Color Transfer with Modulated Flows

Maria Larchenko, Alexander Lobashev, Dmitry Guskov, Vladimir Vladimirovich Palyulin

Skolkovo Institute of Science and Technology, Moscow 121205, Russia
mariia.larchenko@gmail.com, lobashevalexander@gmail.com

Abstract

In this work, we introduce Modulated Flows (ModFlows), a novel approach for color transfer between images based on rectified flows. The primary goal of the color transfer is to adjust the colors of a target image to match the color distribution of a reference image. Our technique is based on optimal transport and executes color transfer as an invertible transformation within the RGB color space. The ModFlows utilizes the bijective property of flows, enabling us to introduce a common intermediate color distribution and build a dataset of rectified flows. We train an encoder on this dataset to predict the weights of a rectified model for new images. After training on a set of optimal transport plans, our approach can generate plans for new pairs of distributions without additional fine-tuning. We additionally show that the trained encoder provides an image embedding, associated only with its color style. The presented method is capable of processing 4K images and achieves the state-of-the-art performance in terms of content and style similarity.

Code — https://github.com/maria-larchenko/modflows

Introduction

Color adjustment is one of the most commonly used image editing operations. While minor corrections can often be made quickly, achieving a precise color palette typically requires more time and attention to details.

Classical Methods. The idea to modify an image using features of another image appeared in the early 2000s (Jacobs et al. 2001). Soon the problem of *example-based color transfer* was formulated in the following way (Reinhard et al. 2001). A pair of images known as "content" and "style" is introduced. The aim of the transfer is to alter the colors of the content image to fit the colors of the style image without visible distortions and artifacts.

The pioneering works on the color transfer have already considered it as a problem of optimal transport (Morovic and Sun 2003). For instance, one would prefer to keep the shades of red as close to each other as possible. Technically, one defines a distance in the color space and tries to fit the desired color distribution with a minimal effort. This effort can be seen as a transportation cost, i.e. the problem can be formulated within the framework of optimal transport (OT) theory.

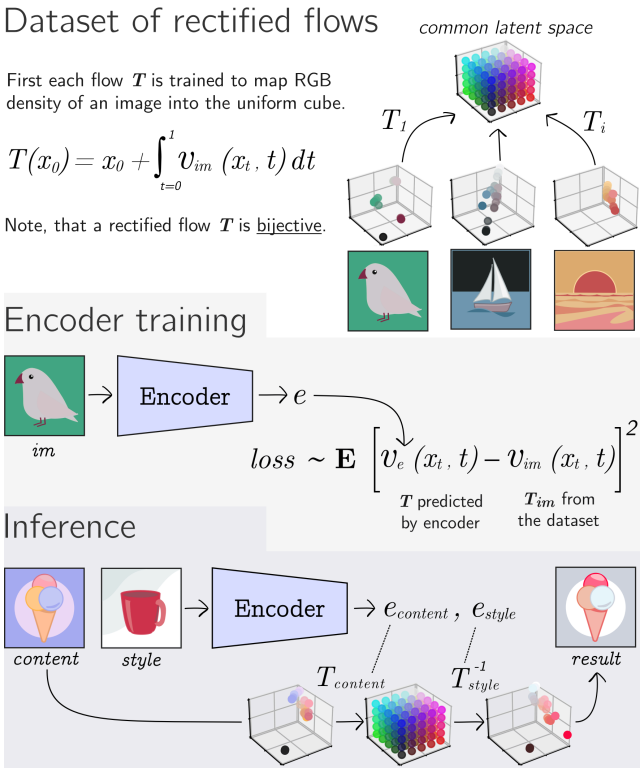

Figure 1: A proposed scheme of training and inference. Color transfer is a composition of a forward content and an inverse style flows applied to the content image.

In general case, an exact solution of OT problem is hard to obtain.Discretization of distributions allowed Morovic and Sun (2003) to employ optimal histogram matching, but explicit calculation of the transport cost still was computationally heavy; for this reason other histogram-based approaches dropped the optimality constraint and considered the simpler mass preserving transport problem (Neumann and Neumann 2005; Pitie, Kokaram, and Dahyot 2005).

Pitié and Kokaram (2007) first switched to a continuous formulation of OT problem in color transfer. Under several

