KOALA: Fast and Memory-Efficient Latent Diffusion Models via Self-Attention Distillation

Youngwan Lee[1,2] Kwanyong Park[1] Yoorhim Cho[3] Yong-Ju Lee[1] Sung Ju Hwang[2,4]

[1]Electronics and Telecommunications Research Institute (ETRI), South Korea
[2]Korea Advanced Institute of Science and Technology (KAIST), South Korea
[3]Sookmyung Women's University, South Korea
[4]DeepAuto.ai, South Korea

project page: https://youngwanlee.github.io/KOALA/

Abstract

This work explores an empirical yet effective strategy for compressing Stable Diffusion XL (SDXL) through a knowledge distillation (KD) scheme. We first design a more efficient U-Net using layer-level pruning. Secondly, we explore how to effectively distill the generation capability of SDXL into an efficient U-Net and eventually identify four essential factors, the core of which is that self-attention is the most crucial part. With our efficient U-Net and self-attention-based KD strategy, we build our efficient text-to-image models, called KOALA-1B &-700M, while reducing the model size up to 54% and 69% of the SDXL model. In particular, the KOALA-700M is over twice as fast as the SDXL while still maintaining satisfactory generation quality. Moreover, unlike SDXL, our KOALA models can generate 1024px high-resolution images on consumer-grade GPUs (e.g., 8GB VRAM). We hope that thanks to its balanced speed-performance tradeoff, our KOALA models can serve as a cost-effective alternative to SDXL.

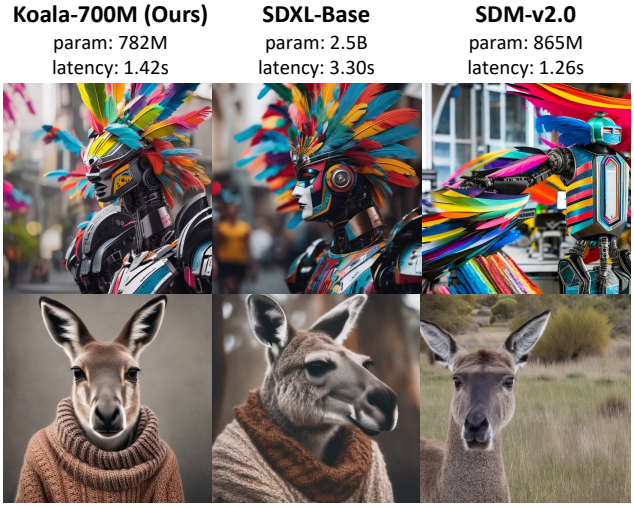

Figure 1. **Qualiatative comparison with SDM-v2.0 and SDXL.** With the following settings: FP-16 precision, $\mathbf{1024^2}$ resolution, and 25 denoising steps on NVIDIA 4090 GPU. The prompts are described in Appendix B.

1. Introduction

The emergence of the Stable diffusion models (SDMs) [20, 25] not only advance text-to-image synthesis but also revolutionize derivative applications such as image editing [5, 35] and text-to-video generation [3]. Furthermore, a more recent version of the SDMs, SDXL [20], enables to generate **higher resolution** images of $\mathbf{1024^2}$ with significantly improved quality. However, its massive computation costs and large model size require expensive hardware equipment and thus incur huge costs.

To alleviate this computation burden, prior works attempt to either reduce the required sampling steps (*i.e.*, step-distillation) [19, 28] or compress the model architecture [15, 17] through the knowledge distillation (KD) scheme [6, 9]. For the architectural compression, BK-SDM [15] exploits KD to compress computationally heavy U-Net [27] part in SDM-v1.4 [24]. BK-SDM builds a compressed U-Net by simply removing some blocks and allows the compressed U-Net to mimic the last features at each stage of the original U-Net. However, the compression method proposed by BK-SDM achieves a limited compression rate (33% in Tab. 1) when applied to the larger SDXL than SDM-v1.4, and the strategy of feature distillation for SDXL has *not yet been fully explored*.

