BK-SDM: Architecturally Compressed Stable Diffusion for Efficient Text-to-Image Generation

Bo-Kyeong Kim [1] **Hyoung-Kyu Song** [1] **Thibault Castells** [1] **Shinkook Choi** [1]

Abstract

Exceptional text-to-image (T2I) generation results of Stable Diffusion models (SDMs) come with substantial computational demands. To resolve this issue, recent research on efficient SDMs has prioritized enabling fewer sampling steps and utilizing network quantization. Orthogonal to these directions, this study highlights the power of classical architectural compression for general-purpose T2I synthesis by introducing block-removed knowledge-distilled SDMs (BK-SDMs). We eliminate several residual and attention blocks from the U-Net of SDMs, obtaining over a 30% reduction in the number of parameters, MACs per sampling step, and latency. We conduct distillation-based pretraining with only 0.22M LAION pairs (fewer than 0.1% of the full training pairs) on a single A100 GPU. Despite being trained with limited resources, our compact models can imitate the original SDM by benefiting from transferred knowledge and achieve competitive results against larger multi-billion parameter models on the zero-shot MS-COCO benchmark. Moreover, we show the applicability of our lightweight pretrained models in personalized generation with DreamBooth finetuning.

1. Introduction

Stable Diffusion models (SDMs) (Rombach & Esser, 2022; Rombach et al., 2022) are one of the most renowned open-source models for text-to-image (T2I) synthesis, and their exceptional capability has begun to be leveraged as a backbone in several text-guided vision applications (Brooks et al., 2023; Blattmann et al., 2023). SDMs are T2I-specialized latent diffusion models (LDMs) (Rombach et al., 2022), which employ diffusion operations (Ho & Salimans, 2021; Song et al., 2021) in a latent space to improve compute efficiency. Within a SDM, a U-Net (Ronneberger et al., 2015) conducts an iterative sampling procedure to gradually eliminate noise from random latents and is assisted by a text encoder (Radford et al., 2021) and an image decoder (Van Den Oord et al., 2017) to produce text-aligned images. This process still involves excessive computational demands (see Figure 1), which often hinder the utilization of SDMs despite their rapidly growing usage.

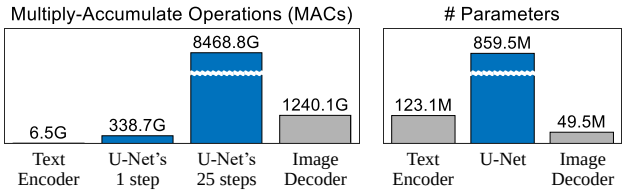

Figure 1. Computation of the major components in SDM-v1. The denoising U-Net is the main processing bottleneck. THOP (Zhu, 2018) is used to measure MACs in generating a 512×512 image.

To tackle this issue, numerous studies toward efficient SDMs have been introduced (see Appendix A for details). Meng et al. (2023; 2022) reduce the number of denoising steps by distilling a pretrained diffusion model to guide an identically architectured model with fewer sampling steps. Li et al. (2023); Hou & Asghar (2023) employ post-training quantization. Chen et al. (2023) enhance the implementation of SDMs for better compatibility with GPUs. However, the removal of architectural elements in diffusion models has not been investigated despite the established efficacy of structured pruning (Li et al., 2017; Liu et al., 2021).

This study unlocks the immense potential of classical architectural compression in attaining smaller and faster diffusion models. We eliminate multiple residual and attention blocks from the U-Net of a SDM and pretrain it with feature-level knowledge distillation (KD) (Romero et al., 2015; Heo et al., 2019) for general-purpose T2I. Despite being trained with only 0.22M LAION pairs (less than 0.1% of the entire training pairs) (Schuhmann & Beaumont, 2022) on a single A100 GPU, our compact models can mimic the original SDM by leveraging transferred knowledge. Our work effectively reduces the computation of SDM-v1 and attains competitive zero-shot results on par with multi-billion parameter models (Ramesh et al., 2021; Ding et al., 2021; 2022). Moreover,

[1]Nota Inc., South Korea. Correspondence to: Bo-Kyeong Kim <bokyeong1015@gmail.com>.

Work presented at the ES-FoMo Workshop at ICML 2023, Honolulu, Hawaii, USA.

Table 1. Minor impact of eliminating the mid-stage from the U-Net of SDM-v1.4 (Rombach & Esser, 2022) on zero-shot MS-COCO performance. Any retraining is not performed for the mid-stage removed model. For evaluation details, see Section 3.1.

Model	Performance		# Params	
	FID ↓	IS ↑	U-Net	Whole
SDM-v1.4	13.05	36.76	859.5M	1032.1M
No Mid-Stage	15.60	32.33	762.5M (-11.3%)	935.1M (-9.4%)

we present the application of our light pretrained models in customized T2I with DreamBooth (Ruiz et al., 2023).

2. Compression Method

Our primary focus is the U-Net (Ronneberger et al., 2015; Dhariwal & Nichol, 2021), the most compute-heavy component in SDMs. Conditioned on the text and time-step embeddings, the U-Net performs multiple denoising steps on latent representations. At each step, the U-Net produces the noise residual to compute the latent for the next step (see Figure 3). We reduce this per-step computation, leading to block-removed knowledge-distilled SDMs (BK-SDMs).

2.1. Compressed U-Net Architecture

We compress the 1.04B-parameter SDM-v1, and the proposed models are referred to as (see Appendix B for the detailed architectures):

- BK-SDM-Base (0.76B parameters) obtained with Section 2.1.1 (fewer blocks in outer stages).
- BK-SDM-Small (0.66B) with Sections 2.1.1 and 2.1.2 (mid-stage removal).
- BK-SDM-Tiny (0.50B) with Sections 2.1.1, 2.1.2, and 2.1.3 (further inner-stage removal).

2.1.1. Fewer Blocks in the Down and Up Stages

This approach is closely aligned with DistilBERT (Sanh et al., 2019) which halves the number of layers for improved compute efficiency and initializes the compact model with the original weights by benefiting from the shared dimensionality. In the original U-Net, each stage with a common spatial size consists of multiple blocks, and most stages contain pairs of residual (R) (He et al., 2016) and cross-attention (A) (Vaswani et al., 2017; Jaegle et al., 2021) blocks. We hypothesize the existence of some unnecessary pairs and use the following removal strategies, as shown in Figure 3.

For the down stages, we maintain the first R-A pairs while eliminating the second pairs, because the first pairs process the changed spatial information and would be more important than the second pairs. This design choice does not harm the dimensionality of the original U-Net, enabling the use of the corresponding pretrained weights for initialization.

Figure 2. Visual results of the mid-stage removed U-Net without retraining. See Figure 11 for an enlarged version.

For the up stages, while adhering to the aforementioned scheme, we retain the third R-A pairs. This allows us to utilize the output feature maps at the end of each down stage and the corresponding skip connections between the down and up stages. The same process is applied to the innermost down and up stages that contain only R blocks.

