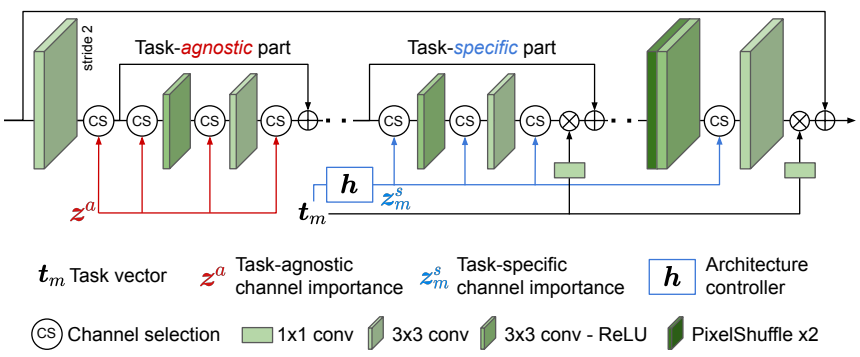
Searching for Controllable Image Restoration Networks
– *Supplementary Document* –

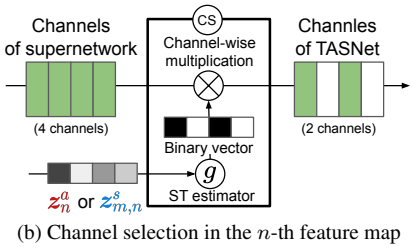
A. Implementation details

Supernetwork architecture. We use the network architecture of CResMD as the supernetwork in the proposed search algorithm. Our supernetwork consists of 32 enhanced residual blocks which have a ReLU activation layer between two convolution layers with 64 filters of the kernel size 3×3. The first convolution layer with a stride of 2 downscales the input images, and the last upsampling module consists of PixelShuffle layer, two convolution layers, and a ReLU activation layer. Global skip connection adds the input image to the output of the upscaling module. A task vector scales the residual feature map in the location of 32 local connections and 1 global connection by a 1×1 convolution layer with channel-wise multiplication.

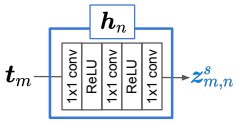
TASNet architecture. The proposed algorithm determines the number of shared layers and selects the channels of each shared or non-shared layer. Figure A(a) illustrates the TASNet architecture. For the shared layers (task-agnostic part), the channels that are not selected at the end of training are pruned in the final model. On the other hand, the non-shared layers (task-specific part) adaptively select their channels *w.r.t* the input task vector. During the training, the channels are virtually selected by channel-wise multiplication to the binary vectors, as described in Figure A(b). Our channel selection modules are located at all feature maps after the initial PixelShuffle layer of CResMD. The architecture controller consists of 3 fully-connected layers with ReLU activation function, as described in Figure A(c). In the task-agnostic part of the supernetwork, the residual scaling modules are removed to make the feature maps independent to specific tasks.



(a) Overview of TASNet architecture



(b) Channel selection in the n-th feature map



(c) Architecture controller in the n-th feature map

Figure A: TASNet architecture. (a) During searching for the number of shared early layers (task-*agnostic* part) in the supernetwork, z^a determines where to prune in the task-*agnostic* part. By contrast, z^s_m selects channels in the remaining layers (task-*specific* part) specialized in the m-th task-vector t_m (the control factor of restoration levels). We omit the notations for the feature map index n and the channel index c for simplicity. (b) In the network training phase, channel-wise multiplication between a binary vector and a feature map operates as virtual channel selection (CS) for the differentiable neural architecture search process. (c) Architecture controller consists of fully connected layers and predicts task-*specific* channel importance from the task vector.

Hyperparameters for the search algorithm. TASNet sets the hyperparameters α, γ, M, λ_1, and λ_2 as 0.9, 0.9, 64, $5\times e^{-11}$, and $1 \times e^{-2}$, respectively. The mini-batch consists of 64 image patches with 64×64 resolution. The initial learning rate is 1×10^{-4}. TASNet is trained for 1×10^6 iterations using Adam optimizer [45] with the learning rate decay of ×0.5 after the first half of training.

