PCA-Based Knowledge Distillation Towards Lightweight and Content-Style Balanced Photorealistic Style Transfer Models

Tai-Yin Chiu  
University of Texas at Austin  
chiu.taiyin@utexas.edu  
Danna Gurari  
University of Colorado Boulder  
danna.gurari@colorado.edu

![Figure 1. Exemplification of our PCA-based knowledge distillation resulting in models that are more lightweight, faster, and achieve a better content-style balance for photorealistic style transfer than existing models. We demonstrate its versatility by reporting results for when our approach is applied to two backbone architectures: VGG (Ours-VGG) and MobileNet (Ours-Mob). Compared to WCT$^2$, our smallest model uses only 0.7% of the parameters (73K vs. 10.12M) while producing results over 6x faster (0.04s vs. 0.27s). Additionally, our distilled models consistently transfer stronger style effects than WCT$^2$ [48] and preserve content better than PhotoWCT-HFR (an improved variant of PhotoWCT [24]), PhotoWCT$^2$ [11], and a CKD [43] distilled model.](image)

Abstract

Photorealistic style transfer entails transferring the style of a reference image to another image so the result seems like a plausible photo. Our work is inspired by the observation that existing models are slow due to their large sizes. We introduce PCA-based knowledge distillation to distill lightweight models and show it is motivated by theory. To our knowledge, this is the first knowledge distillation method for photorealistic style transfer. Our experiments demonstrate its versatility for use with different backbone architectures, VGG and MobileNet, across six image resolutions. Compared to existing models, our top-performing model runs at speeds 5-20x faster using at most 1% of the parameters. Additionally, our distilled models achieve a better balance between stylization strength and content preservation than existing models. To support reproducing our method and models, we share the code at https://github.com/chiutaiyin/PCA-Knowledge-Distillation.

1. Introduction

Photorealistic style transfer is the task of rendering an image (content image) in the style of a reference image (style image) to create a photorealistic result. Examples are shown in Fig. 1. A key challenge the community has focused on tackling since the seminal neural network-based algorithm for this task [30] has been how to simultaneously achieve a good balance between stylization strength and content preservation while running fast to better support practical applications [11, 24, 46, 48].

The status quo for modern photorealistic style transfer models is to use autoencoders. As illustrated in Fig. 2(a), the basic model uses a pre-trained VGG-19 [40] as the encoder to extract content and style features, then a feature transformation to adapt the content feature with respect to the style feature, and finally a decoder to invert the adapted feature to a stylized image. A limitation of this framework is that the speed of autoencoder-based approaches is limited by the large size of the VGG-19 backbone.

This size limitation is further amplified in state-of-the-art models, as they extend the basic autoencoder framework by using multiple feature transformations to better capture style effects. For instance, PhotoWCT [24] and PhotoWCT$^2$ [11] perform coarse-to-fine feature transformations to sequentially adapt the coarse content feature (i.e., $relu_{4,1}$ content feature from VGG) to the fine content feature (i.e., $relu_{1,1}$ content feature from VGG) with respect to the corresponding style features. This means the fine style feature is added on top of the coarse style feature and so can produce strong style effects. However, as

---

1VGG-19 is favored in artistic [7, 15, 16, 19, 20, 22, 23, 39] and photorealistic [4, 11, 24, 30, 46, 48] style transfer research due to its simple architecture with no complex multiple branches and residual modules, making VGG features easier to interpret for content and style.
shown in Fig. 1, the strong style effects may in turn introduce artifacts. Other models such as WCT2 [48] and PhotoNAS [4] perform fine-to-coarse feature transformations to adapt the fine content feature first and the coarse later to transfer weaker style effects and avoid artifacts. However, as shown in Fig. 1, the style effects may be poorly captured since adding the coarse style later partially overshadows the former fine style. Altogether, we observe that the state-of-the-art photorealistic style transfer models not only suffer from large sizes and slow speeds, but also poor balance between content preservation and stylization strength.

To address these issues, (1) we propose a **PCA-based knowledge distillation**, which we motivate from PCA theory [3], to distill the most important knowledge for style representation from a source model to a smaller encoder. To demonstrate the generalizability of our distillation method, we apply it to two backbone architectures as the source models: VGG-19 and MobileNet [18]. (2) We integrate the blockwise decoder training of PhotoWCT2 [11] into our PCA-based knowledge distillation to concurrently implement a **smaller pairing decoder** which reproduces the encoder features to perform coarse-to-fine feature transformations. The resulting model is shown in Fig. 2(b). Experiments demonstrate that our smaller models reflect better style than WCT2 and PhotoNAS due to coarse-to-fine feature transformations, preserve better content than PhotoWCT and PhotoWCT2 due to knowledge distillation, and incur faster speeds. These benefits are exemplified in Fig. 1. To the best of our knowledge, our method is the first knowledge distillation algorithm for photorealistic style transfer.

