Appendix

A.1 Hyper-parameters for Imagen fine-tuning and sample generation.

The quality, diversity, and speed of text-conditioned diffusion model sampling are strongly affected by multiple hyper-parameters. These include the number of diffusion steps, where larger numbers of diffusion steps are often associated with higher quality images and lower FID. Another hyper-parameter is the amount of noise-conditioning augmentation (Saharia et al., 2022b), which adds Gaussian noise to the output of one stage of the Imagen cascade at training time, prior to it being input to the subsequent super-resolution stage. We considered noise levels between 0 and 0.5 (with images in the range [0,1]), where adding more noise during training degrades more fine-scale structure, thereby forcing the subsequent super-resolution stage to be more robust to variability in the images generated from the previous stage.

During sampling, we use classifier-free guidance (Ho & Salimans, 2022; Nichol et al., 2021), but with smaller guidance weights than Imagen, favoring diversity over image fidelity to some degree. With smaller guidance weights, one does not require dynamic thresholding (Saharia et al., 2022b) during inference; instead we opt for a static threshold to clip large pixel values at each step of denoising. Ho et al. (Ho & Salimans, 2022) identify upper and lower bounds on the predictive variance, $\Sigma_\theta(x_t, t)$, used for sampling at each denoising step. Following (Nichol & Dhariwal, 2021) (Eq. 15) we use a linear (convex) combination of the log upper and lower bounds, the mixing parameter for which is referred to as the logvar parameter. Figures 3 and 4 show the dependence of FID Val, and training set IS and Classification Accuracy Scores on guidance weight and logvar mixing coefficient for the base model at resolution 64×64 and the $64 \rightarrow 256$ super-resolution model. These were used to help choose model hyper-parameters for large-scale sample generation.

Below are further results relate to hyperparameter selection and its impact on model metrics.

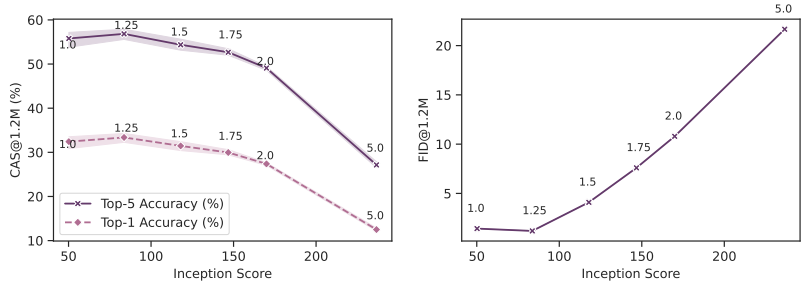

Figure A.1: **Left**: Training set CAS vs IS Pareto curves for train set resolution of 64×64 showing the impact of guidance weights. **Right**: FID Train vs IS Pareto curves for resolution of 64x64 showing the impact of guidance weights.

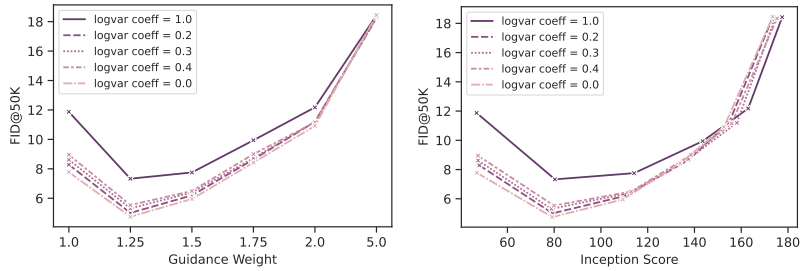

Figure A.2: Sampling refinement for 64×64 base model. **Left**: Validation set FID vs. guidance weights for different values of log-variance. **Right**: Validation set FID vs. Inception score (IS) when increasing guidance from 1.0 to 5.0.

