
SplitFlow: Flow Decomposition for Inversion-Free Text-to-Image Editing

Sung-Hoon Yoon^{1*}, Minghan Li^{1*}, Gaspard Beaudouin², Congcong Wen^{1,3},
Muhammad Rafay Azhar¹, and Mengyu Wang^{1,4†}

¹Harvard AI and Robotics Lab, Harvard University

²École des Ponts, Institut Polytechnique de Paris ³New York University Abu Dhabi

⁴Kempner Institute for the Study of Natural and Artificial Intelligence, Harvard University

{syoon13,mli4}@meei.harvard.edu, gaspard.beaudouin@eleves.enpc.fr,
cwen2@meei.harvard.edu, rafayazhar@college.harvard.edu,
mengyu_wang@meei.harvard.edu

Abstract

Rectified flow models have become a *de facto* standard in image generation due to their stable sampling trajectories and high-fidelity outputs. Despite their strong generative capabilities, they face critical limitations in image editing tasks: inaccurate inversion processes for mapping real images back into the latent space, and gradient entanglement issues during editing often result in outputs that do not faithfully reflect the target prompt. Recent efforts have attempted to directly map source and target distributions via ODE-based approaches without inversion; however, these methods still yield suboptimal editing quality. In this work, we propose a flow decomposition-and-aggregation framework built upon an inversion-free formulation to address these limitations. Specifically, we semantically decompose the target prompt into multiple sub-prompts, compute an independent flow for each, and aggregate them to form a unified editing trajectory. While we empirically observe that decomposing the original flow enhances diversity in the target space, generating semantically aligned outputs still requires consistent guidance toward the full target prompt. To this end, we design a projection and soft-aggregation mechanism for flow, inspired by gradient conflict resolution in multi-task learning. This approach adaptively weights the sub-target velocity fields, suppressing semantic redundancy while emphasizing distinct directions, thereby preserving both diversity and consistency in the final edited output. Experimental results demonstrate that our method outperforms existing zero-shot editing approaches in terms of semantic fidelity and attribute disentanglement. The code is available at <https://github.com/Harvard-AI-and-Robotics-Lab/SplitFlow>.

1 Introduction

Flow-based generative models have demonstrated superiority in synthesizing images and also in text-to-image generation task. Building on these advances, recent research has actively explored image editing methods that modify a given image to align with a target prompt. Due to the nature of diffusion [20, 7, 23] and flow-based generative models [1, 13, 15], which generate samples from noise through iterative refinement, editing typically requires an inversion step to estimate the corresponding noisy latent representation. However, this inversion process is often imperfect and fails to recover an exact latent that reconstructs the original image. As a result, the editing process may suffer from semantic drift, visual distortion, or inconsistent attribute manipulation. Even

* Equal contribution. † Corresponding author.

when a precise inversion allows near-perfect reconstruction, an empirical trade-off arises between reconstructability and editability. Latents that are highly optimized to match the input image tend to be rigid and entangled, meaning that adjusting a single attribute often leads to unintended changes in unrelated features [2].

Recently, some efforts have been made to improve the editing process based on rectified flow models [19, 26, 3, 29, 31] and also inversion-free method that directly maps the source and target distribution [11], but these methods show limited semantic fidelity due to gradient entanglement and prompt-latent misalignment. In particular, when a complex target prompt contains multiple semantic attributes (e.g., "a german shepard dog with black sunglasses jumping on the grass with mouth opened"), a single editing trajectory guided by the full prompt often leads to entangled gradients and conflicting directions in the semantic space. This makes it difficult to isolate and control the influence of individual attributes, resulting in either under-edited or overly distorted outputs.

Motivated by these issues, we propose **SplitFlow**, which decomposes the flow induced by the target prompt into independent sub-target flows derived from semantically decomposed sub-prompts. By computing independent editing flows for each sub-prompt and aggregating them into a unified trajectory while mitigating flow conflict, our method achieves improvements in both fidelity and editability. Performance on the PIE-Bench benchmark [10] demonstrates that the proposed approach outperforms prior methods.

The contributions of this paper are threefold:

- We demonstrate that decomposing the editing flow improves the fidelity of the edited image, especially in preserving background consistency.
- We propose SplitFlow, which progressively approximates the target latent through flow decomposition followed by flow aggregation.
- We introduce a projection-and-aggregation method that aligns sub-target flows with the target direction, while preserving their semantic diversity and alleviating potential conflicts velocity field that arise during integration.

2 Related Work

Diffusion and Rectified Flow. Diffusion models [21, 18] synthesize images by gradually reversing a noising process that transforms data into Gaussian noise. Starting from random noise, they iteratively denoise through learned score functions, generating high-quality outputs over many timesteps. Rectified Flow (RF) [13, 15] introduces a more stable and efficient generation process via a straight trajectory in latent space.

Text-to-Image Inversion and Editing. Text-guided image editing aims to modify a given image to match a target text prompt. Most diffusion-based editing pipelines rely on first mapping the image back into the model’s latent space, typically in the form of Gaussian noise. DDIM [21] enables approximate inversion via a deterministic trajectory, but suffers from cumulative errors due to its linear approximation. To improve inversion accuracy, optimization-based approaches [16, 25, 8] have been proposed, though at a high computational cost. On the editing side, early methods achieve localized edits by fine-tuning diffusion models [5, 12] or manipulating cross-attention maps [6, 24]. While effective, these methods still operate within the diffusion framework and depend heavily on either inversion or fine-tuning, which are often inefficient for practical use.

Recent efforts have explored RF models for editing, which offer more stable and efficient sampling trajectories than diffusion models. RF-Inversion [19] formulates editing as an optimal control problem to balance editability and fidelity, RF-Solver [26] incorporates attention injection to guide editing, and FireFlow [3] employs higher-order ODE solvers to improve inversion accuracy. Despite their advantages, these methods still depend on iterative re-noising procedures, making them prone to cumulative error. FTEdit [29] reduces errors by performing iterative average at each inversion timestep, which is basically equivalent to increasing the number of sampling steps.

