EDITOR'S CHOICE: EVALUATING ABSTRACT INTENT IN IMAGE EDITING VIA ATOMIC ENTITY ANALYSIS

Anonymous authors
Paper under double-blind review



Figure 1: **Evaluating Abstract Image Editing**. Given a context image and an abstract instruction from our ABSTRACTEDIT test set, our ENTITY-RUBRICS framework assesses the edited image by decomposing the scene at a granular level, yielding a per-edit rank that determines the final instruction-following score. As shown, some entity edits lead to better prompt alignment (e.g., the man's expression), while others introduce unnecessary changes that hurt preservation (the crowd).

ABSTRACT

Humans naturally communicate through abstract concepts like "mood". However, current image editing benchmarks focus primarily on explicit, literal commands, leaving abstract instructions largely underexplored. In this work, we first formalize the definition and taxonomy of abstract image editing. To measure instruction-following in this challenging domain, we introduce ENTITY-RUBRICS, a framework that breaks down abstract edits into individual, entity-level assessments and achieves strong correlation with human judgment. Alongside this framework, we contribute ABSTRACTEDIT, the first benchmark dedicated to abstract image editing across diverse real-world scenes. Evaluating 11 leading models on this dataset reveals a fundamental challenge: standard architectures struggle to balance intent and preservation, commonly defaulting to under-editing or over-editing. Our analysis demonstrates that driving meaningful improvements relies heavily on integrating advanced LLM text encoders and iterative thinking. Looking forward, our entity-based paradigm can generalize beyond assessment to serve as a reward model, enable models to correctly interpret abstract communication, or highlight specific failures in test-time critique loops. Ultimately, we hope this work serves as a stepping stone toward seamless multimodal interaction, closing the gap between rigid machine execution and the natural, open-ended way humans communicate.

1 INTRODUCTION

Image generation models have made significant progress in following explicit commands (Brooks et al., 2023; Betker et al., 2023), yet their ability to interpret abstract instructions remains a critical bottleneck (Liu & Chilton, 2022; Fu et al., 2023). In creative applications, humans rarely think in terms of exhaustive, technical specifications; instead, they communicate through high-level concepts, such as altering an image's mood or season (Norman, 2014; Zamfirescu-Pereira et al., 2023; Oppenlaender, 2024). Natural communication relies on a shared world understanding, allowing humans to convey complex ideas without specifying every granular detail (Grice, 1975; Clark & Brennan, 1991). However, current visual generative models still struggle to achieve this level of *abstract* semantic alignment.

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.

Prior work has recognized this issue but has typically approached it from narrow angles (§2). Studies have predominantly focused on limited implicit instructions (Pu et al., 2025; Zeng et al., 2025; Yu et al., 2025), or on physical reasoning (He et al., 2025; Zeng et al., 2025; Yu et al., 2025), where target outcomes are singular and easily quantifiable, leaving little room for true abstraction. This hesitancy to fully tackle abstract instructions in real-world scenes stems from their inherent difficulty: abstract prompts are highly subjective, allowing many valid interpretations. Therefore, in this work, we establish a framework to tackle the broader challenge of abstract instruction following through a formal problem definition, and a novel evaluation suite.

First, we argue that instruction-based **image editing** provides a uniquely highly practical testbed for bridging this gap compared to a pure text-to-image setup. Generally, image editing is considered more complex than text-to-image generation due to a dual objective: the model must adhere to the text instruction while rigorously preserving the original image context. However, when evaluating abstract instructions, this dual constraint is actually a profound advantage. The requirement to preserve the context image significantly narrows the infinite space of possible interpretations, anchoring the abstract request to a grounded visual baseline and making systematic evaluation tractable.

Next, we define *abstract image editing* (§3). It is determined by both the text instruction and the context image, and is measured along two axes spanning human intent: Identification and Specificity. This concept can be conveyed by asking a simple question: *would a majority of human observers identify the exact same target entities to be edited, and would they expect the exact same visual transformations?* If either of these conditions fails, the prompt is considered an abstract edit. Explicit instructions, such as "add mud to the paws", provide a clear roadmap. However, the intent to "make the dog look like it just returned from a long rainy trip" (Fig. 2) is entirely abstract, specifying neither target nor transformation. Such abstract intent allows diverse valid edits, such as making the fur wet, moving to a bathtub, or adding an umbrella, requiring the model to autonomously deduce multiple required edits. This definition of abstractness as a subjective, multi-path problem naturally dictates a shift toward **granularity** and **precision** in the evaluation.

To achieve this granularity, we draw inspiration from factuality evaluation in NLP. There, complex generated text is assessed by decomposing it into "atomic facts" and verifying each one against a source document (Gekhman et al., 2023; Min et al., 2023; Honovich et al., 2022; Wei et al., 2024). We posit that abstract visual transformations can be similarly evaluated by analyzing the discrete modifications applied to image entities. Despite the subjectivity of what and how to edit, results must be objectively grounded in the input narrative. Since holistic transformations are more than a "sum of parts" we treat entity-level consistency as a necessary, but not sufficient, requirement for alignment.

Operationalizing this perspective, we propose **ENTITY-RUBRICS**, a VLM-based, novel granular automatic evaluation framework. ENTITY-RUBRICS treats individual image entities, such as objects and stuff, as the atomic units of assessment (§4), and provides a highly interpretable rationale, as opposed to existing evaluation methods which at most provide only a single global score and a short explanation (Ku et al., 2024; Yang et al., 2025). Since abstract prompts allow multiple valid interpretations, target edits are not predetermined. We therefore prioritize precision over recall: rather than checking if a model captured every possible way to interpret a prompt, we verify that its chosen edits are logically grounded in the prompt. In doing so, our approach captures multiple local edits and separates intentional transformations from arbitrary changes that degrade preservation (Fig. 1), while strongly correlating with human judgment.

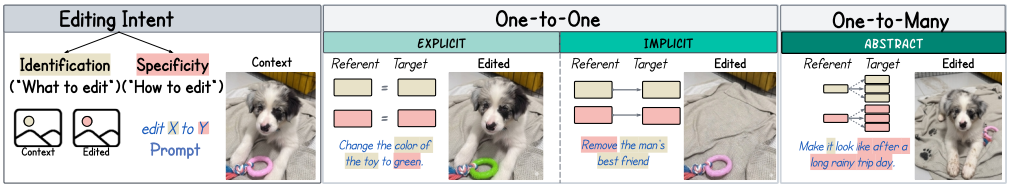


Figure 2: **Taxonomy of Image Editing Intent**. We formalize intent across two orthogonal axes: Identification ("what to edit") and Specificity ("how to edit"). Intent is categorized based on the mapping between these axes: Explicit and Implicit editing provide a one-to-one mapping through high consensus on both axes, whereas abstract editing introduces a one-to-many relationship.

To address the lack of structured resources, we introduce ABSTRACTEDIT (§5), a diverse dataset of editing scenarios designed specifically to evaluate abstract intent. The test set consists of 470 human-verified abstract samples spanning four core domains, ranging from physical and logical changes to emotional and social shifts. Each sample is a triplet consisting of a context image, an abstract edit instruction, and a corresponding explicit instruction serving as a representative concrete example. To support model development, we further provide a training expansion of 4k samples. ABSTRACTEDIT is produced via a novel, fully automated VLM-powered pipeline designed to ensure the systematic generation of contextually relevant, diverse prompts grounded in real-world scenes.

Leveraging the ABSTRACTEDIT and the ENTITY-RUBRICS framework, we benchmark 11 leading models (§6) and reveal a stark performance gap. The models exhibit divergent behaviors: closed-source models thrive on abstract intent but often over-edit and sacrifice original context, whereas open-source models exhibit variable performance, relying heavily on explicit instructions and frequently under-editing by missing latent requirements. We find that advanced LLM text encoders and iterative "thinking" mechanisms are key to boosting abstract instruction following. Ultimately, our analysis confirms that abstract prompts unlock model creativity and a quantifiable manifold of visual diversity.

In summary, our work provides a comprehensive framework to formalize and benchmark abstract image editing. Our core contributions are: (1) a formal definition of *abstract* image editing; (2) an ENTITY-RUBRICS evaluation framework; (3) The ABSTRACTEDIT benchmark and its automated generation pipeline; and (4) an in-depth analysis comparing open- and closed-source editing models. Ultimately, this framework moves the field beyond rigid instruction-following toward alignment with complex human intent, providing a roadmap to master the one-to-many nature of real-world editing.

2 RELATED WORK

Instruction-Guided Image Editing Datasets. Early image editing models relied on local masks (Avrahami et al., 2022; Joseph et al., 2024; Yuan et al., 2024), but frameworks like InstructPix2Pix (Brooks et al., 2023) and MagicBrush (Zhang et al., 2023) shifted the paradigm toward instruction-guided editing to facilitate natural human communication (Clark & Brennan, 1991). However, these datasets primarily focus on simple, explicit commands (Fu et al., 2023; Ma et al., 2024; Qian et al., 2025b). To push the boundaries of edit complexity, subsequent datasets incorporated multi-step instructions (Yang et al., 2025), implicit prompts requiring specific domain knowledge (Huang et al., 2024; Jia et al., 2025), or hypothetical scenarios within the physical domain (He et al., 2025; Zeng et al., 2025; Yu et al., 2025) (see §5 and Table 1 for comparison). Yet, these benchmarks evaluate rigid, one-to-one visual mappings rather than true abstract instructions, which require models to translate vague intents into specific edits that align with a user's mental model (Norman, 2014). To bridge this gap, we formalize abstract image editing along the axes of Identification and Specificity, introducing the ABSTRACTEDIT, a benchmark specifically designed to evaluate abstract image editing.

Evaluating Instruction Alignment. Evaluating image editing requires balancing instruction following with context preservation (Brooks et al., 2023). However, encoding-based metrics like CLIP (Radford et al., 2021) or DINO (Siméoni et al., 2025) struggle with fine-grained semantics (Thrush et al., 2022; Liang et al., 2022; Hu et al., 2023; Zhang & Wan, 2025), often rewarding aggressive over-editing over logical though subtle changes. To address this, VQA metrics (Lin et al., 2024; Hu et al., 2023) attempt localized checks, but their reliance on rigid proxy questions is conceptually misaligned with abstract instructions and penalizes valid visual diversity. Alternatively, recent VLM judges (Ku et al., 2024; Yang et al., 2025) capture broader semantics but typically output holistic scores that lack interpretable, localized explanations. Crucially, none address how to unpack an abstract instruction to systematically verify its execution. To bridge this gap, we introduce ENTITY-RUBRICS, a framework that measures alignment to high-level intents through atomic, entity-level verification.

Approaches to Complex Abstract Instructions. Recent methods address complex instructions using LLMs to decompose abstract prompts into explicit sub-tasks (Ji et al., 2025; Yu et al., 2025; Iakovleva et al., 2024; Zeng et al., 2025). Building on this, some frameworks execute these simple steps sequentially (Yeh et al., 2025; Yang et al., 2025), or introduce agentic loops with iterative visual feedback (Li et al., 2025; Wang et al., 2024; Fang et al., 2025; Yao et al., 2026). While effective, these computationally expensive strategies restrict visual diversity. Shifting the semantic burden to the text encoder forces rigid interpretations and leaves the diffusion model's innate mapping abilities

Table 1: **Comparison of benchmark subsets for complex image editing**. Size refers to the test set size; Dom indicates the number of covered domains categorized as Low $(1-2)$, Medium $(3-4)$, and High (≥ 5); Ctx Img counts the unique context images; Nat. denotes if prompts are natural (✓) or templated (✗); and Global/Local specifies the edit spatial scope. eDoF (editing Degree of Freedom) describes the intent mapping: one-to-one $(1:1)$ or one-to-many $(1:N)$ abstract.

Bench.	Subset	Size	Dom	Ctx. Img	Nat.	Glob/ Loc	eDoF	*Example Instruction*
CompBench (Jia et al., 2025)	Implicit	100	L	100	✗	L	1:1	*Remove the farthest tiger from the water*
SmartEdit (Huang et al., 2024)	Implicit	60	M	30	✗	L	1:1	*Remove the object that can be used to have meals*
EditWorld (Zeng et al., 2025)	Logic	60	L	60	✗	-	1:1	*What happens if a hole appears in the balloon?*
AnyEdit (Yu et al., 2025)	Implicit	10k	L	10k	✗	-	1:1	*What would happen if the man can't catch the wave?*
EMU-Edit (Sheynin et al., 2024)	Global	440	M	219	✓	G	1:N	*Change the scene to night time*
Kontext (Labs et al., 2025)	Global	262	M	88	✓	G	1:N	*Make this image real*
AbstractEdit	**Full Abs.**	**470**	H	257	✓	G,L	1:N	*Make the dog look like after a long rainy trip day*

unoptimized. Instead, our pipeline provides scalable data, including a 4k training set, enabling future work to directly optimize diffusion models for abstract edits. Complementing this, Entity-Rubrics offers entity-level diagnostics to target specific failures without exhaustive trial-and-error.

3 Abstract Image Editing: Taxonomy and Definition

This work focuses on the underexplored complexity of *semantic abstractness in human intent*, as opposed to previous studies that characterize complexity through instruction count (Yang et al., 2025; Wang et al., 2025b) or knowledge-based implicit references (Huang et al., 2024; Qian et al., 2025a). By shifting the focus of complex image editing toward interpretation, abstract editing measures a model's ability to translate underspecified intent into concrete visual modifications. This process quantifies the "generative guesswork" required to bridge the conceptual gap between the text instruction (p) and context image ($\mathcal{I}_c$) and a satisfying result.

To formalize abstract image editing, we define two orthogonal axes spanning the human intent space: *identification* and *specificity*. **Identification** pinpoints the exact semantic or visual entities targeted by the edit, whether modifying existing elements or inserting new ones (the "what"). **Specificity** determines the precise nature and extent of the visual transformations applied to those elements (the "how"). Conceptually, the generative guesswork required to resolve these axes can be viewed as an *Editing Degree of Freedom ($eDoF$)*. Operating as a continuous spectrum, $eDoF$ represents the autonomy a model must exercise to interpret an instruction. While by nature a strictly theoretical construct, we could define $eDoF$ based on the size of the interpretation set $\mathcal{K}$, which represents all visually distinct edits that satisfy the text prompt p given the context image $\mathcal{I}_c$: $eDoF \propto |\mathcal{K}(p|\mathcal{I}_c)|$.

The magnitude of this interpretation set dictates the edit's mapping (Fig. 2). When an instruction is clear along both axes, it yields a strict *one-to-one* mapping where $|\mathcal{K}| \approx 1$ and the lowest $eDoF$. We divide these strictly mapped cases into two sub-types (Fig. 2). **Explicit** (Exp) edits directly map the literal instruction perfectly to the outcome (e.g., "turn the toy green"). **Implicit** (Imp) edits also maintain a one-to-one mapping but require additional computational capabilities, such as specific domain knowledge (understanding "man's best friend" is a dog) or robust visual detection (e.g. detecting the colors when instructed to "swap the sofa's color with the toy's color"). Conversely, when ambiguity exists along the identification or specificity axes, the interpretation set expands into a *one-to-many* mapping ($|\mathcal{K}| \gg 1$). These underspecified instructions demand a high $eDoF$ and are categorized as **abstract** (Abs) edits, as they remain open to a vast array of valid visual interpretations.

In practice, applying these axes is challenging because both offer endless possibilities. For identification, a target can be anything from a simple "dog" to a complex interaction like "the dog's gaze on the toy" (Kress & Van Leeuwen, 2020). For specificity, any chosen target can then be transformed into a multitude of valid ways. To manage this complexity in our evaluation (§4), we adopt a precision-over-recall approach. Rather than attempting to map the entire infinite space of what and how a model could have edited, we focus on verifying that the specific edits the model actually performed are logically grounded in the instruction. Further formalization is detailed in §A.

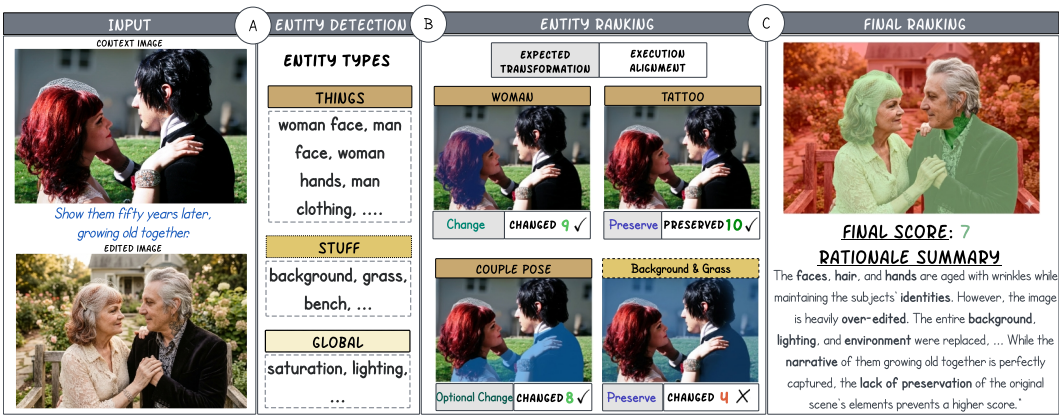


Figure 3: **Overview of the three-stage VLM-based ENTITY-RUBRICS evaluation framework**. (A) Entity Detection identifies relevant entities. (B) Entity Ranking assigns an expected transformation to each entity (*Change*, *Optional*, *Preserve*) and measures its execution alignment in the edited image. (C) Final Scoring aggregates these into a global rank and rationale. Results are visualized directly on-image via a red (incorrect) to green (correct) scale.

4 ENTITY-RUBRICS: AUTOMATIC EVALUATION FRAMEWORK

Evaluating abstract intent in image editing is uniquely challenging because a single instruction often triggers changes across multiple entities in various ways. As shown in Fig. 3, "making a couple look older" involves more than just graying a woman's hair, it entails adding facial wrinkles, updating outfits to reflect age, and potentially adjusting gaze to suggest a lifelong partnership. Effective evaluation must capture these nuances, distinguishing between essential transformations and optional creative choices for a weighted performance assessment. Furthermore, because these instructions are complex, models often succeed at some edits while failing at others. Current methods, whether based on image-text join-encoding as CLIP (Radford et al., 2021) or zero-shot VLM judgments (Ku et al., 2024; Yang et al., 2025), lack the precision to account for this multi-entity coordination and fail to provide the granular feedback necessary for such a high-dimensional task. To address this, we introduce **ENTITY-RUBRICS**, a novel automatic evaluation framework centered on the granular inspection of entities, which inherently supports fast-to-interpret visual reporting.