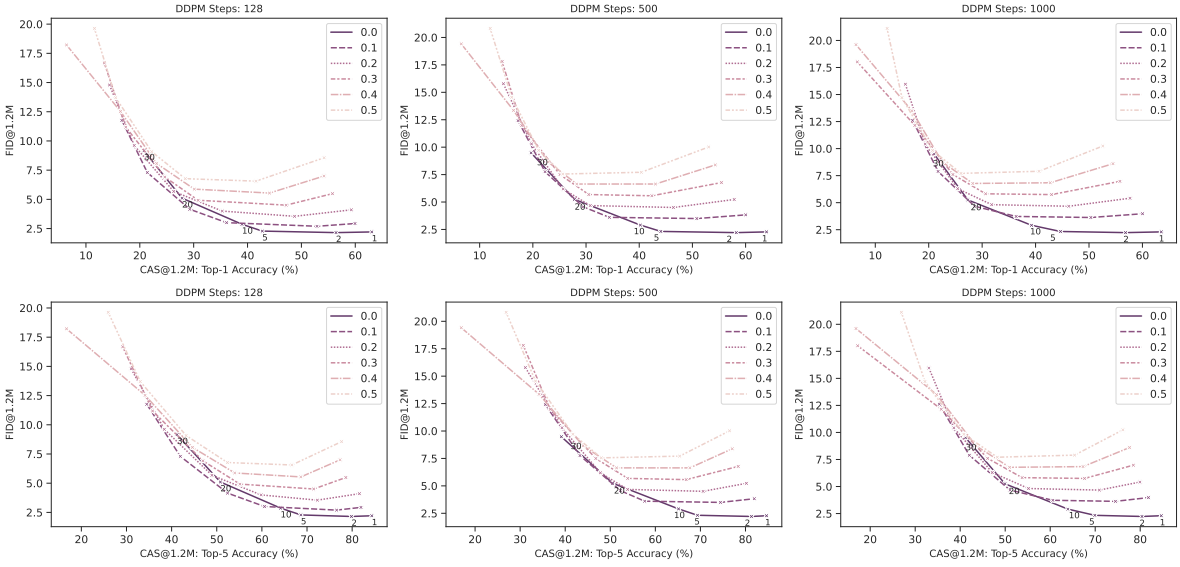

Figure A.3: Top-1 and Top-5 training set classification accuracy score (CAS@1.2M) vs FID Train (FID@1.2M) Pareto curves (sweeping over guidance weight) showing the impact of conditioning noise augmentation at 256×256 when sampling with different number of steps. As indicated by number overlaid on each trend line, guidance weight is decreasing from 30 to 1.