Inversion-Free Editing. Unlike traditional inversion-based approaches, which require recovering the noise latent that originally generated the image, inversion-free methods bypass this step and directly optimize in the image or latent space. InfEdit [30] introduces an early inversion-free

framework based on the consistency models [22]. FlowEdit [11] extends this idea to RF models by directly manipulating image-space velocity differences. However, it lacks directional selectivity and often struggles with global edits. These works highlight that explicitly modeling semantic flow differences enhances editing control and efficiency.

Multi-Task and Flow Decomposition. Our method builds upon insights from gradient conflict resolution in multi-task learning [32, 17, 14], where conflicting objectives are resolved via adaptive re-weighting. Similarly, we propose to semantically decompose the target prompt into multiple sub-flows and adaptively aggregate them for consistent and diverse edits. To our knowledge, our work is the first to incorporate flow decomposition and aggregation into text-based image editing framework.

3 Preliminaries

3.1 Flow Matching and Rectified Flow

Let $\mathbb{R}^d$ denote the data space, and let $x_0 \in \mathbb{R}^d$ be an initial data point sampled from a source distribution p_0 . Flow Matching (FM) methods [15, 13, 1] aim to learn a time-dependent vector field $v_\theta(x_t, t) : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, such that the solution to the following ODE transports $x_0 \sim p_0$ to $x_1 \sim p_1$, where p_1 denotes the target distribution:

$$dx_t = v_\theta(x_t, t)dt \quad (1)$$

The solution x_t describes a continuous trajectory defined by the ODE, starting from the initial point x_0 and reaching the target point x_1 . The ground-truth vector field $v^*(x_t, t)$ governs this trajectory and induces a distributional path x_t that satisfies the boundary conditions $x_{t=0} = x_0$ and $x_{t=1} = x_1$. The objective of FM is to learn a parameterized vector field $v_\theta(x_t, t)$ that approximates $v^*(x_t, t)$ by minimizing the following regression loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_t, t} [\|v_\theta(x_t, t) - v^*(x_t, t)\|_2^2], \quad (2)$$

where $\|\cdot\|_2^2$ denotes mean square error. Here, FM enables deterministic sampling via ODEs, avoiding the stochastic noise accumulation seen in diffusion models. By leveraging known couplings between distributions, it ensures both interpretability and fidelity. This leads to faster, more stable sampling and a principled connection to optimal transport, making FM a strong candidate for generative modeling. Rectified Flow (RF) [15], a special case of FM, aims to learn an ODE whose solution closely follows straight-line trajectories between pairs of points sampled from x_0 and x_1 . The ground-truth vector field is defined as $v^*(x_t, t) = x_1 - x_0$, where x_t is the linear interpolation between x_0 and x_1 . The training objective can thus be formulated as:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_t, t} \|v_\theta(x_t, t) - (x_1 - x_0)\|_2^2, \quad \text{where } x_t = (1 - t)x_0 + tx_1. \quad (3)$$

By eliminating the need for stochastic sampling and instead following deterministic linear paths, RF enables faster and more stable performance in downstream tasks such as ODE-based generative modeling and image editing.

3.2 Inversion-free Image Editing

Image editing aims to transform a source image x_0^{src} , guided by a source text prompt φ^{src} , into an edited image x_0^{tgt} using a target text prompt φ^{tgt} . In RF, the trajectory between noise and image is assumed to be linear, meaning that the noisy latent at any timestep can be obtained via linear interpolation. Leveraging this property, inversion-free image editing method [11] bypass the need for latent inversion by estimating the vector field difference between the source and target images at each timestep. This enables the construction of a transition path in the clean image space, gradually mapping the source image toward the target. Specifically, the target-aligned latent at timestep t , denoted as x_t^{FE} , can be approximated by:

$$x_t^{FE} = x_0^{src} + x_t^{tgt} - x_t^{src}. \quad (4)$$

The evolution of this trajectory follows the ODE:

$$dx_t^{FE} = v_\theta^\Delta(x_t^{tgt}, x_t^{src}) dt \approx v_\theta^\Delta(x_{t-1}^{FE} + x_t^{src} - x_0^{src}, x_t^{src}) dt, \quad (5)$$

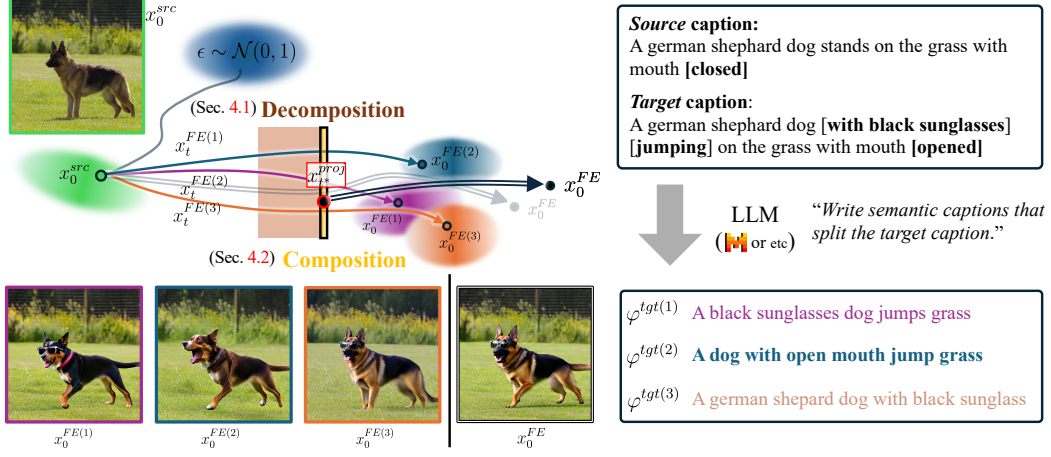


Figure 1: Simplified visual illustration of the proposed SplitFlow. Purple, Blue, Orange line indicates independent sub-target flow. The aggregation is done on a certain timestep. After the aggregation, we use a single, unified flow. The notation in this figure follows the paper.

