NoHumansRequired: Autonomous High-Quality Image Editing Triplet Mining

Maksim Kuprashevich
mvkuprashevich@gmail.com

Grigorii Alekseenko
grigoriyalexeenko@gmail.com

Irina Tolstykh
irinakr4snova@gmail.com

Georgii Fedorov
fedorov.ssu@gmail.com

Bulat Suleimanov
motjumi@gmail.com

Vladimir Dokholyan
doholyan.vs@gmail.com

Aleksandr Gordeev
asegordeev@gmail.com

SALUTEDEV LLC

Abstract

Recent advances in generative modeling enable image editing assistants that follow natural language instructions without additional user input. Their supervised training requires millions of triplets ⟨original image, instruction, edited image⟩, yet mining pixel-accurate examples is hard. Each edit must affect only prompt-specified regions, preserve stylistic coherence, respect physical plausibility, and retain visual appeal. The lack of robust automated edit-quality metrics hinders reliable automation at scale. We present an automated, modular pipeline that mines high-fidelity triplets across domains, resolutions, instruction complexities, and styles. Built on public generative models and running without human intervention, our system uses a task-tuned Gemini validator to score instruction adherence and aesthetics directly, removing any need for segmentation or grounding models. Inversion and compositional bootstrapping enlarge the mined set by $\approx 2.6\times$*, enabling large-scale high-fidelity training data. By automating the most repetitive annotation steps, the approach allows a new scale of training without human labeling effort. To democratize research in this resource-intensive area, we release* ***NHR-Edit****, an open dataset of 720k high-quality triplets, curated at industrial scale via millions of guided generations and validator passes, and we analyze the pipeline's stage-wise survival rates, providing a framework for estimating computational effort across different model stacks. In the largest cross-dataset evaluation, it* ***surpasses all public alternatives****. We also release* ***Bagel-NHR-Edit****, a fine-tuned Bagel model with state-of-the-art metrics.*

Project page, dataset, and model: https://riko0.github.io/No-Humans-Required/

1. Introduction

Recent acceleration in generative modeling has facilitated image-editing assistants that follow natural language instructions. Creating such editors is a multi-stage process, starting with **foundational pre-training** on large, often noisy datasets (e.g., Brooks et al. [4], Ge et al. [9], Hui et al. [14], Wei et al. [28], Ye et al. [33], Yu et al. [34], Zhang et al. [36], Zhao et al. [40]). This stage adapts a base text-to-image model to execute diverse edits and preserve unedited regions. Next, **initial SFT** on smaller, curated datasets elevates performance on specific tasks; ObjectDrop [5] and OmniPaint [23] have shown that as few as 2500-3300 pairs of real photos can teach a model to remove shadows and reflections in object removal task. The third stage, **continual supervised fine-tuning (SFT) and preference optimization** [20, 27], handles more complex edits and improves quality but presents a data bottleneck. It is constrained by reliance on human annotators to review millions of pixel-level edits, which is not the best use of expert attention.

Existing large-scale data collection methods have fundamental drawbacks. Cascades of external tools, e.g., for grounding [16], segmentation [15], and inpainting [24], create visual artifacts and can corrupt the data — if an imperfect "remove" edit with inpainting artifacts is inverted into an "add" operation, the model may learn to use artifacts as spatial cues rather than understanding the instruction's semantics, effectively **poisoning** the training data. Approaches like 3D rendering [7] lack realism and scalability, while video frame extraction [17] depends on complex, error-prone auxiliary models. A lack of reliable validation metrics for detecting subtle defects persists; although MLLMs are now used as evaluators [28, 29, 33], we found even top models like Gemini 2.5 Pro [10] insufficient, and we therefore fine-tuned a Gemini-2.0-flash [11] validator on human scoring data (Sec. 3.2).

We posit that the potential of a model after initial SFT is under-exploited. By utilizing its new abilities and sensitivity to stochastic initialisation, the editor itself can generate unlimited high-quality synthetic data. To realize this, we introduce an end-to-end triplet-mining pipeline. For each instruction, the framework generates multiple candidate edits. These are pre-filtered, then judged by our fine-tuned validator, which selects the single best edit that meets our strict quality standards (Algorithm 1). This self-contained framework unlocks several capabilities for continual learning:

- **Direct complexity measurement for curricula:** Instruction difficulty for the *current* model is quantified by counting attempts for a successful edit, providing a direct signal for an **easy-to-hard learning curriculum**.
- **Targeted weakness correction:** Rare successes on complex tasks can be mined by running the model repeatedly to harvest a targeted dataset that fixes that weakness.
- **Compositional edit synthesis:** Complex training data can be created by combining multiple instructions. For example, a single instruction can execute two additions, one deletion, and a global style change in one pass.
- **Flexible input sourcing:** The framework uses real and synthetic inputs. Real images provide authentic scenarios, while synthetic images enable exploration of the long-tail, including **impossible-to-photograph scenarios** (e.g., a corgi in a spacesuit on a rocket).
- **Unparalleled simplicity and flexibility:** The framework is model-agnostic and requires no external specialist models for segmentation, depth estimation, or grounding.

To demonstrate effectiveness, we release NoHumansRequired Dataset (**NHR-Edit**), a public dataset of 720k rigorously validated triplets (for representative samples, see Figure 1 and Figures C.8-C.19 in Appendix). Building on this data, we release **Bagel-NHR-Edit**, a LoRA-tuned BAGEL [8] variant trained on NHR-Edit that surpasses the base model on two benchmarks. Our primary contribution is this end-to-end pipeline, a powerful engine for advancing research in self-improving generative models [6, 39].

