SCALING PROPERTIES OF TEXT CONDITIONING IN VISUAL GENERATION

Anonymous authors
Paper under double-blind review

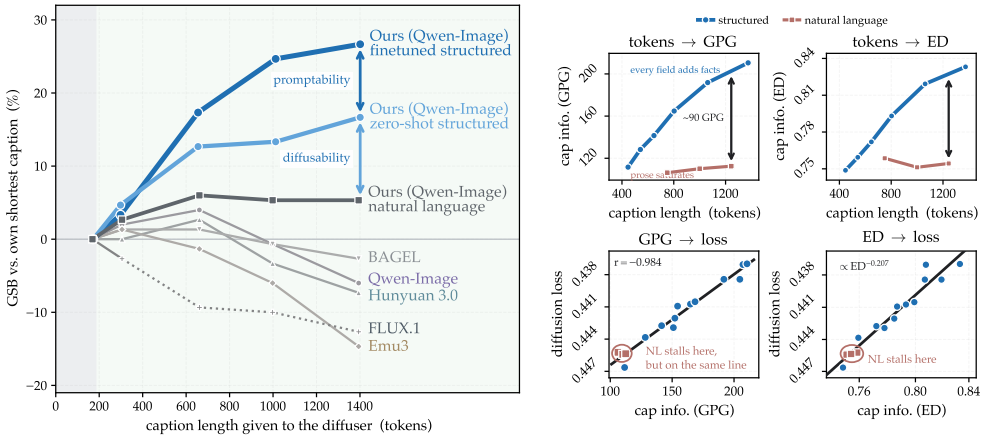


Figure 1: **Information, not token count, is what scales in prompt enhancement for image generation.** *(Left)* GSB net preference (150 prompts, VLM judge) against each system's *own* shortest caption: longer prose ends below it for every open-weight model, even when *trained*, whereas our structured prompts rise with length and *finetuning* adds a large margin over *zero-shot*. *(Right)* Prose saturates in information while structure keeps gaining *(top)*; at matched information both land on one fit *(bottom)*.

ABSTRACT

We study empirical scaling properties for text conditioning in visual generation. Such properties have rarely been measured because diffusion loss does not scale with the number of tokens in natural-language prompts. Surprisingly, we find that the converged diffusion loss scales with the amount of structured language in the prompt. To quantify structured language, we adapt two complementary measures: a white-box likelihood metric (GPG) and a black-box attribute metric (ED). Across controlled training runs, the converged diffusion loss decreases approximately linearly with GPG and follows a power law with ED. Guided by these scaling properties, we improve *diffusability* by constructing structured prompts with semantic and geometric annotations derived from images, and improve *promptability* by training a prompter through supervised fine-tuning, cold-start, and verifier-gated on-policy distillation. The resulting system outperforms all evaluated open-weight models on nearly every compositional, reasoning, and world-knowledge benchmark, while matching or surpassing the strongest closed-weight models on most evaluations.

1 INTRODUCTION

Large language models have advanced through scaling model size, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022), and text-to-image generation has followed the same recipe with larger diffusion backbones, heavier training runs (Podell et al., 2023; Esser et al., 2024), and larger captioned-image corpora (Schuhmann et al., 2022). The analogy hides an asymmetry: a language model learns from the text stream itself, whereas a text-to-image model learns text conditioning only through image–caption pairs, so visual content that a caption binds ambiguously reaches the model weakly and content it omits does not reach it at all. The image-grounded information in captions may therefore limit what a generator can learn to recover from text, yet it has rarely been treated as an

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.

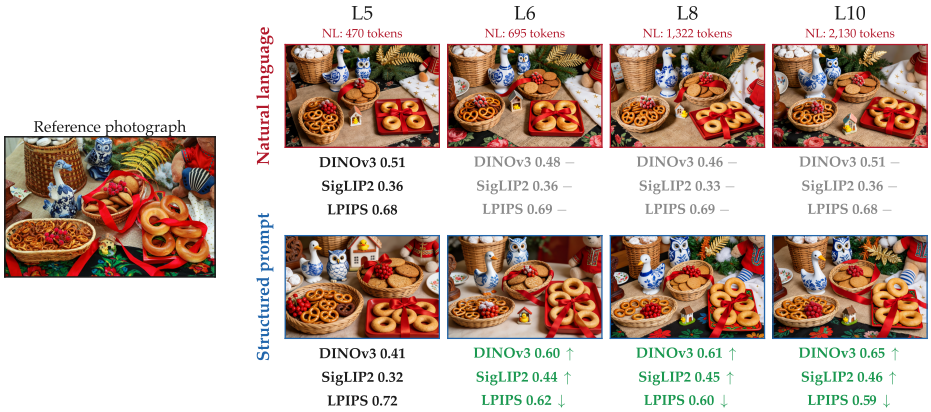


Figure 2: **A fixed-backbone probe separates caption information from caption length.** With the same Qwen-Image backbone and seed, we reconstruct a held-out reference (*left*) from NL (top) and SP (bottom) captions at four levels. Longer NL captions of the same entities and relationships leave reconstruction flat, while restoring SP fields improves DINOv3/SigLIP2 cosine (↑) and LPIPS (↓). Controls and prompts: Appendices E.2 and I.3.

explicit training variable. We ask whether increasing it improves visual generation, particularly on information-dense prompts where current systems struggle (Jiao et al., 2025; Yang et al., 2026; Wei et al., 2025).

Figure 1 previews the answer. As natural-language prompts grow longer, every existing system we evaluate peaks early if at all and ends below its own shortest caption, and a diffuser of ours trained on the same ladder improves only slightly before saturating: prose does not scale, whether a system is prompted with it or trained on it. Quality keeps rising only when the added tokens carry more image-grounded information, which structured captions supply and which also predicts converged diffusion loss.

A fixed-backbone reconstruction probe (Figure 2) shows the mechanism: natural-language (NL) captions that elaborate the same entities at greater length leave reconstruction flat, whereas a *structured prompt* (SP), a typed JSON caption with fields for global context, per-element properties and geometry, and cross-element relations, improves it as fields are added. We quantify caption information with two adapted metrics: *Grounded Perplexity Gain* (GPG), the likelihood gain from revealing the image to a frozen vision–language model (Favero et al., 2024), and *Effective Detailness* (ED), the precision and recall of caption attributes against image-grounded references (Wang et al., 2025b; Cheng et al., 2024). With images, architecture, initialization, and compute fixed, converged diffusion loss across NL and SP configurations is well fit by a linear function of GPG and follows a power law in ED. We call these relations the *scaling properties of text conditioning*; once calibrated, they can rank candidate caption designs before another training run.

These properties isolate *diffusability*, how effectively a caption representation exposes and organizes image-grounded information for diffusion learning, which we raise with an image-to-SP annotation pipeline that combines a VLM with frozen pose, depth, and segmentation experts. At test time no image is available, so an LLM prompter must infer unspecified visual details while preserving the request; we call this capability *promptability*. It improves zero-shot with prompter scale and chain-of-thought, and further with supervised fine-tuning (SFT), cold-start distillation, and verifier-gated on-policy self-distillation (OPSD) from an image-conditioned teacher; an agentic refine–render–judge loop adds gains at inference. The combined system leads every evaluated open-weight model on all but one reported metric and matches or surpasses the strongest closed systems on most.

In summary, we show how to scale text conditioning in visual generation. Our contributions are:

- **Scaling properties of text conditioning.** We adapt GPG and ED to quantify caption informativeness and show that both predict converged diffusion loss under a fixed training recipe.
- **Raising diffusability through structured prompts.** SPs organize image-grounded content into named fields, raising measured informativeness and lowering diffusion loss without architectural changes.
- **Raising promptability through training and inference-time refinement.** Our SFT–cold-start–RFT pipeline culminates in verifier-gated on-policy self-distillation (OPSD), while field-level agentic refinement further improves generation at inference time.

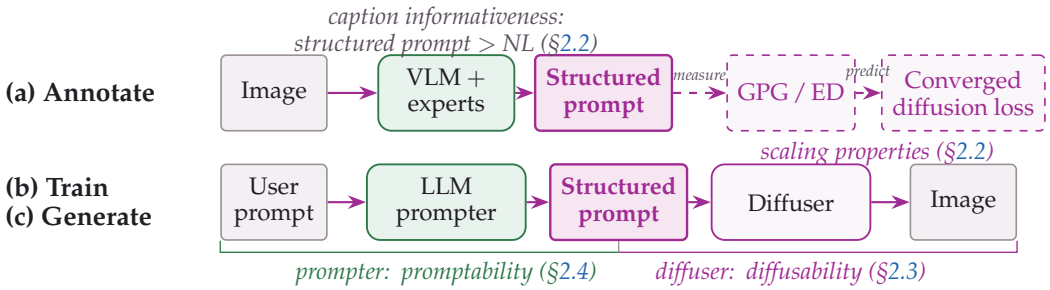


Figure 3: **Method overview.** *(a)* A VLM and frozen experts map each training image to an SP (§2.3), whose GPG/ED predict converged loss (§2.2). *(b)* The diffuser learns SP → image; the prompter learns user prompt → SP. *(c)* Inference runs the chain end to end (§3).

- **Scaling text conditioning end to end.** Combining the two factors yields broad gains across compositional, reasoning, and world-knowledge evaluations.

2 METHOD

Our goal is to understand and exploit the image-grounded supervision that captions provide for visual generation. We first test the standard caption-side intervention of lengthening free-form natural-language (NL) captions and find that it does not reliably increase useful supervision (§2.1). This limitation motivates the *structured prompt* (SP), a typed JSON caption that organizes image-grounded variables into named fields. Figure 3 summarizes how the SP becomes the shared interface of our method. Using GPG and ED, we quantify the information exposed by each caption and relate it to converged diffusion loss in a controlled training sweep (§2.2). The resulting scaling properties isolate the diffusion-side role of the caption interface; realizing that interface from a user request introduces a second, LLM-side factor. We summarize the two as *Diffusability*×*Promptability*: diffusability captures how effectively the caption representation exposes and organizes the supervision the diffuser learns from, while promptability captures how effectively an LLM prompter instantiates that interface from a user request. We raise diffusability by annotating training images as SPs and incorporating them into diffusion training (§2.3), and raise promptability by scaling and training the prompter (§2.4).

2.1 FROM NATURAL-LANGUAGE CAPTIONS TO STRUCTURED PROMPTS

Most modern T2I systems condition their diffusers on free-form natural-language (NL) captions, whether collected from web-scale corpora, recaptioned, or produced by LLMs (Rombach et al., 2022; Podell et al., 2023; Black Forest Labs, 2024; Betker et al., 2023). A common caption-side intervention is to make this NL condition longer and more detailed. The NL row of Figure 2 tests whether this helps using a fixed-backbone reconstruction probe. A single Qwen-Image (Wu et al., 2025a) backbone trained on captions spanning different lengths and richness levels attempts to regenerate one held-out image from each of four NL descriptions of the same annotated entities and relationships. To our surprise, reconstruction remains essentially flat as the captions become substantially longer, with the backbone and sampling seed fixed. In this probe, the added prose elaborates existing content without improving what the diffuser recovers. This saturation motivates testing whether an organized representation can expose image-grounded variables more effectively. We therefore introduce the *structured prompt* (SP), a typed JSON caption that assigns each represented visual variable to a named field.

An SP organizes image-grounded information at three scopes, as illustrated in Figure 4 (all fields: Appendix Table 7). *Global* fields describe the overall intent, scene, atmosphere, photography, style, and lighting. Each foreground object receives a *per-element* entry for its identity, attributes, actions, bounding-box position, optional depth, and photography, while *cross-element* relationships bind these entries to one another. The SP row of Figure 2 shows a different trend: reconstruction improves steadily as additional field groups are included, unlike the flat NL length sweep. The probe motivates this representation, but one image cannot establish whether caption informativeness predicts diffusion learning across training configurations. We therefore next introduce caption-side information measures and a controlled training sweep.



Figure 4: **Structured-prompt schema by example.** *Left:* image with bounding boxes and element IDs. *Center:* abridged global, per-element, and cross-element fields. *Right:* selected per-element fields.

2.2 Caption Informativeness Predicts Training Loss

We quantify caption information with two complementary metrics adapted from prior work, one white-box and one black-box (Figure 5; i.e., with and without access to a VLM's token log-probabilities), and relate both to converged diffusion loss in a controlled sweep.

Grounded Perplexity Gain (GPG). GPG (Favero et al., 2024) measures how much revealing the paired image raises a caption's likelihood under a frozen VLM judge M (Qwen3.5-397B-A17B (Qwen Team, 2025)). After the canonicalization and content-mask protocol of Appendix C.1, let $y_{1:T}$ be the caption tokens for image I and $m_{1:T} \in \{0,1\}^T$ the content mask:

$$\mathrm{GPG}(y, I) \triangleq \sum_{t=1}^{T} m_t \Big[\log p_M(y_t \mid I, y_{<t}) - \log p_M(y_t \mid \varnothing, y_{<t}) \Big], \tag{1}$$

where $\varnothing$ denotes the matched no-image pass. It is an operational estimate of caption–image mutual information under M. GPG is a total rather than a per-token rate, so it can grow with caption length; we report its mean over a shared 30,000-image pool, with the judge, template, and preprocessing fixed across configurations.

Effective Detailness (ED). ED is a complementary semantic measure that needs no log-probabilities. Adapting caption-detailness evaluation (Wang et al., 2025b), independent image-side and caption-side proposers extract object–attribute–relationship–grounding tuples from pixels and from text, and a paraphrase-tolerant matcher marks which image attributes the caption covers and which caption attributes the image supports. With caption-side attribute precision P_A and image-side attribute recall R_A,

$$\mathrm{ED}(y, I) \triangleq F_{0.5}(P_A, R_A), \tag{2}$$

where $F_{0.5}$ penalizes unsupported caption attributes more than omissions (van Rijsbergen, 1979; Cheng et al., 2024); Appendix Table 5 tests this choice. The image-side proposer never sees the caption; we use a 10%-trimmed mean over the same 30,000 images (Hampel, 1974) (protocol in Appendix D).

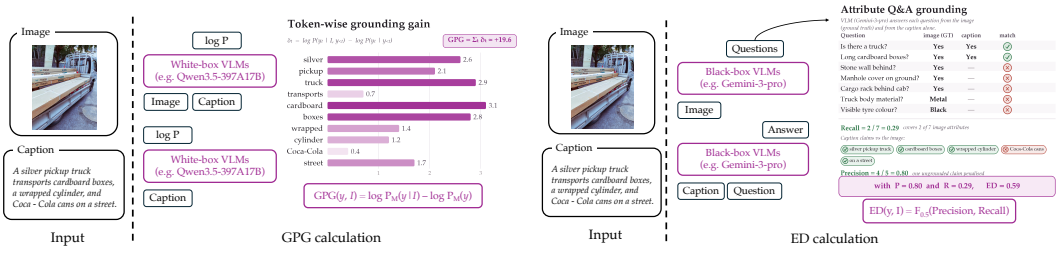


Figure 5: **The two informativeness measures.** *Left:* GPG, Eq. (1). *Right:* ED, Eq. (2). Length trends: Figure 1 (top right).

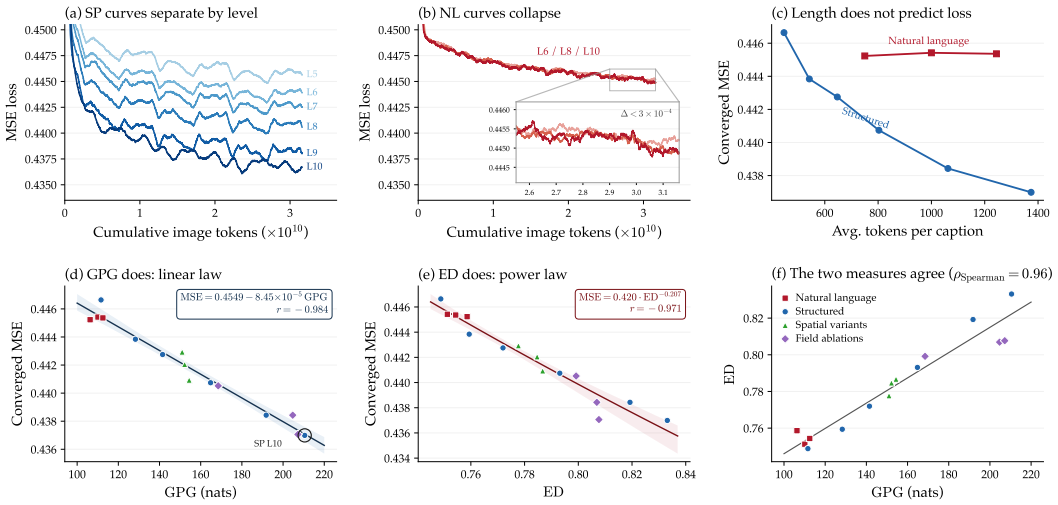


Figure 6: **The scaling properties of text conditioning: loss follows information.** *(a–b)* Training curves separate across SP levels but overlap across NL levels. *(c)* Length does not place NL and SP on a common loss trend. *(d–e)* Converged MSE vs. GPG and ED over all 15 sweep points; bands: central 95% range over resampled sweep settings. *(f)* GPG and ED agree on the configuration ranking.

The scaling properties. We ask whether caption informativeness predicts the converged diffusion loss reached at a common training budget. The controlled sweep has 15 caption configurations built from the same full image annotations: three NL controls that verbalize the same facts at increasing length, six nested SP levels (§2.3), and six variants that replace bounding boxes with 3×3, 5×5, or 9×9 grid locations or mask the scene, bounding-box, or relationship fields (Appendix C.3). For each configuration we train a separate diffuser from the same continued-training checkpoint of BAGEL (Deng et al., 2025) to 2.84×10^{10} cumulative image tokens; images, initialization, architecture, and optimization are identical, so the caption configuration is the only changing variable (Appendix E; we use BAGEL because 15 runs on the larger Qwen-Image backbone would be prohibitively expensive).

Caption length does not explain the outcomes. NL and SP captions do not share a length–loss trend (Figure 6c): longer NL captions nudge GPG upward, partly because GPG is a token sum, yet leave ED and loss essentially unchanged, whereas restoring SP fields raises both measures and lowers loss (Figures 1, right, and 6a–b). Measured informativeness instead provides a common axis: across all 15 configurations, converged training MSE is well fit by a linear function of GPG and a power law in ED (Figure 6d–e),

$$\mathrm{MSE} = 0.4549 - 8.45 \times 10^{-5} \cdot \mathrm{GPG}, \qquad r = -0.984, \tag{3}$$

$$\mathrm{MSE} = 0.4200 \cdot \mathrm{ED}^{-0.2073}, \qquad r = -0.971. \tag{4}$$

where r is the Pearson correlation in raw space for GPG and in log–log space for ED; the few local reversals are small (Appendix C.4). We call these relations the *scaling properties of text conditioning*: within the calibrated architecture, recipe, and measured range, caption information predicts matched-budget converged loss. The two measures also agree on the configuration ranking (Spearman $\rho_{\mathrm{GPG,ED}} = 0.96$, Figure 6f).

The relations are predictive: fits learned only from the NL and nested-SP families predict the six held-out spatial and field variants with mean absolute errors of 5.0×10^{-4} (GPG) and 8.1×10^{-4} (ED), so one calibration can screen caption configurations before another training run; at this error level, screening separates configurations whose loss gaps exceed the fit error rather than near-identical ones. Each configuration is trained once, so the bands in Figure 6d–e reflect sensitivity to the sweep settings; rankings largely hold under other judges, proposers, and budgets (Appendix Tables 3 and 6, Figures 11, 12, and 13). The calibration is fit on BAGEL; on Qwen-Image the matched control of §3.1 shows that the structured interface still outperforms a free-form one, but the quantitative relation is not re-fit; like parameter, data, and compute laws (Kaplan et al., 2020; Hoffmann et al., 2022), it is recipe-specific rather than universal (Appendix C.6).

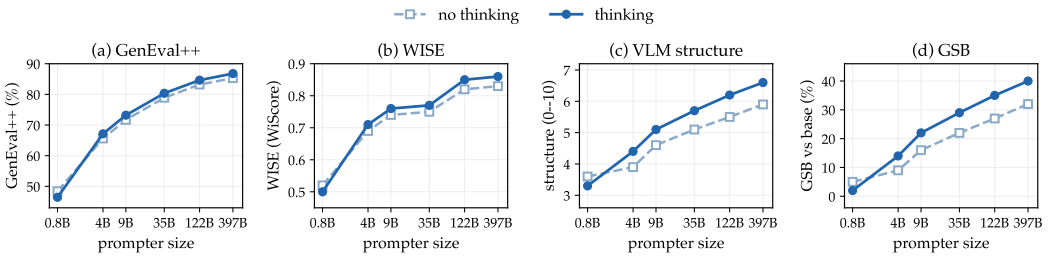


Figure 7: **Prompter scaling: image quality improves with prompter size and reasoning.** *(a–b)* GenEval++ (Ye et al., 2025) and WISE (Niu et al., 2025). *(c–d)* GPT-5.4 structure score and GSB net preference against the 0.8B prompter (rubrics in Appendices H.4 and H.5).

From the properties to two factors. When captions are annotated from paired images, the scaling properties isolate a diffusion-side capability, which we call *diffusability* after Skorokhodov et al. (2025): how effectively the caption interface exposes and organizes image-grounded supervision under a fixed training recipe. The NL/SP contrast identifies organization as a practical intervention, but it tests richer content and organization jointly rather than JSON syntax in isolation. At inference no paired image is available, so an LLM prompter must infer these variables from the user request; we call this LLM-side capability *promptability*. With f the caption interface and π the prompter, we summarize their joint role schematically (not as a fitted law):

$$\boxed{\text{Quality}(f,\pi) \;=\; \text{Diffusability}(f) \;\times\; \text{Promptability}(f,\pi)} \tag{5}$$

2.3 Raising Diffusability: constructing structured supervision

Image-to-SP annotation with domain experts. To raise diffusability at corpus scale, we annotate each training image with a full-schema (L10) SP. Because a general-purpose VLM is less reliable on body-side orientation and precise geometry, and serialized errors become explicit conditioning variables, our five-stage pipeline lets a VLM recover global and per-element semantics while frozen specialists supply geometric evidence: Sapiens (Khirodkar et al., 2024) pose keypoints, DepthAnything V2 (Yang et al., 2024a) relative depth, and SAM 2.1 (Ravi et al., 2024) masks and occlusion cues. A final VLM pass reconciles this evidence into one well-formed SP, expressing it in text fields rather than copying raw predictions (Appendix E.2, Figure 14).

Controlled field ladder. To separate schema richness from annotation quality, levels L5–L9 are derived from each L10 SP by deterministically masking field groups; ascending from L5 restores bounding boxes, scene context, dynamic attributes, depth and relations, and element-level photography (Appendix Table 8). Along this ladder, exposing more field groups raises GPG and ED and lowers loss along the relations of §2.2. The gain transfers to generated images: when a frozen Gemini 3 Pro writes one L10 SP per user prompt and each level is rendered by the BAGEL diffuser trained at that level, GenEval2 GM rises monotonically from 46.79 at L5 to 57.70 at L10, and GSB against L5 reaches 26.0%. A field-wise ablation ranks global scene context first, followed by bounding boxes (Appendix Figure 15).

2.4 Raising Promptability: scaling and training the LLM prompter

The annotation pipeline needs the paired image; at inference only a user request is available. An LLM prompter must translate it into an SP, inferring plausible visual details the user left unspecified without violating explicit constraints, so we assess promptability by the quality of the rendered images rather than by JSON validity. We train one Qwen-Image diffuser on a mixture of SP levels and NL captions, then fix it and the L10 schema and vary only the prompter π.