2 Related Works

**Modern photorealistic style transfer models.** To overcome the slow speed of the pioneering neural network-based algorithm [30], most modern photorealistic style transfer models are based on autoencoders [10, 11, 24, 48]. Yet, as explained in Sec. 1, these models are still relatively slow due to their large sizes. Additionally, such models result in a poor balance between stylization strength and content preservation. For instance, PhotoWCT [24] realizes strong stylization strength by coarse-to-fine feature transformations, but suffers poor content preservation for two reasons: artifacts from strong stylization strength and missing high-frequency details due to its lossy structure. WCT2 [48] improves the content preservation from PhotoWCT by performing fine-to-coarse feature transformations to transfer weaker style effects in order to reduce artifacts and also introduces wavelet-based skip connections to reinforce the high-frequency component construction. However, sacrificing stylization strength means WCT2 weakly captures the style. PhotoWCT2 [11] combines the merits of PhotoWCT and WCT2: it introduces high-frequency residuals to improve the high-frequency detail construction from PhotoWCT and realizes coarse-to-fine feature transformations to maintain strong stylization strength. Yet, PhotoWCT2 still can generate artifacts, likely because of the coarse-to-fine feature transformations. Our experiments demonstrate that models distilled with our PCA knowledge distillation simultaneously address the aforementioned limitations of existing models: our models are smaller, faster, and achieve a better content-style balance.

**Knowledge distillation for vision tasks.** Given a neural network (teacher/source model) for a specific task, knowledge distillation (KD) aims to efficiently train a smaller network (student/target model) for the same task by leveraging the knowledge acquired in the source network.

Most KD algorithms address high-level vision tasks. For example, following the pioneering KD algorithm [17], most follow-up algorithms address image classification [12, 27, 31, 32, 34–36, 42, 44, 47, 49]. Others address object detection [6], semantic segmentation [28] and clothing matching [29]. Yet, these algorithms are not applicable to the

---

Footnote: An exception is a 2020 method [46] which learns to compute a pixel-wise linear mapping to map the content pixel values to the stylized pixel values. However, its code is not available.
low-level photorealistic style transfer task. For example, it is sufficient for prior work that the coarse features of two dog images from a late layer of a distilled image classifier to be similar such that the final classification layer can classify them to the same class (i.e., dog class). In contrast, photorealistic style transfer should preserve image content and so it requires that the features differ so that the different content in the dog images can be reconstructed.

The closest work to ours is collaborative knowledge distillation (CKD) [43] for artistic style transfer. However, CKD is different from ours in several respects. First, CKD is empirical without theoretical explanation. Second, with no theory behind, there is no guideline for target model size selection. Third, CKD is poor at preserving content and so is not applicable to photorealistic style transfer. We experimentally demonstrate the advantages of our PCA-based approach over CKD.

3. Method

We now describe our PCA-based algorithm for distilling lightweight, fast, high quality models for photorealistic style transfer. For simplicity, we describe it when used with the most popular backbone architecture for this task: VGG-19. However, to show the generalizability of our approach, we will also apply the algorithm to distill a model from a MobileNet-based source model in the experiments.

3.1. Background: how to represent style

It has been shown that a style representation is valid (i.e., can catch the style of an image) if the distance between style representations of the style image and the stylized image is a Maximum Mean Discrepancy [25]. Under this scheme, the covariance of an image feature $\mathbf{F}$ extracted at an intermediate layer of VGG-19 is one feasible style representation. Formally, by reshaping $\mathbf{F} \in C \times H \times W$ ($C$, $H$, $W$: channel length, height, and width of $\mathbf{F}$) to a 2D matrix of shape $C \times HW$, the covariance is calculated as $\bar{\mathbf{F}}\bar{\mathbf{F}}^T$, where $\bar{\mathbf{F}}$ is the centralized feature $\mathbf{F} - \mu(\mathbf{F})$, and $\mu(\mathbf{F})$ is the mean of $HW$ column vectors of $\mathbf{F}$. We leverage this style representation in our method. For brevity, in what follows, we exclude the coefficient of the covariance matrix.

Often multiple layers of a network (e.g. $\text{relu}_N$, $N = 1, 2, 3, 4$) are simultaneously used in order to extract a style representation that captures both the coarse and fine style of an image [15, 25]. We follow this approach in our work.

3.2. PCA-based knowledge distillation

Our PCA-based knowledge distillation algorithm consists of two parts, which we describe below: global eigenbasis derivation and blockwise PCA knowledge distillation.

Teacher/Source model. We consider VGG-19 from its input to the $\text{relu}_4$ layer as the source model which we name $\text{ENC}$. From $\text{ENC}$ we distill the style knowledge to a smaller target model which we name $\text{enc}$, whose structure follows the source model but uses a smaller channel length at each layer. More specifically, the style knowledge will be distilled from the $\text{relu}_N$ layers of $\text{ENC}$ to the counterpart layers in $\text{enc}$ which we call $\text{relu}_N\text{enc}$.

