COMPVIS-BENCH: BENCHMARKING STRUCTURAL UNDERSTANDING OF COMPOSITE VISUALIZATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world visualizations combine charts, maps, and glyphs whose components are connected through shared data fields. Understanding these *composite visualizations* requires recovering not only each component's visual encoding, but also the data relationships that connect the components. We introduce COMPVIS-BENCH, a reconstruction-based benchmark for evaluating this structural understanding in vision-language models. Its normalized mark-group representation preserves data semantics, encoding structure, and cross-group field correspondences while abstracting away presentation details and alternative composition hierarchies. We build an expert-annotated seed set and extend it through controlled structural modifications, randomized rendering, and manual verification, yielding 1,600 verified visualizations. Across 12 large vision-language models, local mark groups and encodings are recovered substantially more reliably than the field correspondences that connect them. These results expose a central limitation of current models: recognizing the parts of a complex visualization does not imply understanding how those parts jointly represent data.

1 INTRODUCTION

Composite visualizations (Figure 1) combine multiple data-encoding systems, such as charts, maps, or glyphs, through juxtaposition, overlay, nesting, and repetition (Javed & Elmqvist, 2012; Chen et al., 2021; Deng et al., 2023a). For example, a category encoded by color in a pie chart may also organize positions along a bar chart's axis. Understanding the composition requires recognizing that these encodings refer to the same field, while avoiding false correspondences based on similar appearance alone. For vision-language models, this poses a broader question: can local visual recognition support globally consistent data binding across heterogeneous encodings? Composite visualizations provide a concrete setting for investigating this ability. Although they can appear in infographics, we focus on data and encoding relationships rather than narrative or decoration.

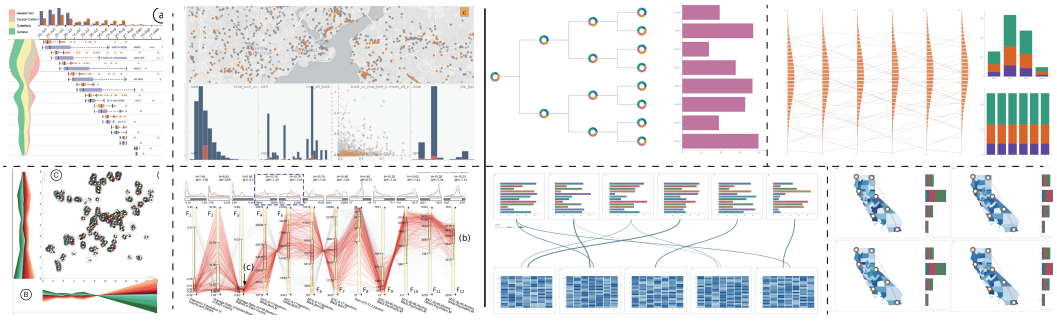

Figure 1: Example visualizations from seed and extension sets. Left: images from the seed set; right: images from the extension set, which is derived and expanded from the seed set.

Cross-component comparison and information integration depend on these bindings: a model can recognize each chart correctly yet connect unrelated fields, producing an incorrect interpretation of the whole. Evaluating this ability requires distinguishing component recognition from relational

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.

recovery. Structured perception (Xia et al., 2026; Peng et al., 2026) provides explicit targets, but primarily focuses on individual charts or encoding systems, with limited representation of relationships among multiple systems. Generation-based evaluation (Wu et al., 2025; Yang et al., 2025; Goswami et al., 2025) offers a more holistic test through reconstruction, but couples understanding with the ability to implement a visualization and reproduce its presentation. Neither directly tests binding recovery among correctly identified components.

We introduce COMPVIS-BENCH to diagnose this ability through structured reconstruction from rendered images. Its explicit annotations allow components, encodings, and field correspondences to be evaluated separately. Our representation retains *relational data semantics* and *encoding structure*, specifying how fields are encoded and shared across groups of visual marks, while abstracting away *presentation details* (Figure 2). We normalize alternative composition hierarchies and decomposition granularities into a fixed mark-group representation, preserving cross-group correspondences through shared field identities. This supports direct comparison with reference annotations without requiring executable visualization code. Multi-level metrics evaluate local and global recovery, while matched-group metrics measure correspondence recovery among identified components.

Using this representation, we construct an expert-annotated seed set of 121 visualizations from VAID (Ying et al., 2024). We extend the seed annotations with new structural and relational configurations and render them with varied presentation choices, yielding 1,600 manually verified extensions. Evaluation of 12 large vision-language models (LVLMs) reveals a clear gap between local encoding recovery and cross-group field correspondence. Correspondence errors persist among successfully matched mark groups, showing that component identification alone does not account for the difficulty of recovering consistent relational structure. Two datasets are available at `https://github.com/CompVis-Bench/dataset`.

Our contributions are threefold:

- ◇ We introduce COMPVIS-BENCH, a reconstruction-based benchmark for structural understanding of composite visualizations, with a normalized representation and metrics separating local recovery from cross-group binding.
- ◇ We construct an expert-annotated seed set and extend it through structural and relational modifications, yielding 1,600 manually verified visualizations.
- ◇ We benchmark 12 LVLMs, revealing a gap between local encoding recovery and cross-system correspondence, with binding errors even among identified components.

2 RELATED WORK

Composite visualization design. Composite visualizations combine multiple visual representations into a coordinated design, allowing related data to be expressed through multiple encoding systems (Javed & Elmqvist, 2012; Chen et al., 2021; Deng et al., 2023a). At the semantic level, such compositions are organized around relationships among the underlying data fields: different visual representations may share fields, expose complementary attributes, or encode related portions of the data. At the visual level, these relationships are expressed through composition mechanisms such as juxtaposition, overlay, nesting, shared visual encodings, and coordinated repetition. The resulting spatial and visual relationships provide observable evidence of how the participating encoding systems are connected through their underlying data.

Evaluation of visualization understanding. Visualization understanding is commonly evaluated through question answering (Masry et al., 2022; Wang et al., 2024; Zhu et al., 2025). Such benchmarks can require substantial visual and logical reasoning, but evaluate the information needed for individual queries rather than explicitly recovering the organization of the visualization as a whole. A closer line of work targets structured chart perception (Poco & Heer, 2017; Xia et al., 2026; Peng et al., 2026). These approaches recover structures such as single-chart visual encoding specifications or data-centric representations derived from chart content. For example, prior work reconstructs encoding specifications from individual chart images (Poco & Heer, 2017), while structured parsing methods represent chart data through header–entity–value relations or related tabular structures (Xia et al., 2026; Peng et al., 2026). CompVis-Bench instead targets the relational organization of *composite* visualizations, requiring models to jointly recover multiple encoding systems, their field bindings, and the cross-system relationships induced by shared data fields and composition.

Evaluation of visualization generation. Faithful visualization generation can be viewed as a strong operational test of understanding: reproducing a reference visualization requires a model to interpret its content and structure well enough to realize them correctly. Accordingly, chart-to-code benchmarks evaluate generated programs and outputs through execution, layout and appearance matching, perceptual similarity, and data recovery (Wu et al., 2025; Yang et al., 2025; Li et al., 2025). These metrics provide useful measures of generation quality, but do not isolate structural understanding. Code-based measures depend on successful implementation, appearance-based measures are sensitive to presentation choices, and image-level similarity does not directly capture the underlying data and encoding relationships. ChartEval (Goswami et al., 2025) moves toward structural evaluation by parsing generated and reference charts into scene graphs and comparing their objects, attributes, and relations. For composite visualizations, the effectiveness of such structure-based evaluation additionally depends on the accuracy of automatic parsing and the consistency of structural decomposition, particularly when multiple encoding systems admit alternative but semantically equivalent descriptions. CompVis-Bench instead adopts a normalized representation of the retained data and encoding relationships, abstracting away representational ambiguity and presentation-level choices.