In this work, our goal is to build more efficient U-Nets by distilling the generation capability of significantly larger U-Net in SDXL [20]. To this end, we first design two efficient U-Nets, **KOALA-1B** and **KOALA-700M**, using not only block removal but also *layer-wise pruning* to reduce

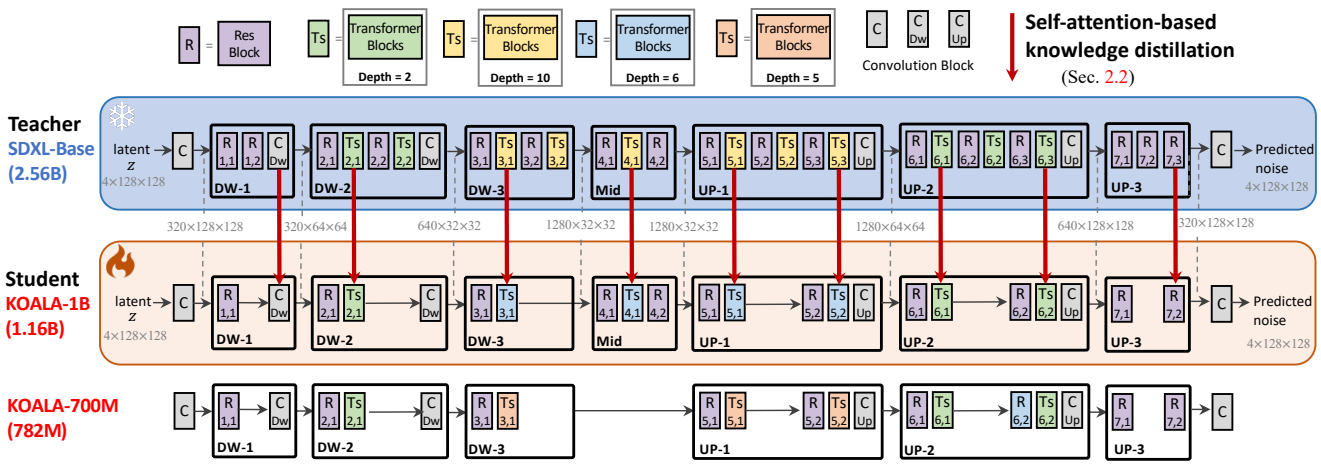

Figure 2. **Overview of KnOwledge-DistillAtion in LAtent diffusion model based on SDXL and architecture of KOALA.** We omit skip connections for simplicity. We perform feature distillation in transformer blocks using self-attention layers.

the model size of the SDXL's U-Net by up to 54% and 69% (vs. BK's method: 33%). Furthermore, we investigate how to effectively distill SDXL as a teacher model and find ***four essential factors*** for feature-level knowledge distillation. The core of these findings is that *self-attention* features are the most crucial for distillation.

With the proposed strategies, we train an efficient text-to-image synthesis model on top of SDXL [20], called KOALA, by only replacing SDXL's U-Net with our efficient U-Net. KOALA is trained with 1024^2 resolution on a smaller *publicly available* LAION-Aesthetics-V2-6+ [30], which has only 8M text-image pairs. Our efficient KOALA models consistently outperform BK-SDM [15]'s KD methods. Furthermore, our smaller model, KOALA-700M, shows better performance than widely used SDM-v2.0 [25], while having a similar model size and inference speed. Lastly, to validate its practical impact, we perform inference analysis on a variety size of ***consumer-grade* GPUs** (*e.g.*, memory of 8GB, 11GB, and 24GB), and the results show that SDXL cannot be mounted on an 8GB GPU, whereas our KOALA-700M can run on it while still maintaining satisfactory image quality as shown in Fig. 1.

Our main contributions are as follows:

1. We design two efficient denoising U-Net with model sizes (1.13B/782M) that are more than twice as compact and faster than SDXL's U-Net (2.56B).
2. We perform a comprehensive analysis of the knowledge distillation strategies for SDXL, finding four essential factors for feature distillation.