In the following, we let $\mathbf{F}_{N,k} \in C_N \times H_{N,k}W_{N,k}$ be the feature of the image $I_k$ extracted at the $\text{relu}_N\text{enc}$ layer of $\text{ENC}$ and $\mathbf{F}_{N,k}^t \in C_N \times H_{N,k}W_{N,k}$ ($C_N \ll C_N$) be the feature extracted at the counterpart $\text{relu}_N\text{enc}$ layer of $\text{enc}$. The above notations are summarized in Fig. 2.

**PCA for style knowledge distillation: image-dependent eigenbases.** We represent the style of a photograph $I_k$ using the covariance matrices of the features from the source/target model. Formally, in the distillation from $\text{relu}_N\text{enc}$ to $\text{relu}_N\text{enc}$, we want the covariance of the distilled feature $\bar{\mathbf{F}}_{N,k}^t(\bar{\mathbf{F}}_{N,k}^e)^T$ to capture the most important information in the covariance of the source feature $\bar{\mathbf{F}}_{N,k}^e\bar{\mathbf{F}}_{N,k}^t$ needed for stylization. Moreover, we want to do this distillation for multiple layers: distill $\text{relu}_N\text{enc}$ to the $\text{relu}_N\text{enc}$ layer for $N = 1, 2, 3, 4$.

This kind of problem is classical dimension reduction with PCA. In PCA, we treat $\bar{\mathbf{F}}_{N,k}$ as a collection of $H_{N,k}W_{N,k}$ numbers of $C_N$-dimensional data points. PCA seeks an orthonormal transformation $\mathbf{W}_{N,k} \in \mathbb{R}^{C_N \times C_N}$ to map these data points to a $C_N$-dimensional space, so that the covariance $\mathbf{W}_{N,k}\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T\mathbf{W}_{N,k}^T$ of the mapped feature $\mathbf{W}_{N,k}\bar{\mathbf{F}}_{N,k}$ preserves the essential information in the original covariance $\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T$. Mathematically, PCA solves the following optimization problem [2, 3],

$$\max_{\mathbf{W}_{N,k}} \quad \text{tr}(\mathbf{W}_{N,k}\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T\mathbf{W}_{N,k}^T),$$

where the constraint $\mathbf{W}_{N,k}\mathbf{W}_{N,k}^T = 1$ is the orthonormality property of $\mathbf{W}_{N,k}$.

Intuitively, we can think of the trace function as a metric to evaluate the information in a covariance matrix and we search an orthonormal basis to maximize the metric value. According to PCA, the $C_N$ row vectors of the solution $\mathbf{W}_{N,k}$ to Eq. (1) are the $C_N$ leading eigenvectors of the covariance $\bar{\mathbf{F}}_{N,k}\bar{\mathbf{F}}_{N,k}^T$. With $\mathbf{W}_{N,k}$, we train the target
model $\text{enc}$ to produce at the $\text{relu}N,I,e$ layer a feature $\mathbf{F}_{N,k}$ whose centralized form $\overline{\mathbf{F}}_{N,k}$ is $\mathbf{W}_{N,k}\overline{\mathbf{F}}_{N,k}$.

However, the image-dependent transformations $\mathbf{W}_{N,k}$’s for distillation are problematic, which can be mathematically explained as follows. First, the $\text{relu}N,I,e$ layer of the distilled model $\text{enc}$ defines a $C_N$-dimensional space $S_{C_N}$:

$$S_{C_N} = \{ f \in \mathbb{R}^{C_N} \mid f \text{ is a column vector of } \overline{\mathbf{F}}_{N,k}, \forall I_k \}. \quad (2)$$

Therefore, for any two images $I_{k_1}$ and $I_{k_2}$, the column vectors of $\overline{\mathbf{F}}_{N,k_1}$ and $\overline{\mathbf{F}}_{N,k_2}$ should be in the same space $S_{C_N}$. However, the orthonormal transformations $\mathbf{W}_{N,k_1}$ and $\mathbf{W}_{N,k_2}$ calculated from the covariances $\overline{\mathbf{F}}_{N,k_1} \overline{\mathbf{F}}_{N,k_1}^T$ and $\overline{\mathbf{F}}_{N,k_2} \overline{\mathbf{F}}_{N,k_2}^T$ may be different, so the column vectors of the mapped feature $\mathbf{W}_{N,k_1} \overline{\mathbf{F}}_{N,k_1}$ may reside in a different $C_N$-dimensional space than the column vectors of $\mathbf{W}_{N,k_2} \overline{\mathbf{F}}_{N,k_2}$, which conflicts with the former argument. Thus, if we distill with the image-dependent eigenbases $\mathbf{W}_{N,k}$’s, it results in a sub-optimal space that does not catch the style of any images well. We will show this in Sec. 4.3.