2.1.2. Removal of the Entire Mid-Stage

Surprisingly, removing the entire mid-stage from the original U-Net (marked with red in Figure 3) does not noticeably degrade the generation quality for many text prompts while effectively reducing the number of parameters (see Table 1 and Figure 2). This is consistent with the minor role of inner layers in the U-Net generator of GANs (Kim et al., 2022).

Integrating the mid-stage removal with fewer blocks in Section 2.1.1 further decreases computational burdens (Table 3) at the cost of a slight decline in performance (Table 2). Therefore, we offer this mid-stage elimination as an option, depending on the priority between compute efficiency (using BK-SDM-Small) and generation quality (BK-SDM-Base).

2.1.3. Further Removal of the Innermost Stages

For additional compression, the innermost down and up stages can also be pruned from BK-SDM-Small (see Appendix B), leading to our lightest model BK-SDM-Tiny. This implies that outer stages with larger spatial dimensions and their skip connections play a crucial role in the U-Net.

2.2. Distillation Pretraining for General-purpose T2I

We train the compact U-Net to mimic the behavior of the original U-Net. We use the pretrained-and-frozen encoders to obtain the inputs of the U-Net. Given the latent representation z of an image and its paired text embedding y, the task loss for the reverse denoising process (Ho et al., 2020; Rombach et al., 2022) is computed as:

$$\mathcal{L}_{\text{Task}} = \mathbb{E}_{z,\epsilon,y,t}\Big[||\epsilon - \epsilon_{\text{S}}(z_t, y, t)||_2^2\Big], \quad (1)$$

where $\epsilon \sim N(0, I)$ and $t \sim \text{Uniform}(1, T)$ denote the noise and time step sampled from the diffusion process, respec-

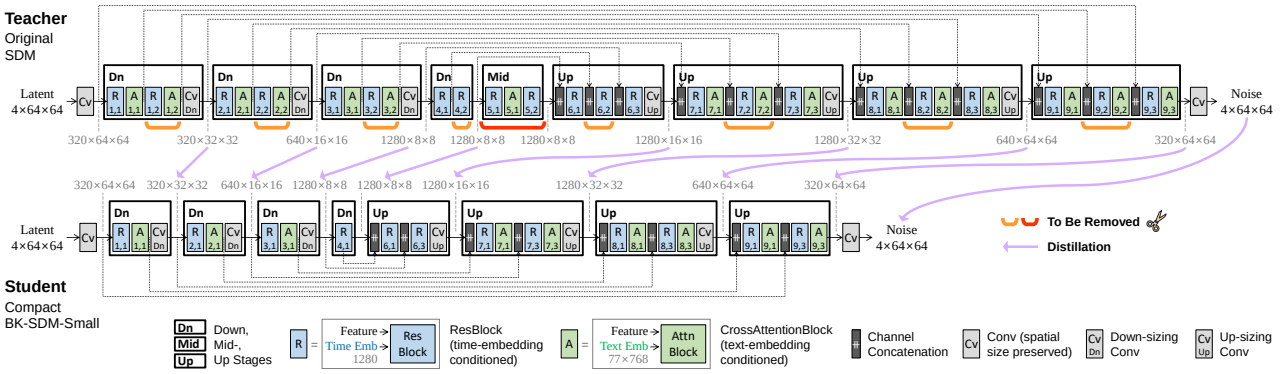

Figure 3. U-Net architectures and KD-based pretraining process. The compact U-Net student is built by eliminating several residual and attention blocks from the original U-Net teacher. Through the feature and output distillation from the teacher, the student can be trained effectively yet rapidly. See Appendix B for other architectures of BK-SDMs and Appendix C for the block details.

tively, and $\epsilon_{\mathrm{S}}(\circ)$ indicates the output of our compact U-Net student. For brevity, we omit the subscripts of $\mathbb{E}_{z,\epsilon,y,t}[\circ]$ in the following notations.

The compact student is also trained to imitate the outputs of the original U-Net teacher, $\epsilon_{\mathrm{T}}(\circ)$, with the following output-level KD objective (Hinton et al., 2014):

$$\mathcal{L}_{\mathrm{OutKD}} = \mathbb{E}\Big[||\epsilon_{\mathrm{T}}(z_t, y, t) - \epsilon_{\mathrm{S}}(z_t, y, t)||_2^2\Big]. \quad (2)$$

A key to our approach is the utilization of feature-level KD (Romero et al., 2015; Heo et al., 2019) that provides abundant guidance for the student's training:

$$\mathcal{L}_{\mathrm{FeatKD}} = \mathbb{E}\Big[\sum_l ||f_{\mathrm{T}}^l(z_t, y, t) - f_{\mathrm{S}}^l(z_t, y, t)||_2^2\Big], \quad (3)$$

where $f_{\mathrm{T}}^l(\circ)$ and $f_{\mathrm{S}}^l(\circ)$ represent the feature maps of the l-th layer in a predefined set of distilled layers from the teacher and the student, respectively.

The final objective is formalized as below. We simply set the loss weights λ_{OutKD} and $\lambda_{\mathrm{FeatKD}}$ as 1, which is effective in empirical validation without hyperparameter tuning.

$$\mathcal{L} = \mathcal{L}_{\mathrm{Task}} + \lambda_{\mathrm{OutKD}}\mathcal{L}_{\mathrm{OutKD}} + \lambda_{\mathrm{FeatKD}}\mathcal{L}_{\mathrm{FeatKD}}. \quad (4)$$

2.3. Application: Smaller Personalized SDMs

To emphasize the benefit of our lightweight pretrained SDMs, we use a popular finetuning scenario for personalized generation. DreamBooth (Ruiz et al., 2023) enables T2I diffusion models to create contents about a particular subject using just a few input images. Our compact models reduce finetuning cost and produce high-quality images based on the inherited capability of the original SDM.

Table 2. Zero-shot results on 30K prompts from MS-COCO validation set (Lin et al., 2014) at 256×256 resolution. For our models, the results with the minimum FID and the final 50K-th iteration are reported (see Appendix E.2 for detailed analysis).

Model	FID ↓	IS ↑	# Params	Data Size
SDM-v1.4 (Rombach et al., 2022)	13.05	36.76	1.04B	600M
Small Stable Diffusion (Pinkney, 2023)	12.76	32.33	0.76B	229M
BK-SDM-Base (Ours) @ Min FID	13.57	29.22	0.76B	0.22M
BK-SDM-Base (Ours) @ Final Iter	15.76	33.79	0.76B	0.22M
BK-SDM-Small (Ours) @ Min FID	15.93	29.61	0.66B	0.22M
BK-SDM-Small (Ours) @ Final Iter	16.98	31.68	0.66B	0.22M
BK-SDM-Tiny (Ours) @ Min FID	16.54	29.84	0.50B	0.22M
BK-SDM-Tiny (Ours) @ Final Iter	17.12	30.09	0.50B	0.22M
DALL·E†* (Ramesh et al., 2021)	27.5	17.9	12B	250M
CogView‡* (Ding et al., 2021)	27.1	18.2	4B	30M
CogView2†* (Ding et al., 2022)	24.0	22.4	6B	30M
Make-A-Scene‡ (Gafni et al., 2022)	11.84	-	4B	35M
LAFITE$^{\ddagger\sharp}$ (Zhou et al., 2022)	26.94	26.02	0.23B	3M
GALIP (CC3M)† (Tao et al., 2023)	16.12	-	0.32B	3M
GALIP (CC12M)† (Tao et al., 2023)	12.54	-	0.32B	12M
GLIDE‡ (Nichol et al., 2022)	12.24	-	5B	250M
LDM-KL-8-G$^{\ddagger\sharp}$ (Rombach et al., 2022)	12.63	30.29	1.45B	400M
DALL·E-2† (Ramesh et al., 2022)	10.39	-	5.2B	250M

† and ‡: FID from (Tao et al., 2023) and (Rombach et al., 2022), respectively. * and $^{\sharp}$: IS from (Ding et al., 2022) and (Rombach et al., 2022), respectively.