2. Approach

In this section, we first propose an efficient U-Net architecture in Sec. 2.1. Then, we explore how to effectively distill the knowledge from U-Net in SDXL [20] into the proposed efficient U-Net in Sec. 2.2.

U-Net	SDM-v2.0	SDXL-1.0	BK-SDXL	**KOALA-1B**	**KOALA-700M**
#Param.	865M	2,567M	1,717M	1,161M	782M
CKPT size	3.46GB	10.3GB	6.8GB	4.4GB	3.0GB
Tx blocks	[1, 1, 1, 1]	[0, 2, 10]	[0, 2, 10]	[0, 2, 6]	[0, 2, 5]
Mid block	✓	✓	✓	✓	✗
Latency	1.13s	3.13s	2.42s	1.60s	1.25s

Table 1. **U-Net Comparison.** Tx means Transformer. SDM-v2.0 [25] uses 768^2 resolution, while SDXL and KOALA models use 1024^2 resolution. Latency is measured with FP16, and 25 denoising steps in NVIDIA 4090 GPU. CKPT means the trained checkpoint file.

2.1. Efficient U-Net architecture

We devise a compressed U-Net that is more *suitable* for SDXL by introducing *layer-level pruning* compared to that of BK-SDM [15] which uses only block-level pruning. Similar to BK-SDM, we first remove the residual-transformer blocks pair at each stage. Specifically, in the encoder part (`DW-i`), each stage has two alternating pairs of a residual block and transformer blocks. We remove the last pair of residual-transformer blocks at each stage. In the decoder part (`UP-i`), we remove the intermediate pair of residual-transformer blocks. Furthermore, focusing on the fact that the majority of the parameters are concentrated on the transformer blocks at the lowest features, as shown in Fig. 2, we reduce the number of layers (*i.e.*, depth) in the transformer blocks from 10 to 5 or 6 at the lowest features (*i.e.*, `DW-3`, `Mid` and `UP-1` in Fig. 2). As a result, we design two types of compressed U-Net, KOALA-1B and KOALA-700M. More details of the proposed U-Nets are demonstrated in Tab. 1 and Fig. 2. Note that we remove `Mid` block in KOALA-700M for additional model compression. Our KOALA-1B model has 1.16B parameters, making it twice as compact as SDXL (2.56B). Meanwhile, KOALA-700M, with its 782M parameters, is comparable in size to SDM-v2.0 (865M).

Distill type	HPSv2
SD-loss	25.53
SA	**26.74**
CA	26.11
Res	26.27
FFN	26.48
LF (BK-SDM [15])	26.63

(a) **Distillation type.**

Distill loc.	HPSv2
SD-loss	25.53
DW-2	25.32
DW-3	25.57
Mid	25.66
UP-1	**26.52**
UP-2	26.05

(b) **Distill stage.**

SA loc.	HPSv2
SA-bottom	**26.74**
SA-interleave	26.58
SA-up	26.48

(c) **SA location.**

Combination	HPSv2
Baseline (SA only)	26.74
SA + LF at DW-1 & UP-3	**26.98**
SA + Res at DW-1 & UP-3	26.94
SA + LF all	26.83
SA + Res all	26.80
SA+CA+Res+FFN+LF all	26.39

(d) **Combination.**

Table 2. **Analysis of feature level knowledge distillation of U-Net in SDXL [20].** SA, CA, and FFN denote self-attention, cross-attention, and feed-forward net in the transformer block. Res is a convolutional residual block and LF denotes the last feature (same in BK [15]).