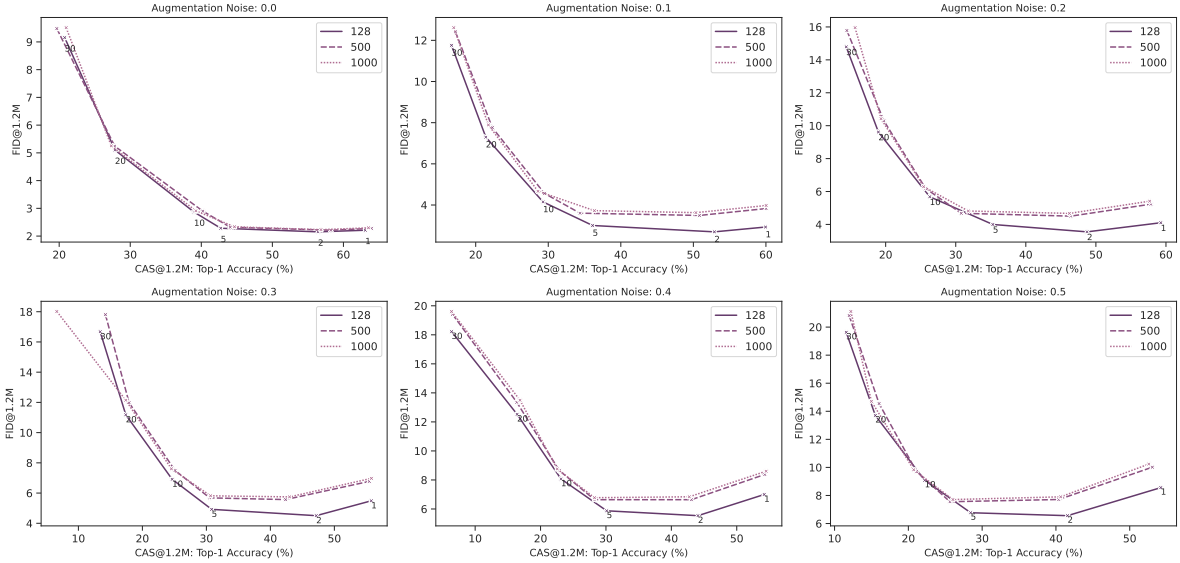

Figure A.4: Top-1 and Top-5 training set classification accuracy score (CAS@1.2M) vs FID Train (FID@1.2M) Pareto curves (sweeping over guidance weight) showing the impact of conditioning noise augmentation at 256×256 when sampling with different number of steps at a fixed noise level. As indicated by number overlaid on each trend line guidance weight is decreasing from 30 to 1. At highest noise level (0.5) lowering number sampling step and decreasing guidance can lead to a better joint FID@1.2M and CAS@1.2M. At lowest noise level (0.0) this effect is subtle and increasing sampling steps and lower guidance weight can help to improve CAS@1.2M.

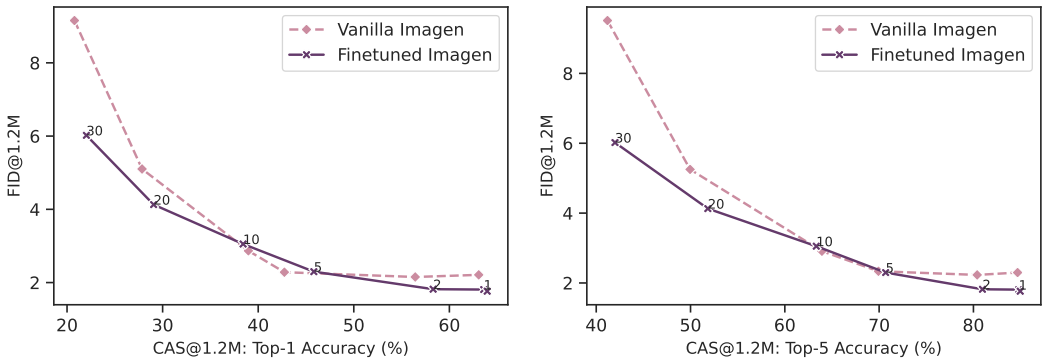

Figure A.5: Fine-tuning of SR model helps to jointly improve classification accuracy as well as FID of the vanilla Imagen.

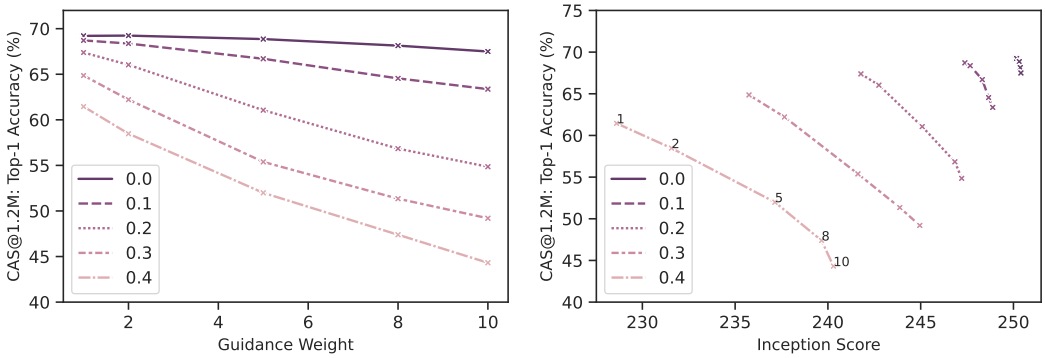

Figure A.6: Sampling refinement for 1024×2014 super resolution model. **Left**: CAS vs. guidance weights under varying noise conditions. **Right**: Training set CAS vs. Inception score (IS) when increasing guidance from 1.0 to 5.0 under varying noise conditions.

A.2 Class Alignment of Imagen vs. Fine-Tuned Imagen

What follows are more samples to compare our fine-tuned model vs. the Imagen model are provided in Figures A.7, A.8, and A.9. In this comparison we sample our fine-tuned model using two strategies. First, we sample using the proposed vanilla Imagen hyper-parameters which use a guidance weight of 10 for the sampling of the base 64×64 model and subsequent super-resolution (SR) models are sampled with guidance weights of 20 and 8, respectively. This is called the high guidance strategy in these figures. Second, we use the proposed sampling hyper-parameters as explained in the paper which includes sampling the based model with a guidance weight of 1.25 and the subsequent SR models with a guidance weight of 1.0. This is called the low guidance weight strategy in these figures.