where the difference in velocity fields guided by the source and target prompts is defined as: $v_t^\Delta(x_t^{tgt}, x_t^{src}) = v_\theta(x_t^{tgt}, t, \varphi^{tgt}) - v_\theta(x_t^{src}, t, \varphi^{src})$. Since the source image is known, x_t^{src} can be directly obtained by linearly interpolating between it and a randomly sampled noise. In contrast, the target image is unknown and cannot be directly interpolated. Therefore, x_t^{tgt} is approximated using the previous timestep x_{t-1}^{FE} and Eq. (4) as follows: $x_t^{tgt} \approx x_{t-1}^{FE} + x_t^{src} - x_0^{src}$.

Based on the estimated velocity field difference, the entire editing process can be implemented as an iterative trajectory in the clean image space. Starting from the source image, we initialize the path with $x_0^{FE} = x_0^{src}$. At each timestep t , the edited latent is updated using the approximated velocity difference: $x_{t-1}^{FE} = x_t^{FE} + \Delta_t v_\theta^\Delta(x_t^{tgt}, x_t^{src})$. This iterative procedure continues until the trajectory converges to the desired target image.

4 Method

Long target prompts often contain multiple attributes and complex semantics, resulting in a large semantic gap between the source and target in the latent space. This gap makes direct editing challenging, as entangled gradients can degrade edit quality. Providing simultaneous guidance for all attributes may lead to *conflicting flows*, which can cause semantic drift or even failure in the editing process. To address this challenge, we propose **SplitFlow**, an editing framework that progressively approximates the target latent through *flow decomposition* (Sec. 4.1) followed by *flow composition* (Sec. 4.2). Specifically, we first decompose the semantic complexity of the target prompt into a set of sub-target prompts and compute an independent flow for each, enabling latent directional components to be isolated and manipulated separately. These sub-flows are then aggregated into a unified flow that semantically aligns with the original target prompt. The editing process proceeds along this unified trajectory, resulting in more stable, diverse edits.

4.1 Progressive Target Approximation with Flow Decomposition

In image editing tasks, the semantic gap between a source prompt φ^{src} and a target prompt φ^{tgt} is often complex and high-dimensional. Direct transformation from source to target may lead to unstable or imprecise editing results. To address this, we propose to decompose the overall semantic transition into a sequence of intermediate sub-target prompts $\{\varphi^{tgt(i)}\}_{i=1}^N$, where N is the number of sub-target prompts. This decomposition simplifies the editing task into semantically controllable and progressively guided transformations.

To perform the decomposition, we leverage a Large Language Model (LLM) as a prompt reasoning engine, represented as a function f_{LLM} . We construct a composite input sequence by concatenating an instruction prompt ψ , the source prompt φ^{src} , and the target prompt φ^{tgt} . Then we feed the input

to f_{LLM} to generate a sequence of sub-target prompts that represent incremental semantic transitions:

$$\{\varphi^{tgt(i)}\}_{i=1}^N = f_{\text{LLM}}(\Phi), \quad \Phi = \text{concat}[\psi, \varphi^{src}, \varphi^{tgt}]. \quad (6)$$

Each sub-target prompt $\varphi^{tgt(i)}$ captures a localized semantic component that contributes to the overall transformation from φ^{src} to φ^{tgt} . For example, as shown in Fig. 1, given a source prompt $\varphi^{src} = \text{"A german shepherd dog stands on the grass with mouth **closed**"}$ and a target prompt $\varphi^{tgt} = \text{"A german shepherd dog with black sunglasses jumping on the grass with mouth **opened**"}$, and instruction prompt $\psi = \text{"Write semantic captions that split the target caption."}$, the resulting sub-target prompts may be: $\{\varphi^{tgt(i)}\}_{i=1}^N = \{ \text{"A black sunglasses dog jumps grass"}, \text{"A dog with open mouth jump grass"}, \text{"A german shepherd with black sunglasses"} \}$. The number of sub-target prompts N is adaptively determined by f_{LLM} based on the degree and complexity of the semantic difference. In most cases, $N \leq 3$, which yields a compact yet effective semantic trajectory for guided editing.

Following the formulation of the baseline (Eq. (4)–(5)), we extend the framework to handle each sub-target prompt individually. For each sub-target prompt $\varphi^{tgt(i)}$, we define a corresponding flow $x_t^{\text{FE}(i)}$ governed by an independent velocity field. Specifically, the sub-target flow is expressed as follows:

$$x_t^{\text{FE}(i)} = x_0^{src} + x_t^{tgt(i)} - x_t^{src}, \quad (7)$$

where x_0^{src} denotes the initial latent of the source image, x_t^{src} is the interpolated latents at timestep t between the source image and a Gaussian noise. Accordingly, the ODE governing each decomposed sub-target flow is given by the sub-target relative velocity field $v_t^\Delta(x_t^{tgt(i)}, x_t^{src})$:

$$dx_t^{\text{FE}(i)} = v_t^\Delta(x_t^{tgt(i)}, x_t^{src}) \cdot dt, \quad v_t^\Delta(x_t^{tgt(i)}, x_t^{src}) = v_\theta(x_t^{tgt(i)}, t, \varphi^{tgt(i)}) - v_\theta(x_t^{src}, t, \varphi^{src}).$$

Since $x_t^{tgt(i)}$ is not directly observable during inference, we approximate it based on the previously updated latent in Eq. (7) as: $x_t^{tgt(i)} \approx x_{t-1}^{\text{FE}(i)} + x_t^{src} - x_0^{src}$. The decomposition phase starts from η_{max} and proceeds until η_{dec} .