Our framework is inspired by factuality research in the NLP domain, which frames semantic alignment as Natural Language Inference (NLI) (Bowman et al., 2015; Maynez et al., 2020; Kryscinski et al., 2020), and verifies information by decomposing complex claims into "atomic facts" for individual validation (Gekhman et al., 2023; Min et al., 2023; Honovich et al., 2022; Wei et al., 2024; Chern et al., 2023). As illustrated in Fig. 3, we adapt this concept to the visual domain by treating image entities as "atomic units". We decompose the evaluation into discrete logical steps, leveraging the VLM's reasoning to perform a per-entity alignment and necessity assessment, given the context image, the candidate edited image and the instruction. This granular focus ensures that the final evaluation score is grounded in specific visual evidence, verifying both the correctness of modifications and the necessity of preservation, rather than relying on a global, surface-level explanation.

As illustrated in Fig. 3, the ENTITY-RUBRICS evaluation framework consists of three main stages: entity detection, entity ranking, and final scoring. The first stage, **(A) Entity Detection**, identifies all entities, including global attributes, within both the context and the edited images. To ensure comprehensive coverage, the VLM is prompted to partition these entities into three semantic groups: *things* (distinct objects such as "woman's face"), *stuff* (amorphous background (Caesar et al., 2018); environmental elements like "grass"), and *global* (global attributes as lighting). By categorizing entities this way, the framework captures intent-driven changes across every layer of the image.

Next, the **(B) Entity Ranking** stage serves as the core of the framework. It addresses the tension between the expected edit and the actual candidate edited image produced by an image editing model. This stage utilizes a two-phase design, which we validate via ablations in §H.1. First, the VLM determines the Expected Transformation for each entity. It iterates over all detected entities from

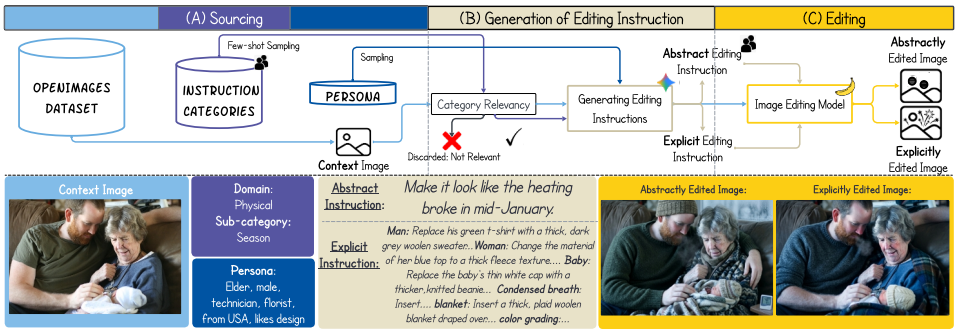


Figure 4: **The AbstractEdit Automatic Curation Pipeline**. (A) Sourcing: Context images (Azure) from OpenImages are compiled alongside manual category examples (Purple) and diverse personas (Blue). (B) Instruction Generation: After filtering the images for category relevancy, an LLM, prompted by few-shot examples and a random persona, generates paired abstract and explicit instructions (Beige). (C) Editing: Both instructions are applied to the context image to produce the final edited pairs (Yellow). Bottom: A generated AbstractEdit example.

the context image to assess their expected state based solely on the abstract instruction. This phase represents the identification axis, acting as a theoretical grounding step to determine what requires modification. Specifically, it assigns an expected state of $Change$, $Optional\ Change$, or $Preserve$ before any visual comparison occurs, using only the context image and the text prompt as input.

Then, the evaluation moves to examining the edited image to analyze the modifications present in the candidate result. In this phase, Execution Alignment, the VLM first assesses whether a change occurred for each entity. If a change is detected, it is followed by a qualitative assessment of how the entity was transformed or preserved, producing a concise description of the modification (not shown in the figure). These two phases culminate in a per-entity rank, where the VLM calculates a score based on how well the observed changes align with the expected transformations and whether they increase overall alignment with the instruction. To ensure holistic quality, the VLM is also prompted to provide a global failure profile that audits the image for missing changes or over-editing artifacts not captured at the individual entity level (not shown in the figure).

Finally, **(C) Final Scoring and Rationale** aggregates all previous findings into a comprehensive 1–10 assessment, prioritizing instruction adherence, overall image coherence, and the expected level of change (*under-editing* for missed changes and *over-editing* for unexpected edits, both extracted as boolean flags). By combining the data from all entity categories with the global examination, the framework generates a score and a written rationale. A score of 10 reflects perfect alignment where all required changes are executed flawlessly while preserving the essential context of the context image. Lower scores are assigned based on the severity of issues such as missed requirements, disjointed narratives, or over-editing artifacts. This synthesis ensures the final evaluation is a grounded reflection of performance across every image layer. Recognizing the black-box nature of the VLM primarily at this stage, we demonstrate the explainability of the final score through entity-level features in §H.2. The full prompt architecture is detailed in §C.

To validate our automatic evaluation, we conducted an Amazon Mechanical Turk study collecting 1,080 responses across 60 stratified AbstractEdit samples (see §D). Three annotators per task evaluated edits from the six top models, assessing instruction following, content preservation, image quality, and granular per-entity alignment. Aggregating responses via mean vote, the study achieved moderate Inter-Annotator Agreement (Fleiss' κ of 0.47). Crucially, Entity-Rubrics achieves a Spearman's ρ of 0.66 with human judgments, confirming it as a high-fidelity proxy for instruction following while outperforming alternative metrics like VIE (0.54) and image-text CLIP (0.41).

5 The AbstractEdit Dataset: Abstract Image Editing

Evaluating abstract image editing requires benchmarks that capture the inherent complexities of one-to-many mappings. Our analysis of the broader landscape of complex editing (Table 1) reveals that existing commonly adopted datasets are often restricted by template-based language and narrow

domain diversity. Most critically, these frameworks suffer from a diagnostic gap: they treat the editing process as a black box, failing to decouple high-level semantic inference from low-level generative execution. Without this distinction, it cannot be determined if a failure is due to misinterpreting the concept or an inability to perform the edit.

To address these gaps, we present ABSTRACTEDIT, a dedicated benchmark for evaluating abstract image editing across diverse, real-world scenes. It is structured to support direct comparisons between paired abstract and explicit instructions. By providing both prompt types for each context image, the explicit edits serve as a direct baseline to isolate and quantify a model's abstract interpreting capabilities. The ABSTRACTEDIT test set contains 470 human-verified samples (via manual author review) to ensure they strictly adhere to our definition of abstract editing, as well as logically grounded. Additionally, we provide a 4k training set. However, as training is highly challenging and beyond our current scope, we defer it to future work, providing preliminary results in §I.

ABSTRACTEDIT spans 12 categories across four domains: Physical, Logical, Emotional, and Social. The domains demand varying generative autonomy, testing models' ability to resolve the inherent ambiguity of abstract intent. For instance, the Emotional domain demands high $eDoF$; Given a prompt to "infuse empathy and a deep bond with animals" (Fig. 1), the model should infer that the matador's aggressive posture should be replaced with a nurturing stance. Unlike benchmarks focusing on object detection or implicit references, ABSTRACTEDIT targets abstract instructions that necessitate coordinated local and global edits across multiple entities. For additional details see §B.

As illustrated in Fig. 4, ABSTRACTEDIT is generated via an automatic pipeline designed to capture a broad spectrum of authentic human intents through diverse and natural instructions. Our pipeline operates in three phases: (A) Sourcing, (B) Instruction Generation, and (C) Image Editing. The **Sourcing (A**) phase establishes the creative constraints of the dataset. We begin by selecting 1,300 natural images from the Open Images v7 opensource dataset (Kuznetsova et al., 2020), prioritizing complex scenes with multiple interacting entities to ensure the model has a rich "canvas" to edit. To guide the abstraction, we manually define a taxonomy of the categories (e.g., *Physical: Seasonal*), including instruction examples for each. We then utilize synthetic personas, defined by traits like age and expertise, to avoid predictable templates. Randomizing these traits spans a broad linguistic spectrum; a practical persona might dictate to "make the room look freezing to test the insulation", whereas a poetic persona might say "infuse the room with the biting chill of a mid-January morning".

Instruction Generation (B) then translates these images, instruction examples and personas into actionable tasks. For every context image, we first check semantic relevance to ensure the chosen category is grounded in the context image, for instance, we would not apply a "Physical: Pose" edit to an image of an empty desert. For valid pairings, we use a few-shot prompting approach where an LLM (Gemini 2.5 Pro (Comanici et al., 2025)), adopting a specific random persona, generates two types of instructions: an abstract intruction representing underspecified human intent (e.g., "The heating just broke"), and explicit instructions detailing the concrete visual modifications that "solve" the abstraction (e.g., "Add frost to window edges and place a wool blanket on the sofa"). We then manually verify the abstractness of each sample of the test set. In the **Editing phase (C**), we execute both prompts to produce the final image pairs.

6 RESULTS & EXPERIMENTAL SETUP

We evaluate 11 state-of-the-art image editing models selected for their diverse parameter scales and architectures. This comprises 8 open-source (OS) models, two of which include thinking variants, and 5 closed-source models (see §E). Table 2 presents the primary ENTITY-RUBRICS scores on the ABSTRACTEDIT test set, computed using Gemini 3 Flash, and includes human evaluations for the top six performers, with Fig. 6 providing a qualitative example of these abstract edits. Fig. 5 further illustrates the performance gap between abstract and explicit prompts. Supplementary preservation and instruction-following metrics, and complementary results are deferred to the Appendix (§E and §F) as well as qualitative results (§G), and the use of ENTITY-RUBRICS as a critique-and-revise mechanism (§H.4). Our analysis reveals several key trends:

Prompting Trade-offs in Open vs. Closed Models. Performance gap exists between closed-source and open-source models, with closed-source models achieving a high mean score of 9.2 compared to top open-source performers like Qwen-Image-Edit (7.48). As presented in Fig. 5, this divide

Table 2: **Abstract Instruction Following Performance**. **Left Table**: Open-Source, OS with thinking modes, and closed models ranked by Abstract ENTITY-RUBRICS scores and human annotations. **Failure Profile:** Under-Editing ← | → Over-editing. Domains: **Emotional**, **Logical**, **Physical**, and **Social**. **Right Fig. 5**: Prompt Type Comparison: Explicit (blue) and Abs. (striped bordeaux).

Cat.	Model	Hum ↑	Abs ↑	F. Prof.	Emo-tional	Logi-cal	Physi-cal	So-cial
Open Source	Qwen-Image-Edit	-	**7.48** ±2.86		8.00 ±2.89	6.89 ±3.02	**7.43** ±2.92	**8.14** ±2.41
	FLUX.2 [dev]	8.50	7.26 ±2.83		**8.29** ±2.04	**7.34** ±2.89	6.85 ±3.24	7.16 ±2.57
	HiDream-E1	-	5.38 ±3.15		7.91 ±2.12	4.18 ±3.05	4.97 ±3.23	6.56 ±2.63
	FLUX.1-Kontext [dev]	7.69	5.10 ±3.46		6.17 ±3.48	4.34 ±3.44	5.21 ±3.62	5.72 ±3.16
OS w/ Thinking	Step1X-Think-Reflect	7.97	**6.90** ±3.20		**7.63** ±3.06	**7.39** ±3.22	5.39 ±3.64	**7.14** ±2.50
	Step1X	-	6.55 ±3.41		7.61 ±3.06	6.90 ±3.38	5.36 ±3.83	6.64 ±2.97
	Bagel-Think	-	5.80 ±3.50		5.85 ±3.43	4.84 ±3.67	**5.97** ±3.49	6.92 ±2.91
	Bagel	6.61	4.45 ±3.42		4.89 ±3.45	3.31 ±3.21	4.90 ±3.37	5.53 ±3.29
Closed Source	Gemini 3.1 Flash	9.66	**9.52** ±1.37		**9.62** ±1.33	9.47 ±1.34	**9.43** ±1.64	**9.62** ±1.19
	GPT-Image	9.67	9.34 ±1.37		9.49 ±1.43	9.19 ±1.42	9.36 ±1.46	9.49 ±1.20
	Gemini 3 Pro	-	9.27 ±1.78		9.50 ±1.40	**9.59** ±1.13	8.46 ±2.80	9.35 ±1.42
	Seed-Dream	-	9.21 ±1.76		9.43 ±1.44	9.21 ±1.71	8.88 ±2.33	9.39 ±1.38
	Gemini 2.5 Flash Image	-	8.67 ±2.40		9.23 ±1.70	9.09 ±2.04	7.57 ±3.26	8.73 ±2.05

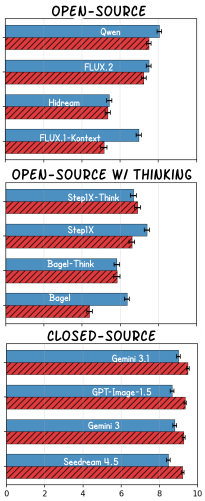




Figure 6: Qualitative example edited by different models, representative from each model-category.

is driven by a fundamental difference in instruction processing. Closed-source models thrive on abstract instructions, improving by up to 9% with tighter score distributions when presented with high-level intent. In contrast, open-source models generally need explicit roadmaps to maintain alignment. However, this reliance on explicit instructions creates a severe trade-off for open-source architectures. Lengthy text often overwhelms these models, leading to excessive over-editing as text tokens dominate the original image context. Although switching to abstract prompts universally reduces this over-editing across all models (avg. −13.3%), it merely shifts the failure mode. Unable to decipher high-level intents, open-source models fail to execute the edit entirely, trading over-editing for severe under-editing. Thus, while open-source models require explicitness to function, abstraction serves as a necessary regularizer for preserving the original image.

Across domains (Table 2), we find that open-source models perform better on Emotional and Social domains (mean 6.88) than on Physical and Logical ones (5.7). We hypothesize that this gap is a function of feature alignment. Emotional and Social edits often map to strongly-encoded semantic patterns (e.g., color shifts or facial expressions) robustly represented in the generated image latent space. In contrast, Physical and Logical domains require "multi-hop" compositional reasoning to resolve complex spatial configurations, object interactions, and constraints that cannot be satisfied by simple latent pattern retrieval alone.

Failure Analysis. Open-source and closed-source models exhibit opposite failure modes. Open-source models tend toward under-editing, failing to capture latent requirements in abstract prompts (Table 2; purple bar of Failure Profile); this results in inflated preservation metrics (Appendix Table 6) and is further analyzed in §H.3. Conversely, closed-source models are prone to over-editing (red bar), discarding the original context to satisfy underspecified instructions.

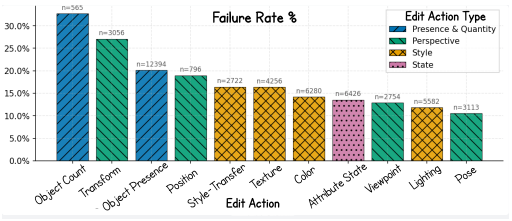


Figure 7: **Failure Rate per Entity Edit Action**. Normalized by the number of occurrences of each edit type across all models.

Action-specific analysis (Fig. 7), based on the entity-ranking stage of our evaluation (§4), reveals granular bottlenecks. The highest failure rates ($> 30\%$) involve Object Counts, suggesting that numerical grounding remains the primary hurdle in image editing (Dahary et al., 2025; Binyamin et al., 2025; Tewel et al., 2024). This is followed by Perspective (scale, rotation, position) and Object Presence, which often yield missing or malformed entities. In contrast, models reliably handle "surface" edits like Style, Texture, and Attribute State. This confirms that the primary bottleneck in abstract instruction following is structural reasoning rather than aesthetic adjustment.

Thinking and Text Encoders as Drivers of Performance in Abstract Instruction Following. Two key drivers improve instruction following for abstract prompts: advanced LLM backbones and iterative thinking processes. Among open-source models, the choice of text encoder is a primary performance factor. For instance, FLUX.2 [dev] utilizes a Mistral and T5 configuration (7.26) to outperform the CLIP-based Flux-Kontext (5.1), while Qwen-Image-Edit leverages a Qwen-VL-2.5 backbone. These are the highest-scoring open-source models. Beyond architecture, iterative thinking mechanisms significantly stabilize performance on abstract tasks. Step1X with its thinking mode improves its ENTITY-RUBRICS score by 5.3% over the base model, achieving a high human instruction-following score of 7.97. Similarly, Bagel-Think demonstrates a 30.3% increase in its score compared to the standard Bagel model. The efficacy of these methods suggests that the observed performance gap between closed-source and open-source models is largely driven by the adoption of similar inference-time thinking processes (Google DeepMind, 2026). Furthermore, these thinking models offer a mixed benefit: they boost alignment on abstract instructions ($+1.4$ points for Bagel) but suffer a 'precision tax' on explicit prompts (Fig. 5), where overthinking hurts direct execution.

Abstractness Unlocks Diversity. To examine the link between abstraction and creativity, we analyze how prompt types affect the diversity of the edited images. Using leading open-source (FLUX.2) and closed-source (Gemini 3.1) models, we generate 2,000 images by processing 100 random samples across 10 seeds for each prompt type, abstract and explicit (Appendix Fig. 19). We measure the semantic and visual diversity of these outputs via Vendi Score (Friedman & Dieng, 2022) on DINOv3 (Siméoni et al., 2025) features. We found that abstract prompts consistently produce a higher mean and density of diverse visual concepts than explicit ones. While this shift toward variety is conceptually expected, our results provide a clear quantification of the effect. The gap widens in the closed-source model, indicating that while explicit instructions force models toward a narrow output, abstract prompts allow them to explore a much wider manifold of creative interpretations.

7 CONCLUSIONS AND DISCUSSION

In this work, we formalized abstract image editing, shifting the evaluation paradigm from holistic scoring to granular diagnostic inspection via our ENTITY-RUBRICS framework and the ABSTRACTEDIT benchmark. We demonstrate that treating image entities as atomic units of assessment provides interpretable, entity-level feedback that correlates with human judgment. Our analysis reveals open-source models are trapped in an under/over-editing trade-off that iterative thinking only superficially mitigates. Still, as the first framework to formalize abstract editing, our evaluation remains a first-order baseline that does not capture cross-entity relations or sub-entity components.

Looking forward, ABSTRACTEDIT can be expanded across diverse domains to test abstractness in new contexts, while ENTITY-RUBRICS can be augmented with agentic capabilities like external image retrieval. Crucially, the utility of ENTITY-RUBRICS extends beyond assessment: its granular feedback can actively steer generation, either as a reward signal for preference alignment or for test-time scaling. Ultimately, we hope our evaluation suite catalyzes the development of models that move beyond executing technical blueprints to fluidly navigating the "one-to-many" interpretation space of abstract human communication.