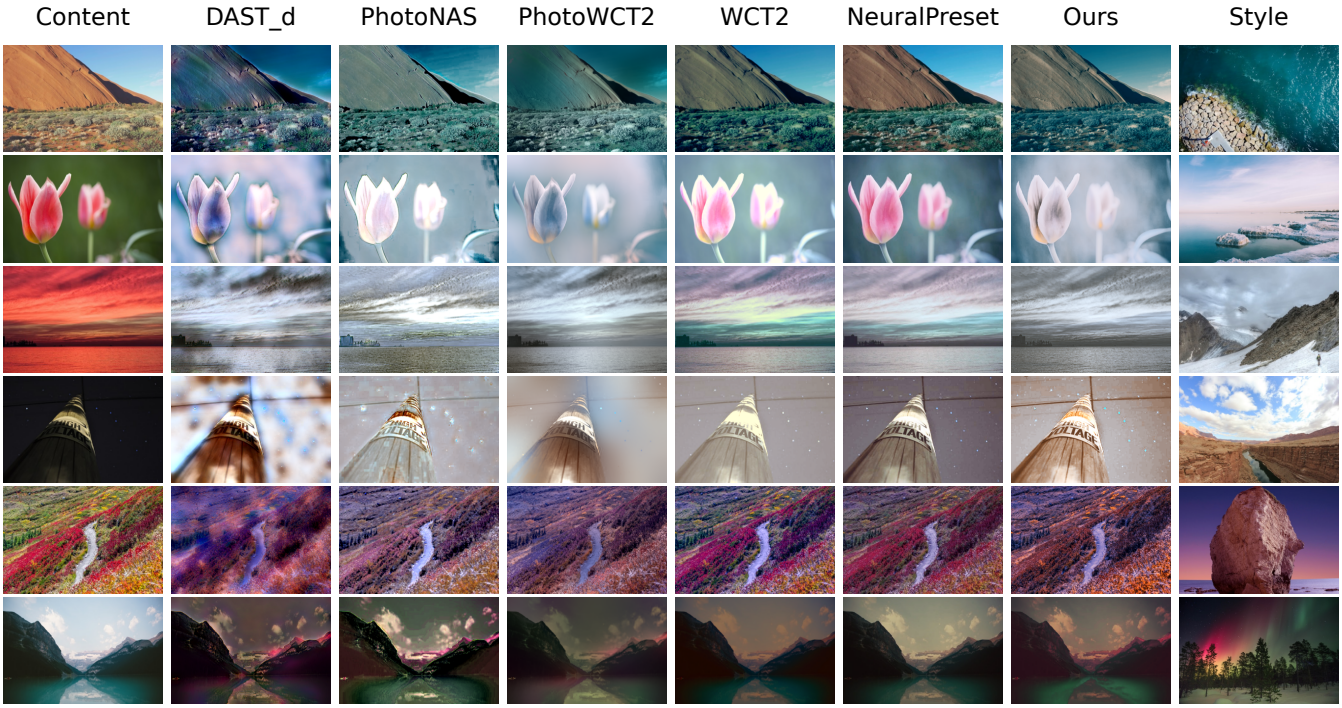

Figure 2: Qualitative comparison. Examples from Unsplash Lite test set. Our model achieves the most exact match with the reference palette without visible distortion.

simplifications (e.g. that color distributions are Gaussian), authors proposed Monge-Kantorovitch Linear (MKL) algorithm, which is still a strong competitor (Mahmoud 2023).

Neural Methods. Gatys, Ecker, and Bethge (2016) turned the research in a different direction, adapting deep convolutional neural networks (CNNs) for a high-level style extraction. The algorithm referred as Neural Style (Johnson 2015) could perform color transfer when applied to a pair of photos. However, the transfer was not ideal. It targeted a painting technique and textures, blending into stylized photos a reference color palette together with unwanted patterns.

The ability of deep CNNs to separate a color style from a content has inspired follow-up studies, primarily focusing on artifact removal. This has resulted in a series of algorithms such as DPST (Luan et al. 2017), WCT (Li et al. 2017), PhotoWCT (Li et al. 2018), WCT2 (Yoo et al. 2019), PhotoNAS (An et al. 2020), PhotoWCT2 (Chiu and Gurari 2022), DAST (Hong et al. 2021) and Deep Preset (Ho and Zhou 2021). The last algorithm, aimed at automatic retouching, achieves high quality in terms of the absence of artifacts but it does not suit the color transfer task well. Nevertheless, we have included the Deep Preset in comparison to give reference scores for image retouching.

Two of the most recent studies are closely related to our work. The first one is Sparse Dictionaries (Huang et al. 2023), the method based on discrete optimal transport applied to learned style dictionaries. The algorithm is reported to be rather slow compared to other methods and its code is unavailable at the moment.

The second method is the Neural Preset approach proposed by Ke et al. (2023). It executes *color style transfer* in RGB space using a multilayer perceptron, with its hidden weights predicted by an encoder network. It achieves impressive visual quality and is capable of processing of high-resolution images. The results for Neural Preset were obtained via the officially distributed application since the training code and model were not released. Due to the test set containing over a thousand images, we included Neural Preset only in the qualitative comparison.

However, we dedicated significant effort to reproducing this method. We believe that training of Neural Preset heavily depends on the random color filter adjustment strategy. In particular, the authors reported using 5000 Look-Up Table (LUT) filters, which are not publicly available. These LUT filters are designed by domain experts, and acquiring such a large number of them proved to be challenging. As a workaround, we used random monotone color filters for augmentation (Lobashev 2024). While our re-implementation successfully avoids visual artifacts, it only slightly conveys the reference visual style and has minimal impact on the color distribution. Therefore, we see the main strength of the original Neural Preset in its ability to capture the effects of hand-crafted LUT filters and we treat them as an essential part of the dataset, which could not be fully replicated by random color perturbations alone.

In order to address these limitations, we aimed on developing a model that could be trained without additional LUT filters, could be quickly applied to new images and considers the color transfer problem from the optimal transport point of view. To this end, we utilize rectified flows with parame-

ters, predicted by an encoder network. In order to simplify the training process, we introduce a uniform latent (or intermediate) space. The rectified flows transport the color distribution of a given image to the latent space. Upon application of a particular style, we use the inverse rectified flow to transfer color distribution back from the uniform distribution to target distribution of the style image.

Our contribution. The contribution of this paper can be summarized as follows:

- We present a novel method for color transfer based on rectified flows and a shared latent distribution. After training on a set of optimal transport plans, our approach can generate plans for new pairs of distributions without requiring additional training.
- We produce the dataset of 5896 flow-image pairs and train the generalizing encoder model.
- We show that the encoder-predicted vector of weights is an image embedding associated with its palette.