2.2. Exploring Knowledge distillation for SDXL

Now we explore how to effectively distill the knowledge of U-Net in SDXL [20] into the proposed compact U-Net described in Sec. 2.1. Prior work [15] that attempts to distill an early series of stable diffusion (*i.e.*, SDM-v1.4 [24]) directly follows traditional knowledge distillation literature [6, 26]. The compressed student U-Net model S_θ is jointly trained to learn the target task and mimic the pre-trained U-Net of SDM-v1.4 as a teacher network. Here, the target task is the reverse denoising process [11], and we denote the corresponding learning signal as $\mathcal{L}_{\text{task}}$. Besides the task loss, the compressed student model is trained to match the output of the pre-trained U-Net at both output and feature levels. $\mathcal{L}_{\text{out}}$ and $\mathcal{L}_{\text{feat}}$ represent the knowledge distillation (KD) loss at the output- and feature-level, respectively. For designing the feature-level KD-loss, BK-SDM [15] simply considers only the last feature map of the teacher $f_T^i(\cdot)$ and student network $f_S^i(\cdot)$ at each i stage as follows:

$$\mathcal{L}_{\text{featKD}} = \min_{S_\theta} \mathbb{E}_{z,\epsilon,c,t} || \sum_i f_T^i(z_t, t, c) - f_S^i(z_t, t, c) ||_2^2, \tag{1}$$

where t and c denote given diffusion timestep and text embeddings as conditions. Thus, the feature distillation approach for text-to-image diffusion models has ***not been sufficiently explored***, leaving room for further investigation.

In this work, we extensively explore feature distillation strategies to distill the knowledge from the U-Net of SDXL effectively to our efficient U-Net, KOALA-1B. We start from a baseline trained only by $\mathcal{L}_{\text{task}}$ and add $\mathcal{L}_{\text{featKD}}$ without $\mathcal{L}_{\text{outKD}}$ to validate the effect of feature distillation. More training details are described in Sec. 3 and Appendix A. From the experiments as shown in Tab. 2, We summarize our insights into ***four important findings*** as follows.

F1. Which feature type is effective for distillation? BK-SDM [15] demonstrated that distilling the last features (`LF`) at U-Net stages benefits overall performance when applied to shallow U-Net of early SDM-v1.4 [24]. However, with the increasing complexity of U-Net and its stage, relying solely on `LF` may not be sufficient to mimic the intricate behavior of the teacher U-Net. Thus, we revisit which features provide the richest guidance for effective knowledge distillation. We focus on key intermediate features from each stage: outputs from the self-attention (`SA`), cross-attention (`CA`), and feedforward net (`FFN`) in the transformer block, as well as outputs from convolutional residual block (`Res`) and `LF`. Tab. 2a summarizes the experimental results. While all types of features help obtain higher performance over the naïve baseline with only the task loss, distilling ***self-attention features*** achieves the most performance gain. Considering the prior studies [16, 32, 34] which suggest that `SA` plays a vital role in capturing semantic affinities and the overall structure of images, the results emphasize that such information is crucial for the distillation process.

F2. Which stage is most effective for distillation? In addition, we ablate the significance of each self-attention stage in the distillation process. Specifically, we adopt an `SA`−based loss at a single stage alongside the task loss. As shown in Tab. 2b, the results align with the above understanding: distilling self-attention knowledge within the ***decoder*** stages significantly enhances generation quality. In comparison, the impact of self-attention solely within the encoder stages is less pronounced. Consequently, we opt to retain more `SA` layers within the decoder (see Fig. 2).

F3. Which SA's location is effective in the transformer blocks? At the lowest feature level, the depth of the transformer blocks is 6 for KOALA-1B, so we need to decide which locations to distill from the 10 transformer blocks of teacher U-Net. We assume three cases for each series of transformer blocks; (1) `SA-bottom`: $\{f_T^l \mid l \in \{1,2,3,4,5\}\}$, (2) `SA-interleave`: $\{f_T^l \mid l \in \{1,3,5,7,9,10\}\}$, and (3) `SA-up`: $\{f_T^l \mid l \in \{6,7,8,9,10\}\}$ where l is the number of block. Tab. 2c shows that `SA-bottom` performs the best while `SA-up` performs the worst. This result suggests that the features of the early blocks are more significant for distillation. Therefore, we adopt the `SA-bottom` strategy in all experiments.

