Appendix of BK-SDM

A U-Net Architecture and Distillation Retraining

Figs. 15 and 16 depict the U-Net architectures and distillation process, respectively. Our approach is directly applicable to all the SDM versions in v1 and v2 (i.e., v1.1/2/3/4/5, v2.0/1, and v2.0/1-base), which share the same U-Net block configuration. See Fig. 17 for the block details.

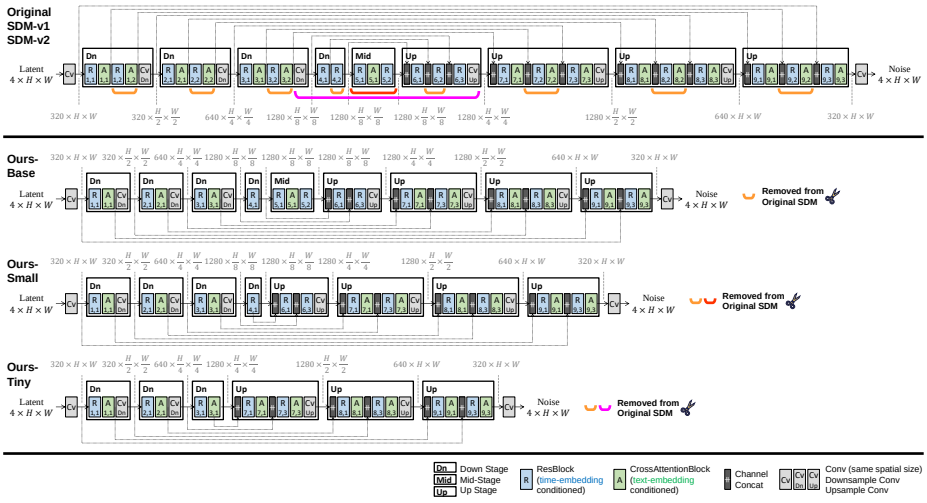

Fig. 15: U-Net architectures of SDM-v1, SDM-v2, and BK-SDMs.

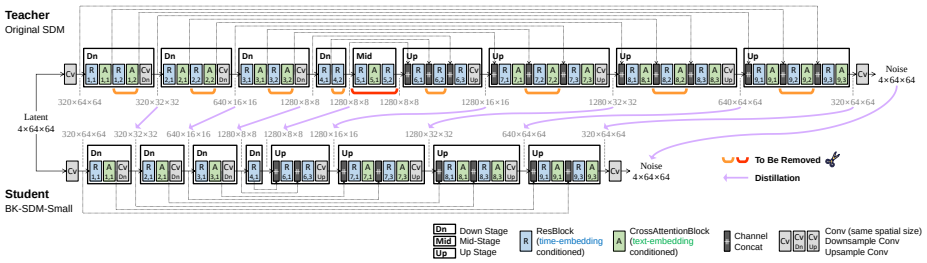

Fig. 16: Distillation retraining process. The compact U-Net student is built by eliminating several residual and attention blocks from the original U-Net teacher. Through the feature and output distillation from the teacher, the student can be trained effectively yet rapidly. The default latent resolution for SDM-v1 and v2-base is $H = W = 64$ in Fig. 15, resulting in 512×512 generated images.

Fig. 17 shows the details of architectural blocks. 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 attention heads. The number of groups for the group normalization is set to 32. The differences between SDM-v1 and SDM-v2 include the number of attention heads (8 for all the stages of SDM-v1 and [5, 10, 20, 20] for different stages of SDM-v2) and the text embedding dimensions (77×768 for SDM-v1 and 77×1024 for SDM-v2).

