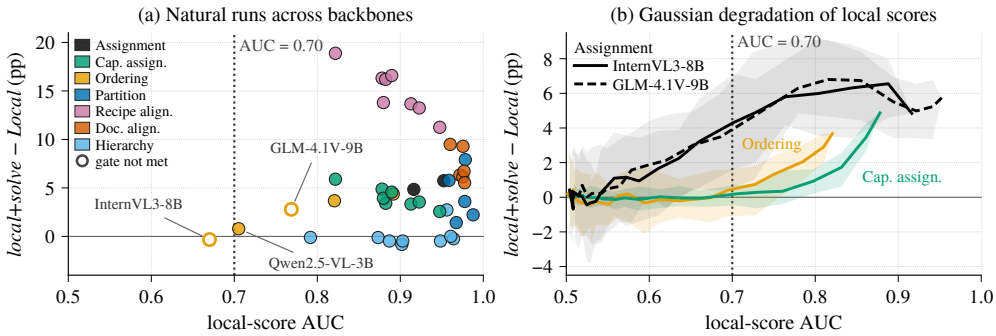



Figure 3: **External-assembly gain versus local-score informativeness. (a)** Natural runs across backbones and controlled settings (43 points from 35 scoring runs), with the three weak ordering runs labelled. Open markers fail the pre-run gate. **(b)** Gaussian degradation for assignment, capacitated assignment, and ordering, where the gain over *Local* approaches zero as the local scores become uninformative. Dotted lines mark the pre-specified AUC threshold of 0.70.

We screen new extensions and backbone replications with a pre-specified *local-signal gate*: an eight-instance probe must reach AUC $\geq 0.70$. Failing extensions are stopped or kept as negative controls, while failing backbone runs are flagged (App. A.3). A noise sweep places the zero-gain crossings at AUC 0.57–0.73 for assignment, capacitated assignment, and ordering (Fig. 3b). Natural runs show the same trend: the three weak ordering scorers (AUC 0.67–0.77) gain at most +2.8pp (Fig. 3a). Thus, 0.70 serves as a practical screening threshold rather than a universal boundary, and a small probe can still yield a false negative (App. E.2).

### 4.6 Is the Assembly Deficit Backbone-Specific?

Across Qwen2.5-VL-7B, InternVL3-8B (Zhu et al., 2025), and GLM-4.1V-9B-Thinking (GLM-V Team, 2025b), with the same task instances, local queries, solvers, and metrics, *local+solve* significantly outperforms *Global* in 20 of the 21 backbone–setting pairs (seven controlled task–dataset settings, three backbones, item-level McNemar $p < 10^{-3}$). The sole exception is GLM on recipe alignment, where *Global* scores higher. On capacitated assignment, all three backbones have the same full-run local-score AUC of 0.88, while *Global* accuracy ranges from 24.5% to 67.2%. Thus, local-score informativeness alone does not explain the variation in global prediction (App. E.2).

## 5 Conclusion

VLMs can be right in parts but wrong overall. We identify this *assembly deficit* across vision-language tasks: models can make strong local decisions while failing to combine them into the required global structure. In six of seven settings, querying the decisions separately recovers most of the gap before the task-specific solver is applied. Post-training on *local+solve* outputs nearly internalizes assembly on the trained task, while multi-family *local+solve* pseudo-labels improve *Global* prediction on two unseen structural families. From 7B to 72B, the deficit narrows but persists in most families. Across Qwen2.5-VL-7B, InternVL3-8B, and GLM-4.1V-9B-Thinking, *local+solve* significantly outperforms *Global* in 20 of 21 backbone–setting pairs. These results point to assembly as a distinct, learnable capability, and local and global performance provide complementary information. Global performance alone does not reveal whether an error comes from the underlying local decisions or from combining them, while local performance alone does not test whether those decisions can be assembled into the required structure.

**Limitations.** Our framework is designed for tasks with a task-defined feasible set $\mathcal{Y}$ and requires specifying the local decisions used for scoring. We demonstrate our strictest shared-context control on RecipeQA. For the other structural families, local queries use only information from the same task instance but may present the subset relevant to each decision. These experiments establish the local–global separation behaviorally but do not identify the internal mechanism that produces it. Finally, the inference cost of *local+solve* grows with structure size.

### ETHICS STATEMENT

This work uses open-weight models and public datasets, whose stated terms are listed in App. A.6. We use the Voxel51 distribution of GroundUI-18k, whose dataset card lists training and fine-tuning models for UI element localization among its direct uses. The dataset card of the original AgentStudio release describes GroundUI-18k as a test set and asks users not to use it for training. Our transfer experiments use at most 408 GroundUI-derived training examples per adapter (sizes per experiment in App. F). These adapters are not released. We do not evaluate them under, or claim results for, the original GroundUI-18k benchmark protocol. GroundUI-derived examples are used only in our controlled transfer analyses, and the one training–evaluation overlap (19 ScreenSpot screens) is excluded in App. A.6, which leaves the result essentially unchanged.

### REPRODUCIBILITY STATEMENT

All experiments use open-weight VLMs and public datasets or benchmarks, whose splits, evaluated sizes, and stated terms are listed in App. A.6. Public benchmark comparisons use their native metrics, while the controlled probes use the family-specific metrics defined in App. A.1. Full task definitions and solver specifications are provided in App. A.1, prompts in App. I, and post-training settings in App. F. We will release per-item predictions, stored local score matrices, prompts, and the reproduction and analysis scripts, including those for the backbone-replication and gate-sensitivity experiments. Experimental protocols were recorded with timestamps before evaluation, and we will release them with the code. We therefore use the term "pre-specified" rather than "preregistered."

### AI USE STATEMENT

We used generative AI assistants for several aspects of this work. They assisted with drafting and editing parts of the text for readability and terminology consistency, with implementing and revising parts of the experimental pipeline, analysis code, and figure-generation scripts, and with searching for, summarizing, formatting, and checking related literature and references. For experimental analysis, they assisted with checking analysis scripts and organizing stored experimental outputs. All prompts used for model evaluation were reviewed and finalized by the authors before evaluation. The authors conceived the research question, developed the research methodology, defined the experimental protocols, and made all final conceptual, methodological, experimental, analytical, and reporting decisions. Translation and dataset cleaning or reformatting were not AI-assisted.

The authors reviewed all AI-assisted work. Every numerical result from our experiments was produced by scripts that we re-ran from stored per-item outputs, and an automated validation script checks manuscript values against a registry of recorded results. AI-assisted experimental code was validated by first reproducing the previously recorded Qwen2.5-VL-7B results before being applied to additional backbones. Every cited reference was checked against its publisher or official repository page. The authors take full responsibility for the final content of this work.

## REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. TallyQA: Answering complex counting questions. In *AAAI*, 2019.

Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. Sort Story: Sorting jumbled images and captions into stories. In *EMNLP*, 2016.

Daechul Ahn, Yura Choi, Hyeonbeom Choi, Seongwon Cho, San Kim, and Jonghyun Choi. What happens when: Learning temporal orders of events in videos. In *WACV*, 2026.

Iñigo Alonso, Gorka Azkune, Ander Salaberria, Jeremy Barnes, and Oier Lopez de Lacalle. Vision-language models struggle to align entities across modalities. In *Findings of ACL*, pp. 18846–18862, 2025.

Adrita Anika and Md Messal Monem Miah. Evaluating LLMs' reasoning over ordered procedural steps. In *Findings of IJCNLP-AACL*, pp. 2259–2267, 2025. doi: 10.18653/v1/2025.findings-ijcnlp.139. URL https://aclanthology.org/2025.findings-ijcnlp.139/.

Anthropic. Introducing Claude 3.5 Sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, 2024.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1–3):89–113, 2004.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *NeurIPS*, 2024.

Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024.

Yoeng-Jin Chu and Tseng-Hong Liu. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14: 1396–1400, 1965.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2Web: Towards a generalist agent for the web. In *NeurIPS*, 2023.

Jack Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71B(4): 233–240, 1967.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models. In *ICLR*, 2024.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *ICML*, 2023.

Gemini Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. Grammar-constrained decoding for structured NLP tasks without finetuning. In *EMNLP*, 2023.

GLM-V Team. GLM-4.6V: Open source multimodal models with native tool use. `https://z.ai/blog/glm-4.6v`, 2025a.

GLM-V Team. GLM-4.5V and GLM-4.1V-Thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025b.

Boyu Gou, Ruohan Wang, Boyuan Zheng, Yanan Xie, Cheng Chang, Yiheng Shu, Huan Sun, and Yu Su. Navigating the digital world as humans do: Universal visual grounding for GUI agents. In *ICLR*, 2025.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, pp. 14953–14962, 2023.

James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982. doi: 10.1148/radiology.143.1.7063747.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2): 65–70, 1979.

David W. Hosmer and Stanley Lemeshow. *Applied Logistic Regression*. Wiley, 2nd edition, 2000.

Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. SugarCrepe: Fixing hackable benchmarks for vision-language compositionality. In *NeurIPS*, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *NAACL*, 2016.

Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. A data-driven approach for learning to control computers. In *ICML*, pp. 9466–9482, 2022.

Mohamed Fazli Imam, Chenyang Lyu, and Alham Fikri Aji. Can multimodal LLMs do visual temporal understanding and reasoning? The answer is No! *arXiv preprint arXiv:2501.10674*, 2025.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *ICLR*, 2023.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. Language models can solve computer tasks. In *NeurIPS*, 2023.

Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2 (1-2):83–97, 1955.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. ScreenSpot-Pro: GUI grounding for professional high-resolution computer use. In *ACM MM*, 2025.

Wei Li, William Bishop, Alice Li, Chris Rawles, Folawiyo Campbell-Ajala, Divya Tyamagundlu, and Oriana Riva. On the effects of data scale on UI control agents. In *NeurIPS*, 2024.

Kevin Qinghong Lin, Linjie Li, Difei Gao, Zhengyuan Yang, Shiwei Wu, Zechen Bai, Stan Weixian Lei, Lijuan Wang, and Mike Zheng Shou. ShowUI: One vision-language-action model for GUI visual agent. In *CVPR*, 2025.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *ECCV*, 2024.

Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. LLM+P: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. In *ICLR*, 2018.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *ICCV*, pp. 2034–2044, 2025.

Zesen Lyu, Dandan Zhang, Wei Ye, Fangdi Li, Zhihang Jiang, and Yao Yang. Jigsaw-Puzzles: From seeing to understanding to reasoning in vision-language models. In *EMNLP*, 2025.

Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. HRDoc: Dataset and baseline method toward hierarchical reconstruction of document structures. In *AAAI*, 2023a.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *NeurIPS*, 2024.

Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. CREPE: Can vision-language foundation models reason compositionally? In *CVPR*, 2023b.

Gonzalo E. Mena, David Belanger, Scott W. Linderman, and Jasper Snoek. Learning latent permutations with Gumbel-Sinkhorn networks. In *ICLR*, 2018.

Fanqing Meng, Jin Wang, Chuanhao Li, Quanfeng Lu, Hao Tian, Tianshuo Yang, Jiaqi Liao, Xizhou Zhu, Jifeng Dai, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. MMIU: Multimodal multi-image understanding for evaluating large vision-language models. In *ICLR*, 2025.

Atsuyuki Miyai, Jingkang Yang, Jingyang Zhang, Yifei Ming, Qing Yu, Go Irie, Yixuan Li, Hai Helen Li, Ziwei Liu, and Kiyoharu Aizawa. Unsolvable problem detection: Robust understanding evaluation for large multimodal models. In *ACL*, 2025.

OpenAI. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Pedro Orvalho, Marta Kwiatkowska, Guillem Alenyà, and Felip Manyà. Reliable reasoning with large language models via preference-based maximum satisfiability. *arXiv preprint arXiv:2605.29687*, 2026.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of EMNLP*, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, pp. 5687–5711, 2023.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. UI-TARS: Pioneering automated GUI interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Pritish Sahu, Karan Sikka, and Ajay Divakaran. Challenges in procedural multimodal machine comprehension: A novel way to benchmark. In *WACV*, 2022.

T. V. Sethuraman, Savya Khosla, Aditi Tiwari, Vidya Ganesh, Rakshana Jayaprakash, Aditya Jain, Vignesh Srinivasakumar, Onkar Kishor Susladkar, Srinidhi Sunkara, Aditya Shanmugham, Rakesh Vaideeswaran, Abbaas Alif Mohamed Nishar, Simon Jenni, Rohan Maheshwari, and Derek Hoiem. Stress tests REVEAL fragile temporal and visual grounding in video-language models. *arXiv preprint arXiv:2602.11244*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of Bits: An open-domain platform for web-based agents. In *ICML*, pp. 3135–3144, 2017.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *ICLR*, 2025.

Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via Python execution for reasoning. In *ICCV*, pp. 11888–11898, 2023.

Raphael Tang, Crystina Zhang, Xueguang Ma, Jimmy Lin, and Ferhan Ture. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *NAACL*, 2024.

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. MuirBench: A comprehensive benchmark for robust multi-image understanding. In *ICLR*, 2025a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *NeurIPS*, 2024a.

Xiaochen Wang, Heming Xia, Jialin Song, Longyu Guan, Qingxiu Dong, Rui Li, Yixin Yang, Yifan Pu, Weiyao Luo, Yiru Wang, Xiangdi Meng, Wenjie Li, and Zhifang Sui. Beyond single frames: Can LMMs comprehend implicit narratives in comic strip? In *Findings of EMNLP*, pp. 6436–6452, 2025b. doi: 10.18653/v1/2025.findings-emnlp.342. URL https://aclanthology.org/2025.findings-emnlp.342/.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. "My answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of ACL*, 2024b.

Xiyao Wang, Zhengyuan Yang, Linjie Li, Hongjin Lu, Yuancheng Xu, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Scaling inference-time search with vision value model for improved visual comprehension. In *ICCV*, pp. 1173–1184, 2025c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Penghao Wu, Yushan Zhang, Haiwen Diao, Bo Li, Lewei Lu, and Ziwei Liu. Visual Jigsaw post-training improves MLLMs. In *ICLR*, 2026.

Zhiyong Wu, Zhenyu Wu, Fangzhi Xu, Yian Wang, Qiushi Sun, Chengyou Jia, Kanzhi Cheng, Zichen Ding, Liheng Chen, Paul Pu Liang, and Yu Qiao. OS-ATLAS: Foundation action model for generalist GUI agents. In *ICLR*, 2025.

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes. In *EMNLP*, 2018.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. *arXiv preprint arXiv:2310.11441*, 2023.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023.

Yunan Zeng, Yan Huang, Jinjin Zhang, Zequn Jie, Zhenhua Chai, and Liang Wang. Investigating compositional challenges in vision-language models for visual grounding. In *CVPR*, 2024.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. VideoLLaMA 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025a.

Haowei Zhang, Jianzhe Liu, Zhen Han, Shuo Chen, Bailan He, Volker Tresp, Zhiqiang Xu, and Jindong Gu. Visual question decomposition on multimodal large language models. In *Findings of EMNLP*, 2024a.

Jianrui Zhang, Mu Cai, and Yong Jae Lee. Vinoground: Scrutinizing LMMs over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*, 2024b.

Jianshu Zhang, Dongyu Yao, Renjie Pi, Paul Pu Liang, and Yi R. Fung. VLM2-Bench: A closer look at how well VLMs implicitly link explicit matching visual cues. In *ACL*, 2025b.

Yuanhan Zhang, Bo Li, Haotian Liu, Yong Jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model. `https://llava-vl.github.io/blog/2024-04-30-llava-next-video/`, 2024c.

Bingchen Zhao, Yongshuo Zong, Letian Zhang, and Timothy Hospedales. Benchmarking multi-image understanding in vision and language models: Perception, knowledge, reasoning, and multi-hop reasoning. *arXiv preprint arXiv:2406.12742*, 2024.

Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng Yan. AgentStudio: A toolkit for building general virtual agents. In *ICLR*, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models. In *ICLR*, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *SIGIR*, 2024.

## NOTATION AND ABBREVIATIONS

Tab. 7 collects the principal notation and abbreviations used in the main text and the appendix.

Table 7: **Notation and abbreviations.** The last column shows where each item is defined or first used.

| Symbol or term | Meaning | Defined in |
|---|---|---|
| *Prediction modes and quantities* | | |
| VLM, GUI | vision-language model, graphical user interface | §1 |
| *Global* (G) | the VLM predicts the complete structure directly from the task input | §3 |
| *Local* | the local scores are decoded without the task-specific solver (row-wise argmax, Borda ranking, thresholded connected components, or best parent per section) | §3, Tab. 4 |
| *local+solve* (*l+s*) | the task-specific solver is applied to the same local scores under the feasible set | §3 |
| *assembly deficit* | local questions answered accurately while the global structure is wrong, a behavioral notion | §1 |
| gap | *local+solve* minus *Global*, in percentage points (pp) | §4.3, Fig. 2 |
| *Formulation and scores* | | |
| $r$, $s_\theta(r)$ | a task-specific local decision and the frozen VLM's score for it | Eq. (1) |
| $y$, $\hat{y}$, $\mathcal{Y}$, $R(y)$ | a global structure, the prediction, the task-defined feasible set, and the local decisions that make up $y$ | Eq. (1) |
| $x_1, \ldots, x_n$ and $z_1, \ldots, z_m$ | the items and the candidates of one instance | §3.2 |
| $S[i, c]$ | score for assigning item $x_i$ to candidate $z_c$ | §3.2 |
| $s(i, j)$ | score that a pairwise relation (precedence, same group, parent) holds between $x_i$ and $x_j$ | §3.2 |
| $C$ | RecipeQA score matrix combining forward and inverse queries | App. A.2 |
| $L$ | number of steps and images of a public ordering item, or of objects in a MATE scene | Apps. A.2, D.3 |
| $k$, $m$ | items and candidates per instance in the appendix (the $n$ and $m$ of §3.2), such as instructions, photos, or pages versus targets, steps, sections, or marks. $m$ also counts PSC presentations, and $k$ also indexes answer tokens in App. A.2 | App. A.1 |
| $n{=}\ldots$ | sample size (not the $n$ of §3.2) | per table |
| CI, sd | confidence interval, standard deviation | per table |
| *Evaluation* | | |
| AUC | area under the ROC curve of local scores for correct versus incorrect local choices | §4.5, App. A.3 |
| local-signal gate | pre-specified screen requiring AUC $\geq 0.70$ on an eight-instance probe | §4.5, App. A.3 |
| McNemar, sign test, Holm | paired item-level test, paired instance-level test, multiplicity correction | App. B |
| *Baselines and training* | | |
| PSC, CoT | permutation self-consistency, chain-of-thought | §4.1, App. C.3 |
| $T$, top-$p$, $N$ | sampling temperature, nucleus threshold, and number of samples for self-consistency | Apps. C.3, I |
| LoRA, SFT, GRPO | low-rank adaptation, supervised fine-tuning, group relative policy optimization | App. F |
| *local+solve* or *Global* pseudo-labels | training targets produced by *local+solve* or by *Global*'s own answers | §4.4 |
| D1, D3 | adapters trained with the same budget on assignment only or on assignment, partition, and ordering | App. F.2 |
| Set-of-Mark | letter labels rendered on GUI targets | App. D.1 |
| DP | dynamic programming (lattice DP for monotone alignment) | App. A.1 |
| ARI | adjusted Rand index for partitions | App. F.4 |

# A Additional Method and Evaluation Details

## A.1 Task and solver definitions

The controlled probes use the local-scoring interface of §3.2. Bidirectional assignment scoring is used only for the public RecipeQA ordering task (App. A.2). App. I provides the exact prompts.

Tab. 8 summarizes the local variable, feasible set, solver, metric, and solver-free *Local* decoding for each controlled formulation. Throughout, $S[i, c]$ denotes an assignment score and $s(i, j)$ a pairwise relation score. The model never sees ground truth. It is used for dataset construction and evaluation and, for experiments governed by the local-signal protocol, only for the pre-specified screening decision of App. A.3.

**Assignment.** Hungarian matching (Kuhn, 1955) returns the highest-scoring bijection, whereas *Local* selects each target independently and may therefore reuse a target. Assignment queries each instruction in one direction (instruction to mark).

