BlazeEdit: Generalist Image Editing on Mobile Devices with Image-to-Image Diffusion Models

Anonymous CVPR submission

Paper ID 2

Abstract

*The remarkable generation quality of modern diffusion models often comes at the cost of massive parameter counts, which necessitate server-side inference with significant computational costs and potential privacy risks. Consequently, there is growing momentum toward developing efficient on-device alternatives. While recent efforts have optimized text-to-image models for mobile hardware, they remain relatively bulky, typically ranging from 0.5B to 1B parameters. We present BlazeEdit, a highly efficient, generalist image-to-image diffusion model tailored for on-device deployment. By identifying that many practical image editing tasks do not require text-based guidance, we eliminate the text-conditioning components and develop a multi-task architecture that consolidates object removal, outpainting, tone correction, relighting, and sticker generation into a single, compact model of only **195M** parameters. BlazeEdit achieves a substantial reduction in download size and memory overhead while maintaining competitive generation quality. It completes a full inference pass in just **290ms** on a Pixel 10, delivering a seamless, privacy-preserving, and lightning-fast experience for generalist image editing on the edge.*

1. Introduction

Diffusion and flow-based generative models [1, 9, 14, 15, 20, 21] have achieved remarkable visual quality across a variety of digital content creation domains. However, many state-of-the-art diffusion models [4] employ heavy multimodal transformers whose parameter counts can reach up to 20B [25], requiring server-side inference on high-end GPUs/TPUs. This not only incurs significant computational costs, but also raises potential concerns regarding the privacy of uploaded personal photos.

Consequently, there is growing interest in developing on-device diffusion models that can run locally and efficiently. Recent efforts, such as SnapFusion [13], SnapGen [3], and MobileDiffusion [32], have made impressive strides in reducing the denoiser size and latency for text-to-image generation. Nevertheless, these models remain relatively heavy, as the text encoder increases the total model size. For instance, SnapGen leverages multiple text encoders including CLIP [17] and Gemma-2-2B [22], adding $\sim$2B parameters to its 0.38B denoiser. The large model size often presents a download barrier for mobile applications, where bandwidth is frequently limited.

In this paper, we challenge the prevailing assumption that a general-purpose mobile editing tool must be built upon a text-to-image foundation. We observe that a significant subset of common image editing actions, such as removing background objects, changing the aspect ratio of an image (*e.g.*, portrait to square or landscape), and creating stylized stickers from photos, can be sufficiently guided by the input image plus a user-provided mask, bypassing the need for text-based conditioning.

Building on this insight, we develop a pure image-to-image framework and introduce BlazeEdit, a highly efficient, generalist image editor tailored for mobile devices. Our contributions are summarized as follows:

- We develop a pretraining pipeline specifically for image-to-image diffusion models. By leveraging masked reconstruction as the pretraining objective, we endow the base model with foundational inpainting and outpainting capabilities, facilitating data-efficient downstream finetuning.
- We repurpose the mask value as a universal task indicator. Combined with a jointly trained image-and-mask encoder, this allows for simultaneous multi-task finetuning that enables knowledge transfer across tasks.
- We achieve a single, compact model of only **195M** parameters that consolidates five distinct editing tasks—object removal, outpainting, tone correction, relighting, and sticker generation. With an inference latency of just **290ms** on a Pixel 10, our model delivers a highly interactive user experience.

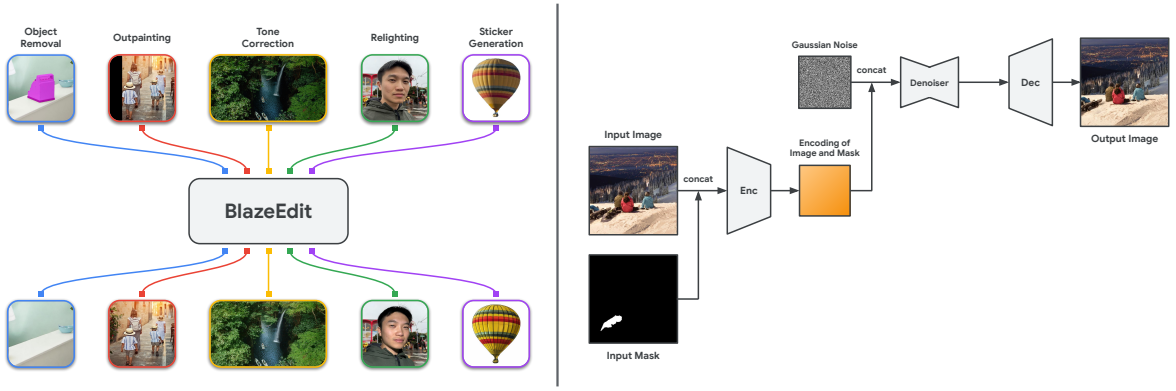

Figure 1. *(Left)* BlazeEdit is a generalist image-to-image editing framework designed for mobile devices. We achieve a single, compact model that consolidates five distinct editing tasks. *(Right)* BlazeEdit follows the latent diffusion paradigm, and employs a trainable image-and-mask encoder to provide the conditioning signal to the denoiser. The input resolution is 512×512, and latent resolution is 64×64.