3 Dataset Construction

We construct CompVis-Bench to evaluate structural understanding of composite visualizations. We manually annotate a seed set and extend it with new structural and relational configurations.

3.1 Scope of Annotation

Reconstructing every detail of a visualization is neither necessary nor desirable for evaluating structural understanding. A rendered composite visualization contains information at different levels: the data relationships being represented, the encoding structure through which they are expressed, and the presentation choices used for their visual realization. We distinguish these as:

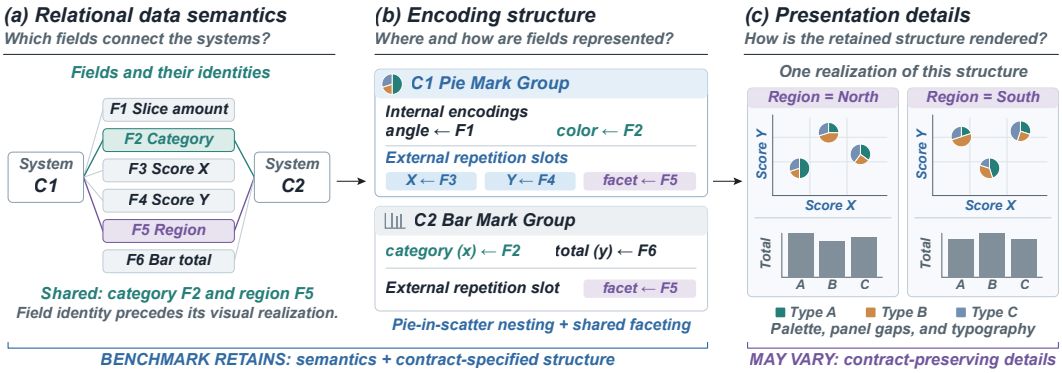

Figure 2: Annotation scope. A composite visualization is represented at three levels: (a) relational semantics, capturing data-field relationships across mark groups; (b) encoding structure, specifying how those fields are represented; and (c) presentation details, describing their visual realization. The benchmark retains the first two levels while abstracting away presentation-level variation.

Relational data semantics concerns the data relationships connecting the participating mark groups. In this work, the primary relationship of interest is *data-field sharing*: whether distinct mark groups represent the same underlying data field. Which fields are represented and how their identities correspond across mark groups therefore constitute the semantic target.

Encoding structure provides the grounding through which these relationships are expressed and verified. It includes the constituent mark groups, their required structural attributes, observable visual relations among them, and the bindings of data fields to internal encoding channels or external repetition roles. A field-to-color binding, for example, is retained because it specifies how a field is represented. Likewise, assigning a field to the horizontal rather than vertical position channel changes the reconstruction target even when the same field remains present.

Presentation details are realization choices that may vary without changing the relational semantics or encoding structure retained by the benchmark. These include exact hues, typography, spacing, as well as auxiliary elements such as axes, labels, legends, captions, and annotations.

Our annotation target retains the **first two levels**. Presentation details may vary as long as they leave the retained relational and structural information unchanged. Although presentation choices are not themselves part of the reconstruction target, their rendered manifestations provide visual evidence for recovering the underlying encoding structure and data relationships. This also motivates the inverse task considered in our benchmark: given only a rendered visualization, a model must use observable visual cues to reconstruct the mark groups and their field bindings, and from this structural grounding recover the underlying relational semantics.

Additional scope constraints. We retain only encoding dimensions that are directly interpretable from the rendered visualization. Position is explicitly represented by two spatial dimensions, with additional position dimensions introduced when the layout contains further spatial subdivisions. Other visual channels generally correspond to a single effective field. Multiple fields are permitted only when a channel contains perceptually separable dimensions—for example, when hue and luminance independently encode different fields—although no such cases occur in our dataset. We also do not require instance-level spatial identifiers such as bounding boxes or coordinates. Mark groups are identified through their encoding and relational properties rather than explicit localization.

3.2 Normalized Representation

High-level composition operations such as *concat*, *layer*, *facet*, and *nested* (Ying et al., 2024) can induce ambiguity in how a composite visualization is decomposed into structural units. This ambiguity arises in two ways. **First**, the appropriate *minimum granularity* is not always self-evident: for example, a grouped bar chart may be treated as a single encoding system or decomposed into multiple overlaid bar groups. **Second**, the effective granularity may change under composition, particularly with nesting, where a structure that is treated as an individual unit at one level may itself contain multiple encoding systems at a finer level. As a result, the same rendered visualization can admit multiple hierarchical descriptions that differ in their basic units, hierarchy depth, and intermediate composition nodes, even when they preserve the same underlying data relationships.

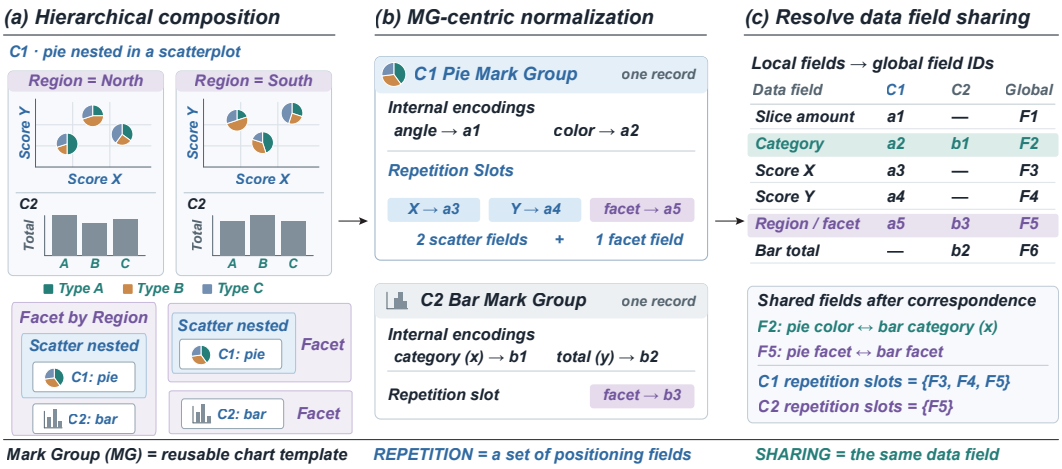

Figure 3: Mark-group relational representation. (a) Alternative hierarchical descriptions can arise from composition order and mark-group granularity. (b) Hierarchy-dependent bindings are flattened into internal bindings and group-level external field sets. (c) Shared data fields are assigned consistent visualization-wide identities across mark groups.

We therefore establish a fixed **mark-group granularity** in the annotation guideline and use the resulting mark groups as the basic reconstruction entities. Under this granularity, each structure is represented either as an individual mark group or as **repetitions** of the same mark group over different data slices, allowing us to abstract away the particular composition hierarchy. We treat *facet*, *nested*, and other simlilar composition uniformly as repetiton whenever they instantiate the

same encoding system over different subsets of the data. When multiple composition levels repeat the same mark group, we flatten these levels and collect their organizing fields as a set of **repetition slots**. Figure 3(b) illustrates this with repeated pie charts: the nested scatterplot contributes two repetition slots, while an additional facet contributes a third. This flattening removes explicit hierarchy-dependent structure and attaches the repetition slots directly to the resulting mark groups.

Flattening removes the explicit composition hierarchy, but the relationships induced by that hierarchy need not be lost. To preserve them, occurrences of the same underlying data field are assigned a common visualization-wide identity, which retains correspondences across mark groups. In Figure 3(b), for example, the repeated pie group has three repetition slots, one of whose fields is also shared with the repeated bar group. This shared field aligns the two repetitions, linking each barchart instance to the corresponding subset of pie-chart instances. Thus, the underlying cross-group correspondence is preserved through field sharing, as illustrated in Figure 3(c). The same mechanism also captures non-hierarchical coordination: for example, mark groups using a coordinated color palette or visually shared axes can be represented as sharing the corresponding color or position field, regardless of the particular presentation used to express that coordination.