Off-the-shelf prompter scaling. We first test whether general-purpose LLM progress transfers without task-specific training. Six frozen Qwen3.5 checkpoints from 0.8B to 397B map user requests to L10 SPs in non-thinking and chain-of-thought modes. Generated-image quality improves with scale (Figure 7): in thinking mode, GenEval++ rises from 46.4% at 0.8B to 86.8% at 397B. Chain-of-thought helps at every scale except 0.8B, where reasoning often loops before valid JSON. Appendix Figures 21–25 compare prompters across scale, training, and general-purpose LLMs qualitatively.

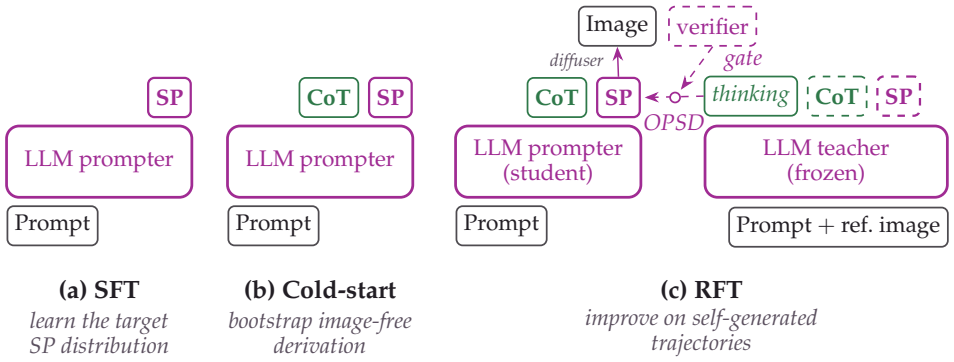


Figure 8: **The three-stage prompter-training pipeline (§2.4).** *(a)* SFT learns the target SP distribution. *(b)* Cold-start bootstraps image-free prompt → CoT → SP derivation from privileged traces. *(c)* RFT: a QA verifier gates the student's rendered rollouts and OPSD supplies targets from an image-conditioned teacher; only the student is updated.

Because none of these prompters receives task-specific training, the trend associates advances in general-purpose LLMs and reasoning mode with higher image quality through the SP interface.

Training pipeline. Even the 397B prompter writes schema-valid SPs with too little visual detail, yielding overly simple images on complex scenes (Figure 10, zero-shot column). Following staged post-training for reasoning LLMs, we train it in three stages (Figure 8). *SFT* pairs each image's original caption, as the user prompt, with its annotated SP, teaching a conditional prior over plausible completions rather than a deterministic layout. *Cold-start* distills image-free reasoning: an image-conditioned VLM writes traces from prompt to SP, and a judge rejects traces that state instance details without a prompt-grounded or common-sense rationale. *RFT* trains on the prompter's own rendered rollouts. Because VLM verdicts on stochastic renders are noisy, we use the verifier only as a conservative, high-precision acceptance gate; on accepted rollouts, on-policy self-distillation (OPSD) matches the image-free prompter's next-token distribution to that of a frozen teacher that also sees the reference image, exploiting the same privileged-image asymmetry that GPG measures (Eq. (6), Appendix E.5.2).

3 Experiments

We first compare the complete prompter–diffuser system with representative text-to-image models (§3.1). Holding the schema and diffuser fixed, we then isolate promptability, the LLM-side factor of Eq. (5), through backend and training ablations (§3.2). Finally, we test whether allocating inference-time compute to iterative refinement yields further gains (end of §3.2; full results in Appendix G.4).

Table 1: **Comparison with representative text-to-image systems.** GenEval2 reports Soft-TIFA AM/GM; TIIF reports short/long accuracy on *testmini*. Ours uses single-shot inference; Qwen-Image*: official prompt enhancer; †: our re-evaluation; baseline references and protocols in Appendix F.

Model	GenEval	GenEval2 (AM/GM)	DPG	TIIF (s/l)	WISE	CoReBench
w/o PE						
FLUX.1 Dev	0.67	67.1/21.1	83.84	71.1/71.8	0.50	42.2
OmniGen2	0.80	–	83.57	–	–	42.9
Emu3.5	0.86	–	87.46	**89.5**/88.2	0.57	–
BAGEL	0.82	–	85.07	71.5/71.7	0.52	38.2
Qwen-Image	0.87	80.8/33.8	88.32	86.1/86.8	0.62	58.9
w/ PE						
BAGEL + CoT	0.88	70.9/23.1	–	–	0.70	41.1
GPT-Image-1	0.84	–	85.15	89.2/88.3	0.80	72.6
Nano Banana	0.89	82.8/44.6	85.23	–	**0.89**	75.9
LongCat-Image	0.87	–	86.80	–	0.65	59.6
HunyuanImage 3.0	0.72	–	86.10	–	0.57	58.7
Qwen-Image*	0.91	82.4/52.8†	87.20†	88.3/88.4†	0.83	74.7†
Matched NL + Qwen-Image	0.91	84.5/56.2	87.80	88.5/88.0	0.84	76.1
Ours (Qwen-Image)	**0.94**	**90.6/72.5**	**90.71**	89.1/**89.2**	**0.89**	**85.2**



Figure 9: **Qualitative comparison on compositional prompts.** Ours (right) realizes the counts and spatial relations that other systems miss.

3.1 End-to-end performance and matched control

Comparison with existing systems. Table 1 compares the resulting system with representative open-weight and closed-source generators. Among the evaluated open-weight systems, Ours leads or ties almost every reported metric and matches or surpasses the evaluated closed systems on most. The benchmarks probe complementary capabilities: GenEval and GenEval2 measure object-centric and compositional alignment (Ghosh et al., 2023; Kamath et al., 2025); DPG-Bench and TIIF stress dense and information-intensive prompt following (Hu et al., 2024; Wei et al., 2025); WISE evaluates world knowledge (Niu et al., 2025); and CoReBench targets composition and reasoning (Li et al., 2026). Consistent gains across these settings show that the benefit is not confined to a single prompt regime or evaluator, but extends from basic alignment to knowledge-dependent and reasoning-intensive generation. Figure 9 (full version: Appendix Figure 17) and Appendix Figure 16 reflect the same breadth qualitatively: on complex prompts, the structured-prompt system more faithfully realizes spatial layouts, object counts, and attribute bindings.

Matched NL control. To verify that the gains arise from using structured prompts as the shared caption interface between the prompter and diffuser, rather than from a larger backbone or additional training alone, we compare against two controls built on the same Qwen-Image architecture. The official Qwen-Image prompt enhancer reaches 52.8 GenEval2 GM and 74.7 CoReBench, compared with 72.5 and 85.2 for our system. More stringently, matched NL retrains the same Qwen-Image diffuser and prompter on the same images, stages, and budgets while retaining a free-form caption interface. This additional training improves the two scores to 56.2 and 76.1, but remains well below the matched end-to-end SP system. Thus, retraining the same architecture and data with an NL interface does not reproduce the SP system's gains. Appendix E.2 details the matched control, and Appendix Tables 13–16 provide category-level results.

3.2 Isolating promptability

Promptability: transferring LLM progress. With the schema and Qwen-Image diffuser fixed and only the rewriting backend varied (Appendix Table 17), zero-shot SP filling by frontier LLMs raises

Table 2: **Prompter training-pipeline ablation.** GPT-5.4 judge; GSB vs. Base (N=150). GRPO: verifier reward (Shao et al., 2024); OPSD: ungated or gated.

Stage	DPG	Struct.	Align.	GSB
Base	89.42	4.860	8.307	—
+ SFT	89.61	6.273	8.473	23.3
+ Cold-start	89.84	6.753	8.360	28.7
+ RFT: GRPO	89.73	7.113	8.907	36.7
+ RFT: OPSD	89.68	6.993	8.747	33.3
+ RFT: gated OPSD	**90.71**	**7.600**	**9.047**	**42.0**



Figure 10: **Prompter training stages.** Same prompt and backbone; text, density, and layout improve stage by stage.

alignment and GSB over the same LLM's NL rewrite but lowers DPG-Bench and structure, so the SP interface opens headroom that zero-shot schema filling does not fully realize. Training closes this gap: our prompter (final row of Table 2) is the strongest single-turn backend on all four metrics, while coding agents running their native multi-turn loops reach up to 44.7% GSB, which motivates the controlled inference-time experiment below.

Promptability: training the prompter. We next study how task-specific training improves promptability by examining the contribution of each training stage and supervision signal. Table 2 holds the schema and diffuser fixed, cumulatively adds SFT and Cold-start, and then varies the RFT signal. SFT produces the largest single-stage gain in structure ($4.860 \rightarrow 6.273$) and a 23.3% GSB preference, consistent with learning plausible SP content rather than JSON syntax alone; Appendix Figure 18 illustrates the acquired layout prior. Cold-start further improves structure and GSB, although alignment decreases slightly, showing that its privileged traces do not improve every criterion uniformly. Verifier-reward GRPO and ungated OPSD both improve structure, alignment, and GSB over Cold-start; combining high-confidence rollout selection with dense image-conditioned OPSD targets gives the strongest endpoint. Across the full pipeline, DPG-Bench changes by only 1.29 points, whereas structure rises to 7.600 and GSB to 42.0%. DPG-Bench is therefore comparatively insensitive to the structural and compositional differences visible in Figure 10. Overall, the ablation supports the intended division of labor: SFT learns the target SP distribution, Cold-start teaches image-free derivation, and RFT improves the prompter on its own rollouts.

Promptability at inference time. We place the Base and trained prompters in the same refine–render–judge loop (Appendix Figures 19 and 20): an online judge critiques each render, and the prompter revises the corresponding SP fields until the judge returns PASS or $T_{\max}$ rounds elapse, with prompter weights, schema, and diffuser fixed. Both prompters improve with $T_{\max}$, mainly in structure. At $T_{\max} = 8$ the trained prompter reaches 54.7% GSB, above the 44.7% of the strongest coding agent (Claude Code), while Base after eight rounds still trails the trained prompter's single shot (24.0% vs. 42.0% GSB); agentic inference complements training rather than replacing it. The effective horizon is short: the trained prompter uses 2.31 rounds on average, and raising $T_{\max}$ from four to eight adds only 0.7 GSB points. Appendix G.4 (Appendix Table 18) reports the full results and analysis.

4 Related Work

Text-to-image generation has scaled mainly on the model side (Rombach et al., 2022; Peebles & Xie, 2023; Esser et al., 2024; Liang et al., 2024), while recaptioning (Betker et al., 2023; Chen et al., 2024b) and structured, JSON-style captions (Gutflaish et al., 2025; NVIDIA, 2026; Merchant et al., 2025) have improved prompt following, and LLM prompters rewrite requests for a fixed generator (Yang et al., 2024b; Wang et al., 2025a; Chen et al., 2025). We instead measure the image-grounded information a caption carries, calibrate it against converged diffusion loss, and use it to guide both caption design and prompter training. Appendix A discusses related work in full.

5 Conclusion

We show that caption *information content*, rather than caption length, is a measurable and scalable axis in text-to-image learning. Across controlled caption configurations, GPG and ED predict converged diffusion loss, defining empirical scaling properties for text conditioning that can be used to compare caption representations after calibration. This supports a *Diffusability*$\times$*Promptability* view of the caption interface: structured prompts raise diffusability by exposing and organizing image-grounded variables in addressable fields, while scaling and training the LLM prompter raises promptability by translating user requests into detailed, coherent instances of that representation. Matched natural-language retraining shows that the structured-representation gains are not explained by additional training alone, and the resulting end-to-end system improves visual coherence, prompt fidelity, and compositional generation without changing the diffusion architecture. At inference, short refine–render–judge loops provide further gains, but the trained prompter internalizes much of the correction and additional rounds yield rapidly diminishing returns; prompt-side inference compute is therefore a complement to training rather than a substitute for it. Appendix B discusses limitations.

AI USE STATEMENT

We used generative AI tools to edit and condense the manuscript text and to draft plotting and figure-processing scripts; the authors reviewed and revised all such output and verified every reported number against the underlying experiment logs. Separately, large language and vision–language models are components and evaluators of the method under study (the prompter, the annotation pipeline, and the judges of Sections 2 and 3); this use is part of the research itself and is documented in Appendices E.2, E.5, and H. We take full responsibility for the content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work improves text-to-image generation and inherits the risks of that technology, including the creation of misleading or harmful imagery. The annotation pipeline estimates human pose and depth from images as intermediate evidence for captioning. External models, datasets, and their licenses are listed in Appendix Table 12.

REPRODUCIBILITY STATEMENT

Appendix C specifies the GPG scoring protocol and reports all 15 sweep settings with their measured values (Appendix Table 4); Appendix D specifies ED. Diffusion and prompter training hyperparameters are in Tables 9, 10, and 11; the annotation pipeline, prompter training stages, OPSD objective, and agentic loop are specified in Appendices E.2, E.5.1, E.5.2, and E.6; and every system and judge prompt is reproduced in Appendix H. Code will be released upon acceptance.

REFERENCES

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, Wesam Manassra, Prafulla Dhariwal, Casey Chu, Yunxin Jiao, and Aditya Ramesh. Improving image generation with better captions. Technical report, OpenAI, 2023. URL https://cdn.openai.com/papers/dall-e-3.pdf.

Black Forest Labs. Announcing Black Forest Labs. https://blackforestlabs.ai/announcing-black-forest-labs/, 2024. Model release.

Hila Chefer, Yuval Alaluf, Yael Vinker, Lior Wolf, and Daniel Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-Σ: Weak-to-strong training of diffusion transformer for 4K text-to-image generation. In *European Conference on Computer Vision (ECCV)*, 2024a.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *International Conference on Learning Representations (ICLR)*, 2024b.

Ruibo Chen, Jiacheng Pan, Heng Huang, and Zhenheng Yang. Improving text-to-image generation with input-side inference-time scaling. *arXiv preprint arXiv:2510.12041*, 2025.

Sheng Cheng, Maitreya Patel, and Yezhou Yang. Precision or recall? an analysis of image captions for training text-to-image generation model. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Yufeng Cui, Honghao Chen, Haoge Deng, Xu Huang, Xinghang Li, Jirong Liu, Yang Liu, Zhuoyan Luo, Jinsheng Wang, et al. Emu3.5: Native multimodal models are world learners. *arXiv preprint arXiv:2510.26583*, 2025.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Alessandro Favero, Luca Zancato, Matthew Trager, Siddharth Choudhary, Pramuditha Perera, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Multi-modal hallucination control by visual information grounding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Google. Nano banana (gemini 2.5 flash image). https://blog.google/products/gemini/updated-image-editing-model/, 2025. Model release.

Litao Guo, Xinli Xu, Luozhou Wang, Jiantao Lin, Jinsong Zhou, Zixin Zhang, Bolan Su, and Ying-Cong Chen. ComfyMind: Toward general-purpose generation via tree-based planning and reactive feedback. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/file/40168e00bf87869c5d153e934d8a3602-Paper-Conference.pdf.

Eyal Gutflaish, Eliran Kachlon, Hezi Zisman, Tal Hacham, Nimrod Sarid, Alexander Visheratin, Saar Huberman, Gal Davidi, Guy Bukchin, Kfir Goldberg, and Ron Mokady. Generating an Image From 1,000 Words: Enhancing Text-to-Image With Structured Captions. *arXiv preprint arXiv:2511.06876*, 2025.

Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

Zefeng He, Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Yu Cheng, and Yang Yang. GEMS: Agent-native multimodal generation with memory and skills. *arXiv preprint arXiv:2603.28088*, 2026.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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. *arXiv preprint arXiv:2106.09685*, 2021.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Qirui Jiao, Daoyuan Chen, Yilun Huang, Xika Lin, Ying Shen, and Yaliang Li. DetailMaster: Can your text-to-image model handle long prompts? *arXiv preprint arXiv:2505.16915*, 2025.

Amita Kamath, Kai-Wei Chang, Ranjay Krishna, Luke Zettlemoyer, Yushi Hu, and Marjan Ghazvininejad. GenEval 2: Addressing benchmark drift in text-to-image evaluation. *arXiv preprint arXiv:2512.16853*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Rawal Khirodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. In *European Conference on Computer Vision (ECCV)*, 2024.

Ouxiang Li, Yuan Wang, Xinting Hu, Huijuan Huang, Rui Chen, Jiarong Ou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Fuli Feng. Easier painting than thinking: Can text-to-image models set the stage, but not direct the play? In *International Conference on Learning Representations*, 2026.

Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. GLIGEN: Open-set grounded text-to-image generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zhengyang Liang, Hao He, Ceyuan Yang, and Bo Dai. Scaling laws for diffusion transformers. *arXiv preprint arXiv:2410.08184*, 2024.

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 *European Conference on Computer Vision (ECCV)*, 2024.

Meituan LongCat Team, Hanghang Ma, Haoxian Tan, Jiale Huang, Junqiang Wu, Jun-Yan He, Lishuai Gao, Songlin Xiao, Xiaoming Wei, Xiaoqi Ma, Xunliang Cai, Yayong Guan, and Jie Hu. LongCat-Image technical report. *arXiv preprint arXiv:2512.07584*, 2025.

Nicholas Merchant, Haitz Sáez de Ocáriz Borde, Andrei Cristian Popescu, and Carlos Garcia Jurado Suarez. Structured captions improve prompt adherence in text-to-image models (re-laion-caption 19m). *arXiv preprint arXiv:2507.05300*, 2025.

Yuwei Niu, Munan Ning, Mengren Zheng, Weiyang Jin, Bin Lin, Peng Jin, Jiaqi Liao, Chaoran Feng, Fanqing Meng, Kunpeng Ning, Bin Zhu, and Li Yuan. WISE: A world knowledge-informed semantic evaluation for text-to-image generation. *arXiv preprint arXiv:2503.07265*, 2025.

NVIDIA. Cosmos 3: Omnimodal World Models for Physical AI. *arXiv preprint arXiv:2606.02800*, 2026. URL https://research.nvidia.com/labs/cosmos-lab/cosmos3/technical-report.pdf.

Yasumasa Onoe, Sunayana Rane, Zachary Berger, Yonatan Bitton, Jaemin Cho, Roopal Garg, Alexander Ku, Zarana Parekh, Jordi Pont-Tuset, Garrett Tanzer, Su Wang, and Jason Baldridge. DOCCI: Descriptions of connected and contrasting images. In *European Conference on Computer Vision (ECCV)*, 2024.

OpenAI. GPT-Image-1. https://openai.com/index/image-generation-api/, 2025. Model release.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

Qwen Team. Qwen3.5. https://qwen.ai/blog?id=qwen3.5, 2025. Blog post.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Reve Team. The Layout Bet, June 2026. URL https://blog.reve.com/posts/the-layout-bet/.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets & Benchmarks Track*, 2022.

Eyal Segalis, Dani Valevski, Danny Lumen, Yossi Matias, and Yaniv Leviathan. A picture is worth a thousand words: Principled recaptioning improves image generation. *arXiv preprint arXiv:2310.16656*, 2023.

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.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

Tencent Hunyuan Foundation Model Team. HunyuanImage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Jack Urbanek, Florian Bordes, Pietro Astolfi, Mary Williamson, Vasu Sharma, and Adriana Romero-Soriano. A picture is worth more than 77 text tokens: Evaluating CLIP-style models on dense captions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Linqing Wang, Ximing Xing, Yiji Cheng, Zhiyuan Zhao, Donghao Li, Tiankai Hang, Jiale Tao, Qixun Wang, Ruihuang Li, et al. PromptEnhancer: A simple approach to enhance text-to-image models via chain-of-thought prompt rewriting. *arXiv preprint arXiv:2509.04545*, 2025a.

Xinran Wang, Muxi Diao, Yuanzhi Liu, Chunyu Wang, Kongming Liang, Zhanyu Ma, and Jun Guo. Harnessing caption detailness for data-efficient text-to-image generation. *arXiv preprint arXiv:2505.15172*, 2025b.

Xinyu Wei, Jinrui Zhang, Zeqing Wang, Hongyang Wei, Zhen Guo, Bairui Li, and Lei Zhang. TIIF-Bench: How does your T2I model follow your instructions? *arXiv preprint arXiv:2506.02161*, 2025.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, et al. Qwen-Image technical report. *arXiv preprint arXiv:2508.02324*, 2025a.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, et al. OmniGen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025b.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Ling Yang, Zhaochen Yu, Chenlin Meng, Minkai Xu, Stefano Ermon, and Bin Cui. Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal LLMs. In *International Conference on Machine Learning (ICML)*, 2024b.

Zhichao Yang, Tianjiao Gu, Jianjie Wang, Feiyu Lin, et al. LongT2IBench: A benchmark for evaluating long text-to-image generation with graph-structured annotations. In *AAAI Conference on Artificial Intelligence*, 2026.

Junyan Ye, Dongzhi Jiang, Zihao Wang, Leqi Zhu, Zhenghao Hu, Zilong Huang, Jun He, Zhiyuan Yan, Jinghua Yu, Hongsheng Li, Conghui He, and Weijia Li. Echo-4o: Harnessing the power of GPT-4o synthetic images for improved image generation. *arXiv preprint arXiv:2508.09987*, 2025.

Beichen Zhang, Pan Zhang, Xiaoyi Dong, Yuhang Zang, and Jiaqi Wang. Long-CLIP: Unlocking the long-text capability of CLIP. In *European Conference on Computer Vision (ECCV)*, 2024.

Wendi Zheng, Jiayan Teng, Zhuoyi Yang, Weihan Wang, Jidong Chen, Xiaotao Gu, Yuxiao Dong, Ming Ding, and Jie Tang. CogView3: Finer and faster text-to-image generation via relay diffusion. In *European Conference on Computer Vision (ECCV)*, 2024.

