Color Style Transfer with Modulated Flows

Maria Larchenko [1] **Alexander Lobashev** [1] **Dmitry Guskov** [1] **Vladimir Palyulin** [1]

Abstract

In this work, we introduce Modulated Flows (ModFlows), a novel approach for color style 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 unseen images. We 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.

1. Introduction

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

Classical Methods. The idea of image modifications based on features of another image appeared in the early 2000s under the name "image analogies" (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" in the current literature 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 proposed solution (called as ColorTransfer, CT) treats images as 3D distributions in the $l\alpha\beta$ color space (Ruderman et al., 1998) and adjusts the mean and the variance along the main axes (i.e. marginal moments).

The pioneering works on the color transfer have already considered it as a problem of the optimal transport (Morovic & 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 mass distribution with a minimal effort. The effort needed can be defined as a transportation cost, i.e. the problem can be formulated within the framework of optimal transport (OT) theory. In general case, the exact solution of OT problem is hard to obtain. For instance, in the case of discrete distributions the optimal histogram matching could be utilized (Morovic & Sun, 2003). However, an exact calculation of the transport cost was computationally heavy; for this reason other histogram-based approaches dropped the optimality constraint and considered the color transfer as the mass preserving transport problem (Neumann & Neumann, 2005; Pitie et al., 2005).

Unfortunately, the first attempts suffered from artifacts and additional (sometimes even manual) adjustments were still needed to work around their limitations. Pitié & Kokaram (2007) were the first who switched to a continuous formulation of OT problem in color transfer. The authors made several simplifications, assuming color distributions to be Gaussian. Referred to as Monge-Kantorovitch Linear (MKL) (Mahmoud, 2023) it is still a strong competitor, as shown in Fig. 2.

Neural Methods. Gatys et al. (2016) turned the research into a different direction. It adapted deep convolutional neural networks (CNN) for a high-level style extraction. The algorithm (often referred as Neural Style (Johnson, 2015)) was shown to be able to perform an example-based color transfer when applied to a pair of images. The transfer was not ideal though. Neural Style targets a painting technique and textures. Hence, it blends into a stylized image not only a reference color palette but also 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.,

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Maria Larchenko <mariia.larchenko@gmail.com>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