3.3 SEED-SET ANNOTATION PROCESS

We construct the manually annotated seed set of COMPVIS-BENCH by re-annotating selected visualizations from the VAID corpus (Ying et al., 2024) using the representation defined above. The annotation was conducted by three authors of this paper, all of whom have experience with composite visualizations and more than four years of experience in visualization design and implementation.

Selection. VAID contains a broad collection of carefully designed visualizations from visual analytics systems. We selected visualizations containing multiple encoding systems, together with a small number of compositionally complex single-encoding idioms that form useful boundary cases for our representation. For example, sunburst and icicle plots may appear compositionally complex while constituting a single encoding system under our definition. We further excluded figures that were incomplete, insufficiently resolved, or visually ambiguous, where the rendered image did not provide sufficient evidence for reliable annotation.

Guideline calibration. Before full annotation, the three annotators independently annotated a shared subset of visualizations. Disagreements were used to refine the mark-group granularity, mark-group type vocabulary, applicable encoding channels, and field-naming conventions. Cases admitting multiple plausible decompositions were jointly reviewed until a consistent and perceptually grounded interpretation was established. The resulting guideline and vocabulary were then fixed.

Conventions. **First,** we distinguish mark-group types when similar encodings are realized under different coordinate systems. For example, normal and radial barchart are treated as different types. For geographic coordinates, we use geo point, geo line, and geo area marks as the basic granularity (Deng et al., 2023b). **Second,** we do not represent a graph as a separate composite entity. Instead, we first identify the mark groups and associate each link structure with the set of mark-group IDs it connects, allowing links to span multiple groups. An ordinary node–link graph is therefore represented as a scatterplot mark group with links; if the node marks are replaced by nested visualizations, the nested result is represented directly as the corresponding repeated mark group, with the links attached to it. Link source and target are treated as structural connectivity rather than encoding channels, since they are determined by the connected instances and rarely participate in cross-group field sharing. Other link properties, such as width or color, may still encode data fields.

Annotation procedure. Primary annotation was performed solely from the rendered visualization. After the image-based annotation was completed, the annotations were cross-validated using available sources, including VAID data-field information, figure captions, and original publications when necessary. Each visualization was annotated in four stages:

1. *Mark-group and repetition identification.* The constituent mark groups were identified according to the fixed granularity, together with whether each mark group was repeated.
2. *Data field and encoding annotation.* For each mark group, the represented data fields and their internal encoding channels were annotated. For repeated mark groups, the fields governing the organization of repeated instances were additionally recorded as repetition bindings.

3. *Field sharing.* Occurrences referring to the same underlying data field were assigned a common visualization-wide field identity, from which cross-mark-group field sharing was derived.
4. *Consistency verification.* The resulting annotation was checked for consistency in mark-group boundaries, repetition structure, field bindings, and visualization-wide field identities.

Review and adjudication. Each visualization was first annotated by one author and independently reviewed by a second. During review, the figure caption or corresponding publication could be consulted when necessary to verify the inferred data semantics. Disagreements concerning mark-group identification, repetition structure, field bindings, or field correspondence were discussed by the two annotators, with unresolved cases adjudicated by the third author.

Statistics. Seed-set (Figure 1) contains **121** visualizations, **268** mark groups spanning **38** mark-group types, and **584** annotated data fields. Among these visualizations, **87** contain repeated mark groups and **68** contain shared data fields. Overall, **114** visualizations exhibit cross-mark-group field sharing or repetition, while the remaining **7** are compositionally complex single-encoding idioms.

During annotation, we also recorded recurring presentation choices in VAID that serve as visual cues for data-field sharing across mark groups. These choices were treated as presentation-level design patterns and consolidated into a shared design pool rather than as strict or exhaustive semantic annotations. These patterns capture recurring image-level realizations and support controlled presentation variation; they do not imply a one-to-one correspondence between a visual cue and a shared field, nor do they constitute a complete reconstruction of the source visualization's styling.

3.4 Dataset Extension

Normalized annotations from seed-set provide verified structural building blocks that can be modified and combined to construct a broader range of composite visualizations. Our extension proceeds in two stages: we first generate new annotations by extending their encoding structures and data relationships, and then render each annotation with sampled presentation settings.

Annotation-based Extension. We design a set of extension directions over the normalized representation and use GPT-6-Astra to generate new annotations accordingly. These directions include adding new mark groups and mark-group types, introducing or extending repeated and nested structures, composing existing structures, and modifying data-field sharing across mark groups. We also include contrastive cases in which multiple visual components coexist without corresponding field-sharing or structural relationships. Each extension follows the corresponding annotation rules. Together with each annotation, GPT-6-Astra generates a mock-data script whose fields and sharing relationships are consistent with the generated structure.

Rendering. We develop a randomized renderer that instantiates each generated annotation using its corresponding mock data. For each example, the renderer samples from an expanded set of presentation choices based on those recorded during seed-set annotation, including color palettes, relative placement of mark groups, spatial partitioning, and the concrete arrangement of repeated or nested structures. These choices determine the visual realization while preserving the encoding structure and data-field relationships specified by the annotation. Each annotation is rendered once, and the resulting image–annotation pair is added to the candidate set for subsequent verification.

Verification. All extended examples are manually reviewed before inclusion. Each example is independently checked by two authors for visual clarity, plausibility of the composite design, and consistency between the rendered visualization and its annotation. Field-sharing relations are retained only when the correspondence is visually recoverable; relations that exist in the underlying data but are not perceptually evident are removed. The reviewers also verify mark-group identification, repetition structure, field bindings, and cross-group field correspondence. Disagreements are discussed by the two reviewers, with unresolved cases adjudicated by a third author. Minor inconsistencies are corrected, while examples requiring substantial revision are excluded.

Statistics. The extension set (Figure 1) contains **1,600** visualizations, **5,273** mark groups spanning **37** mark-group types, and **14,245** annotated data fields. Among these visualizations, **1,486** contain repeated mark groups and **912** contain cross-mark-group field sharing (**2,162** shared fields in total), including repetition-position fields. Overall, **1,517** visualizations exhibit cross-mark-group field sharing or repetition, while the remaining **83** contain multiple mark groups with neither relation.

4 EXPERIMENT

4.1 EXPERIMENT SETUP

Our experimental setup specifies three key components of the evaluation: the reconstruction task presented to each model, the procedure used to obtain a valid prediction for each visualization, and the alignment required before predictions can be compared with the reference annotations.

Task. Given only a rendered visualization, the model is asked to reconstruct the corresponding annotation defined in section 3. The prediction consists of a set of mark groups (MGs),

$$\widehat{Y} = \{\widehat{u}_1, \ldots, \widehat{u}_n\}, \qquad \widehat{u}_i = (\widehat{g}_i, \widehat{t}_i, \widehat{r}_i, \widehat{E}_i, \widehat{E}_i^{\mathrm{rep}}), \tag{1}$$

where $\widehat{g}_i$ is a visualization-local MG identifier, $\widehat{t}_i$ is its MG type, and $\widehat{r}_i$ indicates whether the MG is repeated. $\widehat{E}_i$ contains its internal encoding-channel–data-field bindings, while $\widehat{E}_i^{\mathrm{rep}}$ contains the bindings that organize repeated instances when $\widehat{r}_i$ is true. Data fields are represented by visualization-local identities, with the same identity used when the same underlying field appears across different MGs. The task therefore requires the model to recover both individual MG structures and field correspondences across MGs.