F4. Which combination is the best? In SDXL's U-Net, as shown in Fig. 2, there are no transformer blocks at the highest feature levels (*e.g.*, `DW-1&UP-3`); consequently, self-attention features cannot be distilled at this stage. Thus, we try two options: the residual block (`Res at DW-1&UP-3`) and the last feature (`LF at DW-1&UP-3`) as BK-SDM [15]. To this end, we perform SA-based feature distillation at every stage except for `DW-1` and `UP-3`,

Model	Param.(Whole/U-Net)	HPSv2	CompBench
SDM-v1.4 [24]	1.04B/860M	26.95	0.3150
SDM-v2.0 [25]	1.28B/865M	27.13	0.3672
DALL.E-2 [23]	6.5B	26.95	0.4268
SDXL-Base [20]	3.46B/2.6B	27.73	0.4441
BK-SDXL-700M [15]	1.68B/782M	27.26	0.3723
KOALA-700M	1.68B/782M	27.43	0.3791
BK-SDXL-1B [15]	2.06B/1.16B	27.12	0.3719
KOALA-1B	2.06B/1.16B	27.44	0.3912

Table 3. **Visual aesthetics evaluation** using HPSv2 [38] (Left) and **Image-text alignment evaluation** using T2I-CompBench [12] (Right).

where we use the above two options, respectively. In addition, we try additional combinations: `SA+LF all`, `SA+Res all`, and `SA+CA+Res+FFN+LF all` where `all` means all stages. Tab. 2d demonstrates that adding more feature distillations to the SA-absent stage (*e.g.*, `DW-1&UP-3`) consistently boots performance, and especially `LF at DW1&UP3` shows the best. Interestingly, both `+LF all` and `+Res all` are worse than the ones at only `DW-1&UP-3` and `SA+CA+Res+FFN+LF all` is also not better, demonstrating that the SA features are not complementary to the other features.

With these findings, we build a **K**n**O**wledge-distill**A**tion-based **LA**tent diffusion model with our efficient U-Nets, called KOALA. We train our KOALA models with the following objectives: $\mathcal{L}_{\text{task}} + \mathcal{L}_{\text{outKD}} + \mathcal{L}_{\text{featKD}}$ where we apply our findings to $\mathcal{L}_{\text{featKD}}$.

3. Experimental results

Implementation details. We train the proposed efficient U-Net in SDXL [20] for 200K iterations with a batch size of 128 on publicly available LAION-Aesthetics V2 6+ [29–31] for reproducibility. For a fair comparison to our counterpart BK-SDM [15], we train our efficient U-Nets with their distillation method under the same data setup (*e.g.*, BK-SDXL-1B and -700M in Tab. 3). More training details are described in Appendix A.

Evaluation metric. Recently, several works [2, 20, 38] have claimed that FID [8] is not closely correlated with visual fidelity. Therefore, instead of FID, we use Human Preference Score (HPSv2) [38] as a visual aesthetics metric. For image-text alignment, we use the T2I-compbench [12], which is a more comprehensive benchmark than CLIP score [7].

3.1. Main results

We compare our KOALA-700M/1B models with popular open-sourced Stable diffusion models series [20, 24, 25] and DALLE-2 [23] in Tab. 3. Our KOALA-700M & KOALA-1B models based on SDXL [20] consistently achieve both higher HPS and CompBench average scores than the BK [15] models (BK-SDXL-700M & 1B) equipped with our efficient U-Net. Furthermore, our KOALA-700M surpasses SDM-v2.0 [25] with a comparable U-Net size, which is widely used in the community. In addition, our KOALA models show higher HPS but lower Compbench than DALLE-2 [23], which has a much larger model size (6.5B). We speculate that the different tendency between DALLE-2 and our model may stem from data used for training. Because the LAION-Aesthetics data we used focuses on higher aesthetic images than multiple objects with various attributes, our model is vulnerable to texts with different attribute properties. Lastly, when measuring latency and memory usage on a consumer-grade GPU (*e.g.*,11GB of VRAM), as shown in Fig. 3, SDXL cannot run with FP32 precision, whereas our KOALA-700M operates twice as fast using both FP16 and FP32 precision, showing speeds comparable to SDM-v2.0.