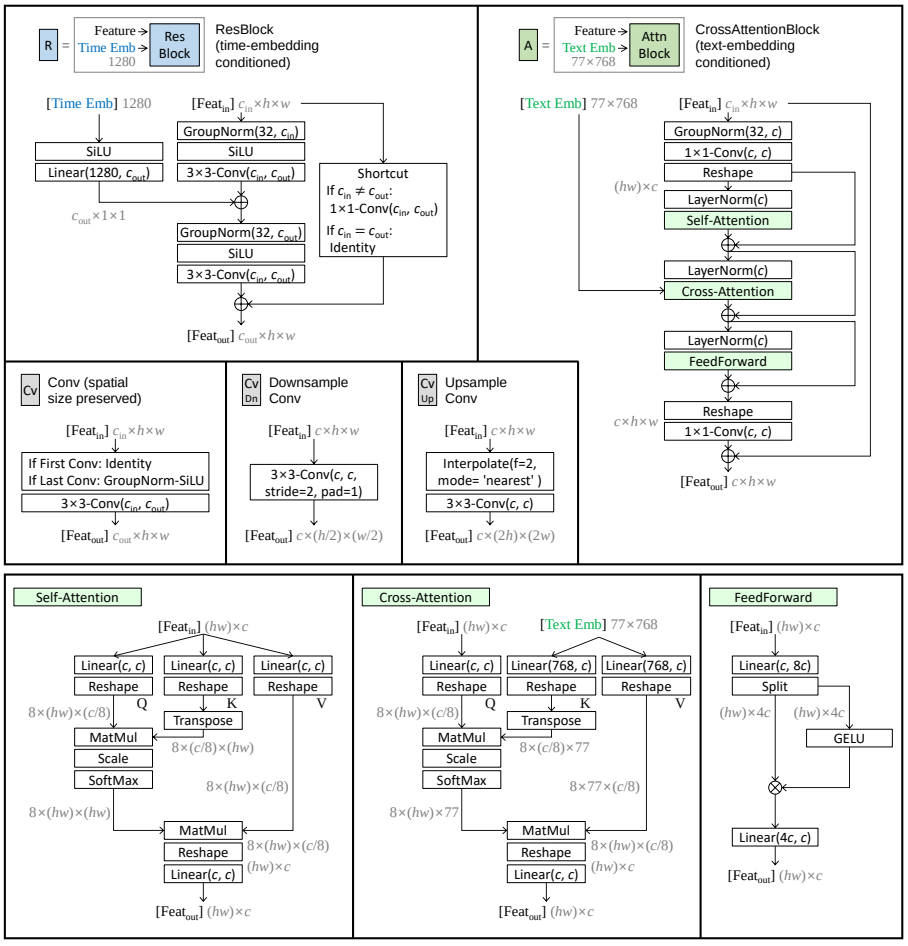

Fig. 17: Block components in the U-Net.

B Impact of Mid-stage Removal

Removing the entire mid-stage from the original U-Net does not noticeably degrade the generation quality for many text prompts while effectively reducing the number of parameters. See Fig. 18 and Tab. 10. Retraining is not performed.

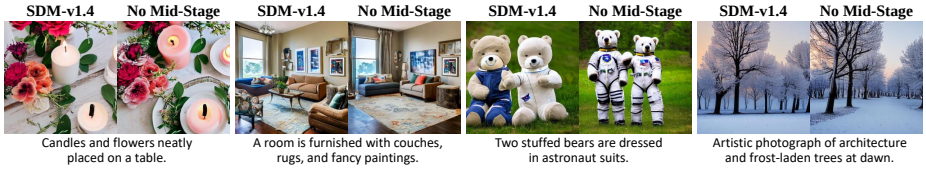

Fig. 18: Visual results of the mid-stage removed U-Net from SDM-v1.4 [64].

Table 10: Minor impact of eliminating the mid-stage on MS-COCO 256×256 30K.

Model	Performance		# Parameters	
	FID ↓	IS ↑	U-Net	Whole
SDM-v1.4 [64]	13.05	36.76	859.5M	1032.1M
Mid-Stage Removal	15.60	32.33	762.5M (-11.3%)	935.1M (-9.4%)