**PCA for style knowledge distillation: global eigenbases (i.e., image-independent).** We instead introduce a global, image-independent eigenbasis $\mathbf{W}_{N,g} \in \mathbb{R}^{C_N \times C_N}$. This definition defines a unique $C_N$-dimensional space that on average catches the style of different images well.

Our solution entails modifying Eq. (1) towards solving the following optimization problem for $\mathbf{W}_{N,g}$:

$$\min_{\mathbf{W}_{N,g}} \frac{1}{M} \sum_{k=1}^{M} \text{tr} (\mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k} \overline{\mathbf{F}}_{N,k}^T \mathbf{W}_{N,g}^T), \quad (3)$$

where $M$ represents the number of photographs in a virtually infinitely large image dataset, and the solution of $\mathbf{W}_{N,g}$ is the eigenbasis of $\frac{1}{M} \sum_{k=1}^{M} \overline{\mathbf{F}}_{N,k} \overline{\mathbf{F}}_{N,k}^T$, which is not analytically attainable. While Eq. (3) seems approximately solvable using mini-batch gradient descent to minimize the loss $- \sum_{k \in B_t} \text{tr} (\mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k} \overline{\mathbf{F}}_{N,k}^T \mathbf{W}_{N,g}^T) / |B_t|$, where $B_t$ is a batch of sampled images at the $t$-th iteration of gradient descent, the minus of a trace as a loss function makes the gradient descent process unstable. This is because such a loss has no lower bound and so a gradient descent algorithm is prone to keep minimizing the loss but ignore the constraint $\mathbf{W}_{N,g} \mathbf{W}_{N,g}^T = \mathbf{I}$.

To bypass it, we rewrite Eq. (3) as the equivalent form:

$$\min_{\mathbf{W}_{N,g}} \frac{1}{M} \sum_{k=1}^{M} \| \mathbf{W}_{N,g}^T \mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k} - \overline{\mathbf{F}}_{N,k} \|_2^2, \quad (4)$$

where $\mathbf{W}_{N,g}^T \mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k}$ is the reconstructed feature of $\overline{\mathbf{F}}_{N,k}$ from the mapped feature $\mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k}$. The meaning behind the equivalence is that the orthonormal basis $\mathbf{W}_{N,g}$ that on average maximizes the covariance information in the $C_N$-dimensional space should also on average minimize the reconstruction loss in the original $C_N$-dimensional space. Unlike the trace function in Eq. (3) which is unbounded, the L2-norm of Eq. (4) has a lower bound of zero and so Eq. (4) is approximately solvable with mini-batch gradient descent. Moreover, since $\mathbf{W}_{N,g}$’s ($N = 1, 2, 3, 4$ for $\text{relu}N,I,e$ layers) are independent, they can be solved together with the same batch of sampled images at an iteration of gradient descent.

To summarize, to derive $\mathbf{W}_{N,g}$’s ($N = 1, 2, 3, 4$), we use mini-batch gradient descent to solve Eq. (4) for four $\mathbf{W}_{N,g}$’s at once: At the $t$-th iteration of gradient descent, we sample a batch $B_t$ for the following minimization problem:

$$\min_{\mathbf{W}_{N,g}} \frac{1}{|B_t|} \sum_{N=1}^{4} \sum_{k \in B_t} \| \mathbf{W}_{N,g}^T \mathbf{W}_{N,g} \overline{\mathbf{F}}_{N,k} - \overline{\mathbf{F}}_{N,k} \|_2^2$$

and calculate the gradient of the objective in Eq. (5) to update $\mathbf{W}_{N,g}$’s. In our implementation, we use the MS-COCO dataset with random crop data augmentation, a batch size $|B_t|$ of 8, and train $\mathbf{W}_{N,g}$’s for five epochs. Fig. 3(a) exemplifies the derivation with a batch of one image.

With global eigenbases $\mathbf{W}_{N,g}$’s ($N = 1, 2, 3, 4$), we can distill style information from $\text{relu}N,I,e$ layers of the source model $\text{ENC}$ to $\text{relu}N,I,e$ layers of the target model $\text{enc}$.

**Blockwise PCA knowledge distillation.** To realize coarse-to-fine feature transformations in a distilled model, in addition to distilling the style information to the distilled encoder $\text{enc}$, our approach also must implement a pairing decoder $\text{dec}$ to reproduce encoder features. We now describe our distillation approach, which integrates PCA knowledge distillation with the blockwise training strategy for the decoder in PhotoWCT$^2$ [11] to achieve this.