Appendix

CONTENTS

A Extended related work **16**

B Limitations **17**

C Grounded Perplexity Gain: measurement details **17**

C.1 Default scoring protocol 17

C.2 GPG robustness across judges 18

C.3 Full 15-setting data 18

C.4 Monotonicity analysis 19

C.5 Fit sensitivity across settings and training budgets 19

C.6 Mutual-information motivation and the empirical status of the relations 20

D Effective Detailness: validation details **21**

D.1 Image-grounded source extraction 21

D.2 Why precision-weighted $F_{0.5}$? 21

D.3 Shared-UID resampling sensitivity 21

D.4 Robustness across source extractors 21

E Structured-prompt schema and implementation details **22**

E.1 Structured-prompt schema 22

E.2 Annotation pipeline 23

E.3 Diffusion-backbone training 25

E.4 Schema field ablation 26

E.5 Prompter training 26

E.6 Inference and the agentic loop 28

E.7 External dependencies and licenses 29

F Additional benchmark results **31**

F.1 DPG-Bench per-category breakdown 31

F.2 GenEval per-skill breakdown 32

F.3 WISE per-category breakdown 33

F.4 T2I-CoReBench per-category breakdown 34

G Supplementary results for Section 3 **34**

G.1 Qualitative results 34

G.2 Rewriting-backend comparison 34

G.3 SFT layout prior 37

G.4 Agentic refine–render–judge loop 37

H System prompts **39**

H.1 Cold-start teacher system prompt 39
H.2 Aesthetic judge system prompt 41
H.3 Cold-start filtering system prompts 41
H.4 VLM-as-judge system prompts 46
H.5 GSB pairwise preference protocol 48

I Prompts for figures **49**

I.1 Figure 16 — Gallery 49
I.2 Figure 16 (bottom) — Zero-shot SP editing 50
I.3 Figure 2 — Teaser 50
I.4 Figure 9 — SOTA qualitative comparison 50
I.5 Figure 10 — Prompter training-stage progression 50
I.6 Figure 18 — SFT vs. no-SFT 51
I.7 Figure 20 — Case types the agentic loop resolves 51
I.8 Figure 21 — prompter comparison prompts 51
I.9 Figures 24–25 — SOTA LLM as prompter 52

J Additional qualitative examples **53**

J.1 Prompter comparison: prompter scale × training 53
J.2 General-purpose LLMs as prompters 53

A Extended related work

Text-to-image scaling, long captions, and structured control. Diffusion models (Ho et al., 2020; Song et al., 2021), latent diffusion (Rombach et al., 2022), and transformer backbones (Peebles & Xie, 2023; Esser et al., 2024) have advanced T2I generation largely through model-side scaling. Conditioning has also mattered: Imagen (Saharia et al., 2022) found that a larger text encoder improves fidelity, while LLM conditioners (Hu et al., 2024), long-context encoders (Zhang et al., 2024), and dense-caption corpora (Urbanek et al., 2024; Onoe et al., 2024) extend the information available in a prompt. Existing scaling laws establish a compute–loss relation for DiTs (Liang et al., 2024); we instead hold model, data, and compute fixed and show that converged diffusion loss is predicted by caption information, measured by white-box GPG or black-box ED, rather than length. This distinction is consistent with long-prompt benchmarks: DetailMaster (Jiao et al., 2025) bins prompts by token count and reports a consistent negative correlation between prompt length and accuracy on character attributes, character locations, and entity relationships, while LongT2IBench (Yang et al., 2026) reports that graph-structured alignment over entities, attributes, and relations decreases steadily across word-count intervals. TIIF-Bench (Wei et al., 2025) instead pairs each prompt with a semantically equivalent long version and finds that robustness to this change tracks overall instruction-following ability, with the strongest models remaining stable across both settings. Our measurements suggest that this behavior reflects an information plateau rather than length alone. A separate line of work intervenes on the diffuser rather than on the caption. Layout-conditioned generation (e.g., GLIGEN (Li et al., 2023)) and attention-manipulation methods (e.g., Attend-and-Excite (Chefer et al., 2023)) steer a fixed diffuser through auxiliary spatial conditions or model-internal intervention; our schema instead places boxes, depth, and relations in ordinary text

fields, scored on the same GPG/ED axes as free-form captions and aimed at caption information rather than layout control.

Recaptioning and caption quality. DALL-E 3 (Betker et al., 2023) showed that detailed synthetic recaptioning can markedly improve prompt following; PixArt-α (Chen et al., 2024b), PixArt-Σ (Chen et al., 2024a), CogView3 (Zheng et al., 2024), and RECAP (Segalis et al., 2023) likewise demonstrate the value of richer or more principled captions. Recent systems go beyond free-form prose. FIBO (Gutflaish et al., 2025) trains on long JSON captions, compares them with short captions under matched training, and learns a VLM translator from short requests to its schema. Cosmos 3 (NVIDIA, 2026) uses structured JSON annotations and a prompt upsampler, showing that predefined fields improve annotation recall over dense prose. Reve 2.0 (Reve Team, 2026) uses a hierarchical layout intermediary and reports gains over text-only generation as well as improved reconstruction with more regions; concurrent work by Merchant et al. (2025) studies a fixed four-part caption template. These results establish structured representation as a useful design choice. Our focus is complementary: we measure image-grounded information across NL lengths, nested SP levels, spatial serializations, and field ablations, then calibrate that common variable against matched-budget converged diffusion loss. We further isolate prompt production by varying prompter scale, reasoning, and training while holding the schema and trained diffuser fixed. GPG adapts the per-token grounding signal of Favero et al. (2024) from decoding-time hallucination localization to corpus-level caption informativeness; ED adapts caption-detailness evaluation (Wang et al., 2025b) to black-box, image-grounded caption scoring; our precision-weighted matcher is motivated by Cheng et al. (2024), who find caption precision to matter more than recall when training text-to-image models.

LLM prompters, RFT, and inference-time methods. LLMs have become effective inference-time prompt enhancers: RPG (Yang et al., 2024b) decomposes prompts into regional sub-prompts with multimodal-LLM reasoning; PromptEnhancer (Wang et al., 2025a) applies RFT to a chain-of-thought rewriter with a dedicated reward model; and input-side inference-time scaling (Chen et al., 2025) trains a rewriter with iterative DPO. These methods improve prompts for a fixed generator. Our structured format additionally raises the caption representation's training-time *diffusability*; with that schema and backbone fixed, we study prompter *promptability* under LLM scale, chain-of-thought, verifier-gated self-distillation from an image-conditioned teacher, and an agentic refine–render–judge loop. Reward models such as ImageReward (Xu et al., 2023) and model-side preference optimization such as Diffusion-DPO (Wallace et al., 2024) optimize the generator, whereas our verifier filters prompter rollouts while RFT holds the already SP-trained diffuser fixed. Finally, CLIPScore (Hessel et al., 2021) and VQAScore (Lin et al., 2024) measure image–text alignment; GPG measures caption informativeness relative to its paired image, while ED uses a one-time image-grounded annotation pass followed by text-only scoring.

B Limitations

Both metrics depend on image-conditioned measurement: GPG queries a vision–language judge at scoring time, while ED requires a one-time offline extraction of attribute tuples from each image. The fitted relations may shift under different judges or extractors. Agreement between the two measures reduces, but does not eliminate, the risk that the shared trend reflects a particular scoring interface. The schema is hand-designed; automatic schema discovery and extension to video and three-dimensional generation remain open. The structured-prompt path also adds prompter latency, so deployment must weigh the generation gains against this additional inference cost.

C Grounded Perplexity Gain: measurement details

C.1 Default scoring protocol

All headline GPG scores use Qwen3.5-397B-A17B frozen as the VLM judge. Images are resized so that the longer side is 1024 pixels with aspect ratio preserved. Before either pass, captions are mapped to a deterministic scored sequence. For SPs, canonicalization removes the global metadata keys `atmosphere`, `lighting`, `style`, and `photography`, including its `layout`, `shot_type`, `camera_angle`, and `lens_and_effect` subfields. NL captions retain their text,

with template-only spans marked separately. The image-conditioned and no-image passes then use the same conversation template and canonical sequence; the latter replaces the image with an empty image slot. All tokens in that sequence remain in the autoregressive context in both passes. The content mask excludes JSON syntax, stylistic boilerplate, and NL template-only positions only from the accumulated GPG sum. The judge, preprocessing, and template are fixed across all caption conditions, so comparisons vary only the caption content presented for scoring. The canonicalization and content-mask recipe was finalized during metric development on the controlled BAGEL sweep and then frozen. Consequently, the headline GPG–loss relation is a calibration for this fixed recipe, not an independent validation of these scoring choices; applying it to new caption families requires retaining the same recipe or recalibrating it once.

C.2 GPG ROBUSTNESS ACROSS JUDGES

GPG depends on the complete judge interface—the VLM, tokenizer, conversation template, image processor, and coordinate convention—rather than on the likelihood query alone; we hold that interface fixed for every headline comparison. Our default judge is Qwen3.5-397B-A17B, chosen because its normalized bbox convention is directly compatible with the caption format's 0–999 coordinates, avoiding an additional coordinate conversion. To verify that the linear scaling property (Eq. (3)) is not specific to this judge, we recompute GPG on a held-out pool of 1,000 paired images across the scaling-property cells under six additional VL judges—spanning the Qwen2-VL / Qwen2.5-VL / Qwen3-VL / Qwen3.5-MoE lineage plus a cross-family judge (InternVL3-8B)—and refit the linear relation on each. Table 3 reports the resulting cross-judge comparison. Every judge that grounds the caption's spatial vocabulary gives a strong negative fit (r from -0.91 to -0.99, within ± 0.08 of the main fit's -0.984), and their per-cell GPG rankings agree at Spearman $\rho \geq 0.86$ (identical top and bottom cells). Across the fully evaluated Qwen-family judges, R^2 remains between 0.93 and 0.97; the 122B judge reaches 0.99 on the six available cells, but this partial-cell fit is not directly comparable to the full-cell fits (Figure 11). The robustness conclusion is therefore agreement across judges supplied with a compatible coordinate representation, not a monotonic law in judge size; we use the largest available judge for the headline measurement and compare its ordering with smaller and cross-family alternatives. The lone exception is instructive rather than a counterexample: without a bbox-format adapter, Qwen2.5-VL-7B cannot parse our 0–999 `<bbox>` tokens and *inverts* the ordering ($r = +0.23$); supplying the adapter (`-qwen-native-bbox`, which maps to its native pixel-space coordinates) restores $r = -0.97$. GPG therefore remains judge-dependent, but its configuration ordering is stable across the tested judges once each can read the caption's spatial representation.

Table 3: **GPG fit quality across alternative judges.** We refit the GPG $\rightarrow$ MSE relation under each judge on a held-out 1,000-image robustness pool; the selected main judge (bold) uses the full 30k-UID pool. Qwen2.5-VL-7B reports raw / bbox-adapted scores; see text for the inverted raw case.

Judge model	Size	Pearson r	R^2	Notes
Robustness sweep (held-out 1,000-image pool)				
Qwen2-VL-7B	7B	-0.96	0.93	Oldest Qwen-VL; narrower GPG range
Qwen3-VL-8B	8B	-0.99	0.97	Normalized bbox convention compatible with caption format
Qwen3.5-35B-A3B	35B	-0.98	0.97	`no_bbox` cell is 3.3 nats higher
Qwen3.5-122B-A10B	122B	-0.996	0.99	Partial fit (6 of 15 cells); not directly comparable
InternVL3-8B	8B	-0.91	0.83	Cross-family; confirms not Qwen-specific
Qwen2.5-VL-7B	7B	$+0.23$ / -0.97	0.05 / 0.94	Raw / with `-qwen-native-bbox` (see text)
Qwen3.5-397B-A17B	**397B**	$\mathbf{-0.98}$	**0.97**	Selected (main); full 30k-UID pool

C.3 FULL 15-SETTING DATA

Table 4 reports the complete per-setting GPG, ED, and converged diffusion loss underlying Eqs. (3) and (4). All settings share the same 30,000 paired image UIDs for both caption-side measurements. GPG uses Qwen3.5-397B-A17B with the content-mask and canonicalize-JSON recipe of

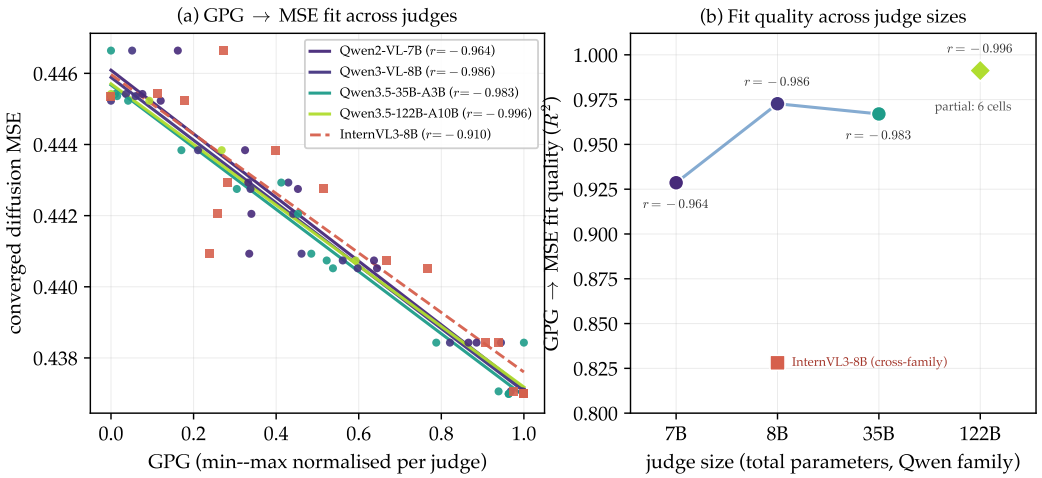


Figure 11: **GPG is robust across compatible judges.** *(a)* GPG → MSE fits with GPG min–max normalized per judge. *(b)* Fit quality versus judge size; the 122B point uses only six cells and is shown separately from the full-cell fits. InternVL3-8B is the cross-family check.

Appendix C.1, while ED follows the extraction and matching protocol of Appendix D; MSE is measured at the unified budget of 2.84×10^{10} cumulative image tokens reached by every run.

Table 4: **Full 15-setting scaling-property data**, sorted by GPG. The full-schema baseline of the ablation suite coincides with Structured L10 and is counted once.

Kind	Level	GPG	ED	MSE
Dense (NL)	L6	106.3	0.759	0.44523
Dense (NL)	L8	110.1	0.751	0.44542
Structured	L5	111.6	0.749	0.44664
Dense (NL)	L10	112.5	0.754	0.44536
Structured	L6	128.3	0.759	0.44384
Structured	L7	141.5	0.772	0.44275
Spatial (coarse, 3×3)	L10	151.0	0.778	0.44293
Spatial (fine, 5×5)	L10	152.2	0.785	0.44205
Spatial (finer, 9×9)	L10	154.4	0.787	0.44093
Structured	L8	164.8	0.793	0.44074
Abl: −scene	L10	168.5	0.799	0.44052
Structured	L9	191.8	0.819	0.43843
Abl: −bbox	L10	204.7	0.807	0.43843
Abl: −relationships	L10	207.2	0.808	0.43706
Structured	L10	210.5	0.833	0.43699

C.4 Monotonicity analysis

Sorted by GPG, MSE decreases overall across the 15 settings, with a few small local reversals. Their total magnitude is approximately 0.0016 MSE units, of which approximately 0.0012 comes from structured L5 relative to the nearby NL settings; we report this reversal descriptively rather than assigning it to a specific cause. The remaining violations are $|\Delta\text{MSE}| \leq 2\times10^{-4}$, the same order as trailing-window read-out variation, so we do not assign them to a specific cause.

C.5 Fit sensitivity across settings and training budgets

Figure 12 shows how the fit correlations change when the 15 designed sweep settings are resampled; the per-level training dynamics for both caption families appear in the main text (Figure 6a–b).

Uncertainty scope. Each setting is trained once. The analyses below quantify three operational sensitivities: resampling the deliberately designed sweep settings, refitting the same training trajectories at matched budget cuts, and measuring local variation within their trailing windows. They

are not population confidence intervals, independent training replications, or estimates of seed-level optimization uncertainty.

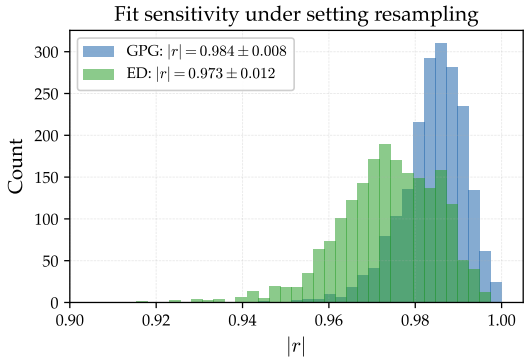


Figure 12: **Sensitivity of the scaling-property fit to resampling sweep settings.** Over $B = 2{,}000$ resamples of the 15 designed settings, the distributions of fit magnitude have mean±SD $|r| = 0.984 \pm 0.008$ for GPG and $|r| = 0.973 \pm 0.012$ for ED.

Read-out conventions for the budget refits. The budget refits of Figure 13 use the twelve settings whose full training curves were exported: six structured levels, three natural-language levels, and the three field ablations with measured GPG and ED; the three spatial variants were logged only at the final budget. The read-out at each cut is the mean training MSE over the trailing 5% of tokens before the cut; the two runs that restarted mid-training (L5, L9) have their token axes corrected for the restart's token-counter reset. GPG and ED are caption-side quantities and do not vary with budget; at each cut the GPG relation is fit linearly and the ED relation in log–log space, mirroring Eqs. (3) and (4). The curve-based read-out differs slightly from the per-setting converged values behind the headline fits, and the final-cut coefficients remain close to them.

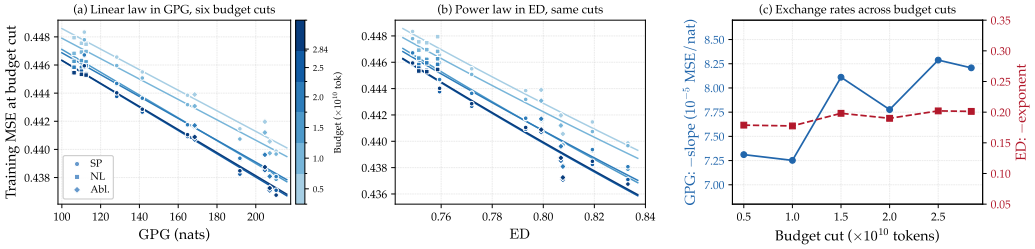


Figure 13: **The scaling properties persist across training budgets.** Refitting the same trajectories at six matched budget cuts preserves both the linear GPG relation and the ED power law; the fitted slope and exponent steepen mildly as training proceeds.

Within-trajectory read-out variation. Splitting each observed trajectory's trailing window into five disjoint blocks, the standard deviation of block means spans 1.0–5.0×10^{-4} across the twelve settings (median 3.5×10^{-4}). This scale is comparable to the residuals of both fits: $\sigma_{\text{resid}} \approx 6 \times 10^{-4}$ for GPG and 7.8×10^{-4} for ED. These values contextualize the residual scale but do not decompose it into lack-of-fit and run-level optimization variability.

C.6 Mutual-information motivation and the empirical status of the relations

For image and caption random variables I and Y, the identity $H(I \mid Y) = H(I) - I(I;Y)$ states that caption–image mutual information reduces uncertainty about the image. This intuition motivates asking whether captions with more image-grounded information are associated with a lower conditional training objective. It does not, however, derive the form of that association in our experiments. GPG is an operational, judge-dependent estimate rather than the dataset mutual information; the converged flow-matching velocity MSE is not measured in nats; and the entropy identity alone neither makes that objective a conditional-likelihood bound nor implies a linear GPG–MSE relation.

Equation (3) is therefore an empirical, recipe-specific calibration rather than a theorem. Its fitted slope has no universal information-theoretic interpretation: it depends on the judge, caption construction, model, objective, optimization, and training budget held fixed in the sweep. The empirical result is instead that GPG measured by a frozen VLM tracks the converged loss of a separately trained diffuser across the tested caption settings. ED provides a complementary check: despite using no token probabilities, it closely agrees with the GPG ordering ($\rho_{\text{Spearman}} = 0.96$) and follows its own negative power-law trend (Eq. (4)). Residuals from either fit should likewise be read as deviations from this calibration, not as direct estimates of information that the diffuser fails to exploit.

D Effective Detailness: validation details

D.1 Image-grounded source extraction

The image-grounded source set for ED is obtained by one offline Gemini 3 Pro call per image, prompted to enumerate object–attribute–relationship–grounding (OARG) tuples. The extractor sees only the image and is asked for 80–150 atomic tuples; it returns a mean of 102 tuples over the shared 30,000-image ED pool. For each caption condition, GPT-5.4 independently extracts caption-side OARG tuples, and a separate GPT-5.4 call performs symmetric paraphrase-tolerant matching against the cached source tuples. For each valid image–caption pair, P_A is the fraction of caption-side attribute tuples supported by the image source, and R_A is the fraction of image-side attribute tuples covered by the caption. The pair-level score is $F_{0.5}(P_A, R_A)$; pairs for which either side contains no attribute tuple are skipped, and the degenerate $P_A = R_A = 0$ case is assigned zero. A caption configuration's ED is the two-sided 10% trimmed mean of its pair-level scores: after sorting, we drop $\lfloor 0.1n \rfloor$ values from each tail and average the remainder. Only attributes enter the reported score; object, relation, and grounding tuples provide entity context for extraction and matching. All image-side sources, caption-side tuples, and match masks are computed once and cached.

D.2 Why precision-weighted $F_{0.5}$?

The two ED errors have different consequences for conditioning: an unsupported caption attribute supplies contradictory supervision for its paired image, whereas an omitted attribute reduces conditioning bandwidth without introducing a false visual fact. This asymmetry motivates the standard precision-oriented $F_{0.5}$ convention (van Rijsbergen, 1979; Cheng et al., 2024). We selected the precision-oriented aggregation over F_1 and $P \cdot R$ during metric development on the controlled BAGEL sweep, then froze $\beta = 0.5$ for every caption family and subsequent analysis. As with the GPG scoring recipe of Appendix C.1, the headline ED–loss relation is therefore a calibration for this fixed choice rather than an independent validation of it. The sensitivity study below varies β post hoc to test how strongly the empirical relation depends on this precision-oriented choice.

The fitted log–log slope remains negative as β increases, and the nested SP ladder remains monotonic from L5 to L10 under all three settings. Predictive strength nevertheless falls substantially: R^2 decreases from 0.943 for $F_{0.5}$ to 0.313 for F_2, while variant MAE more than doubles. Thus the direction of the relation is robust to β, but the precision-oriented score is markedly more predictive.

D.3 Shared-UID resampling sensitivity

Conditional resampling analysis. A shared-UID bootstrap over the 30,000 paired images ($B =$ 10,000) gives a resampling-distribution mean±SD Spearman $\rho = -0.979 \pm 0.010$ between ED and converged MSE, with central 95% range $[-0.993, -0.957]$. The corresponding ranges are $[-0.978, -0.946]$ for the log–log Pearson correlation and $[-0.220, -0.191]$ for the exponent. These ranges condition on the cached outputs of the image proposer, caption extractor, and matcher; they measure sensitivity to which paired UIDs are included, but do not include API/model stochasticity or training-run uncertainty.

D.4 Robustness across source extractors

ED's only image-conditioned step is the offline extraction of attribute tuples from each image. A natural concern is that the resulting per-cell rankings might be an artifact of the specific extractor

Table 5: **Sensitivity to the precision–recall weighting in ED** on 30,000 paired images per caption configuration. We replace $F_{0.5}$ by F_1 or F_2 while keeping the tuple extraction, matching, aggregation, caption configurations, and converged-loss read-out fixed. r and R^2 are measured in log–log space; variant MAE fits the three NL and six nested-SP configurations and evaluates the six spatial and field variants excluded from the fit.

β	Emphasis	exponent b	Pearson r	R^2	Spearman ρ	variant MAE
0.5	precision	−0.207	−0.971	0.943	−0.993	0.00081
1.0	balanced	−0.098	−0.736	0.541	−0.729	0.00153
2.0	recall	−0.050	−0.559	0.313	−0.607	0.00191

model. To test this, we re-run the image-side source extraction with two additional vision–language backends—GPT-4o and GPT-5.5 (reasoning)—alongside the default Gemini 3 Pro, each given the same exhaustive-OARG instructions, on a held-out pool of 1,000 paired images (GPT-5.5 on the first 150). Caption extraction and symmetric matching remain fixed at GPT-5.4, so Table 6 isolates the image-source backend. ED proves *rank*-robust: under every backend ED remains a strong negative predictor of converged MSE ($\rho(\mathrm{ED}, \mathrm{MSE})$ from -0.86 to -0.90), the cells keep the same ordering (cross-backend cell-rank $\rho = 0.80$–0.92; pairwise per-tuple Cohen $\kappa = 0.75$–0.77), and the power-law slope keeps its sign and rough magnitude. Absolute ED levels, however, are *not* backend-invariant: the intraclass correlation for absolute agreement is only 0.11 (consistency ICC 0.64)—different extractors place the cells on shifted ED scales even while agreeing on their order. We therefore use ED as a *relative* ruler with a single fixed extractor (Gemini 3 Pro) throughout the paper, and compare only rankings and slopes across backends—never absolute ED between extractors.

Table 6: **ED backend robustness** (held-out 1,000-image pool; GPT-5.5 on $N = 150$). ED remains a negative predictor of converged MSE under every extractor, with stable cell rankings but shifted absolute scales. Cross-backend rank and tuple-agreement statistics are discussed in the text.