Image quality measure. In this work, we utilize three widely used image quality measures, PSNR, SSIM, NIQE [44], and BRISQUE [46] to evaluate the quality of images produced by models. PSNR and SSIM are full-reference measures in that the restored images are compared with the original clean images. On the other hand, NIQE and BRISQUE are no-reference evaluation metrics, in which the restored image quality is measured without referring to the original image. Images with higher

PSNR, higher SSIM, lower NIQE, and lower BRISQUE scores are considered to have better quality. However, measuring image quality during adjusting restoration levels has not been studied thoroughly. Thus, we visualize extensive qualitative results in both the main manuscript and the supplementary document.

Degradation in non-blind test set. For fair comparisons in non-blind setting, we construct CBSD68 dataset with the combinations of three levels and three types of degradation; Gaussian blur with $r \in \{0, 2, 4\}$, Gaussian noise with $\sigma \in \{0, 25, 50\}$, and JPEG compression with $q \in \{None, 60, 10\}$. Among the 27 combinations of degradation, we omit $(r, \sigma, q) = (0, 0, None)$ which generates identical images to the original. PSNR, SSIM, NIQE, and BRISQUE in all tables of this paper report the average scores on CBSD68 dataset with the 26 combinations of degradation.

Computation cost metric. We measure the computation costs of the networks in FLOPs and latency. FLOPs is a classical device-agnostic metric and exponentially increases by image resolution. Since latency is device-dependent, we measure latency on CPU with single-core (CPU latency (single)), CPU with multi-core (CPU latency (multi)), and GPU (GPU latency). We use Intel i7-5960X CPU which has 16 cores and GeForce RTX 2080 Ti GPU. The computation costs reported in this paper are average scores to generate images with 27 restoration levels unless otherwise mentioned. The task vectors $\boldsymbol{t} \in \mathbb{R}^3$ represent the 27 restoration levels by $t_d \in \{0, 0.5, 1\}$.

B. Additional experiments

Balancing the hyperparameters. Table A presents the ablation study of hyperparameters λ_1 and λ_2 which balance the trade-off between the network computation cost and the number of shared layers while minimizing Equation (8) of the main paper. The models trained with small λ_1 and large λ_2 have large portions of shared layers, and thus they are efficient in generating multiple (27) images (② *vs.* ③ and ⑤ *vs.* ④). In contrast, the models trained with the opposite balance between λ_1 and λ_2 are efficient for a single inference (② *vs.* ① and ⑤ *vs.* ⑥).

Table A: Ablation study of hyperparameters λ_1 and λ_2 on CBSD68.

Ex.#	λ_1	λ_2	#Shared layer	PSNR	FLOPs	
					Single inference	Multiple inferences
①		1×10^{-3}	18 %	25.67 dB	35.2 G	23.1 G
②	5×10^{-11}	1×10^{-2}	62 %	25.75 dB	52.9 G	7.5 G
③		1×10^{-1}	99 %	25.48 dB	125.5 G	4.8 G
④	5×10^{-12}		99 %	25.46 dB	154.6 G	6.0 G
⑤	5×10^{-11}	1×10^{-2}	62 %	25.75 dB	52.9 G	7.5 G
⑥	5×10^{-10}		16 %	25.50 dB	15.4 G	1.9 G

Extra qualitative results. We present more qualitative comparisons between CResMD and TASNet in the blind setting where users have to generate diverse restored images by controlling the restoration levels (task vectors) for unknown degradation of an input image. Recall that CResMD incurs three problems in this scenario: artifacts in the generated images, over-smoothed outputs, and uneven modulation across the task vectors. Figure B and C show that CResMD produces output images with undesired and visually unpleasing artifacts. Figure D presents less artifacts in the outputs of CResMD, but the outputs are over-smoothed compared to the outputs of TASNet even for the true task vector. Figure E also shows over-smoothed outputs for CResMD when restoring the input images with high restoration levels for denoising and dejpeg. By contrast, TASNet maintains the sharp textural details of the input image and removes visually unpleasing noise and compression artifacts of the input. Figure F exemplifies the problem of uneven modulation for CResMD. While CResMD produces images with negligible changes for lower values of deblurring level, it exhibits drastic changes for higher levels. In contrast to CResMD, TASNet demonstrates more even modulation across the different task vectors and generates smoothly-varying images. Figure G, H, and I presents modulation scenarios for a *real-word image* with unknown degradation, in which modulations with various task vectors are inevitable to find the visually pleasing images. These results demonstrate that CResMD sometimes generates severely destructive artifacts (especially in Figure G) and overly-smooth outputs (especially in Figure H) during the modulation process whereas TASNet generates plausible images for various task vectors.