4.2 Flow Composition

While decomposing the flow, as described in Sec. 4.1, allows us to disentangle the gradients and achieve independent transformations, the ultimate objective in the image editing task is to perform editing aligned with the target prompt faithfully. Therefore, we devise a method to compose the previously generated sub-target flows into a unified flow that adheres to the full semantics of the target prompt. Considering the formulation of the ODE as shown in Eq. 1, the velocity field computed for each sub-target prompt can be interpreted as a gradient-like vector in latent space that guides the latent representation toward the target state. In multi-task learning (MTL), it is well known that gradients from different tasks can conflict, leading to unstable optimization or degraded performance on each task. To address this issue, various strategies such as gradient projection and orthogonality constraints have been proposed. Inspired by gradient conflict resolution methods [17, 14] in MTL, we propose a method that mitigates interference between sub-target velocity fields while effectively achieving the desired image editing objectives.

4.2.1 Latent Trajectory Projection (LTP).

To enforce global semantic consistency with the target flow while maintaining the diversity of each sub-target flow, we project each sub-target latent (conditioned on $\varphi^{tgt(i)}$) onto the target latent (conditioned on φ^{tgt}): x_t^{FE} . To perform this projection, we normalize the target latent as $\hat{x}_t^{\text{FE}} = x_t^{\text{FE}} / \|x_t^{\text{FE}}\|_2$. Here, the projection of each sub-target latent $\{x_t^{\text{FE}(i)}\}_{i=1}^N$ onto $\hat{x}_t^{\text{FE}}$ is computed as follows:

$$x_t^{\text{proj}(i)} = \left(\langle x_t^{\text{FE}(i)}, \hat{x}_t^{\text{FE}} \rangle \right) \hat{x}_t^{\text{FE}}, \quad (8)$$

where the inner product $(\langle \cdot, \cdot \rangle)$ is computed along the channel dimension of the latent. By projecting the sub-target latent onto the target latent, we ensure that the overall editing process remains consistent with the intended semantic shift. We then aggregate the projected sub-target latent to form the *projected latent*:

$$x_t^{\text{proj}} = \frac{1}{N} \sum_{i=1}^N x_t^{\text{proj}(i)}. \quad (9)$$

While both the target latent x_t^{tgt} and the projected sub-target latents $\{x_t^{\text{proj}(i)}\}_{i=1}^N$ are aligned along the same semantic direction, their origins differ fundamentally: the former stems from a unified trajectory conditioned on the full target prompt, whereas the latter are partial trajectories derived from sub-prompts and aligned through projection. This approach retains the global coherence of the target trajectory while preserving localized variations introduced by sub-target prompts, enhancing both semantic consistency and editing diversity.

4.2.2 Velocity Field Aggregation (VFA).

After projection, to further enhance flow diversity, we introduce Velocity Field Aggregation (VFA), which combines the velocity fields of sub-target flows. To quantify the directional consistency among sub-target flows, we compute the cosine similarity between their relative velocity fields with respect to the source latent. Specifically, we first compute the relative velocity vector between $x_t^{\text{proj}(i)}$ and x_t^{src} as follows:

$$\mathbf{g}_i := v_t^\Delta(x_t^{\text{proj}(i)}, x_t^{src}) = v_\theta(x_t^{\text{proj}(i)}, t, \varphi^{tgt(i)}) - v_\theta(x_t^{src}, t, \varphi^{src}) \quad (10)$$

The cosine similarity S_{ij} between the i -th and j -th sub-target prompt is defined as:

$$S_{ij} = \langle \hat{v}_t^{\Delta(i)}, \hat{v}_t^{\Delta(j)} \rangle, \quad \hat{v}_t^{\Delta(i)} = \frac{v_t^\Delta(x_t^{\text{proj}(i)}, x_t^{src})}{\|v_t^\Delta(x_t^{\text{proj}(i)}, x_t^{src})\|}. \quad (11)$$

Here, $\hat{v}_t^{\Delta(i)}$ is the normalized from $v_t^\Delta(x_t^{\text{proj}(i)}, x_t^{src})$ to compute cosine similarity. This metric captures the angular agreement between projected velocity directions, thereby reflecting the consistency of semantic changes introduced by each sub-target prompt. By applying the softmax operation to the cosine similarity map $S \in \mathbb{R}^{N \times N \times H \times W}$, we obtain a weight map $w \in \mathbb{R}^{N \times H \times W}$ that determines the relative contribution of each velocity field at every spatial location (h, w) in the latent grid:

$$\bar{v}_t^\Delta(h, w) = \sum_{i=1}^N w_i(h, w) \cdot v_t^{\Delta(i)}(h, w), \quad w_i(h, w) = \frac{\exp\left(\sum_{j \neq i} S_{ij}(h, w)\right)}{\sum_k \exp\left(\sum_{j \neq k} S_{kj}(h, w)\right)}. \quad (12)$$

Combining the projected latent from Eq. (9) with the aggregated velocity field $\bar{v}_t^\Delta$, the latent after aggregation can be updated as follows:

$$x_t^{\text{FE}} \leftarrow x_t^{\text{proj}} + \bar{v}_t^\Delta \cdot dt. \quad (13)$$

Since the proposed VFA adaptively weights the sub-target velocity fields based on their semantic alignment, it not only suppresses the influence of redundant flows but also emphasizes those with distinct semantic directions. This enables the model to preserve editing diversity while maintaining coherent alignment with the target prompt. Also, note that the LTP and VFA are applied at the end of the decomposition phase (η_{dec}).