2. Related Work

Our research builds upon two main pillars of generative modeling: methodologies for creating instruction-based editing data and the paradigm of model self-improvement through preference optimization.

2.1. Methodologies for Editing Data Generation

Creating high-quality editing data is a foundational challenge, with existing approaches presenting unique trade-offs.

Pipelines on Real-World Data. A common strategy is a cascade of models to edit real images, like in AnyEdit [34] and ImgEdit [33], which use pipelines for detection [16], segmentation [15], and inpainting [24]. Each stage can propagate errors, and global edits struggle to preserve details. Video-based methods like Step1X-Edit add complexity with pipelines for motion estimation and background filtering [41]. These approaches can also suffer from dataset bias (Schuhmann et al. [22]).

Fully Synthetic Generation. Synthetic generation offers more control but has its own drawbacks. Methods range from 3D rendering [7], which is labor-intensive and lacks photorealism, to diffusion-based techniques [9, 14, 40] that can introduce artifacts, alter details, or generate data misaligned with real-world distributions.

Specialist Models. OmniEdit [28] trains specialized models for each task (e.g., inpainting, attribute modification) integrated into similar pipelines. While ensuring quality for simple tasks, this inherits cascade complexity and error propagation issues and cannot handle complex, compositional instructions.

Our work differs by using the editor model itself as the data source, creating a simple framework that bypasses complex pipelines and specialist models.

2.2. The Metric Gap in Image Editing

Evaluation is a key challenge, as traditional, reference-based metrics (e.g., LPIPS [37], DINO [21], CLIP-Score [12]) correlate poorly with human preference and are unsuitable for our generative framework. While MLLM-based reward models have emerged in related fields (IQA, T2I, T2V) [30, 31, 38], their use in editing was pioneered by VIEScore [29], which showed GPT-4o judgments align well with human preferences. Subsequent work like OmniEdit and ImgEdit built on this by distilling judgments or fine-tuning MLLMs. However, curating data for SFT demands higher precision. We found that even top models like Gemini 2.5 Pro [10] are unreliable for detecting subtle editing flaws (Fig. C.7). We therefore developed a specialized validator by fine-tuning Gemini-2.0-flash [11] on human preference data to achieve the necessary sensitivity.

2.3. Self-improvement and Iterative Learning

A model generating its own data for self-refinement is a highly effective concept, proven in NLP [20, 26] and extended to generative vision [35]. Our framework is an automated engine applying these preference alignment techniques to image editing. Algorithms like DPO [27] and KTO [2] require scalable preference-labeled data, which our pipeline automatically provides. By solving

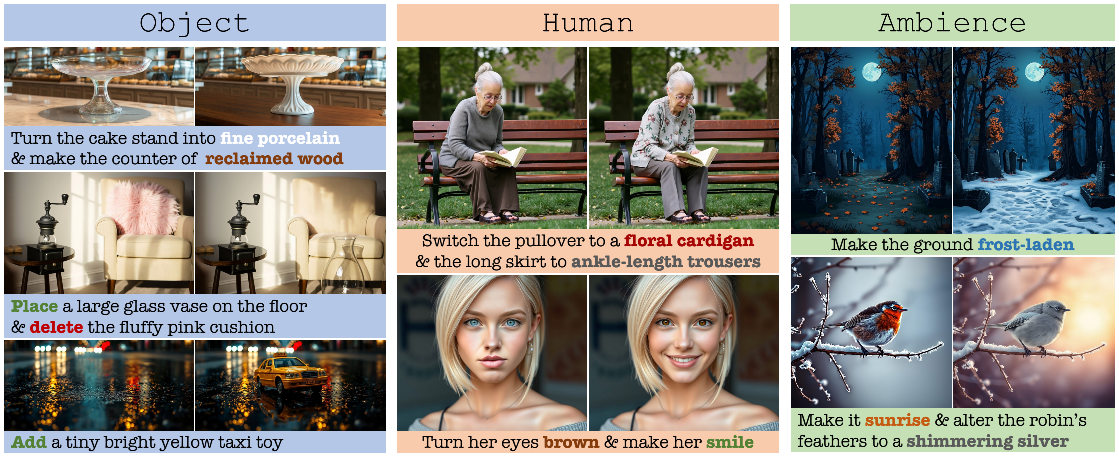
Object
Human
Ambience
Turn the cake stand into fine porcelain
& make the counter of reclaimed wood
Switch the pullover to a floral cardigan
& the long skirt to ankle-length trousers
Make the ground frost-laden
Place a large glass vase on the floor
& delete the fluffy pink cushion
Add a tiny bright yellow taxi toy
Turn her eyes brown & make her smile
Make it sunrise & alter the robin's
feathers to a shimmering silver

Figure 1. High-quality samples from our **NHR-Edit** dataset.

the data generation and labeling bottleneck, our work enables applying these powerful self-improvement techniques to instruction-based image editing.

3. Methodology

This section details our autonomous triplet-mining pipeline, which comprises four modules: (i) a prompt engineer for generating consistent text-to-image (T2I) and image-to-image (I2I) instructions; (ii) a T2I generator; (iii) an instruction-guided image editor; and (iv) a multi-stage validation stack.