Figure B: **Deblur modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure C: **Deblur and dejpeg modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

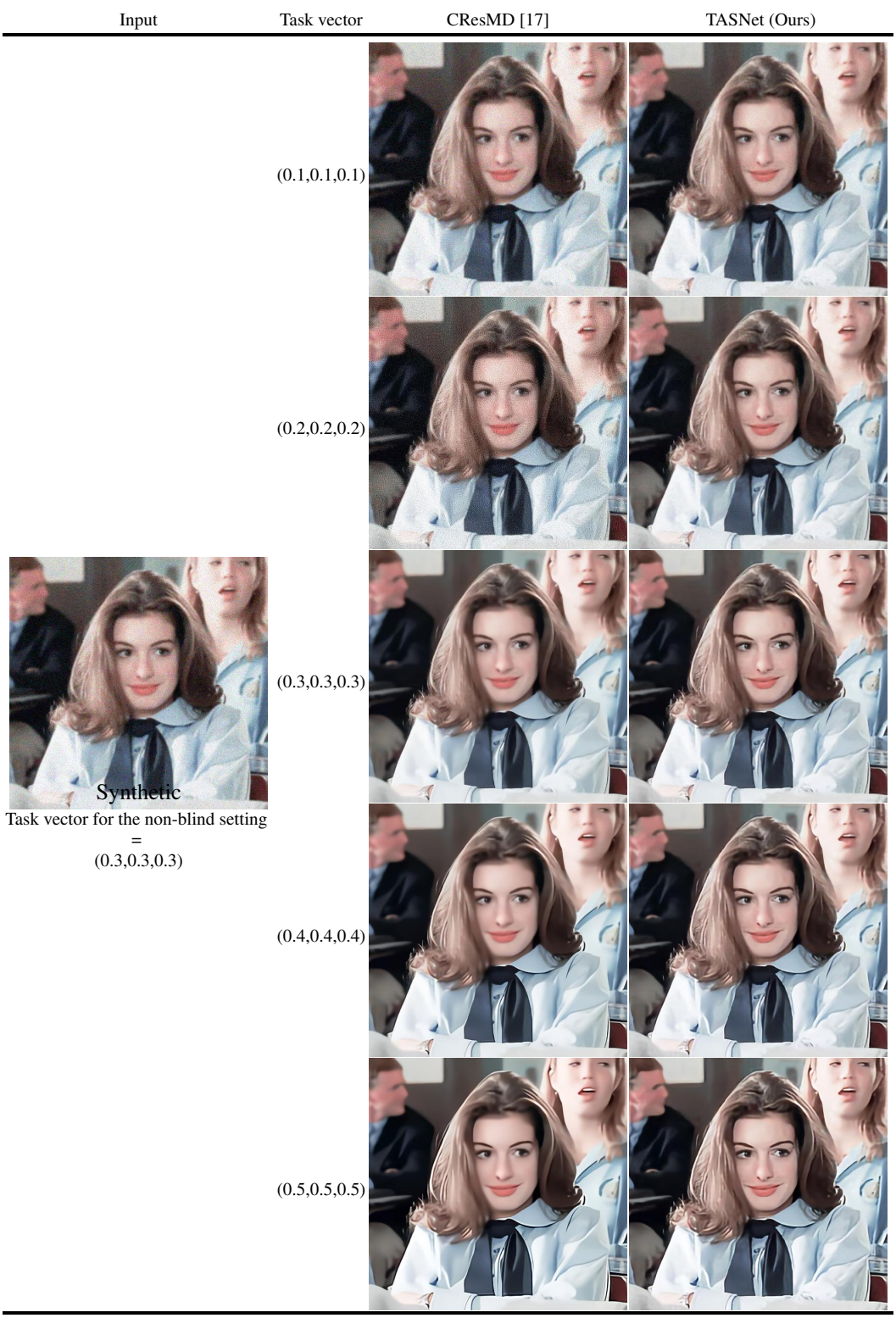


Figure D: **Deblur, denoise, and dejpeg modulation examples to the image with blur, noise, and jpeg compression**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts and over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure E: **Denoise and dejpeg modulation examples to the image with noise and jpeg compression**. Our TASNet generates less over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