Background

Problem setting

In RGB space an image can be associated with a continuous 3-dimensional probability density function. We denote the density functions as π_0 for a content image and as π_1 for a style one. Here the random variables $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ represent pixels taken from the correspondent images. The color transfer problem may be defined as finding **a deterministic transport map** $T(X_0) = X_1$, where $T : \mathbb{R}^D \to \mathbb{R}^D$ is a change of variables, i.e.

$$\pi_0(x) = \pi_1(T(x)) \left|\det J_T(x)\right|, \tag{1}$$

where $J_T(x)$ is the Jacobian of T taken at point x.

Monge's optimal transportation. By introducing a cost function $c : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$, one arrives to a minimization problem. For instance, the quadratic cost function $c(x, y) = \|x - y\|^2$ gives a total expected cost of a transport map T

$$\text{Cost}\,[T] = \mathbb{E}\big(\,\|X_1 - X_0\|^2\,\big) = \int_{\mathcal{X}_0} (T(x) - x)^2 \pi_0(x) dx. \tag{2}$$

Finding of the optimal deterministic map T^* that minimizes the $\text{Cost}\,[T]$ for a fixed cost function is called Monge problem. It does not always have a solution. However, the quadratic cost function and the continuous density functions π_0, π_1 with finite second moments guarantee that a solution always exists and it is unique (Villani et al. 2009). In some cases T can be obtained explicitly. For monochrome images $X_0, X_1 \in \mathbb{R}$ and monotonically increasing cumulative distribution functions F_0, F_1 the optimal transport map $T(x)$ reads

$$T(x) = F_1^{-1}\big(F_0(x)\big). \tag{3}$$

In practice it is possible to construct $T(x)$ even when F_1, F_2 do not have an inverse (Neumann and Neumann 2005). Below we make the use of this fact by proposing a new content metric, a normalized gray-scale image.

Another important case for a known $T^* : \mathbb{R}^D \to \mathbb{R}^D$ is matching of two multivariate Gaussian distributions. Mentioned earlier MKL by Pitié and Kokaram (2007) relies on the Gaussian approximations and this result.

Monge–Kantorovich formulation. A correspondence between $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ can be non-deterministic. Instead of transport mapping T one could consider a **transport plan** $\pi(X_0, X_1)$ (also called a coupling), a joint probability distribution with marginals π_0 and π_1,

$$\int_{\mathcal{X}_0} \pi(x, y) dx = \pi_1(y), \quad \int_{\mathcal{X}_1} \pi(x, y) dy = \pi_0(x). \tag{4}$$

An example of a transport plan that always exists is a trivial coupling $\pi = \pi_0 \times \pi_1$, a plan where initial and target random variables are independent.

Monge–Kantorovich problem is to find $\pi^*(X_0, X_1)$ that minimizes the expected cost

$$\text{Cost}\,[\pi] = \mathbb{E}\big(c(X_0, X_1)\big) = \int_{\mathcal{X}_0 \times \mathcal{X}_1} c(x, y)\pi(x, y) dx dy. \tag{5}$$

Let $\Pi(\pi_0, \pi_1)$ be all possible couplings of π_0 and π_1. Then the **optimal transport cost** between the initial and target distributions is

$$\text{C}(\pi_0, \pi_1) = \inf_{\pi \in \Pi(\pi_0, \pi_1)} \int_{\mathcal{X}_0 \times \mathcal{X}_1} c(x, y) d\pi(x, y). \tag{6}$$

The optimal transport cost is tightly connected with the **Wasserstein distance** between two distributions. Note that the equation above is written for an unspecified cost function, i.e. the axioms of distance are not satisfied. By replacing a cost $c(x, y)$ with a proper distance function $d(x, y)$ (the quadratic cost suits this purpose) one gets a Wasserstein distance of order one

$$\text{W}(\pi_0, \pi_1) = \inf_{\pi \in \Pi(\pi_0, \pi_1)} \int_{\mathcal{X}_0 \times \mathcal{X}_1} d(x, y) d\pi(x, y). \tag{7}$$

Rectified flows

The optimal transport problem can be approximately solved by Rectified flows (Liu, Gong, and Liu 2022). Its key idea is in converting an arbitrary initial coupling into a deterministic transport plan. The new transport plan guarantees to yield no larger transport cost than initial one simultaneously for all convex cost functions. First, the independent pairs (X_0, X_1) from the trivial transport plan are sampled

$$\pi_{\text{trivial}}(X_0, X_1) = \pi_0(X_0) \times \pi_1(X_1). \tag{8}$$

Secondly, a linear interpolation between the initial and target samples is introduced by setting$X_t = tX_1 + (1-t)X_0$. With this, one trains a neural network $v_\theta(X_t, t)$ to minimize the loss

$$\min_\theta \int_{t=0}^{1} \mathbb{E}_{(X_0, X_1) \sim \pi_{\text{trivial}}} \left[\|X_1 - X_0 - v_\theta(X_t, t)\|^2\right] \text{d}t. \tag{9}$$

Given a trained rectified flow one can transport samples from the initial distribution π_0 to the target distribution π_1 in a deterministic way by numerically solving the ordinary differential equation (ODE)