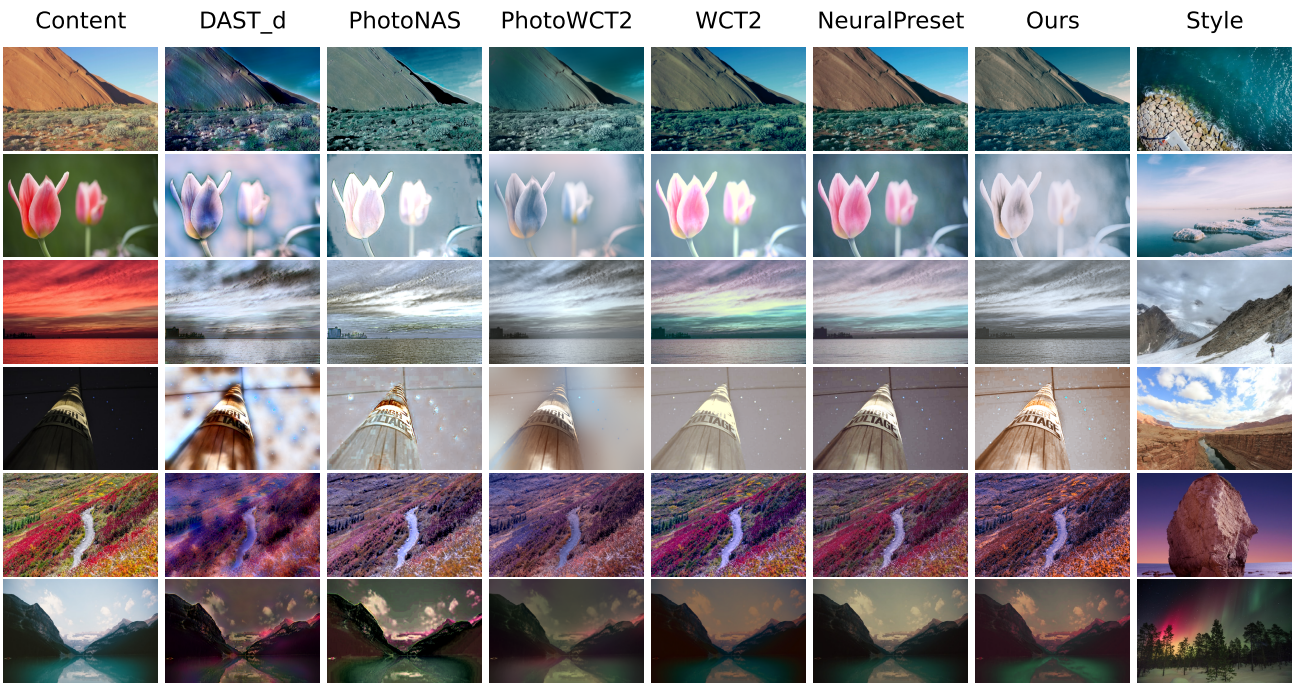

Figure 1. Qualitative comparison. Examples from Unsplash Lite test set. Our model achieves the most exact match with the reference palette without visible distortion (zoom in for better quality).

2019), PhotoNAS (An et al., 2020), PhotoWCT2 (Chiu & Gurari, 2022) and DAST (Hong et al., 2021).

A step aside was taken by Deep Preset (Ho & Zhou, 2021) which is based on the U-net architecture. Deep Preset is aimed at automatic retouching and achieves the high quality in terms of absence of artifacts. It changes the color distribution only slightly. While this is desirable in the case of retouching, it does not suit the color transfer task well. Nevertheless, we have included Deep Preset in comparison to give a reference values of scores for image retouching.

Two most recent studies are closely related to our work. The first one is the Neural Preset approach proposed by Ke et al. (2023), where the color transfer is executed in RGB space by a multilayer perceptron with the hidden weights predicted by an encoder network. The algorithm could be trained in self-supervised fashion. It achieves impressive visual quality and is capable of processing of high-resolution images. However, it heavily depends on external Look-Up-Table (LUT) filters. Designing a diverse set of LUT filters of a high quality requires domain expertise and time. One may treat them as a part of the dataset. These filters, along with the model, are not currently available. Since the test set consists of more than a thousand of images, we only included Neural Preset in qualitative comparison. The results for Neural Preset were obtained via officially distributed application.

The second one is Sparse Dictionaries (Huang et al., 2023), the method based on a discrete optimal transport applied to learned style dictionaries. However, the algorithm is slow compared to the other methods based on neural networks and its code is also unavailable at the moment.

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 parameters, 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 describe a new method of color transfer based on rectified flows and the common latent distribution.
- 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 strongly associated with a color style of the image.

Figure 2. Transfer results and corresponding distributions in RGB space (zoom in for better quality).

2. Background

2.1. 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 & 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 Pitié & 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 in general. 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,

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

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

Monge–Kantorovich minimization finds $\pi^*(X_0, X_1)$ 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

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

2.2. Rectified flows

The optimal transport problem can be approximately solved by Rectified flows (Liu et al., 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

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

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

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

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)

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

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

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

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

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

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

3. 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 universal intermediate distribution U.

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 $X \sim \mathcal{I}$
    $Z = \mathrm{Flow}_\theta(X)$
    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

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

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

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 et al., 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 $\text{Enc}(\mathcal{I}_i) = e_i$ 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

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

where $\text{Enc}(\mathcal{I}_i) = e_i$ 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.

The Algorithm 1 gives the pseudo-code for the proposed solution.

4. Experiments and metrics

Dataset. To implement the approach described above one needs a dataset of images with sufficiently different color styles and resolutions. We construct our dataset by combining DIV2K (Ignatov et al., 2019) and CLIC2020 (Toderici et al., 2020) (designed for image compression challenges) with a subset of "laion-art-en-colorcanny" (ghoskno, 2023) (mostly consisting of cakes). 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.

Encoder. EfficientNet B6 is used as an encoder model (Tan & 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 & 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).

Metrics. The seminal work (Gatys et al., 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 color style 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).

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 & Tu, 2015; Niklaus, 2018) and LDC (Soria et al., 2022) models. The variants of the colorless representation are demonstrated in Fig. 3. The difference between colorless images is evaluated with DISTS (Ding et al., 2020) producing the content score.

Table 1 contains average style distances and aggregated scores for compared methods. The **aggregated score** is calculated as a distance to the ideal point p, similarly with (Ke et al., 2023),

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

Please refer to the Table 3 (supplemental materials) for the same evaluation with SSIM.

Search of similar color styles. Once trained, the output vector of parameters e could serve as an embedding of an image color style. 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. 9 and Fig. 10 in supplemental materials.