2. Related Work

There have been significant recent advancements in developing generalist text-to-image diffusion models that unify diverse, multi-modal editing tasks within single architectures [5, 12, 16, 25–28, 31]. However, their reliance on heavy multi-modal transformers and massive parameter counts renders them impractical for on-device execution. A growing body of research has focused on compressing text-to-image diffusion models for mobile devices. Pioneering works such as SnapFusion [13], SnapGen [3], and MobileDiffusion [32] have significantly reduced the model parameters to the 0.5B $\sim$ 1B range through extensive architectural investigation and advanced distillation. Our work further shrinks the model size by shifting to an image-to-image framework that still supports a wide variety of practical editing tasks. Compared to the pixel-space image-to-image framework presented in Palette [19], our model works in latent space and introduces a task-agnostic pretraining phase, enabling fast inference and data-efficient finetuning.

3. Method

3.1. Model Architecture

Figure 1 presents an overview of the BlazeEdit framework. We detail our design below.

Latent Diffusion with Jointly Trained Encoder. Following established practice, we adopt latent diffusion [18] to reduce the computational overhead of the iterative denoising process. Given a frozen encoder $\mathcal{E}$ and decoder $\mathcal{D}$, we seek to train a denoiser ϵ_θ that models the conditional distribution $p(\mathcal{E}(\mathbf{y}) \mid \mathbf{x}, \mathbf{m})$, where $\mathbf{x}$, $\mathbf{y}$, $\mathbf{m}$ denote the input image, output image, and mask, respectively. The prevailing approach in pixel-space image-to-image frameworks, such as Palette [19], provides the masked image as a conditioning input to the denoiser. However, we find this approach suboptimal when adapted to the latent space. Specifically, if the conditioning input is simply changed to the latent representation of the masked image $\mathcal{E}(\mathbf{m} \odot \mathbf{x})$, the denoiser will struggle to preserve structural fidelity around the masked region, especially for object removal. This happens even if the autoencoder has no difficulty reconstructing the masked image, *i.e.*, $\mathcal{D}(\mathcal{E}(\mathbf{m} \odot \mathbf{x})) \approx \mathbf{m} \odot \mathbf{x}$. We attribute this to the suboptimal latent representations. Since the autoencoder is optimized mainly for reconstruction, its latent space is not inherently structured to decouple the original image content from the superimposed mask.

To address this, BlazeEdit introduces a secondary, trainable encoder $f_\theta(\texttt{concat}[\mathbf{x}, \mathbf{m}])$ that processes the concatenated input image and mask. The output of f_θ is fed to the denoiser as a conditioning input. Unlike the frozen, reconstruction-oriented autoencoder, f_θ is jointly optimized with the denoiser to extract task-relevant features that the model can more easily leverage. We find this joint training critical for maintaining fine-grained details and structural integrity. Furthermore, for object removal tasks where the mask specifies the object to be removed, providing f_θ with the full input image $\mathbf{x}$ allows the model to more effectively infer and eliminate shadows cast by the object.

Efficient Denoiser and Lightweight Decoder. The denoiser follows the U-ViT architecture [10]. Specifically, we employ ResNet [7] blocks at higher resolutions to maintain the spatial inductive bias while saving memory, and self-attention [23] blocks at lower resolutions to capture long-range dependencies and improve accelerator utilization. To further reduce model size and inference latency, we prune the width and depth of the decoder $\mathcal{D}$. We then train it for image reconstruction with the encoder $\mathcal{E}$ frozen, following MobileDiffusion [32]. The resulting lightweight decoder has only 6M parameters, and shows minimal degradation in reconstruction quality.

3.2. Pretraining via Masked Reconstruction

A significant bottleneck in developing a generalist image editor is the scarcity of high-quality, task-specific datasets. Existing literature typically follows one of two paths: (1) adapting a pretrained text-to-image model by adding task-specific LoRAs [11, 32], or (2) training an image-to-image model from scratch but limited to a small number of tasks, such as colorization and JPEG restoration [19], where massive paired data can be easily synthesized. The first approach inevitably involves parameters for text understanding and text-image alignment, which are unnecessary for purely image-conditioned tasks. On the other hand, the second approach struggles to scale to more complex editing tasks where the training data are costly to obtain.