AI USE STATEMENT

In this work, we employed generative AI tools to assist with data processing, writing code for the manual annotation framework, visualizing results, and resolving LaTeX formatting errors. We did not rely on AI for our baseline implementations, the core evaluation framework, or generating fundamental research ideas. Furthermore, AI was used for routine text editing, such as grammar and word choice refinements. All AI-generated outputs were strictly reviewed by the authors. Specifically, we verified the functionality of the annotation framework and its data-saving formats, and manually checked all grammar adjustments in the evaluation prompts. We take full responsibility for the final content, claims, and artifacts presented in this paper.

ETHICS STATEMENT

The ability of image generation models to understand abstract instructions has significant societal implications with both positive applications and potential risks. Our analysis provides valuable insights into model behavior, helping users navigate the gap between rigid explicit commands and open ended human intent. By revealing how different models interpret abstract instructions, our findings can guide users in crafting more effective prompts, making AI assisted design more accessible to those without extensive prompt engineering expertise. Furthermore, our evaluation provides developers with a concrete metric to identify and fix failure modes like context destruction, leading to safer and more reliable models. However, the primary risk of abstract editing lies in latent model biases. When an instruction is ambiguous, the model must fill in the unspecified details. Left unchecked, models may rely on harmful stereotypes to resolve this ambiguity and exacerbate representation biases. By formalizing the axes of identification and specificity, our taxonomy exposes exactly where these inferential leaps occur. We strongly encourage researchers to use our dataset not just for capability benchmarking but as an auditing tool to uncover how different architectures interpret socially sensitive abstractions.

REPRODUCIBILITY STATEMENT

To ensure the full reproducibility of our benchmark and evaluation framework, the complete ABSTRACTEDIT including all context images, abstract/explicit paired instructions, and human verification data, will be released as open-source. The codebase for our ENTITY-RUBRICS framework, along with all scripts required to reproduce the evaluation and further experiments detailed in the appendix, such as the test-time critique-and-revise experiments, will also be made publicly available upon publication. For precise implementation details, the exact model versions, baseline configurations, and prompt templates are comprehensively outlined in Appendix §E and §C. Finally, a complete breakdown of the computational resources, GPU hours, and API usage required for dataset generation and model evaluation is provided in Appendix §E.1.

REFERENCES

Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18208–18218, 2022.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13242–13251, 2025.

Black Forest Labs. FLUX.2: Analyzing and enhancing the latent space of FLUX – representation comparison, 2025. URL https://bfl.ai/research/representation-comparison.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 632–642, 2015.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, et al. Hidream-i1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv preprint arXiv:2505.22705*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*, 2023.

Herbert H Clark and Susan E Brennan. Grounding in communication. 1991.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Omer Dahary, Yehonathan Cohen, Or Patashnik, Kfir Aberman, and Daniel Cohen-Or. Be decisive: Noise-induced layouts for multi-subject generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

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

Rongyao Fang, Chengqi Duan, Kun Wang, Linjiang Huang, Hao Li, Shilin Yan, Hao Tian, Xingyu Zeng, Rui Zhao, Jifeng Dai, et al. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*, 2025.

Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Yang Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. *arXiv preprint arXiv:2309.17102*, 2023.

Peter Gardenfors. *Conceptual spaces: The geometry of thought*. MIT press, 2004.

Zorik Gekhman, Jonathan Herzig, Roee Aharoni, Chen Elkind, and Idan Szpektor. Trueteacher: Learning factual consistency evaluation with large language models. *arXiv preprint arXiv:2305.11171*, 2023.

Google DeepMind. Image generation with Gemini | Gemini API | Google AI for developers. Online Documentation, 2026. URL `https://ai.google.dev/gemini-api/docs/image-generation#gemini-image-editing`. Accessed: April 6, 2026.

Google Developers. Gemini API documentation: Image generation and editing. `https://ai.google.dev/gemini-api/docs/image-generation#gemini-image-editing`, 2026. Accessed: April 18, 2026.

Herbert P Grice. Logic and conversation. In *Speech acts*, pp. 41–58. Brill, 1975.

Qingdong He, Xueqin Chen, Chaoyi Wang, Yanjie Pan, Xiaobin Hu, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xiangtai Li, and Jiangning Zhang. Reasoning to edit: Hypothetical instruction-based image editing with visual reasoning. *arXiv preprint arXiv:2507.01908*, 2025.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. True: Re-evaluating factual consistency evaluation. *arXiv preprint arXiv:2204.04991*, 2022.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20406–20417, 2023.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8362–8371, 2024.

Ekaterina Iakovleva, Fabio Pizzati, Philip Torr, and Stéphane Lathuilière. Specify and edit: Overcoming ambiguity in text-based image editing. *arXiv preprint arXiv:2407.20232*, 2024.

Jaeseok Jeong, Junho Kim, Yunjey Choi, Gayoung Lee, and Youngjung Uh. Visual style prompting with swapping self-attention. *arXiv preprint arXiv:2402.12974*, 2024.

Liya Ji, Chenyang Qi, and Qifeng Chen. Instruction-based image editing with planning, reasoning, and generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17506–17515, 2025.

Bohan Jia, Wenxuan Huang, Yuntian Tang, Junbo Qiao, Jincheng Liao, Shaosheng Cao, Fei Zhao, Zhaopeng Feng, Zhouhong Gu, Zhenfei Yin, et al. Compbench: Benchmarking complex instruction-guided image editing. *arXiv preprint arXiv:2505.12200*, 2025.

KJ Joseph, Prateksha Udhayanan, Tripti Shukla, Aishwarya Agarwal, Srikrishna Karanam, Koustava Goswami, and Balaji Vasan Srinivasan. Iterative multi-granular image editing using diffusion models. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 8107–8116, 2024.

Gunther Kress and Theo Van Leeuwen. *Reading images: The grammar of visual design*. Routledge, 2020.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9332–9346, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.750. URL https://aclanthology.org/2020.emnlp-main.750/.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12268–12290, 2024.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Hongyu Li, Manyuan Zhang, Dian Zheng, Ziyu Guo, Yimeng Jia, Kaituo Feng, Hao Yu, Yexin Liu, Yan Feng, Peng Pei, et al. Editthinker: Unlocking iterative reasoning for any image editor. *arXiv preprint arXiv:2512.05965*, 2025.

Victor Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Y Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.

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*, pp. 366–384. Springer, 2024.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025.

Vivian Liu and Lydia B Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proceedings of the 2022 CHI conference on human factors in computing systems*, pp. 1–23, 2022.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516, 2024.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 1906–1919, 2020.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, 2023.

Donald A Norman. Some observations on mental models. In *Mental models*, pp. 7–14. Psychology Press, 2014.

OpenAI. *Image generation*, December 2025. URL `https://developers.openai.com/api/docs/guides/image-generation`. GPT-Image-1.5 Model Documentation.

Jonas Oppenlaender. A taxonomy of prompt modifiers for text-to-image generation. *Behaviour & Information Technology*, 43(15):3763–3776, 2024.

Yuandong Pu, Le Zhuo, Songhao Han, Jinbo Xing, Kaiwen Zhu, Shuo Cao, Bin Fu, Si Liu, Hongsheng Li, Yu Qiao, et al. Picabench: How far are we from physically realistic image editing? *arXiv preprint arXiv:2510.17681*, 2025.

Yusu Qian, Eli Bocek-Rivele, Liangchen Song, Jialing Tong, Yinfei Yang, Jiasen Lu, Wenze Hu, and Zhe Gan. Pico-banana-400k: A large-scale dataset for text-guided image editing. *arXiv preprint arXiv:2510.19808*, 2025a.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. Gie-bench: Towards grounded evaluation for text-guided image editing. *arXiv preprint arXiv:2505.11493*, 2025b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation, 2025. *URL https://arxiv. org/abs/2509.20427*, 6.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8871–8879, 2024.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Yoad Tewel, Rinon Gal, Dvir Samuel, Yuval Atzmon, Lior Wolf, and Gal Chechik. Add-it: Training-free object insertion in images with pretrained diffusion models. *arXiv preprint arXiv:2411.07232*, 2024.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5238–5248, 2022.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8228–8238, 2024.

Ye Wang, Ruiqi Liu, Jiang Lin, Fei Liu, Zili Yi, Yilin Wang, and Rui Ma. Omnistyle: Filtering high quality style transfer data at scale. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7847–7856, 2025a.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025b.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37: 128374–128395, 2024.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, et al. Long-form factuality in large language models. *Advances in Neural Information Processing Systems*, 37:80756–80827, 2024.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025a. URL `https://arxiv.org/abs/2508.02324`.

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

Siwei Yang, Mude Hui, Bingchen Zhao, Yuyin Zhou, Nataniel Ruiz, and Cihang Xie. Complex-edit: Cot-like instruction generation for complexity-controllable image editing benchmark. *arXiv preprint arXiv:2504.13143*, 2025.

Mingde Yao, Zhiyuan You, King-Man Tam, Menglu Wang, and Tianfan Xue. Photoagent: Agentic photo editing with exploratory visual aesthetic planning. *arXiv preprint arXiv:2602.22809*, 2026.

Chun-Hsiao Yeh, Yilin Wang, Nanxuan Zhao, Richard Zhang, Yuheng Li, Yi Ma, and Krishna Kumar Singh. Beyond simple edits: X-planner for complex instruction-based image editing. *arXiv preprint arXiv:2507.05259*, 2025.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 26125–26135, 2025.

Tianshuo Yuan, Yuxiang Lin, Jue Wang, Zhi-Qi Cheng, Xiaolong Wang, Jiao GH, Wei Chen, and Xiaojiang Peng. Flexedit: Marrying free-shape masks to vllm for flexible image editing. *arXiv preprint arXiv:2408.12429*, 2024.

J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. Why johnny can't prompt: how non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–21, 2023.

Bohan Zeng, Ling Yang, Jiaming Liu, Minghao Xu, Yuanxing Zhang, Pengfei Wan, Wentao Zhang, and Shuicheng Yan. Editworld: Simulating world dynamics for instruction-following image editing. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 12674–12681, 2025.

Huixuan Zhang and Xiaojun Wan. Re-thinking the automatic evaluation of image-text alignment in text-to-image models. *arXiv preprint arXiv:2506.08480*, 2025.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

APPENDIX

A TAXONOMY AND DEFINITION

A.1 CLARIFICATIONS

Context Image as part of the definition. The importance of the context image $\mathcal{I}_c$ lies in its role as the anchor for the instruction p, as the degree of generative guesswork is a function of the image-text pair rather than the text alone. An instruction can be interpreted differently depending on the visual environment; for instance (see Figure 8), the prompt "make it more formal" might require adding a suit to a person in a Physical context, whereas in an Artistic context, it could involve shifting the image composition toward symmetry. Furthermore, the visual state of $\mathcal{I}_c$ determines the $eDoF$ level by defining the available constraints. Consider the instruction "adjust for a younger audience". In a photo of a serious, high-end restaurant, the model faces high $eDoF$ as it must creatively synthesize sweeping changes to the lighting, decor, and plating to match the concept. Conversely, in an image of a dark and grayscale coloring book page, the same instruction becomes relatively explicit with low $eDoF$ because the visual context clearly points to applying a vibrant, bright color palette. Thus, the context image provides the necessary grounding to resolve underspecified intent and quantify the model's reasoning.

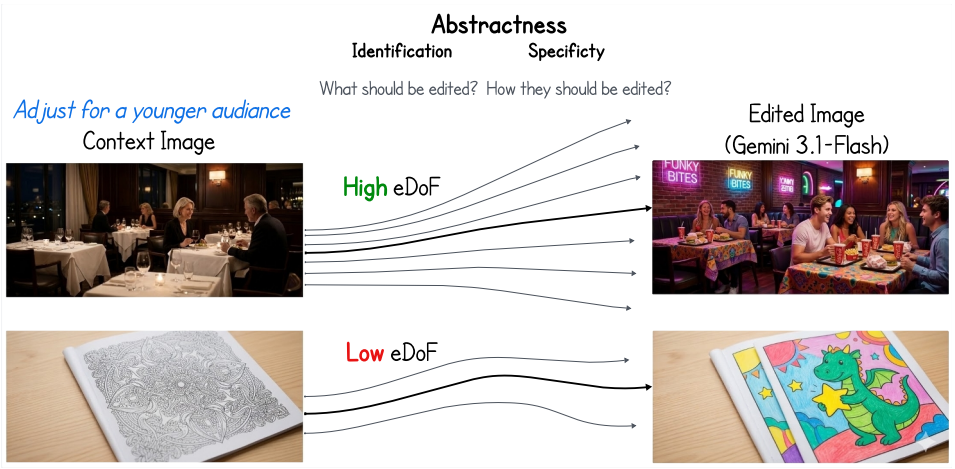


Figure 8: Context-Dependent $eDoF$. The same instruction, "Adjust for a younger audience", requires varying levels of generative guesswork depending on the context image $\mathcal{I}_c$. In the High $eDoF$ case (top), the model must navigate a broad probability space to resolve both Identification (what to change) and Specificity (how to change it). In the Low $eDoF$ case (bottom), the visual environment provides explicit constraints, narrowing the model's interpretive path.

Conceptualizing Editing Degrees of Freedom. Editing Degrees of Freedom (eDoF) represents the continuous spectrum of valid interpretations an editing instruction can evoke. Quantifying this spectrum is inherently challenging, as assigning a strict metric to human imagination and semantic ambiguity quickly transitions from a computational problem into a philosophical one. Given this difficulty, it is more practical to conceptualize eDoF in ranges, specifically by mapping transitions from deterministic one-to-one instructions to open-ended one-to-many (or one-to-few) instructions based on the number of plausible outcomes. Explicit instructions represent the lowest eDoF, dictating direct and unambiguous changes. Still within the one-to-one regime are implicit instructions, which, rather than requiring abstract reasoning, rely on external computer vision capabilities like object detection or external knowledge retrieval to resolve the referent. Conversely, abstract instructions represent the highest eDoF, encompassing one-to-many scenarios such as spatial referencing, artistic transformations, and style transfer, as demonstrated in Figure 9. While this work primarily focuses on the semantic and contextual abstractness of human intent in image editing, expanding this framework to fully capture these broader categories of interpretation remains an exciting avenue for future research.

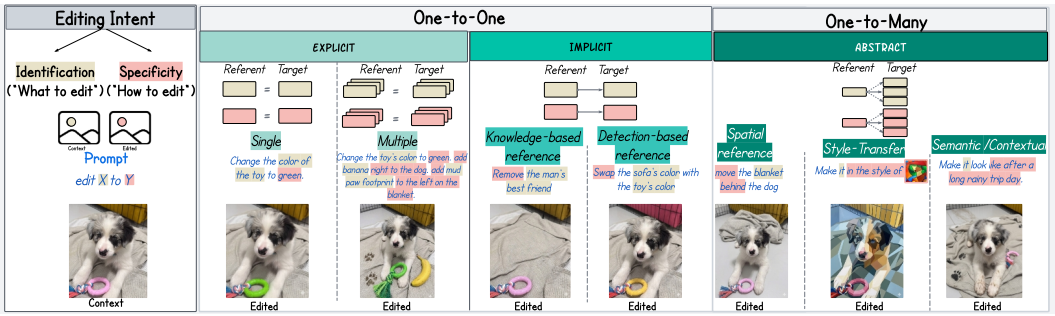


Figure 9: Expanded Cases of Editing Intent.

eDoF and Domains. Crucially, a sample's position on the eDoF scale is inherently tied to its semantic domain (Gardenfors, 2004). Physically bounded edits, such as changing the season to winter, usually carry lower interpretive variance and naturally align with lower $eDoF$ scores. Edits involving complex social or emotional shifts, such as making a scene appear formal, introduce vastly higher interpretive freedom, placing them at the upper limit of the scale. Furthermore, this level of abstractness is strictly independent of spatial resolution. Instructions can manifest as highly abstract local modifications (e.g., "make the shirt more formal") or completely explicit global transformations (e.g., "reduce brightness by 1 in each color channel").

From the Theoretical Consensus to the Tractable Practice. The practical implementation of this taxonomy is challenged by the continuous visual space abstract editing spans, making a consensus-based definition difficult to quantify. Beyond the mathematical intractability of mapping an infinite interpretation set, consensus is fractured by multi-level subjectivity, ranging from human cultural variance to the architectural biases of VLM judges and the spatial constraints of different generative seeds. It also suffers from the ambiguity of granularity, which directly impacts our core axes: Identification becomes a recursive problem, where the target can shift from a general entity (a "dog") to hyper-local components (the "pose of its ear") without a natural stopping point. Similarly, Specificity expands into second-order complexity ($O(n^2)$) when an edit targets the relational interaction between entities (e.g., "make them look in love") rather than an isolated attribute, requiring the model to identify and modify the mutual dependency between multiple entities. To navigate this, our framework adopts a "rough-cut" approach based on three distinct methodological choices. First, we prioritize precision over recall in our evaluation; rather than attempting to account for the exhaustive recall of all valid creative interpretations, we focus on the precision of a model's editing, ensuring the chosen output is logically grounded and verifiable. Second, we utilize a high-reasoning VLM as a proxy for a mutual judge across all experiments, employing fixed generative seeds within the image editing models to minimize stochastic variance. Third, we constrain the problem of granularity by grounding both identification and specificity to the immediate requirements of the visual context, effectively limiting the scope of modification to what is necessary for a successful edit.

Implicit vs Low Abstractness. We utilize two dimensions to differentiate between them. An implicit instruction refers to a target entity or transformation indirectly, but still maintains a one-to-one mapping with the target (similar to an explicit instruction, but requiring additional reasoning). For example, in Table 1, "Remove the farthest tiger from the water" uniquely identifies a single tiger, but requires spatial and relational processing to determine which one. In contrast, an explicit instruction would directly name the specific tiger, while an abstract instruction would be open to interpretation (e.g., "Edit the tiger to look scarier").

A.2 Potential Abstractness Expansions

While the specificity axis is central to framing abstract image edits, it alone provides an incomplete definition.

Causality Considerations. Specificity can be framed as a directional space where a single edit triggers a sequential cascade of subsequent changes. This "edit cascade" is defined by the causal rela-

tionships expected between explicit atomic edits. For example, the instruction "make an earthquake" in a crowded street scene implies a direct causal trajectory: cracked ground $\rightarrow$ structural damage $\rightarrow$ panicked pedestrians. Expanding the Specificity axis to include these trajectories allows for a more accurate formalization of abstract editing. However, this approach introduces significant challenges inherent to causality itself, such as determining the ground truth order of effects in multi-turn edits and distinguishing between correlation and true causation. While we believe integrating causality is essential for a complete understanding of the field, it remains beyond the scope of this work; we hope to inspire future research into this intersection.