3. Experiments

This section presents key results of BK-SDMs. Appendix D provides detailed experimental setups, and Appendix E presents in-depth analyses and additional results.

3.1. General-purpose T2I Generation

Table 2 shows the zero-shot T2I results on 30K samples from the MS-COCO 256×256 validation set. Despite being trained with only 0.22M samples and having fewer than 1B parameters, our compressed models demonstrate competitive performance on par with previous large pretrained models. Despite the absence of a paper support, we include the model (Pinkney, 2023) that is identical in structure to BK-SDM-Base for comparison. This model benefits from

Table 3. The impact of per-step compute reduction of the U-Net on the entire SDM. The number of sampling steps is indicated with the parentheses, e.g., U-Net (1) for one step. The full computation (denoted by "Whole") covers the text encoder, U-Net, and image decoder. All corresponding values are obtained on the generation of a single 512×512 image with 25 denoising steps. The latency was measured on Xeon Silver 4210R CPU 2.40GHz and NVIDIA GeForce RTX 3090 GPU.

Model	# Params		MACs			CPU Latency			GPU Latency		
	U-Net	Whole	U-Net (1)	U-Net (25)	Whole	U-Net (1)	U-Net (25)	Whole	U-Net (1)	U-Net (25)	Whole
SDM-v1.4	860M	1033M	339G	8469G	9716G	5.63s	146.28s	153.00s	0.049s	1.28s	1.41s
BK-SDM-Base (Ours)	580M (-32.6%)	752M (-27.1%)	224G (-33.9%)	5594G (-33.9%)	6841G (-29.5%)	3.84s (-31.8%)	99.95s (-31.7%)	106.67s (-30.3%)	0.032s (-34.6%)	0.83s (-35.2%)	0.96s (-31.9%)
BK-SDM-Small (Ours)	483M (-43.9%)	655M (-36.5%)	218G (-35.7%)	5444G (-35.7%)	6690G (-31.1%)	3.45s (-38.7%)	89.78s (-38.6%)	96.50s (-36.9%)	0.030s (-38.7%)	0.77s (-39.8%)	0.90s (-36.1%)
BK-SDM-Tiny (Ours)	324M (-62.4%)	496M (-51.9%)	206G (-39.5%)	5126G (-39.5%)	6373G (-34.4%)	3.03s (-46.2%)	78.77s (-46.1%)	85.49s (-44.1%)	0.026s (-46.9%)	0.67s (-47.7%)	0.80s (-43.2%)

Table 4. Personalized generation with finetuning over different pretrained models. Our compact models can preserve subject fidelity (CLIP-I) and prompt fidelity (CLIP-T) of the original SDM with reduced finetuning (FT) cost and fewer parameters. See Table 6 for additional results with DINO scores.

Pretrained Model	CLIP-I ↑	CLIP-T ↑	FT Time†	FT Mem‡	# Params
SDM v1.4	0.725	0.263	881.3s	23.0GB	1.04B
BK-SDM-Base (Ours)	0.717	0.260	622.3s	18.7GB	0.76B
BK-SDM-Small (Ours)	0.705	0.259	603.6s	17.2GB	0.66B
BK-SDM-Tiny (Ours)	0.693	0.261	559.3s	13.1GB	0.50B
Ours-Base, Batch Size 64	0.708	0.262	622.3s	18.7GB	0.76B
- No KD & Random Init.	0.465	0.191	622.3s	18.7GB	0.76B
- No KD & Pretrained Init.	0.669	0.258	622.3s	18.7GB	0.76B

Per-subject FT time† and GPU memory‡ for 800 iterations with a batch size of 1 on NVIDIA RTX 3090.

Figure 4. Visual comparison with Ding et al. (2022); Zhou et al. (2022); Tao et al. (2023) on zero-shot MS-COCO benchmark. See Figure 12 for an enlarged version with additional results.

far more training resources, i.e., two-stage KD relying on two teachers (SDM-v1.4 and v1.5) and a much larger volume of data with significantly longer iterations.

Figures 4 and 12 depict synthesized images of different models with some MS-COCO captions. Our compressed models inherit the superior ability of SDM and produce more photorealistic images compared to the AR-based (Ding et al., 2022) and GAN-based (Zhou et al., 2022; Tao et al., 2023) baselines. Noticeably, the same latent code results in a shared visual style between the original and our compact SDMs (4th–7th columns in Figure 4), similar to the observation in transfer learning for GANs (Mo et al., 2020).

Table 3 shows how the compute reduction for each sampling step of the U-Net affects the overall compute of the SDM. The per-step reduction effectively decreases MACs and inference time by more than 30%. Notably, BK-SDM-Tiny has 50% fewer parameters compared to the original SDM.

Figure 5. Visual results of personalized generation. Each subject is marked as "a [identifier] [class noun]" (e.g., "a [V] dog"). See Figure 13 for an enlarged version with additional results.

3.2. Personalized T2I Generation with DreamBooth

Tables 4 and 6 compare the results of DreamBooth finetuning (Ruiz et al., 2023) with different pretrained models. BK-SDMs can preserve 95%∼99% performance of the original SDM with the reduced finetuning cost and number of parameters. Figures 5 and 13 depict that our models can accurately capture the subject details and generate various scenes. Over the models pretrained with a batch size of 64, we observe the impact of KD pretraining on personalized synthesis. The baselines without KD fail to generate the subjects entirely or cannot maintain the identity details.

4. Conclusion

We uncover the potential of architectural compression for general-purpose text-to-image synthesis with a renowned model, Stable Diffusion. Our block-removed lightweight models are effective for zero-shot generation, achieving competitive results against large-scale baselines. Distillation is a key of our method, leading to powerful pretraining even under very constrained resources. Our work is orthogonal to previous works for efficient diffusion models, e.g., enabling fewer sampling steps, and can be readily combined with them. We hope our study can facilitate future research on structural compression of large diffusion models.

Acknowledgements

We express our gratitude to the Microsoft Startups Founders Hub program for their invaluable contribution of Azure credits, fueling the research and development efforts of the NetsPresso R&D project.

References

Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023.

Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Chen, Y.-H., Sarokin, R., Lee, J., Tang, J., Chang, C.-L., Kulik, A., and Grundmann, M. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *CVPR Workshop*, 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021.

Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021.