A schematic diagram of our $\text{enc-dec}$ model is shown in Fig. 2. We split the encoder $\text{enc}$ into a series of blocks $\{\text{enc}_1, \text{enc}_2, \text{enc}_3, \text{enc}_4\}$ where the output of $\text{enc}_N$ is the $\text{relu}N,I,e$ layer, and the decoder $\text{dec}$ into a series of blocks $\{\text{dec}_3, \text{dec}_4, \text{dec}_2, \text{dec}_1\}$ where the output of $\text{dec}_N$ is the $\text{relu}(N-1),I,d$ layer which reproduces the $\text{relu}(N-1),I,d$ feature. That is, the decoder takes as input the $\text{relu}4,I,e$ feature from $\text{enc}$ to progressively reproduce $\text{relu}3,I,e$, $\text{relu}2,I,e$, and $\text{relu}1,I,e$ features and the reconstructed image. During stylization, ZCA feature transformations [9, 23] are placed at the $\text{relu}4,I,e$ layer and the $\text{relu}N,I,d$ layers ($N = 3, 2, 1$) to sequentially adapt the coarse $\text{relu}4,I,e$ content feature to the fine $\text{relu}1,I,d$ content feature with respect to the corresponding style features.

To implement our $\text{enc-dec}$ model, each pair of $\text{enc}_N$ and $\text{dec}_N$ are trained together with other pairs fixed by minimizing the encoder distillation loss $\mathcal{L}_{\text{enc}}^N$, and the decoder implementation loss $\mathcal{L}_{\text{dec}}^N$. The four pairs are trained sequentially.

---

\(^4\)Derivation of the equivalence is in the Supplementary Materials.
Figure 3. PCA-based knowledge distillation for photorealistic style transfer consists of two steps: (a) global eigenbasis derivation and (b) blockwise PCA knowledge distillation. As explained for Eq. (4), the $W_N$ that on average maximizes the style information in the distilled space also minimizes the feature reconstruction loss $\|\bar{F}_N - W_N^T W_N F_N\|_2^2$. With $W_N$, style knowledge is blockwisely distilled from the source model block $ENC_N$ to the target model block $enc_N$ in the order from $N = 1$ to $N = 4$ by minimizing the encoder distillation loss (Eq. (6)) and the decoder implementation loss (Eq. (7)).

In the encoder distillation, given the image $I_k$ we want to train the encoder block $enc_N$ to make its centralized output $\bar{F}_{N,k}$ close to the feature $W_{N,g} F_{N,k}$, which is mapped from the centralized output $F_{N,k}$ of $ENC_N$ by the global eigenbasis $W_{N,g}$. Instead of taking $\|\bar{F}_{N,k} - W_{N,g} F_{N,k}\|_2^2$ as the loss function, we observe the following feature reconstruction loss for encoder distillation converges better:

$$L^{enc}_N(I_k) = \|W_{N,g}^T \bar{F}_{N,k} - \bar{F}_{N,k}\|_2^2.$$  \hspace{1cm} (6)

This loss also effectively makes $enc_N$ produce $\bar{F}_{N,k}$ equal to $W_{N,g}\bar{F}_{N,k}$.\footnote{We prove this in the Supplementary Materials.} We suspect the better convergence is due to its stronger constraint: $L^{enc}_N(I_k)$ imposes $C_N H_{N,k} W_{N,k}$ scalar constraints, while $\|\bar{F}_{N,k} - W_{N,g} F_{N,k}\|_2^2$ imposes $C_N^{-1} H_{N,k} W_{N,k}$ scalar constraints, which are $C_N / C_N^2$ times (~10x) looser and might be casually fulfilled.

In the decoder implementation, given the image $I_k$ we want to make the output $F^d_{N-1,k}$ of $dec_N$ to reproduce the input $F^e_{N-1,k}$ of $enc_N$ and the reconstructed image $I_{k,rec}$ close to the input $I_k$. Collectively, we minimize $L^{dec}_N$ consisting of three terms:

$$L^{dec}_N(I_k) = \|F^d_{N-1,k} - F^e_{N-1,k}\|_2^2 + ||I_{k,rec} - I_k||_2^2 + ||F_{N,k,rec} - F_{N,k}||_2^2,$$ \hspace{1cm} (7)

where the third term is perceptual loss used to boost the image reconstruction. Note that when $N = 1$, there is no first term for feature reproduction.

To summarize, when training the pair of $enc_N$ and $dec_N$, we solve the following optimization problem:

$$\min_{enc_N, dec_N} L^{enc}_N(I_k) + L^{dec}_N(I_k).$$ \hspace{1cm} (8)

### 3.3. Channel lengths of the target model

We now show how we select the channel lengths $C_N^j$’s of the target model $enc$. Following the rule of thumb [5, 13, 14, 33, 37] in PCA dimension reduction to keep the most important information, the target layer $relu_{N,I}$ of the channel length $C_N^j$ should preserve 85% of the variance information from the source layer $relu_{N,I}$. The channel length that preserves this amount of information can be determined with a data-driven approach.

