FLUX.1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space

Black Forest Labs

Figure 1: Consistent character synthesis with *FLUX.1 Kontext*. Generated images can be iteratively used as context for new generations, enabling applications such as story-telling and ... **A:** show text-to-image starting image, add prompts for image in the grid. should be more clear what is edited how

Abstract

We present *FLUX.1 Kontext*, a generative flow-matching model for unified image generation and editing that performs in-context learning by extracting semantic concepts from text and input images to generate accurate views without task-specific training. *FLUX.1 Kontext* uses a simple sequence concatenation approach to handle both local editing and generative in-context tasks in a single architecture.

Current editing models exhibit degradation in character consistency and stability across multiple turns. In contrast, *FLUX.1 Kontext* improves the preservation of objects and characters, enabling robust iterative workflows. The model achieves

Preprint. Under review.

state-of-the-art performance with interactive inference speed, matching proprietary systems while delivering faster generation suitable for interactive applications.

Further, we present KontextBench, a comprehensive benchmark with 1,026 image-prompt pairs covering five task categories: local editing, global editing, character reference, style reference and text editing. Detailed evaluations show the superior performance of *FLUX.1 Kontext* in terms of both single-turn quality and multi-turn consistency, setting new standards for unified image processing models.

1 Introduction

Images are a foundation of modern communication and form the basis for areas as diverse as social media, e-commerce, scientific visualization, entertainment, and memes. As the volume and speed of visual content increases, so does the demand for intuitive but faithful and accurate image editing. Professional and casual users expect tools that preserve fine detail, maintain semantic coherence, and respond to increasingly natural language commands. The advent of large-scale generative models has changed this landscape, enabling purely text-driven image synthesis and modifications that were previously impractical or impossible [15, 34, 35, 33, 2, 10, 40, 19].

Traditional image processing pipelines work by directly manipulating pixel values or by applying geometric and photometric transformations under explicit user control [12, 45]. In contrast, generative processing uses deep learning models and their learned representations to synthesize content that seamlessly fits into the new scene. Two complementary capabilities are central to this paradigm (make Figure X for viz)

- **Local editing.** Local, limited modifications that keep the surrounding context intact (e.g. changing the color of a car while preserving the background or replacing the background while keeping the subject in the foreground). Generative inpainting systems such as LaMa [44], LatentDiffusion inpainting [35], RePaint [28], and the Stable Diffusion Inpainting variants[1][2] make such context-aware edits instantaneous; see also Palette [38] and Paint-by-Example [47]. Beyond inpainting, ControlNet [49] enables mask-guided background replacement, while DragGAN [31] offers interactive point-based geometric manipulation.
- **Generative editing.** Extraction of a visual concept (e.g. a particular figure or logo), followed by its faithful reproduction in new environments, potentially synthesized under a new viewpoint or rendering in a new visual context. Similarly to *in-context learning* in large language models, where the network learns a task from the examples provided in the prompt without any parameter updates [6], the generator adapts its output to the conditioning context on the fly. This property enables personalization of generative image and video models without the need for finetuning [36] or LoRA training [17, 24, 18]. Early works on such training-free subject-driven image synthesis include *IP-Adapter* [48] or retrieval-augmented diffusion variants [7, 3].

Recent Advances. InstructPix2Pix [5] and subsequent work [4] demonstrated the promise of synthetic instruction-response pairs for fine-tuning a diffusion model for image editing, while learning-free methods such as Textual Inversion[11] and its variants [] enable image modification with off-the-shelf, high-performance image generation models [35, 33]. Subsequent instruction-driven editors such as Emu Edit [42], OmniGen [46], HiDream-E1[13] and IceEdit [50] - extend these ideas to refined datasets and model architectures maybe be a tad more specific, depending on related work. Huang et al. [18] introduce in-context LoRAs for diffusion transformers on specific tasks, where each task needs to train dedicated LoRA weights. Novel proprietary systems embedded in multimodal LLMs (e.g., GPT-Image[30] and Gemini Native Image Gen [21]) further blur the line between dialog and editing. Generative platforms such as Midjourney[29] and RunwayML[37] integrate these advances into end-to-end creative workflows.

Shortcomings of recent approaches. In terms of results, current approaches struggle with three major shortcomins: (i) instruction-based methods trained on synthetic pairs inherit the shortcomings

[1] https://huggingface.co/runwayml/stable-diffusion-inpainting
[2] https://huggingface.co/stabilityai/stable-diffusion-2-inpainting

(a) Input image

(b) *"remove the thing from her face"*

(c) *"she is now taking a selfie in the streets of Freiburg, it's a lovely day out."*

(d) *"it's now snowing, everything is covered in snow."*

Figure 2: **Iterative, instruction-driven editing.** Starting from a reference photo (a), our model successively applies three natural-language edits—first removing an occlusion (b), then relocating the subject to Freiburg (c), and finally transforming the scene into snowy weather (d). Identity, pose, clothing, and overall photographic style remain consistent throughout the sequence.

of their generation pipelines, limiting the variety and realism of achievable edits; (ii) maintaining the accurate appearance of characters and objects across multiple edits remains an open problem, hindering story-telling and brand-sensitive applications; (iii) in addition to lower quality compared to denoising-based approaches, autoregressive editing models integrated into large multimodal systems often come with long runtimes that are incompatible with interactive use.