3.1. Automated Mining Pipeline

Figure C.6 and Algorithm 1 overview the pipeline (full prompts can be found in Appendix A). The process starts with initial constraints (e.g., topic, style) which are used by a prompt engineering module (Algorithm 1a) to produce a T2I prompt (p_{t2i}) and corresponding edit instructions ($\{p_e\}_k$), as shown in Listing 1. While supplied manually here, these constraints could be automated.

For each T2I prompt, the pipeline generates N candidate source images (I_0) using different random seeds (Algorithm 1b). Each source image undergoes M edit attempts for every instruction p_e. This yields a large pool of candidate triplets $\langle I_0, p_e, I_e \rangle$, which are subjected to a coarse pre-filtering step before final validation (see Sec. 3.2). In the final stage, for each unique pair $\langle I_0, p_e \rangle$, the highest-quality edited image $I_e^\star$ is selected by maximizing the geometric mean of its scores ($\sqrt{s_{\text{aes}} \cdot s_{\text{adh}}}$, see Algorithm 1). We chose this metric because it enforces a balance between aesthetic quality and instruction adherence, proving particularly robust for highly imbalanced scores where a candidate excels on one criterion but fails on the other. This prevents the selection of, for instance, a visually pleasing but semantically incorrect edit. The winning image is added to the final dataset $\mathcal{D}$ only if both of its scores exceed predefined quality thresholds.

Listing 1 Example of a generated T2I prompt and its corresponding edit instructions.

```
\\ T2I prompt
"prompt": "A living room with a large
window: a small cactus on the
windowsill, a half-eaten bowl of cereal
 on the coffee table, a remote control,
 a crocheted blanket, and a dog toy on
the rug.",
\\ I2I prompts for editing
"edits": [
    "Get rid of that cactus.",
    "Remove the cereal bowl.",
    "No remote control, thanks.",
    "Lose the crocheted blanket.",
    "Eliminate the dog toy.",
    "Remove the cactus, cereal, remote,
 blanket, and toy"
]
```

3.2. Validation Framework

Robust validation is a key challenge in automated triplet mining. Our two-stage process uses a **Qwen-VL 72B** pre-filter to discard obvious failures, reducing calls to the more expensive final validator. While this open-source model cannot filter all noise, it is effective. The second stage uses a specialized **Gemini 2.0 Flash** model, fine-tuned on a curated corpus, to assign final aesthetic and instruction adherence scores.

Algorithm 1 Pipeline Pseudocode

Algorithm 1a: SamplePromptsDesign

```
Require: Task description in $\mathcal{P}_{A.1}$
Ensure: Set $\mathcal{P} = \big\{(p_{\text{t2i}}, \{p_e\}_k)\big\}_m$
$\mathcal{P} \leftarrow$ OpenAI o3$\big(\mathcal{P}_{A.1}\big)$
return $\mathcal{P}$
```

Algorithm 1c: Autonomous Triplet-Mining Pipeline

```
Require: Task description in $\mathcal{P}_{A.1}$, parameters $N, M$,
    $T_{\text{aes}}, T_{\text{adh}}$
Ensure: Final dataset $\mathcal{D}$
$\mathcal{D} \leftarrow \emptyset$, Pool $\leftarrow \emptyset$
$\mathcal{P} \leftarrow$SamplePromptsDesign$(\mathcal{P}_{A.1})$ {1a}
for all $(p_{\text{t2i}}, \{p_e\}_k) \in \mathcal{P}$ do
    Pool $\leftarrow$
    Pool$\cup$TripletMining$(p_{\text{t2i}}, \{p_e\}_k, N, M)$ {1b}
end for
for all distinct $\langle I_0, p_e \rangle$ in Pool do
    $\mathcal{S} \leftarrow \{I_e \mid \langle I_0, p_e, I_e \rangle \in \textit{Pool}\}$
    $s_{\text{aes}}(I_e),\ s_{\text{adh}}(I_e) \leftarrow$ Gemini$\big(I_0, p_e, I_e, \mathcal{P}_{A.2}\big)$ for
    every $I_e \in \mathcal{S}$
    $\mathcal{S} \leftarrow \{I_e \in \mathcal{S} \mid s_{\text{aes}} \geq T_{\text{aes}} \land s_{\text{adh}} \geq T_{\text{adh}}\}$
    if $\mathcal{S} \neq \emptyset$ then
        $I_e^\star \leftarrow \arg\max_{I_e \in \mathcal{S}} \sqrt{s_{\text{aes}}(I_e)\, s_{\text{adh}}(I_e)}$
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{\langle I_0, p_e, I_e^\star \rangle\}$
    end if
end for
$\mathcal{D} \leftarrow \mathcal{D}\cup$ApplyInversions$(\mathcal{D})$ 3.6
$\mathcal{D} \leftarrow$BCFilter$(\mathcal{D}, T_{\text{inv,aes}}, T_{\text{inv,adh}})$ 3.6
$\mathcal{D} \leftarrow \mathcal{D}\cup$ApplyBootstraps$(\mathcal{D})$ 3.6
return $\mathcal{D}$
```