Figure A.7: Example 1024×1024 images from vanilla Imagen (first row) vs. fine-tuned Imagen sampled with Imagen hyper-parameters (high guidance, second row) vs. fine-tuned Imagen sampled with our proposed hyper-parameter (low guidance, third row). Fine-tuning and careful choice of sampling parameters help to improve the alignment of images with class labels, and also improve sample diversity. Sampling with higher guidance weight can improve photorealism, but lessens diversity.

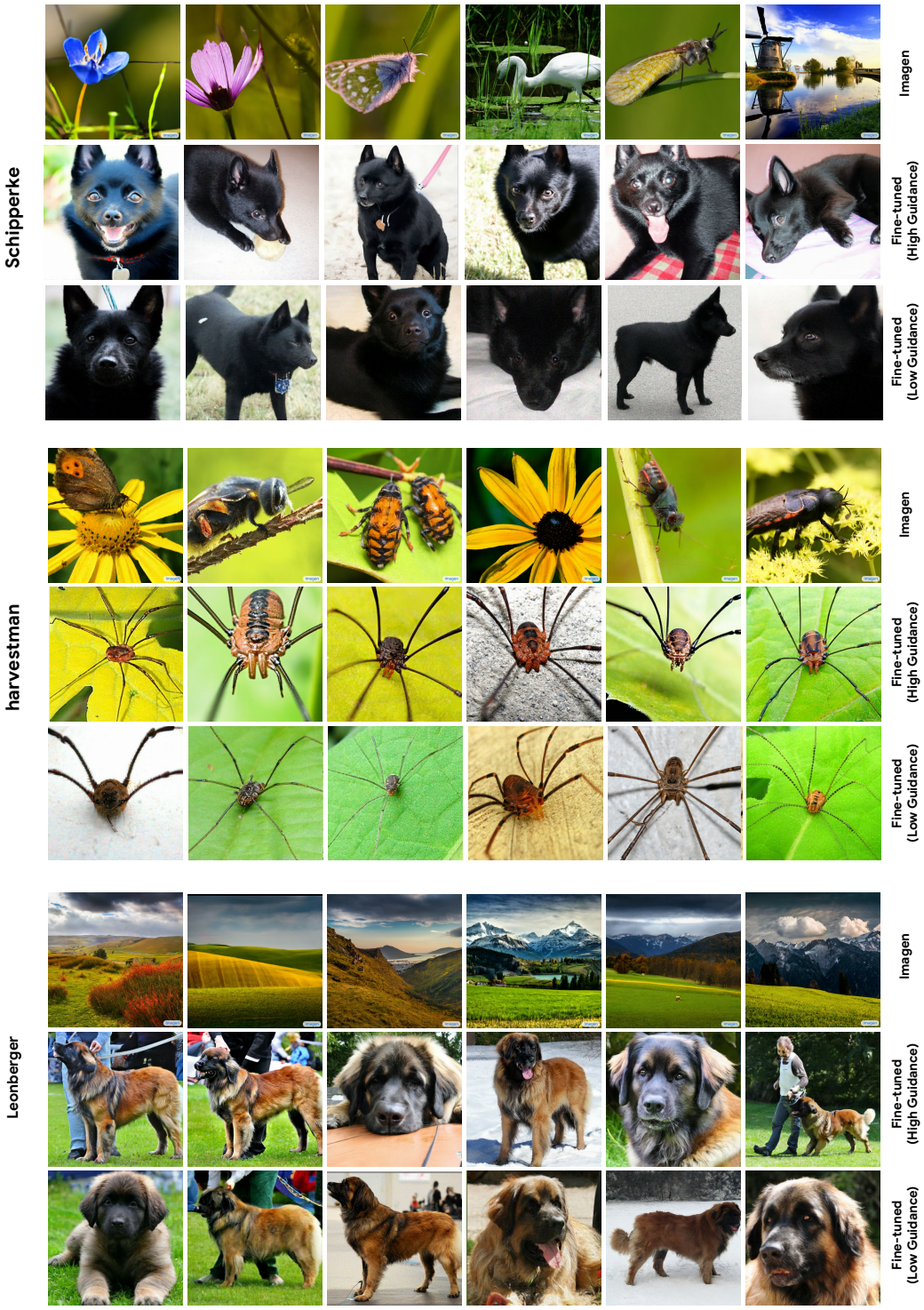

Figure A.8: Example 1024×1024 images from vanilla Imagen (first row) vs. fine-tuned Imagen sampled with Imagen hyper-parameters (high guidance, second row) vs. fine-tuned Imagen sampled with our proposed hyper-parameter (low guidance, third row). Fine-tuning and careful choice of sampling parameters help to improve the alignment of images with class labels, and also improve sample diversity. Sampling with higher guidance weight can improve photorealism, but lessens diversity.