**Capacitated assignment.** We solve the many-to-one objective exactly by reducing it to a rectangular assignment problem solved with the Hungarian algorithm. Hungarian matching on the relative scores $S[i, j] - \max_l S[i, l]$ selects one distinct photo for each step, and all remaining photos take their highest-scoring step.

**Ordering.** We antisymmetrize the two presentation orders as

$$w_{ij} = \frac{s(i, j) - s(j, i)}{2},$$

and exhaustively select the permutation $\pi$ maximizing $\sum_{a<b} w_{\pi_a, \pi_b}$ over the $5! = 120$ orders. *Local* uses the corresponding Borda ranking. The raw pairwise decisions (the sign of $w_{ij}$) score $74.8\%$, against $75.0\%$ for *local+solve*. Longer public sequences use Borda initialization with insertion refinement (App. I).

**Monotone alignment.** A lattice DP allows each successive item either to remain in the current step/section or advance to the next, with boundary conditions requiring every block to be non-empty. Backtracking recovers the optimal alignment. Document windows with equal numbers of pages and sections have only one feasible monotone cover and are excluded (17 of 58, see *Controlled dataset construction* below).

**Partition.** Using pairwise same-group log-odds $s(i, j)$ as edge weights, we exhaustively solve correlation clustering (Bansal et al., 2004), selecting the partition that maximizes the sum of scores within groups. *Local* instead thresholds the pair scores at zero and takes connected components, so a single false-positive edge can merge two groups. The raw pairwise decisions (same-group log-odds above zero) score $93.7\%$, against $94.4\%$ for *local+solve*.

**Hierarchy.** We remove leading section numbers of the form "1.2" from the listed titles and use Chu–Liu–Edmonds (Chu & Liu, 1965; Edmonds, 1967) with a virtual root to obtain a valid forest, allowing multiple top-level sections. This does not remove numbering as a cue. The pattern misses some numbered titles (for example "C.1" or "1.1Installation"), keeps words such as "Chapter 1", and the page renders that both modes receive show numbered headings. Both modes receive the same section titles and corresponding page content. On HRDoc-Simple, however, the *Global* renders are lower-resolution, as described below. The *Global* prompt attaches the $n$ first-page renders in the same order as the titles, so both modes share a page-identity cue: $207/578$ ground-truth children start on their parent's page, where the pairwise query sees two identical renders (*Local* parent accuracy $73\%$ on these children vs. $49\%$ on the others). For MMLongBench (Ma et al., 2024), the target tree is taken from the embedded two-level PDF outline. For HRDoc-Simple (Ma et al., 2023a), we derive a hierarchy task from the released section classes and `contain`/`equality` relations: second-level sections are assigned to their annotated parent, consecutive top-level sections are grouped into 6–12-node windows, and each node is represented by the page containing its heading. This yields 187 windows and 1,896 items. This is a derived task from the official annotations, not HRDoc's original semantic-unit or layout task. On HRDoc-Simple the *Global* prompt receives the page renders capped at $401{,}408$ pixels, whereas the local queries use the processor's default resolution (about $0.49$M pixels for these $596\times842$ and $612\times792$ pages), so *Global* sees about $80\%$ of the pixels that the local queries see. On MMLongBench, both modes use the default resolution.

**Hierarchy *Global* scoring.** The pre-specified hierarchy metric parses each *Global* output as a whole window. A single missing or malformed `item:parent` line makes the window invalid, and an invalid window scores zero on all of its items. Lines that name an item as its own parent (for example `5:5`) are dropped by the parser, so they also invalidate the window. At 7B, $8/84$ MMLongBench and $44/187$ HRDoc-Simple windows are invalid, 5 and 28 of them because of a self-parent line. Every table keeps the pre-specified window-level metric. Scored item by item instead (each parsed line counts for its own item, and a self-parent line or a missing

Table 8: **Task and solver definitions** for the six structural families. Some families are instantiated on multiple dataset settings, described below. *Local* denotes solver-free decoding of the same local scores used by *local+solve*.

| Family | Local variable | Feasible set | Solver | Metric | *Local* decoding |
|---|---|---|---|---|---|
| Assignment | $S \in \mathbb{R}^{k\times k}$: instruction–target scores | bijections | Hungarian | item | best target per instruction |
| Capacitated assignment | $S \in \mathbb{R}^{k\times m}$: photo–step scores | many-to-one; every step non-empty | Hungarian reduction | item | best step per photo |
| Ordering | $s(i,j)$: pairwise precedence scores | permutations | exhaustive search | pairwise | Borda ranking |
| Monotone alignment | $S \in \mathbb{R}^{k\times m}$: photo–step or page–section scores | non-empty contiguous blocks | lattice DP | item | best candidate per item |
| Partition | $s(i,j)$: pairwise same-group scores | partitions | exact correlation clustering | pairwise | threshold + connected components |
| Hierarchy | parent scores plus root scores over section nodes | rooted forests | Chu–Liu–Edmonds | parent | best parent or root |

item counts as wrong), 7B *Global* reaches $49.6\%$ on MMLongBench and $46.5\%$ on HRDoc-Simple, and the gaps to *local+solve* are $+18.7$pp ($p{=}3{\times}10^{-22}$) and $+8.4$pp ($p{=}2{\times}10^{-14}$) rather than $+24.6$ and $+19.2$pp. At 72B the item-level MMLongBench gap is $-0.6$pp instead of $+2.7$pp.

**Controlled dataset construction.** All randomized sampling in these controlled datasets uses seed 20260823, except the HRDoc-Simple windows (seed 20260918). Assignment uses GroundUI-18k (Zheng et al., 2025): one random four-instruction subset per screen with distinct, separated target boxes, giving 222 screens and 888 instructions (App. D.2). Capacitated assignment and monotone alignment use the RecipeQA validation split, retaining recipes with 3–7 steps, 6–16 photos, and every step illustrated (218 recipes, 2,341 photos). This set and the public RecipeQA ordering questions both come from the validation split (159 of the 218 recipes also appear among the 929 questions), and both are evaluated without training on them. Ordering samples five screens from each of the 215 MoTIF-derived mobile trajectories in GroundUI-18k with at least five time-consistent screens and shuffles their presentation order. Partition draws six screenshots from two or three GroundUI-18k applications (150 instances, 117 applications). Document alignment cuts MMLongBench (Ma et al., 2024) PDFs into non-overlapping windows of 3–7 consecutive top-level outline sections covering 6–16 pages, yielding 58 windows and 624 pages from 25 documents before excluding the 17 degenerate equal-count windows. The remaining 41 windows contain 506 evaluated pages, and document alignment is reported on them everywhere. The 17 excluded windows (118 of 624 pages, 16 from one slide deck whose outline names every slide) have as many pages as sections, so the monotone cover is unique and *local+solve* scores $100\%$ without the model. Their post hoc exclusion is a disclosed deviation from the pre-specified sample. On all 58 windows, 7B gives *Global* $55.4\%$, *Local* $71.5\%$, flow $72.8\%$, and *local+solve* $92.5\%$, and, for the controls of App. C.3, self-consistency $55.3\%$ and constrained decoding $60.4\%$. The 3B/7B/32B/72B gaps are $+69.9/+37.0/+32.7/+12.0$pp. On the 41 windows, 7B gives $63.2/84.4/85.2/90.7\%$ and the gaps are $+67.8/+27.5/+23.5/+13.2$pp, all significant (3B/7B/32B/72B: $p \le 6{\times}10^{-11}$). All values use the example-free *Global* prompt. The order constraint's value over the flow decode shrinks from $+19.7$ to $+5.5$pp ($p{=}4{\times}10^{-6}$), and the uniform contiguous split constant scores $61.1\%$. Hierarchy windows contain consecutive top-level outline sections and their direct children (6–12 nodes). Of 92 constructed windows, the first 8 are used only for the pre-run probe, and the remaining 84 (843 nodes, 25 documents) are evaluated. Among the 2,341 RecipeQA photo entries used in the controlled recipe settings, 200 are byte-identical to another entry of the same recipe: 191 are the same photo stored twice in the RecipeQA image archive (under two split folders) with the same step label, and 9 are photos that the recipe itself reuses at a different step. Removing these entries changes every affected RecipeQA controlled result (the two RecipeQA rows of Tab. 16: capacitated assignment and recipe alignment) by at most $0.5$pp and does not alter any conclusion.

### A.2 Bidirectional assignment scoring and direction routing

For the public RecipeQA ordering task, we score each correspondence in both directions, for $L$ text items and $L$ observations. The controlled families query each correspondence in a single direction (Tab. 29).

Let $p_\theta(\cdot \mid \cdot)$ denote the frozen VLM's answer-token probability, and let $q_{\mathrm{fwd}}$ and $q_{\mathrm{inv}}$ denote fixed forward and inverse query templates.

The forward query asks which observation matches each text item and produces $S_{\mathrm{fwd}} \in \mathbb{R}^{L\times L}$:

$$S_{\mathrm{fwd}}[i,k] = \log p_\theta(k \mid a_i,\, o_{1:L},\, q_{\mathrm{fwd}})\,, \tag{2}$$

where $k$ denotes the answer token associated with observation $o_k$.

Table 9: **Direction ablation, accuracy and cost** (7B, accuracy values are percentages): forward-only, inverse-only, and combined decode on RecipeQA ordering, the public task that uses bidirectional scoring. Each cell is accuracy with, in parentheses, the token cost as a multiple of the single global-prediction pass when the forward image prefix is cached (each inverse query still processes its own image). Without caching, forward-only costs $3.1\times$ and the combined decode $5.0\times$.

| task (*local+solve* decode) | forward | inverse | combined | note ($\Delta$ vs. best single) |
|---|---|---|---|---|
| RecipeQA ordering (combined) | 82.2 (1.3×) | **93.0** (1.9×) | **93.3** (3.2×) | inverse saturates → +0.3pp (n.s.) |

The inverse query asks which text item matches each observation and produces $S_{\text{inv}} \in \mathbb{R}^{L\times L}$:

$$S_{\text{inv}}[k,i] = \log p_\theta(i \mid o_k,\, a_{1:L},\, q_{\text{inv}})\,, \tag{3}$$

where $i$ denotes the answer token associated with text item $a_i$. Thus, $S_{\text{fwd}}$ and $S_{\text{inv}}^\top$ share the same text-by-observation orientation. These are conditional answer-token log-probabilities over the listed candidates, not likelihoods of the observations themselves.

Using both directions reduces dependence on a single query direction. When both carry usable signal, we combine them as

$$C = S_{\text{fwd}} + S_{\text{inv}}^\top, \tag{4}$$

and, since the RecipeQA answer is one of four listed orderings, we choose the listed ordering with the highest summed score in $C$. A pre-specified label-free routing check, described below, drops a direction only when it shows no above-chance signal. Query cost and caching are detailed in App. A.5.

**Direction routing.** The bidirectional scoring above retains both directions unless one is indistinguishable from chance. For each direction, agreement is the fraction of probe decisions whose top-ranked candidate matches that of an independent frozen auxiliary reference model, and a direction is dropped only when the bootstrap confidence interval of its agreement minus the $1/L$ chance level includes zero. The reference model makes one routing decision per task and never contributes scores to the final matrix $C$, so it provides no per-instance evidence. In a separate subsampling stability analysis ($4{,}000$ resamples of $n$ items per task, deciding only when a $t$-interval over the agreement gaps excludes zero), decisions from $n{=}3$ items pick the direction that is more accurate on the full labeled data in at least $99\%$ of deciding draws where a wrong route is costly, and the check mostly abstains where the two directions are equivalent. Rewording both scoring prompts leaves the reported RecipeQA accuracy nearly unchanged ($93.3/92.6/93.8\%$) while the forward direction alone swings by up to 17pp.

**Direction ablation.** Tab. 9 shows that on RecipeQA, where the inverse direction saturates, combining the two directions is nearly neutral ($+0.3$pp, n.s.). Against the VQAScore-style control in Tab. 3, forward-only trails ($82.2\%$ vs. $90.4\%$), and the reported gain comes primarily from inverse scoring.

### A.3 Local-signal gate

*local+solve* can enforce a task's structural constraints, but it cannot create semantic information that is absent from the VLM's local scores. If those scores do not distinguish correct local decisions from incorrect ones, *local+solve* should not be expected to recover the answer. We therefore use a pre-specified *local-signal gate* to screen whether the chosen local queries expose usable signal before running experiments governed by this protocol at full scale.

For each such experiment, we measure how well the local scores separate correct from incorrect choices using AUC on an eight-instance pre-run probe (48–690 local decisions, depending on the task). AUC $0.5$ corresponds to chance discrimination, and we require AUC $\geq$ $0.70$ to proceed to the full run of an extension. For the backbone replications, probe and full run were launched together and every family is reported, so there the gate flags results rather than stopping runs. Probe construction differs by experiment. For the agentic extensions of App. D, probe items are excluded from the evaluated samples (for Mind2Web this exclusion was applied after the run, App. D). For the backbone replications of App. E.2, every family is screened, and the probe uses the first eight instances of each evaluated set, except for hierarchy, which uses the document parent–child probe of Tab. 10. For the original Qwen2.5-VL-7B controlled runs, only hierarchy has a recorded eight-instance pre-run probe, and the remaining families are characterized retrospectively by their full-run AUCs (Tab. 10, right). In all cases, probe labels are used only for the pre-specified screening decision and do not tune the scoring rule, model, or threshold.

The $0.70$ threshold was fixed in the pre-specified protocol before any probe was run and was never revised. It is motivated by a conventional ROC rule of thumb that AUC below $0.70$ indicates poor discrimination (Hosmer

Table 10: **Uncertainty in the pre-run local-signal gate.** Left: AUC on the eight-instance pre-run probes, with approximate 95% AUC confidence intervals computed from the logged positive and negative counts using the Hanley–McNeil analytic variance estimate (Hanley & McNeil, 1982). The 3×3 jigsaw is not listed because its AUCs are a post hoc full-run diagnostic, not a pre-run probe. Right: full-run AUCs for selected controlled local-scoring rules under Qwen2.5-VL-7B. The two blocks are independent summaries, and rows are not paired.

| *pre-run gate (eight-instance probe)* | | | | *controlled full runs* | | |
|---|---|---|---|---|---|---|
| probe | AUC | pos/neg | 95% CI | setting | AUC | pos/neg |
| AndroidControl partition | 0.995 | 39/81 | [0.98, 1.00] | Cap. assign. / Recipe align. | 0.878 | 2,341/9,575 |
| Doc. parent–child | 0.963 | 51/639 | [0.93, 1.00] | Ordering | 0.821 | 1,017/1,133 |
| AndroidControl ordering | 0.866 | 36/44 | [0.78, 0.95] | Partition | 0.978 | 717/1,533 |
| Mind2Web precedence | 0.740 | 22/26 | [0.60, 0.88] | Hierarchy (parent) | 0.948 | 578/7,296 |
| Mind2Web action–screen | 0.679 | 32/96 | [0.57, 0.79] | Hierarchy (root) | 0.827 | 265/578 |
| Charades letter | 0.653 | 87/425 | [0.59, 0.72] | Alignment (documents) | 0.972 | 506/1,781 |
| Charades Yes/No | 0.631 | 87/425 | [0.56, 0.70] | HRDoc (parent) | 0.887 | 1,032/16,850 |
| Mind2Web anchor alignment | 0.616 | 48/96 | [0.52, 0.72] | HRDoc (root) | 0.886 | 864/1,032 |

& Lemeshow, 2000). Here it is intended as a weak screening criterion rather than a guarantee that *local+solve* will improve performance. The pass/fail decisions are also insensitive to its exact value: passed probes score 0.740 or higher and failed probes 0.68 or lower (Tab. 10), so any threshold from 0.70 to 0.74 yields the same decisions. Of the eight screening probes in Tab. 10, four pass and four fail. Two further sets of probes are not listed there and fall in the same window: the twelve backbone probes of Tab. 25 (ten pass at 0.88 or higher, two fail at 0.57 and 0.62) and the RecipeQA ordering anchor probe (0.944, pass).

Because the gate is estimated from only eight instances, it is necessarily noisy. Tab. 10 reports Hanley–McNeil 95% intervals (Hanley & McNeil, 1982). These analytic intervals do not account for within-instance dependence, so we treat them as approximate rather than cluster-robust. For example, the Mind2Web precedence probe passes at AUC 0.740, although its interval includes the 0.70 threshold. The gate is therefore a pre-run screening step, not the final measurement. For the selected controlled Qwen scoring rules shown in the right block of Tab. 10, full-run AUCs range from 0.82 to 0.98.

The observed pass and fail cases are consistent with this interpretation. The 3×3 natural-image jigsaw (80 TallyQA (Acharya et al., 2019) test images, nine 168px patches each) was run in full before the gate existed, so it has no pre-run probe. A post hoc diagnostic on all 80 instances gives horizontal and vertical seam-adjacency AUCs of 0.71 and 0.59 (480 adjacent versus 5,280 non-adjacent ordered pairs per axis, ties counted as half, or 0.69 and 0.58 with ties counted as errors). The vertical scores carry little signal, and exact structured inference remains near chance (13.5% versus 11.1% chance), so we retain this task as a negative control. Multi-query temporal grounding on Charades (Sigurdsson et al., 2016) also fails: on the same eight probe videos, the original letter scoring rule reaches AUC 0.65, an independent Yes/No rule 0.63, and a SigLIP2 scorer (Tschannen et al., 2025) 0.57 (0.60 on all 150 constructed windows). We therefore stop after the probe rather than running the full experiment. Conversely, document parent–child scoring passes the gate with AUC 0.963, and the corresponding hierarchy experiment succeeds.

**Threshold sensitivity.** To characterize how external assembly depends on local-score informativeness, we progressively degrade the stored local scores with Gaussian noise and re-run the same decoders and solvers while leaving *Global* unchanged (Fig. 4, right). We use 20 seeds at each noise level and continue until the full-run AUC approaches 0.5. Assignment uses the InternVL3-8B and GLM-4.1V-9B score matrices because the original Qwen2.5-VL-7B assignment matrix was not stored.

For assignment, capacitated assignment, and ordering, the *local+solve–Local* gain approaches zero as the local scores become uninformative. The estimated zero-gain crossings are AUC 0.57 $[0.53, 0.63]$ for assignment, 0.73 $[0.64, 0.78]$ for capacitated assignment, and 0.70 $[0.62, 0.75]$ for ordering (seed-percentile intervals). In partition, the two alignment settings, and hierarchy, structural constraints themselves provide an advantage even under nearly random scores, so the *local+solve–Local* gap does not directly measure score informativeness.

Natural runs across Qwen2.5-VL (3B to 72B), InternVL3-8B, and GLM-4.1V-9B show the same qualitative pattern (Fig. 4, left). In particular, the three weak ordering runs (InternVL3-8B, Qwen2.5-VL-3B, and GLM-4.1V-9B) at AUC 0.67, 0.71, and 0.77 yield assembly gains of $-0.3$, $+0.8$, and $+2.8$pp, respectively, with only the last significant. Together, the synthetic and natural results place the pre-specified 0.70 threshold near the region where the additional benefit of external assembly becomes small or unreliable. We therefore treat 0.70 as a weak pre-run screening threshold, not a universal boundary or a guarantee of improvement. Because the degradation is synthetic and isotropic, real weak scorers may fail differently.