Our Solution. We introduce *FLUX.1 Kontext*, a flow-based generative image processing model that matches or exceeds the quality of state-of-the-art black-box systems while overcoming the above limitations. *FLUX.1 Kontext* is a simple flow matching model trained using only a velocity prediction target on a concatenated sequence of context and instruction tokens.

In particular, *FLUX.1 Kontext* offers:

- **Unified capability:** A single model covers both classic local editing and generative, in-context image generation.
- **Character consistency:** *FLUX.1 Kontext* excels at identity preservation, including multiple, iterative edit turns.
- **Interactive speed:** *FLUX.1 Kontext* is *fast*. Summary of the main latency differences to the competition here
- **Iterative application:** Fast inference and robust consistency allow users to refine an image through multiple successive edits with minimal visual drift (see Figure X).

This report is structured as follows. Section 3 describes the architecture and training strategy of *FLUX.1 Kontext*. Comprehensive quantitative and qualitative analyzes, as well as additional applications of *FLUX.1 Kontext* are presented in Section 4. Finally, **??** closes with an outlook for future work.

Table 1: Reconstruction quality comparison across different VAE architectures. All metrics computed on 4,096 image pairs. Values are mean ± standard error (rounded).

Model	PDist ↓	SSIM ↑	PSNR ↑
Flux-VAE	**0.332** $\pm$ 0.003	**0.896** $\pm$ 0.004	**31.1** $\pm$ 0.08
SD3-TAE[3]	0.746 $\pm$ 0.004	0.774 $\pm$ 0.014	27.9 $\pm$ 0.06
SDXL-VAE [33]	0.890 $\pm$ 0.005	0.748 $\pm$ 0.006	25.9 $\pm$ 0.07
SD-VAE[4]	0.949 $\pm$ 0.005	0.720 $\pm$ 0.004	25.0 $\pm$ 0.07

2 FLUX.1

FLUX.1 is a rectified flow transformer [10, 26, 27] trained in the latent space of an image autoencoder [35]. The following section summarizes the main architectural design choices that simplify and improve the performance and stability of the model training over previous diffusion and flow transformer models []. We follow Rombach et al. [35] and train a convolutional autoencoder with an adversarial objective from scratch. By scaling up the training compute and using 16 latent channels, we improve the reconstruction capabilities compared to related models; see Table 1. Furthermore, FLUX.1 is built from a mix of multimodal [10] and fused DiT [32] blocks. Multimodal blocks employ separate weights for image and text tokens, and *mixing* is done by applying the attention operation over the concatenation of tokens. After passing the sequences through the multimodality blocks, we discard the text tokens and apply 38 fused DiT blocks to the image tokens - see Figure 3 for a visualization.

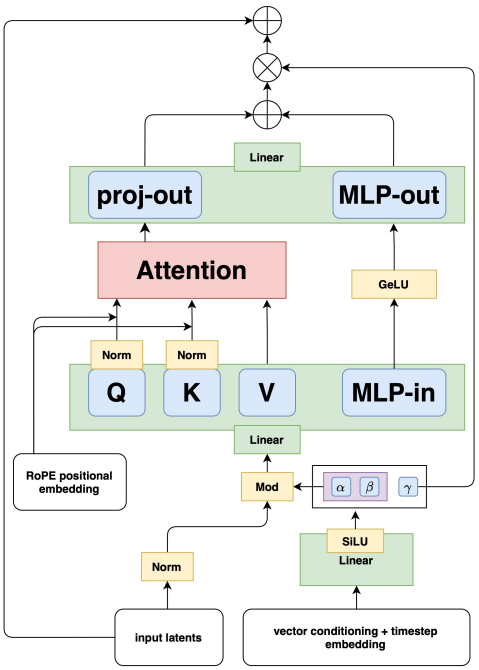

Figure 3: A fused DiT block equipped with rotary positional embeddings

To improve GPU utilization, we leverage *fused* feedforward blocks inspired by Dehghani et al. [8], which i) reduce the number of modulation parameters in a feedforward block by a factor of 2 and ii) fuse the attention input- and output linear layers with that of the MLP, leading to larger matrix-vector multiplications and thus more efficient training and inference. We utilize factorised two–dimensional Rotary Positional Embeddings (2D RoPE) [43]. Every latent token is indexed by its spatial coordinates (h, w) and an optional axial index t; the query/key vectors are then rotated by axis-specific frequencies, preserving dot-product magnitudes while enabling resolution-agnostic modeling.

3 *FLUX.1 Kontext*

Our goal is to learn a model that can generate images conditioned jointly on a text prompt and one or more reference images. More formally, we aim to approximate the conditional distribution

$$p_\theta(x \mid y, c) \tag{1}$$

where x is the target image, y is a context image (or $\varnothing$), and c is a natural-language instruction. Unlike classic text-to-image generation, this objective entails learning *relations between images themselves*—mediated by c—so that the same network can i) perform image-driven edits when $y \neq \varnothing$, and ii) create novel content from scratch when $y = \varnothing$.