C Block-level Pruning Sensitivity Analysis

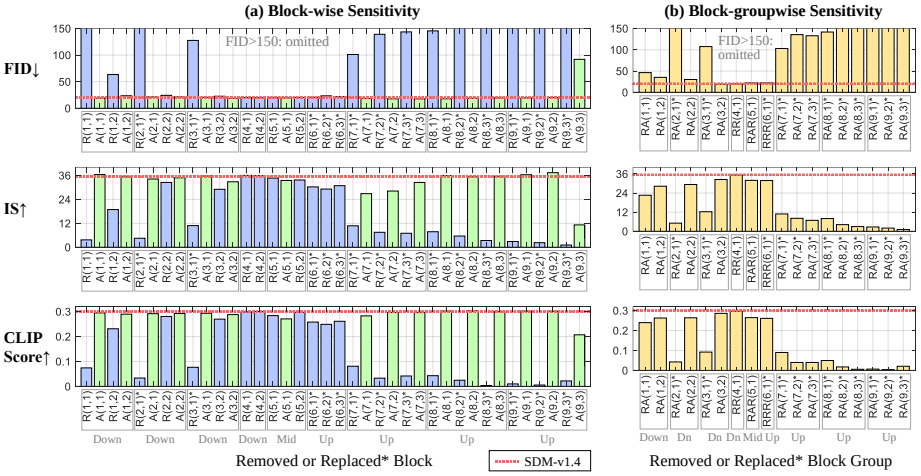

Fig. 19: Analyzing the importance of (a) each block and (b) each group of paired/triplet blocks in SDM-v1.4. Evaluation on MS-COCO 512×512 5K. The block notations match Fig. 15. Whenever possible (i.e., with the same dimensions of input and output), we remove each block to examine its effect on generation performance. For blocks with different channel dimensions of input and output, we replace them with channel interpolation modules (denoted by "*") to mimic the removal while retaining the information. The results are aligned with our architectural choices (e.g., removal of innermost stages and the second R-A pairs in down stages).