$$\frac{dZ_t}{dt} = v_\theta(Z_t, t) \tag{10}$$

Algorithm 1: Encoder training

Require: trained image-flow pairs $(\mathcal{I}, \theta)$

repeat
  get batch $\boldsymbol{\mathcal{I}} = \{\mathcal{I}\}_i^N$, $\boldsymbol{\theta} = \{\theta\}_i^N$
  for $i = 1, \dots N$ **do**
    sample $X \sim \mathcal{I}$
    $Z = \mathrm{T}_\theta(X)$
    collect $t \sim \mathrm{Uniform}\,[0, 1]$
    collect $Z_t = tZ + (1-t)X$
    collect $v_t = v_\theta(Z_t, t)$
  end for
  Randomly reflect and rotate $\mathcal{I} \in \boldsymbol{\mathcal{I}}$
  $\boldsymbol{e} = \mathrm{Enc}(\boldsymbol{\mathcal{I}})$
  $\boldsymbol{t} = \{t\}_i^N$, $\boldsymbol{Z_t} = \{Z_t\}_i^N$, $\boldsymbol{v_t} = \{v_t\}_i^N$
  Apply $\boldsymbol{e}$ as parameters for ModFlow to get $\boldsymbol{v_e}(\boldsymbol{Z_t}, \boldsymbol{t})$
  Take gradient step with respect to Enc weights on $\nabla\, \mathbb{E}\left[\|\boldsymbol{v_t} - \boldsymbol{v_e}(\boldsymbol{Z_t}, \boldsymbol{t})\|^2\right]$
until converged

for $t \in [0, 1]$ with $Z_0 \sim \pi_0$. Thus, for this particular case the deterministic transport map reads

$$T_{\text{1-rectified}}(Z_0) = Z_0 + \int_{t=0}^{1} v_\theta(Z_t, t)dt. \tag{11}$$

The deterministic transport map $T_{\text{1-rectified}}$ gives rise to the deterministic transport plan $\pi_{\text{1-rectified}}$,

$$\pi_{\text{1-rectified}}(X_0, X_1) = \pi_0(X_0) \times \delta(X_1 - T_{\text{1-rectified}}(X_0)). \tag{12}$$

This transport plan has a much lower transport cost than the naïve transport plan $\pi_{\text{trivial}}(X_0, X_1)$.

Method

Our method is inspired by the increasing rearrangement coupling (Villani et al. 2009) given by Eq. 3. The transfer task is complicated as we want the model to generalize well across all possible pairs (π_i, π_j) of color distributions. However, having the opportunity to learn bijective mappings, one could greatly simplify the task by introducing a common intermediate distribution U.

The distribution U is implicitly present in the increasing rearrangement, such that for any random variable $X \sim \pi, X \in \mathbb{R}$ having monotonically increasing CDF

$$\begin{gathered} F(x) = \int_{-\infty}^{x} d\pi(y) \quad \text{it holds that} \\ U = F(X) \sim \mathrm{Uniform}\,[0, 1]\,. \end{gathered} \tag{13}$$

Therefore, for a pair of such random variables $X_i, X_j \in \mathbb{R}$ a composition $T = F_j^{-1} \circ F_i$ is a transport plan that traverses through a $\mathrm{Uniform}\,[0, 1]$ distribution.

We are extending this idea to random variables $X_i \in \mathbb{R}^D$ by learning bijective mappings $T_i : \mathbb{R}^D \to \mathbb{R}^D$ such that $T_i(X_i) = U^D$, where U^D is random variable in $\mathbb{R}^D$ with all components uniformly distributed in $[0, 1]$. For any pair X_i, X_j we define $T(X_i) = X_j$ as $T = T_j^{-1} \circ T_i$

Here rectified flow offers three important benefits. Firstly, as a solution of ordinary differential equation 9 it is bijective. Secondly, it keeps the marginal distributions close to the desired ones. Lastly, the rectification step allows us to substantially increase the inference speed without adding the transport cost. Thus, we are able to efficiently compute T as a composition.

During the experiments we observed that lightweight shallow models with a number of trained parameters ranging from approximately 500 to 10,000 could work as color transfer flows. The number of parameters lies in the same range with an output vector length of encoders so one may hope to use the output vector as flow parametrization, thus generalizing the approach.

The proposed method consists of two stages:

1. Produce a dataset of flow-image pairs, where flows' weights θ_i are trained to map a color distribution X_i of an image $\mathcal{I}_i$ into the uniform cube U. We follow (Liu, Gong, and Liu 2022) with an interpolation $X_t = t\,U + (1-t)X_i$

$$\min_{\theta_i} \int_{t=0}^{1} \mathbb{E}_{(U, X_i) \sim \pi_{\text{trivial}}} \left[\|U - X_i - v_{\theta_i}(X_t, t)\|^2\right] dt. \tag{14}$$

2. Train the encoder on batches from the dataset, such that the output vector $\mathrm{Enc}(\mathcal{I}_i) = e_i$ is a flow parametrization for an image $\mathcal{I}_i$.

Note, that the second stage does not include any distances $d(\theta, e)$. A flow parameterized by the encoder (or the **modulated flow**) is not obliged to have the same architecture as models in a dataset. We train the encoder using the loss function that allows a distillation

$$\min_{\mathrm{Enc}} \int_{t=0}^{1} \mathbb{E}_{(Z_i, X_i) \sim \pi_{\text{1-rectified}}} \left[\|Z_i - X_i - v_{e_i}(Z_{it}, t)\|^2\right] dt, \tag{15}$$

where $\mathrm{Enc}(\mathcal{I}_i) = e_i$ and target Z_i is generated from a X_i by trained flow θ_i

$$Z_i = T_{\text{1-rectified}}(X_i) = X_i + \int_{t=0}^{1} v_{\theta_i}(Z_t, t)dt \tag{16}$$

and Z_{it} are points sampled from an interpolation line connecting original X_i with its target Z_i

$$Z_{it} = tZ_i + (1-t)X_i. \tag{17}$$

The predicted velocity $v_{e_i}(\cdot, t)$ is given by the modulated flow with e_i weights. Generally, it is not advised to take the dimension of e much higher than the bottleneck of selected encoder.