To that end, let $x \in \mathcal{X}$ be an output (target) image, $y \in \mathcal{X} \cup \{\varnothing\}$ an optional *context* image, and $c \in \mathcal{C}$ a text prompt. We model the conditional distribution $p_\theta(x \mid y, c)$ such that the same network handles

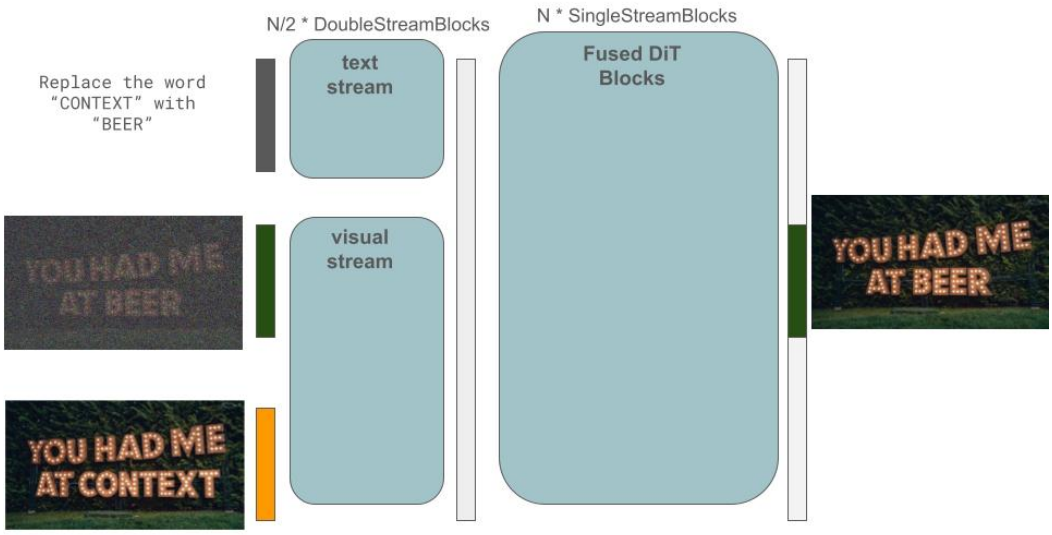

Figure 4: PLACEHOLDER. High-level overview of *FLUX.1 Kontext*, with input and context image on the left. Details in Section 3. make nice. add rope.

in-context and local edits when $y \neq \varnothing$ and free *text-to-image generation* when $y = \varnothing$. Training starts from a FLUX.1 text-to-image checkpoint, and we collect and curate millions of relational pairs $(x \,|\, y, c)$ for optimization.

Token sequence construction. Images are encoded into latent tokens by the frozen FLUX auto-encoder. All context tokens are *appended* to the image token sequence and are fed into the visual stream of the model (see Figure 3):

$$s = \big[y_1, \ldots, y_{N_y},\, c_1, \ldots, c_{N_c}\big]. \tag{2}$$

This simple *sequence concatenation* i) supports different input/output resolutions and aspect ratios, and ii) readily extends to multiple images $y_1, y_2, \ldots, y_N$. Channel-wise concatenation of x and y was also tested but degrades performance.

We encode positional information via 2D RoPE embeddings, where the embeddings for the context y receive a constant offset for all context tokens. We treat the offset as a 'virtual time step' that cleanly separates the context and target blocks while leaving their internal spatial structure intact. Concretely, if a token position is denoted by the triplet $\mathbf{u} = (t, h, w)$, then for each context token we set

$$\mathbf{u}_{y_i} \;=\; (\,t + \tau,\; h,\; w\,), \qquad \tau > 0, \tag{3}$$

so that the entire context block is phase shifted along the t axis while preserving its internal spatial geometry.

Rectified-flow objective. We train with a rectified flow–matching loss

$$\mathcal{L}_\theta = \mathbb{E}_{t\sim p(t),\, x, y, c}\big[\, \|v_\theta(z_t, t, s) - (\varepsilon - x)\|_2^2 \big], \tag{4}$$

where z_t is the linearly interpolated latent between x and noise $\varepsilon \sim \mathcal{N}(0, 1)$; $z_t = (1-t)x + t\varepsilon$. We use a shifted logit-normal schedule (see Appendix A.2) for $p(t; \mu, \sigma = 1.0)$, where we shift the mode μ depending on the data resolution during training.