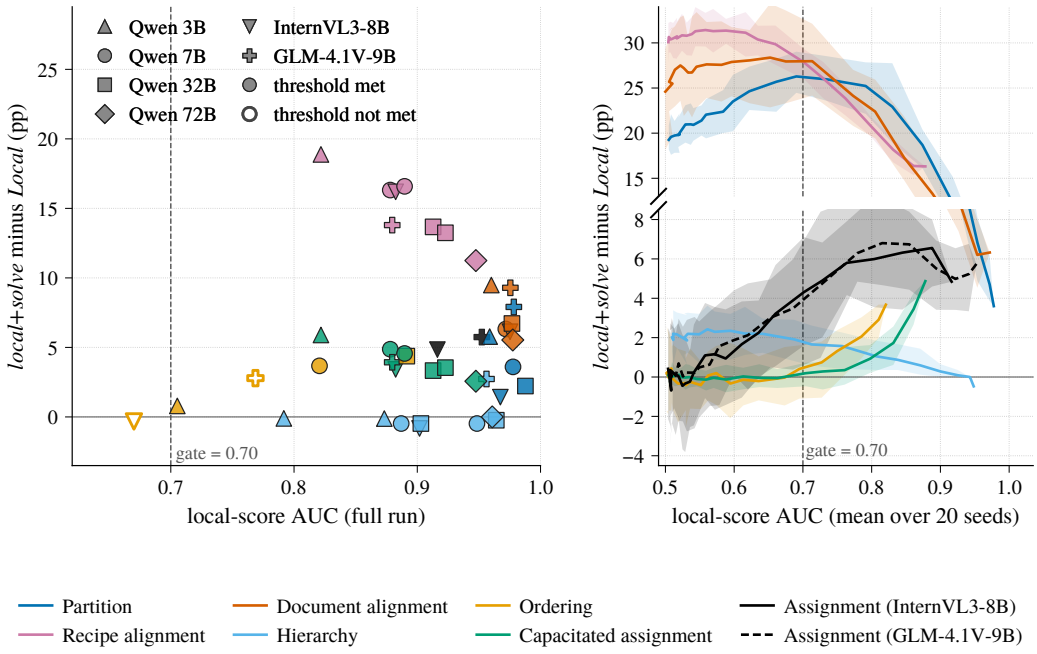



Figure 4: **External-assembly gain versus local-score informativeness, all families** (compact main-text version in Fig. 3). Left: *local+solve* minus *Local* against full-run local-score AUC for 43 family-level points from 35 scoring runs (capacitated assignment and recipe alignment share one score file per model), covering Qwen2.5-VL 3B to 72B, InternVL3-8B, and GLM-4.1V-9B across the controlled families and dataset settings. Filled markers meet the 0.70 threshold, judged on the eight-instance pre-run probe where one was run (InternVL3-8B, GLM-4.1V-9B, and Qwen2.5-VL-7B hierarchy) and on the full-run AUC otherwise. The two open markers are the InternVL3-8B and GLM-4.1V-9B ordering runs, whose probes fall below 0.70 (their full-run AUCs are 0.67 and 0.77). Right: the same gain as Gaussian noise progressively degrades the stored local scores (Qwen2.5-VL-7B, except assignment, which uses the InternVL3-8B and GLM-4.1V-9B matrices). Curves show the mean over 20 seeds and shaded bands show seed variation. The dashed vertical line marks the pre-specified AUC threshold of 0.70. The right panel uses a broken $y$-axis because some structural constraints retain a substantial advantage even when the local scores are weak.

### A.4 Automatic solver selection

Each structural formulation of App. A.1 uses a solver specified in advance. We test whether this routing step can be performed from the text prompt alone, without images or local scores. The selector receives only the text task prompt and a fixed library of the six solvers of Tab. 8, each described by the feasible structure it enforces and shown in five shuffled orders, and answers with one letter. The target is the solver used by our formulation, so this tests selection from a predefined library, not discovery of a new solver. Three prompt sets are used: the *Global* prompts of nine settings (*Original*, three instances each: the six controlled families with alignment on recipes and documents, plus ScreenSpot $k$=3 assignment and HRDoc-Simple hierarchy), the same tasks with constraint statements reworded to avoid the library's terminology (*Paraphrase*), and native-style descriptions of five public or extension settings (*Public/extension*: RecipeQA ordering and text–image assignment, AndroidControl session partition and screen ordering, Mind2Web ordering). This gives 27, 27, and 15 distinct prompts (135, 135, and 75 decisions). A lexical baseline that picks the description with the highest TF–IDF cosine similarity to the prompt reaches 81.5% on the original prompts but 29.6% and 20.0% on the other two sets (chance 16.7%).

Qwen2.5-VL-32B and 72B choose the solver used by our formulation on 97.8–100% of decisions in all three sets (Tab. 11), with their few errors on capacitated assignment. The 7B model falls to 75.6% on the paraphrases, and the 3B model is at the lexical level there. The fallback sensitivity of Tab. 12 averages these nine settings, substitutes *Local* for a wrongly routed setting, and is computed from the recorded Qwen2.5-VL-7B results, not rerun. Routing is therefore reliable within a predefined library at 32B and above, but the structural

Table 11: **Solver-selection accuracy (%).** Each distinct prompt is evaluated under five shuffled orders of the fixed six-solver library. The target is the solver used by the corresponding formulation in this paper.

| Selector | Original | Paraphrase | Public/ext. |
|---|---|---|---|
| Lexical TF–IDF | 81.5 | 29.6 | 20.0 |
| Qwen2.5-VL-3B | 63.7 | 31.1 | 72.0 |
| Qwen2.5-VL-7B | 92.6 | 75.6 | 94.7 |
| Qwen2.5-VL-32B | 98.5 | **100.0** | **100.0** |
| Qwen2.5-VL-72B | 97.8 | 99.3 | **100.0** |

Table 12: **Solver-routing sensitivity.** Macro-average over the nine solver-selection settings, computed from the recorded Qwen2.5-VL-7B *Local* and *local+solve* results while varying only the routing selector. Routing errors fall back to *Local*, and values are computed rather than obtained from end-to-end reruns.

| Routing | Mean acc. |
|---|---|
| *Local* (no solver) | 74.0 |
| Task-defined | **79.4** |
| Auto, 7B selector | 78.2 |
| Auto, 32B selector | **79.4** |
| Auto, 72B selector | 79.3 |

formulation and the library itself are still specified by us, the selector sees text only, and the number of distinct task descriptions is limited.

### A.5 Efficiency

For bidirectional text–image assignment, local scoring issues $2L$ VLM queries per instance: $L$ forward and $L$ inverse. Each forward query shows all $L$ images, so without caching the $L$ forward queries process the same $L$-image prefix $L$ times ($O(L^2)$ image tokens). Caching the key–value states of that shared prefix removes this repetition in the forward half ($O(L)$). It does not help the inverse half: each inverse query shows its single image after a different text prefix, so it still processes its own image tokens (only the vision-encoder features could be reused).

We count prefill tokens per instance and divide by the tokens of one *Global* pass (prompt plus generated tokens). With the forward prefix cached, bidirectional scoring on RecipeQA costs $3.2\times$ one *Global* pass, the sum of the forward ($1.3\times$) and inverse ($1.9\times$) halves of Tab. 9. Without caching, it costs $5.0\times$. These counts are exact for the 929 items. VQAScore-style scoring, by comparison, requires $L^2$ single-image comparisons. The recovery from local scoring does not rely on repeated complete-structure generation, but it costs several *Global* passes on this task.

Tab. 13 profiles the six controlled families using the Qwen2.5-VL-7B logs of the reported runs on one H200 in bfloat16. Query counts are exact, whereas wall-clock times are rounded to the minute. For candidate-selection families, local scoring uses one query per item against the task-provided candidates. GUI assignment instead presents the shared marked screen as the candidate interface. On the logged runs, local scoring is cheaper than global prediction for the document setting, about $1.2\times$ its time for the recipe settings, and about $2.8\times$ for GUI assignment. Ordering, partition, and hierarchy require quadratic pairwise or parent–child scoring, making them more expensive: their logged scoring times are approximately 3–$13\times$ those of the corresponding global predictions.

### A.6 Datasets, splits, and terms of use

Tab. 14 lists the datasets behind the evaluations of the main text and of Apps. B–E, with the split we draw from, the tasks we derive, the evaluated size, and the terms stated by the release. Construction details are in App. A.1 (controlled families), App. D (GUI extensions and MATE), and App. E.3 (public anchors of Tab. 3). Where a release states no dataset license, the table reports the terms given in the dataset paper or on the project page. We use all datasets for non-commercial research.

We obtained GroundUI-18k from its Voxel51 re-release on Hugging Face (`Voxel51/GroundUI-18k`, FiftyOne format), whose dataset card lists "training and fine-tuning models for precise UI element localization"

Table 13: **Query and wall-clock profile of *local+solve* on the controlled families** (Qwen2.5-VL-7B, one NVIDIA H200 GPU, bfloat16, timing logs of the reported runs, with the example-free ordering and partition *Global* runs, and with document alignment timed on all 58 windows). Queries = local queries per instance ($k$ instructions, $k$ photos or pages, $n$ section nodes, five screens or six screenshots per instance). Images = renders encoded per query. Scoring and global prediction are total wall-clock over the pre-specified sample. Hierarchy scoring ran as two concurrent halves of 42 windows each (28 and 23 min), so its serial time is at most their sum.

| Family | Instances | Queries/instance | Images/query | Total queries | Scoring | *Global* |
|---|---|---|---|---|---|---|
| Assignment (GUI marks) | 222 screens | $k$=4 | 1 | 888 | 7 min | 3 min |
| Cap. assign. / alignment (recipes) | 218 recipes | $k \in [6, 16]$ | 1 | 2,341 | 3.8 min | 3.1 min |
| Alignment (documents) | 58 windows | $k$ pages | 1 | 624 | 2 min | 3 min |
| Ordering (GUI trajectories) | 215 | $5 \cdot 4$=20 | 2 | 4,300 | 38 min | 8 min |
| Partition (app grouping) | 150 | $\binom{6}{2}$=15 | 2 | 2,250 | 23 min | 8 min |
| Hierarchy (PDF outlines) | 84 windows | $n(n-1)+n$ | 2 / 1 | 8,717 | $\leq$ 51 min | 4 min |

Table 14: **Datasets used for evaluation.** Sizes are evaluated items after the exclusions described in the referenced appendices. Terms are those stated by the official release (repository, dataset card, or project page). "None stated" means that we found no dataset license there.

| Dataset | Split | Derived tasks | Evaluated size | Stated terms |
|---|---|---|---|---|
| GroundUI-18k (Zheng et al., 2025) | single split (Voxel51 re-release) | assignment, ordering (MoTIF-derived trajectories), partition | 222 screens (888 instr.), 215 sequences, 150 instances | Original: MIT, test set, not for training. Voxel51: no stated license, training listed (see text) |
| RecipeQA (Yagcioglu et al., 2018) | validation | capacitated assignment, recipe alignment, visual ordering with four-step context (Tab. 3) | 218 recipes (2,341 photos), 929 questions | per-recipe licenses listed in the release |
| MMLongBench-Doc (Ma et al., 2024) | full | document alignment, hierarchy | 41 windows (506 pages), 84 windows (843 nodes) | CC BY-NC 4.0, research use only |
| HRDoc-Simple (Ma et al., 2023a) | test | hierarchy (derived from released relations) | 187 windows (1,896 items) | none stated (source papers CC-licensed) |
| MMIU (Meng et al., 2025) | temporal, visual, jigsaw tasks | 4-way ordering and 2×2 jigsaw (Tab. 3) | 200 per task | CC BY 4.0 |
| VIST (Huang et al., 2016) | SIS test | pairwise ordering of jumbled image–caption pairs (Sort-Story protocol, App. E.3) | 5,050 stories (5 of 5,055 lack images) | none stated (CC-licensed Flickr photos) |
| ScreenSpot (Cheng et al., 2024) | full | $k$=3 assignment, native grounding anchor | 190 screens (570 items), anchor 1,272 items | none stated for data (code Apache-2.0) |
| AndroidControl (Li et al., 2024) | test | session partition, screen ordering | 150 and 200 instances | Apache-2.0 (repository) |
| Multimodal-Mind2Web (Deng et al., 2023) | test_domain | screen ordering | 241 trajectories (233 evaluated) | OpenRAIL (dataset card) |
| MATE (Alonso et al., 2025) | full | cross-modal assignment (Tab. 3) | 5,500 items | none stated (CLEVR-based scenes) |
| MiniWoB++ (Liu et al., 2018) | live environment | closed-loop assignment and ordering (click-checkboxes-soft, multi-orderings) | 200 episodes per task | MIT (repository) |

among its direct uses. The card of the original AgentStudio release (`agent-studio/GroundUI-18K`) instead describes the data as a test set and asks that it not be used for training. The transfer analyses of §4.4 and Apps. F.2 and F.4 fit LoRA (low-rank adaptation) (Hu et al., 2022) adapters on GroundUI-18k items. The assignment adapters learn to match instructions to Set-of-Mark boxes on UI screenshots (App. D.2), which is closest to this listed use. The partition and ordering adapters (D3 and the joint adapter of App. F.2) learn to group screenshots by application and to order the screens of one session, which the card does not list. We use these adapters only in experiments that analyze the *assembly deficit*. They are not released, and we claim no GroundUI-18k benchmark performance from them. All other GroundUI-18k results come from frozen models. GroundUI-18k also contains the screenshots of all $1{,}272$ ScreenSpot items. Of the $190$ ScreenSpot $k{=}3$ screens, $19$ are among the training screens of the assignment adapter in Tab. 27. They hold $57$ of the $570$ targets, and $55$ of these $57$ coincide with a training target on the same screen (box IoU above $0.5$), with different instruction wording. Excluding the $19$ screens leaves the frozen $74.9\%$ unchanged and moves the adapter's accuracy from $95.4\%$ to $95.7\%$. The post-training controls of App. F.3 additionally evaluate on random samples of MMStar (Chen et al., 2024), MathVision (Wang et al., 2024a), and MM-UPD (Miyai et al., 2025) with our own short-answer prompts rather than the benchmarks' released protocols, and the other sets used only in App. F (arXiv and manual outlines, the RecipeQA post-training items, and the diagnostic rows) are described where they appear.

# B Complete Structural Results

This section expands the six-family summary of Tab. 1 to all eight controlled task–dataset instantiations. The settings span GUI screenshots, natural photographs, and document pages, while preserving the same comparison between *Global*, *Local*, and *local+solve*. All prediction modes use the same task instance, and ground truth is never shown to the model. Its screening use is described in App. A.3. To keep the controlled diagnosis comparable across families, each family uses one pre-specified local-query form rather than tuning the query interface separately for each dataset.

## B.1 Full results

Tab. 16 reports the complete controlled results. In all eight instantiations, *local+solve* significantly outperforms *Global* ($p \leq 5{\times}10^{-15}$). Much of this recovery is already present in *Local*, while the additional benefit of enforcing the global constraint varies by structure: it is largest for alignment and remains meaningful for the assignment, ordering, and partition settings. For hierarchy, *Local* and *local+solve* differ only in how exact score ties are broken.

After Holm correction (Holm, 1979) over the $14$ comparisons of Tab. 15, every positive gap remains significant at item and instance level, the weakest being partition versus *Local* (instance-level sign test $p{=}.015$, Holm $p{=}.029$). The sole non-positive gap, hierarchy versus *Local* ($-0.5$pp), is non-significant throughout and comes only from how exact score ties are broken.

## B.2 Effect of global constraints

Structural constraints help most when the local decisions decoded without the task-specific solver conflict. Enforcing one-to-one matching adds $+5.7$pp on square assignment ($+5.4$ to $+8.5$pp for $k{=}2$ to 6) but only $+0.3$pp once the candidate set has slack (App. D.2), and capacity bounds add $+4.9$pp. Relative to the order-free flow decode of the same scores, requiring assignments to follow the sequence without moving backward adds $+11.5$pp on recipes, where the constrained decode reaches parity with an oracle given the true per-step counts ($p{=}.44$), and $+5.5$pp on documents ($85.2\% \rightarrow 90.7\%$). Because the recipe photos are given in their true chronological order, a blind baseline that splits them into equal contiguous blocks, one block per step, already scores $63.6\%$ on recipe alignment. *Global* ($61.9\%$) falls below it and *Local* ($64.0\%$) only matches it, while *local+solve* reaches $80.3\%$. Hierarchy is the opposite regime: row-wise parent predictions already form valid forests on all $84$ instances, and the solver departs from them only on $9$ outlines where the parent scores tie exactly, so the total score is unchanged and accuracy moves by $-0.5$pp ($p{=}.29$).

Table 15: **Multiplicity-corrected primary comparisons** on the seven pre-specified dataset settings preceding HRDoc-Simple (Qwen2.5-VL-7B). Gap is *local+solve* minus the named prediction mode in percentage points. McNemar tests items. The sign test counts instances (screens, recipes, document windows, sessions, or outlines) on which *local+solve* has more (+) or fewer (−) correct decisions. Holm correction is applied to the 14 tests in each column. Settings follow Tab. 16.

| | | | Item level | | | Instance level | | |
|---|---|---|---|---|---|---|---|---|
| Family | Dataset | *local+solve* vs. | gap | McNemar $p$ | Holm | sign +/− | sign $p$ | Holm |
| Assignment | GUI marks (888) | *Global* | +17.0 | $8.4\times10^{-29}$ | $6.7\times10^{-28}$ | 77/10 | $5.9\times10^{-14}$ | $5.3\times10^{-13}$ |
| Assignment | GUI marks (888) | *Local* | +5.7 | $8.4\times10^{-7}$ | $1.7\times10^{-6}$ | 68/22 | $1.2\times10^{-6}$ | $6.2\times10^{-6}$ |
| Cap. assign. | RecipeQA photos→steps (2,341) | *Global* | +23.3 | $5.1\times10^{-70}$ | $6.1\times10^{-69}$ | 152/31 | $2.3\times10^{-20}$ | $2.8\times10^{-19}$ |
| Cap. assign. | RecipeQA photos→steps (2,341) | *Local* | +4.9 | $4.1\times10^{-16}$ | $2.4\times10^{-15}$ | 99/25 | $1.3\times10^{-11}$ | $1.0\times10^{-10}$ |
| Alignment | recipes, temporal (2,341) | *Global* | +18.4 | $5.1\times10^{-58}$ | $5.6\times10^{-57}$ | 146/37 | $1.6\times10^{-16}$ | $1.6\times10^{-15}$ |
| Alignment | recipes, temporal (2,341) | *Local* | +16.3 | $4.6\times10^{-83}$ | $6.0\times10^{-82}$ | 170/7 | $1.0\times10^{-41}$ | $1.5\times10^{-40}$ |
| Alignment | doc pages→sections (506) | *Global* | +27.5 | $5.4\times10^{-25}$ | $3.8\times10^{-24}$ | 31/6 | $4.1\times10^{-5}$ | $1.7\times10^{-4}$ |
| Alignment | doc pages→sections (506) | *Local* | +6.3 | $6.7\times10^{-8}$ | $2.7\times10^{-7}$ | 15/0 | $6.1\times10^{-5}$ | $1.8\times10^{-4}$ |
| Ordering | GUI trajectories (2,150 pairs) | *Global* | +21.5 | $2.1\times10^{-50}$ | $2.1\times10^{-49}$ | 156/38 | $3.6\times10^{-18}$ | $3.9\times10^{-17}$ |
| Ordering | GUI trajectories (2,150 pairs) | *Local* | +3.7 | $1.0\times10^{-7}$ | $3.0\times10^{-7}$ | 97/39 | $7.0\times10^{-7}$ | $4.2\times10^{-6}$ |
| Partition | app grouping (2,250 pairs) | *Global* | +29.4 | $3.1\times10^{-146}$ | $4.4\times10^{-145}$ | 118/6 | $4.4\times10^{-28}$ | $5.7\times10^{-27}$ |
| Partition | app grouping (2,250 pairs) | *Local* | +3.6 | $4.0\times10^{-12}$ | $2.0\times10^{-11}$ | 19/6 | 0.015 | 0.029 |
| Hierarchy | PDF outline trees (843) | *Global* | +24.6 | $1.9\times10^{-33}$ | $1.7\times10^{-32}$ | 60/14 | $6.2\times10^{-8}$ | $4.4\times10^{-7}$ |
| Hierarchy | PDF outline trees (843) | *Local* | −0.5 | 0.29 | 0.29 | 1/5 | 0.22 | 0.22 |

Table 16: **Full controlled-probe results** across six structural families and eight task–dataset instantiations. *Global* predicts the complete structure, *Local* decodes the local scores without the task-specific solver, and *local+solve* applies that solver to the same scores. Values are percentages, and bold marks the row maximum. Ordering, partition, and document alignment use the example-free *Global* prompt, hierarchy includes the page images, degenerate document windows are excluded, and invalid *Global* outputs count as wrong (Apps. C.4 and A.1). The two hierarchy rows of *Local* and *local+solve* differ only in how exact score ties are broken.

| Family | Solver | Setting (scored units) | *Global* | *Local* | *local+solve* |
|---|---|---|---|---|---|
| Assignment | Hungarian | GUI marks (888) | 69.6 | 80.9 | **86.6** |
| Capacitated assignment | Hungarian reduction | RecipeQA photos→steps (2,341) | 45.6 | 64.0 | **68.9** |
| Ordering | exhaustive search | GUI trajectories (2,150 pairs) | 53.5 | 71.3 | **75.0** |
| Partition | correlation clustering | app grouping (2,250 pairs) | 65.0 | 90.8 | **94.4** |
| Monotone alignment | lattice DP | RecipeQA photos→steps (2,341) | 61.9 | 64.0 | **80.3** |
| Monotone alignment | lattice DP | document pages→sections (506) | 63.2 | 84.4 | **90.7** |
| Hierarchy | Chu–Liu–Edmonds | PDF outline nodes (843) | 43.8 | **68.8** | 68.3 |
| Hierarchy | Chu–Liu–Edmonds | HRDoc-Simple nodes (1,896) | 35.7 | **55.4** | 54.9 |

# C Diagnostic Controls and Stronger Global Inference

This section reports controls that locate the deficit (Apps. C.1 and C.2) and stronger *Global* inference variants (Apps. C.3 and C.4).