Spatial Considerations. The Specificity axis is further influenced by a critical challenge that was previously addressed Tewel et al. (2024): determining where to insert a new object, or where to move an existing one, and at what scale relative to the surrounding context. This extends the problem beyond semantics into the realm of spatial reasoning. Even when an entity is explicitly named, its position and scale may remain entirely abstract, requiring the model to infer placement based on scene understanding. Specifically, we identify *abstract location*, where the model must determine a plausible placement for an object when the instruction is general, such as deciding where to "Add Snoopy", and abstract size, where the model must determine the appropriate scale of an object for actions like adding, replacing, or transforming. These two dimensions introduce significant spatial ambiguity into the editing process. While our test set includes physical domain categories such as size and point of view, the broader uncertainty regarding object placement remains beyond the scope of this work.

Relational Interaction Considerations. A more holistic evaluation would ideally account for the interactions between entities within a scene. However, this introduces substantial scalability and complexity issues; for n entities, the relational space expands at $O(n^2)$. Assessing the nuanced changes in these dependencies, such as spatial shifts or physical contact, is computationally demanding and difficult to quantify. Consequently, while these interactions are vital for scene coherence, evaluating them requires a careful balance between modeling the full relational web and maintaining a tractable evaluation framework.

Generalizing to Stylistic Intents. While artistic style transfer (Jeong et al., 2024; Wang et al., 2025a; Yao et al., 2026) involves global transformations, these works primarily evaluate familiarity with pre-existing visual distributions, such as specific artistic movements. In contrast, ABSTRACTEDIT focuses on situational intents that do not map to a single stylistic category. For example, a style prompt dictates a uniform aesthetic change, whereas an abstract prompt like "make the dog look like it just returned from a long rainy trip" requires the model to autonomously deduce and execute specific semantic edits, such as adding mud or wetting fur. Consequently, style transfer represents a constrained form of abstraction relying on specific external knowledge, while our work addresses the open-ended interpretation of complex human intent. Integrating elements of artistic reference with open-ended editing instructions remains a compelling direction for future work, potentially bridging the gap between concept-driven stylization and intent-driven content modification.

B DATASET STATISTICS & CURATION DETAILS

Domain and Subcategory Distribution of ABSTRACTEDIT. Fig. 10 and Fig. 11 details the composition of the ABSTRACTEDIT benchmark across its four primary domains and their fine-grained categories. The quantitative distribution: Logical (186), Social (141), Physical (101), and Emotional (41), is intentionally designed to reflect the varied cognitive demands required for abstract image editing. The **Logical** domain is the most heavily represented, driven primarily by the *Insertion Goal* (89) and *Commonsense Goal* (84) subcategories. This prioritization is motivated by the fact that functional, deductive reasoning, inferring *what* specific objects or state changes are required to fulfill an abstract need (e.g., "prepare for rain" resolving to "add an umbrella"), forms the foundational backbone of abstract human instruction. The **Social** domain follows closely, including complex human-centric categories such as *Culture* (47), *Socio-Economic* (44), and *Role* (37). The strong emphasis on this domain challenges the model's capacity to translate abstract sociological concepts into concrete visual markers (e.g., clothing, environment), a task that requires deep semantic grounding often absent in standard diffusion models. The **Physical** domain focuses on abstract intents that resolve into structural modifications, such as shifting *POV/Composition* (37) or *Size* (23) to alter

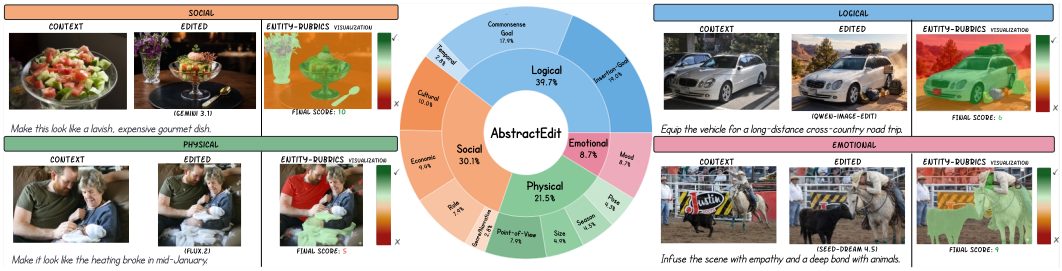


Figure 10: Composition and Evaluation of the ABSTRACTEDIT Benchmark. The composition of the ABSTRACTEDIT benchmark is illustrated across four primary domains and 12 subcategories in the middle panel. Surrounding this distribution are representative samples from each domain, featuring context images paired with candidate edits produced by various models. Each output is overlaid with our ENTITY-RUBRICS automatic granular evaluation, providing a visual layout of performance that ranges from red (incorrect) to green (correct) at the entity level.

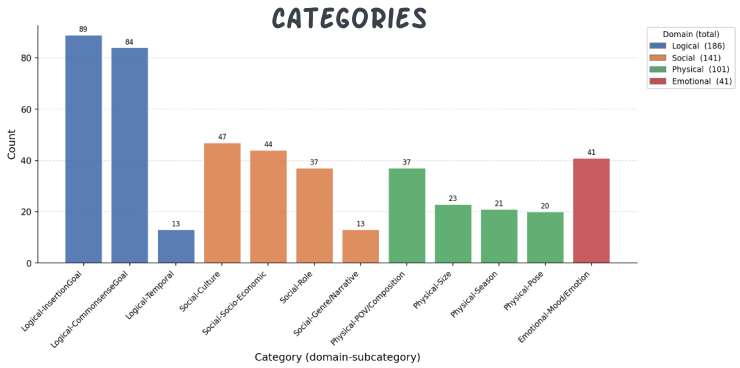


Figure 11: ABSTRACTEDIT distribution of categories.

the perceived focus or power dynamic of a scene. Finally, while the **Emotional** domain has the lowest sample count, its singular focus on *Mood/Emotion* (41) provides a critical test of atmospheric interpretation, requiring the model to manipulate global image semantics, such as lighting, weather, and color grading, to evoke subjective feelings. Together, this hierarchical distribution ensures the benchmark rigorously tests a wide spectrum of abstraction, ranging from objective problem-solving to subjective atmospheric translation. See Table 3 for the domain distribution of the train set (4000 samples).

Table 3: Category Distribution in ABSTRACTEDIT Training Set.

Domain	**Count**	**Percentage (%)**
Social	1,470	35.7%
Physical	1,470	35.7%
Logical	882	21.4%
Emotional	294	7.1%
Total	**4,116**	**100.0%**

Context Image Composition and Density. Figure 13 illustrates the semantic diversity of the base context images selected from the OpenImages v7 dataset (Kuznetsova et al., 2020). The top distribution demonstrates the variety of distinct entity types present within individual samples; the majority of images contain between 2 and 12 unique categories, peaking prominently at 5 distinct entity types [24]. This guarantees that the benchmark evaluates conceptual editing within compositionally rich environments rather than simplified, single-object backgrounds. The frequency

analysis of these entities (middle) reveals a strong prevalence of human-centric and everyday subjects, heavily dominated by categories such as "Clothing" (254 occurrences) [19, 32], "Person" (232 occurrences) [34, 40], and "Human body" (196 occurrences)[37, 41]. This grounds the dataset in relatable, real-world scenarios, which is a prerequisite for supporting the complex interpersonal and structural interactions required by the Social and Physical evaluation domains. Finally, the instance count distribution (bottom) highlights the high visual density of the chosen images, with most samples containing between 15 and 35 distinct bounding box instances [106, 123], and a notable peak at 16 instances [63, 107]. This deliberate level of visual clutter introduces realistic scene distractors, ensuring that models must not only interpret the abstract intent but also accurately localize the edit without inadvertently corrupting a dense surrounding context.

Abstract and Explicit Prompt Lengths. Figure 12 presents the distribution of prompt lengths, measured in words, for both abstract and explicit instructions across the four evaluation domains (Emotional, Logical, Physical, and Social) in ABSTRACTEDIT. The most prominent observation is the stark contrast in length and variance between the two prompt types, which quantitatively illustrates the "semantic gap" inherent in abstract editing. Abstract instructions are consistently brief, averaging roughly 10 to 11 words as was limited in the curation process (< 15 words), directly reflecting their high-level, underspecified nature. Conversely, the paired explicit instructions are significantly longer and exhibit substantial variance, acting as comprehensive, step-by-step blueprints for execution. Among the domains, the **Social** category requires the most extensive explicit instructions (averaging 209 words), likely due to the complexity of describing interpersonal dynamics and contextual scene modifications. In contrast, the **Logical** domain requires the fewest words on average (143 words) to explicitly ground the intended edits. This massive length discrepancy highlights the heavy lifting required, by either the human annotator or the model's text encoder, to translate a concise abstract concept into a complete set of executable visual steps.

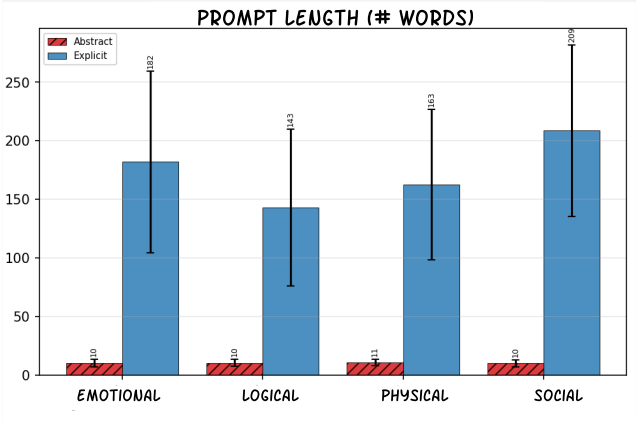


Figure 12: Explicit vs Abstract Prompt Length.

Personas. To ensure instruction diversity, we employ a high-dimensional persona sampling strategy that simulates a wide array of human backgrounds and behaviors (see the prompt box). By sampling features such as age, country, profession, and personality as independent variables, the model generates instructions spanning different cultural contexts and linguistic styles. This approach captures the intersection of a user's technical ability, ranging from AI enthusiasts to tech-phobic individuals, with their specific creative motivations like artistic expression or practical utility. By grounding each request in a unique combination of visual language and hobby, the system prevents mode collapse and produces a dataset that reflects the authentic variety of real-world image editing needs. This feature space yields 10^{10} unique persona combinations.

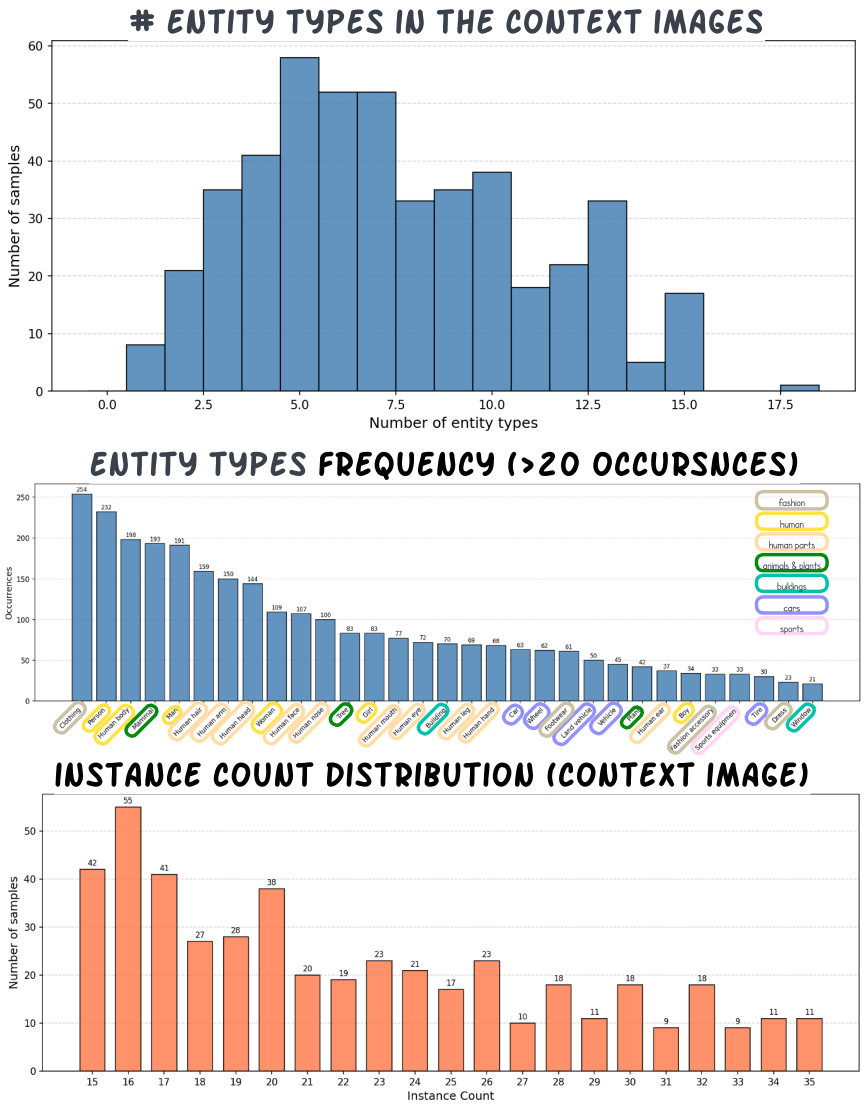


Figure 13: Distribution of Common Entities in ABSTRACTEDIT Context Images (OpenImages).

Automatic Curation pipeline of ABSTRACTEDIT Curation

```
random_paersona = persona_prompt_generator()
system_prompt_2 = f"""

{random_paersona} Your goal is to generate creative and detailed
    editing
instructions for an image based on a set of approved categories.
    Your goal
is also to translate creative, high-level ideas (**Abstract Prompts
    **) into
specific, technical, and deterministic steps (**Explicit
    Instructions**).
Each response must be sampled at random from the distribution.

## CONTEXT
- **Image Entities**: You will be given a list of entities detected
    in the
image: {', '.join(entities)}. You can detect additional entities if
    you
think they are relevant for the edit, but you cannot remove any of
    the
provided entities.
- **Relevant Categories**: You will be given a list of categories,
    each with
an example of a high-level creative prompt.

## YOUR TASK
Your overall task is to generate editing instructions for a list of
    categories.
This list includes all of the 'Relevant Categories' provided to you,
     **PLUS one
new, creative category of your own invention (the category name will
     start
with NEW_ prefix)** that you believe is a good fit for the image.
For **EACH** category in this combined list (both provided and self-
    generated),
you must generate a complete set of editing instructions by
    following these steps:

1. **Create an Abstract Prompt (The "What")**: Write a new, creative
    , and
high-level prompt inspired by the category's theme. This defines the
     overall
'vibe' of the edit. The instruction must be open to interpretation
    in terms of
How exactly to achieve it, I.e, different people will interpret it
    differently.
2. **Probablity Sampling**: Ensure that each generated response is
    sampled
randomly from the distribution.
3. **Define Entity-Specific Edits (The "How")**:
    - **Discovery**: Actively identify more relevant entities in the
         image or
    context that were not provided in the {entities} list. Include
        them in
    your explicit list.
        For each relevant 'Image Entity', determine its role in the
            edit. This
        is the technical process. It is precise, measurable, and
            leaves no
        room for interpretation. It focuses on concrete properties:
```

```
        Write a clear, precise, and actionable instruction for the
            entity with
        specific guidance. Keep it blank if no change is needed to
            the entity
        to advance the alignment to abstract prompt.
        -- **Granularity**: Relate to the relevant entity components
             (e.g., for
        a 'person', specify edits for their clothing, hair, and
            posture in
        separate sentences).
        -- **Density**: Aim for a high density of edits. Each
            sentence must
        contain exactly ONE technical change. Multiple sentences can
             be
        generated for the same entity as long as they refer to
            different
        components or aspects of edit of the entity.
        -- **Technical Properties**: Use specific values: "Increase
            roughness
        to 80%," "Shift hue to #228B22," "Reduce focal depth to
            2.0," or
        "Add 15% film grain."
4. **Define a General Instruction**: Write instructions that
    describes any
global changes to the image's overall e.g., style, lighting, mood,
    or color
palette. This applies to the entire scene.
5. **Insertion/Removal Entities**: You may suggest new entities to
    be
inserted in the image to better align with the abstract prompt as an
     explicit
instruction specifying the entity and its placement.
    These should be realistic and contextually appropriate. If no
    insertions/removals are needed, leave this blank.
    For each insertion, you MUST provide coordinates or relative
        placement
    (e.g., "Place a [Object] 10cm to the left of [Entity] at a 30-
        degree angle").

## OUTPUT REQUIREMENTS
- You **must** process every category provided in the input.
- You MUST return your complete analysis of all categories by
    calling the
`DiverseImageAnalysisResult_abstractExplicit` function.
- You must ensure diversity in your generated prompts by sampling
    randomly
from the distribution.
- Each abstract prompt must not be longer than 15 words, and has to
    keep its
abstractness while being specific to the image content and user
    context.
- Each abstract prompt has to look like user-generated by the
    persona mentioned
in the beginning, suitable for them using a mobile phone on the go,
    trying to
quickly convey a creative idea.
- Your generated **explicit instructions** and **general instruction
    ** must
**NEVER** use subjective or interpretive words. only specific
    technical clear
words for specific change.
- High-density output is mandatory: provide as many specific, atomic
instructions as possible per entity.
```

```
- Avoid adjectives like "whimsical", "vast", "vibrant", which are
    not
user-common.

**Begin your analysis for the following categories. Examples are
    provided only
for inspiration. Do not use them as templates. Follow each category
    guidelines:**
{relevant_cats_with_examples_str}
"""
```

Persona-Driven Instruction Diversity

Feature Space $\mathcal{F}$:

Age: Gen Alpha, Gen Z, Millennial, Gen X, Boomer, Elderly, Teenager, Toddler-parent, Young Adult, Retiree
Country: Japan, Brazil, Norway, Kenya, India, USA, France, Australia, Egypt, Mexico
Gender: Non-binary, Female, Male, Agender, Genderfluid, Trans-masculine, Trans-feminine, Bigender, Queer, Undisclosed
Hobby: Urban Exploration, Gardening, Cyber-gaming, Birdwatching, Cosplay, Minimalist Design, Oil Painting, Skateboarding, Baking, Street Photography
Profession: Florist, Data Scientist, History Teacher, DJ, Architect, Mechanic, Social Media Influencer, Nurse, Student, Chef
Tech Skill: Smartphone novice, Power user, Professional Editor, Casual social poster, Technophobic, AI enthusiast, Tablet artist, Analog lover, Gamer, Software Dev
Visual Lang: Cinematic, Abstract, High-Contrast, Pastel, Gritty/Industrial, Surreal, Symmetrical, Over-saturated, Muted/Desaturated, Low-Poly
Personality: Sarcastic, Highly optimistic, Melancholic, Efficient/Direct, Whimsical, Detail-oriented, Impatient, Poetic, Skeptical, Cheerful
Name: Yuki, Mateo, Astrid, Akachi, Priya, Jax, Elodie, Liam, Zaynab, Santiago
Motivation: Artistic expression, Quick fix, Social media clout, Memory preservation, Creating a gift, Practical utility, Boredom, Satire, Visual storytelling, Professional portfolio

System Prompt Template $\mathcal{S}$:

*"You are an expert image editing assistant. You currently simulate how a user named **{name}** from **{country}**—a **{age}** **{gender}** who works as a **{profession}** and enjoys **{hobby}**—would generate creative abstract editing instructions. This user is a **{tech_skill}** motivated by **{motivation}**. They describe things using a **{visual_lang}** visual language and have a **{personality}** personality."*

Domains and Categories: Guidelines

Emotional-Mood/Emotion: Fundamentally alters the emotional atmosphere of the image through changes in color, light, and context, or of people/animals emotions. *Requirement: Use different moods / emotions such as joyful, peaceful, sad, angry, fearful, surprised, curious, proud, lonely, hopeful.*
Physical-Material: Manipulates the sensory surface qualities and physical materials of objects. *Requirement: Must contain objects with distinct, changeable materials such as smooth, rough, glossy, soft, grainy, metal, wood, glass, fabric, stone.*
Physical-Disaster/Event: Introduces a transformative, chaotic, or disastrous event into the scene, even if not logical for the initial setting. *Requirement: Use both natural and human-made events such as earthquake, flood, wildfire, hurricane, volcanic eruption, war, protest, explosion, building collapse, oil spill.*
Physical-POV/Composition: Changes the camera angle, framing, or composition to alter the viewer's narrative relationship to the scene.
Physical-Pose: Alters a subject's body language to convey a different emotion, status, or intent. *Requirement: Must contain a 'person', 'animal', or anthropomorphic figure.*
Physical-Season: Changes the time of year, including atmospheric conditions and indirect clues for indoor scenes.
Physical-Size: Alters the scale of an exisiting object or subject in the image, without mentioning a very specific size.
Social-Culture: Transports the subjects or scene into a different cultural context, affecting fashion, architecture, or historical period. *Requirement: Best with people, fashion, and architecture that can be re-interpreted. Use different nationalities and cultures [e.g. Thai, Indian, Nigerian, Moroccan, Peruvian, Greek, Scottish, Ethiopian, Mexican, Maori, Mongolian, Egyptian, Brazilian, Norwegian].*
Social-Environment/Setting: Changes the primary location to a new environment with a distinct character or narrative purpose.
Social-Genre/Narrative: Re-contextualizes the scene to take place during a specific social gathering, ceremony, or ritual. *Requirement: Use different events [e.g. wedding, birthday, festival, protest, parade, funeral, carnival, market, concert, sports event, religious ceremony].*

It can apply a distinct visual philosophy or artistic movement or specific artistic style to the entire image.
Social-Role: Recasts a person into a new professional or archetypal role by altering their appearance and context. *Requirement: Must contain a 'person'. Use different professions [e.g. firefighter, chef, astronaut, doctor, teacher, artist, musician, athlete, scientist, pilot, engineer, farmer, dancer, writer, actor].*
Social-Socio-Economic: Adjusts the visual narrative of wealth, poverty, or societal status through changes to the environment and personal effects. *Requirement: Use states such as homeless, wealthy, poor, middle-class, luxurious, impoverished, unemployed, white-collar, upper-class, working-class.*
Logical-Temporal: Manipulates the timeline by showing the scene at a different point in its story (past, future, before, after). *Requirement: The image must imply an action or process that can be shifted in time.*
Logical-CommonsenseGoal: Creatively disrupts the scene's logical purpose or common-sense state. *Requirement: The scene must have a clear function that can be completed or subverted, without mentioning how to do it. Do not mention how to change, focus on the goal.*
Logical-InsertionGoal: Adds a general object or subject (something, an object, an equipment, etc) that has a clear, logical purpose within the scene. *Requirement: Ask to add **something**, not a specific object, DO NOT mention what to add, the focus is on the goal. The goal has to relate to the overall scene purpose.*

C ENTITY-RUBRICS EVALUATION

Evaluation Prompt Templates. Our ENTITY-RUBRICS evaluation employs a two-stage prompting strategy. The first call establishes a theoretical baseline of expectations from the context image alone (see Figure C), while the second call performs a comprehensive comparison between the context and edited images against those expectations (Figure C). The output schema is presented in Figure C.

Call 1: Entity Expectations Prompt

Role: Expert Visual Analyst
Inputs: Instruction (p), Context Image ($\mathcal{I}_c$).
Analysis Steps:

- **Step 1: Define Categories.** Identify *Things* (distinct people/animals/objects; multiple instances must be differentiated with unique IDs and parts), *Stuff* (general environmental and structural elements), and *Global* (abstract image properties like luminance or spatial complexity).
- **Step 2: Generate Element List.** Create a comprehensive master list of all entities and parts present in $\mathcal{I}_c$ relevant to the edit.
- **Step 3: Per-Element Evaluation.** Assign each entity its group and evaluate its editing expectation:
  - `EXPECTED_CHANGE`: Highly expected to change. Obvious and necessary changes that everyone would expect (e.g., adding snow for a "winter scene").
  - `OPTIONAL_CHANGE`: May change to improve alignment without hurting identity preservation. Not everyone would think of it, but it is a positive creative interpretation (e.g., adding an umbrella for "make it rainy").
  - `EXPECTED_PRESERVATION`: Should be preserved and NOT change. Irrelevant to the instruction.

Output Format: JSON object containing `entity_expectations`.

Call 2: Comprehensive Evaluation Prompt

Role: Micro-Detail Observer and Expert Visual Analyst
Inputs: Instruction (p), Context Image ($\mathcal{I}_c$), Edited Image ($\mathcal{I}_e$), Entity Expectations.
Analysis Steps:

- **Steps 1 & 2: Define & Generate List.** Expand the master list by scanning both images to include new elements introduced in $\mathcal{I}_e$.
- **Step 3: Per-Element Evaluation.** For each item, determine:
  - *Group & Change Description:* Describe the literal visual delta between the images.
  - *Edit Action:* Categorize the change type (e.g., `POSITION`, `STYLE_TRANSFER`).
  - *Reality vs. Expectation:* Compare actual change to the baseline expectation:
    - `GOOD_EXPECTED_CHANGE`: Changed, and needed to. Explicitly requested or mandatory.
    - `BAD_EXPECTED_CHANGE`: Changed and needed to, but actively hurts alignment (failure of core intent).
    - `GOOD_OPTIONAL_CHANGE`: Changed without being obvious, but actively improves alignment (positive creative interpretation).
    - `BAD_OPTIONAL_CHANGE`: Changed without being obvious, but actively hurts alignment or preservation (negative creative interpretation).
    - `GOOD_EXPECTED_PRESERVATION`: Did not change, and this is the correct outcome.
    - `BAD_EXPECTED_PRESERVATION`: Did not change, but should have to successfully fulfill the instruction.
- **Step 4: Global Assessment.** Evaluate missing changes, over-editing, and overall narrative coherence.
- **Step 5: Overall Score Synthesis.** Synthesize an overall 1–10 alignment score:
  - **10 (Perfect Alignment):** Perfectly embodies the instruction with all required changes executed flawlessly. Essence is preserved; narrative is cohesive.
  - **8–9 (Strong Alignment):** Strongly reflects the instruction with most required changes executed well. Minor imperfections exist but do not detract overall.
  - **6–7 (Moderate-High Alignment):** Good level of alignment. Most required changes executed, but some minor imperfections or missed requirements exist.
  - **4–5 (Moderate Alignment):** Some required changes executed, but noticeable imperfections, missed requirements, or signs of over-editing. Disjointed narrative.
  - **2–3 (Low Alignment):** Many required changes missing; significantly different from what was requested. Substantial over-editing or preservation issues.
  - **1 (Very Low Alignment):** Fails to align. Required changes mostly missing; output unrelated to instruction. Incoherent narrative and severe preservation issues.

Output Format: JSON containing `entity_evaluations` and `global_evaluations`.

Entity-Rubrics Evaluation: Output Schema

```
{
  "entity_evaluations": {
    "<entity_name>": {
      "group": "things" | "stuff" | "global",
      "change_description": "string",
      "change_occured": boolean,
      "edit_action": "NO_CHANGE" | "OBJECT_PRESENCE" |
                     "OBJECT_COUNT" | "POSITION" |
                     "TRANSFORM" | "POSE" | "VIEWPOINT" |
                     "COLOR" | "TEXTURE" | "LIGHTING" |
                     "STYLE_TRANSFER" | "ATTRIBUTE" |
                     "STATE" | "OTHER",
       "edit_expectation": "EXPECTED_CHANGE" |
                           "OPTIONAL_CHANGE" |
                           "EXPECTED_PRESERVATION"
      "edit_execution": "GOOD_EXPECTED_CHANGE" |
                        "BAD_EXPECTED_CHANGE" | ...,
      "entity_edit_rationale": "string",
      "entity_overall_score": integer [1-10],
    }
  },
  "global_evaluation": {
    "missing_changes": boolean,
    "over_editing": boolean,
    "overall_narrative_coherence": boolean,
    "final_rank": integer [1-10],
    "final_rationale": "string"
  }
}
```

Entities Granularity. When defining entities, whether they are things, stuff (i.e., background elements like sky, floor, or walls), or global features, there is a significant hierarchy of resolution to consider. For instance, a “dog” can be further decomposed into its mouth, ears, and paws, extending even further to specific attributes like the “left canine” or the “tongue”. The depth of granularity required is highly context-dependent, as not every resolution is relevant to a specific edit. While an instruction for a man to move from "standing" to “sitting” may not require detailing individual body parts for instruction-following alignment, an edit where a man’s mood changes from sad to happy necessitates explicit references to the mouth or eyes. Consequently, both our dataset curation for explicit instruction generation and our evaluation framework prioritize high density and granularity. As presented in the prompt box B, we define Granularity as the requirement to relate to relevant entity components, such as specifying edits for a person’s clothing, hair, and posture in separate sentences, and we define Density as the maintenance of a high concentration of edits by ensuring each sentence contains exactly one technical change, allowing for multiple sentences to be generated for the same entity as long as they refer to distinct components or different aspects of the transformation.

Over-editing and Under editing. Over-editing and under-editing are evaluated in the third stage as part of the main criterion: ’whether it met the expected level of change’. After the entity-based analysis, the VLM aggregates the boolean indicators of missed-changes or over-editing that were expected. We use boolean values to avoid bias toward the number of entities in the image and then average it over the entire dataset.

Role of Explicit Instructions in the Evaluation Pipeline It is important to emphasize that the ENTITY-RUBRICS evaluation framework does not rely on explicit instructions during the assessment phase. Rather, explicit instructions are utilized solely during the data curation process to enable controlled experiments that compare abstract instruction-following against paired explicit baselines.

The ENTITY-RUBRICS framework analyzes each instruction entirely independently. The resulting scores are non-dependent and are presented side-by-side (as shown in Section 6, Table 2). Specifically, the explicitly paired instructions serve as an anchor for comparison; this ensures that the evaluation of a model's ability to handle complex, abstract instructions is not confounded by its baseline capability to execute simpler, explicit modifications.

D HUMAN EVALUATION

Human Evaluation Questionnaire and Guidelines To rigorously validate our automated evaluation ENTITY-RUBRICS, and assess image editing model performance, we conducted a comprehensive human evaluation study using Amazon Mechanical Turk (AMT). To ensure high-quality and reliable annotations, we restricted the pool to fluent English speakers who successfully passed a qualification task. Ultimately, a curated pool of eight highly qualified workers participated in the study, with each specific editing task being evaluated by three independent annotators to establish consensus. Workers were compensated fairly at a rate of 0.35$ per completed task (which takes approximately 1-2 minutes). As illustrated in Figure 15, the evaluation interface presents the context and edited images side-by-side alongside a structured, two-part questionnaire. Part A focuses on Instruction Following, requiring annotators to rate the overall prompt alignment on a 1–5 scale (Q1) and conduct a fine-grained, entity-level evaluation (Q2) to verify if specific elements were correctly changed or preserved. Part B assesses Preservation and Quality, measuring how accurately unintended areas were maintained (Q3) and scoring the overall realism and visual quality compared to the original image (Q4). To standardize the assessment and minimize subjective bias, workers were provided with detailed visual guidelines and solved examples. The study achieved moderate Inter-Annotator Agreement (quadratic weighted Fleiss' κ of 0.47[1]). As demonstrated in Figure 14, these examples explicitly trained annotators on how to correctly assign scores and how to manually add and evaluate new or missing entities (such as adding "magazine" or "people faces") that frequently emerge during complex, abstract image edits.

Human Evaluation and Ethics This study conducted image annotation tasks involving human crowd-workers. The protocol was determined to be exempt from formal IRB review based on the following criteria: (1) the research involved minimal risk to participants, (2) the tasks did not expose workers to sensitive, offensive, or harmful content, and (3) no Personally Identifiable Information (PII) was collected or stored, ensuring participant anonymity.

[1]IAA breakdown: instruction following 0.54, image preservation 0.61, per-entity alignment 0.41, and image quality 0.28.

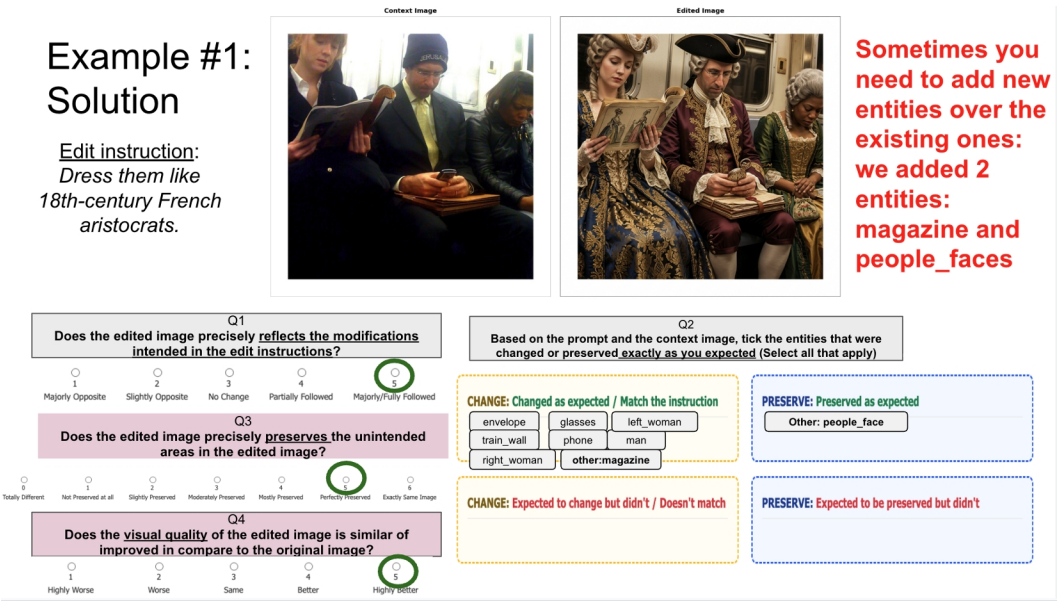


Figure 14: Annotation guideline example for Amazon Mechanical Turk workers.

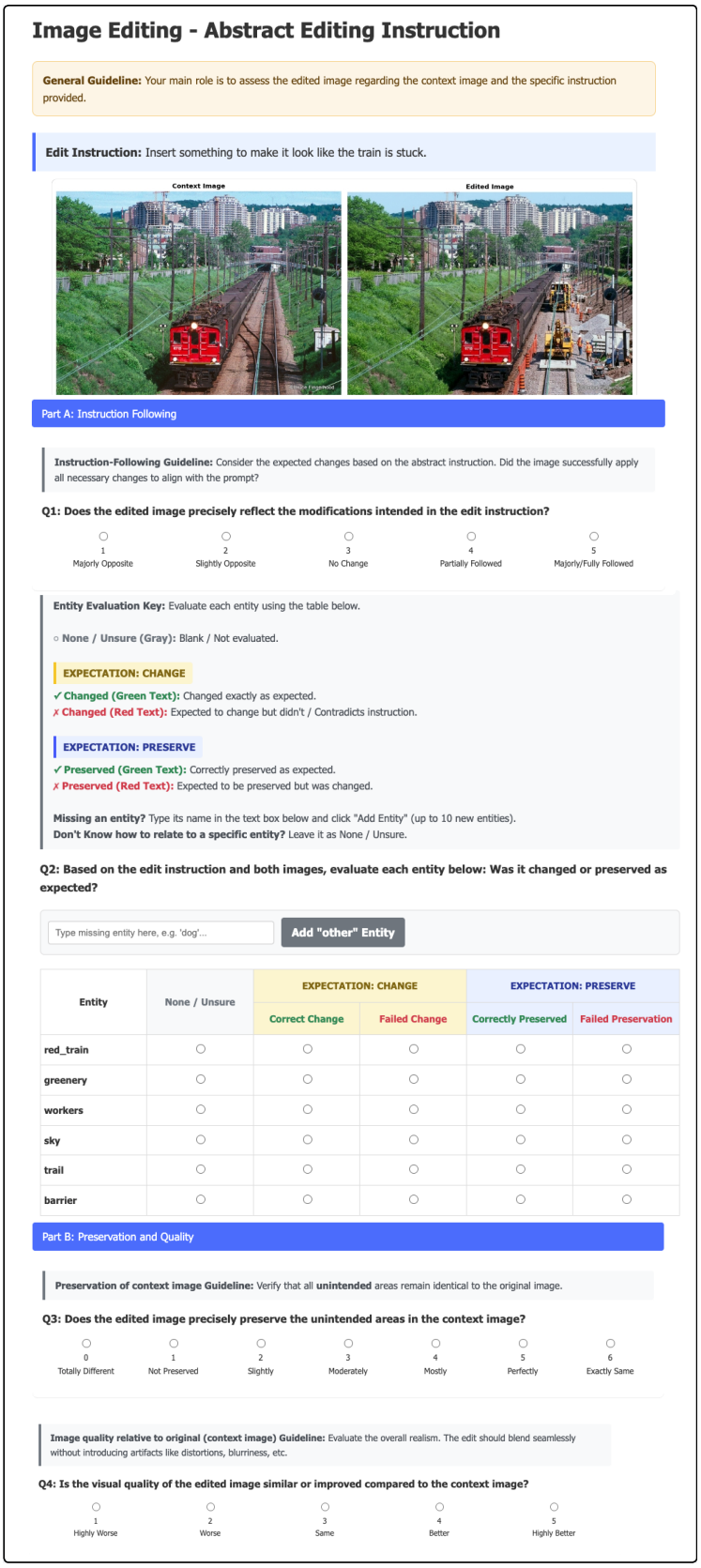


Figure 15: Human Evaluation Questionnaire in Amazon Mechanical Turk (AMT).