When sampling pure text–image pairs ($y = \varnothing$) we omit all tokens y, preserving the text-to-image generation capability of the model. verify

Adversarial Diffusion Distillation Sampling of a flow matching model obtained by optimizing Equation (4) typically involves solving an ordinary or stochastic differential equation [26, 1], using

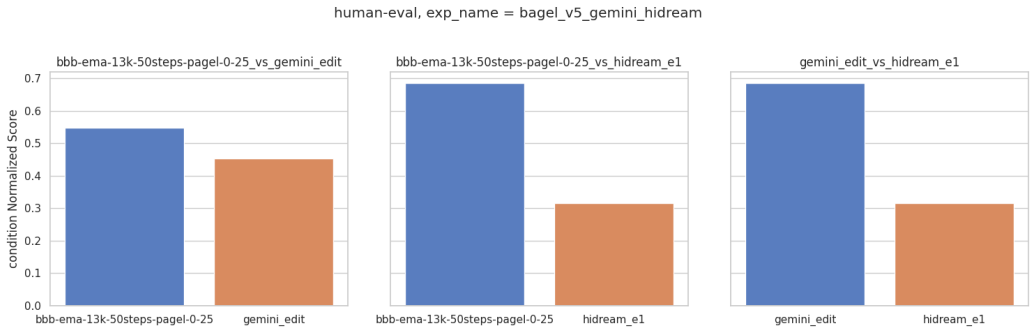

Figure 5: make this nice ofc. *FLUX.1 Kontext* is preferred to contemporary in-context LLMs and diffusion models, here Gemini2] and HiDream-E1]

50-250 guided [14] network evaluations. While samples obtained through such a procedure are of good quality for a well-trained model v_Θ, this comes with a few potential drawbacks: Firstly, such multi-step sampling is slow, rendering model-serving at scale expensive and hindering low-latency, interactive applications. Moreover, guidance may occasicanally introduce visual artifacts (e.g. high contrast) in the denoising process. We tackle both challenges using latent *adversarial diffusion distillation* [39, 40], reducing sampling step-count while increasing sample quality through adversarial training.

Implementation details. Starting from a pure text-to-image checkpoint, we fine-tune the model jointly on image-to-image and text-to-image tasks following Equation (4). We use FSDP2 [25] with mixed precision: all-gather operations are performed in `bfloat16` while gradient reduce-scatter uses `float32` for improved numerical stability. We use selective activation checkpointing [23] to reduce maximum VRAM usage. To improve throughput, we use *Flash Attention 3* [41] and regional compilation of individual Transformer blocks.

4 Evaluations & Applications

4.1 KontextBench – Crowd-sourced Real-World Benchmark for In-Context Tasks

motivation of why a new bench is needed? - other ppl have benches that are restricted in some ways, - omnigen used emu-edit and dreambench (from dreambooth). however emu edit are all images of lower-resolution and both the image as well as (image+)edit distribution doesn't match real world usecase. moreover its an edit only bench, whereas we are also do in-contenxt generation. dreambench is nice but small but doesn't span broad coverage of inputs or tasks. Magicbrush is another dataset people use, came before emu edit and the input images are same source as Emu edit (not good), its also instruction editing only

From emu edit paper . First, the InstructPix2Pix benchmark [2], which is intrinsically biased due to its reliance on generated Stable Diffusion [21] input images, and GPT3 [3] generated instructions. Consequently, it is unclear whether its results will truly mirror the performance on real input images, with genuine user instructions. Unlike InstructPix2Pix, the second benchmark, MagicBrush [29], uses a diverse set of authentic input images from the MS-COCO benchmark [5, 13], and annotator-defined instructions. Nonetheless, this dataset also suffers from inherent bias. During data collection, annotators were directed to use the DALLE-2 image editing platform [17] to generate the edited images. Thus, this benchmark is biased towards editing instructions that the DALLE-2 editor can successfully follow, which may compromise both its diversity and complexity.

GEdit-bench introduced in Step1x is really nice (better input image distribution) but again only instruction edit and not representative of the full scope of what omni/bagel style models can do in terms edit complexity.

IntelligentBench from ByteDance Bagel (https://arxiv.org/pdf/2505.14683) alleges to bench tasks that require "complex" multimodal reasoning and has some character reference examples. however

(a) Input image

(b) *"make me a matching flower vase, product photography set against a white wall, sitting on a wooden desk, put some nice flowers in it"*

(c) *"change the vase base color to black"*

Figure 6: **Iterative, product-style editing.** Starting from the reference bowl (a), our model first generates a matching flower vase in a tabletop studio setting with fresh flowers (b), and subsequently changes the vase's base color to black while preserving the floral pattern, lighting, and composition (c).

the benchmark is not publicly available the time of writing, is concurrent work, unsure about all the tasks it covers and is only 300 examples.

BOTTOM LINE: ppl can do quite broad set of tasks with Kontext style models, we set out to capture some real-world usecases for this models. hence from our team we crowd sourced several images and prompts for each edit they'd like to see. Note the images were images people wanted to edit and includes personal photos, CC attributed art images, public domain images from sources like Unsplash and Pexels, and AI-generated synthetic images. The edits were also user provided spanning local & global instruct editng, character reference, style reference & text editing. We collect a total of 1026 unique image / prompt pairs starting from 108 images. We also crowd-source users to annotate tags. For simplicity we enforce users to pick exactly one of the five tasks listed above. The distribution is shown in X (show a pie chart)

Category Total Instruction Editing - Local 416 Instruction Editing - Global 262 Text Editing 92 Style Reference 63 Character Reference 193

We found the bench to be quite representative of model capabilities and improvements in both qualitative and quantitative studies. We also observe automatic evaluations proposed in other methods noisy or the dataset is either too small / too big for doing human evals reliably. KontextBench with approx 1000 examples offers the right tradeoff of size and continuous human eval. Some examples are shown in Y.

for examples, follow links here: https://black-forest-labs.slack.com/archives/C087ZFHVDFU/p1748015866756229

add some examples, how to construct, what it covers that others do not - i.e. describe net new value we add by releasing this.