Algorithm 1 provides the pseudo-code for the proposed method of training modulated flows. The term "modulated" refers to the fact that the weights of a flow at inference are produced (or modulated) by the encoder. It is important to note that the original rectified flow approach requires retraining for each new pair of densities, whereas modulation eliminates the need for this process. As demonstrated in the ablation study, using a generalizing model such as the encoder slightly increases the Wasserstein distance from the target distribution. However, it also provides implicit regularization, reducing the average Lipschitz constant of the modulated flows compared to rectified flows trained from scratch (see Table 3), which results in fewer visual artifacts.

Aggregated scores (DISTS)↓				Style distance↓	
Algorithm	Grayscale	Depth	Edge (Xie and Tu 2015)	Algorithm	mean ± std of mean
ModFlows (ours)	**0.129**	**0.217**	0.220	DAST_d	**0.112 ± 0.001**
MKL	0.146	0.227	0.224	ModFlows (ours)	0.123 ± 0.001
CT	0.169	0.234	0.232	DAST_da	0.127 ± 0.001
WCT2	0.170	0.228	0.249	PhotoWCT2	0.129 ± 0.001
PhotoWCT2	0.191	0.236	**0.217**	MKL	0.145 ± 0.001
DAST_d (vanilla)	0.204	0.267	0.224	WCT2	0.163 ± 0.001
DAST_da (adversarial)	0.214	0.282	0.229	CT	0.166 ± 0.001
PhotoNAS	0.224	0.276	0.270	PhotoNAS	0.183 ± 0.002
NeuralPreset*	0.349	0.366	0.360	NeuralPreset*	0.348 ± 0.003
Deep Preset	0.384	0.400	0.387	Deep Preset	0.384 ± 0.004

Table 1: Comparison of algorithms. Please note that NeuralPreset* is our re-implementation.

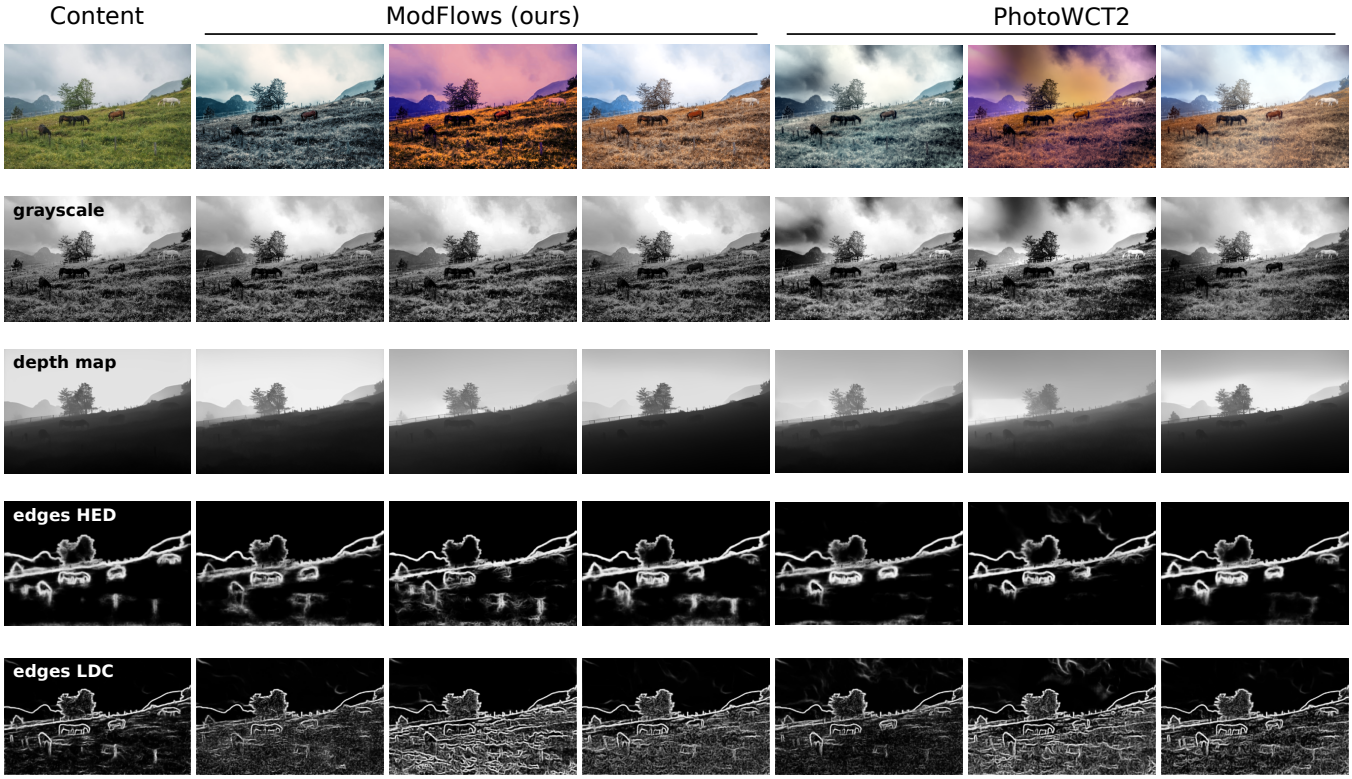

Figure 3: Colorless content metrics. The choice of the best content metric is not obvious. Edges detection by HED model (Xie and Tu 2015) grasps mostly the main objects of a scene, while canny LDC (Soria, Pomboza-Junez, and Sappa 2022) images are capturing the too detailed edges. Both of them are not sensitive to low-frequency artifacts. To show the absence of such artifacts in the Modflows we additionally compute similarity scores between the normalized grayscale images, which are processed to have a linear intensity histogram through histogram matching, and the depth maps (Gui et al. 2024).