D Comparison with Existing Studies

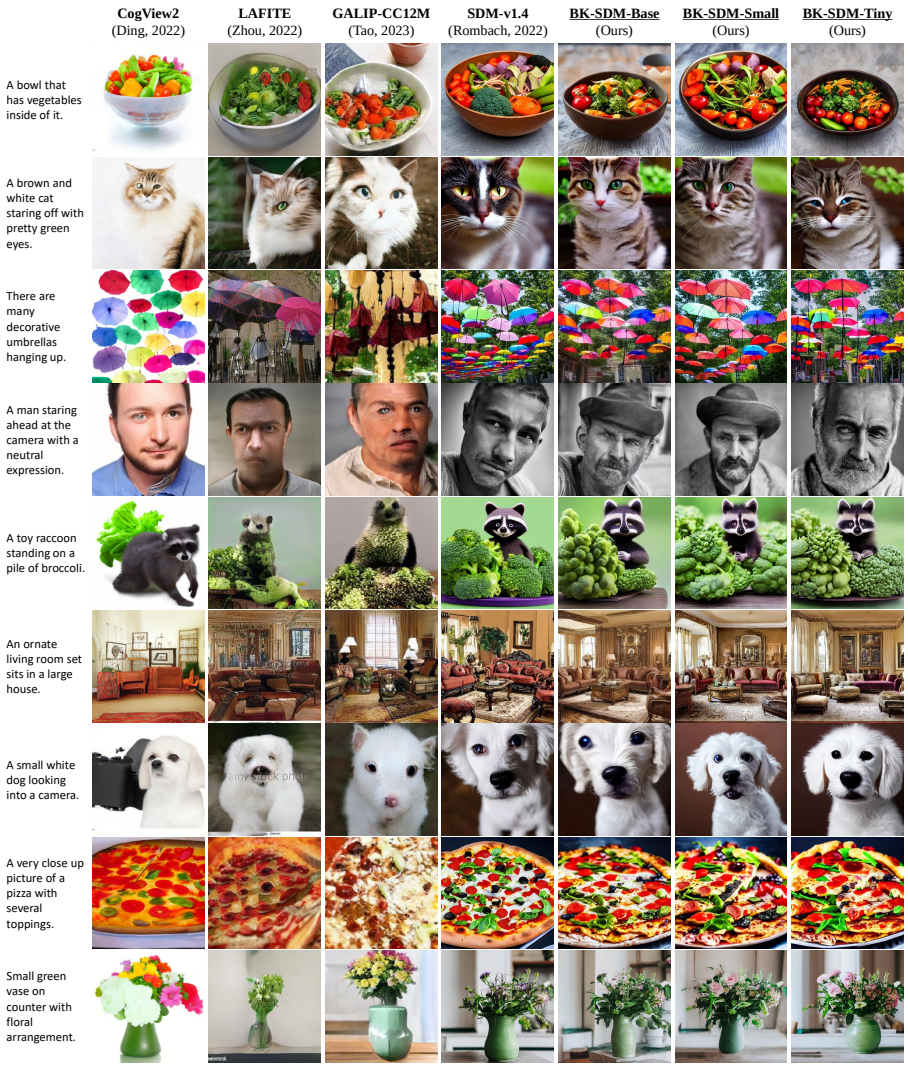

Fig. 20: Zero-shot general-purpose T2I results. The results of previous studies [12,83,94] were obtained with their official codes and released models. We do not apply any CLIP-based reranking for SDM and our models.

E Personalized Generation

Fig. 21: 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.

F Text-guided Image-to-Image Translation

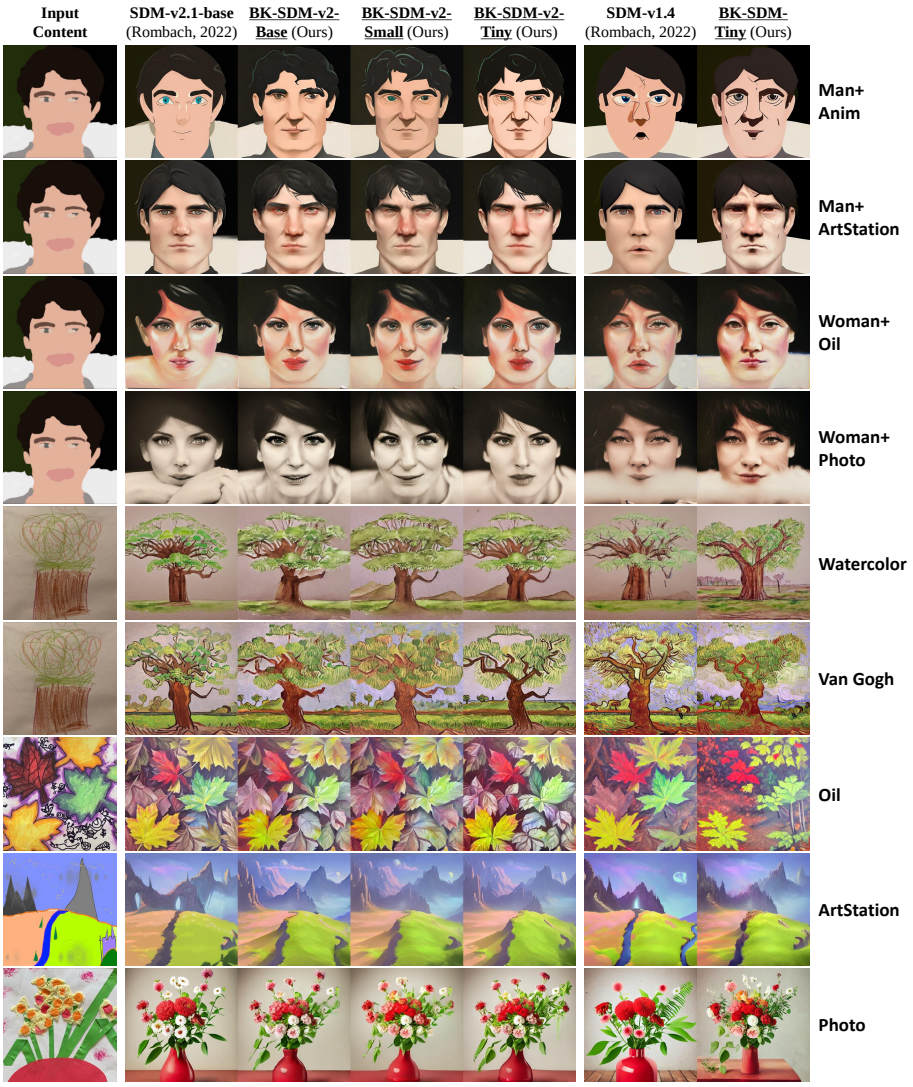

Fig. 22: Results of text-guided image-to-image translation. Our small models effectively stylize input images.