Mathematical Justification of VFA. Here, we mathematically verify why VFA improves both fidelity and editability over mere averaging by showing:

$$\langle \bar{\mathbf{g}}, \mathbf{g}_{\text{avg}} \rangle \geq \|\mathbf{g}_{\text{avg}}\|^2, \quad (14)$$

where $\bar{\mathbf{g}} = \sum_{i=1}^K w_i \mathbf{g}_i$ and $\mathbf{g}_{\text{avg}} = \frac{1}{K} \sum_{i=1}^K \mathbf{g}_i$. Here, S_{kj} is denoted as $\langle \hat{\mathbf{g}}_k, \hat{\mathbf{g}}_j \rangle$ and $a_k = \sum_j S_{kj}$. We first reformulate Eq. (14) in terms of the scores a_i , where the LHS is $\langle \bar{\mathbf{g}}, \mathbf{g}_{\text{avg}} \rangle = \frac{1}{K} \sum_{i=1}^K w_i a_i$ and the RHS is $\|\mathbf{g}_{\text{avg}}\|^2 = \frac{1}{K^2} \sum_{i=1}^K a_i$. Thus, the proposition is equivalent to proving $\sum_i w_i a_i \geq \frac{1}{K} \sum_i a_i$. The proof combines two standard results. First, from Gibbs' inequality, the KL-divergence between the softmax distribution $w = \{w_i\}$ and the uniform distribution $u = \{1/K\}$ is non-negative, which implies: $\sum_{i=1}^K w_i a_i \geq \log(Z/K)$ (14-1), where $Z = \sum_i e^{a_i}$. Second, applying Jensen's inequality to the convex function $f(x) = e^x$ gives $\log(E[e^a]) \geq E[a]$, which in our context is:

$$\log(Z/K) = \log\left(\frac{1}{K} \sum_{i=1}^K e^{a_i}\right) \geq \frac{1}{K} \sum_{i=1}^K a_i \quad (14-2)$$

Table 1: Quantitative comparison results on PIE benchmark. For each model group (diffusion-based and flow-based), the best and second-best values are indicated in bold and underlined, respectively. Ours[†] is the result with a fidelity-enhanced setting.

Method	Model	Structure		Background Preservation				CLIP Similarity	
		Editing	Distance $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	Whole $\uparrow$	Edited $\uparrow$
DDIM [21]	Diffusion	P2P	69.4	17.87	208.80	219.88	71.14	25.01	22.44
DDIM [21]	Diffusion	PnP	28.22	22.28	113.46	83.64	79.05	<u>25.41</u>	<u>22.55</u>
Null-text [16]	Diffusion	P2P	13.44	<u>27.03</u>	<u>60.67</u>	<u>35.86</u>	<u>84.11</u>	24.75	21.86
PnP-Inv [10]	Diffusion	P2P	11.65	27.22	54.55	32.86	84.76	25.02	22.10
PnP-Inv [10]	Diffusion	PnP	24.29	22.46	106.06	80.45	79.68	25.41	22.62
RF-Inversion [19]	Flux	-	40.6	20.82	184.8	129.1	71.92	25.20	22.11
RF-Solver [26]	Flux	RF-Solver	31.1	22.90	135.81	80.11	81.90	26.00	22.88
FireFlow [3]	Flux	RF-Solver	28.3	23.28	120.82	70.39	82.82	25.98	22.94
Flow Edit [11]	Flux	-	27.7	21.91	111.70	94.0	83.39	25.61	22.70
FTEdit [29]	SD3.5	AdaLN	18.17	26.62	80.55	40.24	91.50	25.74	22.27
iRFDS [31]	SD3	-	62.72	19.61	186.39	179.76	74.59	24.54	21.67
FlowEdit [11]	SD3	-	27.24	22.13	105.46	87.34	83.48	<u>26.83</u>	<u>23.67</u>
FlowEdit [11]	SD3.5	-	<u>11.80</u>	<u>26.97</u>	<u>53.68</u>	<u>31.23</u>	89.70	26.18	22.88
SplitFlow(Ours)	SD3	-	25.96	22.45	102.14	81.99	83.91	26.96	23.83
SplitFlow(Ours) [†]	SD3	-	14.55	25.22	68.53	44.96	87.54	26.23	23.01
SplitFlow(Ours)	SD3.5	-	11.68	27.12	52.93	30.61	<u>89.76</u>	26.29	22.89

Chaining inequalities Eq. (14-1) and Eq. (14-2) directly yields the required result:

$$\sum_{i=1}^K w_i a_i \geq \log(Z/K) \geq \frac{1}{K} \sum_{i=1}^K a_i$$

This proof formalises why VFA improves both fidelity and editability: It suppresses conflicts, steering the update toward high-consensus attributes. Empirically, this manifests as higher background preservation and better CLIP similarity in Table 1-2.

When the decomposition phase is ended, the aggregated latent now follows the ODE formulation of the target prompt as described in Sec. 3.2. The decomposed flow stages facilitate fine-grained attribute manipulation without gradient entanglement, while the final unified flow phase ensures alignment with the holistic editing intent. This hybrid strategy improves editing stability and fidelity by integrating diversity-aware adjustments with prompt-level consistency. A detailed algorithmic description of SplitFlow is included in the supplementary material.

5 Experiments

5.1 Experimental Setup

Implementation Details. To show the effectiveness of the proposed method and for a fair comparison with the prior works, we employed the commonly used Stable Diffusion (SD3, SD3.5) [4] rectified flow model. By following the protocol of the baseline [11], we use the same $T = 50$ steps with $\eta_{max} = 33$, which skips the first one-third steps. The CFG values for the source and target are set to 3.5 and 13.5, respectively. The decomposition η_{dec} is set to 28, which lasts for **5 steps** ($\eta_{max} - \eta_{dec}$). To decompose the target prompt, we used Mistral-7B [9].

Dataset. We evaluate our method on Prompt-based Image Editing (PIE) Benchmark [10], which contains 700 images of natural and artificial scenes. In PIE benchmark, ten categories span from random editing written by volunteers to change image style. Each image includes a source prompt, target prompt, editing directive, edit subjects, and editing mask.