## C.1 Shared-context and sequential-query controls

This subsection provides positive evidence for the behavioral isolation in §4.2: matched prediction modes from the same joint RecipeQA prompt ($n$=929), whose answer-digit log-prob readout re-runs to 38.43% exactly.

**The local signal remains accessible despite assembly failure.** Tab. 2 rules out a decoding-only explanation: scoring the listed candidate sequences moves accuracy from 38.4% only to 38.8% ($p$=.90), whereas the shared-context local query reaches 70.8% element accuracy (58.1% when each step is queried in its own context), and matching those local answers to the listed candidates, with no further model call, reaches 73.0% ($p$=$1.1\times10^{-45}$). Asking the local queries in turn (Tab. 17) with earlier answers kept makes the answers more consistent (an image is repeated on 2.3% instead of 70.0% of instances), yet the fifth-turn global question still scores only 34.3% ($p$=.0015 against the same run's global prediction, 38.5%), so sequential local querying does not recover assembly. Global prediction also carries an answer-position bias: the ground-truth option is nearly uniform on RecipeQA (24.4–26.5%), but global prediction chooses option 4 on 44.0% and option 1 on 0.9% of cases, and MMIU shows analogous choice bias. External assembly avoids this failure mode.

Table 17: **Sequential local queries in the same joint prompt** on RecipeQA visual ordering ($n$=929, four-way choice). The four conditions of Tab. 2 are not repeated here. The last row uses per-item contexts.

| condition (same joint prompt) | acc. (%) | note |
|---|---|---|
| local queries in turn, earlier answers kept, element acc | 69.3 | repeats an image: 2.3% vs. 70.0% |
| all four answers correct | 50.1 | independent queries: 27.0% |
| global question asked after those four turns | 34.3 | $p$=.0015 vs. same-run 38.5 |
| *local+solve* (per-item context + solver) | 93.3 | |

Table 18: **Whole-structure view of Tab. 16** (Qwen2.5-VL-7B, percentages). Per-decision is the family metric. Whole structure requires every decision to be correct. Invalid outputs are scored zero, and constraint violations are measured among parsed outputs: a duplicate mark, a step or section with no item, a backward step, or a cycle. Ordering's *Global* prediction folds non-permutations into invalid. Ordering, partition, and hierarchy settings follow Tab. 16. Dashes in the constraint column indicate that every partition labeling is feasible.

| family | prediction mode | per-decision | whole structure | invalid | constraint violated |
|---|---|---|---|---|---|
| Assignment | *Global* | 69.6 | 53.6 | 0.5 | 0.5 |
| | *Local* | 80.9 | 48.6 | 0.0 | 49.5 |
| | *local+solve* | 86.6 | 77.5 | 0.0 | 0.0 |
| Cap. assign. | *Global* | 45.6 | 4.1 | 2.8 | 40.1 |
| | *Local* | 64.0 | 7.3 | 0.0 | 70.2 |
| | *local+solve* | 68.9 | 9.6 | 0.0 | 0.0 |
| Ordering | *Global* | 53.5 | 1.4 | 0.0 | 0.0 |
| | *Local* | 71.3 | 6.5 | 0.0 | 0.0 |
| | *local+solve* | 75.0 | 14.4 | 0.0 | 0.0 |
| Alignment | *Global* | 61.9 | 7.3 | 5.0 | 55.6 |
| | *Local* | 64.0 | 7.3 | 0.0 | 86.7 |
| | *local+solve* | 80.3 | 25.2 | 0.0 | 0.0 |
| Partition | *Global* | 65.0 | 18.7 | 0.7 | — |
| | *Local* | 90.8 | 58.7 | 0.0 | — |
| | *local+solve* | 94.4 | 62.7 | 0.0 | — |
| Hierarchy | *Global* | 43.8 | 7.1 | 9.5 | 0.0 |
| | *Local* | 68.8 | 22.6 | 0.0 | 0.0 |
| | *local+solve* | 68.3 | 22.6 | 0.0 | 0.0 |

### C.2 Output anatomy across the six controlled families

Tab. 18 shows that symptoms depend on structure. In assignment, $55/222$ instances have exactly $2/4$ correct decisions, the signature of a swap under one-to-one matching. Capacitated assignment and monotone alignment violate the stated constraint on $40.1\%$ and $55.6\%$ of parsed outputs. Partition has 12–$14/15$ correct pairs on $8$ instances and $15/15$ on 28, versus 50 and 94 exact pair sets for *local+solve*, and hierarchy is valid but semantically weak. The independent *Local* decoding is more accurate per decision while violating constraints more often (capacity $70.2\%$, alignment $86.7\%$, one-to-one $49.5\%$), separating local signal from global consistency.

### C.3 Hardening the Global baseline

Among the stronger *Global* inference variants we test, none closes the gap to *local+solve* (Tabs. 6 and 19). Five-sample self-consistency, voting separately at each position, moves assignment $69.6\% \rightarrow 68.2\%$, recipes $61.9\% \rightarrow 66.6\%$, and documents $63.2\% \rightarrow 62.8\%$ on the 41 non-degenerate windows, and matching the scoring-query budget with the same vote adds only $+1.1$pp on recipes. With whole-structure voting, five samples move ordering $53.5\% \rightarrow 52.9\%$ and partition $65.0\% \rightarrow 67.8\%$. With one sample per *local+solve* query (compute-matched), whole-structure voting gives $67.3/52.8/68.6/61.2/66.4\%$ on assignment, ordering, partition, recipe alignment, and document alignment (with the prompts of Tab. 1, so ordering, partition, and documents are example-free), changes of $-2.3$ to $+3.6$pp relative to global prediction that are significant only for partition ($p$=$7\times10^{-7}$). On RecipeQA, where global prediction selects one of four candidate digits, sampling

Table 19: **Stronger *Global* inference does not close the deficit.** Qwen2.5-VL-7B unless noted, family metric in %. *l+s* = *local+solve*, *Global* = direct whole-structure prediction. Gap = *local+solve* − *Global*, computed before rounding. All gaps are significant under paired tests. Rows 1–2 use Qwen3-VL-30B-A3B-Thinking on GUI assignment with four instructions per screen. Self-consistency rows use $T$=0.7, top-$p$ 0.9, and vote over complete structures with as many samples as solver queries, with the prompts of Tab. 1.

| global-prediction variant | *Global* | *l+s* | gap |
|---|---|---|---|
| Thinking model, 4k tokens (30B-A3B, GUI $k$=4) | 71.1 | 91.6 | 20.5 |
| Thinking model, 16k tokens (post hoc) | 84.1 | 91.6 | 7.4 |
| Self-consistency, samples = solver queries: ordering | 52.8 | 75.0 | 22.2 |
| partition | 68.6 | 94.4 | 25.9 |
| Prompt without a format example: partition | 65.0 | 94.4 | 29.4 |

that answer five or eight times (the number of *local+solve* queries) and voting gives 36.5% and 36.9% (mean over 20 seeds) against 38.5% for the same run's global prediction. Every gap to *local+solve* remains significant ($p \leq 5\times10^{-21}$). With the fixed partition example, 2,009/2,238 samples copy it and voting remains 61.2%. Constrained decoding gives 69.8/68.0/51.2% on GUI assignment/recipe alignment/document alignment versus 86.6/80.3/90.7% for the solver ($p \leq 2.4\times10^{-27}$). Relative to global prediction (69.6/61.9/63.2%), its lift is confined to recipe alignment, and on documents it falls below global prediction ($p$=$4\times10^{-7}$). On ordering and partition, restricting the output to valid structures changes 0/215 and 1/150 instances (53.5% and 65.3%), because those outputs are already almost always valid. On RecipeQA, global prediction already takes the most probable of the four candidate digits, which is constrained decoding over the valid answers, so the two coincide (38.4%).

**Procedure prompting does not close the gap.** Describing the full procedure in one prompt raises RecipeQA global prediction from 38.4% to 60.4% at 7B, still below *local+solve* (93.3%, $p$=$1.9\times10^{-74}$). At 72B, the same prompt reaches 78.7%, still below *local+solve* (96.4%, $p$=$3.3\times10^{-38}$). The model reasons in free text at full image resolution, and a reply without a final "ANSWER: $n$" line counts as wrong (16 of 929 at 7B, none at 72B). On the five controlled settings of Tab. 6 at 7B, we write the same procedure as three steps (judge each local choice, resolve the choices under the task's constraint, report the result), decode greedily up to 900 new tokens, and use no format example in any prompt. This prompt lowers assignment from 69.6% to 45.7% ($p$=$2.2\times10^{-36}$, also below the 58.9% of the example-free *Global* prompt, $p$=$1.1\times10^{-11}$), ordering from 53.5% to 50.2% ($p$=.014), and document alignment from 63.2% to 37.0% ($p$=$2.7\times10^{-20}$). It raises partition from 65.0% to 69.1% ($p$=.002) and recipe alignment from 61.9% to 64.5% ($p$=.022, also above the 60.1% of the example-free *Global* prompt, $p$=$7.4\times10^{-5}$). All five stay far below *local+solve* ($p < 10^{-41}$). On assignment, 95 of the 222 final answers are invalid (90 omit at least one instruction, 3 name a mark that is not on the screen, and 2 answer "N/A" or "No match"). Crediting the valid pairs of these answers, with missing pairs counted as wrong, gives 66.3%, numerically below *Global* (69.6%, $p$=.057). On ordering, 164 of the 215 final answers return the presentation order unchanged ("1, 2, 3, 4, 5"), against 1 for the example-free *Global* prompt and 64 with the format example (Tab. 20). On partition, 23 of the 150 final answers are invalid (14 leave out a screenshot, 5 place one screenshot in two groups, and 4 end without an "ANSWER" line). On recipe and document alignment, 22 of the 218 and 9 of the 41 final answers are invalid (24 of these 31 omit a photo or page). Of the remaining 196 and 32 answers, 58 and 17, respectively, leave a step or section without any photo or page, and 62 and 4 go backwards, both of which the prompt forbids. A pre-specified single-pass decomposition control (Khot et al., 2023; Press et al., 2023), with self-chosen or our prescribed sub-questions, scores 35.3–44.1% on capacitated assignment and 41.8–42.1% on hierarchy, 25–34pp below *local+solve* ($p < 10^{-33}$). Supplying the compatibility matrix explicitly on the first 60 items shows that 72B can assemble given evidence (100.0% with chain-of-thought (CoT) (Wei et al., 2022)), whereas 7B reaches 50.0% with chain-of-thought and 25.0% when answering directly. Describing the decomposition or the solver procedure inside a single response therefore helps in some settings and hurts in others, but always remains well below external local scoring with structured inference.

### C.4 Prompt-example copying and the example-free Global prompt

Each global-prediction prompt in App. I ends with a fixed format example. The 7B model sometimes copies it (Tab. 20). On app grouping this happens on all 150 instances, making the fixed-example score of 61.2% (Tab. 20) the score of one constant partition. We therefore re-ran every family with an example-free format instruction using the same samples, seeds, and decoding. The only rule fixed before those re-runs keeps the original prompt when the copy rate is at most 10% (assignment, capacitated assignment, recipe alignment). For

Table 20: **Example copying and the example-free prompt** (Qwen2.5-VL-7B, pre-specified samples, family metric in percent, invalid = unparsable *Global* outputs, counted as zero). Copies = instances whose *Global* output reproduces the in-prompt example (for the variable-length families, its first four items). Ordering additionally returns the presentation order unchanged on 64 instances with the example and on 1 without it. $\Delta$ is example-free minus fixed, with a paired McNemar $p$-value.

| | | | Global, fixed example | | Global, example-free | | | |
|---|---|---|---|---|---|---|---|---|
| Family ($n$) | in-prompt example | copies | acc. | invalid | acc. | invalid | $\Delta$ ($p$) | *local+solve* |
| Assignment (222) | `1B 2A 3D 4C` | 21 | 69.6 | 1 | 58.9 | 19 | $-10.7$ ($10^{-8}$) | 86.6 |
| Cap. assign. (218) | `1A 2C 3B 4A ...` | 12 | 45.6 | 6 | 39.0 | 32 | $-6.6$ ($10^{-9}$) | 68.9 |
| Ordering (215) | `3,1,4,2,5` | 59 | 52.9 | 0 | 53.5 | 0 | $+0.6$ (.69) | 75.0 |
| Alignment (218) | `1A 2C 3B 4A ...` | 7 | 61.9 | 11 | 60.1 | 7 | $-1.8$ (.076) | 80.3 |
| Doc. alignment (41) | `1A 2A 3B 4C ...` | 9 | 51.6 | 0 | 63.2 | 1 | $+11.7$ ($10^{-6}$) | 90.7 |
| Partition (150) | `1,3 \| 2,5 \| 4,6` | 150 | 61.2 | 0 | 65.0 | 1 | $+3.8$ (.005) | 94.4 |
| Hierarchy (84) | `1:0 2:1 3:1 4:0 ...` | 12 | 43.8 | 8 | 46.4 | 15 | $+2.6$ (.12) | 68.3 |

the four families above 10% the choice was made after the re-runs: partition and document alignment, whose changes are significant, are reported example-free. Ordering (+0.6pp, $p$=.69) is also reported example-free, because with the example the compute-matched self-consistency vote returns the example itself on 109 of the 215 instances, so the compute-matched self-consistency row of Tab. 6 would measure copying on about half of the instances. This choice narrows the ordering gap. Hierarchy (+2.6pp, $p$=.12) retains the original prompt. The 7B document *Global* output starts with the example's first four pairs (`1A 2A 3B 4C`) on 9 of the 41 windows (22%, against 3 ground-truth windows with that prefix).

Removing the example does not raise global prediction in general, and on no family does the sign of the gap to *local+solve* change. On assignment it lowers global prediction by 10.7pp and invalid outputs rise from 1 to 19. For partition, the 3B and 7B models copy the fixed format example on all 150/150 instances, and removing the example eliminates this failure mode at every size. Document alignment behaves similarly: without the example, global prediction rises from 51.6% to 63.2% at 7B and from 66.0% to 78.1% at 72B. The gap-sign part of the pre-specified prediction holds everywhere, but its $\leq$ 3pp part (assignment, capacitated assignment, and alignment) fails for the smaller models, where removing the example costs up to 11pp because the example serves as a format guide. The one significant reading that depends on the prompt is the 72B recipe-alignment closure: +1.6pp ($p$=.064) with the fixed example and +4.6pp ($p$=$3\times10^{-7}$) without it. At 72B the example also changes the size of the assignment and ordering gaps, which it inflates (App. E.1).

# D Agentic and External Validation

This section reports the GUI-agent evaluations (App. D.1), controls on the constructed GUI assignment setting (App. D.2), and the public MATE benchmark (App. D.3).

## D.1 GUI-agent validation

We first ask whether the local–global separation also appears in a familiar GUI-grounding setting and across models specialized for GUI interaction. We construct a multi-instruction version of ScreenSpot in which each screen contains $k$=3 instructions and three distinct marked targets, with $k$ denoting the number of instructions per screen. Starting from the ScreenSpot annotations, we retain screens with at least three distinct targets and render Set-of-Mark (Yang et al., 2023) labels at those targets, yielding 190 screens.

We evaluate Qwen2.5-VL-7B and several GUI-specialized models on the same $k$=3 screens. UGround is the only exception: because it emits coordinates rather than mark letters, we evaluate it in its native interface (Tab. 22). The $k$=1 column of Tab. 21 is an unpaired reference for conventional single-instruction grounding. Our diagnostic comparison is the paired evaluation of *Global*, *Local*, and *local+solve* on the same multi-instruction screens.

Across the four models of Tab. 21, *local+solve* outperforms both *Global* and *Local*. *Local* outperforms *Global* on every model at the item level but, with the screen as the unit, not for OS-Atlas (below). For example, UI-TARS reaches 66.8% under *Global*, 90.5% under *Local*, and 96.1% under *local+solve* on the paired $k$=3 screens, while its unpaired $k$=1 reference is 82.8%. Thus, the local–global separation is not confined to the primary Qwen backbone or to a single structured task formulation. It also appears in GUI-specialized models when several grounding decisions must be returned as one joint assignment. *local+solve* also beats global prediction on Mind2Web and AndroidControl trajectories and in closed-loop MiniWoB++ (below). On AndroidControl session partition, however, the thresholded *Local* decode is slightly ahead of *local+solve* (pairwise 88.8% vs.

Table 21: Multi-instruction ScreenSpot evaluation. The $k$=1 GroundUI-18k column is an unpaired single-instruction reference, and all $k$=3 methods are evaluated on the same 190 screens. *Global* uses the example-free prompt. Without the example, it often returns the identity assignment `1A 2B 3C` (55, 52, and 64 of 190 screens for Qwen, OS-Atlas, and UI-TARS, against 32 ground-truth identity screens), UI-TARS is invalid on 12.1% of screens (above our 10% rule), and ShowUI is invalid on every screen (3.2% accuracy with the example). Constrained decoding and PSC also use the example-free prompt. Constrained decoding enforces valid one-to-one outputs, and PSC averages over instruction-order permutations, with vote ties resolved toward the *Global* answer (see text). CoT asks for step-by-step reasoning before the pairs (512 tokens), and self-revision shows the model its own first answer and asks it to check and correct it. Both use the example-free prompt, and UI-TARS CoT is invalid on 14.2% of screens (above our 10% rule). Values are item accuracy (%), with invalid-output rates in parentheses. **Bold** indicates *local+solve*.

| | *single-instr.* | *multi-instruction ($k$=3, same 190 screens)* | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Model | $k$=1 reference | *Global* | *Local* | constrained | PSC | CoT | revision | ***local+solve*** |
| Qwen2.5-VL-7B | 80.1 | 68.8 (1.1) | 86.0 | 68.2 | 84.4 | 74.9 (2.6) | 71.4 | **95.3** |
| OS-Atlas-Base-7B | 70.2 | 64.6 (10.0) | 73.5 | 64.9 | 75.6 | 63.5 (1.1) | 67.2 (10.0) | **80.9** |
| UI-TARS-7B-DPO | 82.8 | 66.8 (12.1) | 90.5 | 63.3 | 87.5 | 75.6 (14.2) | 67.5 (8.9) | **96.1** |
| ShowUI-2B | 58.6 | 0.0 (100) | 71.2 | 32.8 | 0.0 (100) | 0.0 (100) | 0.0 (100) | **80.5** |

87.9% at 7B and 89.6% vs. 88.2% at 72B). App. D.2 and App. D.3 add controls for the constructed GUI dataset settings and a public assignment benchmark.

**Dataset settings.** *ScreenSpot* $k$=3: the 190 screens constructed above, with letters in seed-shuffled order, $190 \times 3 = 570$ items. The reproduction anchor runs the native single-grounding protocol on the full 1,272-item benchmark: our harness reaches 78.0% for OS-Atlas-Base-7B against 82.5% published (Wu et al., 2025) and 90.4% for UI-TARS-7B-DPO against 89.5% published for its SFT checkpoint (Qin et al., 2025), both inside the pre-fixed ±5pp tolerance. *AndroidControl*: from the 1,451 test episodes (8,238 steps), session partition has 150 instances (2–3 episodes × 2–3 screens = 6 mixed screens, ground truth = the episode id, task text withheld from the query) and screen ordering has 200 instances (first five screens of one episode, shuffled). *Mind2Web ordering*: 241 real trajectories × 4 screens. Seeds are fixed and pre-specified. The eight probe trajectories used for local-signal screening were scored in the full run and are removed afterward, leaving 233 evaluated trajectories (1,398 pairwise decisions). With them included, the values are 44.7% and 67.8%. For AndroidControl, the probe instances were excluded before the run.