Ding, M., Zheng, W., Hong, W., and Tang, J. Cogview2: Faster and better text-to-image generation via hierarchical transformers. In *NeurIPS*, 2022.

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022.

Hao, Z., Guo, J., Jia, D., Han, K., Tang, Y., Zhang, C., Hu, H., and Wang, Y. Learning efficient vision transformers via fine-grained manifold distillation. In *NeurIPS*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., and Choi, J. Y. A comprehensive overhaul of feature distillation. In *ICCV*, 2019.

Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. In *NeurIPS Workshop*, 2014.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS Workshop*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Hou, J. and Asghar, Z. World's first on-device demonstration of stable diffusion on an android phone. https://www.qualcomm.com/news, 2023.

Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., and Carreira, J. Perceiver: General perception with iterative attention. In *ICML*, 2021.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. Tinybert: Distilling bert for natural language understanding. In *Findings of EMNLP*, 2020.

Jin, Q., Ren, J., Woodford, O. J., Wang, J., Yuan, G., Wang, Y., and Tulyakov, S. Teachers do more than teach: Compressing image-to-image models. In *CVPR*, 2021.

Kim, B.-K., Choi, S., and Park, H. Cut inner layers: A structured pruning strategy for efficient u-net gans. In *ICML Workshop*, 2022.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. In *ICLR*, 2017.

Li, M., Lin, J., Ding, Y., Liu, Z., Zhu, J.-Y., and Han, S. Gan compression: Efficient architectures for interactive conditional gans. In *CVPR*, 2020.

Li, X., Lian, L., Liu, Y., Yang, H., Dong, Z., Kang, D., Zhang, S., and Keutzer, K. Q-diffusion: Quantizing diffusion models. *arXiv preprint arXiv:2302.04304*, 2023.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. In *ICLR*, 2022.

Liu, Y., Shu, Z., Li, Y., Lin, Z., Perazzi, F., and Kung, S.-Y. Content-aware gan compression. In *CVPR*, 2021.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *NeurIPS*, 2022a.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Mangrulkar, S., Gugger, S., Debut, L., Belkada, Y., and Paul, S. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. In *NeurIPS Workshop*, 2022.

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. In *CVPR*, 2023.

Mo, S., Cho, M., and Shin, J. Freeze the discriminator: a simple baseline for fine-tuning gans. In *CVPR Workshop*, 2020.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *CVPR*, 2019.

Pinkney, J. Small stable diffusion. https://huggingface.co/OFA-Sys/small-stable-diffusion-v0, 2023.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *ICML*, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Ren, Y., Wu, J., Xiao, X., and Yang, J. Online multi-granularity distillation for gan compression. In *ICCV*, 2021.

Rombach, R. and Esser, P. Stable diffusion v1-4. https://huggingface.co/CompVis/stable-diffusion-v1-4, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, 2023.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022.

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *NeurIPS*, 2016.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS Workshop*, 2019.

Schuhmann, C. and Beaumont, R. Laion-aesthetics. https://laion.ai/blog/laion-aesthetics, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS Workshop*, 2022.

Shen, H., Cheng, P., Ye, X., Cheng, W., and Abidi, H. Accelerate stable diffusion with intel neural compressor. https://medium.com/intel-analytics-software, 2022.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *ICLR*, 2021.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., and Zhou, D. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *ACL*, 2020.

Tao, M., Bao, B.-K., Tang, H., and Xu, C. Galip: Generative adversarial clips for text-to-image synthesis. In *CVPR*, 2023.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers and distillation through attention. In *ICML*, 2021.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. In *NeurIPS*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Wang, H., Du, X., Li, J., Yeh, R. A., and Shakhnarovich, G. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2023.

Zagoruyko, S. and Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Zhang, L., Chen, X., Tu, X., Wan, P., Xu, N., and Ma, K. Wavelet knowledge distillation: Towards efficient image-to-image translation. In *CVPR*, 2022.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. In *ICLR*, 2023.

Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J., and Sun, T. Towards language-free training for text-to-image generation. In *CVPR*, 2022.

Zhu, L. Thop: Pytorch-opcounter. https://github.com/Lyken17/pytorch-OpCounter, 2018.

Appendix

A. Related Work

Large T2I diffusion models. By gradually removing noise from corrupted data, diffusion-based generative models (Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021) enable high-fidelity synthesis with broad mode coverage. Integrating these merits with the advancement of pretrained language models (Radford et al., 2021; 2019; Devlin et al., 2019) has significantly improved the quality of T2I synthesis. In GLIDE (Nichol et al., 2022) and Imagen (Saharia et al., 2022), a text-conditional diffusion model generates a 64×64 image, which is upsampled via super-resolution modules. In DALL·E-2 (Ramesh et al., 2022), a text-conditional prior network produces an image embedding, which is transformed into a 64×64 image via a diffusion decoder and further upscaled into higher resolutions. SDMs (Rombach & Esser, 2022; Rombach et al., 2022) perform the diffusion modeling in a 64×64 latent space constructed through a pixel-space autoencoder. We use a SDM as our baseline because of its open-access and gaining popularity over numerous downstream tasks (Brooks et al., 2023; Wang et al., 2023; Blattmann et al., 2023; Ruiz et al., 2023).

Efficient diffusion models. Several studies have addressed the slow sampling process of diffusion models. Diffusion-tailored distillation approaches (Meng et al., 2023; 2022; Salimans & Ho, 2022) progressively transfer knowledge from a pretrained diffusion model to a fewer-step model with the same architecture. Fast high-order solvers (Lu et al., 2022a;b; Zhang & Chen, 2023) for diffusion ordinary differential equations boost the sampling speed. Orthogonal to these directions for less sampling steps, our network compression approach reduces per-step computation and can be easily integrated with them. Leveraging quantization techniques (Li et al., 2023; Hou & Asghar, 2023; Shen et al., 2022) and implementation optimizations (Chen et al., 2023) has been applied for SDMs and also can be combined with our models for further efficiency gains.

Distillation-based compression. KD enhances the performance of small-size models by exploiting output-level (Hinton et al., 2014; Park et al., 2019) and feature-level (Romero et al., 2015; Heo et al., 2019; Zagoruyko & Komodakis, 2017) information of large source models. Although this classical distillation has been actively used toward efficient GANs (Li et al., 2020; Ren et al., 2021; Liu et al., 2021; Jin et al., 2021; Zhang et al., 2022), its power has not been explored for structurally compressed diffusion models. Distillation-based pretraining enables small yet capable general-purpose language models (Sanh et al., 2019; Sun et al., 2020; Jiao et al., 2020) and vision transformers (Touvron et al., 2021; Hao et al., 2022). Beyond such models, we show that its success can be extended to diffusion models with iterative sampling steps. Concurrently with our study, a recently released small SDM without paper evidence (Pinkney, 2023) similarly utilizes KD pretraining for a block-eliminated architecture, but it relies on significantly more training resources along with multi-stage distillation. In contrast, our lighter models achieve further reduced computation, and we show that competitive results can be obtained even with much less data and single-stage distillation.