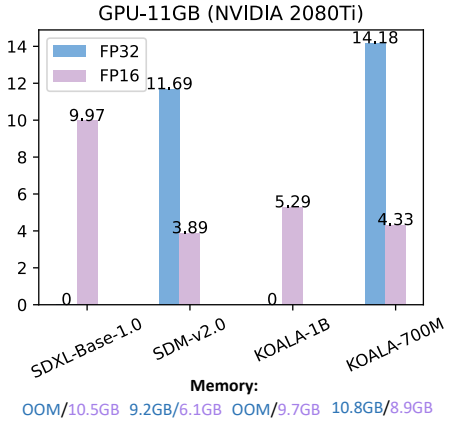

Figure 3. **Latency and memory usage comparison on a consumer-grade GPU (11GB)**. OOM means *Out-of-Memory*. We measure the inference time of SDM-v2.0 with 768px and the other models with 1024px resolution, using 25 denoising steps.

4. Conclusion and Future works

In this work, we propose KOALA, an efficient text-to-image synthesis model, offering a compelling alternative between SDM-v2.0 and SDXL in resource-limited environments. To achieve this, we devise more compact U-Nets and explore effective knowledge distillation strategies. With these contributions, our KOALA-700M model substantially reduces the model size (69%↓) and the latency (60%↓) of SDXL while exhibiting decent aesthetic generation quality.

For future works, as the recent step-distillation methods (*e.g.*, SDXL-Turbo [28] and LCM [19]) are ***orthogonal*** to our approach, it is expected that integrating our efficient KOALA U-Net as a substitute for the SDXL backbone can create synergistic effects, which leads to further speed-up.

Acknowledgement This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-00187238, Development of Large Korean Language Model Technology for Efficient Pre-training).

References

[1] Stable-diffusion-xl-base-1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2023. 7

[2] Eyal Betzalel, Coby Penso, Aviv Navon, and Ethan Fetaya. A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935*, 2022. 4

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 1

[4] Ollin Boer Bohan. Sdxl-vae-fp16-fix. https://huggingface.co/madebyollin/sdxl-vae-fp16-fix, 2023. 7

[5] Jooyoung Choi, Yunjey Choi, Yunji Kim, Junho Kim, and Sungroh Yoon. Custom-edit: Text-guided image editing with customized diffusion models. In *CVPR-Workshop*, 2023. 1

[6] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, 2019. 1, 3

[7] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 4

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 4

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 7

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3, 7

[12] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. In *NeurIPS*, 2023. 4

[13] Gabriel Ilharco, Mitchell Wortsman, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip. https://github.com/mlfoundations/open_clip, 2021. 7

[14] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 7

[15] Bo-Kyeong Kim, Hyoung-Kyu Song, Thibault Castells, and Shinkook Choi. On architectural compression of text-to-image diffusion models. *arXiv preprint arXiv:2305.15798*, 2023. 1, 2, 3, 4, 7

[16] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *CVPR*, 2019. 3

[17] 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. *arXiv preprint arXiv:2306.00980*, 2023. 1

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[19] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 1, 4

[20] 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. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 3, 4, 7

[21] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Stabilityai: Sdxl-base-1.0. https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0, 2023. 7

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 7

[23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 4

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable-diffusion-v1.4. https://github.com/CompVis/stable-diffusion, 2022. 1, 3, 4

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable-diffusion-v2.0. https://github.com/Stability-AI/stablediffusion, 2022. 1, 2, 4

[26] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 1

[28] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023. 1, 4

[29] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-aestheics v2. https://laion.ai/blog/laion-aesthetics/, 2022. 4

[30] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-aesthetics v2 6+. https://huggingface.co/datasets/