Algorithm 1b: TripletMining

```
Require: T2I prompt $p_{\text{t2i}}$, edits $\{p_e\}_k$, parameters
    $N, M$, global GPU-hour budget Budget
Ensure: Candidate pool $\mathcal{C}$
$\mathcal{C} \leftarrow \emptyset$, Jobs $\leftarrow \emptyset$
for $i \leftarrow 1$ to $N$ do
    seed$_i \leftarrow$ Random$(i)$
    $I_0 \leftarrow$ FLUX.1-schnell$(p_{\text{t2i}}, \text{seed}_i)$
    if not Qwen$_{7\text{B}}\big(I_0, p_{\text{t2i}}, \mathcal{P}_{A.5}\big)$ then
        continue
    end if
    for all $p_e \in \{p_e\}_k$ do
        for $j \leftarrow 1$ to $M$ do
            Jobs $\leftarrow$ Jobs $\cup \{(I_0, p_e, \text{Random}(j))\}$
        end for
    end for
end for
while Jobs $\neq \emptyset$ and GPU_hours $<$ Budget do
    sample $(I_0, p_e, s) \sim$ Uniform(Jobs)
    Jobs $\leftarrow$ Jobs $\setminus \{(I_0, p_e, s)\}$
    $I_e \leftarrow$ I2I DiT (internal)$(I_0, p_e, s)$
    $(s_{\text{aes}}, s_{\text{adh}}) \leftarrow$ Qwen$_{72\text{B}}\big(I_0, p_e, I_e, \mathcal{P}_{A.2}\big)$
    if $s_{\text{aes}} \geq T_{\text{aes}}$ and $s_{\text{adh}} \geq T_{\text{adh}}$ then
        check$_p \leftarrow$ Qwen$_{72\text{B}}(I_0, p_e, I_e, \mathcal{P}_{A.3}, \mathcal{P}_{A.4})$
        check$_l \leftarrow$ LowLevelCheck$(I_0, I_e)$
        if check$_p$ and check$_l$ then
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{\langle I_0, p_e, I_e \rangle\}$
        end if
    end if
end while
return $\mathcal{C}$
```

Validator threshold. We set the validator thresholds using an *a priori* rule grounded in the survival curve $S(T)$ (Fig. C.5 in Appendix). The curve shows a gradual decline up to ≈ 4.3 and then enters a broad cliff over $T \in [4.4, 4.9]$ with pronounced drops at $T = 4.5$ (-62.1% of the initial pool) and $T = 4.9$ (-84.0%). To avoid operating exactly at a discontinuity while staying before the collapse regime, we choose the point that maximizes the minimum distance to these two knees. This midpoint yields $T = 4.7$. Additionally, an independent 3 raters audit of 1000 randomly sampled items further indicates that the *residual* errors, i.e., cases where the hard-filter validator makes mistakes, as any model can — are dispersed at high scores and frequently lie at ≥ 4.7; items that pass $T = 4.6$ typically receive very high scores (≥ 4.8). Consequently, raising the threshold from 4.7 to 4.8 removes almost no additional erroneous samples while shrinking the dataset. We therefore adopt the first reliable operating point before the collapse region, $T = 4.7$. We note that an exact operating point could, in principle, be obtained only through a thorough manual audit, ideally yielding *per-category* thresholds. However, such curation is labor-intensive and beyond scope. The survival-curve rule above provides a sufficient and stable choice for our application, as supported by the results in subsection **Human manual audit** and cross-dataset comparison in Sec. 3.7.

Low-level check. The absolute-difference image $D = |I_e - I_0|$ is thresholded (> 40) and analysed with `ConnectedComponents` using 4-connectivity and 32-bit labels; a triplet is discarded if the largest connected component covers $< 0.5\%$ of all pixels flagged as changed. This purely heuristic, optional filter empirically outperforms a raw image-difference threshold. Cutoff level was also found during the threshold analysis of T.

Human manual audit. In a blinded audit of $n = 300$ accepted triplets (Tab. C.4 in Appendix), residual issues were low: 5.0% T2I-inherited imperfections, 4.3% difficult removals under complex lighting or occlusion, 3.3% small residuals after deletion, and 1.6% minor inpainting near the edit area.

3.3. Gemini Validator

While many pipelines use general-purpose models like *GPT-4o* [14, 28, 29] for evaluation, they are not optimized for fine-grained *pixel-level* changes (see Fig. C.7 in Appendix). To obtain reliable estimates, we fine-tuned a `Gemini-2.0-flash` [25] model on a dedicated human-annotated corpus. This corpus was meticulously constructed to cover a wide spectrum of edit qualities, using a combination of an in-house DiT editor and proprietary models like Grok [32] and Gemini. This diverse sourcing ensures the assessor was trained on a broad distribution of potential successes and failures, preventing overfitting. Following HQ-Edit [14], OmniEdit [28] and AnyEdit [34], each image is rated on two five-point scales: (i) **Instruction** score and (ii) **Aesthetics** score. The collected set contains 2998 training and 827 validation examples; every example is judged by two to four independent raters. Inter-rater reliability, as mean pair-wise Spearman correlation, is $\rho = 0.41 \pm 0.09$ for *Aesthetics* and $\rho = 0.64 \pm 0.05$ for *Instruction*, corresponding to *moderate* and *substantial* agreement. The higher consistency on the instruction axis is expected, as semantic correctness is less subjective than aesthetics. To aggregate scores, each rating is first normalized by subtracting the annotator's bias, computed relative to the same triplets they rated. The bias b_j for each rater j is

$$b_j = \underbrace{\frac{1}{|N_j|}\sum_{i\in N_j} s_{i,j}}_{\text{Rater } j\text{'s mean score}} - \underbrace{\frac{1}{|N_j|}\sum_{i\in N_j} \bar{s}_i}_{\text{Mean score of triplets rated by } j} \tag{1}$$

where N_j is the set of triplets rated by rater j, R_i is the set of all raters for triplet i, and $\bar{s}_i = \frac{1}{|R_i|}\sum_{k\in R_i} s_{i,k}$ denotes the mean score of triplet i.