[TODO: Sumith to outline things]

(a) Input image (b) *"tilt her head towards the camera"* (c) *"make her laugh"*

Figure 7: **Sequential, facial-expression editing.** Beginning with the profile reference (a), our model first reorients the subject toward the camera (b) and then changes her expression to a spontaneous laugh (c), while preserving background, clothing and lighting. identity looks quite a bit different from b to c no?

4.2 SOTA Comparison

?? **A:** add section on bakeyness, let's coin it

4.3 Prompting with visual clues

red ellipsis ftw

4.4 Iterative Workflows

Maintaining the accurate appearance of characters and objects across multiple edits is crucial for storytelling, brand-sensitive applications, and the general ability to break down a complex edit into multiple steps. A major limitation of current state-of-the-art approaches is a noticeable visual drift when applying a chain of successive edits. Characters gradually lose their identity, and objects increasingly lose their defining features, drifting further from the source image with each edit. In order to assess the visual drift of different approaches, we focus on human character consistency across local and generative edits as a proxy for general visual drift. We apply a series of successive edits to photos of people and measure the cosine similarity of the AuraFace [9, 20] embeddings of the input image and the corresponding output image and observe 1) a significantly increased AuraFace similarity after one turn when using FLUX.1 Kontext compared to other models and 2) a slower decline in similarity in the following turns.

4.5 More researchy: What else does *FLUX.1 Kontext* learn?

A: optional. stuff like segmenation, bboxes, image correspondences, ...

5 Outlook

We introduced *FLUX.1 Kontext*, a flow-matching model that combines in-context image generation and editing in a single framework. Through simple sequence concatenation and training recipes, *FLUX.1 Kontext* achieves state-of-the-art performance while addressing key limitations: Character drift during multi-turn edits, slow inference, and output quality. Our contributions include a unified

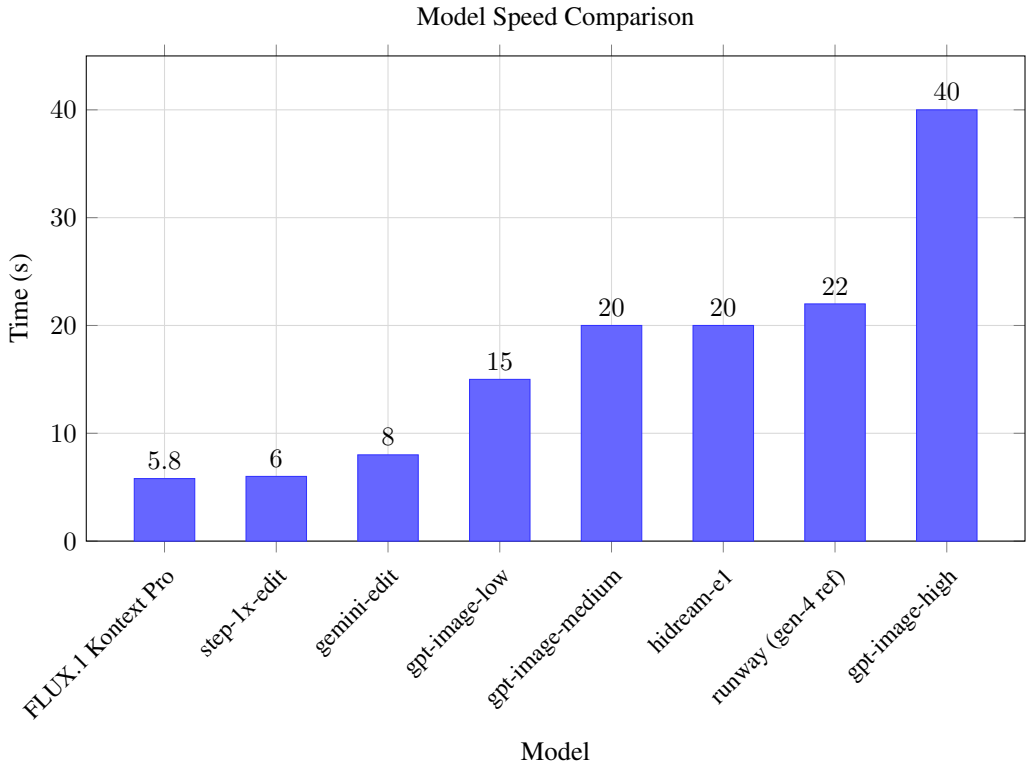

Figure 8: Median inference latency [seconds] update with final values for a 1024×1024 image edit (lower is better).

architecture that handles multiple processing tasks, superior character consistency across iterations, interactive speed, and KontextBench: A real-world benchmark with 1,026 image-prompt pairs. Our extensive evaluations reveal that *FLUX.1 Kontext* is comparable to proprietary systems while enabling fast, multi-turn creative workflows.