For each image $I_k$ in our dataset, we compute the covariance of its feature $F_{N,k}$ extracted at the $relu_{N,I}$ layer. Let $\sigma_{N,k}^j$ be the $j$-th largest eigenvalue of the covariance associated with the $j$-th principal eigenvector $e^j_{N,k}$. The $j$-th explained variance (EV) $\sigma_{N,k}^j / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha$ reflects the portion of the feature variance captured by $e^j_{N,k}$. Then the cumulative EV (CEV) $\sum_{\beta=1}^{C_N^j} \sigma_{N,k}^\beta / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha$ reflects the feature variance captured by the top $C_N^j$ eigenvectors. We use mean cumulative explained variance (mCEV) to determine the value of $C_N^j$. The mean CEV (mCEV) is the average of the CEVs across all images:

$$mCEV(C_N^j) = \frac{1}{M} \sum_{k=1}^{M} \frac{\sum_{\alpha=1}^{C_N^j} \sigma_{N,k}^\alpha}{\sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha} = \frac{C_N^j}{\beta=1} mEV(\beta).$$  \hspace{1cm} (9)

where $M$ is the number of images in the dataset and $mEV(\beta) = \frac{1}{\beta} \sum_{k=1}^{M} (\sigma_{N,k}^\beta / \sum_{\alpha=1}^{C_N} \sigma_{N,k}^\alpha)$ is the mean $\beta$-th
EV. We look for a $C_N^Y$ that satisfies $\text{mCEV}(C_N^Y) \approx 85\%$. We use the MS-COCO [26] training set of 118,287 images because of its large number of diversified images.

The mCEV and mEV for each $reluN$ layer of VGG-19 are shown in Fig. 4. We observe that 85% of the variance information in the original 512 (256, 128, 64)-dimensional space of the $relu4$ ($relu3$, $relu2$, $relu1$) layer can be explained by, on average, 64 (58, 20, 5) principal components. In other words, across the four layers, a small percentage of the original number of channels are needed to preserve 85% of the variance information; i.e., 12.5% (22.7%, 15.6%, 7.8%).

Experimentally, we found that setting $C_4^Y$ to 5 with $C_3^Y$ set to 20 hinders the distillation from the $relu2$ layer to the $relu1$ layer. We suspect it is due to the low reduction rate of 7.8% for the $relu1$ layer. We found experimentally that doubling the reduction rate and setting $C_4^Y$ to 10 overcomes this issue. For our final model, we set the four channel lengths ($C_1^Y$, $C_2^Y$, $C_3^Y$, $C_4^Y$) to (10, 20, 58, 64).

4. Experiments

We evaluate our models with respect to model size (Sec. 4.1), inference time at different image resolutions (Sec. 4.2), and quality of rendered images (Sec. 4.3).

4.1. Model size

Our models. We apply our PCA knowledge distillation to distill two models from VGG and MobileNet. We refer to these models as Ours-VGG and Ours-Mob, respectively.

To assess the benefit of using PCA to derive the channel lengths for a target model, we also evaluate against a PCA-distilled model using the channel lengths empirically selected for CKD [43]: $C_1^Y = 16$, $C_2^Y = 32$, $C_3^Y = 64$, and $C_4^Y = 128$. We call this variant Ours-VGG-CKD.

Following prior work [11], we apply high-frequency residuals (HFR) to all of our models to support good high-frequency detail construction.

Baselines. We compare to four models. One model is distilled with the only prior distillation method for style transfer: CKD [43]. We also evaluate three state-of-the-art non-distilled models: WCT$^2$ [48], PhotoWCT$^2$ [11], and PhotoWCT [24]. To support fair comparison, we apply HFR to reinforce the high-frequency detail construction for those models that lack this feature and so suffer lossy structure for high-frequency details (as discussed in Sec. 2): PhotoWCT [24] and the CKD model. We show in the Supplementary Materials that both variants achieve better content preservation than the original versions, with little impact to model size and speed.

Results. Results are reported in Tab. 1(a).

Compared to the existing models, our models have far fewer model parameters. For example, Ours-VGG uses only 2.8%, 3.4%, and 4.0% of the parameters of WCT$^2$, PhotoWCT-HFR, and PhotoWCT$^2$, respectively.

Given that MobileNet is a lightweight alternative to VGG-19, we observe that Ours-Mob uses fewer parameters than Ours-VGG: it uses only 0.7%, 0.9%, and 1.0% of the parameters of WCT$^2$, PhotoWCT-HFR, and PhotoWCT$^2$, respectively.

Compared to the baseline distilled model, CKD, Ours-VGG-CKD uses fewer parameters. We attribute that to the fact that Ours-VGG-CKD uses a single autoencoder while CKD uses a cascade of four autoencoders.