Baselines and Evaluation Metrics. As this work focuses on image editing based on rectified flow, we compare with the State-of-The-Art (SoTA) editing methods; RF-Inversion [19], RF-Solver [26], FireFlow [3], iRFDS [31], FTEdit [29], and FlowEdit [11]. Diffusion-based models such as DDIM [21, 16, 10] are also included for comparison. We evaluate our method on the PIE-Bench dataset using standard metrics. For assessing reconstruction quality and background preservation, we report image-level metrics: LPIPS [33], SSIM [27], MSE, PSNR, and Structure Distance [10]. To measure semantic alignment with the target prompt, we use CLIP similarity [28].

Table 2: Ablation study on PIE-benchmark. Here *AVG* denotes the case where mere average is applied for latent trajectory aggregation.

Baseline	LTP	VFA	Structure	Background Preservation					CLIP Similarity	
			Distance $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	Whole $\uparrow$	Edited $\uparrow$	
✓			27.24	22.13	105.46	87.34	83.48	26.83	23.67	
AVG			22.28	23.36	92.00	68.26	85.00	26.81	23.67	
✓	✓		26.22	22.37	103.58	83.67	83.76	26.93	23.82	
✓	✓	✓	25.96	22.45	102.14	81.99	83.91	26.96	23.83	

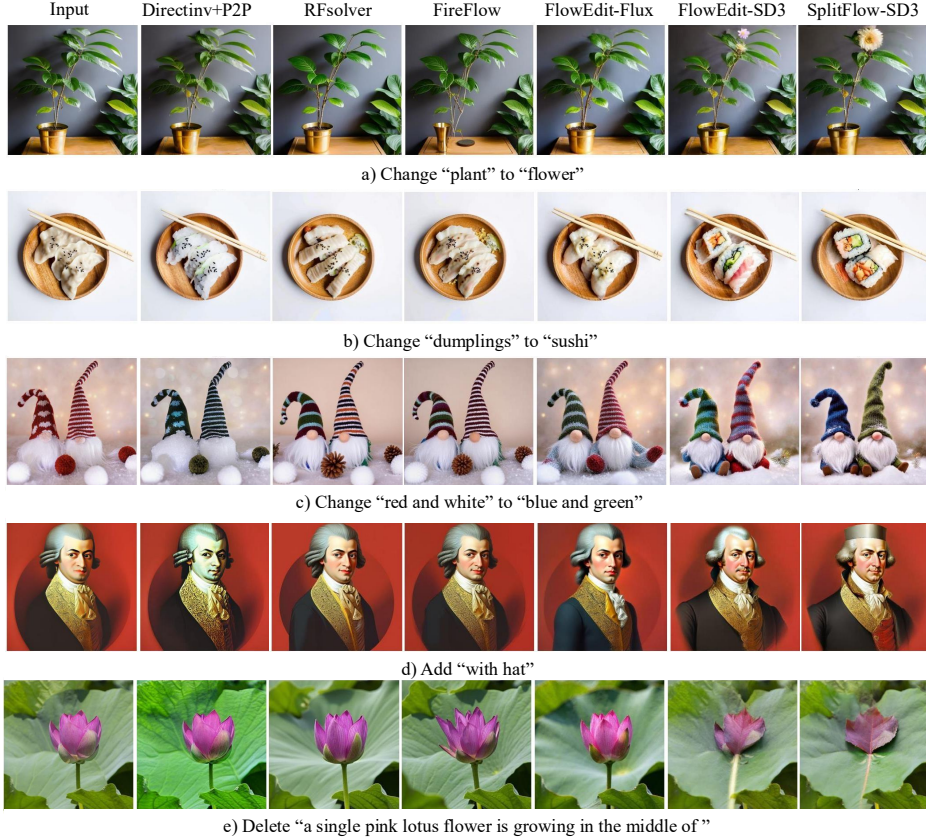


Figure 2: Qualitative comparison of prompt-based image editing methods. Each row corresponds to a specific editing instruction, where the source prompt is modified into a target prompt. From top to bottom, the tasks are: (a) *change* “plant” to “flower”, (b) *change* “dumpling” to “sushi”, (c) *change* “red and white” to “blue and green”, (d) *add* “with hat”, (e) *delete* “a single pink lotus flower is growing in the middle of”. The columns show the input image and the results generated by different models, including Directinv+P2P, RFsolver, FireFlow, FlowEdit-Flux, FlowEdit-SD3, and SplitFlow.

5.2 Main Results

Comparison with State-of-the-art Methods. To demonstrate the effectiveness of the proposed SplitFlow, we conducted experiments as shown in Table 1. Compared to the FlowEdit [11] and iRFDS [31], within the same SD3 model, the proposed SplitFlow not only outperforms in preserving background but also in editing quality. To further demonstrate the effect of the trade-off between fidelity and editability, we omit the noise interpolation during the decomposition phase and directly use the source latent x_0^{src} to better preserve structural details of the input image in the fidelity-enhanced setting[†]. Compared to the state-of-the-art method FTedit [29], our SplitFlow[†] (a fidelity-enhanced variant) achieves superior performance in both Structure Distance and LPIPS, despite FTedit employing a stronger backbone model. Moreover, it significantly outperforms FTedit in both CLIP Similarity metrics, demonstrating better alignment with the target prompt. Also, compared

Table 3: Ablation study of aggregation step η_{dec} on PIE-benchmark.