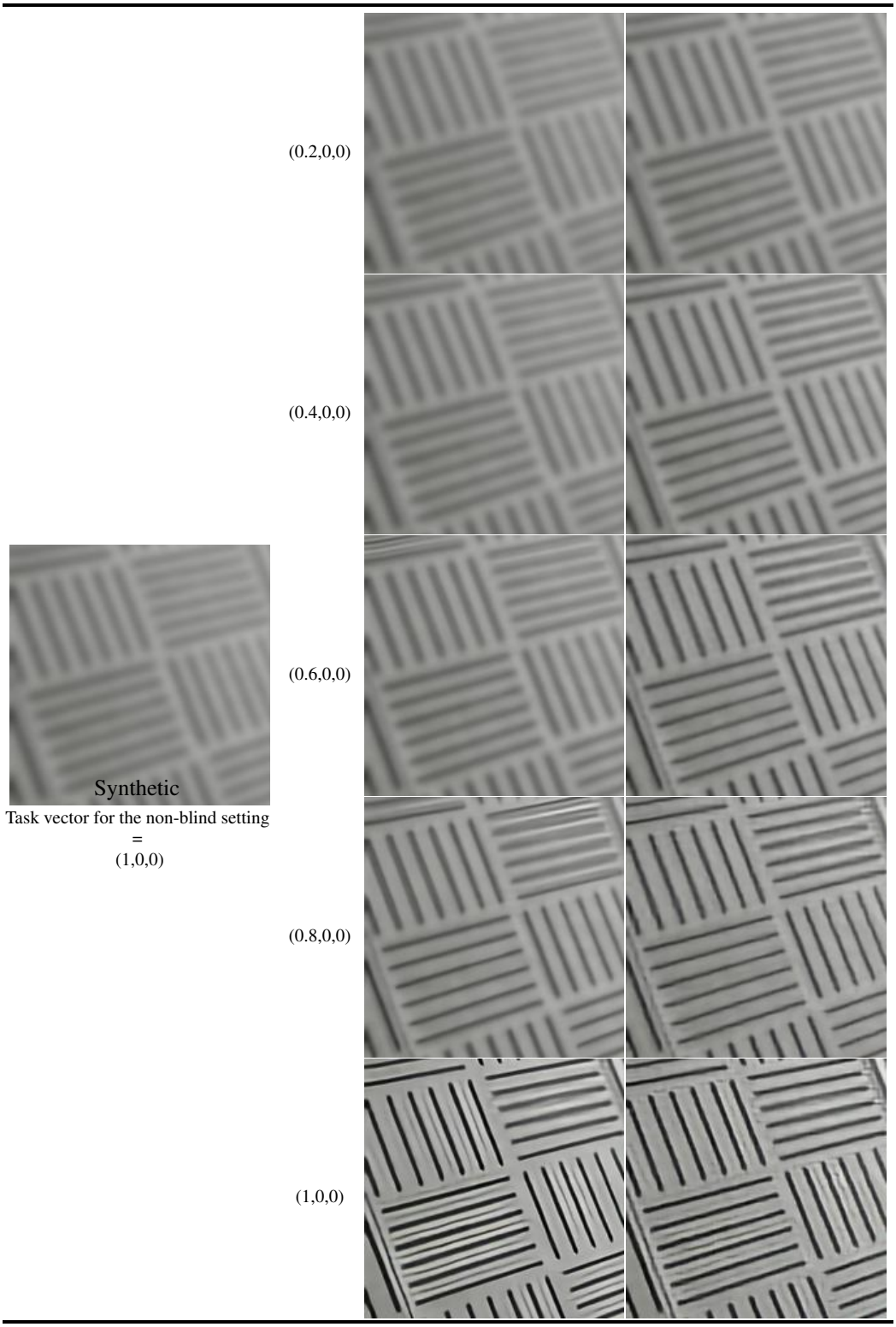


Figure F: **Deblur modulation examples to the image with blur**. Our TASNet generates evenly modulated images with respect to the given restoration level changes. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure G: **Deblur modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure H: **Denoise modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

Figure I: **Deblur, denoise, and dejpeg modulation examples to the real world image on the Internet**. Our TASNet generates diverse images with respect to the given restoration levels (task vectors). TASNet generates less auxiliary visual artifacts and over-smoothed textures. The values of task vector denote restoration levels of (deblur, denoise, dejpeg), respectively.

References

[44] Alan C. Bovik Anish Mittal, Rajiv Soundararajan. Making a completely blind image quality analyzer. *SPL*, 2013. 1

[45] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 1

[46] Anish Mittal, Anush K. Moorthy, and Alan C. Bovik. Blind/referenceless image spatial quality evaluator. In *ASILOMAR*, 2011. 1