When comparing the sizes of models distilled using the channel lengths derived with PCA and those empirically selected for CKD [43], Ours-VGG uses 64.3% parameters of Ours-VGG-CKD. In other words, our theoretically motivated approach results in a more compact model.\footnote{We exclude a couple recent models from this experiments section for the following reasons. Due to the huge size, PhotoNAS [4] cannot handle the smallest considered image resolution (HD) in this paper and so is not used for comparison. LST [22] is another autoencoder-based method, which is designed for artistic style transfer and is insufficient for photorealistic style transfer, as we show in the Supplementary Materials.}

\footnote{Note that a certain amount of variance needs to be preserved so that the model compression does not result in an architecture with too little capacity to effectively learn. Through experimentation, we found the lower bound for the variance information percentage is around 75%, which results in 59K parameters with VGG as the source model. While the distilled model
4.2. Inference time

Intuitively, smaller models should enable faster stylization and support larger image resolutions. We demonstrate these benefits here. We conduct all experiments on an Nvidia RTX8000 GPU with 48GB memory. For completeness, we report CPU times in the Supplementary Materials.

Our models and baselines. We test the same models and baselines as used in Sec. 4.1.

Dataset. We test all models on six resolutions: 1280×720 (HD), 1920×1080 (Full HD), 2560×1440 (Quad HD), 3840×2160 (4K UHD), 5120×2880 (5K), and 7680×4320 (8K). We collect images for testing by downloading an 8K video [1] from YouTube, sampling a frame per second to collect 100 frames, and then downsampling each frame to the other lower resolutions.

Results. We report the speed of different models for stylizing images of different resolutions in Tab. 1.

Both our PCA-based models, Ours-VGG and Ours-Mob, run the fastest for all considered resolutions of the VGG-based models and of all models, respectively. Compared to the fastest non-distilled model PhotoWCT, Ours-VGG is consistently 5-6x faster in the resolutions supported by PhotoWCT, while Ours-Mob is 8x faster in HD image stylization and 12.6x faster in 4K and 5K image stylization. Compared to the slowest non-distilled model WCT, Ours-VGG is 7x faster in HD image stylization and 11x faster in FHD and QHD image stylization, while Ours-Mob is 9x, 16x, and 20x faster in HD, FHD, and QHD image stylization, respectively. Compared to the only baseline distilled model, CKD, Ours-VGG is consistently 2-2.5x faster in all resolutions while Ours-Mob can even run 5.3-5.5x faster in 4K, 5K and 8K resolutions.

Regarding supported resolutions, only distilled models can support the highest resolution tested: 8K. Moreover, WCT is not even able to support 5K resolutions. This finding highlights that distilling models to smaller sizes is critical for supporting higher resolution visual data.

4.3. Content preservation and stylization strength

We assess the quality of our models’ rendered images.

Our models. We again evaluate Ours-VGG, Ours-Mob, and Ours-VGG-CKD, which are described in Sec. 4.1. Additionally, to assess the benefit of using global eigenbases to distill our models, we also evaluate models distilled using image-dependent eigenbases: Ours-VGG-Local and Ours-Mob-Local, where Local denotes local eigenbases.

Baselines. We test the same baselines as used in Sec. 4.1.8

Metrics. We adopt the de facto standard metrics to measure content preservation and stylization strength, which were established in [15]. Specifically, given a stylized image $I_t$, we calculate its content loss $||\tilde{F}_{4,c} - \tilde{F}_{4,t}||^2_2$ from the content image $I_c$ and its style loss $\sum_{N=1}^{N_4}||\text{Cov}(F_{N,s}) - \text{Cov}(F_{N,t})||^2_2$ from the style image $I_s$.

We also assess the quality of an image $I_t$, using standard metrics employed in the image quality assessment community: SSIM [45], FSIM [50], and NIMA [41]. While the first two metrics measure the structure similarity between $I_t$ and $I_c$, the third metric evaluates $I_t$ as a standalone image.

Dataset. We use PST dataset [46], the largest publicly available dataset for evaluating photorealistic style transfer models. It consists of 786 pairs of content and style images, where the resolutions of images are between HD and FHD.

Results. Qualitative results are shown in Fig. 5 and quantitative results are shown in Fig. 6.

Overall, our models distilled with global eigenbases achieve the best balance of content preservation and stylization strength (Fig. 6(a)). This is evident when examining the two extremes for the content-style trade-off: WCT2 sets the upper bound for style loss since it achieves the weakest stylization strength while PhotoWCT-HFR sets the upper bound for content loss since it achieves the worst content preservation. We denote the region set by these two extremes in gray. Lighter gray denotes the region where the trade-off is a smaller amount of stylization strength/content from PhotoWCT-HFR/WCT2 for a larger gain in content preservation/stylization strength. All our distilled models

---

Table 1. Size and inference time of different models, with all distilled models highlighted in light gray. Note that all models except the separated Ours-Mob are distilled from VGG-19. Compared to existing models, our models are smaller and achieve higher speeds.