E EXPERIMENTAL SETUP

Image Editing Models. We primarily evaluate 11 state-of-the-art image editing models, selected to span a diverse range of parameter scales and architectural designs (see Table 4). The open-source models include FLUX.1-Kontext [dev](Labs et al., 2025), FLUX.2 [dev] (Black Forest Labs, 2025), Qwen-Image-Edit-2511 (Wu et al., 2025b;a), Step1X-Edit (Liu et al., 2025), Bagel (Deng et al., 2025), and HiDream-E1-1 (Cai et al., 2025). For Step1X and Bagel, we additionally evaluate their respective thinking variants ("think-and-reflect" and "think" modes correspondingly). The evaluated closed-source models consist of GPT-Image-1.5 (OpenAI, 2025), Seedream-4.5 (Seedream et al.) and the Gemini Family; Gemini-3 Pro Image Preview (Nano Banana), Gemini 3.1 Flash Image Preview (Google Developers, 2026) (Nano Banana 2) and Gemini-2.5 Flash Image (Comanici et al., 2025). We evaluated all models using their default inference parameters (see Table 5).

Instruction Following Evaluation Metrics. We examine three commonly adopted instruction-following metrics alongside our proposed evaluation ENTITY-RUBRICS. First, ΔCLIP computes the difference in cosine similarity between the abstract instruction and the edited image versus the context image; this distance measures the net change in image-text alignment. For the second and third methods, we utilize VLM-based approaches that prompt a VLM with the instruction, the context image, and the candidate edited image to generate a global instruction-following score and a brief explanation. Specifically, VIE-Score (Ku et al., 2024) incorporates penalties for over-editing within its system prompt, while ComplexEdit (Yang et al., 2025) utilizes discrete per-score descriptions to guide the model. For all VLM-based metrics, including ENTITY-RUBRICS, we employ Gemini 3 Flash with a thinking budget as the backbone evaluator. This ensures a high-reasoning, consistent baseline for scoring, with each prompt instructing the VLM to conclude with a single, finalized score.

Preservation Metrics. Alongside the instruction-following evaluation metrics, we measure semantic preservation of the context image, identity preservation as well as image quality based on VLM (Yang et al., 2025) and perceptual similarity to the context image (LPIPS (Zhang et al., 2018)).

E.1 COMPUTATIONAL REQUIREMENTS

Generating the dataset, establishing explicit baselines, and evaluating the models required specific computational resources and API access. Generating the prompt pairs required the Gemini API to create both abstract and explicit instructions for all relevant categories at once per context image, resulting in 1300 API calls across the 1300 context images sourced from the Open Images dataset. For model inference, we utilized a Google Cloud Platform VM equipped with a single NVIDIA A100 80GB GPU. Running inference for the eight open source models to generate the 940 test set samples per model took approximately 150 GPU hours in total. The evaluation phase required a VLM, specifically Gemini 3 Flash, to process two prompt types across 13 models and 470 samples. This evaluation pipeline required two separate API calls per sample to independently assess the initial context image and the final edited image, totaling 24,440 base API calls. This figure represents only the final evaluation pass and does not include the additional compute and API usage required for experimental prompt optimization during the development phase.

Table 4: Architecture Summary of Baseline Image Editing Models. DiT notes Diffusion-Transformer. ✓ and ✗ denote the presence or absence of a dedicated thinking mode, respectively.

Category	Model	Company	Text-Encoder	Image Generator	Size/Params	Thinking
Open-source	Flux-Kontext [dev]	Black Forest Labs	CLIP + T5	DiT	∼12B	✗
	Flux2 [dev]	Black Forest Labs	Mistral + T5	DiT	∼32B	✗
	Qwen-Image-Edit	Alibaba Cloud	Qwen-VL-2.5	DiT	20B	✗
	Step1X-Edit	Stepfun	Qwen-VL	DiT	∼30B	✓
	HiDream-E1-1	HiDream.ai	T5 + CLIPs + Llama	DiT	∼25B	✗
	Bagel	ByteDance	Bagel-MoT	Custom Unified	14B (7B active)	✓
Closed-source	Gemini 2.5 Flash	Google	X	X	-	✗
	Gemini 3 Pro	Google	X	X	-	✗
	Gemini 3.1 Flash	Google	X	X	-	✗
	GPT-Image-1.5	OpenAI	X	X	-	✗
	Seedream-4.5	ByteDance	X	X	-	✗

Table 5: **Model Inference Configurations.**

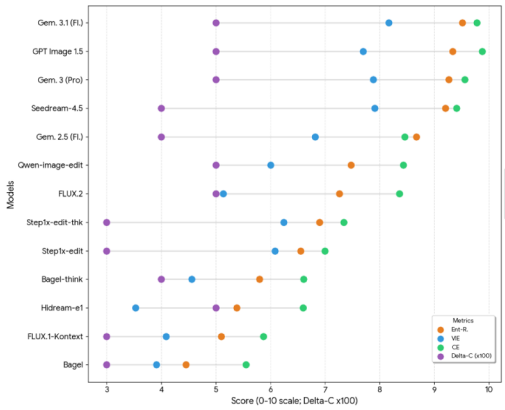
Model	Category	Inference Steps	CFG Scale	Resolution	Thinking	Seed
FLUX.2	Open	20	2.5	1024	✗	42
FLUX.1-Kontext	Open	20	2.5	1024	✗	42
Qwen-image-edit	Open	40	4.0 (Tr)	—	✗	0
Step1x-edit	Open	28	6.0 (Tr)	—	✗	42
Step1x-edit-think	Open	28	6.0 (Tr)	—	✓	42
Bagel	Open	50	4.0 (Tx) 2.0 (Im)	—	✗	42
Bagel-think	Open	50	4.0 (Tx) 2.0 (Im)	—	✓	42
HiDream-e1	Open	28	1.5 (Im)	—	✗	42
Gemini (All)	Closed	—	—	—	✗	42
GPT Image 1.5	Closed	—	—	—	✗	—
Seedream-4.5	Closed	—	—	2048	✗	—

F Complementary Results and Analysis

Discrepancies in evaluation metrics. An analysis of the global results reveals significant limitations in existing automated metrics. We examined CLIP (Radford et al., 2021), encoding-based metric, where we compute the difference between the CLIP score of the edited image with the instruction to the context image with the instruction. Moreover, we examine two commonly adopted VLM-based approaches, VIE-Score (Ku et al., 2024) and ComplexEdit (Yang et al., 2025) of assessing instruction-following. As presented in Table/Figure 16, the ΔCLIP metric exhibits minimal variance between models, proving uninformative and incapable of capturing subtle, fine-grained semantic changes. It highly favors logical inconsistencies and over-editing, as presented in Figure 17. Conversely, while the VLM-based metrics are highly correlated, ComplexEdit demonstrates a strong leniency bias, consistently yielding inflated scores that fail to strictly separate high-quality edits from mediocre ones. While the VIE Score accounts for structural preservation, it frequently over-penalizes legitimate global changes and lacks interpretable, detailed explanations for its scoring. These discrepancies highlight the necessity of our Entity-Rubrics evaluation, which provides a more balanced, diagnostically transparent evaluation.

Figure 16: Performance Comparison of Image Editing Models. **Instruction Following:** Ent-R (Entity-Rubrics), VIE (VIE Score) (Ku et al., 2024), CE (Complex-Edit) (Yang et al., 2025), ΔC (ΔCLIP) (Radford et al., 2021) Scores presented visually as well on the right.

Cat.	Model	Ent-R.	VIE	CE	ΔC
Open	Qwen-image-edit	**7.48**	6.00	8.43	0.05
	FLUX.2	7.26	5.13	8.36	0.05
	Hidream-e1	5.38	3.53	6.60	0.05
	FLUX.1-Kontext	5.10	4.09	5.87	0.03
	Step1x-edit-thk	6.9	6.24	7.34	0.03
	Step1x-edit	6.55	6.08	7.00	0.03
	Bagel-think	5.80	4.56	6.61	0.04
	Bagel	4.45	3.91	5.55	0.03
Closed	Gem. 3.1 (Fl.)	**9.52**	8.17	9.78	0.05
	GPT Image 1.5	9.34	7.70	9.88	0.05
	Gem. 3 (Pro)	9.27	7.88	9.56	0.05
	Seedream-4.5	9.21	7.91	9.41	0.04
	Gem. 2.5 (Fl.)	8.67	6.82	8.46	0.04



Context Preservation and Image Quality. Table 6 details the preservation and perceptual quality metrics across the evaluated models. To establish a human baseline, we report the human preservation score (**Hum-P**), which represents the mean rating from our user questionnaire multiplied by two to align with a standardized scale. Closed-source models, such as Gemini 3.1 Flash and Seed-Dream, demonstrate a balanced performance, maintaining strong identity preservation and high perceptual quality while executing complex abstract edits. Interestingly, the most aggressive preservation scores,

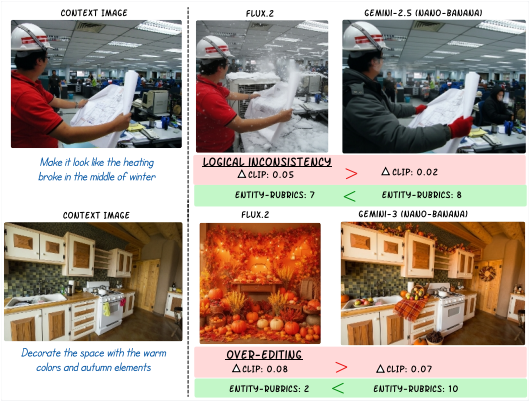


Figure 17: ΔCLIP vs ENTITY-RUBRICS Evaluation Qualitative Comparison (Samples from the ABSTRACTEDIT test set. Logical Inconsistency (top): While the prompt asked for the heating to be "broken," FLUX generated a literal heating generator and snow that didn't exist in the original office, whereas the Gemini 2.5 model correctly inferred that a broken heater means people would wear coats and gloves. CLIP favors the literal alignment. Over-Editing (bottom): When tasked with decorating a kitchen, FLUX.2 completely replaces the room with a dense pile of pumpkins and leaves. CLIP ignores the contextual anchor.

such as the exceptionally low LPIPS distances observed in models like Step1X and Bagel, require careful interpretation. Qualitative analysis reveals that these seemingly superior preservation results are frequently an artifact of severe under-editing. Because these models often struggle to resolve the semantic ambiguity of abstract instructions, they fail to modify the image meaningfully. Consequently, they leave the original context largely untouched, yielding artificially inflated preservation metrics that mask their failure to follow the underlying instruction.

Table 6: ENTITY-RUBRICS Preservation and Image Quality. **Hum-P:** Human ground truth for preservation. **LP:** LPIPS ↓, **Id:** Identity preservation ↑, **Qual:** Perceptual Quality ↑.

Cat.	Model	Hum-P ↑	LP ↓	Id ↑	Qual ↑
Open Source	Qwen-Image-Edit	-	0.50 ±0.22	5.89 ±2.87	5.11 ±1.72
	FLUX.2 [dev]	7.17	0.67 ±0.15	5.51 ±3.26	5.95 ±1.77
	HiDream-E1	-	0.63 ±0.19	4.17 ±3.01	4.85 ±1.57
	FLUX.1-Kontext [dev]	7.64	0.63 ±0.17	6.89 ±3.02	5.74 ±1.73
OS w/ Thinking	Step1X-Think-Reflect	8.65	0.35 ±0.19	8.38 ±2.24	5.77 ±1.76
	Step1X	-	0.33 ±0.19	8.52 ±2.07	5.77 ±1.70
	Bagel-Think	-	-	-	-
	Bagel	7.09	0.30 ±0.30	7.43 ±3.24	5.75 ±2.42
Closed Source	Gemini 3.1 Flash (NB 2)	7.89	0.51 ±0.24	8.17 ±2.24	6.84 ±1.92
	GPT-Image	7.07	0.64 ±0.14	7.39 ±2.39	7.00 ±1.65
	Gemini 3 Pro (NB)	-	0.45 ±0.26	8.25 ±2.43	6.63 ±1.93
	Seed-Dream	-	0.46 ±0.25	8.21 ±2.35	7.29 ±1.79
	Gemini 2.5 Flash Image	-	0.47 ±0.22	8.19 ±2.44	7.04 ±1.75

Text Clues as Abstractness Bypass in Closed-Source Models. We find that closed-source models insert entities with text/sign/cue in their names at a 52% higher rate than open-source models (2.27% vs 1.49% of inserted entities), suggesting they are more prone to increase the alignment to the abstract prompt via text-related visual elements, bypassing challenging visual edits (see examples in Appendix Figure 20).



Figure 18: A qualitative example demonstrating over-editing by HiDream-E1 when given an explicit instruction.

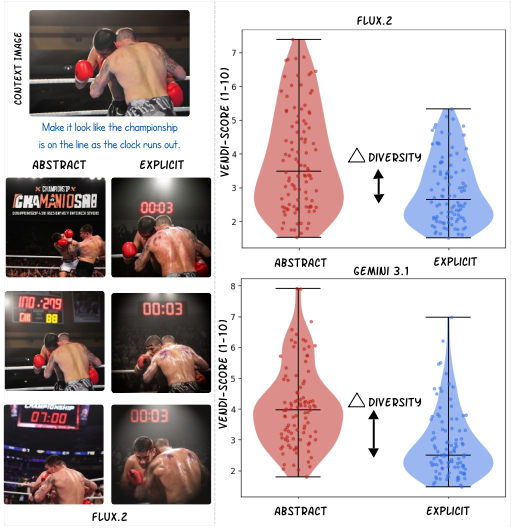


Figure 19: Abstractness Unlocks Diversity. Vendi scores (y-axis) for DINO representations across abstract vs. explicit instructions for FLUX.2 and Gemini 3.1.

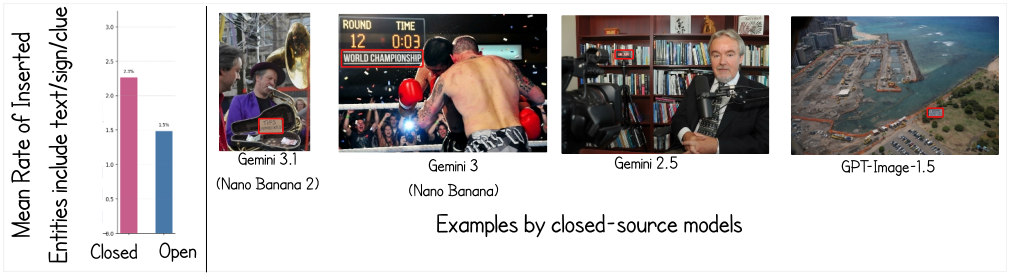


Figure 20: Qualitative Examples across Closed Source models of cases where they insert more textual clues to align better with the abstract instruction.

Complementary Results. Table 7 presents the ENTITY-RUBRICS scores on the explicit prompt, as presented in bar graphs in the main paper.

Figure 19 presents the violin distributions of Vendi scores for image representations (DINOv3 (Siméoni et al., 2025)) generated from both abstract and explicit prompt types across 10 seeds each. We compare two models, FLUX.2 and Gemini 3.1, and provide qualitative examples from FLUX.2 illustrating the limited diversity of explicit prompts relative to the richness accessed through abstract ones. Notably, this diversity gap is more pronounced in closed-source model.

Figure 21 presents a heatmap of ENTITY-RUBRICS scores decomposed by domain. The results highlight that integrating a "thinking" mode yields notable performance improvements (e.g., from Bagel to Bagel-Think, and Step1X to Step1X-Think). For Step1X, the most substantial gains are

observed in the logical and social domains, suggesting that reasoning capabilities help the model decompose complex abstract intents into aligned, explicit sub-instructions. The gains are presented qualitatively as well in Figures 23, 24. Furthermore, while the performance of closed-source models remains relatively stable overall, we observe a significant leap from Gemini 3 Pro to Gemini 3.1 Flash specifically within the physical domain.

Failure Modes by Edit Action. Figure 22 visualizes the failure rates of each model broken down by specific granular edit actions. Most notably, the heatmap exposes a universal bottleneck in spatial and quantitative reasoning. OBJECT COUNT yields the highest failure rates across virtually all models, serving as the primary point of failure even for the most advanced closed-source systems. Similarly, structural edits such as TRANSFORM and POSITION prove significantly more difficult to execute than stylistic changes. Furthermore, analyzing the thinking-enhanced variants reveals a surprising "trade-off" effect. While integrating a thinking phase successfully mitigates failures in semantic and stylistic mappings, such as Bagel-Think dropping its STYLE TRANSFER failure rate from 34 to 9, it can paradoxically destabilize fundamental operations. For instance, Step1X-Think-Reflect exhibits increased failure rates in TRANSFORM and TEXTURE compared to its base counterpart, suggesting that explicit multi-step reasoning might occasionally override robust, low-level visual priors. Finally, the data exposes a sharp dichotomy within the open-source tier: models like FLUX.2 rival closed-source systems on surface-level adjustments like TEXTURE (6) and LIGHTING (9), yet entirely collapse on structural tasks like OBJECT COUNT (41) and POSITION (33), indicating that current open-source alignment is heavily skewed toward appearance rather than spatial composition.