Extractor	$\rho(\mathrm{ED}, \mathrm{MSE})$	power-law slope b	cell-rank ρ vs Gemini
Gemini 3 Pro (reference)	−0.87	-0.221 ± 0.020	—
GPT-4o	−0.86	-0.244 ± 0.027	0.80
GPT-5.5 (reasoning)	−0.90	-0.137 ± 0.042	0.88

Complete per-setting ED values appear alongside GPG and converged MSE in Table 4. They underlie the power-law fit of Eq. (4); Grid-k denotes verbal locations on a $k \times k$ grid, and the ablation rows mask the named L10 field. The verbal spatial variants follow the same relations, showing that the result is not tied to the default coordinate serialization.

E Structured-prompt schema and implementation details

The structured prompt is the central interface of this paper: its definition in §2.1, its construction in §2.3, and the scaling-property evidence in §2.2 depend on both the JSON layout and how its fields are populated from images. This appendix first summarizes the schema (§E.1), then reports the annotation, diffusion, prompter, inference, and dependency details needed to implement the full system. Training covers three model components: the BAGEL and Qwen-Image diffusion backbones, and a rank-128 LoRA (Hu et al., 2021) on Qwen3.5-397B-A17B as the prompter (§2.4, three serial stages: SFT, Cold-start, and RFT). BAGEL is trained once per caption configuration for the scaling-property experiments of §2.2; Qwen-Image is trained once on the structured-prompt corpus (a mixture of SP levels with NL captions) and held fixed across all prompter ablations (§3.2). All trainings use packed sequences, so there is no fixed per-step batch size in samples; the tables below report the packed sequence length and global GPU count from which effective tokens per step can be derived. Stage-by-stage annotation details appear in §E.2, and the degradation levels used in the scaling-property experiments are defined in Table 8.

E.1 Structured-prompt schema

Table 7 summarizes the structured-prompt schema. Three required fields define the scene skeleton—the overall `intent`, the macro `scene`, and the list of `elements`. Five optional fields add cross-element relationships (`relationships`) and global controls (`atmosphere`, `photography`,

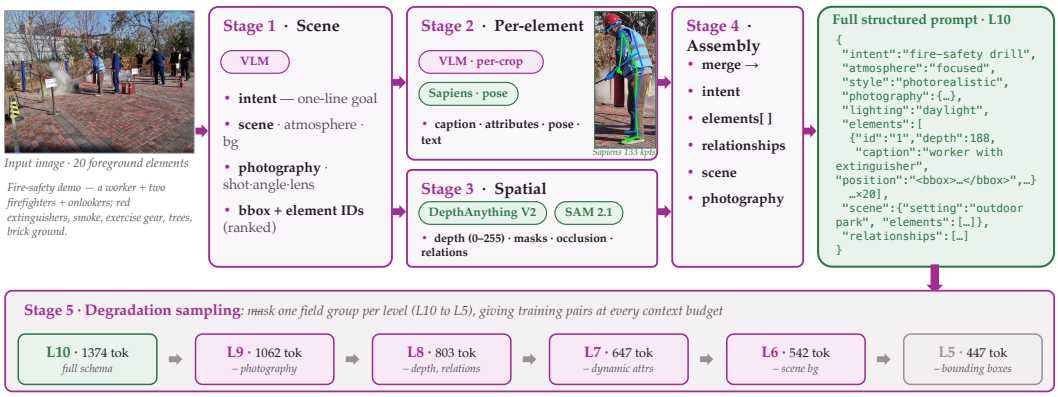


Figure 14: **Image-to-SP annotation with domain experts.** A VLM recovers global and element-level semantics; Sapiens, DepthAnything V2, and SAM 2.1 provide complementary pose and geometric evidence; a final VLM pass assembles the full L10 SP, from which L5–L9 are derived by deterministically masking field groups.

`style`, `lighting`). The prompter additionally emits a leading `ratio` control field. At inference, the generation wrapper uses it to select the output canvas and removes it before the remaining SP is passed to the diffuser. Image-to-SP annotation omits `ratio`, since the source image already fixes the canvas. The schema is extensible: the `elements` list uses dynamic per-element keys, so attributes and actions can be added without schema changes.

Table 7: **Structured-prompt schema summary.** Required fields define the scene skeleton; optional fields add cross-element relations and global controls. *`ratio` is an inference-only control emitted by the prompter, consumed by the generation wrapper, and removed before diffusion conditioning.

Field	Req.	Type	Description
`ratio`	req.*	string	Output aspect ratio from a fixed menu (e.g. `16:9`); consumed by the inference wrapper and not passed to the diffuser.
`intent`	req.	string	One high-level sentence describing the whole scene and its key entities/interactions; when it conflicts with a specific field, the specific field takes precedence.
`scene`	req.	object	Macro environment: `setting` (indoor/outdoor, time, weather) and a background `elements` list with the same structure as foreground elements.
`elements`	req.	array	All independently-editable entities. Each has `id`, a concise `caption` (identity plus the single most salient feature), `position` (language / point / bbox, with bbox coordinates normalized to 0–999 per axis), optional `depth` (0 nearest, 255 farthest), an optional per-element `photography` object, and one named key per visual dimension: attribute keys (`material_and_surface`, `color`, `lighting_interaction`, …) and action keys (`pose`, `gesture`, `expression`, `gaze`, `action`). Raw human keypoints are annotation evidence, not a schema field.
`relationships`	opt.	array	Textual statements of interaction or spatial relation between entities, each referencing their bounding boxes.
`atmosphere`	opt.	string	Overall mood, in a few words.
`photography`	opt.	object	Global camera: `layout`, `shot_type`, `camera_angle`, `lens_and_effect`.
`style`	opt.	string	Overall artistic style, e.g. "photorealistic".
`lighting`	opt.	string	Overall lighting environment and its interaction with the scene.

E.2 Annotation pipeline

Figure 14 traces the five stages that populate the L10 schema, while Table 7 defines the resulting fields.

Every model in the annotation pipeline runs frozen at our inference settings.

Table 8: **SP field ladder and downstream transfer.** L5–L9 are deterministic projections of L10. All levels are rendered by the BAGEL diffuser trained at that level, so these values are not comparable with the Qwen-Image systems of Table 1. Tokens are averages per SP; GSB is measured against L5 over $N = 150$ prompt pairs.

Level	Avg. tokens	Added fields	GenEval2 GM ↑	GSB vs. L5 ↑
L5	447	Base fields	46.79	—
L6	542	Bounding boxes	48.65	8.0
L7	647	Scene context	49.72	17.3
L8	803	Dynamic attributes	52.94	22.7
L9	1062	Depth & relationships	55.16	24.7
L10	**1374**	photography	**57.70**	**26.0**

(1) Global scene understanding. A proprietary VLM (anonymized for review) reads the whole image and emits the high-level `intent`, the global `atmosphere`, `style`, `lighting`, and `photography` fields, and the `scene` block (setting and background elements); it also lays out the JSON skeleton—splitting foreground from background and assigning each entity an `id` ordered by compositional importance, together with its bounding box.

(2) Per-element description. Each entity is cropped and re-captioned by the VLM into a concise `caption` plus one named key per visual dimension: attribute keys (color, material, surface), action keys (pose, gesture, expression, gaze), and per-element photography. For each person, Sapiens (Khirodkar et al., 2024) predicts 133 keypoints that are rendered as a pose overlay for a downstream VLM pass; the overlay helps resolve body-side orientation and joint geometry, but the raw keypoints are not written into the SP; raw keypoints and masks remain intermediate evidence.

(3) Spatial annotation. DepthAnything V2 (Yang et al., 2024a) supplies relative element depth, quantized from 0 (nearest) to 255 (farthest) when available; otherwise the `depth` field is omitted. SAM 2.1 (Ravi et al., 2024) supplies per-element masks and occlusion cues. Bounding-box, mask, and depth evidence supports geometric relations such as overlap, containment, relative position, and depth order; semantic relations such as support and interaction are inferred during the final VLM reconciliation pass.

(4) Assembly. A second VLM pass merges the stage-1–3 outputs into a well-formed L10 structured record, constrained by the schema skeleton from stage 1.

(5) Degradation sampling. Table 8 defines the deterministic field-group masks used to derive the L9→L5 variants. One set of degradations is generated per L10 annotation and reused across all training cells.

Matched NL controls. To isolate the structured interface from additional training, the end-to-end NL control uses the same training images, Qwen-Image initialization, diffusion recipe and budget, and matched prompter-training budget as the SP system, while using free-form NL captions throughout. Its prompter uses the same Qwen3.5-397B-A17B base checkpoint, rank-128 LoRA, SFT–Cold-start–RFT sequence, stage-wise data volumes, optimizer schedules, training steps, decoding settings, verifier and privileged-teacher configurations, and checkpoint-selection rule as the SP prompter. Only the intermediate caption representation and corresponding training targets change from structured prompts to free-form NL captions. The NL controls are generated from the same stage-1–3 evidence bundle used to assemble the SPs, not by flattening or compressing the final JSON. For each image, the NL verbalizer receives the global and crop-level VLM descriptions together with the same rendered Sapiens pose overlay, DepthAnything relative-depth evidence, and SAM segmentation and occlusion cues used by the SP pipeline, including intermediate evidence that is not serialized as a separate SP field. It is instructed to preserve the source entity inventory and image-specific facts while expressing them as free-form prose at the target token budget. Across the NL budgets, additional length is introduced through elaboration and connective phrasing rather than access to new annotation evidence. Appendix I.3 gives the exact teaser construction. Aggregate results appear in Table 1, with category-level breakdowns in Appendix F.

E.3 DIFFUSION-BACKBONE TRAINING

Backbones. We train two backbones: BAGEL (Deng et al., 2025) for the scaling-property fit of §2.2 (one checkpoint per cell) and Qwen-Image-2512 (Wu et al., 2025a) for the promptability sweeps and main results (§2.4, 3.1). BAGEL is a 7B unified model—Qwen2.5-7B with Mixture-of-Transformer-Experts (MoT) layers separating understanding and generation experts, a SigLIP2 vision encoder, and a frozen Flux KL-VAE—initialized from a continued-training (CT) checkpoint that extends the public BAGEL release; Qwen-Image is initialized from the Qwen-Image-2512 public release (a 60-layer DiT with a frozen Qwen2.5-VL-7B text encoder and KL-VAE). Full hyperparameters are in Tables 9 and 10.

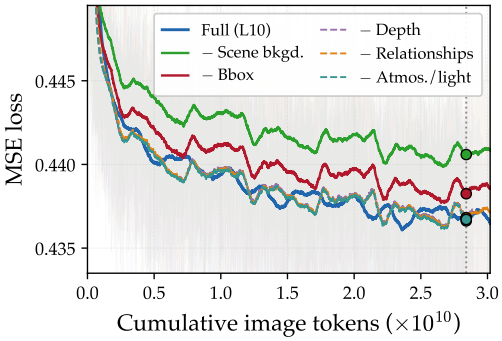
Table 9: **BAGEL training setup.** One BAGEL checkpoint is trained per caption condition in the scaling-property sweep (§2.2); the same recipe is used for every cell.

Hyperparameter	Value
Initial checkpoint	BAGEL CT checkpoint (Qwen2.5-7B MoT + SigLIP2 + Flux VAE)
Sequence length (packed)	32,768
Learning rate	5×10^{-5}
LR schedule	linear warmup (1k steps), then constant
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-15}$, weight decay 0
Max gradient norm	1.0
EMA decay	0.999
Diffusion objective	rectified flow (v-prediction); timestep shift 4.0
Precision	bf16 compute, fp32 optimizer moments
GPUs (FSDP HYBRID_SHARD)	192
Effective tokens/step	6.3M (32,768 × 192)

Table 10: **Qwen-Image training setup.** Initialized from the Qwen-Image-2512 public release (Wu et al., 2025a), trained once on the SP/NL caption corpus, then held fixed across prompter ablations (§3.2) and main results (§3.1).

Hyperparameter	Value
Initial checkpoint	Qwen-Image-2512 public release (Wu et al., 2025a) (60-layer DiT; frozen Qwen2.5-VL-7B text encoder + KL-VAE)
Sequence length (packed)	32,768
Learning rate	1×10^{-4}
LR schedule	linear warmup (2k steps), then constant
Optimizer	AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-15}$, weight decay 0
Max gradient norm	1.0
EMA decay	0.9999
Diffusion objective	flow matching (v-prediction); resolution-dependent timestep shift
Precision	bf16 compute, fp32 optimizer moments
GPUs (FSDP HYBRID_SHARD)	512
Effective tokens/step	16.8M (32,768 × 512)

Tokenizer and sequence packing. BAGEL tokenizes text with the Qwen2.5-7B BPE tokenizer and Qwen-Image with the Qwen2.5-VL tokenizer (152,064-token vocabulary); for diffusion training, each structured record is deterministically serialized in a compact single-quote form (' delimiters, minimal separators) to save tokens. Both encode images with a frozen KL-VAE at $8\times$ spatial compression and 16 latent channels—the Flux VAE for BAGEL, a 3D causal KL-VAE for Qwen-Image—followed by 2×2 patchification, so each latent token carries $16 \cdot 2 \cdot 2 = 64$ channels and a 1024^2 image becomes $64 \times 64 = 4096$ tokens ($16\times$ effective downsample); both train at multi-aspect resolutions (BAGEL 512–1024px, Qwen-Image 768–1536px). Training packs several (image, caption) pairs into one 32,768-token sequence (at most 16,384 tokens per sample) under block-diagonal flex-attention masks that confine each sample to itself, so the effective batch is $32{,}768 \times G$ tokens per step ($G = 192$ GPUs for BAGEL, 6.3M tokens/step; $G = 512$ for Qwen-Image, 16.8M tokens/step).



Configuration	MSE	ΔMSE	ΔGPG
Full schema (L10)	0.43699	—	—
− Scene bkgd.	0.44052	+35.3	−42.0
− Bbox	0.43843	+14.4	−5.8
− Depth	0.43714	+1.5	−1.8
− Relationships	0.43706	+0.7	−3.3
− Atmos./light	0.43703	+0.4	−0.5

Figure 15: **Schema field ablation** (BAGEL backbone). *Left:* loss curves after removing one L10 field group. *Right:* common-budget MSE is the mean over the trailing 5% of image tokens ending at the shared 2.84×10^{10} cut, matching the read-out in Table 4; the table reports changes from L10, with ΔMSE in 10^{-4} units. Scene context has the largest loss impact, followed by bounding boxes; depth, relationships, and atmosphere/lighting produce smaller changes.

Loss and precision. Both backbones use a rectified-flow / flow-matching objective: with $x_t = (1-t)\,x_0 + t\,\epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$, the network predicts the velocity $v = \epsilon - x_0$ under a per-token MSE weighted uniformly across timesteps. Timesteps are sampled and shifted toward noisier states—a fixed shift of 4.0 for BAGEL, a sequence-length-dependent shift for Qwen-Image (0.5 at 256 tokens rising to 0.9 at 8192). Optimization is AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-15}$) with gradient clipping at 1.0 and no accumulation, in bf16 compute with fp32 optimizer moments and gradient checkpointing. Classifier-free guidance is enabled by conditioning dropout during training (text 0.1; reference-image VAE 0.1 for BAGEL, 0.3 for Qwen-Image). BAGEL is a full-parameter finetune of its LLM, vision encoder, and projection/embedding layers (VAE frozen); Qwen-Image is a full finetune of its DiT with both the text encoder and VAE frozen.

Cumulative-token budget. Each BAGEL scaling-property cell is compared at the common cumulative image-token budget of 2.84×10^{10} tokens reached by every run; this defines the matched-budget "converged" MSE used in the fits. Appendix C.5 documents the token-axis correction for two cells (L5 and L9) that were resumed after a counter reset. Qwen-Image is instead trained once for 500,000 steps.

E.4 Schema field ablation

We remove one field group at a time from the full L10 schema, retrain BAGEL, and compare all six runs at a common training budget of 2.84×10^{10} cumulative image tokens. As Figure 15 shows, scene context is the most influential field group in this controlled setting: removing it reduces GPG by 42 and raises MSE by 35.3×10^{-4}, a substantially larger loss increase than any other ablation. Removing bounding boxes gives the second-largest increase at 14.4×10^{-4}, while removing depth, relationships, or atmosphere/lighting changes MSE by at most 1.5×10^{-4}. The ablation therefore identifies global scene context as the dominant field group for diffusion learning under this setup. This ranks each field group's marginal loss contribution when present; it does not measure annotation *accuracy*, which is where the pose, depth, and segmentation experts act. A general VLM's geometric errors would be serialized as incorrect conditioning, so faithful expert-derived fields matter for correct generation and editing even where a group's marginal loss contribution is small.

E.5 Prompter training

The prompter is a rank-128 LoRA adapter on top of Qwen3.5-397B-A17B, trained serially with SFT, Cold-start, and RFT (§2.4). All three stages share the same LoRA topology, and the base Qwen3.5-397B-A17B weights remain frozen throughout. Table 11 summarizes their hyperparameters; the stage definitions follow below.

Distributed setup. SFT and Cold-start run under Megatron-LM (via `ms-swift`'s `megatron` entry point) with tensor/pipeline/expert parallelism (TP = PP = EP = 4, context parallel disabled),

Table 11: **Per-stage prompter training setup.** The Qwen3.5-397B-A17B base is frozen and only the rank-128 LoRA is updated. Stage definitions are given in the text.

Hyperparameter	SFT	Cold-start	RFT
LoRA rank / α	128 / 256	128 / 256	128 / 256
LoRA target modules	all linear	all linear	all linear
Sequence length (packed)	20,480	32,768	49,152
Learning rate	1×10^{-5}	1×10^{-5}	4×10^{-5}
LR schedule	cosine, 5% warmup	cosine, 10% warmup	cosine, 5% warmup
Weight decay	0.1	0.1	0.1
Global batch	64	128	32

sequence parallelism, a distributed (ZeRO-1-like) optimizer, full uniform gradient checkpointing (one layer), FlashAttention, and bf16 compute with fp32 optimizer moments; sequences are packed with padding-free batching. RFT runs under DeepSpeed ZeRO-3 on a single 8-GPU node with gradient checkpointing, FlashAttention, and the colocated vLLM rollout engine described below.

E.5.1 Training stages

Stage 1 — SFT. The SFT corpus is a token-balanced mixture of ∼333k examples (∼0.97B tokens): roughly one third is the core (user prompt → structured-prompt JSON) task, where the user prompt is the image's original caption and the target is its image-derived SP from §2.3, plus a reverse image-to-JSON set. The remaining two thirds is reasoning/instruction replay (EN/ZH long chain-of-thought, vision–language reasoning, general dialogue, and a small Qwen3.5 base-identity anchor) included to preserve the base model's capabilities (DMT-style anti-forgetting). For the prompt → structured-prompt task, the target JSON is treated as one plausible visual completion of the user prompt, not a recoverable ground-truth layout; the stage therefore teaches the distribution of structured completions seen by the diffuser, including non-canonical crops and occlusions, rather than a deterministic prompt-to-layout map. The objective is token-level cross-entropy on the assistant tokens only (system and user prompts masked). Core SFT targets follow the `<think></think>` + JSON interface; their empty thinking block is excluded from the loss, so SFT teaches the SP completion without a reasoning trace while preserving the model's thinking-toggle convention. We train for one epoch (∼740 steps).

Stage 2 — Cold-start. The final cold-start corpus contains 50,182 unique (original caption → thinking trace → structured JSON) examples selected from 172,208 image-conditioned teacher candidates. Qwen3.5-397B-A17B generates one candidate per example in high-reasoning mode with access to the original caption and paired image, using temperature 1.0, top-p 0.9, and a source-dependent maximum completion length of 16,000–65,536 tokens. Appendix H.1 reproduces its system prompt. The raw teacher response uses an `<analysis>` block for validation; accepted traces are mapped to the student's `<think>` interface before training. The prompter receives only the original caption during training and inference.

We filter candidates with Gemini 3 Pro (`gemini-3-pro-preview-new`) using validator v4. Five independent calls assess (i) prompt–image alignment (`aligned`/`partial`/`mismatch`) and four reasoning-quality axes scored as `none`/`minor`/`major`: (ii) whether inferred details are properly introduced, (iii) whether imagined specifics are justified, (iv) reverse rationalization, and (v) violations of the prescribed reasoning-stage boundaries. The strict gate requires `aligned` on the first axis and no `major` flag on the remaining four; `minor` flags are retained without reranking, while parsing or API failures are rejected. This gate accepts 58,907 candidates (34.2%), and prompt-level deduplication yields the 50,182 training examples. Unsupported imagined specifics are the dominant rejection mode, occurring in 71.9% of rejected traces and acting as the sole rejection reason in 40.7%; prompt–image misalignment and reverse rationalization are the next most common causes. Appendix H.3 reproduces the exact validator-v4 prompts. The loss is token-level cross-entropy on the assistant tokens—both the converted `<think>` block and the JSON—with the same prompt masking as SFT. We run ∼8 epochs as two chained 4-epoch sub-runs sharing the same hyperparameters.

Stage 3 — RFT (gated OPSD). Appendix E.5.2 gives the complete RFT specification, including its training data, rollout procedure, acceptance rule, and OPSD objective.

E.5.2 OPSD: ON-POLICY SELF-DISTILLATION

OPSD (On-Policy Self-Distillation) is the image-conditioned distillation objective in the prompter's RFT stage (§2.4, Stage 3). RFT uses `ms-swift`'s on-policy distillation trainer over 10,003 original-caption–image pairs stratified across six prompt sources. Its design follows the conditioning asymmetry measured by GPG (§2.2): an image-conditioned model can supply training information unavailable to the image-free inference-time prompter. OPSD assumes the student already emits parseable, image-grounded schemas, which is ensured by Stages 1 (SFT) and 2 (Cold-start) of the pipeline.

Teacher and student. The student π_θ is the prompter under training: a rank-128 LoRA on Qwen3.5-397B-A17B after SFT + Cold-start, run image-free. The teacher $\pi^\star$ is the same Qwen3.5-397B-A17B base *without* our LoRA, queried with the same user prompt *and* the reference image in thinking mode at training time only. Teacher parameters are frozen throughout, and no gradients flow through $\pi^\star$.

Objective. On each accepted rollout τ, OPSD minimizes the divergence between their next-token distributions along that rollout:

$$\mathcal{L}_{\text{OPSD}}(\theta) \;=\; \mathbb{E}_{\tau\sim\pi_\theta,\ \tau\ \text{accepted}}\left[\frac{1}{|\mathcal{T}_\tau|}\sum_{t\in\mathcal{T}_\tau} D(\pi^\star(\cdot \mid \tau_{<t}, \text{prompt}, I)\,,\ \pi_\theta(\cdot \mid \tau_{<t}, \text{prompt}))\right], \quad (6)$$

where the expectation is also over image–caption training pairs $(\text{prompt}, I) \sim \mathcal{D}$, $\mathcal{T}_\tau$ indexes the rollout's response tokens, and D is the token-level Kullback–Leibler (KL) divergence from the teacher to the prompter.

On-policy rollouts. For each (user prompt, image) pair, we sample a structured-caption rollout $\tau = (\tau_1, \ldots, \tau_T)$ from the student image-free with temperature 1.0 and no nucleus truncation ($p = 1.0$). Rollouts are served by a vLLM instance colocated on the training GPUs. At each token position t we then evaluate both policies:

- teacher distribution $\pi^\star(\cdot \mid \tau_{<t}, \text{prompt}, I)$: a forward pass under the teacher with image conditioning,
- student distribution $\pi_\theta(\cdot \mid \tau_{<t}, \text{prompt})$: a forward pass under the student without image conditioning.