<table>
<thead>
<tr>
<th>Model</th>
<th>(a) Size</th>
<th>(b) Inference time</th>
<th>HD</th>
<th>FHD</th>
<th>QHD</th>
<th>4K</th>
<th>5K</th>
<th>8K</th>
</tr>
</thead>
<tbody>
<tr>
<td>WCT$^2$</td>
<td>10.12M</td>
<td>0.37</td>
<td>0.80</td>
<td>1.20</td>
<td>OOM</td>
<td>OOM</td>
<td>OOM</td>
<td>OOM</td>
</tr>
<tr>
<td>PhotoWCT-HFR</td>
<td>8.35M</td>
<td>0.56</td>
<td>0.84</td>
<td>1.27</td>
<td>2.53</td>
<td>4.36</td>
<td>OOM</td>
<td>OOM</td>
</tr>
<tr>
<td>PhotoWCT$^2$</td>
<td>7.05M</td>
<td>0.32</td>
<td>0.45</td>
<td>0.69</td>
<td>1.26</td>
<td>2.14</td>
<td>OOM</td>
<td>OOM</td>
</tr>
<tr>
<td>CKD</td>
<td>526K</td>
<td>0.09</td>
<td>0.16</td>
<td>0.25</td>
<td>0.53</td>
<td>0.93</td>
<td>2.09</td>
<td></td>
</tr>
<tr>
<td>Ours-VGG-CKD</td>
<td>440K</td>
<td>0.06</td>
<td>0.09</td>
<td>0.13</td>
<td>0.25</td>
<td>0.42</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>Ours-VGG</td>
<td>283K</td>
<td>0.05</td>
<td>0.07</td>
<td>0.11</td>
<td>0.22</td>
<td>0.38</td>
<td>0.82</td>
<td></td>
</tr>
<tr>
<td>Ours-Mob</td>
<td>73K</td>
<td>0.04</td>
<td>0.05</td>
<td>0.06</td>
<td>0.10</td>
<td>0.17</td>
<td>0.38</td>
<td></td>
</tr>
</tbody>
</table>

With 75% of the variance information marginally decreases the content loss from Ours-VGG (2.20e vs. 2.25e) and slightly increases the style loss (4.90e vs. 4.77e), the distillation of this model is unstable and relies on good initialization of parameter values.

---

8For completeness, we show in the Supplementary Materials that CKD’s image quality is not better when using our smaller channel lengths.
Figure 5. Our PCA-distilled models result in stronger style effects than WCT $^2$ and better photorealism than the other methods, achieving a better content-style balance. Please see more stylized images in Supplementary Materials.

Figure 6. (a) Content vs. style losses resulting from different models. Our models distilled with global eigenbases (Our-VGG-CKD, Our-VGG, Our-Mob) can trade-off a smaller amount of stylization strength/content from PhotoWCT/WCT $^2$ for a larger gain in content preservation/stylization strength. (b) Quality scores of stylized images resulting from different models. Our models consistently achieve higher scores.

(Ours-VGG, Ours-Mob, Ours-CKD) achieve a better balance between stylization strength and content preservation, with our most compact model (i.e., Ours-Mob) achieving the best balance. We suspect this is because PCA filters out marginal pattern information (e.g., cloud contours, ripples) that can cause artifacts while preserving the dominant color information. In contrast, we suspect that the high content loss of CKD reflects that CKD poorly filters out the non-essential pattern information, as shown in Fig. 5.

Compared to Ours-VGG-CKD, Ours-VGG further improves the content preservation by trimming extra style information that results in slight artifacts $^9$, which the 9.1% lower style loss of Ours-VGG-CKD mostly contributes to. This highlights that, with the guidance from the PCA theory, we can properly select the channel lengths for our models to achieve a better content-style balance than when empirically selecting from trials.

The benefit of our PCA distillation strategy is validated by comparing models based on local eigenbases (i.e., Ours-VGG-Local and Ours-Mob-Local) to our models based on global eigenbases. We observe a stronger stylization strength in our models based on global eigenbases (qualitative comparisons are shown in Supplementary Materials).

Overall, we observe similar trends to those noted above with the quality metrics (Fig. 6(b)). Our models outperform existing distilled and non-distilled models.

4.4. Limitation

While the primary benefit of our work is to demonstrate that we can create ultra-compact models that run very fast, our work also highlight that stylizing images of higher resolutions is now in reach (i.e., 4K resolutions and beyond). However, we show in the Supplementary Materials that our models occasionally result in artifacts for the larger resolution images, since they contain more high-frequency details which are hotbeds where artifacts form. Future work should establish a new benchmark dataset for photorealistic stylization that includes larger resolution images.

5. Conclusion

We propose the first knowledge distillation for photorealistic style transfer, which is motivated by PCA theory. Compared to existing distilled and non-distilled models, our models are smaller, faster, and achieve a better balance between content preservation and stylization strength.

$^9$The slight artifacts are not clear in the small resolutions in Fig. 5.

We show stylized images of 4K+ resolutions in Supplementary Materials where the slight artifacts become apparent.
References

[29] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen. Structured knowledge distillation for dense predic-


Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In International Conference on Machine Learning, pages 5142–5151. PMLR, 2019. 2