**Local-signal screening.** The AndroidControl and Mind2Web extensions governed by this protocol passed the local-signal gate before their full runs, and the two failed planned extensions are listed with the other probes in App. A.3 (Tab. 10). ScreenSpot-Pro (Li et al., 2025) could not be constructed (only 3 screens have $\geq 3$ distinct-box instructions), so it is cited only as single-grounding context.

***Local* decoding on the $k$=3 set.** Row-wise argmax of the letter scores that *local+solve* uses gives *Local* 86.0/73.5/63.0/90.5/71.2% for Qwen, OS-Atlas, UGround, UI-TARS, and ShowUI. The three items of a screen are dependent (one swap changes two items), so we test with the screen as the unit (sign test on correct items per screen). *local+solve* beats *Local* on every model ($p \leq 3.4\times10^{-5}$), the one-to-one constraint adding 5.6 to 13.0pp, and beats example-free *Global* on every model ($p < 2\times10^{-7}$). *Local* beats example-free *Global* for Qwen, UI-TARS, and ShowUI ($p \leq 8.1\times10^{-6}$) but not for OS-Atlas (64 screens better, 44 worse, $p$=.067, and 63/46, $p$=.13, against the fixed-example *Global*), although an item-level McNemar test, which ignores the dependence, gives $p \leq 9\times10^{-5}$ on every model.

**Single-instruction control ($k$=1).** On the GroundUI-18k sample (222 marked screens, 888 instructions queried one at a time) the panel is strong (UI-TARS 82.8%, Qwen 80.1%, OS-Atlas 70.2%, ShowUI 58.6%), so the multi-instruction deficit of Tab. 21 is not a harness artifact. UGround returns a valid letter on only 1.1% of instructions (0.1% correct) even at $k$=1 because the Set-of-Mark letter interface lies outside its output format, so it is omitted from the letter-interface columns of Tab. 21 (its residual *local+solve* errors are letter-readout errors on correctly grounded targets) and compared in the coordinate-native arm (independent grounding 93.2%, one-to-one constraint 96.1%).

**The specialist panel on the constructed GUI dataset setting.** On the GUI-marks dataset setting of App. D.2, the GUI specialists do not follow the joint answer format. With the example-free prompt, OS-Atlas and UI-TARS are invalid on 36% and 27% of the 222 screens, and UGround and ShowUI on nearly all. A

prompt that adds a per-screen random format example and the sentence "This example shows only the format" removes most invalid outputs, but OS-Atlas, UI-TARS, and ShowUI then return the example verbatim on 18%, 21%, and 58% of screens (4% expected by chance), and UGround remains invalid on 97%. Under the criteria fixed before this run (invalid below 10%, copies near chance), only Qwen2.5-VL-7B qualifies (1% invalid, 5% copies, 71.6% versus 86.6% for *local+solve*, screen-level sign test $p{=}2{\times}10^{-11}$), so we report no specialist *Global* value or gap on this setting. On AndroidControl, UI-TARS *local+solve* reaches 86.4% on partition and 72.1% on ordering, against example-free global prediction of 44.1% (7/150 invalid) and 35.4% (55/200 invalid, above our 10% rule), with instance-level sign tests $p \leq 6{\times}10^{-33}$. With the fixed format example, which UI-TARS copies on 118/150 and 148/200 instances, global prediction scores are 56.3% and 47.3%. UI-TARS *local+solve* still trails Qwen2.5-VL-7B on ordering (72.1% vs. 87.9%), so grounding specialization does not supply trajectory structure.

Table 22: **Additional agentic comparisons** (accuracy in percent). For native-coordinate rows, *local+solve* adds one-to-one matching to independent grounding, which leaves UI-TARS, already at 95.4%, unchanged (95.3%, $p{=}1.0$). The AndroidControl and Mind2Web baselines are example-free global prediction. With the fixed format example, which Qwen2.5-VL-7B copies on all 150 grouping and 72 of the 200 ordering instances, the AndroidControl baselines are 59.3% and 57.2%. Putting every screen in its own group scores 69.1% on session grouping, above both. On AndroidControl ordering, 55.7%/50.9% are constrained generation and five-sample self-consistency with the example-free prompt. On Mind2Web, constrained generation, five-sample self-consistency, PSC ($m{=}8$, Kemeny aggregation, ties to the unshuffled answer on 169 of 233 trajectories, 9.6% if ties count as wrong), and Setwise score 43.1/46.9/39.6/46.4% with the example-free prompt, and the control column shows the strongest (self-consistency). Dashes mark cells that were not run.

| Evaluation | Model | baseline | global-prediction control | *local+solve* |
|---|---|---|---|---|
| ScreenSpot, native coordinates | OS-Atlas | 89.5 | — | **94.9** |
| ScreenSpot, native coordinates | UGround | 93.2 | — | **96.1** |
| ScreenSpot, native coordinates | UI-TARS | **95.4** | — | 95.3 |
| Mind2Web screen ordering | Qwen2.5-VL-7B | 44.4 | 46.9 | **67.9** |
| AndroidControl session grouping | Qwen2.5-VL-7B | 53.2 | — | **87.9** |
| AndroidControl screen ordering | Qwen2.5-VL-7B | 52.9 | 55.7 / 50.9 | **87.9** |

**Adapted baseline controls.** The global-prediction controls of App. C.3 reproduce on the agentic settings. Constrained generation, self-consistency, PSC, and Setwise stay below *local+solve* on Mind2Web ordering ($p \leq 4{\times}10^{-11}$, trajectory-level sign tests), as constrained generation and self-consistency do on AndroidControl ordering (Tab. 22), minimum-Bayes-risk decoding trails the solver's MAP objective by 2.13pp on partition ($p{=}9.1{\times}10^{-14}$), and PSC, the strongest repair for Qwen, OS-Atlas, and UI-TARS (OS-Atlas 75.6%), stays below *local+solve* for every model of Tab. 21 at 6× cost. The PSC column resolves vote ties toward the *Global* answer. This deviates from the pre-specified rule, which counts ties as wrong, in PSC's favor. Ties occur on 63, 71, 50, and 0 of the 570 items, and under the pre-specified rule PSC scores 78.6/70.4/83.9/0.0% for Qwen, OS-Atlas, UI-TARS, and ShowUI. Chain-of-thought and self-revision on ScreenSpot also stay below *local+solve* for every model (Tab. 21, screen-level sign tests $p < 10^{-6}$). On AndroidControl ordering, *local+solve* raises exact-sequence accuracy from 0.5% under example-free *Global* (1.5% with the example) to 47.0%.

**Closed-loop protocol and context.** MiniWoB++ (Shi et al., 2017; Liu et al., 2018) episodes run live in a browser at fixed resolution. The agent is not pixel-only. It sees the screenshot, reads the task text and target words from the page's HTML (which also gives the number of checkboxes), and the Set-of-Mark letters are inserted into the page. The solver's assignment is executed as clicks on the corresponding page elements, and the environment grades success. On click-checkboxes-soft (200 episodes per mode), *local+solve* scores 68.5% vs. global prediction 18.5% (Fisher exact $p{=}1.4{\times}10^{-24}$). The environment does not honor the seed argument, so the two modes ran on independently drawn episodes and the test is unpaired. Reported context: human 73.0% (Humphreys et al., 2022), RCI (GPT-3.5, DOM text) 72.0% (Kim et al., 2023), CC-Net behavior cloning 4.0% (Humphreys et al., 2022), and the systems that exceed human use reinforcement learning with millions of demonstrations (Humphreys et al., 2022) or an instruction-finetuned model with HTML-level access (Furuta et al., 2024). On multi-orderings, where both modes are near ceiling, the closed-loop scores 100.0% vs. 96.5% (200 unpaired episodes per mode, Fisher exact $p{=}.015$).

### D.2 GUI ASSIGNMENT AND RECTANGULAR FEASIBLE SETS

The dataset setting of Tab. 16 assigns $k$ GroundUI-18k (Zheng et al., 2025) instructions one-to-one to Set-of-Mark letter boxes on one screenshot, all arms sharing the marked image and instruction text on frozen backbones. On 222 screens ($k$=4, 888 items) the 7B backbone gives random 25.0%, global prediction 69.6%, row-wise argmax 80.9%, and *local+solve* 86.6% (Tabs. 16, 18, and 15). On the same-size Qwen3-VL-30B-A3B (Bai et al., 2025a) Instruct/Thinking pair, *local+solve* is unchanged by thinking-tuning (91.4% Instruct vs. 91.6% Thinking, $p$=1.0, zero thinking tokens), while the Thinking model's CoT reaches 71.1% at 4,096 tokens (23.4% non-terminating, mean 1,693 thinking tokens) and, in a post hoc run beyond the pre-specified rule (which fixed 4,096 tokens and prescribed reporting the non-termination rate), 84.1% at 16,384 (+13.1pp, $p$=2e−35, non-termination 15/222, mean 2,902 tokens), still 7.4pp behind ($p$=6e−11, rows 1–2 of Tab. 19). On the 207 terminating screens it scores 90.2% against 92.6% ($p$=.013), so its 92.8% on screens terminating at the smaller budget is a selection effect.

**Slack removes the constraint's leverage.** With $m$=8 marks for $k$=4 instructions (163 screens), *local+solve* still beats global prediction (66.9% vs. 49.4%, $p$=1.3e−18), but the one-to-one constraint's gain over row-wise argmax falls to +0.3pp ($p$=.87) from +5.7pp in the square case, so a $k\times k$-vs-$k\times m$ sweep is the cheap diagnostic.

### D.3 MATE: CROSS-MODAL ENTITY ALIGNMENT

MATE (Alonso et al., 2025) renders a CLEVR-style scene of $L \in [3, 10]$ objects with a metadata list missing one attribute type and asks for one attribute of one object across the modality gap. Answering is a binding. We list the objects from the model's own perception (one call, whose perceived object count averages 0.984 of the true count and is exact on 86.2% of scenes), bind them to the metadata entries by Hungarian assignment on the exposed attributes, and read the answer off the binding, without camera matrices or ground-truth boxes. On the full set ($n$=5,500), global prediction is 80.2% (the benchmark reports 80.1%) and enumerate-then-bind reaches 89.6% (McNemar $p$=1.9×10$^{-66}$). For MATE, *Global* is Qwen2.5-VL-7B's greedy generation on the benchmark's own prompt, scored with the benchmark's JSON answer parser, so unparsable outputs (18 of 5,500) count as wrong, and enumerate-then-bind falls back to this answer on the 17 items where the binding does not reach the queried object. A scale check gives 73.8/87.2% at 3B, 80.2/89.6% at 7B, and 91.7/90.6% at 72B (*Global*/enumerate-then-bind), and the 72B reversal is significant (McNemar $p$=.009), unlike the controlled families, whose deficit narrows but does not reverse at 72B (App. E.1). The 32B *Global* run is not comparable because 5,496/5,500 outputs violate the benchmark's required answer format. At 7B, the gain grows with object count, from +1.2pp at three objects to +14.6pp at ten, matching the combinatorial scaling of the benchmark's own difficulty.

## E ROBUSTNESS AND BOUNDARIES

### E.1 SCALE NARROWS BUT DOES NOT CONSISTENTLY ELIMINATE THE DEFICIT

At 72B (Bai et al., 2025b) the *assembly deficit* shrinks but stays open on assignment, capacitated assignment ($p$=10$^{-42}$), partition ($p$=4×10$^{-29}$), ordering ($p$=4×10$^{-14}$), and document alignment ($p$=6×10$^{-11}$), and comes close to closing on recipe alignment ($p$=.064) and MMLongBench hierarchy ($p$=.10), with values in Tab. 24. Constrained decoding helps on recipe alignment and falls below global prediction on documents (App. C.3). A 106B thinking model (GLM-4.6V (GLM-V Team, 2025a)) ties *7B local+solve* on assignment (84.6% vs. 86.6%, $p$=.14, mean 1,237 thinking tokens per screen against four one-token queries). Its outputs are generated with a 4,096-token cap, at which 18 of the 222 assignment screens do not finish thinking and count as wrong. On partition, 3/150 outputs are invalid: GLM-4.6V scores 94.1% against 94.4% ($p$=.59), and 96.0% against 94.5% for *local+solve* on the 147 valid instances. On AndroidControl session partition GLM-4.6V reaches 89.5% against 87.9% for 7B *local+solve* ($p$=.06, 10/150 invalid).

The four-size curve is not monotone (Tab. 24), but the gap remains significant at every size on assignment, capacitated assignment, ordering, partition, and document alignment. The other two families depend on the statistical unit. The hierarchy gaps at 3B and 72B are small (+1.7 and +2.7pp, item-level $p$=.18 and .10, although the 3B window-level sign test gives 22/9, $p$=.03), and at 32B *Global* is ahead by 5.3pp, significant at the item level ($p$=.003) but not per window (25/36, $p$=.2). Recipe alignment at 32B likewise has *Global* ahead at the item level ($p$=.02) with a recipe-level bootstrap CI of $[-3.7, +0.9]$pp, and at 72B the gap is not significant (CI $[-0.7, +3.8]$pp), where direct global prediction plateaus from 32B to 72B (+0.6pp, $p$=.45) while *local+solve* rises +4.3pp ($p$=3×10$^{-11}$).

**Prompt-example copying at 72B.** Tab. 24 uses the original *Global* prompts except for ordering, partition, and document alignment, and App. C.4 checked them for example copying only at 7B. At 72B the model copies

the format example on $46/222$ assignment and $40/215$ ordering instances, above the $10\%$ rule, and on at most $8$ instances of the other families (Tab. 23). Without the example, 72B *Global* rises from $76.5\%$ to $82.1\%$ on assignment, so the gap shrinks from $+14.0$ to $+8.3$pp ($p{=}5{\times}10^{-12}$). Ordering already uses the example-free prompt in Tab. 24 ($64.8\%$, gap $+9.8$pp, $p{=}4{\times}10^{-14}$), and with the fixed example it scores $60.3\%$ (gap $+14.3$pp). Under either prompt both gaps stay significant and below their 7B values, so the 72B reading of Fig. 2 (the gap narrows from 7B and stays significant in five of its six settings) does not depend on the example. Fig. 2 uses the same prompts as Tab. 24, with the three example-free families at every size (72B copies the document example on $4/41$ windows).

Table 23: **Prompt-example copying at 72B** (Qwen2.5-VL-72B, pre-specified samples, family metric in percent). Copies = instances whose fixed-example *Global* output reproduces the in-prompt example (its first four items for the variable-length families). Gaps are *local+solve* minus *Global*, with paired McNemar $p$. Example-free invalid outputs are 3, 8, 0, 2, 0, and 0 in row order. Ordering, partition, and document alignment in Tab. 24 use the example-free prompt. Hierarchy is not listed because its 72B output copies the example on only $3/84$ windows.

| | | *Global* | | gap, fixed example | | gap, example-free | |
|---|---|---|---|---|---|---|---|
| Family ($n$) | copies | fixed | example-free | pp | $p$ | pp | $p$ |
| Assignment (222) | 46 | 76.5 | 82.1 | $+14.0$ | $2{\times}10^{-22}$ | $+8.3$ | $5{\times}10^{-12}$ |
| Cap. assign. (218) | 1 | 63.7 | 60.8 | $+14.9$ | $1{\times}10^{-42}$ | $+17.8$ | $9{\times}10^{-55}$ |
| Ordering (215) | 40 | 60.3 | 64.8 | $+14.3$ | $1{\times}10^{-27}$ | $+9.8$ | $4{\times}10^{-14}$ |
| Recipe alignment (218) | 0 | 85.8 | 82.8 | $+1.6$ | .064 | $+4.6$ | $3{\times}10^{-7}$ |
| Partition (150) | 8 | 82.1 | 86.4 | $+12.1$ | $1{\times}10^{-55}$ | $+7.8$ | $4{\times}10^{-29}$ |
| Doc. alignment (41) | 4 | 66.0 | 78.1 | $+25.3$ | $8{\times}10^{-28}$ | $+13.2$ | $6{\times}10^{-11}$ |

**Replication of the recipe result.** A disjoint sample drawn with the same pre-specified filter (223 recipes, 2,454 photos) gives gaps of $+17.6/-1.9/+1.5$pp at 7B/32B/72B versus $+18.4/-2.1/+1.6$ on the first sample, with the same item-level $p{=}.02$ and recipe-level CI $[-3.9, +0.5]$pp at 32B. Local-score AUC rises with size on both samples, so the 32B result is not a scoring failure.

**Documents and HRDoc-Simple do not close.** On document pages→sections the gap narrows with size but stays significant at every size, from $+67.8$ to $+13.2$pp on the 41 non-degenerate windows (window-level CI $[+6, +19]$pp at 72B) and from $+69.9$ to $+12.0$pp on all 58 windows (App. A.1). From 32B onward the global outputs are almost always monotone ($1/40$ and $0/41$ windows violate the order), and they exceed the uniform contiguous split ($61.1\%$) modestly at 32B ($68.8\%$, $p{=}.006$) and clearly at 72B ($78.1\%$), whereas 7B does not ($63.2\%$, $p{=}.51$). On HRDoc-Simple the gap shrinks to $+3.3$pp at 32B and reopens to $+12.5$pp at 72B, so the small or reversed MMLongBench hierarchy gaps from 32B onward are specific to that sample. Two caveats apply to HRDoc-Simple: its *Global* prompt sees lower-resolution renders than its local queries, and scoring its *Global* output item by item shrinks the 7B gap from $+19.2$ to $+8.4$pp (App. A.1).

### E.2 Further backbones

To test whether the deficit is specific to Qwen2.5-VL, we repeat the six controlled families with the same items, local queries, solvers, metrics, and pre-run gate on InternVL3-8B (Zhu et al., 2025), whose vision encoder, multimodal alignment, and post-training differ from Qwen2.5-VL while its decoder is a Qwen2.5-7B, and on GLM-4.1V-9B-Thinking (GLM-V Team, 2025b), whose decoder is a GLM-4-9B. The *Global* prompt differences are listed in the caption of Tab. 25. Each backbone uses its own processor defaults for image resolution. For the thinking model, local scores are read after an empty think block, and *Global* is generated without thinking with the same empty block and an `Answer:` prefix, so that the answer lines precede any free text. We chose this after a one-screen smoke test in which thinking did not terminate within 4,096 tokens, and we did not run *Global* with thinking. The *Global* parser keeps the answer lines up to the first line of prose. Unfinished or malformed outputs count as wrong, as elsewhere. Tab. 25 reports all six families, including the two ordering runs whose eight-instance probe falls below the gate.