Future work should focus on extending to multiple image inputs via Equation (2), further scaling and reducing inference latency to unlock real-time applications. A natural extension of our approach is to include edits in the video domain. Most importantly, reducing degradation during multi-turn editing would enable infinitely fluid content creation. The release of *FLUX.1 Kontext* and KontextBench provides a solid foundation and a comprehensive evaluation framework to drive unified image generation and editing.

Figure 9: Local Edits: *FLUX.1 Kontext* (top) reliably only changes parts of the image, whereas *GPT-Image-1* (bottom) edits tend to alter the full canvas. The effect is particularly strong when considering multiple, iterative edits.

References

[1] Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2022.

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

[3] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.

[4] Frederic Boesel and Robin Rombach. Improving image editing models with generative data refinement. In *Tiny Papers @ ICLR*, 2024. URL https://api.semanticscholar.org/CorpusID:271461432.

[5] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[7] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. *arXiv preprint arXiv:2209.14491*, 2022.

[8] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[10] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

[11] 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.

[12] Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

[13] HiDream-ai. Hidream-e1: Instruction-based image editing model, 2025. URL https://github.com/HiDream-ai/HiDream-E1.

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[16] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images, 2023.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[18] Lianghua Huang, Wei Wang, Zhi-Fan Wu, Yupeng Shi, Huanzhang Dou, Chen Liang, Yutong Feng, Yu Liu, and Jingren Zhou. In-context lora for diffusion transformers. *arXiv preprint arXiv:2410.23775*, 2024.

[19] Imagen-Team-Google, :, Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brichtova, Andrew Bunner, Lluis Castrejon, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, Hongliang Fei, Nando de Freitas, Yilin Gao, Evgeny Gladchenko, Sergio Gómez Colmenarejo, Mandy Guo, Alex Haig, Will Hawkins, Hexiang Hu, Huilian Huang, Tobenna Peter Igwe, Christos Kaplanis, Siavash Khodadadeh, Yelin Kim, Ksenia Konyushkova, Karol Langner, Eric Lau, Rory Lawton, Shixin Luo, Soňa Mokrá, Henna Nandwani, Yasumasa Onoe, Aäron van den Oord, Zarana Parekh, Jordi Pont-Tuset, Hang Qi, Rui Qian, Deepak Ramachandran, Poorva Rane, Abdullah Rashwan, Ali Razavi, Robert Riachi, Hansa Srinivasan, Srivatsan Srinivasan, Robin Strudel, Benigno Uria, Oliver Wang, Su Wang, Austin Waters, Chris Wolff, Auriel Wright, Zhisheng Xiao, Hao Xiong, Keyang Xu, Marc van Zee, Junlin Zhang, Katie Zhang, Wenlei Zhou, Konrad Zolna, Ola Aboubakar, Canfer Akbulut, Oscar Akerlund, Isabela Albuquerque, Nina Anderson, Marco Andreetto, Lora Aroyo, Ben Bariach, David Barker, Sherry Ben, Dana Berman, Courtney Biles, Irina Blok, Pankil Botadra, Jenny Brennan, Karla Brown, John Buckley, Rudy Bunel, Elie Bursztein, Christina Butterfield, Ben Caine, Viral Carpenter, Norman Casagrande, Ming-Wei Chang, Solomon Chang, Shamik Chaudhuri, Tony Chen, John Choi, Dmitry Churbanau, Nathan Clement, Matan Cohen, Forrester Cole, Mikhail Dektiarev, Vincent Du, Praneet Dutta, Tom Eccles, Ndidi Elue, Ashley Feden, Shlomi Fruchter, Frankie Garcia, Roopal Garg, Weina Ge, Ahmed Ghazy, Bryant Gipson, Andrew Goodman, Dawid Górny, Sven Gowal, Khyatti Gupta, Yoni Halpern, Yena Han, Susan Hao, Jamie Hayes, Jonathan Heek, Amir Hertz, Ed Hirst, Emiel Hoogeboom, Tingbo Hou, Heidi Howard, Mohamed Ibrahim, Dirichi Ike-Njoku, Joana Iljazi, Vlad Ionescu, William Isaac, Reena Jana, Gemma Jennings, Donovon Jenson, Xuhui Jia, Kerry Jones, Xiaoen Ju, Ivana Kajic, Christos Kaplanis, Burcu Karagol Ayan, Jacob Kelly, Suraj Kothawade, Christina Kouridi, Ira Ktena, Jolanda Kumakaw, Dana Kurniawan, Dmitry Lagun, Lily Lavitas, Jason Lee, Tao Li, Marco Liang, Maggie Li-Calis, Yuchi Liu, Javier Lopez Alberca, Matthieu Kim Lorrain, Peggy Lu, Kristian Lum, Yukun Ma, Chase Malik, John Mellor, Thomas Mensink, Inbar Mosseri, Tom Murray, Aida Nematzadeh, Paul Nicholas, Signe Nørly, João Gabriel Oliveira, Guillermo Ortiz-Jimenez, Michela Paganini, Tom Le Paine, Roni Paiss, Alicia Parrish, Anne Peckham, Vikas Peswani, Igor Petrovski, Tobias Pfaff, Alex Pirozhenko, Ryan Poplin, Utsav Prabhu, Yuan Qi, Matthew Rahtz, Cyrus Rashtchian, Charvi Rastogi, Amit Raul, Ali Razavi, Sylvestre-Alvise Rebuffi, Susanna Ricco, Felix Riedel, Dirk Robinson, Pankaj Rohatgi, Bill Rosgen, Sarah Rumbley, Moonkyung Ryu, Anthony Salgado, Tim Salimans, Sahil Singla, Florian Schroff, Candice Schumann, Tanmay Shah, Eleni Shaw, Gregory Shaw, Brendan Shillingford, Kaushik