B. U-Net Architectures of BK-SDMs

Figure 6 depicts the U-Net architectures. Compared to the 1.04B-parameter original SDM (with 0.86B-parameter U-Net), our models are smaller and lighter: 0.76B-parameter BK-SDM-Base (with 0.58B-parameter U-Net), 0.66B BK-SDM-Small (0.49B U-Net), and 0.50B BK-SDM-Tiny (0.33B U-Net). Section C describes the details of block components.

C. Details of Block Components in SDM's U-Net

Figure 7 shows the details of architectural blocks (depicted in Figures 3 and 6). Each residual block (ResBlock) contains two 3-by-3 convolutional layers and is conditioned on the time-step embedding. Each attention block (AttnBlock) contains a self-attention module, a cross-attention module, and a feed-forward network. The text embedding is merged via the cross-attention module. Within the attention block, the feature spatial dimensions H and W are flattened into a sequence length of HW. The number of channels C is considered as an embedding size, processed with 8 attention heads. The number of groups for the group normalization is set to 32. Except the down-sizing part, all the convolutional layers maintain the spatial dimensions by adjusting the stride and padding.

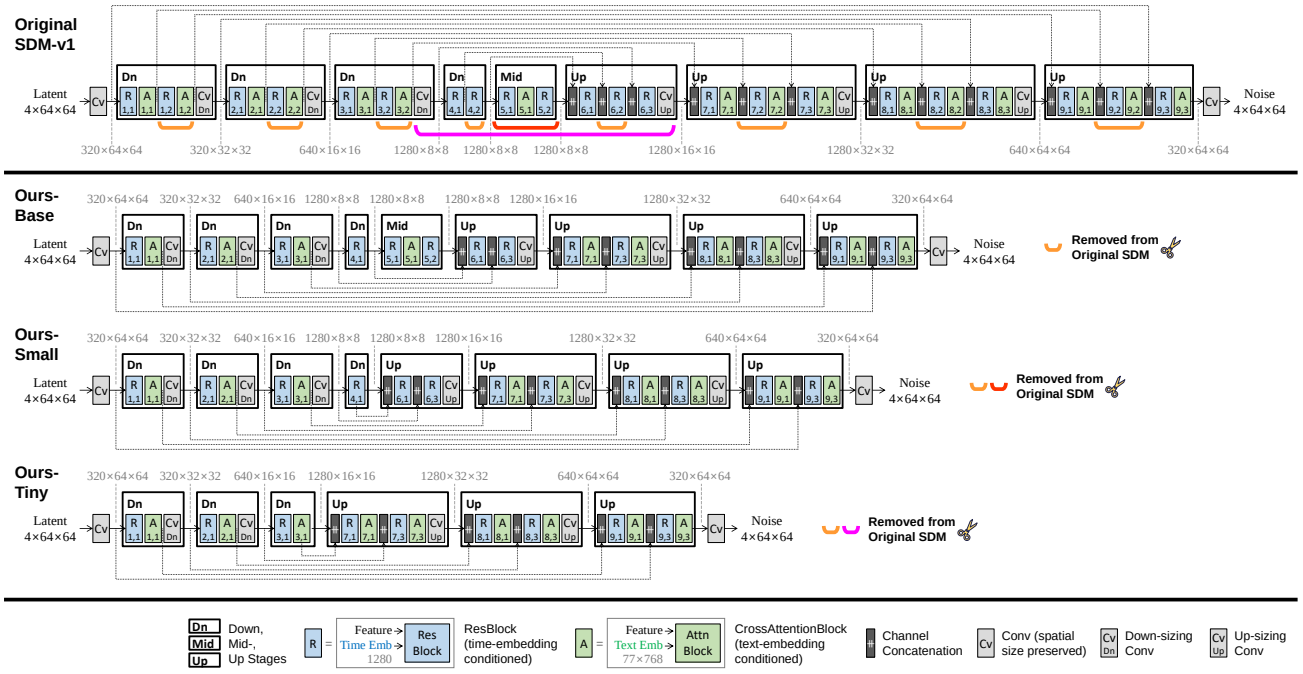

Figure 6. U-Net architectures of the original SDM-v1 and our BK-SDMs.

D. Experimental Setup

D.1. Datasets and Evaluation Metrics

Pretraining. We train our compact SDMs with only 0.22M image-text pairs from LAION-Aesthetics V2 6.5+ (Schuhmann & Beaumont, 2022; Schuhmann et al., 2022), which are significantly fewer than the original training data used for SDM-v1.4 (Rombach & Esser, 2022) (i.e., 600M pairs of LAION-Aesthetics V2 5+ for the resumed training).

Zero-shot T2I evaluation. Following the popular protocol (Ramesh et al., 2021; Rombach et al., 2022; Saharia et al., 2022) to assess general-purpose T2I with pretrained models, we use 30K prompts from the MS-COCO validation split (Lin et al., 2014) and compare the generated images to the whole validation set. We compute Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) to assess visual quality. Moreover, we measure CLIP score (Radford et al., 2021; Hessel et al., 2021) with CLIP-ViT-g/14 model to assess text-image correspondence.

Finetuning for personalized generation. We use the DreamBooth dataset (Ruiz et al., 2023) that covers 30 subjects, each of which is associated with 25 prompts and 4∼6 images. Through individual finetuning for each subject, 30 personalized models are obtained. For evaluation, we follow the protocol of Ruiz et al. (2023) based on four synthesized images per subject and per prompt. We consider CLIP-I and DINO scores to measure how well subject details are maintained in generated images (i.e., subject fidelity) and CLIP-T scores to measure text-image alignment (i.e., text fidelity). We use ViT-S/16 embeddings (Caron et al., 2021) for DINO scores and CLIP-ViT-g/14 embeddings for CLIP-I and CLIP-T.

D.2. Implementation

We use the released version v1.4 of SDM (Rombach & Esser, 2022) as our compression target. We remark that our approach is also applicable to other versions in v1.1–v1.5 with the same architecture and to SDM-v2 with a similarly designed architecture. We adjust the codes in Diffusers library (von Platen et al., 2022) for pretraining our models and those in PEFT library (Mangrulkar et al., 2022) for DreamBooth-finetuning, both of which adopt the training process of DDPM (Ho et al., 2020) in latent spaces.

Distillation-based pretraining. For augmentation, smaller edge of each image is resized to 512, and a center crop of size 512 is applied with random flip. We use a single NVIDIA A100 80G GPU for 50K-iteration pretraining with the AdamW optimizer and a constant learning rate of 5e-5. With the gradient accumulation steps of 4, the total batch size is set to either

64 or 256. With a batch size of 64 for training BK-SDM-Base, it takes about 60 hours for 50K iterations and 28GB GPU memory. With a batch size of 256, it takes about 300 hours and 53GB GPU memory. Training smaller architectures results in 5∼10% decrease in GPU memory usage.

DreamBooth finetuning. For augmentation, smaller edge of each image is resized to 512, and a random crop of size 512 is applied. We use a single NVIDIA GeForce RTX 3090 GPU to finetune each personalized model for 800 iterations with the AdamW optimizer and a constant learning rate of 1e-6. We jointly finetune the text encoder as well as the U-Net. For each subject, 200 class images are generated by the original SDM. The weight of prior preservation loss is set to 1. With a batch size of 1, the original SDM requires 23GB GPU memory for finetuning, whereas BK-SDMs require 13∼19GB memory.