Both backbones reproduce *Global* < *Local* ≤ *local+solve* on assignment, document alignment, and partition, and InternVL3-8B also reproduces the pattern on both RecipeQA settings. GLM-4.1V-9B is the exception on both RecipeQA settings: its direct prediction is already accurate, so *Global* is within 0.2pp of *Local* on capacitated assignment (the solver still adds $+4.1$pp over *Global*, $p{=}7{\times}10^{-5}$) and exceeds *local+solve* on recipe alignment ($83.5\%$ vs. $81.2\%$, $p{=}.01$). Ordering fails the eight-instance gate on both backbones. For InternVL3-8B the full-run AUC is also low (0.67) and *local+solve* is within 0.3pp of *Local*, the pattern the

Table 24: ***Assembly deficit* across four Qwen2.5-VL sizes.** G is global prediction and l+s is *local+solve*, using the family metric in percent on the pre-specified samples of Tab. 16. Every unmarked gap is significant under paired tests ($p \leq 0.02$). The $^\dagger$ cells require the following qualifications: recipe alignment at 32B has item-level $p$=.02 but recipe-level CI $[-3.7, +0.9]$pp, and at 72B $p$=.064 with CI $[-0.7, +3.8]$pp. Hierarchy at 3B/32B/72B has item-level $p$=.18/.003/.10, with the 32B window-level sign test at $p$=.2. HRDoc-Simple at 32B has item-level $p$=.015, window-level $79/68$ ($p$=.41), and document-level $55/33$ ($p$=.025). HRDoc-Simple *Global* sees renders capped at 401k pixels, about $80\%$ of the pixels of its local queries (App. A.1), and its *Global* outputs are invalid on $9/44/0/3$ windows. Its 72B gap has $p=3\times10^{-34}$ and document-level $71/14$. The document row excludes degenerate windows, uses the example-free *Global* prompt, and remains significant at 72B ($p=6\times10^{-11}$, window-level CI $[+6, +19]$pp). Its 3B *Global* output is invalid on 12 windows. Ordering, partition, and hierarchy settings follow Tab. 16. The 32B hierarchy value uses the pre-specified 512-token cap. The 32B partition value uses the example-free prompt at the default 48-token budget ($3/150$ invalid). The 3B ordering *Global* output returns the presentation order on $210/215$ instances, so its 3B *Global* value is essentially that of the shuffled input order.

| | 3B | | | 7B | | | 32B | | | 72B | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Family | G | l+s | gap | G | l+s | gap | G | l+s | gap | G | l+s | gap |
| Assignment | 39.5 | 68.7 | +29.2 | 69.6 | 86.6 | +17.0 | 75.3 | 88.1 | +12.7 | 76.5 | 90.4 | +14.0 |
| Cap. assign. | 20.5 | 60.1 | +39.6 | 45.6 | 68.9 | +23.3 | 67.1 | 72.7 | +5.7 | 63.7 | 78.6 | +14.9 |
| Ordering | 47.5 | 65.3 | +17.8 | 53.5 | 75.0 | +21.5 | 67.0 | 82.5 | +15.5 | 64.8 | 74.6 | +9.8 |
| Alignment, recipes | 32.2 | 73.0 | +40.8 | 61.9 | 80.3 | +18.4 | 85.1 | 83.1 | $-2.1^\dagger$ | 85.8 | 87.4 | $+1.6^\dagger$ |
| Alignment, documents | 22.9 | 90.7 | +67.8 | 63.2 | 90.7 | +27.5 | 68.8 | 92.3 | +23.5 | 78.1 | 91.3 | +13.2 |
| Partition | 54.3 | 91.8 | +37.5 | 65.0 | 94.4 | +29.4 | 85.5 | 95.2 | +9.7 | 86.4 | 94.2 | +7.8 |
| Hierarchy | 30.6 | 32.3 | $+1.7^\dagger$ | 43.8 | 68.3 | +24.6 | 79.8 | 74.5 | $-5.3^\dagger$ | 73.1 | 75.8 | $+2.7^\dagger$ |
| Hierarchy, HRDoc-Simple | 35.2 | 45.9 | +10.8 | 35.7 | 54.9 | +19.2 | 64.3 | 67.6 | $+3.3^\dagger$ | 71.1 | 83.6 | +12.5 |

gate is meant to catch, whereas for GLM-4.1V-9B the full-run AUC is 0.77 and the solver gain is significant (+2.8pp, $p=4\times10^{-5}$), so the eight-instance probe yields a false negative in this case.

**Video-specialized backbones.** On MMIU temporal ordering, whose frames come from video, global prediction stays near the four-way chance level for LLaVA-NeXT-Video-7B (Zhang et al., 2024c) at $22.5\%$, VideoLLaMA3-7B (Zhang et al., 2025a) at $28.5\%$, and Qwen3-VL-8B at $24.5\%$, and under video input packing ($25.5\%$ and $26.0\%$, and $18.5\%$ for Qwen2.5-VL-72B). *local+solve* reaches $29.5\%$, $42.5\%$, and $50.0\%$ on the three backbones, against $51.0\%$ for Qwen2.5-VL-7B. The gain is significant for VideoLLaMA3-7B ($p$=.0034) but not for LLaVA-NeXT-Video-7B, so the recovery tracks backbone strength.

### E.3 Public-task anchors and an additional ordering task

**External anchor provenance (Tab. 3).** Anchors are the benchmarks' own published evaluations, whose strongest entries are 2024–25 frontier systems (GPT-4o, Gemini-1.5, and Claude 3.5 Sonnet (OpenAI, 2024; Gemini Team, 2024; Anthropic, 2024)), and we claim nothing about later systems, for which the regime statement of §4.5 applies. MMIU (ICLR'25): temporal ordering (Penn-Action source, $L$=4–31, where GPT-4o's $28.0\%$ is the best of its 24 evaluated VLMs, none above the $29.0\%$ frequency baseline), visual ordering (Gemini-1.5 $43.5\%$, a split sharing RecipeQA's source, so temporal is the domain-clean row), and $2\times2$ jigsaw (Gemini-1.5 $34.5\%$, GPT-4o $23.5\%$, the benchmark's own random baseline). Tab. 3 reports the strongest published entry per split and a uniform $25\%$ chance for the three four-way splits, whereas the MMIU paper's empirical random rows are $24.5/27.0/23.5\%$. In MMIU-T, 12 of the 200 items list the ground-truth option twice under two letters, and on 4 of them the ground-truth letter is the later copy. Scoring by letter can mark a choice of the other, identical copy as wrong, so it can understate an MMIU-T value by at most 6.0pp (12 items) for any prediction mode, and by at most 2.0pp (4 items) for *local+solve* if its exactly tied duplicate candidates resolve to the first listed copy. MMIU-V and MMIU-J have no such items. RecipeQA visual ordering (EMNLP'18): all 929 visual-ordering questions of the official validation split, each with the dataset's four candidate orderings of four images. Our text context is easier than the official one. The official context lists every step of the recipe (median 6, up to 20 steps). We keep only the four steps whose images are shuffled, identified from the step numbers in the image file names (an annotation that the official input does not provide), in recipe order, and truncate each step text to 200 characters. The step numbers are used only to select these steps and are never shown to the model. This removes the non-illustrated steps as distractors on the 788 questions with more than four steps. *Global*, *local+solve*, and every baseline receive this same four-step input. Rerunning *Global* and *local+solve* with the official context (every step, full text, no file-name information, and the four

Table 25: **Six controlled families (seven task–dataset settings) on three backbones.** Accuracy (%) of *Global*, *Local*, and *local+solve*, the share of invalid *Global* outputs, the eight-instance pre-run probe AUC ($^\dagger$ below the pre-specified 0.70) and the full-run AUC of the same local scores, and paired McNemar $p$ values on items for *local+solve* against *Global* and against *Local*. Qwen2.5-VL-7B values are those of Tab. 1. Recipe alignment shares the capacitated-assignment scores and gate. Partition *Global* uses the example-free prompt on every backbone. InternVL3-8B and GLM-4.1V-9B copy the fixed partition example on only $6/150$ and $5/150$ instances, so the 10% rule of App. C.4 applied per backbone would keep the fixed example, which gives *Global* 46.8% and 64.2% (30 and 40 invalid outputs) and a larger gap. Ordering *Global* uses the example-free prompt for Qwen2.5-VL-7B (App. C.4) and the fixed example for InternVL3-8B and GLM-4.1V-9B, which copy it on $0/215$ and $17/215$ instances, below the 10% rule. Document alignment *Global* uses the example-free prompt for Qwen2.5-VL-7B and InternVL3-8B, which copy the fixed example on $9/41$ and $8/41$ windows (with it, InternVL3-8B scores 55.9%), and the fixed example for GLM-4.1V-9B ($2/41$ copies).

| Family | Backbone | *Global* | *Local* | *local+solve* | invalid % | gate | full AUC | $p$ vs *Global* | $p$ vs *Local* |
|---|---|---|---|---|---|---|---|---|---|
| Assignment (GUI marks) | Qwen2.5-VL-7B | 69.6 | 80.9 | 86.6 | 0.5 | – | – | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 63.9 | 74.4 | 79.2 | 0.5 | 0.88 | 0.92 | $< 10^{-3}$ | $< 10^{-3}$ |
| | GLM-4.1V-9B | 73.1 | 83.1 | 88.7 | 8.6 | 0.97 | 0.95 | $< 10^{-3}$ | $< 10^{-3}$ |
| Capacitated assignment (recipes) | Qwen2.5-VL-7B | 45.6 | 64.0 | 68.9 | 2.8 | – | 0.88 | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 24.5 | 66.0 | 69.4 | 0.0 | 0.92 | 0.88 | $< 10^{-3}$ | $< 10^{-3}$ |
| | GLM-4.1V-9B | 67.2 | 67.4 | 71.3 | 2.3 | 0.90 | 0.88 | $< 10^{-3}$ | $< 10^{-3}$ |
| Alignment (recipes) | Qwen2.5-VL-7B | 61.9 | 64.0 | 80.3 | 5.0 | – | – | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 40.3 | 66.0 | 82.2 | 0.5 | – | – | $< 10^{-3}$ | $< 10^{-3}$ |
| | GLM-4.1V-9B | 83.5 | 67.4 | 81.2 | 0.0 | – | – | 0.01 | $< 10^{-3}$ |
| Alignment (documents) | Qwen2.5-VL-7B | 63.2 | 84.4 | 90.7 | 2.4 | – | 0.97 | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 42.7 | 85.4 | 91.5 | 9.8 | 0.94 | 0.98 | $< 10^{-3}$ | $< 10^{-3}$ |
| | GLM-4.1V-9B | 81.4 | 86.4 | 95.7 | 7.3 | 0.97 | 0.98 | $< 10^{-3}$ | $< 10^{-3}$ |
| Partition (app grouping) | Qwen2.5-VL-7B | 65.0 | 90.8 | 94.4 | 0.7 | – | 0.98 | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 59.2 | 89.9 | 91.3 | 0.0 | 0.97 | 0.97 | $< 10^{-3}$ | 0.001 |
| | GLM-4.1V-9B | 70.6 | 87.5 | 95.4 | 15.3 | 0.99 | 0.98 | $< 10^{-3}$ | $< 10^{-3}$ |
| Ordering (GUI trajectories) | Qwen2.5-VL-7B | 53.5 | 71.3 | 75.0 | 0.0 | – | 0.82 | $< 10^{-3}$ | $< 10^{-3}$ |
| | InternVL3-8B | 51.8 | 61.4 | 61.1 | 0.0 | $0.57^\dagger$ | 0.67 | $< 10^{-3}$ | 0.65 |
| | GLM-4.1V-9B | 59.7 | 66.7 | 69.5 | 0.5 | $0.62^\dagger$ | 0.77 | $< 10^{-3}$ | $< 10^{-3}$ |
| Hierarchy (MMLongBench) | Qwen2.5-VL-7B | 43.8 | 68.8 | 68.3 | 9.5 | 0.96 | 0.95 | $< 10^{-3}$ | 0.29 |
| | InternVL3-8B | 44.2 | 54.4 | 53.6 | 2.4 | 0.88 | 0.90 | $< 10^{-3}$ | 0.04 |
| | GLM-4.1V-9B | 36.9 | 76.7 | 79.5 | 38.1 | 0.98 | 0.96 | $< 10^{-3}$ | $< 10^{-3}$ |

images placed on strictly increasing steps by the monotone alignment solver) gives the same picture: *Global* scores 36.4% (38.5% with the four-step input in the same run, 38.4% in Tab. 3, $p$=.14) and *local+solve* 88.5% (93.3%), a gap of +52.1pp against +54.8pp ($p$=$1.5\times10^{-113}$, 516 items better and 32 worse). The four-step input raises *local+solve* by about 5pp, mostly on recipes with seven or more steps (86.3% versus 93.9%), and does not create the gap. The published supervised result (HTRN (Sahu et al., 2022), 70.7% with BERT and ViT features) and human accuracy (64.0%) use the full context, so Tab. 3 does not list them. MATE (Findings of ACL'25, $n$=5,500): the strongest reported VLM is Claude 3.5 Sonnet at 83.3% (two-shot prompting, mean of the image-to-text and text-to-image settings), and the paper reports 80.1% for Qwen2.5-VL-7B, which our *Global* run reproduces (80.2%). Answers are free-form attribute values or object names, so Tab. 3 lists no uniform chance for MATE (the paper's random baseline is 22.0%).

**Baselines that do not apply (dashes in Tab. 3).** VQAScore-style scoring needs image–text pairs, so it has nothing to score on the image-only MMIU tasks. Setwise's earliest-item selection has no counterpart in the 2×2 spatial jigsaw of MMIU-J. MATE's free-form answers leave no candidate list for PSC to permute and no sequence for Setwise to build, and VQAScore-style scoring would need object regions that the benchmark does not provide.

**VIST Sort-Story.** Under the original Sort-Story protocol (Agrawal et al., 2016), the five image–caption pairs of a VIST story (Huang et al., 2016) are jumbled and no sentence order is given. We evaluate 5,050 SIS test stories (5 lack images) with a pairwise query that asks which of two pairs comes first, in both presentation orders, and *local+solve* (exact search over the 120 orders) reaches 70.6% pairwise order accuracy, against 70.4% for *Local* (Borda count) and the supervised 79.9% reported on all 5,055 test stories. *Global*, which generates the five letters in story order from an example-free prompt (0 unparsable outputs), reaches 67.1%, so the gain of *local+solve* is +3.5pp (95% CI $[+2.9, +4.2]$).

# F DISTILLATION AND TRANSFER

This section covers task-specific internalization (App. F.1), transfer across structural families (App. F.2), broader post-training controls (App. F.3), and transfer across domains, output formats, and datasets (App. F.4). Unless stated otherwise, LoRA uses $r$=16, $\alpha$=32, and dropout 0.05, trained for two epochs with AdamW at a constant learning rate of $10^{-4}$, gradient clipping at 1.0, and an effective batch of 8 (16 for the task-specific RecipeQA adapters and 2 for the hierarchy adapters). The RecipeQA adapters (Apps. F.1 and F.3) adapt only the LLM attention and MLP projections (40.4M parameters, 0.48%). All other adapters (assignment, partition, joint, D1/D3 of App. F.2, their controls, the 2×2 transfer, and hierarchy adapters) select modules by name, which also matches the MLP projections of the vision tower, so they adapt the vision-tower MLPs as well (47.6M parameters, 0.57%, of which 7.2M are in the vision tower).

## F.1 TASK-SPECIFIC INTERNALIZATION

On RecipeQA visual ordering (four-step input of App. E.3, val $n$=929, chance 25.0%), we evaluate three model states under two prediction modes, *Global* and *local+solve*. The three states are the frozen model, a LoRA adapter trained on 2,000 ground-truth-labeled training examples, and an adapter trained on *local+solve* pseudo-labels for the same 2,000 examples. A fourth adapter, trained on *Global*'s own answers for these examples, serves as a control. No training recipe appears among the 929 validation questions. The adapters are trained to predict the single-digit global answer.

| model | *Global* (%) | *local+solve* (%) |
|---|---|---|
| Frozen (no adaptation) | 38.4 | 93.3 |
| FT, ground-truth labels | 93.3 | **95.2** |
| FT, *local+solve* pseudo-labels | 93.0 | 93.3 |
| Frozen, separate run | 38.5 | – |
| FT, *local+solve* pseudo-labels | 92.0 | – |
| FT, *Global* pseudo-labels | 41.3 | – |

Apart from the *Global* pseudo-label control (below), only the frozen model's *Global* prediction is low. Ground-truth supervision and training on *local+solve* pseudo-labels both raise it to the level reached by training-free *local+solve*. Applying *local+solve* to the ground-truth-supervised model gives the highest accuracy (95.2%), so training sharpens the local decisions the solver consumes. The *Global* pseudo-label control fine-tunes on *Global*'s own answers to the same 2,000 items (43.5% correct, against 94.4% for the *local+solve* labels) with the same LoRA recipe and reaches 41.3%, so exposure to the task format alone does not close the gap. The control comes from a separate run (last block of the table), in which the frozen model scores 38.5% and the *local+solve* pseudo-label adapter 92.0%.

## F.2 STRUCTURAL DIVERSITY AND TRANSFER TO UNSEEN FAMILIES

**Joint adapter.** A single LoRA jointly distilled on three families (no ground-truth labels anywhere) trains on 75 assignment, 250 partition, and 83 ordering instances, none of which shares a screenshot with any evaluation window of the three families (checked by image key, file path, and pixel hash). It raises global prediction on all three: assignment 73.3% → 85.5% ($p$=4.8×10$^{-9}$), partition 65.0% → 94.4% ($p$=8.1×10$^{-147}$, example-free prompt), and ordering 51.6% → 67.5% ($p$=1.5×10$^{-12}$). The assignment and ordering bases are the frozen *Global* scores on the unseen 100-item evaluation split of the joint adapter (for ordering, with the fixed format example of the adapter's training prompt), which is why they differ from the full-set values of Tab. 16. On unseen structures, global prediction rises on alignment (61.9% → 70.7%, $p$=4.8×10$^{-24}$, 6/218 invalid) and on hierarchy (43.8% → 50.8%, $p$=3.5×10$^{-4}$, 15/84 invalid). Unseen-alignment *local+solve* stays flat (80.3% → 80.4%, $p$=.92) and remains well above the adapted global prediction, so the solver is still needed on a new structure. Side variants (training on the scoring queries, adding LoRA to the vision-tower attention layers, and reinforcement learning against the solver objective, including a pre-specified dose–response over two seeds and four budgets) sharpen trained-family behavior but keep unseen-alignment *local+solve* within [−0.4, +0.8]pp of its base.

**Structural diversity at a fixed budget.** With 336 examples pseudo-labeled by *local+solve* (Tab. 26), the three-family adapter D3 (assignment, partition, and ordering, 112 each) raises unseen alignment above both the frozen model and the assignment-only adapter (D1) in every one of five seeds (paired $p \leq .003$ against frozen and $p \leq 2\times10^{-25}$ against D1), whereas spending the same budget on assignment alone lowers unseen alignment accuracy and yields 13 to 45 invalid *Global* outputs out of 218. Unseen hierarchy moves in the same direction in every seed, and unseen-alignment *local+solve* stays at its base level. Two controls use the D3 items in the same order and change only the labels (the *Global* pseudo-label control replaces the 3, 4, and

6 items whose *Global* answer is invalid with other training windows). Ground-truth labels and *Global*'s own answers both transfer less than the *local+solve* pseudo-labels (60.9 and 59.3% against 69.0% on alignment, 49.7 and 45.0% against 54.3% on hierarchy), and ground-truth labels vary most across seeds. At seeds 1 and 2, trained-family assignment *Global* rises from 73.3% to 88.5/91.8% with D1 and 87.3/86.3% with D3, so D1, which uses three times as much assignment data, performs better on its training family. D1 and D3 also differ in task prompts and image sources, so the D1–D3 transfer gap reflects this particular mixture at a fixed budget, not structural diversity in isolation. The three-family joint adapter above (408 examples, a different mixing ratio) also transfers to unseen alignment (70.7%), in line with D3.

Given the local scores and the task-specified feasible set $\mathcal{Y}$, the solver output is deterministic, so solver supervision teaches how existing decisions should be reorganized but supplies no new instance-specific evidence. This is consistent with the results above: trained-family global prediction improves, while decoded accuracy on unseen structures stays nearly flat even as score margins sharpen.

