A Hyperparameter Settings

All experiments performed in this paper followed the settings in Tab. 7. In particular, we ensure that their batch sizes are the same for experiments with different n_{step}. Furthermore, for our proposed JDA, the magnitudes of the corresponding sub-policies are shown in Tab. 8. Most of these settings are obtained with modifications based on AutoAugment.

Table 7: This table shows the hyperparameter settings when n_{step} is $\times 1$. At the same time, we achieve the setting where n_{step} is $\times 2$ and $\times 4$ by increasing epoch. Crucially, we use the same batch size of the real input for all methods including JDA and *rotating*.

Datasets	Learning Rate	Optimizer	Weight Decay	Batch Size	Epoch	Scheduled Epoch	Gamma
CIFAR-10	0.1	SGD	1e-4	128	182	$[91, 136]$	0.1
CIFAR-100	0.05	SGD	1e-4	256	240	$[150, 180, 210]$	0.1
ImageNet-1k	0.1	SGD	1e-4	1024	100	$[30, 60, 90]$	0.1

Table 8: This table shows the 14 sub-policies and their hyperparameter settings used in our experiments. Part of this table is copied from [5]. And the execution order of the sub-policies can be found in our released codes.

Operation Name	Description	magnitude in CIFAR-10	magnitude in CIFAR-100
ShearX	Shear the image along the horizontal axis with rate *magnitude*.	0.24	0.15
ShearY	Shear the image along the vertical axis with rate *magnitude*.	0.24	0
TranslateX	Translate the image in the horizontal direction by *magnitude* number of pixels.	$\frac{45}{331}$	$\frac{15}{331}$
TranslateY	Translate the image in the vertical direction by *magnitude* number of pixels.	$\frac{45}{331}$	$\frac{120}{331}$
Rotate	Rotate the image *magnitude* degrees.	6	9
AutoContrast	Maximize the the image contrast, by making the darkest pixel black and lightest pixel white.	-	-
Invert	Invert the pixels of the image.	-	-
Equalize	Equalize the image histogram.	-	-
Solarize	Invert all pixels above a threshold value of *magnitude*.	204.8	153.6
Posterize	Reduce the number of bits for each pixel to *magnitude* bits.	0	8
Brightness	Adjust the brightness of the image. A *magnitude*=0 gives a black image, whereas *magnitude*=1 gives the original image.	0.54	0.27
Contrast	Control the contrast of the image. A *magnitude*=0 gives a gray image, whereas *magnitude*=1 gives the original image.	0.63	0.27
Color	Adjust the color balance of the image, in a manner similar to the controls on a colour TV set. A *magnitude*=0 gives a black & white image, whereas *magnitude*=1 gives the original image.	0.27	0.36
Sharpness	Adjust the sharpness of the image. A *magnitude*=0 gives a blurred image, whereas *magnitude*=1 gives the original image.	0.81	0.45

B Additional Method Comparisons

First, we present a series of computational cost comparisons of SOTA algorithms for knowledge distillation in Tab. 9. Second, we compare the difference in performance between JDA and AutoAugment on knowledge distillation in Tab. 10. The results show that both JDA and CCD achieve the best performance in their respective comparisons.

Table 9: **GFLOPs:** *Giga Floating-point Operations Per Second*. We utilize facebook's open-source project fvcore to calculate GFLOPs. For operators that fvcore does not support statistics, we count their totals in NUO. **NUO:** *The Number of Unsupported Operators*. **TP:** *ThroughPut (images/s)*. We calculated the throughput of all methods from start to finish under an NVIDIA RTX 3080 Ti. Meanwhile, all methods are executed 5000 times to reduce interference. This table presents the comparison results of related knowledge distillation methods on other vital indicators. In general, response-based methods are more portable and reproducible than feature-based methods. Methods that do not use additional modules are more lightweight in training than methods that use additional modules. JDA+CCD does not require additional modules and is very close to the original KD regarding GFLOPs, NUO and TP. Therefore, we can conclude that our proposed JDA+CCD is lightweight.

Methods	Additional Modules	Location of Distillation	GFLOPs↓	NUO↓	TP↑
vanilla KD	No	Response-based	0.4313672	36	16244.3
SPKD	No	Feature-based	0.4314327	50	16192.9
CRD	Yes	Feature-based	0.4355945	115	6726.8
SSKD	Yes	Feature-based	0.4314327	68	14303.8
HSAKD	Yes	Feature-based	1.0127411	93	7679.0
CCD+JDA (ours)	No	Response-based	0.4313672	49	15192.6

Table 10: Performance comparison of JDA and AutoAugment on offline knowledge distillation. All experiments in this table use the same hyperparameter settings. As a result, we find that JDA beats AutoAugment on four teacher-student pairs.

Teacher	WRN-40-2	WRN-40-2	ResNet56	ResNet32×4	VGG13	n_{step}
Student	WRN-16-2	WRN-40-1	ResNet20	ResNet8×4	MobileNetV2	
KD+JDA	76.80%	76.18%	72.37%	76.50%	77.64%	$\times 2$
KD+AutoAugment	76.34%	75.68%	71.97%	75.73%	77.64%	$\times 2$

C Additional Visualization

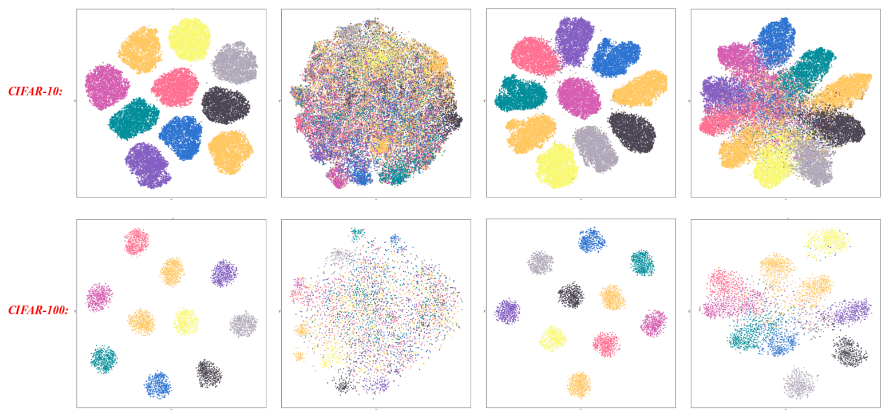

Fig. 6: The figure contains T-SNE visualizations of the output of the teacher model's GAP for eight different scenarios. The four columns from left to right refer to the four cases of $(\mathcal{X}, \mathcal{X})$, $\left(\widetilde{\mathcal{X}}, \mathcal{X}\right)$, $\left(\mathcal{X}, \widetilde{\mathcal{X}} + \mathcal{X}\right)$ and $\left(\widetilde{\mathcal{X}}, \mathcal{X} + \widetilde{\mathcal{X}}\right)$, where (A, B) stands for the teacher model trained with B. Then, we adopt T-SNE to visualize A.