G Deployment on Edge Devices

Our models are tested on NVIDIA Jetson AGX Orin 32GB, benchmarked against SDM-v1.5 [62, 65] under the same default setting of Stable Diffusion WebUI [1]. For the inference, 20 denoising steps, DPM++ 2M Karras sampling [29, 42], and xFormers-optimized attention [34] are used to synthesize 512×512 images. BK-SDM shows quicker generation at 3.4 seconds, compared to the 4.9 seconds of SDM-v1.5 (see Figs. 23 and 26 with BK-SDM-Base trained on 2.3M pairs).

Fig. 23: Deployment on NVIDIA Jetson AGX Orin 32GB.

We also deploy our models on iPhone 14 with post-training palettization [52] and compare them against the original SDM-v1.4 [62, 64] converted with the identical setup. With 10 denoising steps and DPM-Solver [41, 42], 512×512 images are generated from given prompts. The inference takes 3.9 seconds using BK-SDM, which is faster than 5.6 seconds using SDM-v1.4, while maintaining acceptable image quality (see Fig. 24 with BK-SDM-Small trained on 2.3M pairs).

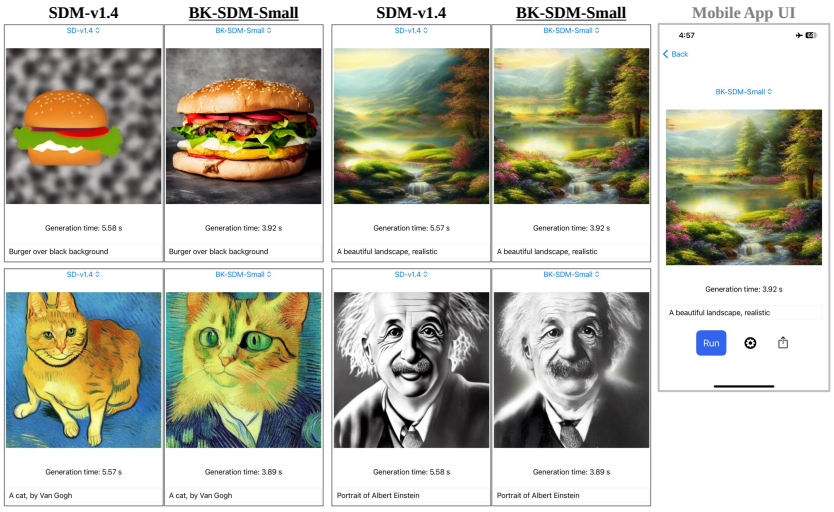

Fig. 24: Deployment on iPhone 14.

Additional results using different models can be found in Fig. 25.

Fig. 25: Additional examples from deployment on edge devices.