Inference. Following the default inference setup, we use PNDM scheduler (Liu et al., 2022) for zero-shot T2I generation and DPM-Solver (Lu et al., 2022a;b) for DreamBooth results. For compute efficiency, we always opt for 25 denoising steps of the U-Net at the inference phase. The classifier-free guidance scale (Ho & Salimans, 2021; Saharia et al., 2022) is set to the default value of 7.5, except the analysis in Figure 9.

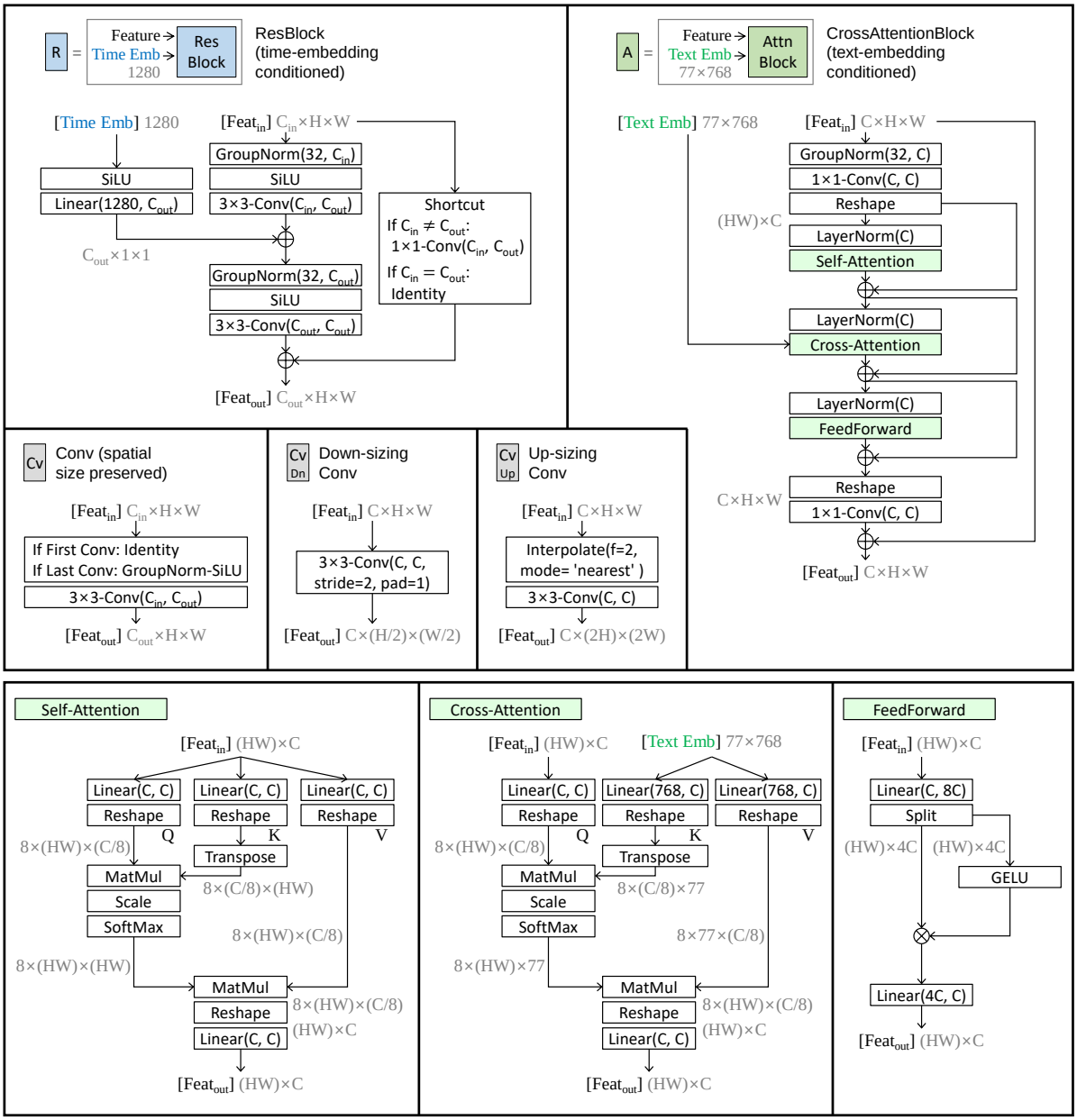

Figure 7. Block components in the U-Net of SDMs.

E. In-depth Analyses and Additional Results

E.1. Ablation Study

Table 5 presents the ablation study with the zero-shot MS-COCO benchmark dataset. The common default settings for the models N1–N9 involve the usage of fewer blocks in the down and up stages (Section 2.1.1) and the denoising task loss (Eq. 1). All the models are drawn at the 50K-th training iteration. We made the following observations.

N1 *vs.* N2. Importing the pretrained weights for initialization clearly improves the performance of block-removed SDMs. Transferring knowledge from well-trained models, a popularized practice in machine learning, is also beneficial for T2I generation with SDMs.

N2 *vs.* N3 *vs.* N4. Exploiting output-level KD (Eq. 2) effectively boosts the generation quality compared to using only the denoising task loss. Leveraging feature-level KD (Eq. 3) further improves the performance by offering sufficient guidance over multiple stages in the student.

N4 *vs.* N5. An increased batch size leads to a better IS and CLIP score but with a minor drop in FID. We opt for a batch size of 256 based on the premise that more samples per batch would enhance the model's understanding ability.

N6, N7, N8, and N9. Despite slight performance drop, the models N6 and N7 with the mid-stage removal have fewer parameters (0.66B) than N4 and N5 (0.76B), offering improved compute efficiency. The further removal of two innermost stages leads to the lightest models N8 and N9 (0.50B).

Table 5. Ablation study on zero-shot MS-COCO 256×256 30K. The common settings include fewer blocks in the down and up stages and the denoising task loss. N5, N7, and N9 correspond to BK-SDM-Base, BK-SDM-Small, and BK-SDM-Tiny, respectively

Model						Performance		
No.	Weight Initialization	Output KD	Feature KD	Batch Size	# Removed Inner Stages	FID ↓	IS ↑	CLIP Score ↑
N1	Random	✗	✗	64	✗	43.80	13.61	0.1622
N2	Pretrained	✗	✗	64	✗	20.45	22.68	0.2444
N3	Pretrained	✓	✗	64	✗	16.48	27.30	0.2620
N4	Pretrained	✓	✓	64	✗	14.61	31.44	0.2826
N5	Pretrained	✓	✓	256	✗	15.76	33.79	0.2878
N6	Pretrained	✓	✓	64	1	16.87	29.51	0.2644
N7	Pretrained	✓	✓	256	1	16.98	31.68	0.2677
N8	Pretrained	✓	✓	64	3	17.28	28.33	0.2607
N9	Pretrained	✓	✓	256	3	17.12	30.09	0.2653
Original SDM-v1.4 (Rombach & Esser, 2022; Rombach et al., 2022)						13.05	36.76	0.2958

E.2. Impact of Distillation on Pretraining Phase

We further analyze the merits of transferred knowledge via distillation, with the models from the pretrained weight initialization. Figure 8 shows zero-shot T2I performance over training iterations. Without KD, training compact models solely with the denoising task loss causes fluctuations or sudden drops in performance (indicated with green and cyan). Compared to the absence of KD, distillation (purple and pink) stabilizes and accelerates the training process, improving generation scores. This clearly demonstrates the benefits of providing sufficient hints for training guidance.