Table 26: **Structural diversity at a fixed distillation budget** (Qwen2.5-VL-7B, 336 examples). D1 uses assignment only and D3 uses assignment, partition, and ordering (112 each), both labeled by *local+solve*. The two controls train on the D3 items in the same order (the *Global* pseudo-label control replaces 3 to 6 items) with ground-truth labels or with *Global*'s own answers. Seeds 1 to 5 are 20260823, 20260824, and 20260903 to 20260905, and the controls use seeds 1 to 3. Unseen alignment is the pre-specified primary outcome. Hierarchy includes page images, and *local+solve* is decoded from the same adapters. Dashes indicate that the run was not performed.

| | frozen | seed 1 | seed 2 | seed 3 | seed 4 | seed 5 | mean ± sd |
|---|---|---|---|---|---|---|---|
| *Unseen alignment, Global (%)* | | | | | | | |
| D1 | 61.9 | 39.8 | 54.5 | 56.9 | 44.9 | 46.6 | 48.5 ± 7.1 |
| D3 | | **72.8** | **64.8** | **68.2** | **69.5** | **69.9** | **69.0** ± 2.9 |
| D3 items, ground-truth labels | | 64.1 | 50.3 | 68.1 | — | — | 60.9 ± 9.3 |
| D3 items, *Global* pseudo-labels | | 55.6 | 62.9 | 59.3 | — | — | 59.3 ± 3.7 |
| *Invalid unseen-alignment Global outputs (/218)* | | | | | | | |
| D1 | 11 | 45 | 25 | 13 | 31 | 29 | |
| D3 | | 7 | 13 | 5 | 7 | 5 | |
| D3 items, ground-truth labels | | 17 | 33 | 8 | — | — | |
| D3 items, *Global* pseudo-labels | | 20 | 13 | 26 | — | — | |
| *Unseen hierarchy, Global, page images (%)* | | | | | | | |
| D1 | 43.8 | 36.5 | 40.0 | 45.9 | 40.9 | 40.7 | 40.8 ± 3.4 |
| D3 | | **54.6** | **59.4** | **55.0** | **47.9** | **54.3** | **54.3** ± 4.1 |
| D3 items, ground-truth labels | | 45.6 | 52.7 | 51.0 | — | — | 49.7 ± 3.7 |
| D3 items, *Global* pseudo-labels | | 46.7 | 45.0 | 43.2 | — | — | 45.0 ± 1.8 |
| *Unseen alignment, local+solve (%)* | | | | | | | |
| D1 | 80.3 | 79.8 | 79.6 | 79.8 | 79.4 | 79.7 | 79.7 ± 0.2 |
| D3 | | 80.4 | 80.9 | 81.5 | 79.7 | 80.4 | 80.6 ± 0.7 |
| D3 items, ground-truth labels | | 80.6 | 80.3 | 80.9 | — | — | 80.6 ± 0.3 |
| D3 items, *Global* pseudo-labels | | 80.1 | 80.1 | 79.5 | — | — | 79.9 ± 0.4 |

### F.3 Broader post-training controls

Four post-training regimes are compared against the same frozen 7B backbone with paired tests: a scaled self-distillation on all 7,558 RecipeQA training items with four examples each (no ground-truth labels), rejection-sampled CoT SFT on 2,600 of these items (1,871 traces whose answer matches the *local+solve* label), group relative policy optimization (GRPO) (Shao et al., 2024) with KL regularization (learning rate $10^{-5}$), and the public Visual Jigsaw 7B release (Wu et al., 2026). Across the four regimes, we detect no significant gains on MMStar (Chen et al., 2024), MathVision (Wang et al., 2024a), and MM-UPD (Miyai et al., 2025) at the pre-specified +3pp sensitivity level. On unseen narration–frame matching, *local+solve* remains stable across all four adapted models. Thus, none of the four regimes measurably improves general benchmarks at this sensitivity. MathVision is evaluated with the answer options of its multiple-choice items ($n$=1,000): the frozen backbone scores 19.5%, scaled self-distillation 19.7%, CoT SFT 18.4%, GRPO 19.5%, and Visual Jigsaw 19.3%, compared with 19.2% for the frozen backbone in Visual Jigsaw's evaluation environment (paired sign tests, all $p \geq .08$).

Table 27: **Self-distilled transfer across domain and output format** (Qwen2.5-VL-7B, no further training). Partition uses ARI. Assignment and hierarchy use item accuracy. Frozen is the model before any adapter. For partition, the same-format frozen cell uses the example-free prompt, because the original prompt returns its format example on all 150 instances (ARI 0.012). The letter-format frozen prompt copies its example on $34/150$ instances. Other cells use the adapter's own prompt. Hierarchy includes page images at the same capped resolution in both arms. The arXiv-outline adapter (trained on the 28 arXiv windows) is also the one evaluated on HRDoc-Simple. Frozen *local+solve* is 0.855 (app grouping), 95.3% (ScreenSpot), 72.1% (manual outlines), 61.4% (arXiv outlines), 54.9% (HRDoc-Simple), and 68.3% (MMLongBench, all 84 windows). Pairwise partition accuracies for the two cells are $65.0 \rightarrow 93.9$ and $73.3 \rightarrow 88.6$. Nineteen of the 190 ScreenSpot screens also occur among the assignment adapter's GroundUI-18k training screens (App. A.6). Excluding them gives 95.7%. Dashes mark cells that were not run.

| | | same format | | new format (letter labels) | |
|---|---|---|---|---|---|
| Adapter | evaluation set | frozen | adapter | frozen | adapter |
| Partition (app grouping) | app grouping, same domain | .248 | **.861** | .340 | **.788** |
| Assignment (GroundUI $k$=4) | ScreenSpot $k$=3, new dataset | 74.9 | **95.4** | — | — |
| Hierarchy (arXiv outlines) | manual outlines, new doc. type | 44.9 | **73.2** | — | — |
| Hierarchy (manual outlines) | arXiv outlines, new doc. type | 33.2 | **61.4** | — | — |
| Hierarchy (arXiv outlines) | HRDoc-Simple, new dataset | 35.7 | **57.4** | — | — |
| Hierarchy (HRDoc-Simple) | MMLongBench, new dataset | 40.8 | **54.1** | — | — |

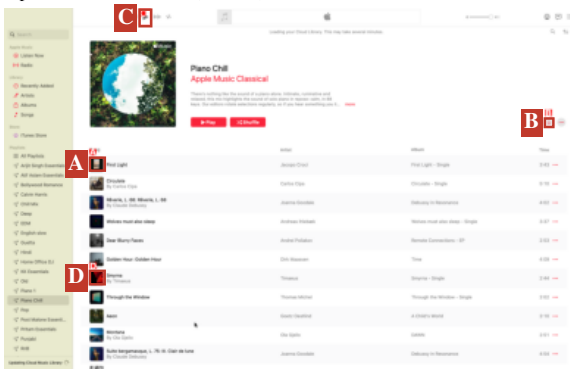
### F.4 Domain, dataset, and output-format transfer

The partition adapter is distilled on 250 GroundUI-18k app-grouping instances and the assignment adapter on 336 GroundUI-18k screens with $k$=4 instructions, both with *local+solve* pseudo-labels. Evaluating the self-distilled adapters without further training on data they were not trained on separates format transfer from structural transfer (Tab. 27). For partition, the frozen same-format prompt returns its format example on all 150 instances, so we measure the in-domain gain against the example-free frozen prompt (ARI 0.248). The adapter's ARI gain is then $+0.613$ in its training format and $+0.448$ with letter labels, so the new format retains about 73% of the in-format gain, and part of what the adapter learns is tied to the output format. Within hierarchy, the adapter distilled on the 28 arXiv windows and the adapter distilled on 28 of the 56 other MMLongBench windows (mostly manuals, with 7 windows from two academic papers) reach the frozen *local+solve* level on the other document type (73.2 vs. 72.1 and 61.4 vs. 61.4), while significantly improving over the frozen *Global* prediction ($p \leq 8\times10^{-17}$), and across datasets the arXiv adapter raises HRDoc-Simple from 35.7% to 57.4%, whereas HRDoc-Simple supervision recovers about half of the frozen *Global*-to-*local+solve* gap on MMLongBench ($40.8\% \rightarrow 54.1\%$).

## G Qualitative Examples

We show fixed-seed examples, mostly from the main Qwen2.5-VL-7B runs (other models are marked in each panel), one per structural family (Figs. 5–9), one for each remaining dataset setting (Fig. 10), and one designated stress case (Fig. 11). These examples are selected to illustrate the failure and recovery mechanisms, not to estimate their frequency. Aggregate frequencies are reported in the quantitative tables. For each family we take, in a fixed random order (seed 20260924), the first instance on which *local+solve* is fully correct while both *Global* and *Local* are wrong. Instances whose input fails a check are skipped: instructions containing auto-generated annotation text or duplicated instructions, and target boxes that do not enclose the named element. For legibility, recipes with more than eight photos and outlines with more than eight sections are excluded. No hierarchy window has *Local* wrong while *local+solve* is right, so that panel drops the requirement that *Local* be wrong and shows the boundary case in which the solver changes nothing (App. B.2). Each panel gives the images, the exact *Global* prompt and local query, and the predictions of *Global*, the baselines run on that setting, *Local*, and *local+solve*, with wrong items in red. Per-item outputs for all instances are released with the code.

Input: marked screenshot (4 marks)




***Global prompt***

A screenshot with 4 UI elements marked with red letters (A, B, C, D).
Instructions:
1: Click on 'First Light' by Jacopo Croci.
2: Click on 'Smyrna'.
3: Click on the 'Play' button.
4: Click on "Download All Songs".

Match each instruction to the single marked element it refers to. Every mark is used exactly once. Reply with exactly 4 pairs like: 1B 2A 3D 4C

***Local query (one per instruction)***

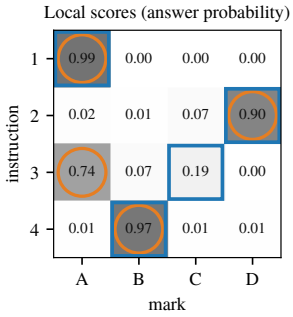
A screenshot with 4 UI elements marked with red letters (A, B, C, D).
Which single marked element does this instruction refer to: '<instruction i>'? Reply one letter.

**Predictions**

| Method | 1 | 2 | 3 | 4 | Note |
|---|---|---|---|---|---|
| Ground truth | A | D | C | B | |
| *Global* | A | D | **B** | **C** | raw reply: 1A 2D 3B 4C |
| Constrained decoding | A | D | **B** | **C** | |
| Self-consistency (5 samples) | A | **A** | **B** | **C** | |
| Thinking model* | A | D | C | B | |
| *Local* | A | D | **A** | B | items 1 and 3 both pick A |
| *local+solve* | A | D | C | B | |

*Qwen3-VL-30B-A3B-Thinking; all other rows Qwen2.5-VL-7B. Red bold: wrong item.

Local scores (answer probability)




circle: *Local* (row argmax) square: *local+solve*

Figure 5: **Assignment (GUI marks).** The instance is the 11th in the fixed order. Of the ten earlier instances, nine fail the instruction checks and one fails the target-box check stated at the start of this appendix. The heat map gives the answer probabilities of the local queries. *Local* lets instructions 1 and 3 take the same mark, and one-to-one matching moves instruction 3 to its second choice.

Input: recipe photos (shown here in time order) and the recipe's steps.

photo 1 photo 2 photo 3 photo 4 photo 5 photo 6 photo 7 photo 8



***Global prompt (with all photos)***

These are 8 photos from one recipe, numbered 1 to 8 in the order shown.
The recipe has 4 steps:
A. Ingredients: To make checked potato fries you'll need:1) 3-4 medium sized […]
B. Creating Checked Pattern: Peal the potatoes and wash them, Now it's time to create the checked […]
C. Flavoring: Now that you have the checked potatoes ready you can add any kind of […]
D. Frying: Take a frying pan and add enough oil to fry the potatoes, Heat oil […]
Assign every photo to exactly one step. Each step has at least one photo.
[Alignment prompt only: The photos are in chronological order; assignments must not go backwards.] Answer for all 8 photos, starting with photo 1, in order. Reply only with photo-step pairs, like: 1A 2C 3B 4A ...

***Local query (one per photo)***

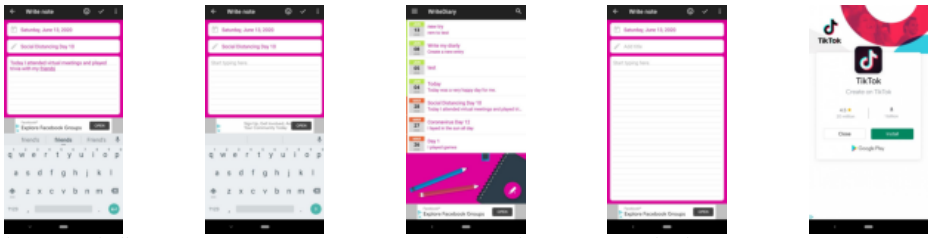
[photo i]
This photo is from a recipe with 4 steps:
A. Ingredients: To make checked potato fries you'll need:1) 3-4 medium sized […]
B. Creating Checked Pattern: Peal the potatoes and wash them, Now it's time to create the checked […]
C. Flavoring: Now that you have the checked potatoes ready you can add any kind of […]
D. Frying: Take a frying pan and add enough oil to fry the potatoes, Heat oil […]
Which step of this recipe does the photo show? Answer with a single letter.

| **Method** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **Note** |
|---|---|---|---|---|---|---|---|---|---|
| Ground truth | A | B | B | B | C | C | D | D | |
| *Capacitated assignment (every step used, order ignored)* | | | | | | | | | |
| *Global* | **B** | B | B | **D** | C | C | D | D | raw (presentation order): 1D 2B 3C 4D 5B 6B 7C 8D |
| *Local* | **B** | B | B | B | C | C | D | **A** | shared by both formulations |
| *local+solve* (flow) | **B** | B | B | B | C | C | D | **A** | |
| *Monotone alignment (photos in time order, steps never go back)* | | | | | | | | | |
| *Global* | A | B | B | B | C | **D** | D | D | raw: 1A 2B 3B 4B 5C 6D 7D 8D |
| Constrained decoding | A | B | B | B | C | **D** | D | D | |
| Self-consistency | **B** | B | B | B | **A** | **D** | D | D | |
| *local+solve* (lattice DP) | A | B | B | B | C | C | D | D | |

All rows Qwen2.5-VL-7B. Red bold: wrong photo.

Figure 6: **Capacitated assignment and monotone alignment (RecipeQA) on the same recipe and the same local scores.** The recipe is selected by the alignment rule, and capacitated assignment fails on it. Photos are shown in time order, and capacitated-assignment outputs are re-indexed to that order. Requiring every step to be used leaves the *Local* errors on photos 1 and 8 in place, whereas requiring that steps never go back corrects both.

Input: five screens from one app session, shown in shuffled order.




***Global prompt (with all 5 screens)***

These are 5 screenshots from one app usage session, numbered 1 to 5 in the order shown (shuffled). Order them chronologically from earliest to latest. Reply only with the order: the five screenshot numbers, earliest first, separated by commas.

***Local query (one per ordered pair)***

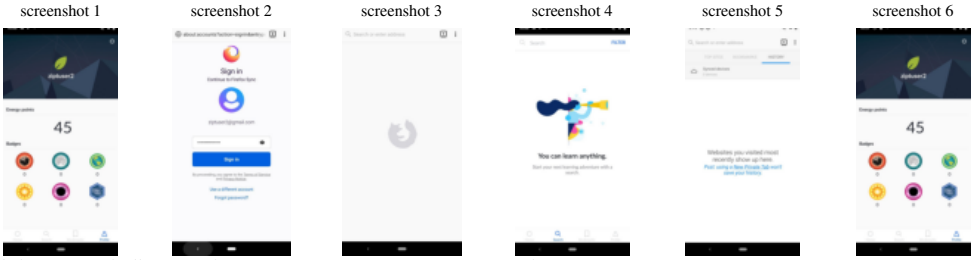
[screen i, screen j]
These are two screenshots from the same app usage session. Was the first screenshot taken before the second one? Answer Yes or No.
(asked for all 20 ordered pairs)

| Method | 1st | 2nd | 3rd | 4th | 5th | Note |
|---|---|---|---|---|---|---|
| Ground truth | 3 | 4 | 2 | 1 | 5 | |
| *Global* | 3 | 4 | **1** | **2** | 5 | pairwise 9/10 |
| Self-consistency (20 samples) | **4** | **2** | **3** | 1 | 5 | pairwise 8/10 |
| *Local* (Borda) | 3 | 4 | **5** | 1 | **2** | pairwise 7/10 |
| *local+solve* (exhaustive) | 3 | 4 | 2 | 1 | 5 | pairwise 10/10 |

All rows Qwen2.5-VL-7B. Cells give the screen placed at each position. Red bold: wrong position.

Figure 7: **Ordering (GUI trajectory).** Cells give the screen placed at each position. *Global* and self-consistency use the example-free prompt shown, as in Tab. 6. Self-consistency votes over 20 sampled complete orders, matching the number of local queries. *Local* is the Borda ranking of the pairwise scores.

Input: six screenshots from 2 applications, in presentation order.




***Global prompt (with all 6 screenshots)***

These are 6 screenshots of applications, numbered 1 to 6 in the order shown. Group them by application: screenshots of the same application go in one group. Reply only with the groups: list the screenshot numbers of each group separated by commas, and separate groups with ' | '. Every number from 1 to 6 appears exactly once.

***Local query (one per pair)***

[screenshot i, screenshot j]
These are two screenshots. Are they from the same application or website?
Answer Yes or No.
(asked for all 15 pairs)

| **Method** | **1** | **2** | **3** | **4** | **5** | **6** | **Note** |
|---|---|---|---|---|---|---|---|
| Ground truth | A | B | B | A | B | A | |
| *Global* | **B** | B | B | A | **A** | A | pairwise 7/15 |
| Self-consistency (15 samples) | **B** | B | B | A | **A** | A | pairwise 7/15 |
| *Local* (threshold) | A | **A** | **A** | A | **A** | A | pairwise 6/15 |
| *local+solve* (exact) | A | B | B | A | B | A | pairwise 15/15 |

All rows Qwen2.5-VL-7B. Letters name groups (matched to the ground truth). Red bold: screenshot in the wrong group.

Figure 8: **Partition (app grouping).** Predicted groups are matched to the ground-truth groups before coloring. Thresholding the pair scores lets a single false positive (screenshots 2 and 6, log-odds $+0.38$) merge the two applications, whereas exact correlation clustering recovers both groups.

Input: the first page of each section (section titles are listed in the prompt).

section 1 | section 2 | section 3 | section 4 | section 5 | section 6 | section 7



***Global prompt (with all section pages)***

A document has 7 sections and subsections, listed in random order. The 7 images are the first pages of these items, in the same order (image 1 is item 1, ..., image 7 is item 7):
1. Dataset Details
2. Hyper-parameters for Evaluation with PaLM2-L
3. Additional Error Analysis
4. Evaluation Details
5. Few-shot Examples for Evaluation with PaLM2-L
6. TimeQA Error Analysis
7. StrategyQA Error Analysis
Recover the hierarchy: for every item, give its parent — another item number, or 0 if it is a top-level section. Answer for all 7 items in order. Reply only with item:parent pairs, like: 1:0 2:1 3:1 4:0 ...

***Local queries (every ordered pair, and every section)***

[first pages of sections a and b]
These are the first pages of two sections of the same document. Section A (first image): "<title a>". Section B (second image): "<title b>". Is section B a direct subsection of section A? Answer Yes or No.

[first page of section a]
This is the first page of the section "<title a>" of a document. Is this section a top-level section (not a subsection of any other section)? Answer Yes or No.

| **Method** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **Note** |
|---|---|---|---|---|---|---|---|---|
| Ground truth | 0 | 4 | 0 | 0 | 4 | 3 | 3 | |
| *Global* (page images) | 0 | **0** | 0 | 0 | **0** | 3 | 3 | 5/7 parents |
| *Local* (best parent) | 0 | 4 | 0 | 0 | 4 | 3 | 3 | 7/7 parents |
| *local+solve* (Chu-Liu-Edmonds) | 0 | 4 | 0 | 0 | 4 | 3 | 3 | 7/7 parents |

All rows Qwen2.5-VL-7B. Parent 0 = top level. Red bold: wrong parent.

Figure 9: **Hierarchy (MMLongBench outline), a boundary case.** The row-wise parent choices already form the correct tree, so *local+solve* equals *Local*, and only *Global* errs. Several sections start on the same page, which gives the page-identity cue discussed in App. A.1.