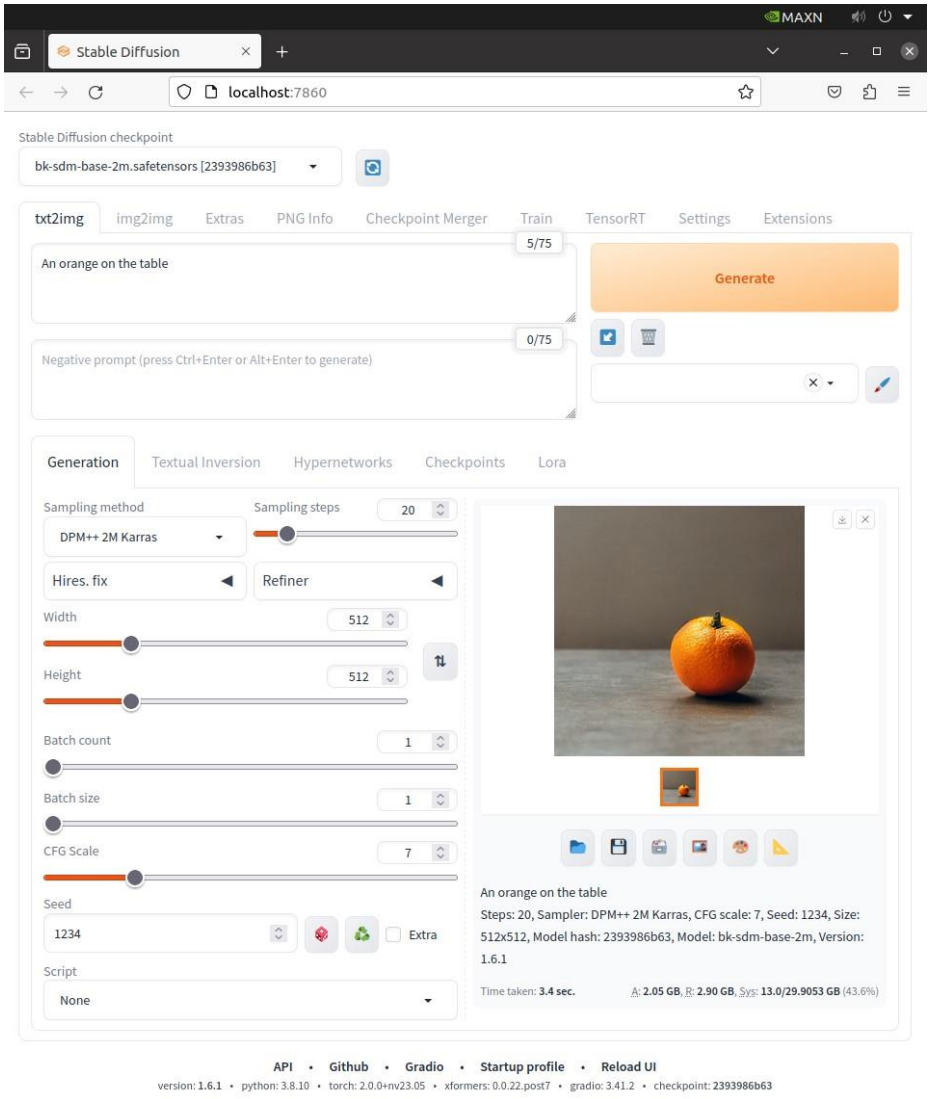

Fig. 26: Stable Diffusion WebUI [1] used in the deployment on AGX Orin.

H Impact of Training Data Volume

Fig. 27 illustrates how varying data sizes affects the training of BK-SDM-Small. Fig. 28 presents additional visual outputs of the following models: BK-SDM-{Base, Small, Tiny} trained on 212K (i.e., 0.22M) pairs and BK-SDM-{Base-2M, Small-2M, Tiny-2M} trained on 2256K (2.3M) pairs.