ChristophSchuhmann/improved_aesthetics_6plus, 2022. 2, 7

[31] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 4

[32] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 3

[33] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 7

[34] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In *CVPR*, 2022. 3

[35] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, 2023. 1

[36] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 7

[37] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers/blob/main/examples/text_to_image/train_text_to_image_sdxl.py, 2023. 7

[38] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 4

Appendix

A. Implementation details

A.1. Training

We base our framework on the officially released SDXL-Base-1.0 [1] and `Diffusers` library [36, 37]. We mainly replace computationally burdened SDXL's U-Net with our efficient U-Net. We keep the same two text encoders, OpenCLIP ViT-bigG [13] and CLIP ViT-L [22], used in SDXL. For VAE, we use `sdxl-vae-fp16-fix` [4], which enables us to use FP16 precision for VAE computation. We initialize the weights of our U-Net with the teacher's U-Net weights at the same block location. We freeze the text encoders, VAE, and the teacher U-Net of SDXL and only fine-tune our U-Net.

We train our KOALA models on LAION-Aesthetics V2 6+ [30] dataset (about 800M text-image pairs) for 200K iterations using four NVIDIA A100 (80GB) GPUs with a resolution of 1024×1024, a discrete-time diffusion schedule [11], size- and crop-conditioning as in SDXL [20], a batch size of 128, AdamW optimizer [18], a constant learning rate of 10^{-5}, and FP16 precision. For a fair comparison to our counterpart BK-SDM [15], we train our efficient U-Nets with their distillation method under the same data setup (*e.g.*, BK-SDXL-1B and -700M in Tab. 3). For the ablation studies in Tab. 2, we train all models for 30K iterations with a batch size of 32 on LAION-Aesthetics V2 6.5+ datasets for fast verification.

A.2. Inference

When generating samples, we also generate images with a resolution of $\mathbf{1024} \times \mathbf{1024}$, FP16-precision and `sdxl-vae-fp16-fix` [4] for VAE-decoder. Note that in the SDXL original paper [20], authors used DDIM sampler [33] to generate samples in the figures while the diffuser's official SDXL code [21] used Euler discrete scheduler [14] as the default scheduler. Therefore, we also use the Euler discrete scheduler for generating samples. With the Euler discrete scheduler, we set the denoising step to 50 only for quantitative evaluation in Tab. 2 and Tab. 3, and set it to 25 for other qualitative results or latency measurements. we set classifier-free guidance [10] to 7.5. We note that we generated samples in Fig. 1 on NVIDIA 4090 GPUs. For measuring latency and memory usage in fair conditions, we construct the same software environments across machines with different GPUs. Specifically, we use `Pytorch` v2.0.1 and `Diffusers` v0.20.0.

A.3. Detailed formulation of training objectives

We detail the two objectives, the $\mathcal{L}_{\text{task}}$ and $\mathcal{L}_{\text{out}}$, which are omitted in the main paper. First, the target task loss $\mathcal{L}_{\text{task}}$ to learn reverse denoising process [11] is summarized as:

$$\mathcal{L}_{\text{task}} = \min_{S_\theta} \mathbb{E}_{z_t,\epsilon,t,c} ||\epsilon_t - \epsilon_{S_\theta}(z_t, t, c)||_2^2, \tag{2}$$

where ϵ_t is the ground-truth sampled Gaussian noise at timestep t, c is text embedding as a condition, and $\epsilon_{S_\theta}(\cdot)$ denotes the predicted noise from student U-Net model, respectively. Second, the output-level knowledge distillation (KD) loss is formulated as:

$$\mathcal{L}_{\text{outKD}} = \min_{S_\theta} \mathbb{E}_{z,\epsilon,t,c} ||\epsilon_{T_\theta}(z, t, c) - \epsilon_{S_\theta}(z, t, c)||_2^2, \tag{3}$$

where $\epsilon_{T_\theta}(\cdot)$ denotes the predicted noise from each U-Net in the teacher model.

B. Represenative prompts in Fig. 1

We use the following prompts for Fig. 1.

- *"A portrait photo of a kangaroo in a sweater."*
- *"A cinematic shot of a robot with colorful feathers."*