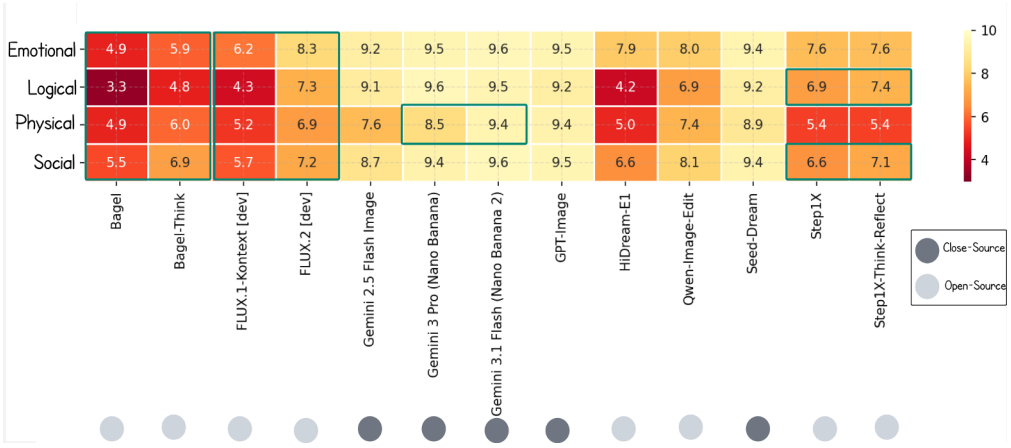


Figure 21: Heatmap Visualization of the ENTITY-RUBRICS scores per model (x-axis) and per domain (y-axis).

Table 7: Summary of model performance (ENTITY-RUBRICS) on Explicit prompts of the ABSTRACTEDIT test set. **Bold** values indicate the winning score within each specific category group.

Cat.	Model	Exp. ↑
Open Source	Qwen-Image-Edit	**8.05** ±2.51
	FLUX.2 [dev]	7.50 ±2.61
	HiDream-E1	5.43 ±2.80
	FLUX.1-Kontext [dev]	6.97 ±2.92
OS w/ Thinking	Step1X-Think-Reflect	6.70 ±3.25
	Step1X	**7.39** ±2.92
	Bagel-Think	5.82 ±3.19
	Bagel	6.35 ±2.79
Closed Source	Gemini 3.1 Flash (NB 2)	**8.99** ±1.97
	GPT-Image	8.66 ±1.98
	Gemini 3 Pro (NB)	8.80 ±2.08
	Seed-Dream	8.47 ±2.23

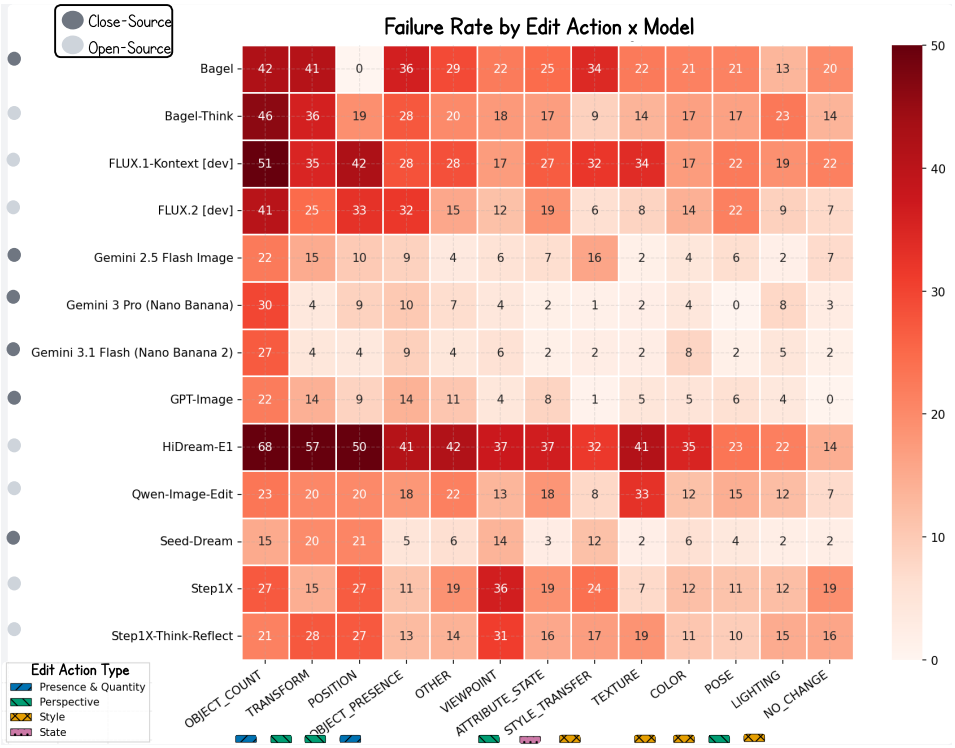


Figure 22: Heatmap illustrating the failure rate across different edit actions (x-axis) for each model (y-axis). More intense red shades correspond to a higher percentage of failure out of all entity-level edits.

G QUALITATIVE RESULTS

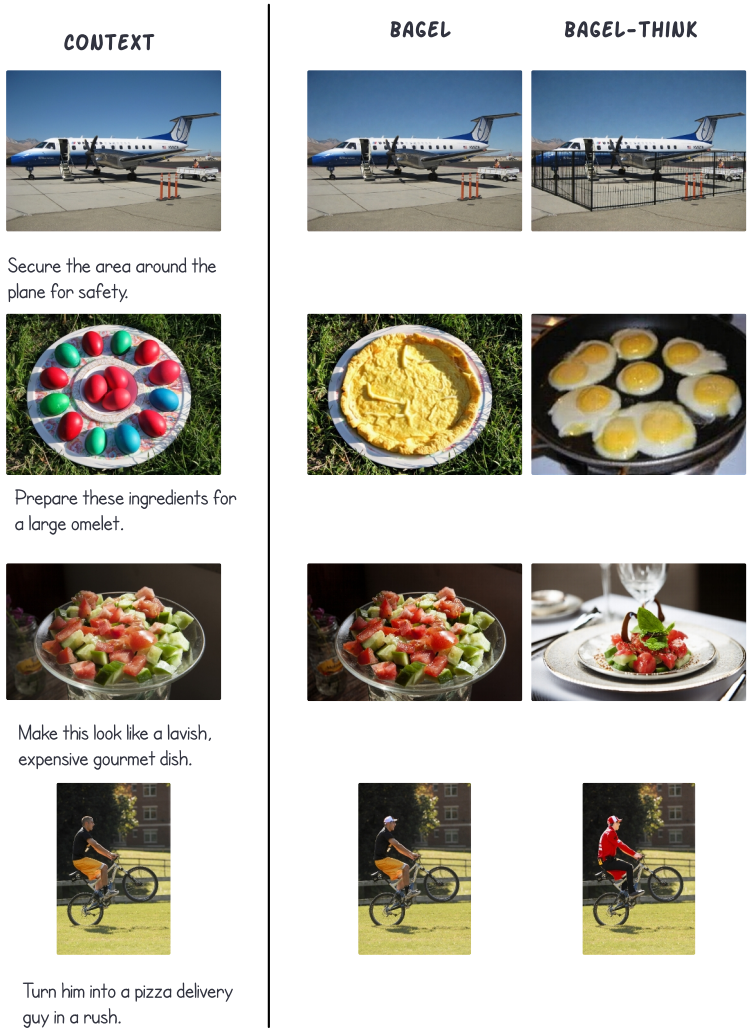


Figure 23: A qualitative comparison of Bagel vs. Bagel-Think.

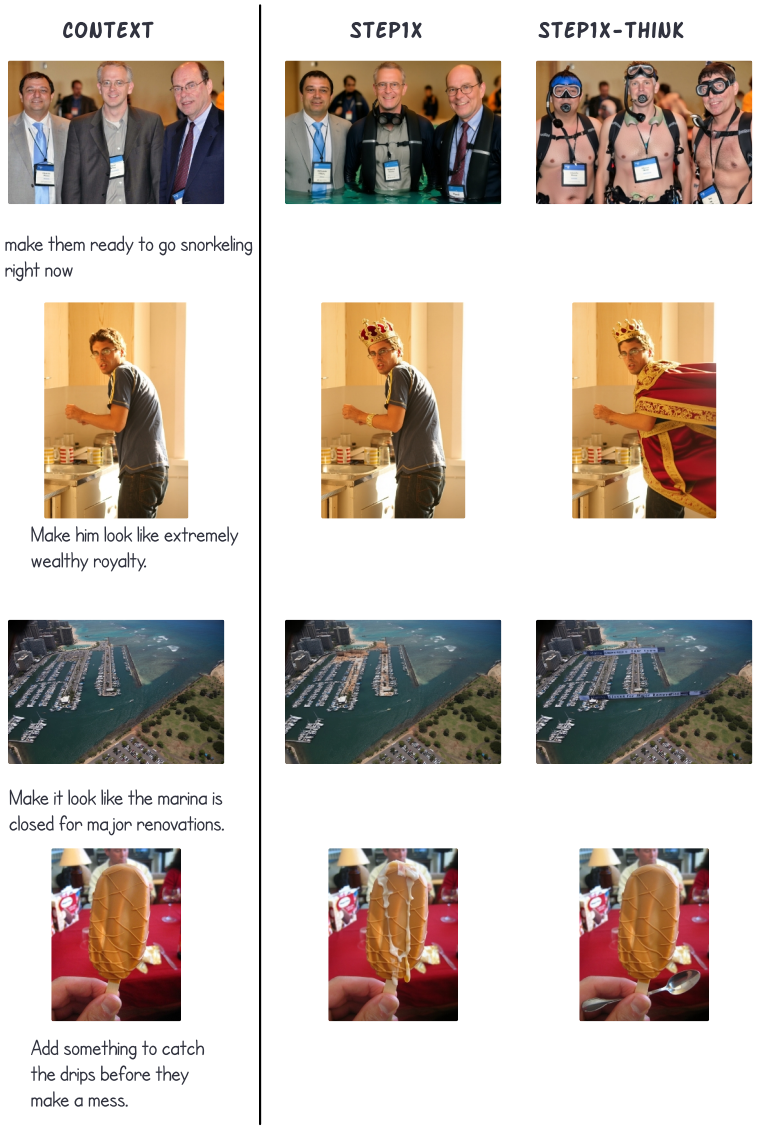


Figure 24: A qualitative comparison of Step1X vs. Step1X-Think-Reflect.



Figure 25: A qualitative comparison of all evaluated models on a sample from the logical domain. Frame colors indicate the corresponding ENTITY-RUBRICS bin scores.



Figure 26: A qualitative comparison of all evaluated models on a sample from the social domain. Frame colors indicate the corresponding ENTITY-RUBRICS bin scores.

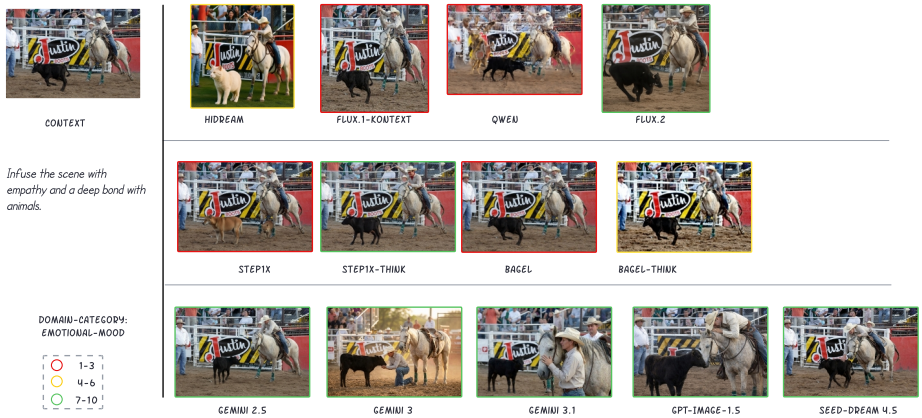


Figure 27: A qualitative comparison of all evaluated models on a sample from the emotional domain. Frame colors indicate the corresponding ENTITY-RUBRICS bin scores.



Figure 28: A qualitative comparison of all evaluated models on a sample from the physical domain. Frame colors indicate the corresponding ENTITY-RUBRICS bin scores.



Figure 29: A qualitative comparison of all evaluated models on a sample from the social domain. Frame colors indicate the corresponding ENTITY-RUBRICS bin scores.

H ENTITY-RUBRICS: FURTHER EXPERIMENTS

H.1 ABLATIONS OF THE TWO-PHASE ENTITY-RUBRICS SECOND STAGE

To analyze the impact of the two-phase design in the ENTITY-RUBRICS evaluation framework, we conduct the following ablation study. The second stage of our ENTITY-RUBRICSframework explicitly separates two phases: determining the expectation per entity, and subsequently assessing alignment for each entity against its edited version and this expectation. We compare our full pipeline against two ablation variants to isolate the contribution of these phases:

1. **Joint-Stage (Expectation + Alignment):** The model sees both images from the start and reasons about "what should happen" and "did it happen" at the same time, in hindsight, rather than committing to a prediction blind beforehand.
2. **No-Expectation (Alignment only):** We remove the entire mental framework of forming and checking an expectation. Now the answer categories themselves no longer mention expectation, necessity, or optionality. Each entity just gets `GOOD_ALIGNMENT` or `BAD_ALIGNMENT`.

We apply these analyses to two open-source baseline models: Flux.1-Kontext and Step1X-Think-Reflect, and examined the change in the evaluation scores and the human correlation ($n = 60$). The results are presented in Table 8: cells show the absolute value, with the relative percentage of that model's own full-pipeline value in parentheses.

Table 8: Ablation results on the two-stage evaluation framework. Cells show the absolute value, with the relative percentage of that model's own full-pipeline value in parentheses.

Metric	FLUX.1 Kontext	Step1X-Think-Reflect
Full Framework: Two-Stage (Ours)		
— Entity Eval Score	5.10 (100.00%)	6.90 (100.00%)
— Human-Eval Corr (r)	0.73 (100.00%)	0.63 (100.00%)
Ablation 1: Joint-Stage		
— Entity Eval Score	5.08 (99.62%)	5.94 (86.04%)
— Human-Eval Corr (r)	0.58 (80.14%)	0.42 (66.93%)
Ablation 2: No-Expectation		
— Entity Eval Score	5.20 (101.88%)	6.09 (88.17%)
— Human-Eval Corr (r)	0.57 (78.21%)	0.36 (57.19%)

Both ablations consistently degrade the framework's alignment with human judgment on both models, and the degradation compounds as more of the expectation-based stage is fully removed (Ablation 2). Neither stage can be omitted or replaced by a simpler joint or direct judgment without a measurable, monotonically compounding loss in human alignment. This indicates that forming an explicit per-entity expectation, and doing so independently, before the alignment assessment, is a necessary component of ENTITY-RUBRICS, not an incidental design choice.

H.2 EXPLAINABILITY OF THE FINAL SCORE VIA ENTITY-LEVEL FEATURES

A primary limitation of relying on a VLM for final scoring is its inherent black-box nature, which can obscure the rationale behind the overall rank. In our initial experiments, we observed that manually aggregating entity-level assessments to compute a final score could trade accuracy for explainability. Setting manual weights preserves strict entity-level explainability but restricts the VLMl's reasoning, preventing it from modeling complex dependencies between entities. To provide greater transparency and explainability without constraining the model, we conduct a linear regression analysis trained to predict the final model scores based entirely on the entity-level features (e.g., `GOOD_PRESERVATION`, `BAD_EXPECTED_CHANGE`).

For each sample, we convert these labels into 23 categorical fractions and fit a ridge-regularized linear regression. This model is evaluated per edit-model and pooled across all edit-models. The resulting standardized RidgeCV coefficients for the all-models pooled fit are presented in Table 9.

Table 9: Standardized RidgeCV coefficients (weights) for the all-models pooled fit, illustrating the impact of entity-level features on the final score.

Feature	Weight
GOOD_EXPECTED_CHANGE	$+0.21$
GOOD_EXPECTED_PRESERVATION	$+0.14$
GOOD_OPTIONAL_CHANGE	$+0.09$
BAD_EXPECTED_CHANGE	-0.20
BAD_OPTIONAL_CHANGE	-0.32
BAD_EXPECTED_PRESERVATION	-0.33
(all group / edit_action weights)	between -0.06 and $+0.07$

The regression explains a substantial share of the final score's variance: $R^2 = 0.50$ in-sample, $R^2 = 0.49$ under 5-fold cross-validation, and $R^2 = 0.45$ under leave-one-model-out validation (i.e., weights fit on 11 edit-models still explain 45% of a completely unseen 12th model's score). Interestingly, the learned weights indicate that violations are consistently penalized more heavily than correct edits are rewarded. Furthermore, unwanted changes to entities that should have been preserved or only optionally modified (BAD_EXPECTED_PRESERVATION, BAD_OPTIONAL_CHANGE) are penalized more strictly than missed expected changes (BAD_EXPECTED_CHANGE). This implies that the model's final ranking is more sensitive to over-editing than to under-editing.

These results demonstrate that, although the final scoring includes a holistic step, much of the variance is captured by the transparent, entity-level criteria inherently exposed by the ENTITY-RUBRICS framework, thereby directly improving score explainability. Furthermore, this analysis opens avenues for future work in image-editing explainability by extending our correlational approach to investigate causal mechanisms linking entity rubrics to final scores via internal model representations.

H.3 ANALYSIS OF UNDER-EDITING (MISSED EXPECTED CHANGES)

This section provides an entity-level analysis of ENTITY-RUBRICS outputs across the test set to examine its behavior in cases of missed expected changes (i.e., under-editing). We filtered for samples where the instruction required at least one specific change (meaning at least one entity was tagged as EXPECTED_CHANGE rather than EXPECTED_PRESERVATION or OPTIONAL_CHANGE) and where all other entities were correctly preserved. We then grouped these samples based on how many of the required entities were actually modified as intended (none, some, or all) and examined the corresponding scores assigned by our framework.

Table 10: Score distribution based on the completion of required edits. The data is pooled across all 12 evaluated edit models, totaling 4,347 images.

Required Edits Actually Made	# Samples	Avg. Score (1–10)	% Scoring ≥ 7
None	129	2.5	0%
Some	233	5.0	29%
All	3,985	9.3	94%

As shown in Table 10, outputs that leave the image "safely" untouched but skip the instruction's actual intent score dramatically lower: 2.5 out of 10 on average, with absolutely none reaching a score of 7 or higher.

This aligns with the linear regression analysis presented in §H.2, which predicts the final score from these same entity-level labels across all 12 evaluated edit models. That regression demonstrated that a missed required edit (BAD_EXPECTED_CHANGE) carries a clear negative weight on the final score. This check confirms that the rubric does not allow a model to artificially inflate its score by performing no action: **under-editing is penalized just as strictly as making unwanted changes elsewhere**.