Model selection and procedure. We evaluate 12 LVLMs (Table 1), selecting representative flagship multimodal models from major proprietary and open/open-weight model families available at the time of evaluation. The evaluated models include GPT-6-Astra (OpenAI, 2026a), GPT-6-Sol (OpenAI, 2026b), Claude-Opus-5.5 (Anthropic, 2026b), Claude-Fable-5.1 (Anthropic, 2026a), Gemini-3.8-Flash (Google, 2026), Grok-4.7 (xAI, 2026), Qwen3.8-Max and Qwen3.8-27B (Qwen Team, 2026), DeepSeek-V4.1-Flash (DeepSeek-AI, 2026), GLM-5.3-Flash (GLM-5 Team, 2026), Kimi-K3 (Kimi Team et al., 2026), and MiniMax-M3 (MiniMax, 2026). For each image–model pair, we repeatedly query the model and retained its first output conforming to the required schema for evaluation. In practice, all models produced a schema-conforming output on the first attempt.

Alignment. Since predicted and reference annotations may use different visualization-local identifiers for the same MGs and data fields, we align MGs and cross-MG field identities before evaluation:

(1) *Mark-group alignment.* We construct type-compatible one-to-one partial mappings between predicted and reference MGs, first maximizing the number of matched groups and then their local structural agreement. Structural agreement compares internal encoding and repetition channels and their dimensionalities, while ignoring within-MG field identities. All mappings with the highest score are retained for field alignment.

(2) *Field alignment.* For each candidate MG mapping, we construct a visualization-wide one-to-one correspondence between predicted and reference fields participating in cross-MG sharing. The field mapping is chosen to maximize preserved cross-MG sharing relations. The jointly optimal MG and field mapping (M, ϕ) is used for all subsequent metrics.

4.2 EVALUATION METRICS

Given the alignment (M, ϕ), we evaluate composite visualization understanding at four levels of increasing completeness. We use F1 for fine-grained recovery, where both missing and additional predictions matter, and exact recall for higher-level completeness, where a reference unit is counted only when fully recovered. For local structural evaluation, we compare encoding channels and repetition structure without considering field identities; field identities are used only at the cross-MG level to determine whether bindings in different MGs refer to the same underlying data field.

L1: Mark-group identification. (a) MG Recovery F1 ($\mathrm{F1}_{\mathrm{mg}}$) measures recovery of MGs, including their types and repetition status. A predicted MG is counted as correct only when it aligns to a reference MG with the same type and repetition status.

L2: Local encoding structure. (b) Internal Encoding Recovery F1 ($\mathrm{F1}_{\mathrm{int}}$) measures recovery of the internal encoding channels that are used. (c) Repetition Recovery F1 ($\mathrm{F1}_{\mathrm{rep}}$) measures recovery of external repetition channels and their dimensionalities. (d) MG Exact Recall ($\mathrm{R}_{\mathrm{mg}}^{\mathrm{ex}}$) measures the proportion of reference MGs whose type, repetition status, internal encoding structure, and repetition structure are all exactly recovered.

L3: Cross-MG field correspondence. (e) Field-Sharing Recovery F1 ($\mathrm{F1}_{\mathrm{shr}}$) measures individual field-sharing relations across distinct MGs. Each relation links two occurrences of the same underlying field, with each endpoint specified by its MG, binding type (*internal* or *repetition*), and encoding channel. A relation is correct only when the shared field and both endpoints match under the global alignment. (f) MG-Pair Exact Recall ($\mathrm{R}^{\mathrm{ex}}_{\mathrm{pair}}$) measures the proportion of reference MG pairs whose complete sharing-relation sets exactly match under the same criterion. We additionally report matched-MG variants, $\mathrm{F1}^{*}_{\mathrm{shr}}$ and $\mathrm{R}^{\mathrm{ex}*}_{\mathrm{pair}}$, computed only over successfully matched MGs, isolating cross-MG correspondence recovery from upstream MG identification errors, indicating how well a model recovers field-sharing relations once relevant MGs have been correctly identified.

L4: Complete reconstruction. (g) Visualization Exact Match ($\mathrm{EM}_{\mathrm{vis}}$) measures the proportion of visualizations whose complete MG inventory, local encoding structures, repetition structures, and cross-MG field correspondences exactly match the reference.

4.3 RESULTS

Model overview. GPT-6-Astra achieves the strongest performance across most metrics, with Claude-Opus-5.5 and Claude-Fable-5.1 also performing consistently well (Table 1). Performance varies more among the remaining models, while DeepSeek-V4.1-Flash consistently yields the lowest scores across most evaluation levels .

Model	L1	L2			L3				L4
	$\mathrm{F1}_{\mathrm{mg}}$	$\mathrm{F1}_{\mathrm{int}}$	$\mathrm{F1}_{\mathrm{rep}}$	$\mathrm{R}^{\mathrm{ex}}_{\mathrm{mg}}$	$\mathrm{F1}_{\mathrm{shr}}$	$\mathrm{F1}^{*}_{\mathrm{shr}}$	$\mathrm{R}^{\mathrm{ex}}_{\mathrm{pair}}$	$\mathrm{R}^{\mathrm{ex}*}_{\mathrm{pair}}$	$\mathrm{EM}_{\mathrm{vis}}$
GPT-6-Astra	93.45	97.86	**87.99**	**83.61**	**63.57**	**78.44**	**53.23**	**67.24**	**45.50**
GPT-6-Sol	90.30	88.26	81.96	76.41	50.87	69.92	41.07	60.19	32.81
Gemini-3.8-Flash	90.32	88.50	82.45	78.23	48.42	70.73	37.59	58.34	34.19
Claude-Opus-5.5	**93.66**	98.01	87.32	81.45	56.63	70.75	45.58	58.61	37.44
Claude-Fable-5.1	91.73	**98.07**	84.49	80.08	55.34	73.17	44.47	60.96	39.88
Grok-4.7	84.91	97.23	76.43	70.80	43.13	73.85	32.40	57.51	28.94
Qwen3.8-Max	87.79	86.32	78.32	76.35	45.03	70.85	34.48	58.18	32.69
Qwen3.8-27B	81.20	79.18	70.73	65.49	33.48	60.40	25.55	51.81	20.50
GLM-5.3-Flash	86.27	84.41	79.26	73.61	46.51	72.39	35.33	57.71	30.19
Kimi-K3	85.21	83.93	75.46	71.63	39.94	65.22	28.78	52.93	26.88
MiniMax-M3	69.81	96.64	58.06	53.58	10.95	53.79	16.41	37.55	10.81
DeepSeek-V4.1-Flash	56.04	53.42	37.38	41.35	2.28	44.08	9.61	28.57	6.44

Table 1: Overall performance on all 1600 visualizations in COMPVIS-BENCH (%).

Cross-MG field sharing remains a major challenge. Table 1 shows that models recover local encoding structure substantially more reliably than cross-MG field correspondence. Internal-channel and repetition recovery remains relatively strong across the 12 models, whereas field-sharing recovery is considerably lower. The matched-MG sharing scores further show that, even after the relevant mark groups have been identified, maintaining consistent field identities across different encoding systems remains difficult. Full sharing scores decrease further because they additionally depend on correctly recovering both endpoint mark groups and their decomposition.

One-relation cases reveal a distinctive failure mode in field sharing. Nearly all models exhibit a pronounced drop in visualization exact match when the reference contains exactly one field-sharing relation (Figure 4(a)). To better understand this behavior, we examine false-positive and false-negative sharing predictions for samples with zero, one, and two reference relations. Both types of errors increase noticeably in the one-relation case, indicating that models are more likely to either miss the existing relation or introduce additional spurious ones. We hypothesize that this reflects a tendency toward two competing interpretations of the relationship between mark groups: models may treat the groups as largely independent components, as in an unrelated dashboard, leading to missed sharing relations; alternatively, once a cross-group correspondence is recognized, they may infer that the groups are more broadly coupled and consequently predict more sharing relations than are actually present. A single isolated sharing relation therefore appears particularly challenging, as it requires preserving a sparse and selective correspondence between otherwise distinct mark groups.