The final score S_i for a triplet is then the mean of the bias-corrected scores:

$$S_i = \frac{1}{|R_i|}\sum_{j\in R_i}\left(s_{i,j} - b_j\right) \tag{2}$$

Using this annotated validation set, we benchmarked our task-specific, fine-tuned `Gemini 2.0-flash` model against its original version, the larger `Gemini 2.5-pro` [25], and `Qwen 2.5 72B`. Table 1 compares the mean absolute error (MAE) and Spearman ρ. Vanilla checkpoints suffer from calibration error, whereas fine-tuning halves the MAE and boosts rank correlation on the instruction axis from 0.36 to 0.82, outperforming even the larger 2.5-pro model. Notably, the fine-tuned model provides high-quality scores directly, without a costly chain-of-thought step, confirming a specialized assessor is a more efficient paradigm for large-scale filtering. To further validate our assessor's robustness, we benchmarked it against the publicly available ImgEdit validator [33] on a per-category basis. Overall, our assessor nearly doubles the rank correlation (overall $\rho = 0.79$ vs. 0.41). Category-level breakdowns — including large gains on *Replace* and *Compose* are provided in Appendix Tab. B.2.

Table 1. Quality metrics of the assessor model on validation data. *I — Instruction, A — Aesthetic.*

Model	I MAE ↓	I ρ ↑	A MAE ↓	A ρ ↑
Qwen 2.5 72B	0.961	0.551	0.839	0.361
Gemini-2.5-pro	0.869	0.609	0.915	0.523
Gemini-2.0-flash	1.241	0.359	1.063	0.245
Gemini-2.0-flash (finetune)	**0.503**	**0.815**	**0.568**	**0.631**

3.4. Image Editing Backbone

Our framework requires an instruction-guided image-to-image (I2I) model that takes a source image I_0 and prompt p_e to produce an edited image $\hat{I}_e$. We use a proprietary, internal diffusion-based editor but treat it as a black box. This modular design ensures no component depends on the editor's internals, allowing it to be swapped with any other I2I model. The external validation stack reinforces this modularity.

3.5. Implementation Details

Component specification. Our pipeline is fully modular; each block can be replaced by any compatible alternative. Unless otherwise noted, we use the following defaults:

- **Prompt engineer.** We query the reasoning-centric *OpenAI o3* model [18] with the template A.1 to jointly emit a text-to-image (T2I) prompt and a set of k logically consistent edit instructions.
- **T2I generator.** Source images are synthesised with *FLUX.1-schnell* [3] at a random resolution (long side $\in [860, 2200]$ px; aspect ratio bounded by 1:6 $\leq$ AR $\leq$ 6:1) using 4 steps.
- **Plausibility gate.** We retain only sample seeds whose captions pass a plausibility check by *Qwen2.5-VL-7B* [19] using (Appendix, Prompt A.5).

- **Instruction-guided editor.** By default we employ our internal I2I DiT model with 18-28 diffusion steps.
- **Soft pre-validation filter.** Candidate edits first pass a coarse screen with *Qwen2.5-VL-72B* using (Appendix, Prompts A.2, A.3, A.4).
- **Hard validation filter.** The fine-tuned Gemini validator (Sec. 3.2) runs at temperature 0.0 with (Appendix, Prompt A.2).

All Qwen-VL calls use the HuggingFace `transformers` default configuration with temperature 10^{-6}.

Configuration. The optimal counts for T2I seeds (N) and edit retries (M) depend on prompt difficulty and represent a fundamental trade-off between dataset diversity, success rate, and computational cost. While a larger M helps with harder samples by trading compute for success probability, a larger N improves diversity. Our choice of $N = 10$ and $M = 5$ was a cost-effective balance for our specific model stack and should not be considered a universal optimum. Practitioners should tune these values based on their editor's capabilities and instruction complexity. For instance, a less capable model may require a higher M to achieve a reasonable success rate. Validation thresholds are fixed at $T_{\text{aes}} = T_{\text{adh}} = 4.7$.

Budget-aware random scheduler. This scheduler allows practitioners to cap total expenditure. It works by enumerating all potential seed-instruction pairs ($N \times k \times M$), queuing those that pass a plausibility test, and then drawing jobs uniformly without replacement until a predefined limit is exhausted. This limit, denoted as `Budget`, is a user-specified cap in GPU-hours (or API-seconds). The final compute, quality, and dataset yield are therefore dictated by this budget, not by the nominal (N, M) values. In future work, this could be extended to adaptive sampling, such as prioritizing difficult categories or continuing retries until a pre-filter success.

3.6. Data Augmentation

The dataset is further refined and expanded through post-processing and augmentation.

Semantic Inversion. Any edit can be inverted by rewriting the instruction into its logical inverse using Gemini 2.5 Flash and Prompt A.6. Crucially, access to the original T2I prompt allows preserving details for a high-quality learning signal. For the example in Listing 1, the inverse of the composite deletion is not a simple addition but a fully specified prompt: "Add a small cactus on the windowsill, a half-eaten bowl of cereal on the coffee table, a remote control, a crocheted blanket, and a dog toy on the rug."