η_{dec}	Structure	Background Preservation				CLIP Similarity	
	Distance $\times 10^3 \downarrow$	PSNR $\uparrow$	LPIPS $\times 10^3 \downarrow$	MSE $\times 10^4 \downarrow$	SSIM $\times 10^2 \uparrow$	Whole $\uparrow$	Edited $\uparrow$
Baseline	27.24	22.13	105.46	87.34	83.48	26.83	23.67
30	25.99	22.41	102.33	83.06	83.88	26.92	23.79
29	26.13	22.41	102.26	82.66	83.85	26.90	23.71
28	25.96	22.45	102.14	81.99	83.91	26.96	23.83
27	25.96	22.44	102.39	82.15	83.91	26.92	23.85
26	25.94	22.44	102.70	82.27	83.91	26.95	23.84

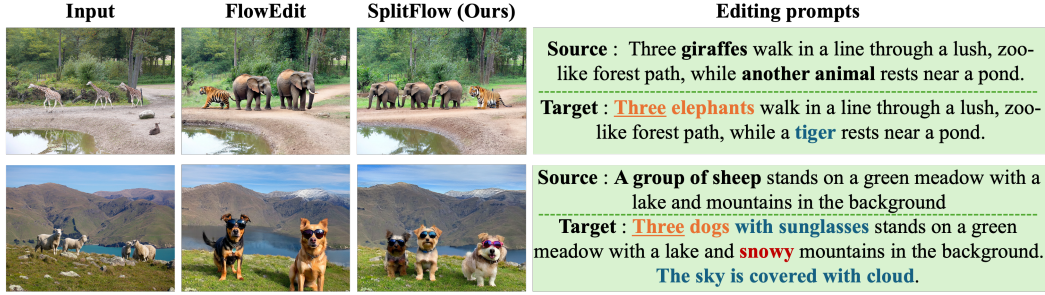


Figure 3: Qualitative comparison results with more complex prompts.

to the methods based on Flux such as RF-Inversion [19], RF-Solver [26], FireFlow [3], SplitFlow[†] outperforms the prior works. Within the SD3.5 model, SplitFlow demonstrates superior background preservation capability compared to prior works.

Qualitative Comparison. The qualitative comparison results are presented in Fig. 2. Across various scenarios—including *change*, *add*, and *delete* object prompts—our proposed *SplitFlow* demonstrates superior editability while effectively preserving the background. For instance, as shown in Fig. 2-(d), which involves adding “with hat” to a portrait of Mozart, other methods fail to generate the hat or distort the original image, whereas our method successfully synthesizes the hat while maintaining the integrity of the source image. Although it is well known that enhancing editability often compromises fidelity, our approach achieves a favorable balance by disentangling gradients within the flow during the editing process with decomposition. In Fig. 3, to further demonstrate the effectiveness of our method, we provide qualitative comparison results with more complex scenario. In the first row, given the source prompt “Three giraffes walk in a line through a lush, zoo-like forest path, while another animal rests near a pond,” the editing prompt requires changing the three giraffes to elephants and the other animal to a tiger. While FlowEdit fails to convert “three giraffes” to “three elephants,” our method successfully performs the transformation and better preserves the semantic detail of “in a line.” Additional qualitative results under complex scenarios are provided in the supplementary material.

5.3 Detailed Analysis

Component Analysis. To validate the effect of the designed component, we conducted an ablation study on PIE-benchmark as shown in Table 2. The result of simply averaging the individual flows after decomposition is reported in the second column of Table 2, denoted as *AVG*. Interestingly, even naive averaging of sub-flows maintains CLIP similarity on par with the baseline [11], while significantly improving background preservation, as reflected in metrics such as PSNR, LPIPS, MSE, and SSIM. We attribute this to the semantic decoupling effect of sub-flows, which localize edits to specific attributes and reduce unintended changes in irrelevant regions. While semantic decomposition and flow separation are key contributions of our work, our overarching objective extends beyond fidelity enhancement. Our goal is to strike a balance between fidelity and editability, ensuring that complex, multi-attribute prompts are both faithfully represented and accurately reflected in the edited outputs. To this end, our proposed components—Latent Trajectory Projection (LTP) and Velocity Field Aggregation (VFA)—go beyond averaging by explicitly aligning sub-flows with the global editing direction and adaptively weighting their contributions. By applying LTP to align each

sub-flow with the target flow, we observe meaningful improvements in CLIP similarity, particularly in the *Edited* metric, which is computed over the foreground mask. Although LTP results in lower background preservation compared to simple averaging, it still outperforms the baseline in all metrics. Furthermore, when VFA is applied, each sub-flow contributes more effectively to the final trajectory. This not only enhances background preservation—similar to the averaging strategy—but also improves CLIP similarity by promoting global semantic alignment while preserving diversity across sub-prompts.

Ablation Study on Aggregation Timestep. In Table 3, we conduct an ablation study on the aggregation timestep η_{dec} to evaluate the effectiveness and robustness of SplitFlow. Across all tested configurations, SplitFlow consistently outperforms the baseline. As η_{dec} decreases—corresponding to a longer decomposition period—we observe improved editability at the expense of background preservation. Considering this trade-off, we set $\eta_{dec} = 28$ for our final configuration. Compared to the baseline, the total number of steps required by SplitFlow can be calculated as $N \times (\eta_{max} - \eta_{dec}) + \eta_{max}$. Since $N \leq 3$ in most cases, the inference steps can be approximated as $3 \times (33 - 28) + 33 = 48$. In practice, FlowEdit requires 57 minutes for inference, whereas SplitFlow takes 83 minutes to process 700 images on the PIE Benchmark. Additionally, prompt decomposition using an LLM takes approximately 20 minutes. Additional detailed ablation studies on LLM, cost analysis are provided in the supplementary material.

Limitations A potential limitation of our method lies in its dependence on the decomposition of the target prompt. Since the editing flows are derived from sub-prompts, the quality and characteristics of the final output can vary depending on the choice of LLMs. Although SplitFlow incurs a higher inference time than the baseline, we emphasize that the LLM serves only as a proxy to facilitate our main contribution—demonstrating that decomposing the editing process into sub-target flows can significantly improve image-editing performance—and proposing a principled method to aggregate these flows effectively. By showing that flow decomposition substantially enhances both fidelity and editability in image editing, this work also opens up new directions for future research. These include developing more effective prompt decomposition techniques using LLMs or vision-language models, as well as exploring optimization-based approaches to mitigate gradient conflicts during flow composition. Additional discussions, including analyses of extreme cases, are provided in the supplementary materials.