To achieve data-efficient scaling across diverse editing tasks while maintaining a compact model footprint, we propose to first pretrain an image-to-image base model from scratch on large-scale task-agnostic datasets, and then finetune it jointly on all available task-specific datasets.

Inspired by the masked image modeling framework [2, 8], we use masked reconstruction as our pretraining objective. We find it critical to employ a diverse set of masks, including random patches, geometric shapes, and strokes within the image, and paddings on the boundary of the image. This not only encourages the model to learn an expressive image representation, but also equips the model with core inpainting and outpainting capabilities, building a generalizable foundation for specialized downstream tasks.

We directly leverage the images from existing large-scale text-to-image datasets for our pretraining. The loss function can be written as:

$$\mathbb{E}_{\mathbf{x},\mathbf{m},\boldsymbol{\epsilon},t}\|\epsilon_\theta(\tilde{\mathbf{x}}_t, f_\theta(\texttt{concat}[\mathbf{m}\odot\mathbf{x},\mathbf{m}]),t)-\boldsymbol{\epsilon}\|^2\,, \quad (1)$$

where $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{1})$ denotes the random noise, and $\tilde{\mathbf{x}}_t$ is the noised version of $\mathcal{E}(\mathbf{x})$ at a randomly sampled diffusion step t. The mask $\mathbf{m}$ is randomly generated on the fly. To prevent degenerate solutions caused by information leak, we mask the image $\mathbf{x}$ when feeding it to the trainable encoder f_θ. This is only necessary during pretraining.

3.3. Multi-Task Finetuning and Distillation

Following pretraining on large-scale, general-purpose image data, we transition to supervised finetuning and step distillation on task-specific datasets.

Universal Task Signaling. We conduct multi-task finetuning on all available downstream tasks simultaneously, as this enables knowledge transfer across tasks. To help the model distinguish between different tasks, we introduce a universal task signaling mechanism encoded directly within the mask values. Specifically, each task i is assigned a unique numerical constant τ_i, which is used to scale the binary mask $\mathbf{m}$. This mask-based conditioning signal allows the model to switch functional modes dynamically without additional parameters. For an input-mask-output triplet $(\mathbf{x},\mathbf{m},\mathbf{y})$ belonging to task i, the finetuning objective is formulated as:

$$\mathbb{E}_{\boldsymbol{\epsilon},t}\|\epsilon_\theta(\tilde{\mathbf{y}}_t, f_\theta(\texttt{concat}[\mathbf{x},\tau_i\cdot\mathbf{m}]),t)-\boldsymbol{\epsilon}\|^2\,, \quad (2)$$

where $\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{1})$ denotes the random noise, and $\tilde{\mathbf{y}}_t$ is the noised version of $\mathcal{E}(\mathbf{y})$ at a randomly sampled diffusion step t.

Adversarial Distribution Matching Distillation. After supervised finetuning, we distill the resulting model for 2-step inference, significantly reducing latency. We find that distribution matching distillation [30] combined with adversarial training [29] works well in this few-step regime.

4. Experiments

4.1. Tasks and Datasets

We first describe the range of image editing tasks supported by BlazeEdit and the task-specific finetuning datasets.

Object Removal aims to remove a user-specified object from a scene and synthesize a plausible background in its place, helping to reduce visual clutter. We use a manually curated dataset of ∼20K image pairs. Following [24], each pair captures a scene before and after an object is physically removed while minimizing other changes.

Outpainting extends the boundaries of an image, which is useful for changing its aspect ratio (*e.g.*, portrait to landscape). We use a high-quality subset of ∼5K images from our pretraining datasets, filtered for aesthetic and diversity.

Tone Correction enhances an image by refining its white balance, saturation, and exposure. We synthesize a dataset of ∼3M image pairs by a high-performance teacher model.

Relighting focuses on portrait enhancement by mitigating unfavorable lighting, such as harsh facial shadows. We use a dataset of ∼100K image pairs synthesized by applying diverse shadow augmentations to portrait photography [6].

Sticker Generation stylizes user-specified subjects within an image (*e.g.*, people, pets, and objects), transforming them into high-quality artistic stickers. We synthesize a diverse dataset of ∼100K image pairs using a high-capacity text-to-image generation model adept at identity-preserving stylization.