Bootstrap Composition. Since each source image I_0 can be successfully edited into multiple distinct images (I_{e1}, I_{e2}, etc.), new triplets can be constructed. Given two successful edits, a new instruction p'_{e2} can be formulated to transform I_{e1} into I_{e2}, yielding a novel compositional triplet $\langle I_{e1}, p'_{e2}, I_{e2} \rangle$ (demonstrated in Fig. 2).

Backward Consistency filter. Semantic inversion guards against trivial forward successes when the T2I misses an object. If the inverse instruction (e.g., "add the cat on the sofa") receives a low score, we drop both the forward and inverse triplets. This optional check depends on the T2I and the validator and serves as an extra quality assurance layer.

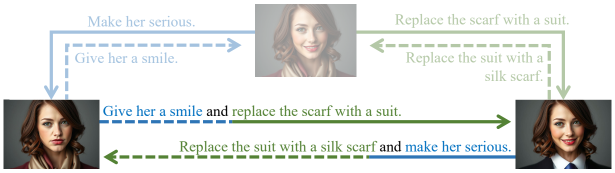
Make her serious.
Give her a smile.
Replace the scarf with a suit.
Replace the suit with a silk scarf.
Give her a smile and replace the scarf with a suit.
Replace the suit with a silk scarf and make her serious.

Figure 2. Solid arrows represent forward instructions, and dashed arrows represent their semantic inversions. Instructions for compositional triplets are aggregated from both forward instructions and inversions.

3.7. NoHumansRequired Dataset

The final pipeline yields a dataset of $720\,088$ high-quality triplets. Table 2 provides a detailed breakdown of data volume changes. Initial generation and editing phases have survival rates of 44% and 43% respectively, with subsequent filtering further refining the set. Augmentation through inversion and composition increases the dataset size by 94.88% and 30.65%.

NHR-Edit presents a variety of editing categories, while also spanning diverse styles, perspectives, and aspect ratios:

- **Removal ($\approx 227k$) and Addition ($\approx 225k$).** The focus is on object removal, as successful inversions provide challenging object addition examples, crucial for improving modern editors (Fig. C.1).
- **27 more diverse operations ($\approx 103k$).** These include complex object manipulations (reshape, change color or texture, degrade and restore), ambience (change background, time of day, weather, season), and human-related editing (emotion, haircut, clothes, accessories) — see Fig. C.2.
- **Almost 300 composite categories ($\approx 165k$).** Bootstrap composition (Sec. 3.6) allows the construction of multi-operation editing triplets, invaluable as complex training data (Fig. C.3).
- **96 various styles.** Spanning from photographic compositions (e.g., DSLR, panorama, wide-angle, aerial) — to

specific artistic choices (oil painting, sketch, anime, crochet, minimalist, etc.) (Fig. C.4).

- **26 aspect ratios.** From 640 × 1600 portraits to 1600 × 640 panoramas. Every image is a well-established composition, generated and edited in its native aspect ratio. The distribution and samples are shown in Tab. C.3.

3.8. Cross-dataset comparison.

We compare our dataset quality against public benchmarks by using our fine-tuned assessor to score 5000 random samples from each. Table 3 reports the mean *Instruction*, *Aesthetics*, and (following OmniEdit) geometric mean scores. With a geometric mean of 4.53, NHR-Edit establishes a new state-of-the-art, significantly outperforming existing datasets, including those with manual curation. This validates that our automated methodology can produce a corpus whose quality is superior to existing benchmarks.

Method note. To justify using our assessor for cross-dataset ranking, we ran a targeted human cross-check on a *sentinel* panel spanning the spectrum in Tab. 3: the lowest-ranked (UltraEdit), a mid-ranked set (HQEdit), and the two highest-ranked (OmniEdit, NHR-Edit). For each dataset we sampled $n = 80$ items and obtained 3 independent crowd annotations under the same instructions as the assessor. Table 4 reports dataset-level geometric means with 95% bootstrap intervals. Across this sentinel panel, assessor and humans induce the *same* ordering (UltraEdit $<$ HQEdit $<$ OmniEdit $<$ NHR-Edit), with substantial interval overlap in $3/4$ cases and both assigning the top rank to NHR-Edit. This probes potential misorderings at the bottom, middle, and top regimes and provides sufficient evidence that the assessor preserves dataset-level rank; we therefore use it to score 5000 samples per dataset in Tab. 3. Minor numerical differences between assessor means in Tab. 3 and Tab. 4 arise from the $n = 80$ subsampling.

4. Experiments

This section investigates if NoHumansRequired Dataset can improve an existing edit method's performance.

4.1. Experimental Setup

We use BAGEL [8], a 14B-parameter open-source multimodal foundation model with a Mixture-of-Transformer-Experts architecture. We performed parameter-efficient adaptation only to the generation expert's attention and feed-forward projection layers using LoRA [13] (rank = 16, alpha = 16, dropout = 0.05, bias = "none", batch size = 16 (it is dynamic, on average 2 per gpu), lr = 2e-5). We refer to this fine-tuned variant as BAGEL-NHR-EDIT. Other BAGEL components are frozen to preserve the model's pretrained capabilities. We chose LoRA for its training stability and substantially lower computational cost compared to full fine-tuning. All BAGEL and BAGEL-NHR-EDIT runs use matched batch size, optimizer, learning rate schedule, precision, and data augmentations.