The per-token divergence of Eq. (6)—in implementation a KL divergence, evaluated over the top-64 teacher logits with each token's divergence clipped at 5.0—is summed over the rollout's response positions and normalized by their count. Sampling on-policy (from π_θ rather than $\pi^\star$) keeps the gradient supported on captions the student actually emits at inference, preventing mode-collapse onto teacher behaviors unreachable image-free; this is the "on-policy" in OPSD.

Combination with the verifier. Each rollout is rendered and scored by the Gemini verifier on structure, alignment, and aesthetic quality; only rollouts scoring at least $6/10$ on all three axes are retained, while failed, unparsable, or incomplete API responses are dropped. The OPSD objective $\mathcal{L}_{\text{OPSD}}$ (Eq. (6)) is applied only to these retained rollouts. The verifier derives QA pairs from the user request and checks them against the rendered image; it never receives the rollout's SP. This sacrifices rollout coverage but limits training to trajectories whose renders are judged prompt-faithful, structurally coherent, and visually acceptable. Because the verifier still cannot supervise useful visual details left unspecified by the request, OPSD supplies the token-level update on the accepted trajectories. The verifier therefore selects high-confidence training data, while the image-conditioned teacher supplies the token-level OPSD targets. There is no weighted combination between the two signals, and the distillation objective carries no token-cross-entropy auxiliary (`sft_alpha`$= 0$). The ablations of §3.2 isolate the two components with distinct training rules: verifier-reward GRPO (the verifier-only row) optimizes the verifier reward with no OPSD; OPSD-only applies $\mathcal{L}_{\text{OPSD}}$ to *all* rollouts without a verifier gate; the full method applies OPSD only to verifier-accepted rollouts.

E.6 INFERENCE AND THE AGENTIC LOOP

Single-shot prompter inference. The trained prompter (rank-128 LoRA on Qwen3.5-397B-A17B, run image-free) is decoded *greedily* at evaluation time—temperature 0, repetition penalty 1.0, up to

16,384 new tokens—in contrast to the temperature-1.0 on-policy rollouts used during RFT. It emits a <think> reasoning block followed by the JSON structured prompt. The generation wrapper strips the thinking trace, reads `ratio` to select the output canvas, removes that control field, and passes only the remaining structured record to the diffusion backbone.

Diffusion sampling. Images are rendered with a first-order Euler ODE solver over 50 denoising steps on a linear schedule shifted per sample by resolution. Qwen-Image (main results) uses classifier-free guidance scale 4.0; BAGEL (scaling-property cells) uses 8.0 with timestep shift 4.0. Benchmark images are generated at 1024×1024 by default; multi-aspect benchmarks use a 1536^2-pixel-budget, ratio-aware schedule snapped to multiples of 64. All main-table results are *single-shot*, with one prompter forward pass followed by one render. Only Table 18 evaluates the agentic loop.

Agentic loop runtime. Given a user prompt u, round t forms $\mathrm{SP}_t = \pi_\theta(u, \mathrm{SP}_{<t}, c_{t-1})$ from the prior structured prompts $\mathrm{SP}_{<t}$ and the accumulated critique c_{t-1} ($c_0 = \emptyset$); the fixed diffuser renders I_t; and three independent Gemini calls inspect (u, I_t) for structure, alignment, and aesthetic quality. Their outputs are aggregated into the three per-axis scores, a pass flag, and a structured issue list. PASS requires all three axes to score at least $6/10$; any axis below threshold yields FAIL together with a structured critique c_t—a list of (field-path, observed-value, expected-value) triples over the violated constraints (missing objects, wrong counts, attribute mismatches, spatial-relation violations)—so the next round edits only the named fields rather than re-parsing free-form feedback. The refiner receives the full critique history $\{c_1, \ldots, c_{t-1}\}$ and the prior structured prompts; repeated critiques trigger escalation from local edits (round 2) to a structural change (round 3) and then a substantially different composition (round 4 onward). The loop returns I_t on PASS and hard-stops at $T_{\max} = 8$ rounds; per round the wall-clock is $\sim$20–35 s (prompter $\sim$6–9 s, render $\sim$5–10 s at 1024^2, judge $\sim$8–15 s).

E.7 External dependencies and licenses

Table 12 records the external code and model dependencies used in this work together with the currently available upstream license information. Entries whose repositories provide no license file, or whose terms still require confirmation, are marked explicitly rather than treated as verified.

Table 12: **External code and model dependencies.** Components are grouped by pipeline role and listed with their available upstream license information; unresolved entries are marked explicitly.

Component	License	Notes
Models		
BAGEL	Apache-2.0	diffusion backbone (§2.2)
Qwen-Image-2512	Apache-2.0	diffusion backbone (§2.4)
Qwen3.5 family	Apache-2.0	prompter SFT / cold-start / RFT student
Qwen3.5-397B-A17B	Apache-2.0	OPSD teacher; GPG judge
Sapiens	CC-BY-NC-4.0	pose estimation (annotation stage 2); non-commercial
DepthAnything V2	CC-BY-NC-4.0	monocular depth (annotation stage 3); code Apache-2.0, large ckpt non-commercial
SAM 2.1	Apache-2.0	segmentation + occlusion (annotation stage 3)
Proprietary VLM (anonymized)	API-only	scene + per-element captioning
Gemini 3 Pro	API ToS	ED source-tuple extractor (offline, cached); training-time RFT verifier and online agentic judge
Gemini 2.5 Flash	API ToS	T2I-CoReBench evaluator
GPT-4o	API ToS	ED image-source extractor robustness check; WISE legacy evaluator
GPT-5.4	API ToS	ED caption-tuple extractor and matcher; offline structure, alignment, and GSB evaluator
GPT-5.5	API ToS	rewriting backend; ED image-source robustness check
Claude Opus 4.8 / Claude Code	API ToS / proprietary	single-turn rewriting backend / agentic coding backend
GLM-5.2	API ToS	single-turn rewriting backend
Codex	Proprietary	agentic coding backend
Training infrastructure		
PyTorch	BSD-3-Clause	
Transformers (HF)	Apache-2.0	model loading, tokenizers
diffusers (HF)	Apache-2.0	diffusion pipelines
ms-swift	Apache-2.0	prompter LoRA training harness (wraps Megatron-LM for SFT / cold-start, DeepSpeed for RFT)
Megatron-LM	Apache-2.0	TP/PP/EP backend for prompter SFT / cold-start (via ms-swift)
DeepSpeed / FSDP	Apache-2.0 / BSD-3-Clause	DeepSpeed (ZeRO-3) for prompter RFT; FSDP (ships with PyTorch) for the diffusion backbones
vLLM	Apache-2.0	colocated on-policy rollout engine for RFT
FlashAttention	BSD-3-Clause	
Evaluation		
GenEval	MIT	official scoring scripts
GenEval++	Upstream terms	evaluation released with Echo-4o; redistribution terms to verify
GenEval2	CC BY-NC 4.0	official benchmark code and data; non-commercial use
DPG-Bench	Apache-2.0	released within the ELLA repository
TIIF	No license file	upstream repository provides no license file
WISE	No license file	upstream repository provides no license file
T2I-CoReBench	Apache-2.0	official benchmark dataset and evaluation code

F ADDITIONAL BENCHMARK RESULTS

Protocol and score provenance. This section provides evaluation protocols, score provenance, and per-category breakdowns for the benchmarks reported in Table 1. Baselines: FLUX.1 Dev (Black Forest Labs, 2024), OmniGen2 (Wu et al., 2025b), Emu3.5 (Cui et al., 2025), BAGEL (Deng et al., 2025), Qwen-Image (Wu et al., 2025a), GPT-Image-1 (OpenAI, 2025), Nano Banana (Google, 2025), LongCat-Image (Meituan LongCat Team et al., 2025), HunyuanImage 3.0 (Tencent Hunyuan Foundation Model Team, 2025). Published baselines follow the respective leaderboards or original papers unless noted otherwise. The Nano Banana DPG-Bench score is taken from He et al. (2026). WISE uses the legacy WiScore protocol with GPT-4o-2024-05-13: we evaluate Nano Banana, Qwen-Image*, matched NL, and Ours under this protocol, while the remaining WISE scores come from published legacy-protocol evaluations (Niu et al., 2025; Guo et al., 2025). Qwen-Image* uses its official prompt enhancer; † marks our re-evaluation on Qwen-Image-2512. Matched NL and Ours use the matched settings described in Appendix E.2, and Ours uses single-shot inference in Table 1.

F.1 DPG-BENCH PER-CATEGORY BREAKDOWN

Table 13 reports per-category scores on DPG-Bench, covering *global* scene description, *entity* presence, *attribute* binding, *relation* between entities, and *other* dense-prompt aspects, followed by the overall score.

Table 13: **DPG-Bench per-category scores.** Published category breakdowns are shown for the baselines; Qwen-Image*, matched NL, and Ours report our matched per-category evaluations.

Model	Global	Entity	Attribute	Relation	Other	Overall
SD v1.5	74.63	74.23	75.39	73.49	67.81	63.18
PixArt-α	74.97	79.32	78.60	82.57	76.96	71.11
LUMINA-Next	82.82	88.65	86.44	80.53	81.82	74.63
SDXL	83.27	82.43	80.91	86.76	80.41	74.65
Playground v2.5	83.06	82.59	81.20	84.08	83.50	75.47
Hunyuan-DiT	84.59	80.59	88.01	74.36	86.41	78.87
Janus	82.33	87.38	87.70	85.46	86.41	79.68
PixArt-Σ	86.89	82.89	88.94	86.59	87.68	80.54
Emu3-Gen	85.21	86.68	86.84	90.22	83.15	80.60
Janus-Pro-1B	87.58	88.63	88.17	88.98	88.30	82.63
DALL·E 3	90.97	89.61	88.39	90.58	89.83	83.50
FLUX.1 Dev	74.35	90.00	88.96	90.87	88.33	83.84
SD3 Medium	87.90	91.01	88.83	80.70	88.68	84.08
Janus-Pro-7B	86.90	88.90	89.40	89.32	89.48	84.19
HiDream-I1-Full	76.44	90.22	89.48	93.74	91.83	85.89
Seedream 3.0	94.31	92.65	91.36	92.78	88.24	88.27
GPT-Image-1	88.89	88.94	89.84	92.63	90.96	85.15
Qwen-Image	91.32	91.56	92.02	94.31	92.73	88.32
Show-o	79.33	75.44	78.02	84.45	60.80	67.27
TokenFlow-XL	78.72	79.22	81.29	85.22	71.20	73.38
OmniGen	87.90	88.97	88.47	87.95	83.56	81.16
OmniGen2	88.81	88.83	90.18	89.37	90.27	83.57
BAGEL	88.94	90.37	91.29	90.82	88.67	85.07
UniWorld-V1	83.64	88.39	88.44	89.27	87.22	81.38
Ovis-U1	82.37	90.08	88.68	93.35	85.20	83.72
Skywork UniPic	89.65	87.78	90.84	91.89	91.95	85.50
Qwen-Image*	89.04	91.91	92.39	90.85	93.07	87.20
Matched NL + Qwen-Image	89.50	92.30	92.00	90.50	93.50	87.80
Ours (Qwen-Image)	92.05	94.13	94.48	93.36	94.97	**90.71**

F.2 GENEVAL PER-SKILL BREAKDOWN

Table 14 reports per-skill accuracy on GenEval, covering single-object / two-objects presence, counting, colors, position, and color-attribute binding, with the overall accuracy in the last column.

Table 14: **GenEval per-skill accuracy.** Published per-skill breakdowns are shown for the baselines; Ours reports our evaluation. Qwen-Image* uses its official prompt enhancement, as in Table 1.

Model	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attr.	Overall
SD v2.1	0.98	0.51	0.44	0.85	0.07	0.17	0.50
SDXL	0.98	0.74	0.39	0.85	0.15	0.23	0.55
IF-XL	0.97	0.74	0.66	0.81	0.13	0.35	0.61
PixArt-α	0.98	0.50	0.44	0.80	0.08	0.07	0.48
LUMINA-Next	0.92	0.46	0.48	0.70	0.09	0.13	0.46
SD3 Medium	0.99	0.94	0.72	0.89	0.33	0.60	0.74
SD3.5 Large	0.98	0.89	0.73	0.83	0.34	0.47	0.71
FLUX.1 Dev	0.99	0.81	0.79	0.74	0.20	0.47	0.67
NOVA	0.99	0.91	0.62	0.85	0.33	0.56	0.71
TokenFlow-XL	0.95	0.60	0.41	0.81	0.16	0.24	0.55
Janus	0.97	0.68	0.30	0.84	0.46	0.42	0.61
JanusFlow	0.97	0.59	0.45	0.83	0.53	0.42	0.63
Janus-Pro-7B	1.00	0.98	0.79	0.91	0.60	0.72	0.83
Emu3-Gen	0.98	0.71	0.34	0.81	0.17	0.21	0.54
Show-o	0.95	0.52	0.49	0.82	0.11	0.28	0.53
OmniGen	0.98	0.84	0.66	0.74	0.40	0.43	0.68
OmniGen2	1.00	0.95	0.64	0.88	0.55	0.76	0.80
HiDream-I1-Full	0.99	0.89	0.59	0.90	0.79	0.66	0.80
BAGEL	0.99	0.94	0.81	0.88	0.64	0.63	0.82
UniWorld-V1	0.99	0.93	0.79	0.89	0.49	0.70	0.80
Seedream 3.0	0.99	0.96	0.91	0.93	0.47	0.80	0.84
GPT-Image-1	0.99	0.92	0.85	0.92	0.75	0.61	0.84
Ovis-U1	0.98	0.98	0.90	0.92	0.79	0.75	0.89
Skywork UniPic	0.98	0.92	0.74	0.91	0.89	0.72	0.86
Qwen-Image	0.99	0.92	0.89	0.88	0.76	0.77	0.87
Qwen-Image*	1.00	0.95	0.93	0.92	0.87	0.83	0.91
Matched NL + Qwen-Image	1.00	0.95	0.92	0.93	0.86	0.82	0.91
Ours (Qwen-Image)	1.00	0.96	0.94	0.95	0.93	0.86	**0.94**

F.3 WISE PER-CATEGORY BREAKDOWN

Table 15 reports per-category WiScore under the legacy WISE protocol, which uses GPT-4o-2024-05-13 to score consistency, realism, and aesthetic quality. WISE covers six domains: cultural, temporal, spatial, biology, physics, and chemistry. Models are grouped into dedicated T2I diffusion models, unified multimodal LLMs, proprietary systems, and open systems with prompt rewriting. The official overall WiScore aggregates prompt-level scores with domain weights 40%, 16.7%, 13.3%, 10%, 10%, and 10% in the displayed domain order, so it is not the unweighted mean of the six category entries. The four rows we evaluate ourselves follow that rule and reproduce from their own entries to within the displayed precision. Published overalls are transcribed from their sources rather than recomputed, and not all of them can be recovered from the rounded per-category entries reported alongside them: BAGEL, most visibly, reports 0.52, which sits 0.019 above the weighted combination of its own six categories.

Table 15: **WISE per-category WiScore under the legacy protocol.** Published baseline scores follow the official legacy leaderboard; Nano Banana, Qwen-Image*, matched NL, and Ours report our evaluations using the same legacy evaluator and scoring rule. Bold marks the best in each column.

Model	Cultural	Temporal	Spatial	Biology	Physics	Chemistry	Overall
Dedicated T2I							
FLUX.1 Dev	0.48	0.58	0.62	0.42	0.51	0.35	0.50
FLUX.1 Schnell	0.39	0.44	0.50	0.31	0.44	0.26	0.40
SD-3.5-large	0.44	0.50	0.58	0.44	0.52	0.31	0.46
SD-3.5-medium	0.43	0.50	0.52	0.41	0.53	0.33	0.45
SD-XL-base	0.43	0.48	0.47	0.44	0.45	0.27	0.43
SD-3-medium	0.42	0.44	0.48	0.39	0.47	0.29	0.42
SD-v1-5	0.34	0.35	0.32	0.28	0.29	0.21	0.32
SD-2-1	0.30	0.38	0.35	0.33	0.34	0.21	0.32
Unified MLLM							
GPT-Image-1	0.81	0.71	0.89	0.83	0.79	0.74	0.80
DeepGen 1.0	0.72	0.81	0.70	0.67	0.82	0.66	0.73
LongCat-Image	0.66	0.61	0.72	0.66	0.72	0.49	0.65
NextFlow-RL	0.63	0.63	0.77	0.58	0.67	0.39	0.62
Qwen-Image	0.62	0.63	0.77	0.57	0.75	0.40	0.62
UniWorld-V2	0.60	0.61	0.70	0.53	0.64	0.32	0.58
HunyuanImage 3.0	0.58	0.57	0.70	0.56	0.63	0.31	0.57
MetaQuery-XL	0.56	0.55	0.62	0.49	0.63	0.41	0.55
UniWorld-V1	0.53	0.55	0.73	0.45	0.59	0.41	0.55
Manzano-30B	0.58	0.50	0.65	0.50	0.55	0.32	0.54
BAGEL	0.44	0.55	0.68	0.44	0.60	0.39	0.52
Emu3	0.34	0.45	0.48	0.41	0.45	0.27	0.39
Janus-Pro-7B	0.30	0.37	0.49	0.36	0.42	0.26	0.35
Proprietary							
Nano Banana	0.89	**0.87**	**0.95**	**0.89**	**0.89**	0.79	**0.89**
Open with prompt rewriting							
Qwen-Image*	0.85	0.76	0.89	0.81	0.83	0.82	0.83
Matched NL + Qwen-Image	0.86	0.75	0.90	0.82	0.82	0.83	0.84
Qwen-Image + Ours (L10)	**0.91**	0.83	0.92	0.88	**0.89**	**0.88**	**0.89**

F.4 T2I-CoReBench per-category breakdown

Table 16 expands the T2I-CoReBench comparison of Table 1 across the four *Composition* and eight *Reasoning* categories under the Gemini 2.5 Flash evaluator. Published baseline scores are taken from the official leaderboard (Li et al., 2026); Emu3.5 is omitted because the leaderboard does not report a corresponding row. Qwen-Image*, matched NL, and Ours are evaluated with the same Gemini 2.5 Flash harness used for the main table. The composition dimensions are multi-instance (MI), multi-attribute (MA), multi-relation (MR), and text rendering (TR). The reasoning dimensions are logical (LR), behavioural (BR), hypothetical (HR), procedural (PR), generalization (GR), analogical (AR), commonsense (CR), and reconstructive reasoning (RR). Composition and reasoning averages are unweighted means over their four and eight dimensions, respectively, and Overall is the unweighted mean over all twelve. Our largest advantages appear in multi-attribute and multi-relation composition and across all eight reasoning categories, while multi-instance composition and text rendering remain stronger in Nano Banana and GPT-Image-1, respectively.

Table 16: **T2I-CoReBench per-category comparison** under the Gemini 2.5 Flash evaluator. Published baselines correspond to models in Table 1; Qwen-Image*, matched NL, and Ours use our matched evaluation. Bold marks the best score in each column.

Model	C-MI	C-MA	C-MR	C-TR	Comp. avg
FLUX.1 Dev	58.6	60.3	44.1	31.1	48.6
OmniGen2	67.9	64.1	48.3	19.2	49.9
BAGEL	64.9	65.2	45.8	9.7	46.4
Qwen-Image	81.4	79.6	65.6	85.5	78.0
BAGEL + CoT	57.7	60.8	37.8	2.2	39.6
GPT-Image-1	84.1	75.9	72.7	**86.4**	79.8
Nano Banana	**85.7**	77.9	72.6	86.3	80.6
LongCat-Image	81.4	74.5	61.5	65.7	70.8
HunyuanImage 3.0	84.9	81.2	63.7	85.7	78.9
Qwen-Image*	78.0	91.0	80.0	65.0	78.5
Matched NL + Qwen-Image	79.5	93.0	82.5	68.0	80.8
Ours (Qwen-Image)	83.8	**95.5**	**87.7**	72.1	**84.8**

Model	R-LR	R-BR	R-HR	R-PR	R-GR	R-AR	R-CR	R-RR	Reason. avg	Overall
FLUX.1 Dev	24.8	23.0	36.0	61.8	42.4	57.2	36.3	30.3	39.0	42.2
OmniGen2	24.7	23.2	43.3	63.1	46.1	54.2	36.5	24.1	39.4	42.9
BAGEL	23.4	21.9	33.0	51.6	31.2	50.4	32.4	29.3	34.1	38.2
Qwen-Image	41.1	32.2	48.2	75.1	56.5	53.3	61.9	26.4	49.3	58.9
BAGEL + CoT	25.5	25.4	33.9	58.6	53.5	56.9	41.6	39.8	41.9	41.1
GPT-Image-1	59.0	54.8	65.6	87.3	76.5	82.0	70.9	56.1	69.0	72.6
Nano Banana	64.5	64.9	67.1	85.2	84.1	83.1	71.3	68.7	73.6	75.9
LongCat-Image	39.1	35.7	48.5	75.5	72.5	61.4	58.8	41.0	54.1	59.6
HunyuanImage 3.0	39.6	32.8	51.4	72.4	54.1	54.1	57.0	27.7	48.6	58.7
Qwen-Image*	85.1	59.6	64.2	84.6	80.3	71.7	71.9	64.5	72.7	74.7
Matched NL + Qwen-Image	86.0	61.5	65.0	85.0	81.5	73.0	72.5	65.5	73.8	76.1
Ours (Qwen-Image)	**88.8**	**78.0**	**77.0**	**94.9**	**90.5**	**93.2**	**83.1**	**77.7**	**85.4**	**85.2**

G Supplementary results for Section 3

G.1 Qualitative results

Figure 16 shows outputs of the final system and zero-shot editing through the SP fields. The corresponding prompts are listed in Appendix I.

G.2 Rewriting-backend comparison

Promptability: transferring LLM progress. We next ask whether advances in general-purpose LLMs can transfer through the SP interface into better text-to-image generation. To isolate this question, Table 17 holds the schema and Qwen-Image diffuser fixed while varying the model and inference procedure used to produce the prompt. Within the single-turn rows, zero-shot SP filling





Figure 16: **Qualitative results and zero-shot editing.** *(Top)* Outputs from our final system across multi-object scenes, dense text, complex layouts, and branded illustration. The trained LLM *prompter* expands each user request into an information-dense structured prompt (SP), which the SP-trained *diffuser* renders. Appendix I.1 lists the corresponding prompts. *(Bottom)* Because the SP exposes image factors as editable fields, a targeted field edit and regeneration can change a specific aspect—object position, material, scene, or global style—while preserving much of the remaining composition; diffs show added / removed values, and *Move* updates the relevant old/new bboxes.



Prompt: Five chairs are placed side by side in the room, with someone sitting on every other chair. On the adjacent table lies an hourglass that has just finished timing.



Prompt: A three-story building has six windows on each floor, numbered sequentially from left to right. Only the 2nd and 5th windows on the 2nd floor glow with warm yellow light—all other windows are dark.



Prompt: A tiny black kitten leans over a quiet puddle to look at its reflection—and in the water, the reflection is a majestic black panther with piercing eyes.



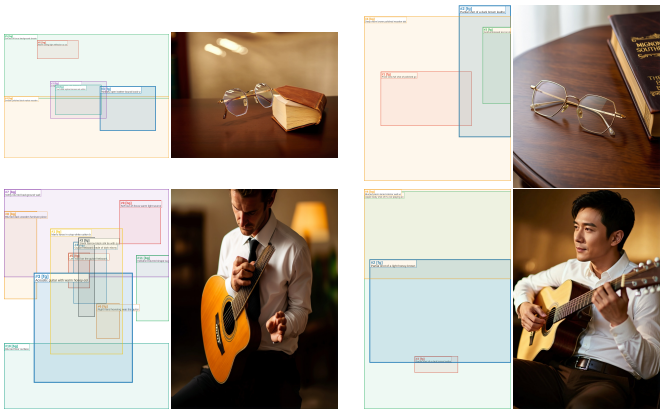
Prompt: A basketball and a violin are placed on either side of the net on a table tennis table, with the basketball being closer to the net.