H.4 PRACTICAL BENEFITS OF ENTITY-RUBRICS: TEST-TIME CRITIQUE AND REVISE

To evaluate the utility of ENTITY-RUBRICS as a test-time feedback mechanism, we implement a critique-and-revise setup using the Step1X open-source model, which natively supports iterative thinking (think + reflect). Specifically, we replace the model's original internal reasoning with the evaluation output from ENTITY-RUBRICS, which includes entity-level assessments, the final score, and the final rationale. Following the configuration of the Step1X-Think-Reflect baseline, we incorporate an early stopping mechanism with a maximum of $N = 3$ turns, requiring no additional fine-tuning of the underlying editor.

The iterative process begins with the initial edit generation (v_0). This edit, along with its ENTITY-RUBRICS feedback, serves as the input context for the subsequent generation turn. We evaluate this pipeline across the entire test set ($n = 469$), with results presented in Table 11.

Table 11: Performance of ENTITY-RUBRICS when used as a test-time critique-and-revise mechanism compared to Step1X baselines.

		Step1X Variants		
Setup	n	**No-Thinking** (v_0)	**+ ENTITY-RUBRICS (Ours)**	**Think+Reflect**
test set	470	6.57	**7.48**	6.90

The results demonstrate that employing ENTITY-RUBRICS as a critique-and-revise mechanism consistently improves performance over both the standard zero-shot baseline and the model's native iterative mechanism. The mean alignment score rises from 6.57 (No-Thinking) to 7.48 with our ENTITY-RUBRICS loop, surpassing the native Step1X-Think-Reflect approach (6.90). Notably, this substantial improvement in alignment is achieved entirely at inference time, without requiring any additional fine-tuning of the underlying editor. See qualitative comparison in Fig. 30.

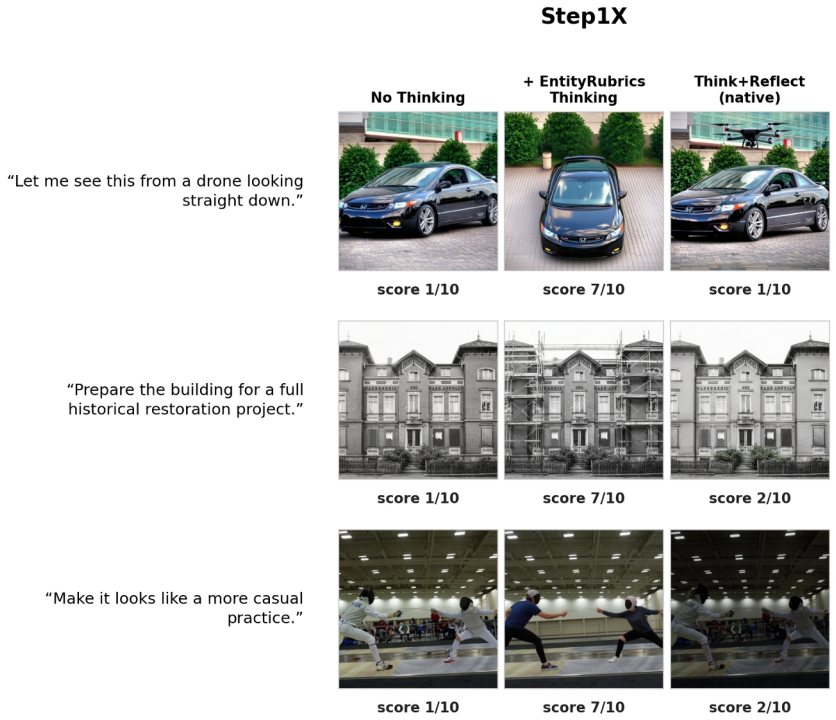


Figure 30: Qualitative comparison between ENTITY-RUBRICS-guided critique-and-revise and the native thinking mechanism in Step1X.

I ABSTRACTEDIT TRAINING SET: QUALITY AND UTILIZATION

I.1 QUALITY AUDIT OF TRAINING INSTANCES

We conduct a quality audit (Table 12) on a stratified random sample of 100 training instances (25 per domain), manually evaluated against their context and paired edited images (generated by Gemini 3.1 Flash).

Table 12: Quality audit results on a stratified random sample of 100 training instances.

Domain	N	Abstract (Yes / No)	Context-related (Yes / No)
Physical	25	**100.0%** / 0.0%	**68.0%** / 32.0%
Logical	25	**100.0%** / 0.0%	**100.0%** / 0.0%
Social	25	**92.0%** / 8.0%	**80.0%** / 20.0%
Emotional	25	**60.0%** / 40.0%	**88.0%** / 12.0%
Overall	**100**	**88.0%** / 12.0%	**84.0%** / 16.0%

Overall, 88% of abstract instructions are valid, and 84% are strongly grounded in their context image, with zero instances showing image-text mismatches or broken target pairings. Abstract instruction failure modes (e.g., the 40% "No" rate in Emotional) primarily stem from two edge cases: instructions that inadvertently incorporate explicit details (e.g., "make the scene look dramatic and sad, tears drop on the man's face") and instructions describing conditions already fully depicted in the context image (e.g., "make it look like a moment of full focus in the game"). Context-related failure modes (16% "No") consist of generic directives untethered to visual anchors unique to that image (e.g., "A quiet sorrow, draped in shadow"). This manual audit confirms that the automatically generated training set maintains high fidelity and validates the pipeline's overall robustness.

I.2 TRAINING SET UTILIZATION: FINETUNING REDUCES UNDER-EDITING FAILURE MODE

Training an image editing model to map abstract instructions to a correct edit, among many possible interpretations, is a non-trivial task, primarily due to the complex, one-to-many distribution mapping inherent to this problem. While the 4k training subset is a complementary asset released to the community rather than a core contribution of our paper, we believe it creates strong opportunities for future work and establishes a necessary first baseline to comprehensively cover the challenge of abstract image editing. While beyond the primary scope of this paper, we leverage this dataset to examine different training objectives designed to mitigate the under-editing failure modes commonly observed in open-source models such as FLUX.1.

We examine three training objectives centered on relativity and granularity: *Supervised Fine-Tuning (SFT)*, *Global Direct Preference Optimization (DPO)* and *Masked DPO*. We begin with SFT as a baseline for fundamental instruction-following, mapping abstract prompts to high-quality target images produced via the set of explicit instructions. The primary limitation of SFT is its inability to provide a directional gradient for nuanced changes; it treats the target as a static goal rather than a relative transformation. To move beyond static reconstruction, we then employ offline Diffusion DPO (Wallace et al., 2024), which introduces a comparative learning signal by training the model to differentiate between chosen (y_c) and rejected (y_r) edited image pairs. However, global DPO often lacks the precision required for abstract tasks, where the model must distinguish which specific entities to modify and which to preserve based on subtle cues. Finally, we propose Entity-Masked DPO to inject spatial granularity into the learning process, an objective that optimizes the policy π_θ specifically within entity-level masks M, as defined by Equation 1:

$$\mathcal{L}_{\text{Entity-DPO}}(\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_c,y_r,M)}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_c|x,M)}{\pi_{\text{ref}}(y_w|x,M)} - \beta \log \frac{\pi_\theta(y_r|x,M)}{\pi_{\text{ref}}(y_r|x,M)}\right)\right] \tag{1}$$

By translating these entity-specific masks M into the latent space, we can concentrate the optimization on the regions most relevant to the abstract prompt. This spatial grounding ensures that the model learns the relationship between the prompt and the visual context at an object-oriented level.

Dataset Curation for DPO. For the Direct Preference Optimization (DPO) objectives, we curated a dataset of paired "chosen" and "rejected" instances. Because comparing two independent image edits is challenging due to inconsistent attribute modifications, we developed a synthetic pipeline to ensure clear preference boundaries. We utilized the fully edited images as the chosen samples (consistent with our SFT data) and generated multiple rejected counterparts for each (see Figure 31). These rejected versions were created by systematically "dropping" the instructions related to a single entity. For instance, if a prompt required modifications to a dog, a blanket, and a cage, we generated three distinct rejected instances where the edits for one of those entities were intentionally omitted. This approach ensures that the model learns to prioritize the completion of every explicit instruction within a multi-entity prompt. For the Entity-Masked DPO objective, we utilize SAM-3 (Carion et al., 2025) to generate entity-specific masks. These masks are downsampled from pixel space to the latent space by including all latents where the entity coverage exceeds a threshold of 0.1.

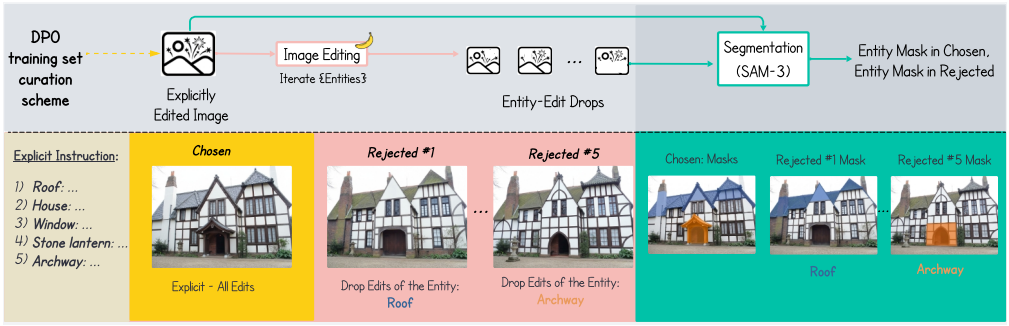


Figure 31: DPO training set curation, where the original explicit image edits serve as the "chosen" samples, while "rejected" pairs are generated by systematically omitting instructions for specific entities.

Training Ablations and Failure Mode Reduction. To validate whether training on our curated data improves editing quality and addresses specific failure modes, we evaluated three LoRA-training approaches on FLUX.1-Kontext: SFT, Global DPO, and Masked-DPO. We assessed all three methods using our automatic ENTITY-RUBRICS framework on the full 469-image benchmark, ensuring that source images in the evaluation set were strictly held out from all training and validation splits.

Table 13: Performance of different training objectives evaluated using the ENTITY-RUBRICSframework. Entity-Masked-DPO significantly reduces under-editing while mitigating the severe over-editing typical of Global DPO.

Training Setup	Under-editing	Over-editing	ENTITY-RUBRICSMean Score (1–10)
Untrained baseline (FLUX.1-Kontext-dev)	48.2%	20.0%	5.10
LoRA-SFT	73.6% (+25.4pp)	6.4%	4.01
Global DPO	44.6% (−3.6pp)	73.8%	4.36
Masked-DPO (Ours)	**24.7% (−23.5pp)**	57.6%	5.01

As shown in Table 13, standard LoRA-SFT does not reduce under-editing; rather, it *increases* it, collapsing toward conservative, near-identity edits. This occurs because mapping one abstract instruction to a fixed explicit target provides no directional signal regarding how much to change—a known behavioral-cloning failure mode.

Conversely, Global DPO's chosen/rejected preference signal successfully reduces under-editing, but without spatial grounding, it frequently over-changes incorrect or peripheral regions, driving the over-editing rate up sharply.

Our proposed Entity-Masked-DPO, which computes the preference loss strictly within the SAM-3 segmented mask of the target entity, nearly halves the untrained baseline's under-editing rate (from

48.2% to 24.7%) while substantially curbing Global DPO's over-editing (reducing it from 73.8% to 57.6%). Note that under-editing and over-editing are independent boolean assessments derived from the same evaluation pass; therefore, an image can trigger both simultaneously (e.g., required changes are absent while unrelated regions are improperly altered).

These results confirm that the curated training set carries a robust, learnable signal for resolving under-editing, and that a preference-based, entity-grounded objective extracts this signal most cleanly. Qualitative examples comparing these behaviors across the different models are provided in Figure 32 and Figure 33.

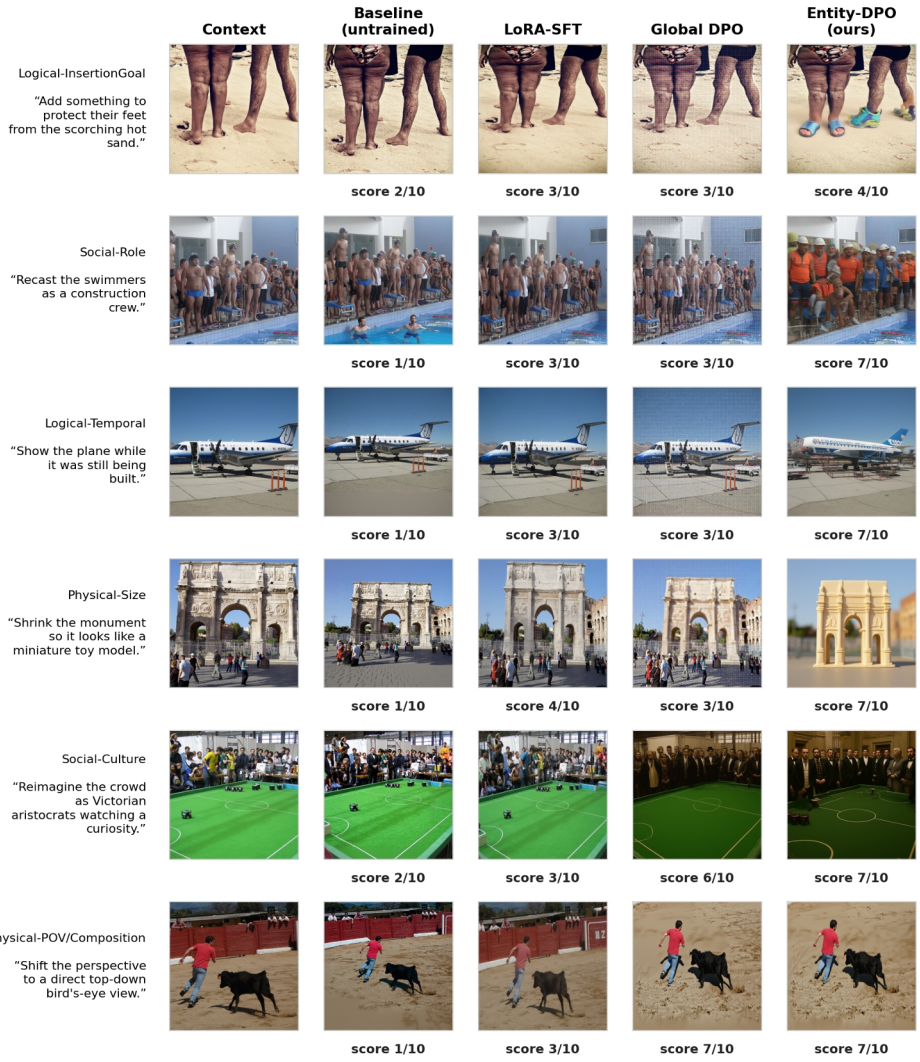


Figure 32: A qualitative comparison of FLUX.1-Kontext training objectives, demonstrating the effect of different fine-tuning methods on reducing under-editing.

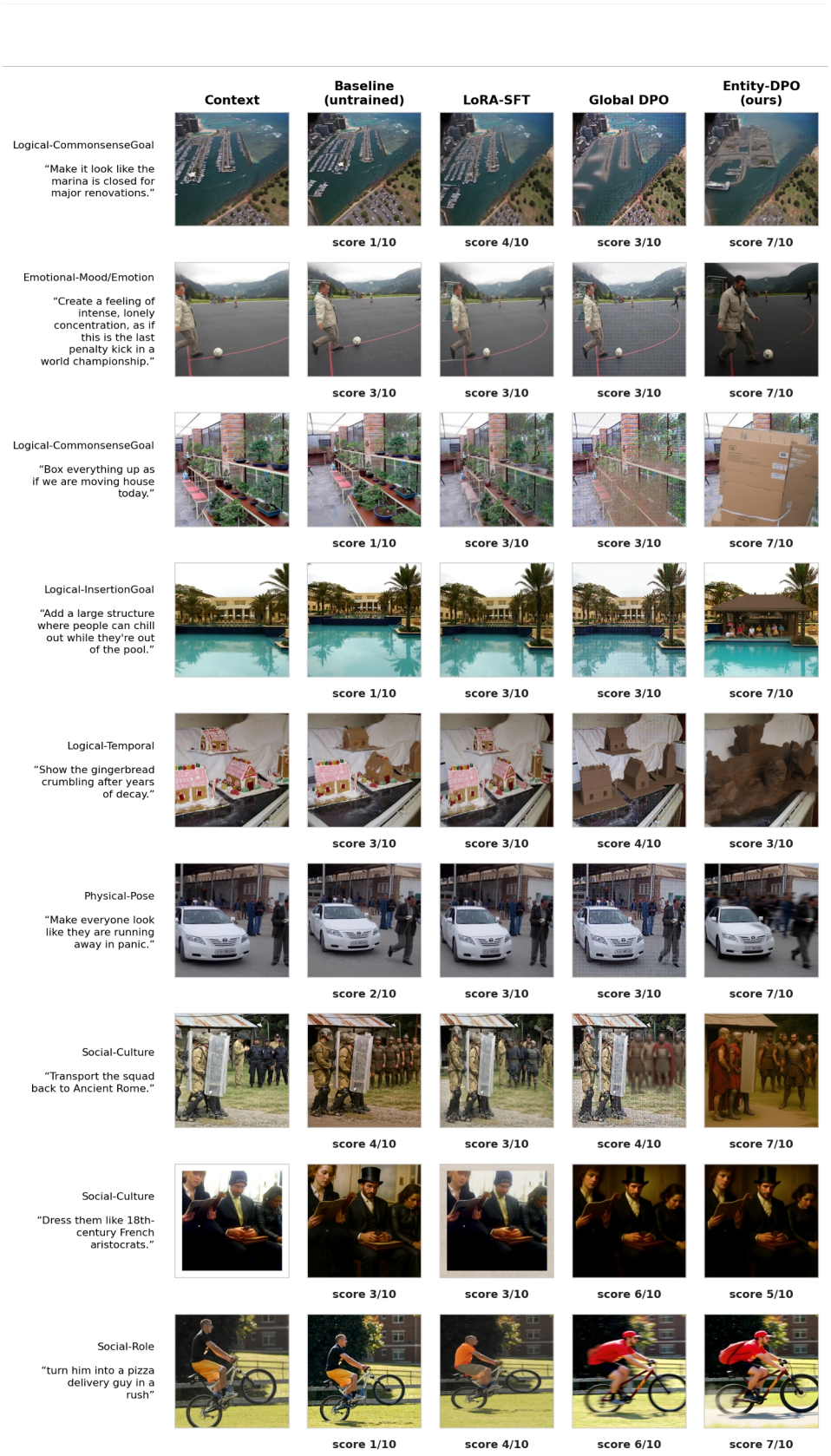


Figure 33: A qualitative comparison of FLUX.1-Kontext training objectives, demonstrating the effect of different fine-tuning methods on reducing under-editing.