6 Conclusion

In this paper, we proposed SplitFlow, a flow decomposition and composition framework designed to address gradient entanglement and semantic conflict that arise in image editing with complex and multi-attribute target prompts. SplitFlow computes independent editing flows for each sub-target prompt and forms a unified trajectory through projection and aggregation, thereby maintaining semantic alignment while mitigating interference between attributes. To this end, we introduced two aggregation strategies: Latent Trajectory Projection (LTP), which aligns the directional components of the latent trajectory to ensure coherence with the global target semantics, and Velocity Field Aggregation (VFA), which adaptively integrates sub-target flows while preserving their semantic diversity. These components enable SplitFlow to effectively balance fidelity and editability—two often conflicting objectives in image editing. Extensive experiments on the PIE-Bench benchmark demonstrate that our method consistently outperforms existing approaches in both visual quality and prompt alignment. Our results confirm that decomposing the editing process into semantically meaningful flows and carefully reassembling them provides a promising direction for accurate, and high-quality text-guided image editing.

References

- [1] Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants, March 2023. arXiv:2209.15571 [cs].
- [2] Yusuf Dalva, Kavana Venkatesh, and Pinar Yanardag. Fluxspace: Disentangled semantic editing in rectified flow models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13083–13092, 2025.

- [3] Yingying Deng, Xiangyu He, Changwang Mei, Peisong Wang, and Fan Tang. FireFlow: Fast Inversion of Rectified Flow for Image Semantic Editing, December 2024. arXiv:2412.07517 [cs].
- [4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.
- [5] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.
- [6] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control, August 2022. arXiv:2208.01626 [cs].
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [8] Vincent Tao Hu, David W. Zhang, Pascal Mettes, Meng Tang, Deli Zhao, and Cees G. M. Snoek. Latent Space Editing in Transformer-Based Flow Matching, December 2023. arXiv:2312.10825 [cs].
- [9] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023.
- [10] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct Inversion: Boosting Diffusion-based Editing with 3 Lines of Code, October 2023. arXiv:2310.01506 [cs].
- [11] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-Free Text-Based Editing Using Pre-Trained Flow Models, December 2024. arXiv:2412.08629 [cs].
- [12] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1931–1941, 2023.
- [13] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.
- [14] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.
- [15] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.
- [16] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text Inversion for Editing Real Images using Guided Diffusion Models, November 2022. arXiv:2211.09794 [cs].
- [17] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.
- [18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj  rn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

- [19] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic Image Inversion and Editing using Rectified Stochastic Differential Equations, October 2024. arXiv:2410.10792 [cs].
- [20] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.
- [21] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.
- [22] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.
- [23] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [24] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.
- [25] Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22532–22541, 2023.
- [26] Jiangshan Wang, Junfu Pu, Zhongang Qi, Jiayi Guo, Yue Ma, Nisha Huang, Yuxin Chen, Xiu Li, and Ying Shan. Taming Rectified Flow for Inversion and Editing.
- [27] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, page 600–612, Apr 2004.
- [28] Chengdong Wu, Ling-Qiao Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. Apr 2021.
- [29] Pengcheng Xu, Boyuan Jiang, Xiaobin Hu, Donghao Luo, Qingdong He, Jiangning Zhang, Chengjie Wang, Yunsheng Wu, Charles Ling, and Boyu Wang. Unveil Inversion and Invariance in Flow Transformer for Versatile Image Editing, March 2025. arXiv:2411.15843 [cs].
- [30] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-Free Image Editing with Natural Language, December 2023. arXiv:2312.04965 [cs].
- [31] Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-Image Rectified Flow as Plug-and-Play Priors, February 2025. arXiv:2406.03293 [cs].
- [32] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.
- [33] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: We not only propose a new method (SplitFlow), but also quantitatively show that the proposed method is effective.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: In the last section (Sec. 6), we discuss the limitation of our work.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: We provide short theoretical proof in the main paper and more details in the supplementary results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: We will release the code upon acceptance for reproducibility. Also, we provide details about our method including algorithm table (in suppl) and implementation details (in Sec. 5).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to code (github: <https://github.com/Harvard-AI-and-Robotics-Lab/SplitFlow>), with scripts to run the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Sec. 5, we specify the dataset we used, and how the baselines are selected. Also, the hyperparameter setting is included.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments were conducted with a fixed random seed, and statistical significance testing is also provided in the supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Additional details on the computational cost, including memory usage and GPU specifications, are provided in the supplementary materials.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The work follows the NeurIPS Code of Ethics; no personal data, hateful, or illicit content is used.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: This work introduces a method for image editing using prompt decomposition and flow-based generation. A potential positive societal impact is the enhanced accessibility and creative control it offers to artists, designers, and accessibility tools through more intuitive image manipulation. However, like other generative models, the proposed method

could be misused to produce manipulated or misleading media content. While the method is intended for research and creative applications, its potential for misuse warrants careful consideration. We will incorporate safeguards such as content provenance mechanisms and watermarking strategies in future deployments.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: No pretrained model is released in this work. The model we use (Stable Diffusion 3) is subject to a restrictive license that governs its usage and distribution. As such, appropriate safeguards are already in place through the licensing terms of the underlying model.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All external assets used in this paper, including pretrained models (Stable diffusion 3), datasets (PIE-benchmark), and code repositories (github), are properly cited with the original source and license information.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new dataset or model checkpoint is released.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The research uses only publicly available images and no human-subject study.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: No human-subject data collection was performed.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Large Language Models (LLMs) were used as an integral part of the methodology, specifically to decompose editing prompts into semantically meaningful intermediate descriptions that guide the diffusion-based image editing process. The use of LLMs (e.g., Mistral-7B-Instruct) impacts the behavior and outcome of the proposed method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.