Figure A.9: Example 1024×1024 images from vanilla Imagen (first row) vs. fine-tuned Imagen sampled with Imagen hyper-parameters (high guidance, second row) vs. fine-tuned Imagen sampled with our proposed hyper-parameter (low guidance, third row). Fine-tuning and careful choice of sampling parameters help to improve the alignment of images with class labels, and also improve sample diversity. Sampling with higher guidance weight can improve photorealism, but lessens diversity.

A.3 High Resolution Random Samples from the ImageNet Model

Figure A.10: Random samples at 1024×1024 resolution generated by our fine-tuned model. The classes are snail (113), panda (388), orange (950), badger (362), indigo bunting (14), steam locomotive (820), carved pumpkin (607), lion (291), loggerhead sea turtle (33), golden retriever (207), tree frog (31), clownfish (393), dowitcher (142), lorikeet (90), school bus (779), macaw (88), marmot (336), green mamba (64).

A.4 Hyper-parameters and model selection for ImageNet classifiers.

This section details all the hyper-parameters used in training our ResNet-based model for CAS calculation, as well as the other ResNet-based, ResNet-RS-based, and Transformer-based models, used to report classifier accuracy in Table 3. Table A.1 and Table A.2 summarize the hyper-parameters used to train the ConvNet architectures and vision transformer architectures, respectively.

For classification accuracy (CAS) calculation, as discussed before we follow the protocol suggested in (Ravuri & Vinyals, 2019). Our CAS ResNet-50 classifier is trained using a single crop. To train the classifier, we employ an SGD momentum optimizer and run it for 90 epochs. The learning rate is scheduled to linearly increase from 0.0 to 0.4 for the first five epochs and then decrease by a factor of 10 at epochs 30, 60, and 80. For other ResNet-based classifiers we employ more advanced mechanisms such as using a cosine schedule instead of step-wise learning rate decay, larger batch size, random augmentation, dropout, and label smoothing to reach competitive performance (Sun et al., 2017). It is also important to emphasize that ResNet-RS achieved higher performance than ResNet models through a combination of enhanced scaling strategies, improved training methodologies, and the implementation of techniques like the Squeeze-Excitation module (Bello et al., 2021). We follow the training strategy and hyper-parameter suggested in (Bello et al., 2021) to train our ResNet-RS-based models.

For vision transformer architectures we mainly follow the recipe provided in (Beyer et al., 2022) to train a competitive ViT-S/16 model and (Touvron et al., 2021) to train DeiT family models. In all cases we re-implemented and train all of our models from scratch using real only, real + generated data, and generated only data until convergence.

Table A.1: Hyper-parameters used to train ConvNet architectures including ResNet-50 (CAS) (Ravuri & Vinyals, 2019), ResNet-50, ResNet-101, ResNet-152, ResNet-RS-50, ResNet-RS-101, and ResNet-RS-152 (Bello et al., 2021).

Model	ResNet-50 (CAS)	ResNet-50	ResNet-101	ResNet-152	ResNet-RS-50	ResNet-RS-101	ResNet-RS-152
Epochs	90	130	200	200	350	350	350
Batch size	1024	4096	4096	4096	4096	4096	4096
Optimizer	Momentum	Momentum	Momentum	Momentum	Momentum	Momentum	Momentum
Learning rate	0.4	1.6	1.6	1.6	1.6	1.6	1.6
Decay method	Stepwise	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Weight decay	1e-4	1e-4	1e-4	1e-4	4e-5	4e-5	4e-5
Warmup epochs	5	5	5	5	5	5	5
Label smoothing	-	0.1	0.1	0.1	0.1	0.1	0.1
Dropout rate	-	0.25	0.25	0.25	0.25	0.25	0.25
Rand Augment	-	10	15	15	10	15	15

Table A.2: Hyper-parameters used to train the vision transformer architectures, i.e., ViT-S/16 (Beyer et al., 2022), DeiT-S (Touvron et al., 2021), DeiT-B (Touvron et al., 2021), and DeiT-L (Touvron et al., 2021).