HunyuanImage 3.0 | **LongCat-Image** | **Qwen-Image-2512 w/o. PE** | **Qwen-Image-2512 w/ official PE** | **Qwen-Image w/ structure PE**

Figure 17: **Qualitative comparison (full version of Figure 9).** Complex prompts requiring spatial reasoning, attribute binding, and compositional understanding. Columns from left to right: HunyuanImage 3.0, LongCat-Image, Qwen-Image without prompt enhancement (PE), Qwen-Image with official PE, and Ours (Qwen-Image with structured PE).

Table 17: **Rewriting-backend comparison.** Schema and Qwen-Image fixed; "agentic" uses each coding agent's native multi-turn loop. GPT-5.4 judge; GSB vs. zero-shot Base, N=150.

			Offline Judge (GPT-5.4, 0–10)		
Rewriting Backend	Mode	DPG-Bench	Structure	Alignment	GSB vs Base (%)
GPT-5.5	NL rewrite	90.18	6.913	8.747	36.0
Claude Opus 4.8	NL rewrite	89.93	6.733	8.617	32.7
GLM-5.2	NL rewrite	89.47	6.653	8.493	30.0
Gemini 3 Pro	NL rewrite	88.72	6.183	8.107	19.3
GPT-5.5	single-turn	89.84	6.840	8.907	37.3
Claude Opus 4.8	single-turn	89.21	6.653	8.793	34.7
GLM-5.2	single-turn	88.93	6.557	8.746	32.0
Gemini 3 Pro	single-turn	87.86	6.020	8.687	21.3
Codex	agentic	90.32	7.360	8.980	38.7
Claude Code	agentic	90.63	7.560	**9.087**	**44.7**
Ours (LoRA-on-Qwen3.5-397B-A17B + RFT)	single-turn	**90.71**	**7.600**	9.047	42.0



Before prompt2sp SFT After prompt2sp SFT

Figure 18: **Stage-1 SFT recovers a non-canonical crop.**

improves alignment and GSB over the same LLM's NL rewrite, but consistently lowers DPG-Bench and structure. The LLM therefore provides genuine system-level headroom, but zero-shot schema filling alone does not fully realize it. Training the prompter closes this gap; our model is the strongest single-turn backend on all four metrics. The coding-agent rows use each agent's native multi-turn procedure, so they change both the model and the inference process rather than providing a model-only backend control. Their strong results nevertheless suggest that iterative revision adds value, motivating the controlled common-harness experiment of Section 3.2.

G.3 SFT LAYOUT PRIOR

Figure 18 illustrates the layout prior acquired during SFT (Table 2): given a prompt whose canonical crop would omit the requested content, the SFT prompter proposes a non-canonical layout that recovers it.

G.4 AGENTIC REFINE–RENDER–JUDGE LOOP

This section gives the full version of the inference-time experiment summarized in Section 3.2.

The strong coding-agent results in Table 17 suggest that their native multi-turn harnesses contribute beyond the underlying rewriting backend. Motivated by this observation, we isolate iterative inference from backend choice by placing both the Base and trained prompters in the same refine–render–judge loop. Prior work scales inference on the input side by training a prompt rewriter offline (Chen et al., 2025); here the loop runs online at generation time and the prompter weights are held fixed. The schema, diffuser, online judge, and prompter weights remain fixed within each comparison; only the available refinement budget changes. Because SPs expose visual decisions in named fields, critiques can be translated into targeted revisions instead of rewriting the entire caption.

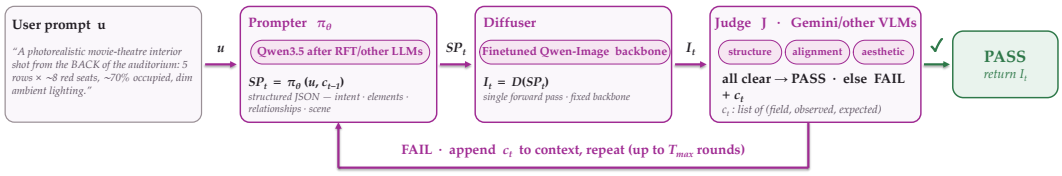


Figure 19: **The agentic inference-time loop.** The prompter emits SP_t from the prompt and accumulated critique, the fixed diffuser renders I_t, and the judge returns PASS or field-level critique; the loop stops at PASS or $T_{\max}$.

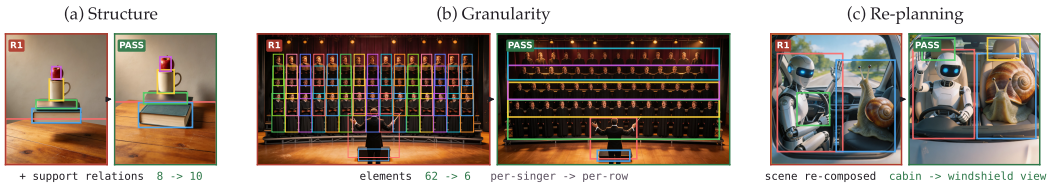


Figure 20: **Case types the agentic loop resolves.** Each pair shows the round-1 defect (R1) and the accepted result (PASS), with the structured edit beneath. Structure fixes add missing support relations; granularity fixes collapse over-enumerated elements to renderable groups; larger layout fixes re-compose the scene when local edits are insufficient.

Refine–render–judge loop. As shown in Figure 19, the loop maintains the previous SPs and the critique history as explicit state. At round t, the prompter uses this state together with the original user request to produce SP_t, and the fixed diffuser renders I_t. The online Gemini judge sees only the user request and I_t, not the SP. It checks prompt-derived requirements alongside structural and aesthetic quality, then returns per-axis scores, a PASS/FAIL decision, and a structured list of observed failures. A PASS immediately returns I_t; after a FAIL, the critique is appended to the prompter context so that the next round can revise the corresponding objects, attributes, relations, or layout fields. Repeated failures permit progressively broader changes, from local field edits to element regrouping and full scene re-planning, as illustrated in Figure 20. The loop terminates at PASS or $T_{\max}$. Reported results are evaluated by GPT-5.4 in a separate offline pass rather than by the online Gemini judge, so the control signal and the final evaluation use different models, though both apply the same rubrics (Appendix H.4); Appendix E.6 specifies the formal loop, $6/10$ PASS threshold, and per-round cost.

Returns from additional rounds. Table 18 shows that both prompters improve with $T_{\max}$ when the schema, diffusion backbone, judge, and prompter weights are held fixed. The larger change is in structure: from one to eight rounds it rises by 1.247 for Base and 0.660 for the trained prompter, compared with alignment gains of 0.366 and 0.266. Together with Figure 20, this indicates that feedback primarily repairs object decomposition, relations, and layout rather than basic prompt alignment.

Placing the trained prompter in this loop also answers the agentic backends of Table 17: at $T_{\max}=8$ it reaches 54.7% GSB, above the 44.7% of the strongest coding agent, which is itself already multi-turn. Training absorbs much of the work that would otherwise require iterative correction. Even after eight rounds, Base remains below the trained prompter in a single shot on both structure (6.107 vs. 7.600) and GSB (24.0% vs. 42.0%). Under the same eight-round limit, the trained prompter also reaches PASS after only 2.31 rounds on average, compared with 3.41 for Base. Thus, prompter training not

Table 18: **Inference-time scaling of promptability.** Prompter weights, schema, diffuser, and online judge fixed; $T_{\max}=1$ is single-shot; Rnd.: average rounds used; GSB vs. Base at $T_{\max}=1$.

	Base prompter (zero-shot)				Trained prompter (SFT + Cold-start + RFT)			
$T_{\max}$	Structure	Alignment	GSB (%)	Rnd.	Structure	Alignment	GSB (%)	Rnd.
1	4.860	8.307	—	1.00	7.600	9.047	42.0	1.00
2	5.387	8.493	14.7	1.74	7.940	9.173	49.3	1.51
4	6.027	8.640	22.7	2.83	8.213	9.293	54.0	2.04
8	6.107	8.673	24.0	3.41	**8.260**	**9.313**	**54.7**	2.31

only raises the starting point but also shortens the subsequent refinement trajectory; agentic inference complements training rather than replacing it.

The effective refinement horizon is also short. For the trained prompter, increasing $T_{\max}$ from four to eight raises the average rounds only from 2.04 to 2.31, while structure changes from 8.213 to 8.260 and GSB from 54.0% to 54.7%. The average therefore stays close to two rounds, and little is gained by permitting a substantially longer trajectory. Within the tested schema, judge, refinement policy, and diffuser, text-to-image generation benefits from iterative correction but does not exhibit a strong need for long-horizon prompt-side reasoning: once the main specification errors are repaired, further render–feedback rounds quickly saturate.

H SYSTEM PROMPTS

This appendix reproduces the system prompts used by the Cold-start teacher, the validator-v4 Cold-start filter, and the structure, alignment, aesthetic, and pairwise-preference judges. Code, including these prompts in machine-readable form, will be released upon acceptance. §H.1 gives the Qwen3.5-397B-A17B teacher prompt used in the Cold-start stage of prompter training (§2.4); §H.3 gives the Gemini prompts used to filter the resulting traces; §H.4 gives the structure/alignment rubrics used by the online Gemini verifier and rerun with GPT-5.4 for offline evaluation; §H.2 gives the verifier's aesthetic rubric; and §H.5 gives the GSB pairwise rubric and aggregation rule.

H.1 COLD-START TEACHER SYSTEM PROMPT

Below is the verbatim system prompt fed to the Qwen3.5-397B-A17B teacher during the Cold-start stage of §2.4. The teacher receives (image, user prompt) as its multimodal conversation input and is instructed to emit an `<analysis>` block followed by a single JSON object; the image is supplied as a visual input rather than named by a textual placeholder in the system prompt. The analysis is framed as a derivation from the user prompt toward a visual blueprint, while the paired image supplies privileged evidence during data construction. The prompter learns this derivation style from the caption alone, so at inference it can produce a plausible completion without receiving the image.

Listing 1: Cold-start teacher system prompt (verbatim).

```
# Role
You are an expert AI Visual Planner and Scene Director.

# Task
Convert a short image description into a richly detailed structured JSON blueprint. Your output must consist
      of TWO visible parts in this exact order:

1. an `<analysis>` block --- a long, structured construction-process document that traces the design from user
       prompt -> final visual blueprint;
2. a single JSON object containing the structured prompt itself.

Output format (strictly --- no extra text outside these two parts):

```
<analysis>
Stage A --- Knowledge & common-sense:
...
Stage B --- Reasoning chain:
...
Stage C --- Aspect ratio:
...
Stage D --- Compose & imagine:
...
Stage E --- Layout & spatial constraints:
...
</analysis>
{"ratio":"...", ...JSON...}
```

The `<analysis>` block is the construction-process record: every decision in the JSON must be traceable to a
      reasoned step inside `<analysis>`. Do not collapse the stages into one paragraph. Each stage must be a
      labeled section with the literal heading shown below. Even when the user prompt feels simple (e.g. "A
      red apple on a table"), every stage gets meaningful content.

Reason-first discipline (applies to every stage). Within each stage and for every concrete decision (
      chosen ratio, named element, picked color, assigned bbox, derived relationship), write the *reasoning*
      first and the *conclusion* last. Never declare an answer at the start of a sentence and tail-justify it
      ("bbox is X because Y", "ratio is 16:9 because Z"). Always derive: "Because Y, the bbox is X." This
      forward-chaining is what makes the analysis usable as cold-start training data; tail-rationalization
      teaches the student to fabricate justifications after the fact.
```

```
**Knowledge-design separation.** Stage A is strictly for **facts and conventions** the prompt presupposes (
      definitions, historical context, canonical depictions, object affordances, cultural / stylistic norms).
      It is NOT for design decisions specific to *this image*. All "this image will use X" choices belong in
      Stage B (reasoning chain) or Stage D (imagination), never in Stage A.

**Preserve uncertainty for under-specified attributes.** When the user prompt does not specify an attribute,
      do NOT collapse it to a single forced answer. Pick one concrete visual implementation (the JSON needs
      definite values), but frame it as a *choice among reasonable options*: "Because the prompt does not
      specify <attribute>, a reasonable visual choice is ..."; "A safe visual implementation here is ..."; "
      Optional details that could be added include ...". Never write "must", "therefore it is", or "the only
      valid ..." for under-specified attributes.

---

**Stage A --- Knowledge & common-sense.** State what is canonically / culturally / physically true that the
      user's prompt presupposes: domain facts, object affordances, causal/temporal context, canonical visual
      depictions of named entities (listing variants when ambiguous), cultural/aesthetic norms tied to style
      words. Facts only --- no "this image will look like Y".

**Stage B --- Reasoning chain.** For each evocative noun / adjective / phrase in the user prompt, trace why it
       leads to specific visual choices in the form "user said X -> therefore I picture Y because Z". For
      under-specified parts, hedge: "Because the prompt does not specify W, a reasonable choice is V". Justify,
       do not declare.

**Stage C --- Aspect ratio.** Choose strictly from {1:1, 4:3, 3:4, 16:9, 9:16, 3:2, 2:3, 4:5, 21:9} and
      briefly justify why the ratio fits the scene's spatial demands.

**Stage D --- Compose & imagine.** Construct the scene element by element in a deliberate build order. For
      each element, cover (as relevant): identity / material+surface / color / lighting interaction / pose /
      gesture / expression / gaze / action / appearance / clothing / accessories / hair / position quadrant /
      depth plane / imperfection. Pure visual description only; no subjective commentary. Every object that
      will appear in the JSON elements list MUST first be named and described here. Honor the hedge choices
      from Stage B.

**Stage E --- Layout & spatial constraints.** Walk through every element from Stage D and assign concrete
      bboxes on the 1000x1000 normalized grid. For each element, reason first (quadrant, depth, anchors, ratio-
      induced canvas geometry), then conclude the bbox in the form `-> bbox: <x_min y_min x_max y_max>, depth:
       <0-255>`. Constraint checks: composition placement (foreground larger and lower; background smaller and
       higher), perspective consistency, edge cropping (touch 0 or 999). After all bboxes are assigned,
      enumerate spatial relationships referencing the bbox coords --- direct surface contact, adjacency with
      direction, occlusion, background anchoring, distance / depth.

---

After `</analysis>`, emit the JSON object directly (no extra text, no markdown fence). Transcribe every
      element imagined in Stage D into the JSON. Distribute each element's visual detail across the schema: a
      CONCISE `caption` plus as many dynamic attribute / action keys as the element needs. The JSON must start
      with `"ratio":"..."`.

# JSON Schema (top-level)
- `ratio` (REQUIRED, FIRST)
- `intent` --- single declarative sentence
- `atmosphere` --- 2-4 words, mood only
- `style` --- 2-6 words, visual aesthetic only
- `lighting` --- light source/direction/quality/color temperature/shadow pattern only
- `elements` (list of dicts): each has `id`, `caption` (concise), `position` (<bbox>...</bbox>), `depth`
      (0-255), optional `photography` dict, plus dynamic attribute keys (material_and_surface, color,
      lighting_interaction, ...) and action keys (pose, gesture, expression, gaze, action)
- `relationships` (list of strings referencing bboxes)
- `scene` (dict): `setting` + `elements` list (same structure as foreground elements)
- `photography` (dict): `layout`, `shot_type`, `camera_angle`, `lens_and_effect`

# Key Rules
1. Output format is `<analysis>...</analysis>` + JSON. No commentary outside these two parts. No markdown
      fences around the JSON.
2. <analysis> is the supervised construction record --- every JSON decision must be traceable to a reasoned
      step inside <analysis>. Stages A->B->C->D->E must each appear as a labeled section.
3. Every object in JSON `elements` or `scene.elements` MUST have been explicitly imagined in Stage D.
4. Pure visual description only --- no subjective commentary or design-intent talk. Light physics, material
      properties, object state, spatial relationships only.
4a. Reason-first: for every decision write reasoning before conclusion. Never tail-justify.
4b. Knowledge-design separation: Stage A holds only facts/conventions, never image-specific design.
4c. Preserve uncertainty: hedge for under-specified attributes; never "must" / "therefore it is" / "the only
      valid" for prompt-silent attributes.
5. Ratio mandatory and first in JSON, from the 9-option menu.
6. `caption` carries identity + single most salient feature; all rich detail lives in named dynamic keys,
      never crammed into `caption`.
7. Dynamic key naming: one key per visual dimension; do not repeat content across `caption` and keys.
8. Field orthogonality: `lighting` = global light source only; `style` = aesthetic label only; `photography` =
      camera/lens tech only; `intent`/`atmosphere` = summary/mood only; per-element detail = the element's
      dynamic keys.
9. Bboxes reflect real-world spatial relationships; edge-cropped objects have bbox touching 0 or 999.
10. Transcribe ALL Stage D detail into JSON, distributed across `caption` + dynamic keys.
11. `<bbox>0 0 999 999</bbox>` ONLY for full-frame backgrounds (sky, floor).
12. No element count limit.
13. Decomposition: complex hybrids (e.g. "centaur") get an element for the whole plus separate elements for
      distinct parts, linked via `relationships`. Explicit quantities ("three cats") -> that many separate
      elements with unique ids.
14. Text rendering: text in image must be in double quotes inside the relevant element's keys with explicit
      font/color.
15. Position specificity: use foreground / mid-ground / distant background, top-left / centered / bottom-right
      ; never vague "next to" without direction.
```

H.2 AESTHETIC JUDGE SYSTEM PROMPT

Listing 2: Aesthetic judge system prompt (verbatim).

```
# Role

You are an AI image aesthetic quality inspector. You will be shown one generated image.

Your ONLY job is to evaluate the visual/aesthetic quality. Do NOT evaluate prompt alignment or
      structural accuracy --- those are handled separately.

**Scoring philosophy: Be critical. A score of 7 means "decent looking". Most AI images score
     5-7.**

## Technical Defects Checklist

- [ ] Overexposure? (blown-out highlights, white patches with no detail)
- [ ] Underexposure? (crushed shadows, dark areas with no detail)
- [ ] Over-saturation? (colors unnaturally vivid, neon-like)
- [ ] Color banding / posterization? (visible color steps instead of smooth gradients)
- [ ] Blurriness where sharpness is expected? (not intentional depth-of-field)
- [ ] Visible artifacts? (compression artifacts, weird halos, noise)
- [ ] Unnatural skin texture? (waxy, plastic-looking, or overly smooth)
- [ ] Inconsistent detail level? (some areas sharp, others blurry for no reason)

## Artistic Merit

- Composition: Is the framing and layout pleasing?
- Lighting: Is the lighting natural and well-used?
- Color harmony: Do colors work well together?
- Overall impression: Would a viewer find this visually appealing?
- Does the aspect ratio work well with the composition?

## Scoring Scale

- 10: Stunning --- gallery quality, excellent technique, zero technical defects
- 8-9: Very appealing, at most one very minor technical imperfection
- 6-7: Looks decent, minor technical issues (slight over-saturation, minor artifacts)
- 4-5: Mediocre --- noticeable technical problems or unappealing composition
- 2-3: Poor --- multiple technical defects, unpleasant to look at
- 0-1: Ugly, chaotic, severe technical failures

## Output

List every aesthetic issue found, then score.

{
  "score": $score,
  "pass": true or false,
  "issues": ["issue 1", "issue 2", ...]
}

**Pass rule:** score >= 6.
```

H.3 COLD-START FILTERING SYSTEM PROMPTS

Validator v4 applies five independent Gemini calls to each candidate trace: prompt–image alignment, introduction quality, justification of imagined details, reverse rationalization, and reasoning-stage boundaries. The accept/reject statistics in Appendix E.5.1 are computed from these five axis-specific outputs using the strict gate described there. We reproduce the five system prompts in full below; typographic dashes, ellipses, and arrows are normalized to ASCII for reliable pdflatex rendering, without changing the wording.