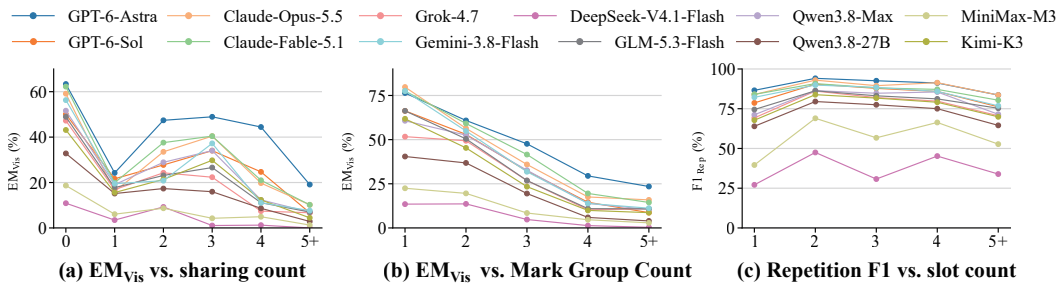

Figure 4: (a) $\mathrm{EM}_{\mathrm{vis}}$ decreases as the reference composite contains more field sharing. (b) $\mathrm{EM}_{\mathrm{vis}}$ generally decreases as the reference contains more mark groupss, with a non-monotonic middle caused by heterogeneous bins. (c) $\mathrm{F1}_{\mathrm{rep}}$ varies with the number of reference repetition slots.

Composite complexity affects global structure recovery. Figure 4(a) shows that visualization exact match generally declines as the number of field-sharing relations increases, although the decrease is relatively gradual apart from the pronounced dip at one relation discussed above. In contrast, Figure 4(b) shows a steeper decline as the number of reference mark groups increases. This suggests that increasing the number of structural components poses a stronger challenge to complete reconstruction than increasing the number of cross-group correspondences alone.

Repetition recovery. Figure 4(c) shows that single-slot repetition is often recovered less reliably than two- or three-slot repetition. The main errors arise from predicting the wrong repetition dimensionality: some single-slot cases are treated as non-repeated, while others are mistaken for two-dimensional repetition when a one-dimensional sequence does not fit in a single row and wraps into multiple rows. By contrast, two- and three-slot repetitions often exhibit clearer spatial organization and are easier to identify. Performance declines again when many repetition slots are present, suggesting that increasingly deep or fine-grained spatial partitioning also makes the repetition structure harder to recover. Overall, repetition recovery remains relatively strong, indicating that the flattened slot-based representation is generally compatible with the structural patterns current LVLMs can recognize, with errors concentrated in ambiguous layouts and highly partitioned structures.

Overall, models recover local structures well but struggle with globally consistent relationships across mark groups. Cross-MG sharing remain the main bottlenecks to complete reconstruction, which highlights the need to evaluate composite visualizations beyond isolated encoding recovery.

5 Conclusion, Limitations, and Future Work

We introduced CompVis-Bench, a benchmark for evaluating structural understanding of composite visualizations. The benchmark represents each visualization through normalized mark groups, encoding bindings, repetition structure, and cross-group field sharing, while abstracting away presentation-specific details. Using this representation, we construct an expert-annotated seed set and expand it through structural and relational extensions, randomized rendering, and manual verification. Experiments on 12 LVLMs show a clear gap between local and global structural understanding: models generally recover individual encoding structures and repetition more reliably than cross-group field correspondences and complete visualization structure. Performance further degrades as the number of mark groups increases, highlighting the difficulty of maintaining globally consistent structure in more complex composite visualizations.

Our benchmark captures only one aspect of visualization understanding. Value extraction, semantic interpretation of chart content, and data-oriented question answering are outside the current scope. Future work could combine structural evaluation with such downstream tasks to study whether, and to what extent, structural understanding supports broader chart comprehension. Coverage is another limitation. The mark-group types, encoding channels, structural patterns, and presentation choices in CompVis-Bench are derived from the current seed set and do not exhaust the design space of composite visualization. Future extensions could incorporate additional visualization types, composition strategies, encoding patterns, providing broader coverage and a stronger test of generalization.

Ethics Statement

This work studies machine learning methods in a research setting and is intended to advance the understanding and development of AI systems. Our experiments rely on publicly available datasets, benchmarks, and/or models, and do not involve human-subject experiments or the collection of personally identifiable or sensitive information.

As with many machine learning and large language model systems, the methods studied in this work may inherit biases, inaccuracies, or other undesirable behaviors from the underlying models and data. In addition, improved capabilities may potentially be used in unintended or harmful applications. We therefore caution against deploying the proposed methods in safety-critical or high-stakes settings without additional evaluation, appropriate safeguards, and human oversight. We limit our claims to the experimental settings evaluated in this paper and report relevant limitations and failure cases where applicable. Any released code, prompts, configurations, or other artifacts are intended primarily to support scientific research and reproducibility. We also aim to respect the licenses and terms of use associated with the datasets, models, and other resources used in this work.

AI Usage Statement

Generative AI tools, including ChatGPT and other large language model based assistants, were used during this work for language editing and improving the clarity of the manuscript. They were also used to assist with code implementation and debugging. All AI-generated suggestions, code, analyses, and text were critically reviewed and independently verified by the authors. AI-generated code was tested as part of the same experimental validation process as manually written code, and AI-assisted interpretations were checked against the underlying experimental results. Generative AI tools were not treated as autonomous researchers and were not relied upon as authoritative sources of scientific claims or citations. The authors take full responsibility for the methodology, experiments, claims, results, and final content of this work.

References

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1, September 2026a. URL https://www.anthropic.com/claude-fable-and-mythos-5-1.

Anthropic. Introducing Claude Opus 5.5, September 2026b. URL https://www.anthropic.com/claude-opus-5-5.

Xi Chen, Wei Zeng, Yanna Lin, Hayder Mahdi Al-Maneea, Jonathan Roberts, and Remco Chang. Composition and configuration patterns in multiple-view visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1514–1524, 2021. doi: 10.1109/TVCG.2020.3030338.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4.1-Flash.

Dazhen Deng, Weiwei Cui, Xiyu Meng, Mengye Xu, Yu Liao, Haidong Zhang, and Yingcai Wu. Revisiting the design patterns of composite visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 29(12):5406–5421, 2023a. doi: 10.1109/TVCG.2022.3213565.

Zikun Deng, Di Weng, Shuhan Liu, Yuan Tian, Mingliang Xu, and Yingcai Wu. A survey of urban visual analytics: Advances and future directions. *Computational Visual Media*, 9(1):3–39, 2023b. doi: 10.1007/s41095-022-0275-7.

GLM-5 Team. GLM-5: From vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Google. Introducing Gemini 3.8 Flash and 3.8 Flash Cyber, September 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/3-8-flash-and-3-8-flash-cyber/.

Kanika Goswami, Puneet Mathur, Ryan A. Rossi, Franck Dernoncourt, Vivek Gupta, and Dinesh Manocha. ChartEval: LLM-driven chart generation evaluation using scene graph parsing. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, pp. 86–93. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.ijcnlp-demo.10.

Waqas Javed and Niklas Elmqvist. Exploring the design space of composite visualization. In *2012 IEEE Pacific Visualization Symposium*, pp. 1–8. IEEE, 2012. doi: 10.1109/PacificVis.2012.6183556.

Kimi Team et al. Kimi K3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Xinhang Li, Jingbo Zhou, Pengfei Luo, Yixiong Xiao, and Tong Xu. ChartAnchor: Chart grounding with structural-semantic fidelity. *arXiv preprint arXiv:2512.01017*, 2025. URL https://arxiv.org/abs/2512.01017.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.findings-acl.177.

MiniMax. MiniMax M3: Frontier coding, 1m context, native multimodality—all in one model, June 2026. URL https://www.minimax.io/blog/minimax-m3.

OpenAI. GPT-6 Astra: A new generation of intelligence, September 2026a. URL https://openai.com/index/gpt-6-astra/.

OpenAI. Introducing GPT-6 Sol and Luna, September 2026b. URL https://openai.com/index/introducing-gpt-6-sol-and-luna/.