Shivakumar, Dennis Shtatnov, Zach Singer, Evgeny Sluzhaev, Valerii Sokolov, Thibault Sottiaux, Florian Stimberg, Brad Stone, David Stutz, Yu-Chuan Su, Eric Tabellion, Shuai Tang, David Tao, Kurt Thomas, Gregory Thornton, Andeep Toor, Cristian Udrescu, Aayush Upadhyay, Cristina Vasconcelos, Alex Vasiloff, Andrey Voynov, Amanda Walker, Luyu Wang, Miaosen Wang, Simon Wang, Stanley Wang, Qifei Wang, Yuxiao Wang, Ágoston Weisz, Olivia Wiles, Chenxia Wu, Xingyu Federico Xu, Andrew Xue, Jianbo Yang, Luo Yu, Mete Yurtoglu, Ali Zand, Han Zhang, Jiageng Zhang, Catherine Zhao, Adilet Zhaxybay, Miao Zhou, Shengqi Zhu, Zhenkai Zhu, Dawn Bloxwich, Mahyar Bordbar, Luis C. Cobo, Eli Collins, Shengyang Dai, Tulsee Doshi, Anca Dragan, Douglas Eck, Demis Hassabis, Sissie Hsiao, Tom Hume, Koray Kavukcuoglu, Helen King, Jack Krawczyk, Yeqing Li, Kathy Meier-Hellstern, Andras Orban, Yury Pinsky, Amar Subramanya, Oriol Vinyals, Ting Yu, and Yori Zwols. Imagen 3, 2024. URL https://arxiv.org/abs/2408.07009.

[20] isidentical. Introducing auraface: Open-source face recognition and identity preservation models. https://huggingface.co/blog/isidentical/auraface, 2024. Accessed: 2025-05-26.

[21] Kat Kampf and Nicole Brichtova. Experiment with gemini 2.0 flash native image generation, 2025. URL https://developers.googleblog.com/en/experiment-with-gemini-20-flash-native-image-generation/.

[22] Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[23] Vijay Anand Korthikanti, Jared Casper, Sangkug Lym, Lawrence McAfee, Michael Andersch, Mohammad Shoeybi, and Bryan Catanzaro. Reducing activation recomputation in large transformer models. *Proceedings of Machine Learning and Systems*, 5:341–353, 2023.

[24] 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.

[25] Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, et al. Torchtitan: One-stop pytorch native solution for production ready llm pre-training. *arXiv preprint arXiv:2410.06511*, 2024.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022.

[29] Midjourney. Midjourney, 2025. URL https://www.midjourney.com/home.

[30] OpenAI. Introducing 4o image generation, 2025. URL https://openai.com/index/introducing-4o-image-generation/.

[31] Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. Drag your gan: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023.

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023. doi: 10.1109/iccv51070.2023.00387. URL http://dx.doi.org/10.1109/ICCV51070.2023.00387.

[33] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.

[35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. doi: 10.1109/cvpr52688.2022.01042. URL http://dx.doi.org/10.1109/CVPR52688.2022.01042.

[36] 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. URL https://arxiv.org/abs/2208.12242.

[37] Inc. Runway AI. Runway | tools for human imagination, 2025. URL https://runwayml.com/.

[38] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–10, 2022.

[39] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.

[40] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation, 2024. URL https://arxiv.org/abs/2403.12015.

[41] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685, 2024.

[42] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023.

[43] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[44] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022.

[45] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[46] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024.

[47] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18381–18391, 2023.

[48] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[49] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[50] Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025.