Aggregated ablation scores (DISTS)↓				
Algorithm	Grayscale	Depth (Gui et al. 2024)	Edge (Xie and Tu 2015)	Style Distance↓
ModFlows (B6), $dim(e) = 8195$	**0.129**	**0.217**	**0.220**	0.123
Rectified flows (8195)	0.137	0.250	0.235	**0.114**
ModFlows (B0), $dim(e) = 515$	0.145	**0.217**	**0.220**	0.141

Table 2: Ablation study

Experiments and Metrics

Dataset. To implement the approach described above one needs a dataset of images with sufficiently diverse color distributions and resolutions. To achieve this diversity we construct our dataset by combining DIV2K (Ignatov, Timofte et al. 2019) and CLIC2020 (Toderici et al. 2020) (designed for image compression challenges) with a subset of "laion-art-en-colorcanny" (Ghoskno 2023). The total number of images is 5,826.

For every image we train a small two-layer MLP with 1024 hidden units (8195 parameters in total) and tanh activation, storing in the dataset 5,826 rectified models. Generation of a model-image pair takes approximately 100k iterations with lr = 5e-4.

Encoder. EfficientNet B6 is used as an encoder model (Tan and Le 2019). For simplicity we set the output dimension to 8195 for it to be the same with the dataset of trained flows. The encoder was trained with Adam optimiser (Kingma and Ba 2014) for 751k iterations with the batch size equals to 8 images. We decreased the learning rate from lr = 5e-4 to lr = 1e-4 after the first 100k iterations.

Test set. Tests were conducted on 1891 content-style pairs selected from Unsplash Lite 1.2.2 (Unsplash 2023). Searches were run on 25,000 Unsplash pictures. Our pictures are generated in 8 steps of ODE solver (16 steps in total for forward and inverse passes).

Style metric. The seminal work (Gatys, Ecker, and Bethge 2016) defines style loss as a distance between Gram matrices of feature maps, taken from convolutional layers of VGG encoder. Despite being capable of extracting a palette, this approach cannot reliably separate a palette from textures. Monge's problem (Eqs. 1 and 2) offers a more precise setting and a straightforward metric, namely, Wasserstein distance, Eq. 7. Therefore we estimate the Wasserstein distance between resulting and reference color distributions taking 6,000 pixel samples for a style metric (Bonneel et al. 2011; Flamary et al. 2021).

Content metric. Contrary to the style, a content metric is not uniquely defined. To measure the amount of visible artifacts we compute a set of colorless metrics based on depth-maps by recently released DepthFM (Gui et al. 2024), normalized grayscale pictures and edge-maps by HED (Xie and Tu 2015; Niklaus 2018) and LDC (Soria, Pomboza-Junez, and Sappa 2022) models. The variants of the colorless representation are demonstrated in Fig. 3. The difference between colorless images is evaluated with DISTS[1] (Ding et al. 2020) producing the content score.

Lipschitz constant. To estimate the regularity of learned color transfer maps we estimate their average Lipschitz constant, Table 3. It could be observed that rectified flow trained from scratch for a given pair of color distributions is more sensitive to input variations than, for instance, MKL and CT, meaning higher amount of visual artifacts. Low value of the Lipschitz constant for ModFlows encoder in comparison to the direct flows demonstrate regularizing effect of our training procedure.

[1]DISTS implementation is taken from "piq" library (Kastryulin, Zakirov, and Prokopenko 2019)

Method	Average Lipschitz Constant
ModFlows (ours)	**37.26 ± 0.79**
CT	51.90 ± 1.03
MKL	55.67 ± 1.19
Rectified flows	91.34 ± 1.26
DAST_d	121.17 ± 1.76
PhotoWCT2	160.12 ± 1.92

Table 3: Average Lipschitz constant of the color transfer map for different methods. Low value of the Lipschitz constant for ModFlows encoder in comparison to the direct flows demonstrate regularizing effect of our training procedure.

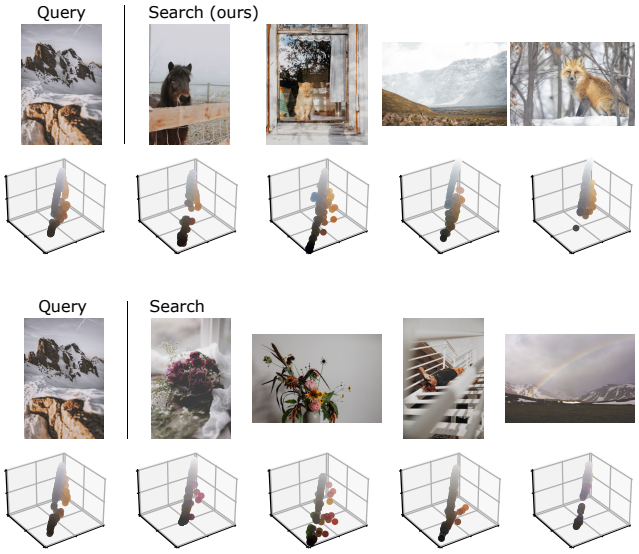

Figure 4: Search for similar color styles in the Unsplash Lite dataset (25k images). The top and second rows show search results based on the output of the ModFlows (B6) model. The third and last rows display results based on image statistics, specifically flattened vectors representing the first and second centered moments of the color distribution.

Comparison with baselines. Table 1 contains average style distances and aggregated scores for compared methods. Please note that NeuralPreset* is our re-implementation of the original work by Ke et al. (2023). It was trained on the same dataset as our method, but the LUT filters were replaced with random color perturbations (Lobashev 2024) since the original color filters and model are not available.