Shangpin Peng, Gengluo Li, Xingyu Wan, Chengquan Zhang, Hao Feng, Binghong Wu, Huawen Shen, Weinong Wang, Ziyi Cai, Zhuotao Tian, Han Hu, Can Ma, and Yu Zhou. ChartArena: Benchmarking chart parsing across languages, scenarios, and formats. *arXiv preprint arXiv:2606.01348*, 2026. URL https://arxiv.org/abs/2606.01348.

Jorge Poco and Jeffrey Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. *Computer Graphics Forum*, 36(3):353–363, 2017. doi: 10.1111/cgf.13193.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. CharXiv: Charting gaps in realistic chart understanding in multimodal llms. In *Advances in Neural Information Processing Systems*, 2024.

Chengyue Wu, Zhixuan Liang, Yixiao Ge, Qiushan Guo, Zeyu Lu, Jiahao Wang, Ying Shan, and Ping Luo. Plot2Code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3006–3028. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.164.

xAI. Introducing Grok 4.7, September 2026. URL https://x.ai/news/grok-4-7.

Renqiu Xia, Haoyang Peng, Hancheng Ye, Mingsheng Li, Xiangchao Yan, Peng Ye, Botian Shi, Yu Qiao, Junchi Yan, and Bo Zhang. StructChart: On the schema, metric, and augmentation for visual chart understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48 (7):8044–8059, 2026. doi: 10.1109/TPAMI.2026.3669664.

Cheng Yang, Chufan Shi, Yaxin Liu, Bo Shui, Junjie Wang, Mohan Jing, Linran Xu, Xinyu Zhu, Siheng Li, Yuxiang Zhang, Gongye Liu, Xiaomei Nie, Deng Cai, and Yujiu Yang. ChartMimic: Evaluating LMM's cross-modal reasoning capability via chart-to-code generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=sGpCzsfd1K.

Lu Ying, Aoyu Wu, Haotian Li, Zikun Deng, Ji Lan, Jiang Wu, Yong Wang, Huamin Qu, Dazhen Deng, and Yingcai Wu. VAID: Indexing view designs in visual analytics system. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2024. doi: 10.1145/3613904.3642237.

Zifeng Zhu, Mengzhao Jia, Zhihan Zhang, Lang Li, and Meng Jiang. MultiChartQA: Benchmarking vision-language models on multi-chart problems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11341–11359. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.566.

A Fixed Type–Channel Constraints

Table 2 lists the fixed set of internal channels permitted for each visualization type.

Table 2: Allowed internal channels for each visualization variation.

Variation	Allowed Internal Channels
bar	color, position.x, position.y
calendar_heatmap	color, position.x, position.y
circlepacking	color, size
connected_scatterplot	color, position.radius, position.theta, position.x, position.y, shape, size
contour	color, position.x, position.y
dotplot	position.x, position.y
euler_venn	color
geo_area	color, position.x, position.y
geo_line	color, position.x, position.y, stroke.width
geo_point	color, position.x, position.y, size
grouped_bar	color, position.x, position.x_offset, position.y, position.y_offset
hexbin	color, position.x, position.y
horizon_chart	color, position.x, position.y
icicle	color, position.x, position.y
link	source, stroke.color, stroke.width, target
multi_boxplot	color, position.x, position.y
multi_line	color, position.radius, position.theta, position.x, position.y
multi_radar	color, position.radius, position.theta
normalized_stacked_bar	color, position.x, position.y
parallel_coordinates	color, position.x, position.y
pie/donut/radial_bar	color, position.radius, position.theta
plain_area	position.x, position.x2, position.y, position.y2
point	color, position.radius, position.theta, position.x, position.y, shape, size
radial_area	color, position.radius, position.theta
radial_tree	color, position.radius, position.theta, stroke.color, stroke.width
rect_heatmap	color, position.x, position.y
single_boxplot	color, position.x, position.y
single_line	position.radius, position.theta, position.x, position.y
single_radar	color, position.radius, position.theta
stacked_area	color, position.x, position.y
stacked_bar	color, position.x, position.y
sunburst	color, position.radius, position.theta
tick_plot	color, position.x
tree	color, position.radius, position.theta, position.x, position.y, size, stroke.color, stroke.width
treemap	color, size
unit_bar	color, position.x, position.y
violin	position.x, position.y
word_cloud	color, size, text

B Annotation Construction and Randomized Rendering

This appendix describes how we construct annotations and how we instantiate them as images. The central design choice is to separate the semantic description of a visualization from its presentation. An annotation specifies what visual units are present and how their data fields are encoded; a renderer then samples a compatible layout and style configuration. Consequently, multiple images can realize the same annotation while preserving its encoding structure and field relationships.

B.1 Annotation target and representation

Our target is a set of independently identifiable chart or mark-group templates. A template is represented by a chart identifier C_i, an allowed variation label, a set of internal encodings, and a set of external encodings. Internal encodings describe channels within the template, such as Cartesian

position, polar position, color, size, shape, text, or stroke properties. External encodings describe properties of the whole template instance: the fields that determine its position in a repeated composition, and, when visible, fields that determine its overall size or color. A link is represented separately and refers to the chart identifiers in `link_targets`; it does not introduce a graph container or duplicate the node marks.

Fields are anonymous identifiers ($F_1, F_2, \ldots$) rather than source-column names. The same identifier is reused only when the image supports the same observable field role. The identifier may therefore occur in both an internal channel and an external position, size, or color property, but each occurrence must remain in its correct scope. Repetition fields are unordered sets of field identities. They specify how a template is indexed by an enclosing composition; they do not count marks, screen axes, or nesting levels. Repeated instances share one template entry instead of being annotated independently.

The annotation vocabulary contains 38 variation labels. Type normalization is applied at annotation time: for example, trees and dendrograms use `tree`, pie, donut, and radial-bar forms use `pie/donut/radial_bar`, and node–link or flow views are decomposed into visible node glyph templates plus a separate `link`. A native tree retains its own hierarchical edges and is not decomposed into a second link. Constant, decorative, or unidentifiable bindings are omitted. In particular, a channel being allowed by the schema is not evidence that the image contains a data binding for that channel.

The formal labels use the external-encoding protocol. Each file contains only the top-level key `charts`; each ordinary entry contains `chart_id`, `variation`, `encodings`, and `external_encodings`, while a link additionally contains `link_targets`. This compact representation is intentionally independent of the synthetic data tables and of the rendering configuration.

B.2 Annotation generation directions

The construction starts from 121 seed visualizations. These seeds are reviewed source examples, rather than images produced by the randomized renderer. We first recover a semantic annotation for each seed: visible MGs, variation labels, internal channels, field identities, whole-instance position fields, and link dependencies. We then generate new annotations by applying controlled transformations to these seed annotations. The transformations change the semantic supervision signal, whereas the later renderer changes only the presentation of a fixed annotation. The annotation transformation directions are listed below.

1. **Variation transformation.** Replace a template by another compatible variation family. The replacement must obey the target variation's channel whitelist and preserve or explicitly replace the observable roles. A grouped form, for example, requires a valid series or offset field.
2. **Reassign channel bindings.** Exchange compatible positional roles, such as $x \leftrightarrow y$ or $\theta \leftrightarrow r$, while retaining the underlying field identities. This is a semantic annotation transformation and is allowed only when the target variation and field types support both roles.
3. **Add a mark group (MG).** Add an independent chart or mark-group template with a fresh chart identifier and independent field IDs. A relation to an existing MG is introduced only by a subsequent explicit sharing operation.
4. **Add a related MG.** Add a chart that reuses one or more existing data fields. Reused F_i identifiers express sharing only when type, unit, entity, and granularity are compatible; otherwise the new MG receives independent fields.
5. **Share data fields.** Merge compatible field roles, including pairwise sharing among several MGs. Equal value ranges or equal screen directions are not sufficient; the relation must have a coherent key, unit, and data role.
6. **Add repetition.** Repeat one MG, a selected group of MGs, or the whole seed composition. The annotation records valid external position fields and one template entry represents all instances. This operation records repetition but does not choose whether the renderer later uses a facet, nested placement, or another layout.