4.2. Main Results

Qualitative Results. In Figure 2, we present the input images and generation samples from BlazeEdit on all five tasks. BlazeEdit achieves compelling and versatile image-to-image editing. It exhibits strong structural reasoning and semantic preservation, seamlessly handling object and shadow removal, boundary extrapolation, stylization, tone correction, and lighting adjustment.

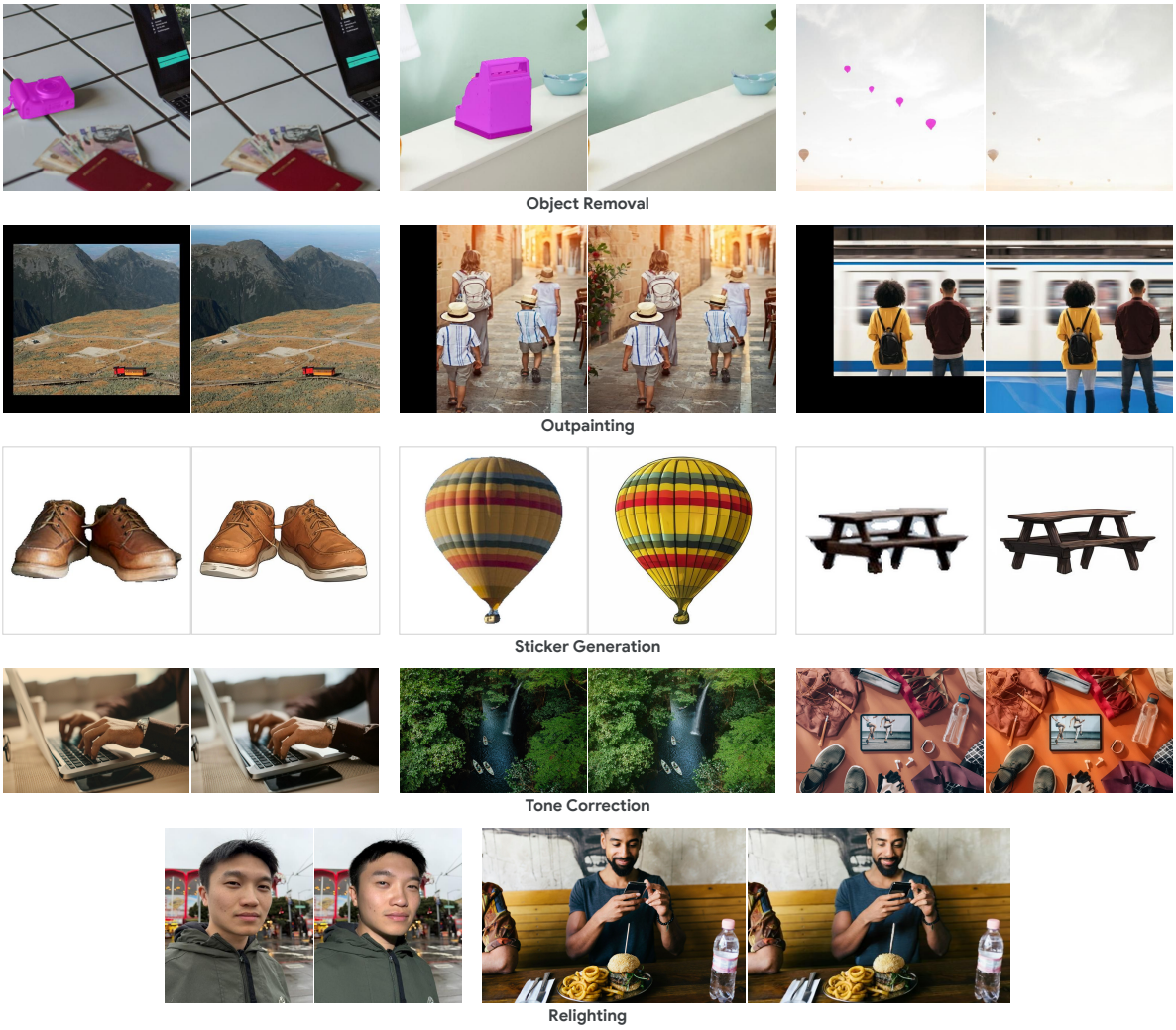

Figure 2. Qualitative results of BlazeEdit on five distinct editing tasks. BlazeEdit achieves compelling and versatile image-to-image editing. It exhibits strong structural reasoning and semantic preservation, seamlessly handling object and shadow removal, boundary extrapolation, stylization, tone correction, and lighting adjustment.