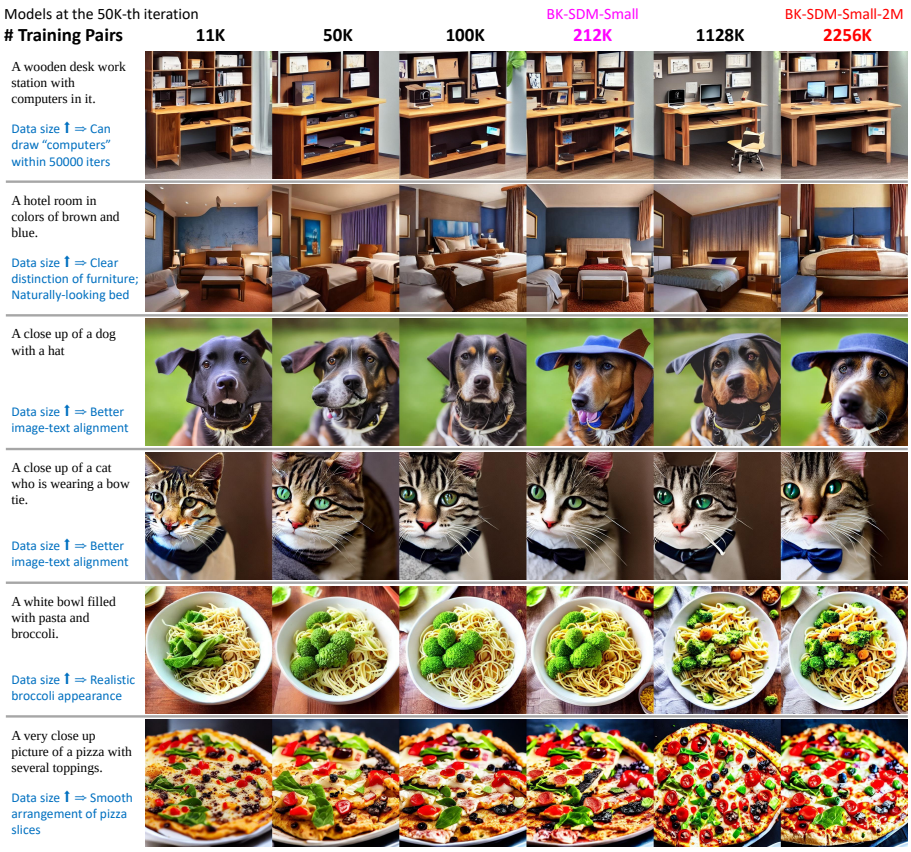

Fig. 27: Varying data quantities in training BK-SDM-Small. As the amount of data increases, the visual outcomes improve, such as enhanced image-text matching and clearer differentiation between objects.

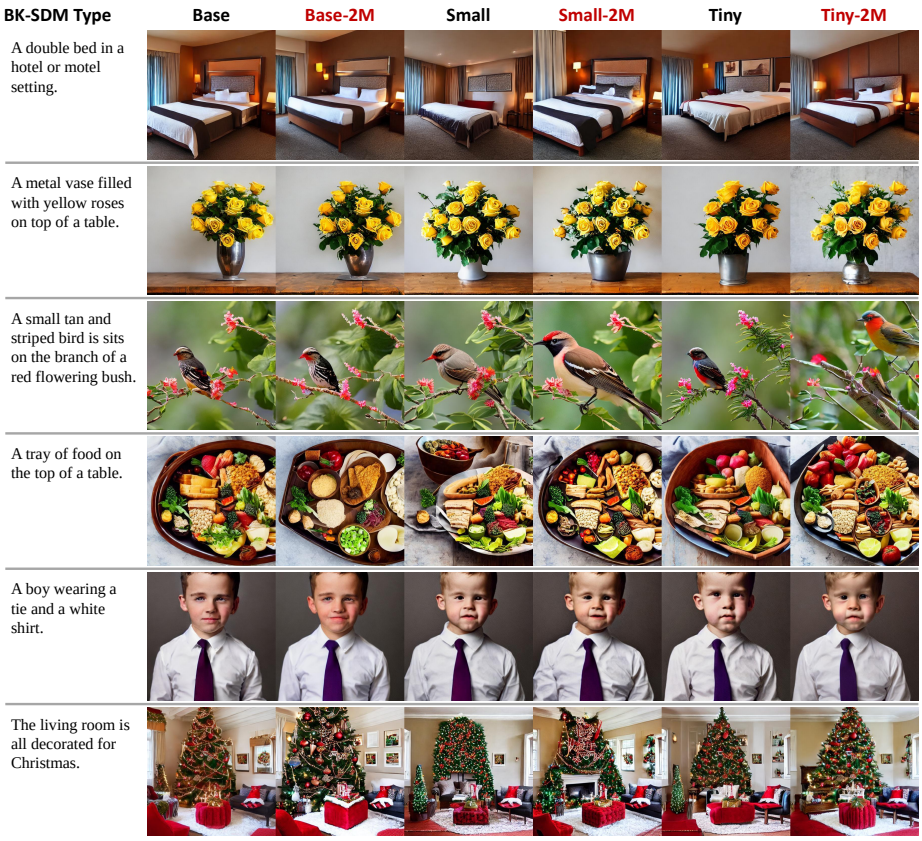

Fig. 28: Results of BK-SDM-{Base, Small, Tiny} trained on 0.22M pairs and {Base-2M, Small-2M, Tiny-2M} trained on 2.3M pairs.