The **aggregated score** is calculated as a distance to the ideal point p, similarly with Ke et al. (2023),

$$\text{aggr. score} = \sqrt{(\text{p} - \text{style score})^2 + (\text{p} - \text{content score})^2} \tag{18}$$

Search of similar color styles. Once trained, the output vector of parameters e could serve as an embedding of a palette. To evaluate its expressive ability we compare e against standard statistics for RGB channels $(\boldsymbol{\mu}, \Sigma)$, that is, the vector of mean values concatenated with flattened covariance matrix. An example of a search is given in Fig. 4.

Ablation Study

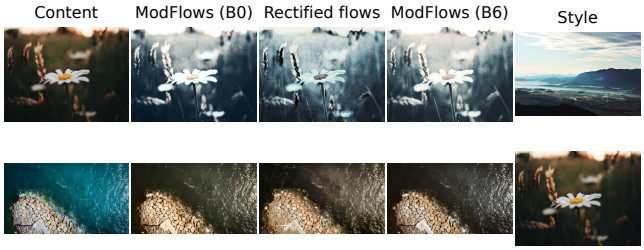

Figure 5: Ablation study. ModFlows models reach a better trade-off between style and content similarity when compared to dataset models used in their training.

All comparisons in this section are computed on a test set described above. We describe qualitatively and numerically the performance of

1. Transfers made with rectified flows (8195 parameters) through the uniform intermediate space.
2. Model based on EfficientNet B6 with output $dim(e)$ = 8195 trained on 5,826 flows-8195 from the main dataset.
3. Model based on EfficientNet B0 with output $dim(e)$ = 515 trained on 4,767 rectified flows (515 parameters) from the laion-art-en-colorcanny.

As the Table 2 proves, the low style distance in transfers made with rectified flows comes with artifacts which are detected by all content metrics, which is shown in Fig. 5. At the same time the generalization done by the ModFlows models reaches a better trade-off between style and content similarity. As expected, providing larger and more diverse dataset along with increased number of parameters results in a better performance.

From our experiments it follows that choosing another color space such as LAB or OkLAB (Ruderman, Cronin, and Chiao 1998) doesn't significantly improve the results. Despite these spaces offers better perceptual distance, they additionally complicate a training procedure, namely, a shape of a suitable shared latent space and the sampling process.

Limitations and Algorithm Tuning

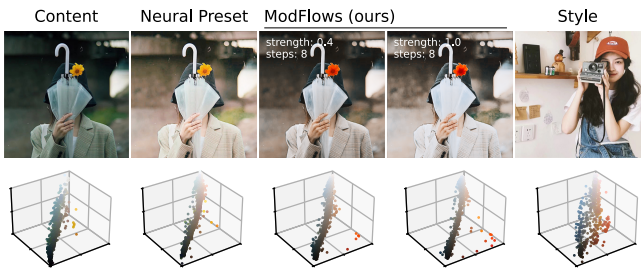

Figure 6: Limitations and algorithm tuning. An example of unintended color switching in two pictures generated with fixed number of steps for ODE solver (steps) and varied percent of interpolation curve passed (strength).

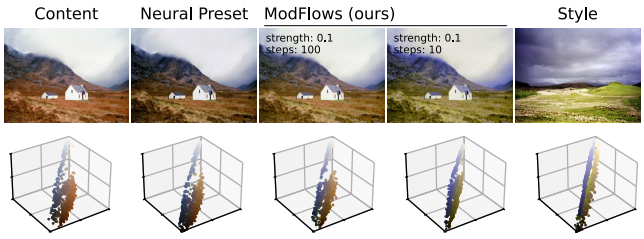

Figure 7: Algorithm tuning. Variation of a number of steps for ODE solver (steps) and a percent of interpolation curve passed (strength) results in different amount of changes for a distribution. In this example, increasing the strength or decreasing the number of steps further leads to the appearance of artifacts.

The framework of the transport theory gives us an opportunity to design an unsupervised algorithm. In the same time it introduces a limitation, that is a greater dependence of the result on the reference image. For example, the method may perform unintended color replacements, such as transforming yellow shades into red ones, Fig. 6. Our method does not provide control over the color of individual objects, as it operates in RGB space without considering semantic information.

The presented model is able to change a color distribution significantly. Hence, in some cases the strength of transformation should be controlled to avoid artifacts and to achieve a satisfying result. In addition to a linear interpolation between original and resulting image, in a rectified flow model there are two parameters of generation process that naturally control the strength of transfer, namely, a number of steps for ODE solver and a percent of interpolation curve passed (strength) after which generation is stopped. The transfer examples where these two parameters are varied are given in Figs. 6 and 7.

Conclusion

We have introduced a novel approach to color transfer, a process that modifies the colors of an image to match a reference palette, such as the color distribution of a style image. Trained on a set of unlabeled images with diverse color styles, our transfer model offers a unique method of performing color transfer as a density transformation in RGB color space. The use of rectified neural ODEs to learn mappings between color distributions is a significant departure from existing methods. The existence of an inverse function of the ODE allows us to introduce a common latent space for all densities. By constructing a transformation as a composition of a forward and an inverse pass through the latent space, we simplifying the training of generalizing model, which is able to predict the mappings for new content-style image pairs.

The proposed approach outperforms existing state-of-the-art neural methods for color transfer. Furthermore, it is not restricted to a specific domain and can be applied to other areas where an image is associated with a distribution, and distribution transfer is needed.

References

An, J.; Xiong, H.; Huan, J.; and Luo, J. 2020. Ultrafast photorealistic style transfer via neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 10443–10450.

Bonneel, N.; Van De Panne, M.; Paris, S.; and Heidrich, W. 2011. Displacement interpolation using Lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*, 1–12.