4.2. Benchmarks and Metrics

We evaluate BAGEL-NHR-EDIT against the BAGEL baseline on GEdit-Bench [17] and ImgEdit-Bench [33], *strictly following the authors' official evaluation protocols*. For **GEdit-Bench**, we use the VIEScore setup with GPT-4o [1] to report Semantic Consistency (*SC*, 0-10), Perceptual Quality (*PQ*, 0-10), and Overall (*O*). For the **ImgEdit-Bench** evaluation, we adopt the original authors' protocol: GPT-4o is used to score edited images across several criteria, each rated on a 1-to-5 scale.

4.3. Results

Table 5 reports mean, standard deviation, and 95% confidence intervals calculated from 3 inference runs with different seeds for each model. BAGEL-NHR-EDIT improves over the baseline on the mean scores for both benchmarks: on *ImgEdit-Bench*, the overall score increases from 3.30 to $\mathbf{3.33}$ $(+0.03)$; on *GEdit-Bench*, the SC/PQ/O scores improve from $7.61/6.18/6.53$ to $\mathbf{7.80/6.56/6.80}$, with deltas of $(\Delta{+0.19}/{+0.38}/{+0.27})$ respectively. Detailed per-category results are in Appendix Tab. C.1 and Tab. C.2.

5. Conclusion

We propose an automated end-to-end pipeline to mine high-quality triplets for instruction-guided image editing. A pre-trained editor generates candidate edits and we retain only successful ones after strict filtering. Instruction inversion and compositional editing produce semantically rich, diverse triplets. Integrating a T2I model broadens stylistic coverage and mitigates overfitting. The pipeline is self-improving: as the editor advances it yields better triplets, creating a feedback loop. We release BAGEL-NHR-EDIT, a LoRA-tuned BAGEL variant that outperforms its baseline on public benchmarks, and `NHR-Edit` to support future research in text-based editing.

Limitations

Our framework is bounded by its component models: it cannot produce triplets for operations the base editor cannot perform, a limitation only partly mitigated by multi-seed sampling. Data quality also depends on the T2I generator and instruction LLM, which can introduce biases from templates or priors. LLM-written instructions may diverge from real user phrasing, though diverse prompting reduces this gap.

Reporting absolute GPU-hours would be misleading as costs depend on chosen models and API pricing. Instead, we provide stage-wise survival rates in Tab. 2 to help estimate required generations and costs for a given model stack.

Table 2. Each stage statistics for 63 292 prompts. Taking 3 072 385 generation attempts, the survival rate can be estimated as 15.3%, excluding the squeezing step.

Processing Stage	**Method / Model**	Δ **(%)**	**Remaining Vol.**
Initial Generation	FLUX.1-schnell	—	1 171 773
Generation Filtering	Qwen-7B	−56.00	515 584
Editing Generation	In-house DiT	+495.90	3 072 385
Editing Filtering	Qwen-72B (Pre-Filter)	−57.00	1 321 126
Low Level Check	Connected Component Analysis	−3.00	1 281 492
Quality Scoring	Gemini Validator (Hard Filter)	−63.21	471 523
Final Selection	ArgMax Selection	−31.01	325 287
Inversion	Gemini 2.5 Flash	+94.88	633 904
Composition	Bootstrap & Concatenation	+30.65	828 212
Backward Consistency Filtering	Gemini Validator (Hard Filter)	−13.06	720 088

Table 3. Quality metrics across editing datasets, sorted in ascending order by geometric mean. The 'Type' column indicates the generation method: **A** for Automatic and **M** for Manual. The asterisk (*) denotes a highly curated automatic dataset.

Dataset	**Type**	**Instr.** ↑	**Aesth.** ↑	**Geom.** ↑
UltraEdit	A	2.67	3.30	2.92
Seed Part 2	M	3.20	3.03	3.09
Seed Unsplash	A	3.01	3.84	3.28
InstructPix2Pix	A	3.17	3.58	3.30
MagicBrush	A	3.62	3.27	3.38
AnyEdit	A	3.39	3.64	3.44
HQ-Edit	A	2.90	4.21	3.45
ImgEdit	A	3.26	3.91	3.49
Seed OpenImages	A	3.42	3.86	3.50
Seed Part 3	M	4.06	4.37	4.13
OmniEdit	A*	4.21	4.35	4.23
NHR-Edit	A	**4.56**	**4.52**	**4.53**

Table 4. Gemini (assessor) vs. Human geometric mean (Geom.), shown as mean ± half-width of the 95% nonparametric bootstrap CI ($B = 2000$) over $n = 80$ items per dataset (3 raters/item), recomputing $Geom.$ per resample.

Dataset	**Gemini Geom.** ↑	**Human Geom.** ↑
UltraEdit	3.00 ± 0.14	3.05 ± 0.15
HQEdit	3.52 ± 0.15	3.54 ± 0.15
OmniEdit	4.30 ± 0.16	4.50 ± 0.15
NHR-Edit	**4.54** ± **0.12**	**4.75** ± **0.09**

Ethics & Societal Impact. NHR-Edit contains only *synthetic* images generated with FLUX.1-schnell from ChatGPT o3 prompts; no photographs of real people are used, so consent/privacy risks tied to real-person imagery are not implicated (though incidental resemblance is possible). We rely on provider safeguards and automated post-filters to reduce NSFW or biased samples, but filtering is imperfect and no manual curation was performed, so some undesirable cases may remain. Because editing models can be misused, the dataset is released for research use only. Prompt diversity was encouraged, yet representation biases may persist; downstream users should assess content, apply safety filters, and comply with applicable laws and policies before deployment.