Notably, our small-size and tiny-size models trained with KD (yellow and red) outperform the bigger base-size models without KD (green and cyan). Additionally, while the best FID score is observed early on for our models, IS and CLIP score exhibit ongoing improvement, implying that judging models solely with FID may be suboptimal.

Figure 9 shows the trade-off curves from different classifier-free guidance scales (Ho & Salimans, 2021; Saharia et al., 2022) $\{2.0, 2.5, 3.0, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5\}$. For the analysis, we use 5K samples from the MS-COCO validation set and our base-size models from the 50K-th iteration. Higher guidance scales lead to better text-aligned images at the cost of less diversity. Compared to the baseline trained only with the denoising task loss, distillation-based pretraining achieves much better trade-off curves.

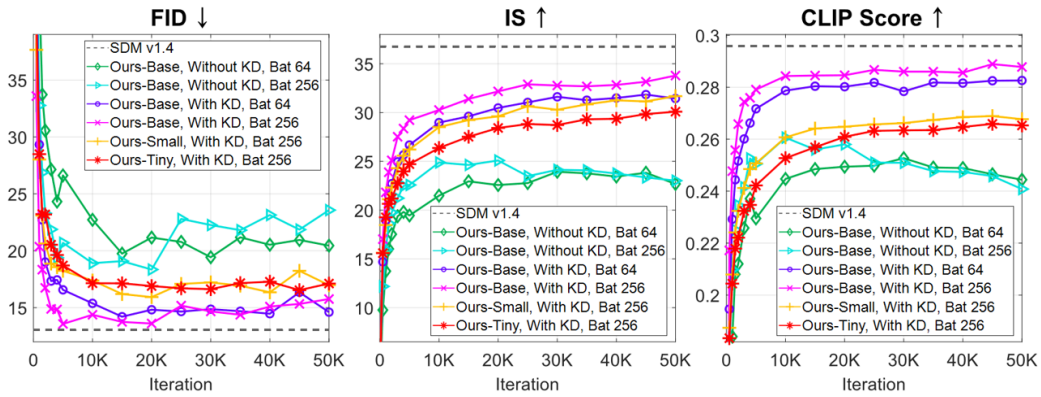

Figure 8. Results on zero-shot MS-COCO 256×256 30K over training progress. The architecture size, usage of KD, and batch size are denoted for our models.

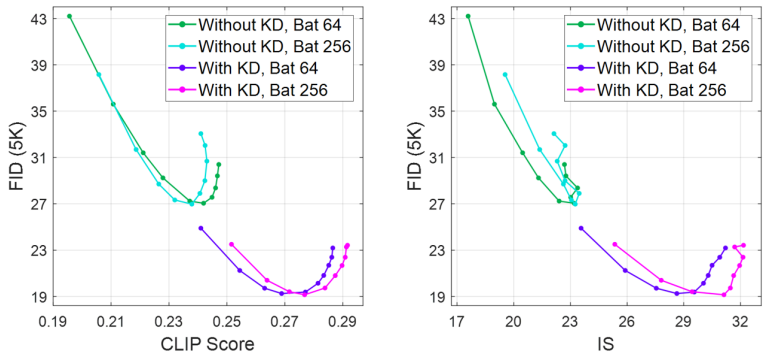

Figure 9. Effect of different classifier-free guidance scales on MS-COCO 512×512 5K.

E.3. Additional DreamBooth Results with DINO Scores

Table 6 shows the results of DreamBooth finetuning (Ruiz et al., 2023) with different pretrained models. BK-SDMs can preserve 95%∼99% performance of the original SDM with the reduced finetuning cost and number of parameters.

Table 6. Personalized generation with finetuning over different pretrained models. Our compact models can preserve subject fidelity (DINO and CLIP-I) and prompt fidelity (CLIP-T) of the original SDM with reduced finetuning (FT) cost and fewer parameters.

Pretrained Model	DINO ↑	CLIP-I ↑	CLIP-T ↑	FT Time†	FT Mem‡	# Params
SDM v1.4 (Rombach & Esser, 2022)	0.728	0.725	0.263	881.3s	23.0GB	1.04B
BK-SDM-Base (Ours)	0.723	0.717	0.260	622.3s	18.7GB	0.76B
BK-SDM-Small (Ours)	0.720	0.705	0.259	603.6s	17.2GB	0.66B
BK-SDM-Tiny (Ours)	0.715	0.693	0.261	559.3s	13.1GB	0.50B
BK-SDM-Base, Batch Size 64	0.718	0.708	0.262	622.3s	18.7GB	0.76B
- Without KD & Random Init.	0.594	0.465	0.191	622.3s	18.7GB	0.76B
- Without KD & Pretrained Init.	0.716	0.669	0.258	622.3s	18.7GB	0.76B

Per-subject finetuning time† and GPU memory‡ for 800 iterations with a batch size of 1 on NVIDIA RTX 3090.

E.4. More Visual Examples

Figure 10 summarizes the capability of BK-SDMs. Figure 11 depicts visual results of the mid-stage removed U-Net without retraining. Figure 12 shows additional visual comparison on the zero-shot MS-COCO benchmark. Figure 13 shows personalized generation results with DreamBooth.

Figure 10. Our compressed Stable Diffusion enables efficient (a) zero-shot general-purpose text-to-image generation and (b) personalized synthesis. Selected samples from BK-SDM-Small with 36% reduced parameters and latency are shown.

Figure 11. Visual results of removing the mid-stage from the U-Net of SDM-v1.4 (Rombach & Esser, 2022). Without retraining, the mid-stage removal can preserve generation quality for many text prompts.