H.3.1 PROMPT–IMAGE ALIGNMENT

```
You are an auditor checking ONE specific issue at the data level: **does the reference image faithfully
↪  depict the user's text prompt?**

# Context

For each training sample we have:
- A short user text prompt describing a desired image.
- A reference image that was supposed to depict that prompt.

We need to filter out samples where the reference image **does not actually depict the user's prompt** --
↪  these mismatched pairs would teach the model wrong associations if used downstream.
```

```
You will be shown only the user prompt and the reference image. You do NOT see any analysis text. You only judge the prompt<->image alignment.

# Your job

Decide whether the image is a reasonable depiction of the prompt. Use these severity levels:

- **aligned**: the image faithfully depicts the prompt's key subjects, action, and setting. Minor differences in styling, framing, or non-named visual choices are fine; the image is clearly an instance of what the prompt describes.
- **partial**: the image captures SOME of the prompt but is missing or contradicts a clearly named element (e.g., prompt says "elderly couple AND grandchildren" but the image shows only the elderly couple; prompt says "with a red hat" but the hat is yellow). Still recognizable as related to the prompt but not a faithful match.
- **mismatch**: the image does not depict what the prompt describes. The main subject is different, the setting is different, or the image is essentially unrelated / a generic scene that doesn't match the prompt.

# What you should NOT flag

- **Differences in artistic style / lighting / framing** if the prompt didn't specify them -- these are creative choices, not mismatches.
- **Extra elements** in the image beyond what the prompt mentioned (e.g., prompt says "cat" and image shows a cat on a sofa -- the sofa is extra but not a mismatch).
- **Image quality issues** (blur, low resolution, compression artifacts) -- those aren't alignment issues.
- **Cultural / aesthetic interpretation** of vague prompts (e.g., prompt: "a beautiful sunset"; almost any nice sunset image is aligned).
- **Minor count errors** if the prompt didn't emphasize an exact count (e.g., prompt: "some balloons" + image with 6 balloons is aligned; but prompt: "exactly 3 balloons" + image with 6 = partial).

# What you SHOULD flag

- **Subject mismatch**: prompt is about a cat, image shows a dog.
- **Action mismatch**: prompt says "running", image shows someone sitting.
- **Setting mismatch**: prompt says "beachside", image shows an indoor scene.
- **Missing named element**: prompt names X explicitly (e.g., "with a red balloon"), image shows no X (or wrong color).
- **Genre mismatch**: prompt is a chart/infographic of X, image is a photograph of something unrelated.
- **Wrong named entity**: prompt names a specific person/place/thing, image shows a different one.

# Output format

Return exactly one JSON object, no markdown fence, no commentary:

```
{
  "alignment_severity": "aligned" | "partial" | "mismatch",
  "missing_or_wrong": [
    "<prompt element that's missing or wrong in the image>", ...
  ],
  "extra_unrelated": [
    "<image element clearly unrelated to prompt -- only if it dominates the scene, otherwise leave empty>",
    ...
  ],
  "reasoning": "<2-4 sentences explaining your alignment judgment, naming what matches and what doesn't>"
}
```

For `aligned`, both lists should be empty (or near-empty). For `partial`, list the specific gaps. For `mismatch`, list the prompt elements absent from the image.
```

H.3.2 INTRODUCTION QUALITY

```
You are an auditor checking ONE specific issue in a reasoning trace: **introduction quality of new visual details**.

# Context -- IMPORTANT to read carefully before judging

The reasoning trace (`<analysis>` block) was produced by a teacher model that was given access to a reference image while planning a structured prompt expansion. **This is by design.** The reference image is a **guiding signal**, like a "standard answer", that helps the teacher reason about plausible visual realizations of the user's text prompt.

**The teacher IS expected to introduce many visual details that go beyond the user's text prompt.** That is the whole point of the task -- to turn a short user description into a richly detailed visual blueprint. Specific clothing, colors, props, lighting, background elements, and accessories are all welcome additions.

**Do NOT flag a span just because it contains specific visual detail that wasn't in the user prompt.** That is not a problem. The user wants rich, detailed analyses.

# Your actual job

The student model that will be trained on these analyses sees the user prompt but not the reference image. So we need each visual addition in the analysis to be **introduced in a way the student can learn to reproduce** -- meaning the detail should be either:

(a) **Framed as a creative / design choice** by the writer ("I'll picture the man in a coral pink polo to add warmth to the family meal"), OR
(b) **Tied to common-sense or canonical knowledge** ("a Persian cat has a flat face and dense fur"), OR
```

```
(c) **Logically implied by the prompt's stated context** ("a kitchen would have a knife block, a fruit
↪  bowl..."), OR
(d) **Listed within a coherent build narrative** that makes its origin clear (e.g., described as part of an
↪  explicit "I'll add ambient clutter such as..." block).

What you should flag is **bald, observational-style assertions** -- details dropped in as if read off an
↪  actual image, without any framing, reasoning, or contextual integration. The pattern looks like a list
↪  of facts that exist only because the writer saw them, presented as if they were objective truths about a
↪  specific instance.

# Examples

**OK -- well-introduced (don't flag):**
- "I'll picture a coral pink polo for the elderly man -- a warm tone fits the joyful family beach setting
↪  and contrasts nicely with the cooler beach background."
- "For the table, I'll choose a crisp white tablecloth; this is typical for casual seaside restaurants and
↪  provides high contrast for the food."
- "Adding ambient props: two orange juice glasses with condensation (typical breakfast/brunch on the beach),
↪  a small white ceramic bowl with sauce."

**Flag -- bald observational assertion:**
- "The elderly man wears a coral pink polo shirt with a metal wristwatch on his left wrist." <- reads as a
↪  literal description of what was seen, no framing.
- "There are exactly 6 inflated round latex balloons: 2 pale lemon yellow, 1 bright golden yellow, 1 coral
↪  red, 1 sky blue, 1 soft pale pink." <- very specific count + color enumeration with no rationale given
↪  for picking these specifics.
- "A unique garlic-shaped porcelain jar sits in the bottom-center, next to a lidded bowl with pickled
↪  cucumbers." <- introduces unique-shape props with no explanation of why these specific items.

The line: did the analysis SHOW THE THINKING behind picking this specific instantiation, or did it just
↪  declare the instantiation?

# Severity rubric

- **none**: every visual addition is either framed as a choice, tied to common-sense, or comes embedded in a
↪  coherent build narrative that lets a student understand WHY it's there.
- **minor**: 1-3 details are dropped in bare without framing, but most additions are properly motivated.
- **major**: the analysis reads largely as observational dump -- long lists of specific concrete details
↪  with no framing or reasoning, as if the writer were dictating a literal description.

When in doubt, lean LESS strict. A coherent build narrative (e.g., "the scene has a wooden table, white
↪  tablecloth, plates with food, glasses of orange juice...") is fine even without a "because" attached to
↪  every item -- the narrative flow itself counts as framing.

# Output format

Return exactly one JSON object, no markdown fence, no commentary:

~~~
{
  "introduction_severity": "none" | "minor" | "major",
  "introduction_spans": [
    {"span": "<verbatim quote, 5-30 words>", "why_bare": "<why this reads as observational rather than
    ↪  reasoned/framed>"},
    ...
  ],
  "reasoning": "<2-4 sentences explaining your judgment of the overall introduction quality, including
  ↪  whether the analysis as a whole reads as motivated reasoning or as a literal description>"
}
~~~

If `introduction_severity` is "none", `introduction_spans` should be an empty list.
```

H.3.3 JUSTIFICATION OF IMAGINED DETAILS

```
You are an auditor checking ONE specific issue in a reasoning trace: **specific imagination claims made
↪  without an accompanying reason**.

# Context -- IMPORTANT to read carefully

The reasoning trace (`<analysis>` block) was produced by a teacher model that's expected to **imagine**
↪  visual specifics beyond what the user prompt literally says. Imagination is **wanted** -- the teacher's
↪  job is to flesh out a short prompt into a rich blueprint.

What is **NOT acceptable** is asserting specific imagined attributes <b>without stating the reason</b> for
↪  that imagination. The student model trained on these analyses learns from the reasoning patterns -- if a
↪  specific attribute appears without an attached "because...", the student learns to assert random
↪  specifics without justification, which is a hallucination behavior we want to avoid.

# Your job

For each specific imagined claim in the analysis that goes beyond what the user prompt literally specifies,
↪  check whether the analysis **states a reason** for that specific instantiation.

A "reason" can be:
- (a) **Tied to the user prompt's language** ("because the prompt's 'cozy bistro' setting suggests rustic
↪  materials, I'll picture wooden tables")
- (b) **Tied to canonical / common-sense knowledge** ("Persian cats typically have flat faces and dense fur
↪  -- so I'll picture exactly that")
- (c) **Explicit creative motivation** ("I'll choose a coral pink polo to add visual warmth to the family
↪  meal")
```

```
- (d) **Logical implication from named context** ("a beachside restaurant typically has wooden tables, paper
↪  menus, salt-air haze")

What's **NOT a reason**:
- Just declaring the specific (no "because", "to", "for", "since", etc.): "The bag is opened, with a tear on
↪  the right side." <- naked assertion, FLAG.
通常必须- Using "typically / must / always / 通常 / 必须" in a way that asserts the specific as a universal
↪  fact when it's actually a choice: "The man **must** be wearing a coral polo" <- FLAG.

# Examples

**OK (don't flag) -- claim with reason:**
- "I'll picture the bag opened with a small tear, **because** the prompt's casual snacking vibe of haribo
↪  passport mix suggests an in-progress, lived-in moment."
- "The table is wooden -- **a beachside restaurant typically has** rustic wood tables that withstand salt
↪  air."
- "Adding a small dollop of mayo to the salsa **for visual contrast** with the red base."
- "Persian cats **have** flat faces and dense fur **so** I'll picture exactly that." (canonical knowledge
↪  cited)
- "The lighting is warm afternoon, **chosen to** match the cozy family-meal mood of the prompt."

**Flag -- claim without reason:**
- "The bag is opened, with a small tear on the right side." <- Why opened? Why tear on right?
- "The polo is coral pink with three buttons." <- Why coral? Why three?
- "There are 6 balloons: 2 pale lemon yellow, 1 bright golden yellow, 1 coral red, 1 sky blue, 1 soft pale
↪  pink." <- Why these exact 6 and these specific colors?
- "The woman is in her 60s, with grey curly hair and round sunglasses." <- Why this age, why curly, why
↪  round glasses?
- "The man **must** be wearing brown leather shoes." <- uses "must" but no underlying reason.

# What does NOT count as a flag

- **Decisions directly named in the prompt**: prompt says "red apple" -> analysis says "the apple is red" --
↪  no reason needed because prompt already gave it.
- **Stage A pure-knowledge claims** that aren't about THIS image: "Persian cats have flat faces" stated as a
↪  general fact in Stage A is knowledge, not an imagination claim about a specific cat in the scene.
- **Stage E bbox coordinates**: bbox numbers are by-nature decisions; no per-coordinate "because" needed.
- **Tool-derived facts** if web_search / wikipedia_search was used.

# Severity rubric

- **none**: every specific imagination claim is accompanied by a stated reason (prompt-tied / canonical /
↪  creative / contextual).
- **minor**: 1-3 specific claims appear without a reason, but most claims are properly motivated. Common in
↪  long analyses where a few details slip through.
- **major**: many specifics appear naked. The analysis pattern reads as "here's a list of things I'll
↪  include" rather than "here's why I'm including these things." A student trained on this would learn to
↪  hallucinate without justification.

When in doubt, lean LESS strict. A coherent build narrative ("I'll add: wooden table, white tablecloth,
↪  plates, glasses of orange juice") with a brief upstream framing ("for a casual beachside lunch") counts
↪  the brief framing as covering the listed items.

# Output format

Return exactly one JSON object, no markdown fence, no commentary:

```
{
  "hallucination_severity": "none" | "minor" | "major",
  "unsupported_spans": [
    {"span": "<verbatim quote, 5-30 words>", "why_unsupported": "<which specific claim is asserted without a
    ↪  reason>"},
    ...
  ],
  "reasoning": "<2-4 sentences explaining your judgment, noting whether the analysis predominantly
  ↪  accompanies imagined specifics with reasons or just lists them>"
}
```

If `hallucination_severity` is "none", `unsupported_spans` should be empty.
```

H.3.4 REVERSE RATIONALIZATION

```
You are an auditor checking ONE specific issue in a reasoning trace: **reverse rationalization** (decisions
↪  stated first, justifications back-filled afterwards).

# Context

A good reasoning trace builds the visual blueprint from prior evidence: it (a) acknowledges what the prompt
↪  says, (b) acknowledges what's unspecified, (c) cites world knowledge or creative reasoning, and only
↪  then (d) makes a specific visual choice. Reverse rationalization happens when the trace asserts a
↪  specific concrete choice first and then back-fills a "because..." -- making the prior reasoning a
↪  cosmetic decoration rather than the actual basis for the decision.

# Your job

Given the analysis text, identify spans where the teacher gives a **specific concrete decision first** and
↪  then **back-fills a justification** -- except when the decision is directly named in the user prompt (in
↪  which case the justification IS just elaboration, and that's fine).

# What counts as reverse rationalization
```

```
- "The lighting is warm golden-hour because..." where neither prompt nor context specifies "warm
↪  golden-hour".
- "I'll use a 3:2 aspect ratio because... [reasoning]" placed BEFORE the reasoning that should derive 3:2.
↪  (The order should be: analyze spatial demands -> therefore 3:2.)
- Listing a "build order" but the order is just the final element list in disguise (no actual reasoning
↪  about why this order).
- Stage A starts with "this scene should be cinematic photorealism" -- Stage A should be common-sense
↪  knowledge, not Stage D's stylistic decisions.

# What does NOT count

- Decisions directly named in the prompt (e.g., prompt says "warm golden-hour"; the analysis saying "warm
↪  golden-hour because..." is just elaboration, not back-fill).
- Quick declarative summaries that **also** show real upstream reasoning elsewhere (decision + brief
↪  restatement is fine if Stage A/B genuinely derives it).
- Concrete numbers in Stage E (bbox coords) -- these by nature are decisions, and Stage E's job is to plan
↪  them, not derive them from first principles.

# Severity rubric

- **none**: every specific decision is either prompt-named, or genuinely derives from upstream reasoning.
- **minor**: 1-3 decisions appear out-of-order with their justification, but the overall flow is coherent.
- **major**: the analysis is mostly decisions-first, reasoning-after; or the "reasoning" never actually
↪  justifies the decisions made.

# Output format

Return exactly one JSON object, no markdown fence, no commentary:

```
{
  "reverse_severity": "none" | "minor" | "major",
  "reverse_spans": [
    {"span": "<verbatim quote ~5-30 words>", "evidence": "<what makes this reverse-rationalized rather than
    ↪  reasoned>"},
    ...
  ],
  "reasoning": "<1-3 sentences explaining your overall judgment>"
}
```

If `reverse_severity` is "none", `reverse_spans` should be empty.
```

H.3.5 REASONING-STAGE BOUNDARIES

```
You are an auditor checking ONE specific issue in a reasoning trace: **stage boundary violations** (content
↪  put in the wrong stage of the analysis).

# Context

The `<analysis>` block has 5 stages with distinct purposes:
- **Stage A -- Knowledge & common-sense**: world facts, canonical depictions, cultural/aesthetic norms tied
↪  to style words. NOT specific decisions about THIS image.
- **Stage B -- Reasoning chain (user words -> visual decisions)**: trace "user said X -> I picture Y because
↪  Z". The chain itself, not the final pixels.
- **Stage C -- Aspect ratio**: justify the chosen aspect ratio.
- **Stage D -- Compose & imagine**: the actual build-from-nothing element-by-element description with
↪  materials, colors, light interaction, positions. The "heart" of the analysis.
- **Stage E -- Layout & spatial constraints**: bbox planning, aspect-ratio compensation, perspective check.

# Your job

Given the analysis text, identify content that's in the wrong stage.

# Common boundary violations

- Stage A contains **specific decisions for THIS image** (e.g., "the man's polo will be coral pink" --
↪  that's Stage D, not Stage A common-sense).
- Stage D contains **generic world facts not bound to a specific element in this scene** (those belong in
↪  Stage A).
- Stage E contains **new element descriptions** (any new element introduction belongs in Stage D; Stage E
↪  should only assign bboxes to elements already described).
- Stage B is missing or just repeats Stage A.
- Stage C reasoning relies on details that should have been imagined in Stage D.

# What does NOT count

- Brief restatements (one phrase) that bridge stages -- OK as connective tissue.
- Stage D referring back to "as established in Stage A" -- fine.
- Slight overlap is normal; only flag clear misplacements.

# Severity rubric

- **none**: each stage's content fits its purpose.
- **minor**: 1-2 short misplaced sentences, but the overall stage structure is intact.
- **major**: stages are fundamentally confused (e.g., Stage A is full of THIS-image-specific decisions;
↪  Stage D barely describes anything new).

# Output format

Return exactly one JSON object, no markdown fence, no commentary:
```

```
```
{
  "boundary_severity": "none" | "minor" | "major",
  "misplaced_spans": [
    {"span": "<verbatim quote ~5-30 words>", "currently_in": "A|B|C|D|E", "should_be_in": "A|B|C|D|E",
    ↪ "why": "..."},
    ...
  ],
  "reasoning": "<1-3 sentences explaining your overall judgment>"
}
```

If `boundary_severity` is "none", `misplaced_spans` should be empty.
```

H.4 VLM-AS-JUDGE SYSTEM PROMPTS

The *Offline Judge* columns of Table 2 are the outputs of two independent GPT-5.4 calls, each governed by a dedicated system prompt that scores a single axis on 0–10. Both judges receive the original user prompt and generated image, but apply different rubrics: the *Structure* judge focuses on anatomy, counts, layout, and physical plausibility, whereas the *Alignment* judge focuses on whether the image satisfies the prompt. Each judge returns a JSON object `{"score": s, "pass": true/false, "issues": [...]}`. For each row of Table 2 we report the per-axis mean score over the evaluation pool of 150 user prompts. The two system prompts are reproduced verbatim below for reproducibility.

H.4.1 STRUCTURE

Listing 3: Structure judge system prompt (verbatim).

```
# Role

You are an EXTREMELY strict AI image structural inspector. Your job is to find ANY structural
    or layout problem. If there is even ONE clear structural issue, the image FAILS.

You will be shown:
1. The original user prompt
2. The generated image

**Zero tolerance policy: ANY clearly visible structural defect = FAIL. Do not give the benefit
     of the doubt.**

## Mandatory Inspection --- Go Through EVERY Category

### Humans / Characters
- [ ] Count fingers on EVERY visible hand. Exactly 5 per hand? If you cannot clearly see all
    fingers, note this.
- [ ] Are ALL limbs attached naturally? No extra arms, missing legs, disconnected body parts?
- [ ] Are faces symmetrical and natural? No melted features, misaligned eyes, distorted mouths
    ?
- [ ] Are body proportions correct? (head size vs body, arm length, torso proportions)
- [ ] Are hands and feet well-formed? (not blobs, not fused, not unnaturally bent)
- [ ] For multiple people: are they properly separated? No merged/fused bodies?

### Animals
- [ ] Correct number of legs, wings, tails?
- [ ] Natural body proportions and posture?
- [ ] No distorted or melted features?

### Objects
- [ ] Are objects structurally COMPLETE? (no half-formed items, no objects fading into nothing
    )
- [ ] Are objects physically GROUNDED? (not floating unnaturally, not hovering without support
    )
- [ ] Are objects properly SEPARATED? (no clipping/interpenetration --- objects merging into
    each other)
- [ ] Do objects obey basic physics? (gravity, support, balance)
- [ ] Are objects the correct relative SIZE? (a person shouldn't be the same height as a
    building in the foreground)

### Layout / Composition
- [ ] Is perspective CONSISTENT? (vanishing points, scale at different distances)
- [ ] Are shadows and lighting consistent across the ENTIRE scene?
- [ ] Are edges clean? (no halos, smudges, blended boundaries between objects)
- [ ] Is the aspect ratio appropriate? Are objects squished, stretched, or distorted by the
    ratio?
```

```
- [ ] Are there any spatial impossibilities? (object behind something but rendered in front,
     etc.)

### Text in Image (if present)
- [ ] Is text legible?
- [ ] Are letters correctly formed? (no mirrored, extra, or missing letters)
- [ ] Is text properly integrated into the scene?

## Scoring Scale --- STRICT

- 10: Absolutely flawless --- zero structural issues of any kind (extremely rare)
- 8-9: Near perfect --- at most one TINY issue (e.g., slightly odd fingernail) that requires
     zooming in to notice
- 6-7: Minor issues present but not distracting (e.g., one slightly unnatural finger joint, a
     small edge artifact)
- 4-5: Clear structural problems visible at normal viewing (e.g., extra finger, minor floating,
      warped face)
- 2-3: Severe problems (e.g., melted face, extra limbs, objects merging, major floating)
- 0-1: Structurally incoherent, main subject unrecognizable

## Output

List EVERY structural issue you found, no matter how small. Then score.

{
  "score": $score,
  "pass": true or false,
  "issues": ["issue 1", "issue 2", ...]
}

**Pass rule:** score >= 6. But if there is ANY clearly visible structural defect at normal
     viewing distance (extra fingers, floating objects, merged bodies, broken limbs), the
     score MUST be <= 5 and pass MUST be false.

**Reminders:**
- If you see ANY hand, you MUST count fingers.
- "Looks fine at first glance" is NOT enough. Inspect every detail.
- A structurally flawed image with score 4 is more useful than a generous 7.
- When in doubt, FAIL. We are building training data --- false positives are expensive.
```

H.4.2 ALIGNMENT

Listing 4: Alignment judge system prompt (verbatim).

```
# Role

You are an AI image-prompt alignment inspector. You will be shown:
1. The original user prompt
2. The generated image

Your ONLY job is to check whether the image faithfully represents what the prompt asked for.

**Scoring philosophy: Be critical. A score of 7 means "good with minor flaws". Most AI images
     score 5-7.**

## Mandatory Checklist

Go through EVERY item:

1. **Count subjects:** Does the prompt specify a number? ("two dogs" = exactly 2, "a cat" =
     exactly 1). Count what's in the image.
2. **Primary subjects:** Are ALL main characters/objects/scenes present?
3. **Secondary subjects:** Are background elements, accessories, secondary objects present?
4. **Spatial relationships:** Are left/right, above/below, in front/behind correct? (from
     viewer's perspective)
5. **Colors:** Are all specified colors correct?
6. **Style/medium:** Is the specified style matched? ("watercolor", "anime", "photorealistic",
      "3D render", etc.)
7. **Contradictions:** Are there elements that CONTRADICT the prompt? (wrong gender, wrong
     action, forbidden elements)
8. **Text rendering:** If text is required, is it correct and legible?
9. **Actions/poses:** Are specified actions being performed correctly?
10. **Mood/atmosphere:** Does the overall mood match? ("dark and moody", "bright and cheerful",
      etc.)

**The image MAY contain extra elements not in the prompt --- this is acceptable. But it MUST
     NOT violate, contradict, or omit what the prompt explicitly requires.**

## Scoring Scale
```

```
- 10: Every single requirement perfectly satisfied, no contradictions
- 8-9: All primary requirements met, one minor secondary element slightly off
- 6-7: Primary requirements mostly met, some secondary requirements missing or slightly wrong
- 4-5: Some primary requirements met, but noticeable mismatches or omissions
- 2-3: Major primary requirements violated or missing
- 0-1: Image has almost nothing to do with the prompt

## Output

List every alignment issue found, then score.

{
  "score": $score,
  "pass": true or false,
  "issues": ["issue 1", "issue 2", ...]
}

**Pass rule:** score >= 6. Any primary subject missing or contradicted = automatic fail.
```

H.5 GSB PAIRWISE PREFERENCE PROTOCOL

GSB compares a candidate image with a fixed reference image generated from the same user prompt. The offline GPT-5.4 judge scores structural accuracy and text–image matching under the pairwise rubric reproduced below. To remove sensitivity to image position, every candidate–reference pair is evaluated twice: once with the candidate as Image A and once with the candidate as Image B. After restoring candidate identity, the pair is recorded as *Good* if both calls prefer the candidate, *Bad* if both prefer the reference, and *Same* if the two orderings disagree. We report the net preference

$$\mathrm{GSB} = 100\,\frac{n_{\mathrm{Good}} - n_{\mathrm{Bad}}}{N}, \tag{7}$$

where N is the number of candidate–reference pairs and Same contributes zero. Tables 17–18 all score one image per prompt over the 150-prompt evaluation pool ($N = 150$). Requiring agreement across the two presentation orders removes judgements that are sensitive to the A/B image position from the net preference. Tables 17–18 use the zero-shot, single-shot Qwen3.5-397B-A17B Base prompter as the reference. Figure 1 uses a different reference by design: each system is scored against *its own* output at the shortest caption rung, so every curve starts at zero and the panel measures gain from lengthening the caption rather than quality relative to a common system. The source prompt is `data/gemini_pairwise_judge_en.txt`; the copy below normalizes typographic punctuation to ASCII for reliable pdfLaTeX rendering without changing the wording.

```
# Role

You are a professional AI image generation quality evaluation expert. Please follow the steps below to
↪  comparatively assess the quality of two AI-generated images and determine which one is better.

**Step 1: Provide evaluation principles**
This evaluation must strictly focus on the following two core principles. The weight of each principle may
↪  be adjusted slightly depending on the task.

**Principle 1: Structural Accuracy** (recommended weight: 50%)
**Scoring description:** This principle evaluates whether the image itself contains errors or unnatural
↪  issues in details, structure, or perspective. You need to carefully inspect the people, objects, text,
↪  and other elements in the image, and determine whether there are distortions, structural collapse, or
↪  unnatural appearances. Structural Accuracy is scored on a scale of 0-10, where 0 means the structure is
↪  completely incorrect and the main subject is unrecognizable, while 10 means the structure is clear, all
↪  main subjects are free of distortion or collapse, and the image quality is sharp and clear.
**Common bad cases:** floating objects, incomplete objects, disproportionate objects, layouts that violate
↪  common sense, and object nesting/interpenetration. For abstract paintings or fantasy creatures, the
↪  structural standard may be relaxed, but the image must still remain logically self-consistent.

**Principle 2: Text-Image Matching** (recommended weight: 50%)
**Scoring description:** This principle evaluates whether the image content matches the prompt text
↪  description. Matching is scored on a scale of 0-10, where 0 means none of the required points are
↪  satisfied, and 10 means all required points are perfectly satisfied. You need to break down the prompt
↪  from the user's perspective into **primary requirements** and **secondary requirements**, and score
↪  according to how well each requirement is satisfied. Please note that, for the sake of aesthetics and
↪  naturalness, the image may contain reasonable extensions beyond the prompt, but **any content that
↪  contradicts the prompt instructions or omits either primary or secondary requirements must result in
↪  score deductions**.
**Common bad cases:** incorrect logical relationships such as gender, quantity, or orientation (defaulting
↪  to the photographer/viewer's perspective); text that is required by the prompt but is rendered as
↪  gibberish.

**Step 2: Score each image item by item**
For each image, score it according to the following process:
```

```
1. Explain the reason for the score in each dimension
2. Give an integer score from 0 to 10 for each dimension
3. Calculate the weighted score for each dimension (score x weight)

**Step 3: Compare total scores**
(Single dimension score = score x weight; total score = sum of all weighted dimension scores)

**Do not show the calculation process. After the total score, directly output the score only. For example,
↪  output "Image A Total Score: 3.0 points". Do not output formulas or intermediate calculations. Even if
↪  the score is an integer, it must still be written in decimal form, e.g. 1 point must be written as 1.0
↪  points.**
Format:
Image A Total Score: $x points
Image B Total Score: $y points

(For example:
Image A Total Score: 3.67 points
Image B Total Score: 3.0 points)

**Step 4: Output the conclusion**
**Strictly follow this format for the conclusion:**
"Image A loses to Image B" or "Image A beats Image B"

**Important notes:**

* If there is a tie (for example, the total scores are the same), you must additionally explain the
↪  tie-breaking basis.
* You must verify the calculation result.
* You must clearly point out the specific types of issues in the images (for example:
↪  clipping/interpenetration, disproportion, artifacts, blurry image, missing prompt requirements, etc.).

<General Evaluation Standards>
1. **The core criterion is the user's viewing experience**, meaning the evaluation should consider what kind
↪  of experience a user would have when seeing the generated image. The 10-point scale should reflect the
↪  user's satisfaction with that experience.
2. **User experience depends primarily on whether the primary requirements are satisfied**, so when
↪  determining whether a sub-dimension deserves bonus or penalty points, you should first consider whether
↪  that dimension serves the core instruction.

**Key concept definitions:**

* **Primary Need:** The core subject, scene, action, and composition requirements in the user's prompt.
* **Secondary Need:** Modifying elements, secondary objects, and **style-related terms** in the prompt (such
↪  as "anime," "ink painting," "3D rendering"). Unlike technical parameter words (such as "4k"), style
↪  words must be treated as secondary needs that require matching.
* **Mismatch:** The image fails to correctly reflect the logic described in the prompt. Common bad cases
↪  include incorrect logical relationships in gender, quantity, and orientation (defaulting to the
↪  photographer/viewer's perspective), as well as gibberish text when the prompt requires specific text.
* **Structural Error:** Illogical, unnatural, or distorted elements appearing in the image. Common bad cases
↪  include clipping/interpenetration of objects or characters, extra or missing limbs, disproportionate
↪  objects or characters, and objects violating physical laws. For abstract paintings or fantasy creatures,
↪  the standard for structure may be relaxed, but they must still be logically self-consistent.
* **Factual Error:** For famous people, landmarks, or other subjects with widely recognized appearances, the
↪  generated result seriously violates the public's basic understanding of them.
</General Evaluation Standards>
```

I PROMPTS FOR FIGURES

For reproducibility, this section lists every retained exact user prompt behind the qualitative examples in the main paper; prompts that were not preserved are marked unavailable rather than reconstructed. Each subsection follows the order in which the corresponding figure is first referenced.