A Image Generation using Flow Matching

A.1 Primer on Rectified Flow Matching

For training our models, we construct forward noising processes in the latent space of an image autoencoder as

$$z_t = a_t x_0 + b_t \varepsilon, \tag{5}$$

with $x_0 \sim p_{data}$, $\varepsilon \sim \mathcal{N}(0, 1)$, and the coefficients a_t and b_t define the log signal-to-noise ratio (log-SNR) [22]

$$\lambda_t = \log \frac{a_t^2}{b_t^2} \tag{6}$$

Further, we use the conditional flow matching loss [26]

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t\sim p(t),\varepsilon\sim\mathcal{N}(0,1)} ||v_\Theta(z_t, t) - \frac{a_t'}{a_t} z_t + \frac{b_t}{2}\lambda_t' \varepsilon||_2^2 \tag{7}$$

For rectified flow models [27], $a_t = 1 - t$ and $b_t = 1$, and thus

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t\sim p(t),\varepsilon\sim\mathcal{N}(0,1),x_0\sim p_{data}} ||v_\Theta(z_t, t) + x_0 - \varepsilon||_2^2 \tag{8}$$

and we sample t from a *Logit-Normal Distribution* [10]: $p(t) = \frac{\exp\left(-0.5\cdot(\text{logit}(t)-\mu)^2/\sigma^2\right)}{\sigma\sqrt{2\pi}\cdot(1-t)\cdot t}$, where $\text{logit}(t) = \log \frac{t}{1-t}$. From the definition of the Logit-Normal Distribution, it follows that a random variable $Y = \text{logit}(t) \sim \mathcal{N}(\mu, \sigma)$.

A.2 Expressing shifting of the timestep schedule via the Logit-Normal Distribution

Previous work on high-resolution image synthesis introduced an additional shift of the timestep sampling (and, equivalently, the log-SNR schedule) via a parameter α [10, 16]. Esser et al. [10] empirically demonstrated that $\alpha = 3.0$ worked best when increasing the image resolution from 256^2 to 1024^2. In the following, we show that this shifting can be expressed via the Logit-Normal Distribution.

Consider the log-SNR of a rectified flow forward process with $\mu = 0$ and $\sigma = 1$:

$$\lambda_t^{0,1} = 2 \log \frac{1-t}{t} = -2\text{logit}(t), \tag{9}$$

where $\text{logit}(t) \sim \mathcal{N}(0, 1)$. Expressing the log-SNR for arbitrary μ and σ gives

$$\lambda_t^{\mu,\sigma} = -2(\sigma \cdot \text{logit}(t) + \mu) = \sigma \cdot \lambda_t^{0,1} - 2\mu\,. \tag{10}$$

The α-shifted log-SNR [10, 16] is obtained as

$$\lambda_t^\alpha = \lambda_t^{0,1} - 2\log\alpha\,. \tag{11}$$

Comparing Equation (10) and Equation (11), we identify $\mu = \log\alpha$ for $\sigma = 1.0$, i.e. a shift of $\alpha = 3.0$ would correspond to a logit-normal distribution with $\mu = \log 3.0 = 1.0986$ and $\sigma = 1.0$.

We can further express the shifted log-SNR as a function of shifted timesteps t'

$$\lambda_{t'} = 2\log\frac{1-t'}{t'} = \sigma\lambda_t^{0,1} - 2\mu = 2\sigma\log\frac{1-t}{t} - 2\mu \tag{12}$$

and solve for t':

$$t' = \frac{e^\mu}{e^\mu + (1/t - 1)^\sigma} \tag{13}$$

For $\sigma = 1.0$ and $\mu = \log \alpha$ this recovers the redistribution function for the timesteps proposed in [10] $t' = \frac{\alpha t}{1+(\alpha-1)t}$, as expected. This generalized shifting formula 10 can be useful both for training and via 13 for inference.

A.3 Data Normalization is implicit Logit-Normal Shifting

We analyze how data normalization implicitly shifts the schedule towards higher noise scales in our framework.

Consider data with zero mean, $\mu_{data} = \mathbb{E}[x] = 0$ for $x \sim p_{data}$. Under a re-normalization operation [35]

$$\tilde{x} = \frac{x}{\sigma_{data}}, \tag{14}$$

the forward process 5 transforms as

$$\tilde{z}_t = a_t \tilde{x} + b_t \varepsilon = a_t \frac{x}{\sigma_{data}} + b_t \varepsilon = \tilde{a}_t x + b_t \varepsilon \tag{15}$$

with $\tilde{a}_t := \frac{a_t}{\sigma_{data}}$, and hence

$$\tilde{\lambda}_t = \log \frac{\tilde{a}_t^2}{b_t^2} = \log \frac{a_t^2}{\sigma_{data}^2 b_t^2} = \lambda_t^{0,1} - 2 \log \sigma_{data} \tag{16}$$

Comparing with 10, we find that re-normalization with σ_{data} is equivalent to a shift $\mu_{data} = \log \sigma_{data}$ of the logit-mean of λ_t.

This implicit shift under a re-normalization should be considered when doing multi-scale training since the standard deviation of the data distribution can vary across different dimensions []. show plot?

Summarizing our findings from A.2 and A.3 can express both the explicit shift of μ, e.g. due to higher data dimensions, and the implicit shift via re-normalization via

$$\mu_{total} = \mu_\alpha + \mu_{data}\,. \tag{17}$$