Table 5. Overall results comparing our BAGEL-NHR-EDIT with the baseline. We report mean ± standard deviation and [95% confidence intervals]. The best results based on the mean are in **bold**.

Bench.	**Metric**	**BAGEL**	**BAGEL-NHR-EDIT**
ImgEdit	Overall	3.30 ± 0.03 [3.23, 3.36]	**3.33** ± **0.02** **[3.28, 3.38]**
GEdit	SC	7.61 ± 0.15 [7.23, 7.98]	**7.80** ± **0.07** **[7.63, 7.97]**
	PQ	6.18 ± 0.15 [5.82, 6.55]	**6.56** ± **0.08** **[6.37, 6.75]**
	O	6.53 ± 0.14 [6.19, 6.87]	**6.80** ± **0.07** **[6.63, 6.98]**

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 7

[2] Megha Bhardwaj and Anant Hans. Aligning text-to-image diffusion models with k-fold tamer preference. *arXiv preprint arXiv:2404.04465*, 2024. 2

[3] Black-Forest-Labs. FLUX.1-schnell. https://

huggingface.co/black-forest-labs/FLUX.1-schnell, 2024. 5

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. 1

[5] Jacopo Burroni, Federico Boin, Federico Amato Galatolo, Oussama Es-sounayni, Marco De Nadai, Federico Becattini, Nicu Sebe, Claudio Baecchi, and Alberto Del Bimbo. Objectdrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. *arXiv preprint arXiv:2403.18818*, 2024. 1

[6] Yongcen Chen, Chen Wang, Yichun Zhao, Jerry Wang, Jialu Han, Yihua Zhu, Ceyuan Zhou, Yujun He, Kewei Wu, Yong-jin Li, Tiezheng Wang, and Yu-gang Wang. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*, 2024. 2

[7] Xueting Cheng, Teli Wang, Zheyuan Liu, Wen-gang Li, Hong-gang Li, Yu-cheng Wang, and Li Wang. Aurora: A system for composing and editing images with rich styles and semantics. *arXiv preprint arXiv:2407.03471*, 2024. 1, 2

[8] 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. 2, 7

[9] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024. 1, 2

[10] Google. Gemini 2.5 Pro Preview Model Card. Model card, Google, 2024. https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro-preview.pdf. 1, 2

[11] Google. Gemini 2.0 Flash Model Card. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash, 2024. 1, 2

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning, 2022. 2

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 7

[14] Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024. 1, 2, 5

[15] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2

[16] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 1, 2

[17] 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. 1, 7

[18] OpenAI. OpenAI o3 and o4-mini System Card. https://openai.com/index/o3-o4-mini-system-card/, 2025. System Card, accessed 18 July 2025. 5

[19] Qwen Team. Qwen2.5-VL-7B-Instruct. https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct, 2024. 5

[20] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Ipo: An identity-preserving-optimization method for aligning lms. *arXiv preprint arXiv:2402.02088*, 2024. 1, 2

[21] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023. 2

[22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. 2

[23] Sang-Hyeon Shin, Jae-Ha Yang, Dong-Hyeok Han, Young-Woon Kim, and Kwang-Hyun Lee. Omnipaint: Mastering object-oriented editing via disentangled insertion-removal inpainting. *arXiv preprint arXiv:2503.08677*, 2025. 1

[24] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. *arXiv preprint arXiv:2109.07161*, 2021. 1, 2

[25] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 5

[26] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2

[27] Bram Wallace, Rafael Rafailov, Kevin Fein, Dorsa Ilas, Stefano Ermon, Christopher Ré, and Nikhil Naik. Diffusion-dpo: Aligning text-to-image models with human preferences. *arXiv preprint arXiv:2311.12908*, 2023. 1, 2

[28] Cong Wei, Zheyang Xiong, Weiming Ren, Xinrun Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. *arXiv preprint arXiv:2411.07199*, 2024. 1, 2, 5

[29] Quanzeng Wu, Jian-hao Wang, Jiachen Wang, Zexin Lin, Jiacheng Gao, Jing Zhang, and Jin Lu. VIEScore: Towards Explainable and Controllable Image-to-Text Evaluation. *arXiv preprint arXiv:2312.14867*, 2023. 1, 2, 5

[30] Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. VisualQuality-R1: Reasoning-induced image quality assessment via reinforcement learning to rank, 2025. 2

[31] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human Preference Score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, 2023. 2

[32] xAI. Grok. https://x.ai/blog/grok, 2023. Accessed: 2025-07-10. 5

[33] Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025. 1, 2, 5, 7

[34] 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*, pages 26125–26135, 2025. 1, 2, 5

[35] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation, 2024. 2

[36] 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. 1

[37] 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 (CVPR)*, 2018. 2

[38] Xuanyu Zhang, Weiqi Li, Shijie Zhao, Junlin Li, Li Zhang, and Jian Zhang. VQ-Insight: Teaching vlms for ai-generated video quality understanding via progressive visual reinforcement learning, 2025. 2

[39] Zekun Zhang, Zheyuan Huang, Yushi Li, Hong Zhou, and Hongsheng Li. Self-improving diffusion models with synthetic data. *arXiv preprint arXiv:2408.16333*, 2024. 2

[40] Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2025. 1, 2

[41] Peng Zheng, Dehong Gao, Deng-Ping Fan, Li Liu, Jorma Laaksonen, Wanli Ouyang, and Nicu Sebe. Bilateral reference for high-resolution dichotomous image segmentation, 2024. 2