7. **Add an external encoding.** Add a whole-instance position, size, or color binding only when its visual effect is observable. Size must change the complete footprint, and external color must be constant within an instance while varying across instances.
8. **Add an internal channel.** Add an observable color, size, shape, text, offset, or stroke channel to an existing variation. The channel must be allowed for that variation, supported by a field with the correct role, and nonconstant in the rendered marks.
9. **Add or revise a link relation.** Add a link template or change its target-template list. Target IDs must exist, be distinct, and not point to the link itself; endpoint pairing and curve geometry remain rendering decisions.
10. **Compose transformations.** Apply several directions sequentially, for example add an MG, share a field, and repeat the resulting group. Each intermediate annotation is checked before the next operation and the complete operation trace is retained in provenance.

The transformations are sampled from a constrained candidate set rather than applied as independent edits. For every candidate, we check the variation-specific channel whitelist, field-ID format and reuse, field type and granularity, external-versus- internal scope, repetition domains, and link references. An addition is accepted only if the resulting annotation still describes a realizable visual structure. In particular, adding an MG does not automatically create a shared field, repetition does not automatically create a nested relation, and an external channel is not inferred from the mere presence of repeated instances. Invalid candidates are discarded before synthetic data or presentation styles are sampled.

The annotation transformations define the semantic target only. In particular, adding repetition records the fields that index repeated instances, but does not decide whether those instances will be displayed as a facet, as nested glyphs, or in another composition. Likewise, an external position, size, or color binding specifies a whole-instance data role, but does not prescribe its pixel coordinates, footprint, or palette. Those choices are deferred to the randomized rendering stage below.

The resulting annotations can therefore be used in several composition settings without changing their semantic content. The renderer may assemble MGs into dashboards, concatenated views, layers, facets, nested views, or linked views when the selected composition is compatible with the annotation. These are rendering constructs, not additional annotation directions.

B.3 Randomized rendering from annotation to image

The second stage is implemented as a renderer-side translation from an annotation to a render specification. The annotation is treated as read-only. A render specification is a self-contained scene object with five parts: (i) a canvas declaration containing width, height, background, and coordinate bounds; (ii) datasets containing typed columns and synthetic rows; (iii) chart specifications containing a chart identifier, variation-specific chart type, dataset identifier, and channel-to-field encodings; (iv) a recursive composition tree; and (v) appearance properties such as palettes, gradients, font settings, and mark defaults. The specification is then passed to the offline SVG renderer and rasterized to PNG.

The composition tree uses a small explicit grammar. A chart node places one chart specification in a frame. A layer node draws compatible chart nodes in the same frame. A concat node allocates horizontal, vertical, angular, or radial regions using gaps and relative weights. A facet node filters a child subtree by one repetition field and allocates one cell per field value. A nested node obtains visible anchors from a parent chart and places a child subtree at those anchors. A linked node adds a link specification whose targets are existing chart nodes. Each node carries its frame, filters, and inherited repetition context, so the renderer can verify that the scene still realizes the annotation's external position fields.

For each annotated field, the renderer samples a domain consistent with its role. Categorical fields receive finite categories, quantitative fields receive numeric sequences or trend-shaped values, and hierarchical fields receive explicit parent and depth relations. Geographic fields use a local geometry asset. Repetition fields are materialized in the data rows so that every selected composition can filter the same template into the corresponding context. Link rows are resolved against the visible chart-template instances named by `link_targets`. These values make the annotated encodings observable; they do not add channels or modify field identities.

The renderer then samples a valid composition. Shared positional fields are used as evidence for compatible choices: a shared x or y may support a layer or an aligned concat, while a view without a justified alignment is placed in an ordinary horizontal or vertical concat. Repetition fields may be realized through a facet, through nested children anchored at visible parent marks, or through a multi-field scaffold. The decision is made by the renderer, not by the annotation generator. Every selected composition node is checked against the inherited external position fields, and the scene is rejected if a template loses a field, is rendered twice, or is left without a valid context.

The rendering pipeline can be summarized as follows:

1. Validate the annotation's variation labels, channel whitelist, field IDs, external encodings, and link targets.
2. Generate synthetic rows and domains using the annotation's field identities and repetition contexts.
3. Compile each template into a chart specification, preserving every internal binding and whole-instance external binding.
4. Sample a compatible composition tree: layer, concat, facet, nested, scaffold, or linked composition. Sample the spatial order, relative weights, gaps, margins, and child footprints subject to geometric constraints.
5. Sample presentation properties such as palettes, gradients, typography, mark shapes, polar style, hierarchy layout, and connector geometry.
6. Render the resulting scene exactly once to SVG and rasterize it to PNG. The PNG is paired with an exact copy of the input annotation.
7. Save the sampled seed, scene, SVG, hashes, and provenance so the same scene can be replayed without making a new random choice.

The randomized renderer therefore changes the visual realization while preserving the semantic object being supervised. A different seed can move MGs, choose a different facet or nested presentation, change the relative panel sizes, or select a different palette, but it cannot change a variation, swap an annotated channel binding, rename an F_i, add a semantic field, or modify a link target.

B.4 Randomized presentation choices

Once the semantic annotation is fixed, randomization operates on presentation-level degrees of freedom only. It does not change a variation label, exchange `position.x` and `position.y`, alter a field identifier, or infer a missing channel. It can, however, decide how the annotated MGs are spatially assembled. The following choices are sampled when the corresponding option is compatible with the annotated structure.

1. **MG assembly and spatial layout.** Assemble the annotated MGs as a layer, an aligned concat, an ordinary horizontal/vertical concat, or a recursively nested composition. Shared x places compatible views in vertically separated panels with a common horizontal coordinate; shared y uses horizontally separated panels with a common vertical coordinate. Without justified alignment, panel order, relative weights, gaps, and margins are sampled subject to readable geometry.
2. **Repeated placement.** Realize external position fields as separate facet cells, nested children anchored at actual parent marks, or a two-dimensional scaffold for multiple independent repetition fields. The field list, domain, and number of semantic templates remain unchanged.
3. **Panel and nested geometry.**
   - Sample facet columns, row/column traversal, wrapping, inter-cell gaps, borders, and label allocation within the image budget.
   - Sample nested child width, height, local margins, permitted offsets, and child-axis visibility. Children must remain inside the parent plot and must not overlap one another.
   - Sample canvas aspect ratio, outer margins, plot padding, clipping, and draw order while keeping every required semantic unit visible.

4. **Color and size.**
   - Sample categorical palettes and sequential gradients for annotated color channels.
   - Apply external color at whole-instance scope rather than confusing it with mark-level color.
   - Let an annotated external size field modulate the footprint of a complete repeated instance; fixed panel size is not treated as a data encoding.
5. **Cartesian, polar, and hierarchical style.**
   - Sample compatible baseline placement, aspect ratio, bar orientation, parallel-coordinate orientation, and axis/guide density without exchanging the annotated x/y binding.
   - Choose pie versus donut when no data-driven radius is annotated, and sample radial bands, angular offsets, and inner-radius ratios without inventing a radius field.
   - Choose tree or cluster layout, treemap tiling strategy, and ordinary stacked-area or streamgraph presentation while retaining the annotated fields.
6. **Marks, typography, and connectors.** Sample point shapes, font size, label visibility, stroke opacity, connector curve family, control-point displacement, draw order, and connector opacity from bounded options. Stroke color and width are mapped only when the annotation contains the corresponding link channels; constant line width is never fabricated.

The probabilities are conditional rather than quota constraints. For example, a layering probability of p is applied only to pairs that share a compatible coordinate system; incompatible charts remain concatenated. Likewise, a nested-placement probability affects only candidate parent–child pairs for which the annotated fields identify a visible anchor. This conditional sampling prevents style variation from changing the semantic interpretation of an annotation.