**Monotone alignment: pages → sections (MMLongBench)**

Sections: A = Appendix: Technical data, B = Appendix: Accessories and spare parts, C = Index. Cells: section of each page.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Ground truth | A | A | A | B | B | B | C | C | C | C | C |
| *Global* | A | A | A | **A** | **A** | **A** | **B** | **B** | **B** | C | C |
| *Local* | A | A | A | B | B | B | C | C | C | C | **A** |
| *local+solve* | A | A | A | B | B | B | C | C | C | C | C |

**Hierarchy: section outline (HRDoc-Simple)**

Sections: 1 = A.3 Annotation Details, 2 = B Examples of Model Prediction, 3 = A.2 Semantic Roles, 4 = A Conversational SRL Dataset, 5 = References, 6 = A.1 Dialogue Dataset: DuConv. Cells: parent (0 = top level).




| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Ground truth | 4 | 0 | 4 | 0 | 0 | 4 |
| *Global* | **0** | **1** | **1** | 0 | 0 | **1** |
| *Local* | 4 | 0 | 4 | 0 | 0 | 4 |
| *local+solve* | 4 | 0 | 4 | 0 | 0 | 4 |

**Assignment: instructions → marks (ScreenSpot, $k = 3$)**

Instructions: 1 = "do not disturb mode", 2 = "adjust keyboard brightness", 3 = "go to music app". Cells: chosen mark.



marked screen, cropped to the marks

| | 1 | 2 | 3 |
|---|---|---|---|
| Ground truth | C | A | B |
| *Global* | **A** | **B** | **C** |
| *Local* | **A** | **B** | B |
| *local+solve* | C | A | B |

**Partition: screens → sessions (AndroidControl)**

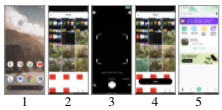
Cells: group of each screen, matched to the ground truth.




| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Ground truth | A | B | A | B | C | C |
| *Global* | A | **A** | A | **A** | C | C |
| *Local* (threshold) | A | B | A | B | C | **D** |
| *local+solve* | A | B | A | B | C | C |

**Ordering: shuffled screens of one episode (AndroidControl)**

Task: "Utilizing the PlantCam app, I want to identify the plant by taking a picture of it and identifying its species." Cells: screen at each position.




| | 1st | 2nd | 3rd | 4th | 5th |
|---|---|---|---|---|---|
| Ground truth | 1 | 5 | 3 | 2 | 4 |
| *Global* | **2** | **3** | **4** | **5** | **1** |
| *Local* (Borda) | 1 | 5 | **2** | **3** | 4 |
| *local+solve* | 1 | 5 | 3 | 2 | 4 |

**Ordering: shuffled screens of one web task (Mind2Web)**

Task: "Find the playful level of a corgi." (akc.org). Cells: screen at each position.




| | 1st | 2nd | 3rd | 4th |
|---|---|---|---|---|
| Ground truth | 1 | 4 | 3 | 2 |
| *Global* | **3** | **2** | **4** | **1** |
| *Local* (Borda) | **4** | **1** | 3 | 2 |
| *local+solve* | 1 | 4 | 3 | 2 |

All rows Qwen2.5-VL-7B. Prompts as in the detailed panel of the same family. Red bold: wrong item. Group letters are matched to the ground truth.

Figure 10: **Remaining dataset settings, compact form.** Each panel applies the selection rule of the detailed panels to one setting of Tab. 16, Tab. 21, or Tab. 22, and each shown instance is the first qualifying instance in the fixed order (no instance was skipped by the input checks). Document alignment draws from the 41 non-degenerate windows. ScreenSpot draws from the 190 screens of Tab. 21 without the stress case of Fig. 11, and Mind2Web from the 233 evaluated trajectories. *Global* rows are the outputs reported in the tables, which use the example-free prompt for document alignment, ScreenSpot, AndroidControl, and Mind2Web. HRDoc-Simple, like the MMLongBench outline, has no window with *Local* wrong and *local+solve* right, so only *Global* is required to be wrong. The public-benchmark outputs (MMIU, RecipeQA ordering, VIST, MATE) and the closed-loop and native-coordinate GUI runs are not included.

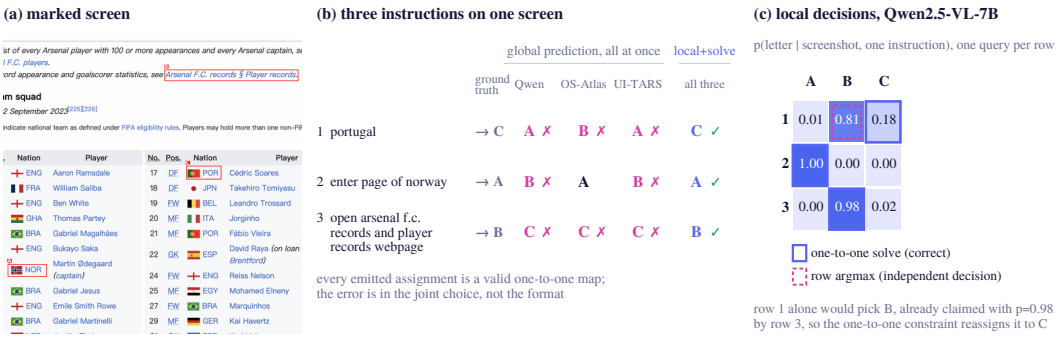



Figure 11: **Designated stress case: a ScreenSpot screen (a Wikipedia page).** With the example-free prompt, global prediction is wrong on eight of the nine instruction–model cells. Qwen and UI-TARS return the presentation-order assignment (1→A, 2→B, 3→C) and OS-Atlas returns 1→B, 2→A, 3→C, while one-to-one assembly of each model's own local scores is exact. The example maximizes global-prediction errors among screens solved exactly by all three *local+solve* runs on which all three global outputs are valid one-to-one maps, with ties broken by the fixed order of the other panels.

# H Implementation Details

**Models and inference.** Tab. 28 lists the exact checkpoints. Qwen2.5-VL is the primary backbone, with its 3B, 32B, and 72B variants used for the scale comparison, and GUI-specialized models and several alternative vision-language backbones test generality across model families. Models run in bfloat16 via HuggingFace `transformers`. Generation-based baselines decode greedily unless noted. Images are capped at 512 px on the long side where memory requires.

**Evaluation.** All conditional scores are single-token log-probabilities: the chat template is applied with the generation prefix, the literal string `Answer:` is appended, and we read the log-softmax over the candidate letter (`A...H`) or digit tokens at the first generated position. Letter queries score every listed candidate at that position, and Yes/No queries score $\log p(\mathrm{Yes}) - \log p(\mathrm{No})$. First-token probabilities can diverge from free-form generated answers (Wang et al., 2024b). That concern targets multi-token free-form answers, whereas our answers are single letter or digit tokens. For multiple-choice *Global* evaluations (the MMIU and RecipeQA columns of Tab. 3), we likewise score the answer digit at the first generated position, avoiding parsing failures. *Global* predictions without listed options (the controlled families, the GUI settings, and MATE) are instead generated and parsed, with unparsable outputs counted as incorrect (App. I). The RecipeQA shared-context control further shows that replacing the answer-digit readout with complete-candidate likelihood scoring changes accuracy only from $38.4\%$ to $38.8\%$ ($p$=.90, App. C.1), so the observed gap is not an artifact of the answer-digit readout.

**Solver implementation.** The assignment solver is SciPy `linear_sum_assignment`. The pairwise ordering query is asked in both presentation orders and antisymmetrized, $M[i,j] = \frac{1}{2}\left(s(i,j) - s(j,i)\right)$, before the solver consumes it. Candidate-permutation MCQs are scored by summing the matrix entries each listed permutation implies.

Table 28: Model checkpoints used in the experiments.

| Model | Checkpoint |
|---|---|
| Qwen2.5-VL-3B-Instruct | `Qwen/Qwen2.5-VL-3B-Instruct` |
| Qwen2.5-VL-7B-Instruct | `Qwen/Qwen2.5-VL-7B-Instruct` |
| Qwen2.5-VL-32B-Instruct | `Qwen/Qwen2.5-VL-32B-Instruct` |
| Qwen2.5-VL-72B-Instruct | `Qwen/Qwen2.5-VL-72B-Instruct` |
| OS-Atlas-Base-7B | `OS-Copilot/OS-Atlas-Base-7B` |
| UGround-V1-7B | `osunlp/UGround-V1-7B` |
| UI-TARS-7B-DPO | `ByteDance-Seed/UI-TARS-7B-DPO` |
| ShowUI-2B | `showlab/ShowUI-2B` |
| Qwen3-VL-8B-Instruct | `Qwen/Qwen3-VL-8B-Instruct` |
| Qwen3-VL-30B-A3B-Instruct | `Qwen/Qwen3-VL-30B-A3B-Instruct` |
| Qwen3-VL-30B-A3B-Thinking | `Qwen/Qwen3-VL-30B-A3B-Thinking` |
| GLM-4.6V-106B | `zai-org/GLM-4.6V` |
| InternVL3-8B (backbone replication) | `OpenGVLab/InternVL3-8B-hf` |
| GLM-4.1V-9B (backbone replication) | `zai-org/GLM-4.1V-9B-Thinking` |
| LLaVA-NeXT-Video-7B | `llava-hf/LLaVA-NeXT-Video-7B-hf` |
| VideoLLaMA3-7B | `DAMO-NLP-SG/VideoLLaMA3-7B` |
| SigLIP2 (auxiliary reference model) | `google/siglip2-large-patch16-512` |

# I Prompt Templates

The public ordering templates below are shown in their RecipeQA instantiation, whose scenario string is "a cooking recipe" inside a prompt frame written for first-person procedure videos. Other domains swap only the scenario wording (e.g., "a cooking recipe" $\rightarrow$ "a ⟨object⟩ in a first-person kitchen video"), and App. A.2 shows *local+solve* is robust to rewording them. Angle brackets are filled per instance. Tab. 29 maps every task to its local and *Global* prompts.

**Bipartite local query.** Forward, one query per action $i$, all $L$ images shown lettered:

```
A first-person video of the procedure '⟨scenario⟩'.  ⟨L⟩
OBSERVATION images of states (random order, lettered):
[Observation A ...]
Which single observation letter shows the state right AFTER
the step '⟨step i⟩'?  Reply one letter.
```

Inverse, one query per observation $k$, that image alone shown, actions listed lettered:

```
A first-person video of the procedure '⟨scenario⟩'.  This
image shows one observed state.  Candidate steps:  [A: ...]
```

> Which single step letter produced the state in this image? Reply one letter.

**Single-set local query.** Pairwise probes over image pairs $(a, b)$. Precedence: "Two photos from one story, A (first shown) and B (second shown). In the correct temporal order, which photo comes FIRST? Reply A or B." with $M[a, b] = \log p(\mathrm{A}) - \log p(\mathrm{B})$. Successor: "Two photos from one story: the FIRST shown, then a candidate NEXT photo. Does the second photo come IMMEDIATELY AFTER the first in the story? Reply Yes or No." with $M[a, b] = \log p(\mathrm{Yes}) - \log p(\mathrm{No})$.

**Global prediction.** MC form (MMIU and RecipeQA rows of Tab. 3), answer read as the log-prob of the option label (letters for MMIU, digits for RecipeQA). For *local+solve* and VQAScore-style scoring on these MCQs, each listed candidate is scored by summing the matrix entries it implies, and the best-scoring candidate is chosen. The RecipeQA prompt:

> A cooking recipe. Steps in correct order: [1. ...4.]
> 4 images of these steps (random order, lettered): [Image A ...]
> Candidate orderings of the images (as image-letter sequences): [1) A-B-C-D ...]
> Which numbered ordering puts the images in the correct recipe order? Reply one digit (1-4).

Free-form generation (internal rows with no listed options): "These $\langle L \rangle$ photos form a story but are shuffled: [...] Output the $\langle L \rangle$ letters in the correct temporal story order, comma-separated (e.g., B,A,D,C,E).", greedy, parsed. Unparsable outputs count as wrong.

**Global-prediction prompts of the six controlled families.** Each states the family's feasibility constraint in words, so the solver receives no structural information that the prompt withholds. The hierarchy target is a forest (several top-level sections per document), and the prompt allows exactly that. Greedy decoding, output parsed, unparsable outputs count as wrong.

> **Assignment.** A screenshot with $\langle k \rangle$ UI elements marked with red letters ($\langle$A, B, ...$\rangle$). Instructions: $\langle$1: ... $k$: ...$\rangle$ Match each instruction to the single marked element it refers to. Every mark is used exactly once. Reply with exactly $\langle k \rangle$ pairs like: 1B 2A 3D 4C
>
> **Capacitated assignment.** These are $\langle k \rangle$ photos from one recipe, numbered 1 to $\langle k \rangle$ in the order shown. The recipe has $\langle m \rangle$ steps: $\langle$A. ... $\rangle$ Assign every photo to exactly one step. Each step has at least one photo. Answer for all $\langle k \rangle$ photos, starting with photo 1, in order. Reply only with photo-step pairs, like: 1A 2C 3B 4A ...
>
> **Monotone alignment (recipes).** As above, with the photos presented in chronological order and the sentence ``The photos are in chronological order; assignments must not go backwards.'' inserted after the capacity sentence.
>
> **(Documents.)** These are $\langle k \rangle$ consecutive pages of one document, numbered 1 to $\langle k \rangle$ in reading order. This part of the document has $\langle m \rangle$ sections: $\langle$A. ...$\rangle$ Assign every page to exactly one section. Each section has at least one page. The pages are in order; assignments must not go backwards. Answer for all $\langle k \rangle$ pages, starting with page 1, in order. Reply only with page-section pairs, like: 1A 2A 3B 4C ...
>
> **Documents, example-free** (the *Global* column of Tabs. 1 and 16, App. C.4). Same prompt with the last sentence replaced by: Reply only with page-section pairs, one per page: the page number followed by its section letter, separated by spaces.
>
> **Ordering.** These are 5 screenshots from one app usage session, numbered 1 to 5 in the order shown (shuffled). Order them chronologically from earliest to latest. Reply only with the order, like: 3,1,4,2,5

> **Ordering, example-free** (the *Global* column of Tabs. 1 and 16, App. C.4). Same prompt with the last sentence replaced by: Reply only with the order: the five screenshot numbers, earliest first, separated by commas.
>
> **Partition.** These are 6 screenshots of applications, numbered 1 to 6 in the order shown. Group them by application: screenshots of the same application go in one group. Reply only with the groups, like: 1,3 | 2,5 | 4,6
>
> **Partition, example-free** (the *Global* column of Tabs. 1 and 16, App. C.4). Same prompt with the last sentence replaced by: Reply only with the groups: list the screenshot numbers of each group separated by commas, and separate groups with ' | '. Every number from 1 to 6 appears exactly once.
>
> **Hierarchy.** A document has ⟨$n$⟩ sections and subsections, listed in random order. The ⟨$n$⟩ images are the first pages of these items, in the same order (image 1 is item 1, ..., image ⟨$n$⟩ is item ⟨$n$⟩): ⟨1. ...⟩ Recover the hierarchy: for every item, give its parent --- another item number, or 0 if it is a top-level section. Answer for all ⟨$n$⟩ items in order. Reply only with item:parent pairs, like: 1:0 2:1 3:1 4:0 ...

**Local queries of the six controlled families.** Each local query shows only the items it names. The hierarchy queries receive the first-page render of each named section.

> **Assignment.** A screenshot with ⟨$k$⟩ UI elements marked with red letters (⟨A, B, ...⟩). Which single marked element does this instruction refer to: '⟨instruction⟩'? Reply one letter. **(ScreenSpot.)** The same first sentence, then: Instruction: ⟨instruction⟩ Which marked element does this instruction refer to? Reply one letter.
>
> **Capacitated assignment and monotone alignment (recipes).** This photo is from a recipe with ⟨$m$⟩ steps: ⟨A. ...⟩ Which step of this recipe does the photo show? Answer with a single letter.
>
> **Monotone alignment (documents).** This page is from a document with ⟨$m$⟩ sections: ⟨A. ...⟩ Which section does this page belong to? Answer with a single letter.
>
> **Ordering.** These are two screenshots from the same app usage session. Was the first screenshot taken before the second one? Answer Yes or No.
>
> **Partition.** These are two screenshots. Are they from the same application or website? Answer Yes or No. **(AndroidControl sessions.)** These are two screenshots from phone app usage. Are they from the same usage session (one user performing one task)? Answer Yes or No.
>
> **Hierarchy.** These are the first pages of two sections of the same document. Section A (first image): ``⟨title $a$⟩''. Section B (second image): ``⟨title $b$⟩''. Is section B a direct subsection of section A? Answer Yes or No. **(Root.)** This is the first page of the section ``⟨title⟩'' of a document. Is this section a top-level section (not a subsection of any other section)? Answer Yes or No.

**Long-sequence permutation decoding and invalid outputs.** For public ordering sequences longer than seven items, we initialize the permutation with the Borda count of the pairwise scores and refine it by insertion, whereas the controlled five-item ordering probes use exact permutation search. Benchmark interfaces without an explicit null option do not add one, and unparsable or incomplete *Global* outputs count as incorrect.

**Baseline prompts.** Generation-based baselines reuse the global-prediction/CoT prompt family above, and full verbatim strings ship with the code. Chain-of-thought uses greedy decoding with 256 new tokens. Self-

Table 29: **Which prompt each task uses.** Local queries are read as single-token log-probabilities. The image column states what the model sees in one local query. Named prompts refer to the verbatim blocks in this section. Agentic and external tasks (App. D) reuse the ordering and partition probes with the scenario wording changed, as in the AndroidControl example above.

| Task (data) | Local decision | Seen per local query | Local query | *Global* prompt |
|---|---|---|---|---|
| Assignment (GUI marks) | instruction → marked letter | marked screenshot | Assignment | Assignment |
| ScreenSpot $k$=3 (Tab. 21) | instruction → marked letter | marked screenshot | Assignment (ScreenSpot) | Assignment |
| Capacitated assignment (RecipeQA photos→steps) | photo → step letter | one photo, step list | Capacitated assignment | Capacitated assignment |
| Monotone alignment (recipes, temporal) | photo → step letter | one photo, step list | same as capacitated assignment | Monotone alignment (recipes) |
| Monotone alignment (doc pages→sections) | page → section letter | one page, section list | Monotone alignment (documents) | Monotone alignment (documents), example-free |
| Ordering (GUI trajectories) | pairwise precedence, Yes/No | two screenshots | Ordering | Ordering, example-free |
| Partition (app grouping) | pairwise same-group, Yes/No | two screenshots | Partition | Partition, example-free |
| Hierarchy (MMLongBench and HRDoc-Simple outlines) | pairwise parent + root, Yes/No | two first-page renders (one for root) | Hierarchy | Hierarchy |
| RecipeQA (Tab. 3) | step → image letter (forward), image → step letter (inverse) | all $L$ images (forward), one image (inverse) | Bipartite local query | *Global*, MC digit |
| VIST Sort-Story (App. E.3) | pairwise precedence, A/B (both presentation orders) | two photos, each with its caption | which moment comes FIRST | *Global*, generated letter order |
| MMIU-T/V (Tab. 3) | pairwise precedence or succession | two images | Single-set local query | *Global*, MC letter |
| MMIU-J (Tab. 3) | position → patch number | masked full image, four patches | which patch belongs at a position | *Global*, MC letter |

consistency samples at $T$=0.7, top-$p$=0.9. With $N$=5 it votes per position on assignment and alignment and over whole structures on ordering and partition, and the compute-matched rows of App. C.3 vote over whole structures with one sample per local query. On RecipeQA it samples the candidate digit from the *Global* answer distribution ($N$=5, and $N$=8 compute-matched). PSC (Tang et al., 2024) uses $m$=8 presentations, the original and seven shuffles. For RecipeQA each shuffle permutes both the image order and the listing order of the candidates, the chosen candidate is mapped back before the vote, and vote ties (178 of 929 items) go to the unshuffled answer (45.9% if ties count as wrong). Setwise (Zhuang et al., 2024) iteratively selects the earliest remaining item, and a reply without a valid image number selects the first remaining image (such fallbacks were not counted). VQAScore-style scoring (Lin et al., 2024) asks the same Qwen2.5-VL-7B backbone a yes/no question per image–text pair, uses $V[i,j] = \log p(\texttt{Yes})$ for the space-prefixed answer token, and feeds that matrix to the same solver. SigLIP2 cosine similarity is used only as the auxiliary reference model (App. A.2). Generate-then-match produces a greedy 48-token post-step caption and matches it with SigLIP2: RecipeQA reaches 87.8%, versus 86.9% for raw step-text matching ($p$=.48) and 93.3% for *local+solve* ($p$=3.3e−6).