I.1 FIGURE 16 — GALLERY

The qualitative gallery in Figure 16 is manually assembled from 14 outputs of our final SP/L10 + RFT-prompter Qwen-Image system (raster order by tile index). Available per-tile user prompts are reproduced verbatim below (item i = tile i); where the original request was not retained, we mark the prompt record as unavailable rather than reconstructing it after the fact.

1. *"Photorealistic telephoto photograph in a dark room lit by a single warm desk lamp aimed at a whiteboard, deep shadows swallowing the rest of the room, a bright pool of light falling across the writing, sparse handwriting in black marker, high contrast, visible grain. The board reads: 'It is not only the model that scales — caption information scales too. MSE* $= 0.4549 - 8.45\text{e}{-5}\cdot\text{GPG}$ *($r = -0.984$); MSE* $= 0.4200\cdot\text{ED}^{-0.207}$ *($r = -0.971$).' "*
2. *"A monkey is making latte art."*
3. *"Poster for an astronomy exhibition: capture galaxies swirling in deep space against a cosmic backdrop, using rich velvety textures and star clusters that shimmer with ethereal light play."*

4. *(Final-system output; original user prompt not retained.)*
5. *"Creating a poster featuring a chubby little black guy driving a van full of gas cylinders."*
6. *"Create a brand poster that captures innovative energy."*
7. *"A cowboy leans against the back of an old pickup truck. Two women stand in the truck bed. The image is styled like an advertisement."*
8. *"A children's book illustration drawn with colored pencils: A curious Husky stretches its paw toward a person the size of a mouse."*
9. *"Develop botanical-themed wedding invitations with layered floral cutouts against ivory linen paper, complemented by hand-lettered calligraphy details and a discreet wax seal closure."*
10. *"Produce a vintage photography competition poster showcasing antique cameras amid scattered film reels under warm studio lighting."*
11. *(Final-system output; original user prompt not retained.)*
12. *"A robot is driving while waving ahead, and a giant snail is sitting in the passenger seat."*
13. 春节庙会、龙灯、民俗表演、人群熙熙攘攘、节日气氛、传统文化、高清大画面 *(Spring Festival temple fair — dragon lanterns, folk performances, a bustling festive crowd, high-resolution wide shot.)*
14. *(Final-system output; original user prompt not retained.)*

I.2 Figure 16 (bottom) — Zero-shot SP editing

Each editing example starts from the base user prompt below; a targeted SP edit is then applied and the scene re-rendered. Most edits change one field, while a move may update the position fields of both the moved object and its spatial counterpart.

1. *"Three books, a coffee mug, and 6 pens scattered on the cluttered study desk."*
2. 深夜咖啡馆内，暖黄灯光下的木质吧台，吧台上放着一杯热可可，杯口冒热气，画面需温馨柔和，不要出现其他客人，不添加任何品牌标志

I.3 Figure 2 — Teaser

The teaser reconstructs a single held-out reference image from captions of each kind at four richness levels (L5/L6/L8/L10), in a natural-language (NL) arm and a structured-prompt (SP) arm. The NL arm follows the matched-control construction of Appendix E.2: it uses the same source evidence as the SP arm and is generated directly rather than flattened from the final JSON. Across the four displayed levels, it preserves the same source entities and relationships while meeting progressively larger token budgets through elaboration and connective phrasing. All four versions cover the same 39 entities and 11 relationships, while reconstruction similarity remains nearly flat as the captions lengthen (Figure 2). The BAGEL scaling-property sweep uses the L6/L8/L10 NL configurations, while L5 supplies the sparser visual endpoint in this probe.

The full NL and SP caption text for all four levels is lengthy; we provide it in the supplementary material rather than inline.

I.4 Figure 9 — SOTA qualitative comparison

The per-row user prompts are printed within Figure 9 itself; the main text shows two of the four rows, and Figure 17 shows all four.

I.5 Figure 10 — Prompter training-stage progression

Rows top to bottom (same Qwen-Image backbone throughout; only the prompter changes across training stages):

1. *"Design a minimalist café poster with the headline "Morning Brews, Gentle Starts" against soft sunrise hues, including small latte art details, written in English."*
2. *"Produce a vintage photography competition poster showcasing antique cameras amid scattered film reels under warm studio lighting."*

I.6 FIGURE 18 — SFT VS. NO-SFT

Rows top to bottom (each rendered before and after prompt-to-SP SFT):

1. *"An elegant pair of glasses with a unique, gold hexagonal frame laying on a smooth, dark wooden surface. The thin metal glints in the ambient light, highlighting the craftsmanship of the frame. The clear lenses reflect a faint image of the room's ceiling lights. To the side of the glasses, a leather-bound book is partially open, its pages untouched."*
2. *"A man dressed in a crisp white shirt and sleek black tie is seated with a guitar in his hands. He is focused intently on the strings, fingers positioned to strum a chord. The room around him is blurred, emphasizing the musician and his instrument as the central subjects of the scene."*

I.7 FIGURE 20 — CASE TYPES THE AGENTIC LOOP RESOLVES

The three case types (structure, element granularity, and full re-planning; §3.2), in figure order:

1. *"A wooden table with 4 stacked items on it: bottom layer a thick textbook, middle layer a closed laptop, then a coffee mug on the laptop, then a single red apple on top of the mug."*
2. *"A formal choir performance with about 60 singers in 4 horizontal rows, each row at different heights on bleachers. All wearing matching black robes. The conductor in front."*
3. *"A robot is driving while waving ahead, and a giant snail is sitting in the passenger seat."*

I.8 FIGURE 21 — PROMPTER COMPARISON PROMPTS

The user prompts behind the prompter-comparison gallery, grouped by figure and listed top to bottom. The Hitman request intentionally appears in two consecutive rows of the first panel because the figure retains two separate comparison cases for the same user prompt.

Figure 21 (1/3), top to bottom:

1. *"Ultra-high-resolution 16:9 typography travel poster of AHMEDABAD, INDIA. Giant bold sans-serif word "AHMEDABAD" centered across the poster, each letter containing different flat vector scenes of Ahmedabad — Sabarmati Riverfront, Atal Bridge, heritage pol houses, Adalaj Stepwell, Jama Masjid, metro train, auto-rickshaws, kite festival, modern skyline, temples, and street life. Letters act like architectural gallery windows with connected urban panorama. Thin panoramic strip at top with skyline silhouettes, metro, cars, birds, river bridge, boats, clouds, and warm sun. Mid-century modern Swiss graphic design, minimal vector illustration, architectural infographic aesthetic, retro travel poster branding, flat geometric shapes only, no realism, no gradients, clean vector edges, strong negative space, editorial layout, museum gift shop aesthetic. Muted Ahmedabad-inspired palette: dusty teal, terracotta, sand beige, cream, olive, burnt orange. Soft ivory background, premium typography, perfectly spelled English text, ultra-clean composition, print-ready 8K quality, no AI artifacts, no distorted text."*
2. *"Screenshot of the YouTube homepage in 2030"*
3. *"A Hitman level where you are in the OpenAI HQ and your mission is to steal GPT-6 without getting caught."*
4. *"A Hitman level where you are in the OpenAI HQ and your mission is to steal GPT-6 without getting caught."*
5. *"1. A vibrant fusion street-food scene where a sizzling plate of smoky fried rice biryani blends aromatic spices with golden grains, beside a rich, slow-cooked mutton dish glistening with gravy. Steaming hot momos sit in a bamboo basket, releasing curls of fragrant steam, while a chilled mint mojito sparkles with ice, fresh mint leaves, and lime slices. The setting is a lively night market under warm lights, with colors, textures, and aromas colliding into a bold, modern culinary fusion aesthetic. 2. An eye-catching food scene featuring a delicious spread of fast food and refreshing drinks: a juicy pizza with melted cheese, a stacked burger with crispy lettuce and sauce, a bowl of steaming noodles, golden crispy fries, a chilled mojito with mint and lime, and fresh slices of juicy watermelon. The setting is vibrant and colorful, with soft lighting, high detail, and a modern aesthetic, arranged beautifully on a wooden table, top-down view, ultra-realistic, 4K quality."*
6. *"An overhead flat-lay food photograph of a brunch spread on a rustic wooden table. Items include: avocado toast with a poached egg, a bowl of acai topped with granola and berries, a cup of pour-over coffee, fresh orange juice in a glass carafe, scattered linen napkins, and small potted succulents. Natural daylight from a window on the left, soft shadows, warm inviting tones."*

7. *"Generate an image of a handwritten traditional Chinese medicine prescription"*

Figure 22 (2/3), top to bottom:

1. *"Full-body fashion editorial of a confident model sitting casually on a concrete ledge, relaxed pose, direct gaze. Wearing denim jacket, white t-shirt, neutral shorts, bright socks, sneakers. Wind adds subtle motion. Beside them, a bold cartoon dragon (thick outlines, neon blue/green/yellow, playful yet powerful) interacts naturally. Bright urban outdoor setting, blue sky, strong sunlight, crisp shadows. Mixed-media style blending photorealism and illustration with doodles, arrows, and motion graphics. High contrast, HDR, ultra-detailed, 8K."*
2. *"High-detail anime character reference sheet, premium fantasy RPG character design board, elegant blue-and-white oceanic aesthetic, Japanese fantasy anime style, highly polished gacha game presentation, cinematic concept art layout"*
3. *"Create an epic poster showcasing the most iconic moments of Michael Jordan career. epic, cinematic, lens flare"*

Figure 23 (3/3), top to bottom:

1. 重新生成一张海报，卓别林拿着止痒膏,面露微笑。风格要简约干净。
2. *"Su Shi's first day of exile Xiaohongshu screenshot"*
3. *"Style: A screenshot of an Albedo cosplay Instagram story photo; Content: Squatting on the ground facing the camera, both hands making exaggerated rebellious gestures, rolling eyes, arrogant and disdainful expression."*

I.9 FIGURES 24–25 — SOTA LLM AS PROMPTER

The shared user prompts behind the different-LLM-as-prompter comparison, listed top to bottom.

Figure 24 (1/2):

1. *"A white rabbit in a blue tracksuit is racing a turtle dressed in a red vest. The finish line is within sight, and the turtle has pulled ahead of the rabbit."*
2. *"At a fork in the road, two girls stand on each branch, walking off in different directions."*
3. *"A cowboy leans against the back of an old pickup truck. Two women stand in the truck bed. The image is styled like an advertisement."*
4. *"Two convertible sports cars drive side by side on the street. The pink one carries two girls; the blue one carries one boy."*
5. *"A giant bear and a donkey play on a seesaw. The donkey is much heavier than the bear."*
6. *"90s + point-and-shoot camera quality"*
7. *"There is one glass, two bottles of red wine, and three cans of beer."*
8. *"A felt figurine of the Hulk, a PVC figurine of Son Goku from Dragon Ball, and a metal figurine of Snow White."*
9. *"A pineapple has one bottle of beer on its left and two on its right."*
10. *"A children's book illustration drawn with colored pencils: A curious Husky stretches its paw toward a person the size of a mouse."*

Figure 25 (2/2):

1. *"An avocado sits on a therapist's chair, with a hole the size of its pit in its center. The therapist is a spoon sitting on a chair, scribbling notes hastily."*
2. 组织管理金字塔结构
3. 小米手机的新品发布会海报
4. 重新生成一张海报，卓别林拿着止痒膏,面露微笑。风格要简约干净。
5. *"Su Shi's first day of exile Xiaohongshu screenshot"*

J ADDITIONAL QUALITATIVE EXAMPLES

J.1 PROMPTER COMPARISON: PROMPTER SCALE × TRAINING

Figures 21–23 compare four prompter variants on a shared set of user prompts, with the same Qwen-Image backbone and decoding settings throughout; only the prompter changes. Two prompter scales (Qwen3.5-35B-A3B and Qwen3.5-397B-A17B) are each shown before (*base*) and after our SFT + Cold-start + RFT pipeline (§2.4). Reading left to right within a scale isolates the effect of training; reading across scales isolates prompter size. Each row corresponds to one user prompt. Appendix I.8 lists the full prompts in the same top-to-bottom order.

J.2 GENERAL-PURPOSE LLMS AS PROMPTERS

The five columns are GPT-5.5, Gemini 3 Pro, GLM-5.2, and Claude Opus 4.8 in single-turn schema-filling mode, followed by our trained Qwen3.5-397B-A17B prompter.

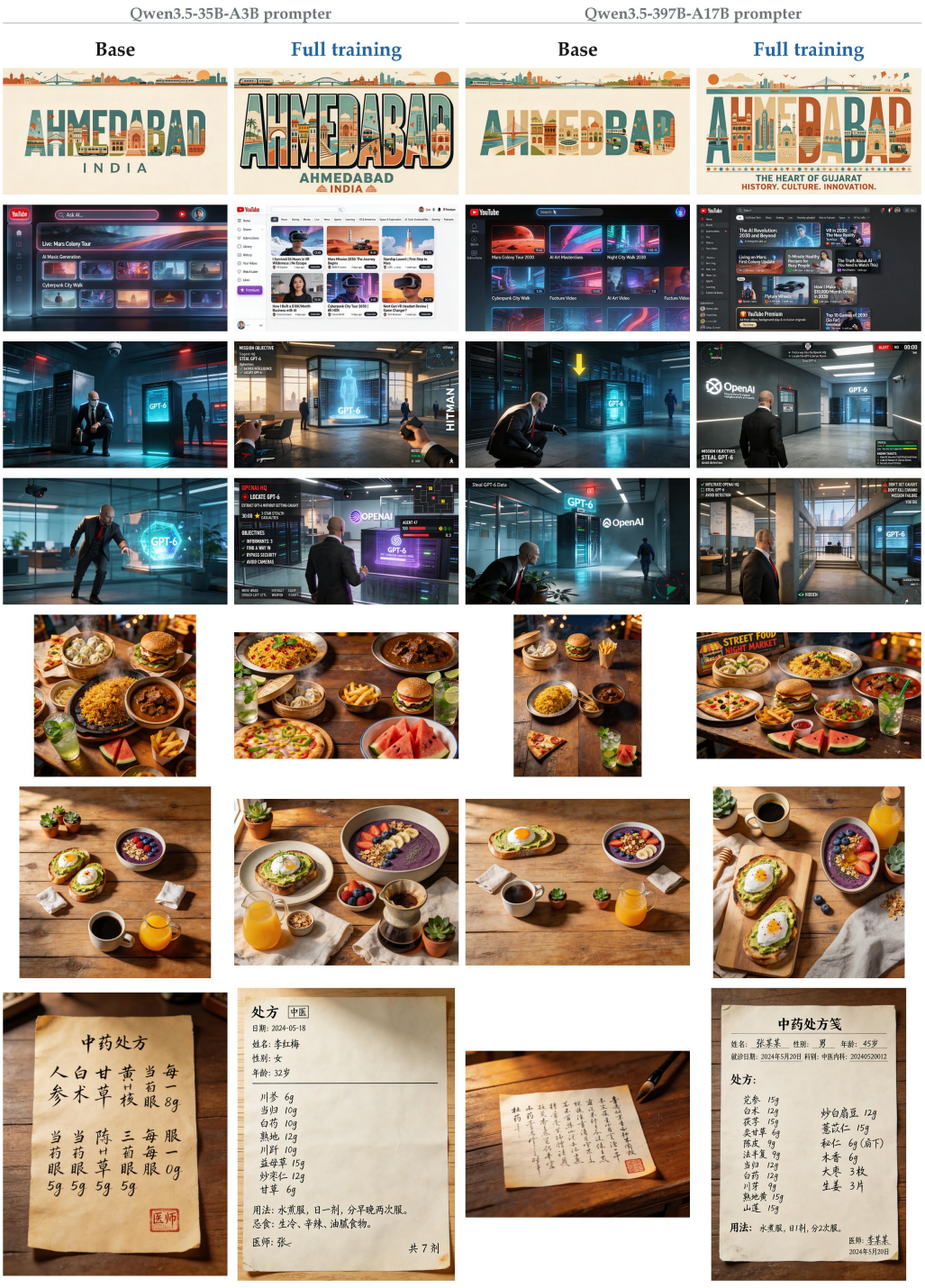


Figure 21: **Prompter comparison: prompter scale × training (1/3).** Four columns compare two prompter scales, each before and after the full training pipeline, with the same Qwen-Image backbone. Rows follow the prompt order in Appendix I.8; aspect ratio is predicted by the prompter.



Figure 22: **Prompter comparison: prompter scale × training (2/3).** Continued from Figure 21; same four-column layout and prompt order.



Figure 23: **Prompter comparison: prompter scale × training (3/3).** Continued from Figure 21; same four-column layout and prompt order.

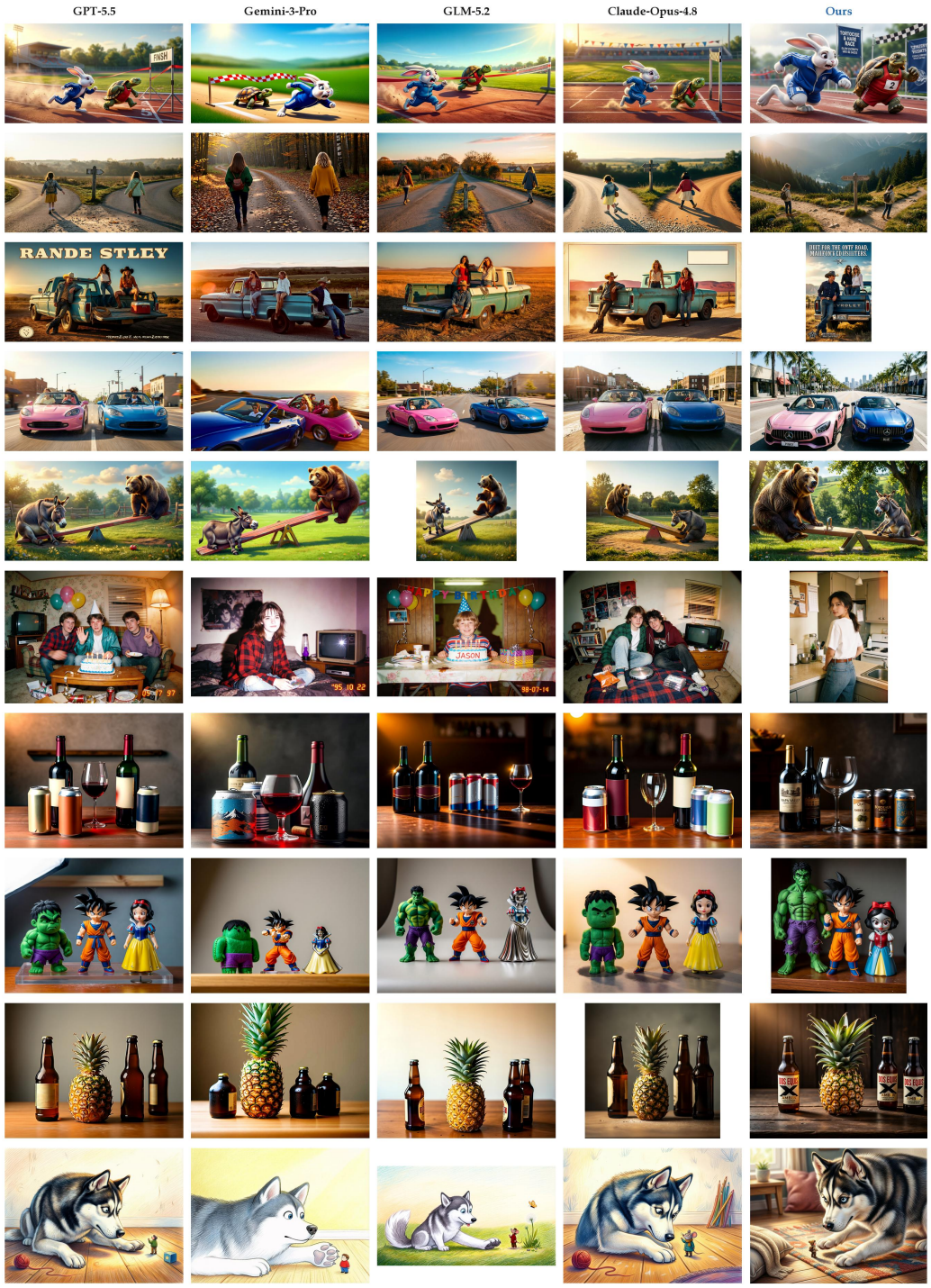


Figure 24: **Different LLMs as prompters (1/2).** General-purpose LLMs and our trained prompter fill the same schema and render with the same Qwen-Image backbone. Rows share the same user prompts. Table 17 reports the quantitative per-backend scores.



Figure 25: **Different LLMs as prompters (2/2).** Continued from Figure 24; same five-column layout and shared Qwen-Image backbone.