For each sample, the global seed and the sample identifier determine a local seed. Thus, changing the order of files does not change the realization of an existing sample. The provenance contains the realized composition tree and presentation decisions, so a saved scene can be rendered again without resampling. Each annotation is rendered once for the candidate set; we do not generate several images and select one based on its appearance.

C Complexity-Conditioned Metric Trends

Figures 5 and 6 show all twelve evaluation metrics after stratifying the 1,600-image extension set by two reference-side complexity counts. Each curve is the mean of the corresponding sample-level scores for one model within a bin; the vertical axis is reported as a percentage. The final bin contains counts of five or more. Sharing and pair metrics are omitted for bins whose references contain no sharing relation, because those bins do not define a sharing task.

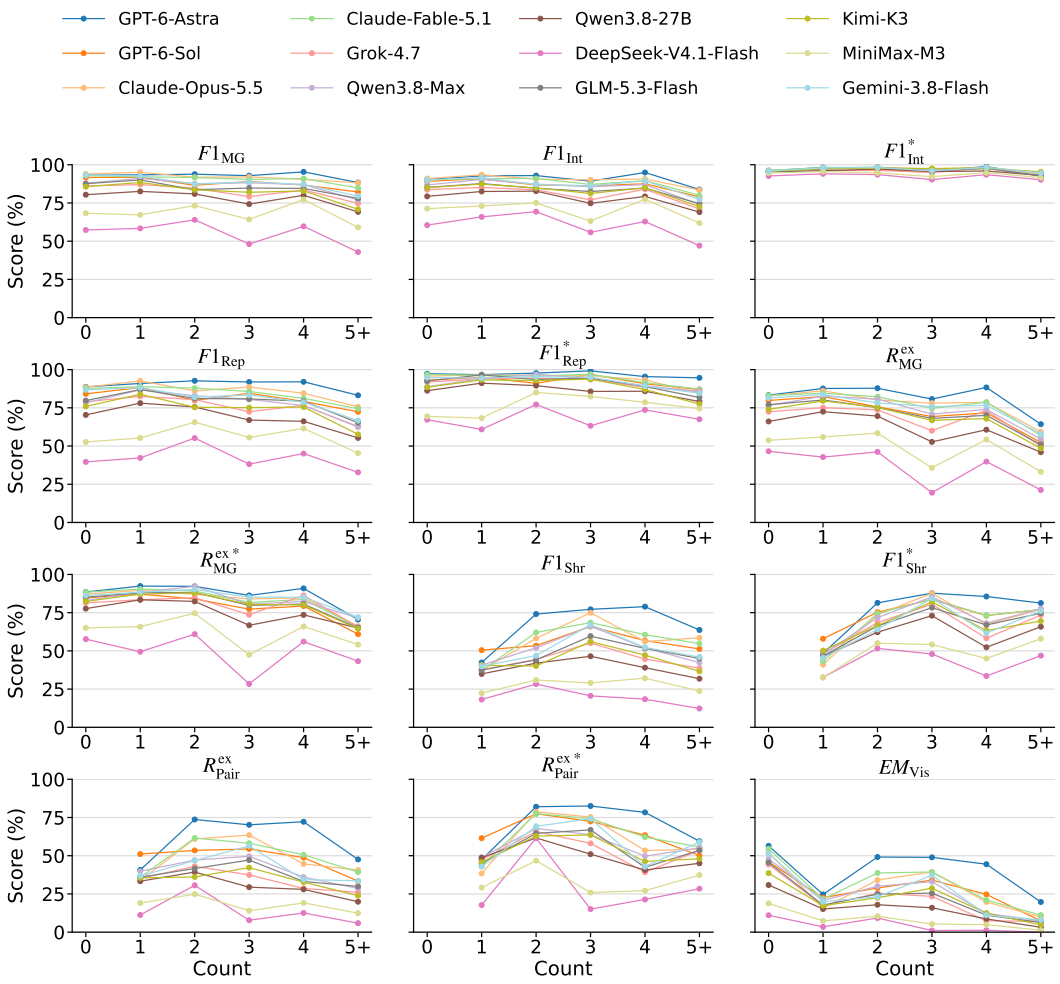

Figure 5: Metric trends by the number of reference sharing facets. A sharing facet is one reference `sharing_all_groups` relation. The twelve panels use the same model colors and axis scale as the main results overview; three panels are placed in each row for appendix readability.

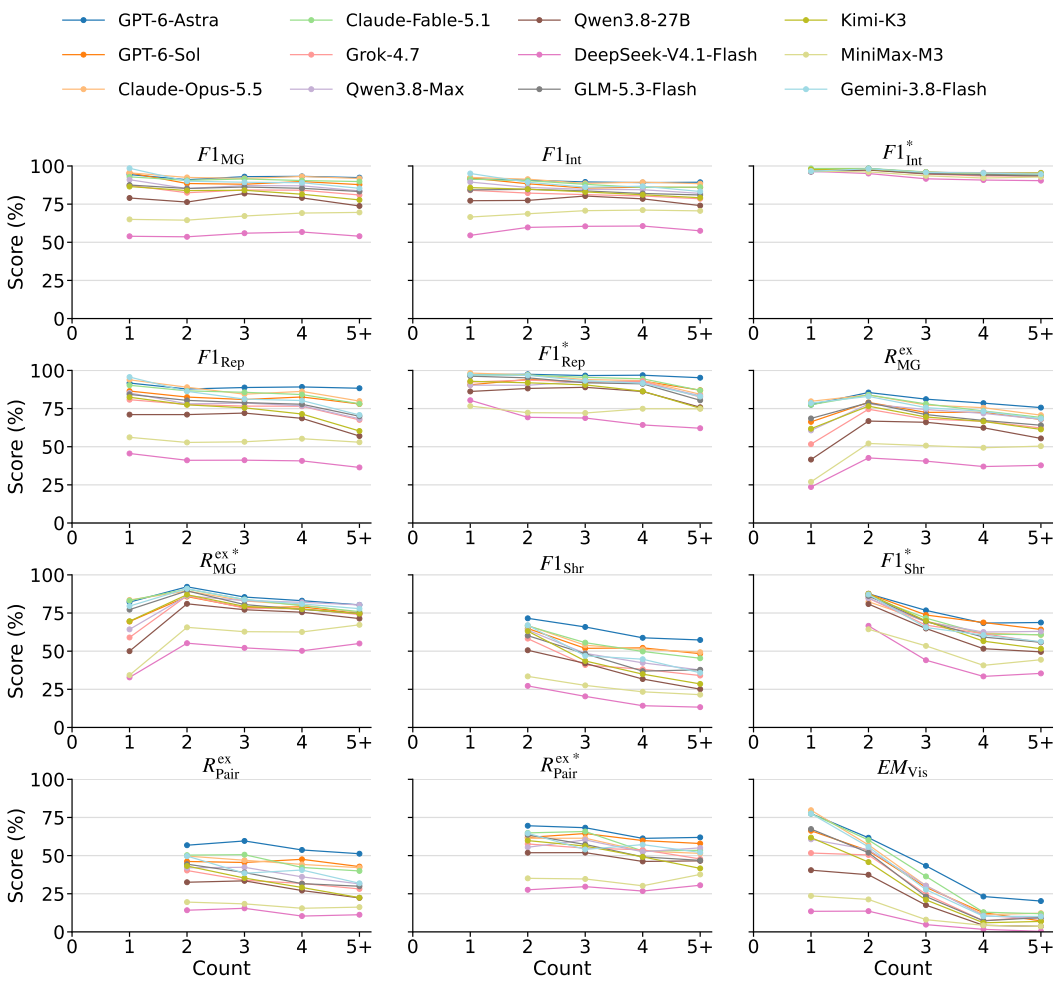

Figure 6: Metric trends by the number of reference mark groups (MGs). The bins are exact counts from zero through four followed by a five-or-more bin.