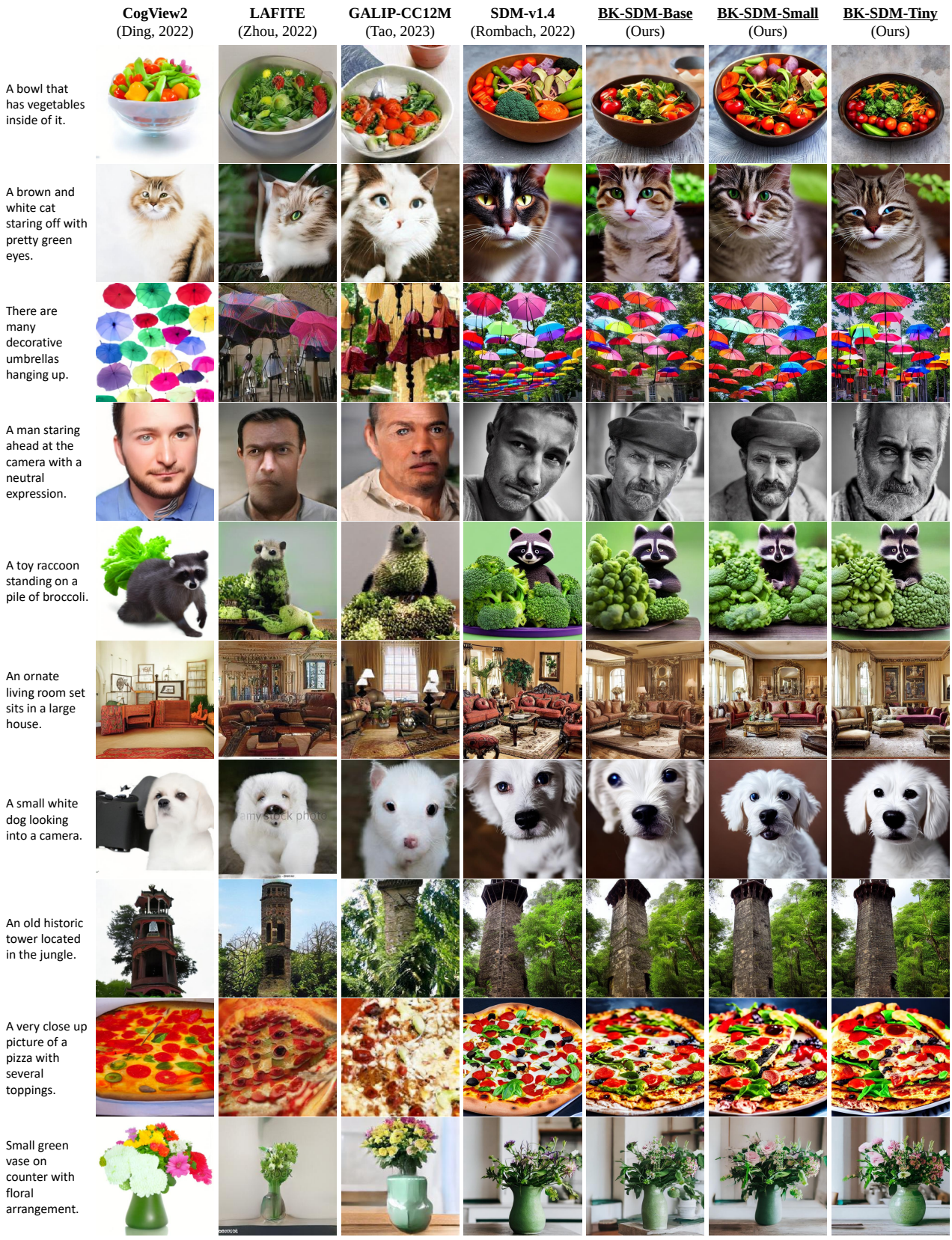

Figure 12. Additional zero-shot T2I results. The results of previous studies (Ding et al., 2022; Zhou et al., 2022; Tao et al., 2023) were obtained with their official codes and released models. We do not apply any CLIP-based reranking for SDM and our models.

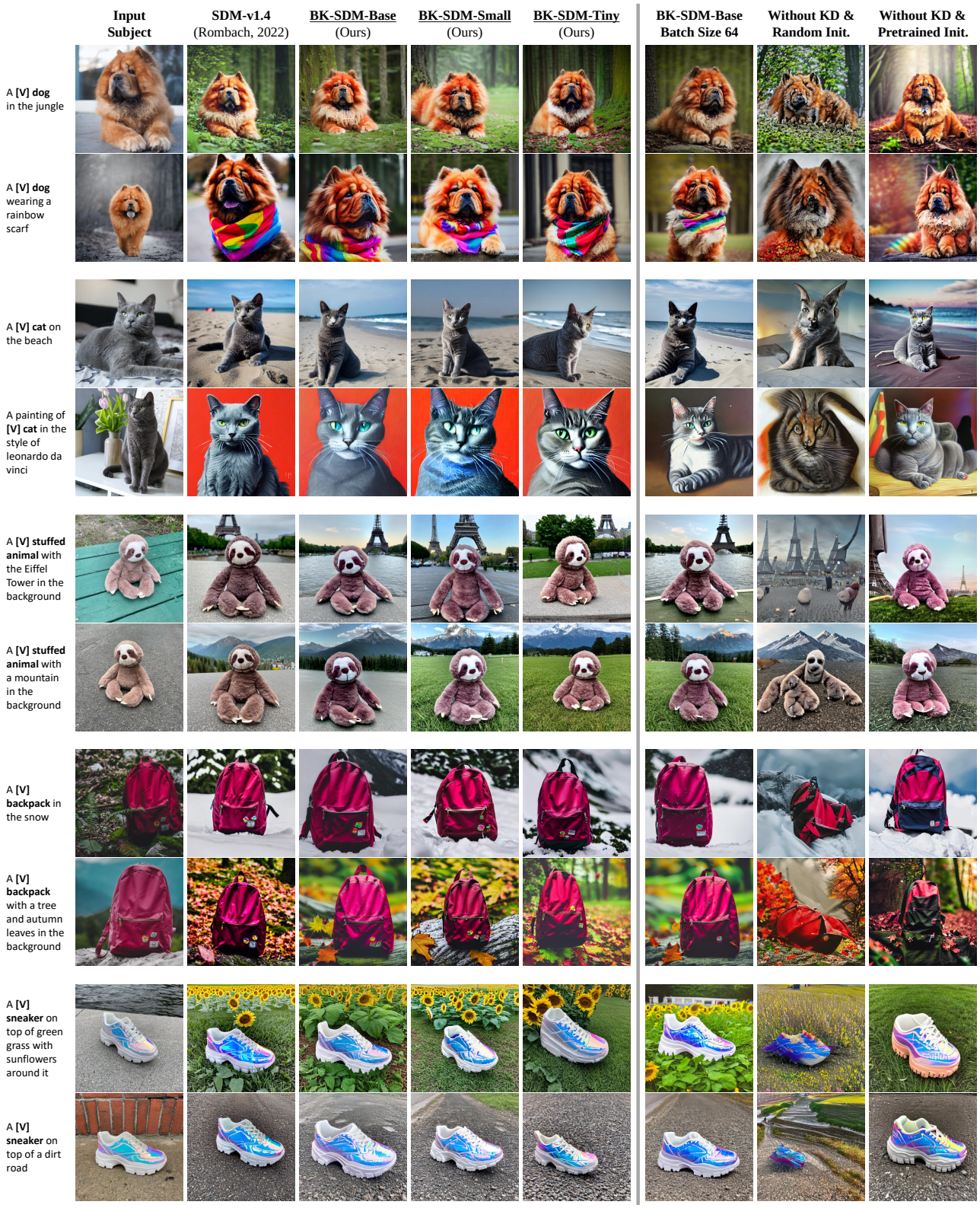

Figure 13. Additional results of personalized generation. Each subject is marked as "a [identifier] [class noun]" (e.g., "a [V] dog"). Similar to the original SDM, our compact models can synthesize the images of input subjects in different backgrounds while preserving their appearance.