I Additional Experiments

More Architectural Exploration. A model that falls between the original and our base size can be achieved by removing the mid-stage (see Tab. 11). Meanwhile, the CLIP criterion can be used to obtain models of various sizes (Tab. 12), though their performance is suboptimal.

Table 11: A model between the original and our base size.

Model	FID↓	IS↑	CLIP↑	# Param, U-Net
Original SD-v2.1-base	13.93	35.93	0.3075	866M
Mid-Stage Removal	14.84	37.29	0.3093	769M
Ours-v2-Base	15.85	31.70	0.2868	584M

Retraining with batch 128, 0.22M data, 50K iters.

Table 12: Additional structural variation. The CLIP-Score criterion can be used to yield models of multiple sizes, but their results are inferior to ours.

Model (# Blocks Removed)		FID↓	IS↑	CLIP↑	U-Net
Base Size	CLIP Criterion (15)	14.06	30.91	0.2787	606M
	Ours-v1-Base (14)	15.02	32.40	0.2841	580M
In-Between	CLIP Criterion (17)	17.65	27.06	0.2553	540M
Small Size	CLIP Criterion (19)	21.86	22.01	0.2283	497M
	Ours-v1-Small (17)	16.83	30.40	0.2668	483M

Retraining with batch 128, 0.22M data, 50K iters.

Effect of Learning Rate (LR). LRs of 5e-5 (used in the main paper) and 2.5e-5 yield good results (see Tab. 13). Extremely high or low LR values are detrimental.

Table 13: Effect of learning rate (LR). BK-SDM-v2-Small.

LR	1.0e-5	2.5e-5	5.0e-5	1.0e-4	2.5e-4
FID↓	15.24	15.69	16.61	17.43	18.93
IS↑	29.77	31.64	31.73	30.58	28.90
CLIP↑	0.2844	0.2906	0.2901	0.2871	0.2775

Retraining with batch 128, 0.22M data, 50K iters.

Analysis of Skip Connections. We remove the second channel concatenation (concat) and R-A pairs in each up stage, while retaining the first and third ones.

For further analysis, we corrupt the features from skip connections by forcibly assigning zero values (see Fig. 29). Consistent with our design, the inner concats are very robust to zeroing and are prunable. Moreover, the second concats are more removable than the others. Note that the first R blocks are often unprunable to utilize the teacher's weights.

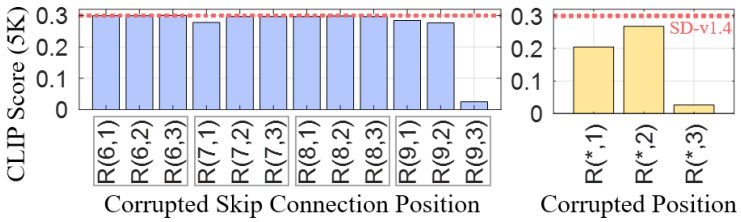

Fig. 29: Analysis of skip connections. We corrupt incoming features from channel concatenation in each up stage (left) and multiple stages (right). Higher scores imply removable units.

J Implementation

We adjust the codes in Diffusers [56] for distillation retraining and PEFT [44] for per-subject finetuning, both of which adopt the training process of DDPM [23] in latent spaces.

Distillation Retraining for General-purpose T2I. 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 retraining with the AdamW optimizer and a constant learning rate of 5e-5. The number of steps for gradient accumulation is always set to 4. With a total batch size of 256 (=4×64), 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 Setup. Following the default setup, we use PNDM scheduler [40] for zero-shot T2I generation and DPM-Solver [41, 42] for DreamBooth results. For compute efficiency, we always opt for 25 denoising steps of the U-Net, unless specified. The classifier-free guidance scale [24, 70] is set to the default value of 7.5, except the analysis in Fig. 10.

Image-to-Image Translation. We use the SDEdit method [47] implemented in Diffusers [56], with the strength value of 0.8.

Distillation Retraining for Unconditional Face Generation. A similar approach to our T2I training is applied. For the 30K-iteration retraining, we use a batch size of 64 (=4×16) and set the KD loss weights to 100.