Model	ViT-S/16	DeiT-S	DeiT-B	DeiT-L
Epochs	300	300	300	300
Batch size	1024	4096	4096	4096
Optimizer	AdamW	AdamW	AdamW	AdamW
Learning rate	0.001	0.004	0.004	0.004
Learning rate decay	Cosine	Cosine	Cosine	Cosine
Weight decay	0.0001	-	-	-
Warmup epochs	10	5	5	5
Label smoothing	-	0.1	0.1	0.1
Rand Augment	10	9	9	9
Mixup prob.	0.2	0.8	0.8	0.8
Cutmix prob.	-	1.0	1.0	1.0

Table A.3: Augmenting the real ImageNet training dataset by adding synthetic images, at resolutions 64×64, 256×256 and 1024×1024, we measure the mean accuracy (on the ImageNet Val set) and standard deviation of ResNet-50 ImageNet classifiers, computed from 10 independent training runs. The baseline Top-1 accuracy of the classifier trained on real data (ie the top row) is 76.39 ± 0.21. These results are plotted in Figure 6.

Train Set Size (M)	64×64		256×256		1024×1024	
	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
1.2	68.63 ± 0.15	88.42 ± 0.10	76.39 ± 0.21	93.25 ± 0.04	76.39 ± 0.21	93.25 ± 0.08
1.5	69.53 ± 0.13	88.65 ± 0.05	76.64 ± 0.12	93.35 ± 0.06	77.05 ± 0.09	93.47 ± 0.07
1.8	69.93 ± 0.08	89.01 ± 0.01	77.09 ± 0.08	93.50 ± 0.01	77.57 ± 0.12	93.65 ± 0.09
2.1	70.53 ± 0.09	89.47 ± 0.08	77.30 ± 0.05	93.61 ± 0.07	77.95 ± 0.07	93.85 ± 0.04
2.4	70.65 ± 0.12	89.71 ± 0.01	77.61 ± 0.08	93.84 ± 0.03	78.12 ± 0.05	94.07 ± 0.06
2.7	70.98 ± 0.10	89.86 ± 0.03	77.49 ± 0.04	93.69 ± 0.08	77.94 ± 0.10	93.91 ± 0.11
3.0	71.37 ± 0.12	89.97 ± 0.05	77.36 ± 0.08	93.65 ± 0.04	77.72 ± 0.13	93.78 ± 0.07
3.3	71.47 ± 0.09	90.13 ± 0.01	77.25 ± 0.07	93.62 ± 0.06	77.58 ± 0.12	93.71 ± 0.04
3.6	71.53 ± 0.09	90.30 ± 0.05	77.16 ± 0.04	93.55 ± 0.04	77.48 ± 0.04	93.66 ± 0.05
4.8	71.98 ± 0.09	90.50 ± 0.05	76.52 ± 0.04	93.18 ± 0.08	76.75 ± 0.07	93.25 ± 0.04
6.0	72.31 ± 0.10	90.69 ± 0.07	76.09 ± 0.08	92.94 ± 0.07	76.34 ± 0.13	92.95 ± 0.06
7.2	72.44 ± 0.11	90.81 ± 0.05	75.81 ± 0.08	92.77 ± 0.08	75.87 ± 0.09	92.71 ± 0.01
8.4	72.65 ± 0.10	90.84 ± 0.10	75.44 ± 0.06	92.62 ± 0.04	75.57 ± 0.07	92.50 ± 0.06
9.6	72.75 ± 0.09	90.90 ± 0.04	75.28 ± 0.10	92.52 ± 0.07	75.10 ± 0.19	92.26 ± 0.07
10.8	72.86 ± 0.11	90.91 ± 0.04	75.11 ± 0.12	92.44 ± 0.04	74.72 ± 0.13	91.96 ± 0.20
12.0	72.98 ± 0.09	91.01 ± 0.03	75.04 ± 0.05	92.31 ± 0.06	74.24 ± 0.09	91.64 ± 0.16