Efficiency Evaluation. In Table 1, we compare the model footprint of BlazeEdit with prior works. BlazeEdit achieves 2× reduction in denoiser parameter count while eliminating the text encoders, which typically range from 0.1B to 2B parameters. Furthermore, Table 2 provides a breakdown of inference latency on Pixel 10 using Edge TPU. BlazeEdit completes a full inference pass in just 290ms, enabling lightning-fast and privacy-preserving image editing directly on the edge.

Table 1. Model footprint comparison. BlazeEdit achieves 2× reduction in denoiser parameters while eliminating text encoders.

Model	Text Encoder	Denoiser #Params
SnapFusion [13]	CLIP-ViT-H	848M
MobileDiffusion [32]	CLIP-ViT-L	386M
SnapGen [3]	CLIP-ViT-L, CLIP-ViT-G, Gemma-2-2B	379M
BlazeEdit	None	189M

Table 2. Inference latency measurements. BlazeEdit completes a full inference pass in just 290ms, enabling lightning-fast and privacy-preserving image editing directly on the edge.

Device	Encoder	Decoder	Denoiser (2 steps)	Overall
Pixel 10 (Edge TPU)	45ms	55ms	190ms	290ms

5. Conclusion

We presented BlazeEdit, a highly efficient image-to-image diffusion model designed for edge-native generalist image editing. We successfully consolidated five diverse editing tasks into a unified 195M-parameter architecture. Our experiments demonstrate that BlazeEdit achieves competitive results while delivering highly interactive user experience on mobile hardware.

References

[1] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. 1

[2] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022. 3

[3] Jierun Chen, Dongting Hu, Xijie Huang, Huseyin Coskun, Arpit Sahni, Aarush Gupta, Anujraaj Goyal, Dishani Lahiri, Rajesh Singh, Yerlan Idelbayev, Junli Cao, Yanyu Li, Kwang-Ting Cheng, S.-H. Gary Chan, Mingming Gong, Sergey Tulyakov, Anil Kag, Yanwu Xu, and Jian Ren. SnapGen: Taming high-resolution text-to-image models for mobile devices with efficient architectures and training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7997–8008, 2025. 1, 2, 4

[4] 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, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024. 1

[5] Tsu-Jui Fu, Yusu Qian, Chen Chen, Wenze Hu, Zhe Gan, and Yinfei Yang. UniVG: A generalist diffusion model for unified image generation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17160–17170, 2025. 2

[6] David Futschik, Kelvin Ritland, James Vecore, Sean Fanello, Sergio Orts-Escolano, Brian Curless, Daniel Sỳkora, and Rohit Pandey. Controllable light diffusion for portraits. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8412–8421, 2023. 3

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 3

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 1

[10] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, 2023. 2

[11] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 3

[12] Duong H. Le, Tuan Pham, Sangho Lee, Christopher Clark, Aniruddha Kembhavi, Stephan Mandt, Ranjay Krishna, and Jiasen Lu. One diffusion to generate them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2671–2682, 2025. 2

[13] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. SnapFusion: Text-to-image diffusion model on mobile devices within two seconds. In *Advances in Neural Information Processing Systems*, 2023. 1, 2, 4

[14] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. 1

[15] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 1

[16] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, Stefano Ermon, Yun Fu, and Ran Xu. UniControl: A unified diffusion model for controllable visual generation in the wild. In *Advances in Neural Information Processing Systems*, 2023. 2

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn 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. 2

[19] 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. 2, 3

[20] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2015. 1

[21] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 1

[22] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison,

Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. 1

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017. 2

[24] Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. ObjectDrop: Bootstrapping counterfactuals for photorealistic object removal and insertion. In *European Conference on Computer Vision*, pages 112–129. Springer, 2024. 3

[25] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-Image technical report, 2025. 1, 2

[26] Bin Xia, Yuechen Zhang, Jingyao Li, Chengyao Wang, Yitong Wang, Xinglong Wu, Bei Yu, and Jiaya Jia. DreamOmni: Unified image generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28533–28543, 2025.

[27] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13294–13304, 2025.

[28] Xingqian Xu, Zhangyang Wang, Gong Zhang, Kai Wang, and Humphrey Shi. Versatile Diffusion: Text, images and variations all in one diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7754–7765, 2023. 2

[29] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in neural information processing systems*, 2024. 3

[30] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 3

[31] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *Advances in neural information processing systems*, 2023. 2

[32] Yang Zhao, Yanwu Xu, Zhisheng Xiao, Haolin Jia, and Tingbo Hou. MobileDiffusion: Instant text-to-image generation on mobile devices. In *European Conference on Computer Vision*, pages 225–242. Springer, 2024. 1, 2, 3, 4