Chiu, T.-Y.; and Gurari, D. 2022. Photowct2: Compact autoencoder for photorealistic style transfer resulting from blockwise training and skip connections of high-frequency residuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2868–2877.

Ding, K.; Ma, K.; Wang, S.; and Simoncelli, E. P. 2020. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5): 2567–2581.

Flamary, R.; Courty, N.; Gramfort, A.; Alaya, M. Z.; Boisbunon, A.; Chambon, S.; Chapel, L.; Corenflos, A.; Fatras, K.; Fournier, N.; Gautheron, L.; Gayraud, N. T.; Janati, H.; Rakotomamonjy, A.; Redko, I.; Rolet, A.; Schutz, A.; Seguy, V.; Sutherland, D. J.; Tavenard, R.; Tong, A.; and Vayer, T. 2021. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78): 1–8.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2414–2423.

Ghoskno. 2023. Color-Canny ControlNet. https://huggingface.co/datasets/ghoskno/laion-art-en-colorcanny.

Gui, M.; Fischer, J. S.; Prestel, U.; Ma, P.; Kotovenko, D.; Grebenkova, O.; Baumann, S. A.; Hu, V. T.; and Ommer, B. 2024. DepthFM: Fast Monocular Depth Estimation with Flow Matching. arXiv:2403.13788.

Ho, M. M.; and Zhou, J. 2021. Deep preset: Blending and retouching photos with color style transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2113–2121.

Hong, K.; Jeon, S.; Yang, H.; Fu, J.; and Byun, H. 2021. Domain-Aware Universal Style Transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 14609–14617.

Huang, J.; Wang, H.; Weiermann, A.; and Ruzhansky, M. 2023. Optimal Image Transport on Sparse Dictionaries. *arXiv preprint arXiv:2311.01984*.

Ignatov, A.; Timofte, R.; et al. 2019. PIRM challenge on perceptual image enhancement on smartphones: report. In *European Conference on Computer Vision (ECCV) Workshops*.

Jacobs, C.; Salesin, D.; Oliver, N.; Hertzmann, A.; and Curless, A. 2001. Image analogies. In *Proceedings of Siggraph*, 327–340.

Johnson, J. 2015. Neural Style. https://github.com/jcjohnson/neural-style.

Kastryulin, S.; Zakirov, D.; and Prokopenko, D. 2019. PyTorch Image Quality: Metrics and Measure for Image Quality Assessment. Open-source software available at https://github.com/photosynthesis-team/piq.

Ke, Z.; Liu, Y.; Zhu, L.; Zhao, N.; and Lau, R. W. 2023. Neural Preset for Color Style Transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14173–14182.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, Y.; Fang, C.; Yang, J.; Wang, Z.; Lu, X.; and Yang, M.-H. 2017. Universal style transfer via feature transforms. *Advances in neural information processing systems*, 30.

Li, Y.; Liu, M.-Y.; Li, X.; Yang, M.-H.; and Kautz, J. 2018. A closed-form solution to photorealistic image stylization. In *Proceedings of the European conference on computer vision (ECCV)*, 453–468.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Lobashev, A. 2024. Python Implementation of Random Monotone Color Filters. https://github.com/alobashev/monotone_color_filters.

Luan, F.; Paris, S.; Shechtman, E.; and Bala, K. 2017. Deep photo style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4990–4998.

Mahmoud, A. 2023. Python Implementation of Colour Transfer Algorithm Based on Linear Monge-Kantorovitch Solution. https://github.com/mahmoudnafifi/colour_transfer_MKL.

Morovic, J.; and Sun, P.-L. 2003. Accurate 3d image colour histogram transformation. *Pattern Recognition Letters*, 24(11): 1725–1735.

Neumann, L.; and Neumann, A. 2005. Color style transfer techniques using hue, lightness and saturation histogram matching. In *CAe*, 111–122.

Niklaus, S. 2018. A Reimplementation of HED Using PyTorch. https://github.com/sniklaus/pytorch-hed.

Pitié, F.; and Kokaram, A. 2007. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. In *4th European conference on visual media production*, 1–9. IET.

Pitie, F.; Kokaram, A. C.; and Dahyot, R. 2005. N-dimensional probability density function transfer and its application to color transfer. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, 1434–1439. IEEE.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; and Shirley, P. 2001. Color transfer between images. *IEEE Computer graphics and applications*, 21(5): 34–41.

Ruderman, D. L.; Cronin, T. W.; and Chiao, C.-C. 1998. Statistics of cone responses to natural images: implications for visual codingStatistics of cone responses to natural images: implications for visual coding. *JOSA A*, 15(8): 2036–2045.

Soria, X.; Pomboza-Junez, G.; and Sappa, A. D. 2022. Ldc: Lightweight dense cnn for edge detection. *IEEE Access*, 10: 68281–68290.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Toderici, G.; Shi, W.; Timofte, R.; Theis, L.; Balle, J.; Agustsson, E.; Johnston, N.; and Mentzer, F. 2020. Workshop and Challenge on Learned Image Compression (CLIC2020).

Unsplash. 2023. Unsplash Lite Dataset 1.2.2. https://unsplash.com/data.

Villani, C.; et al. 2009. *Optimal transport: old and new*, volume 338. Springer.

Xie, S.; and Tu, Z. 2015. Holistically-Nested Edge Detection. In *Proceedings of IEEE International Conference on Computer Vision*.

Yoo, J.; Uh, Y.; Chun, S.; Kang, B.; and Ha, J.-W. 2019. Photorealistic style transfer via wavelet transforms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9036–9045.