Table 1. Comparison of algorithms

Aggregated scores (DISTS)↓				Style distance↓	
Algorithm	Grayscale	Depth	Edge (Xie & Tu, 2015)	Algorithm	mean ± std
ModFlows (ours)	**0.129**	**0.217**	0.220	DAST_d	**0.112 ± 0.039**
MKL	0.146	0.227	0.224	ModFlows (ours)	0.123 ± 0.049
CT	0.169	0.234	0.232	DAST_da	0.127 ± 0.042
WCT2	0.170	0.228	0.249	PhotoWCT2	0.129 ± 0.055
PhotoWCT2	0.191	0.236	**0.217**	MKL	0.145 ± 0.060
DAST_d	0.204	0.267	0.224	WCT2	0.163 ± 0.065
DAST_da	0.214	0.282	0.229	CT	0.166 ± 0.064
PhotoNAS	0.224	0.276	0.270	PhotoNAS	0.183 ± 0.069
Deep Preset	0.384	0.400	0.387	Deep Preset	0.384 ± 0.171

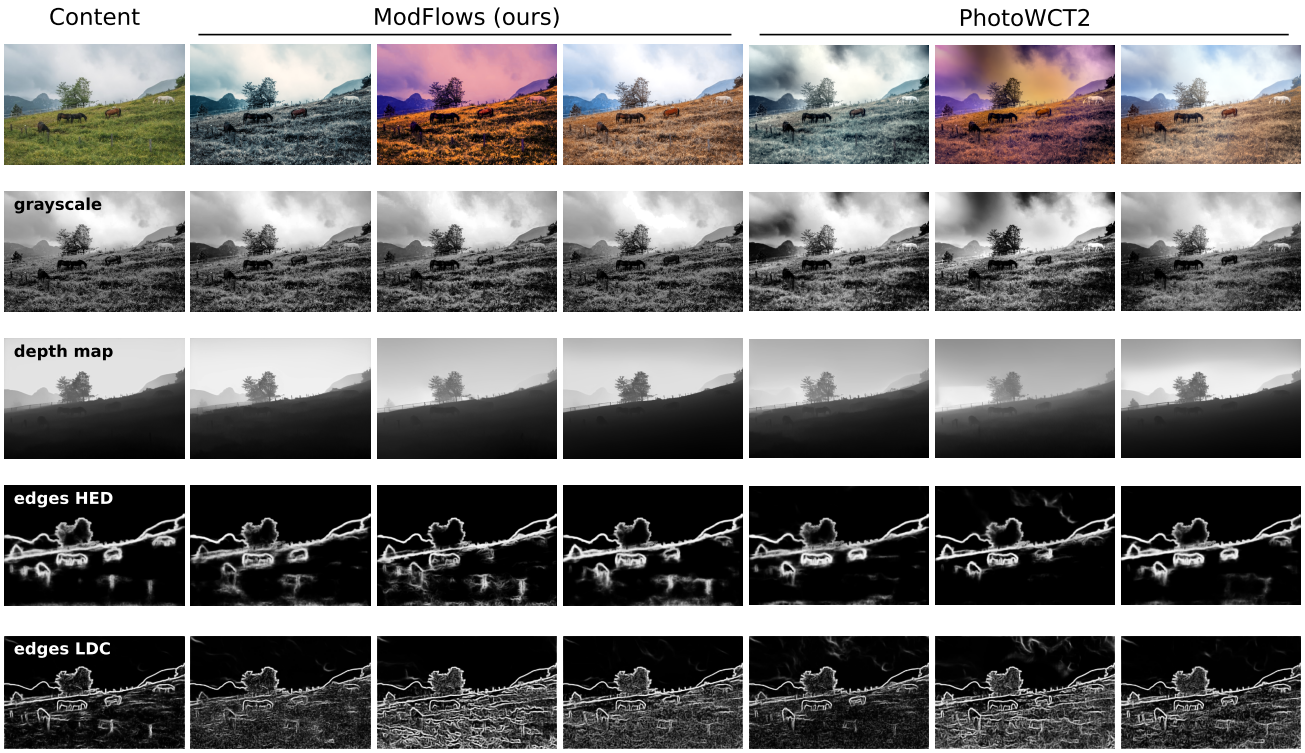

Figure 3. Colorless content metrics. The choice of the best content metric is not obvious. Edges detection by HED model (Xie & Tu, 2015) grasp mostly the main objects of a scene, while canny LDC (Soria et al., 2022) images are capturing the too detailed edges. Both of them are not sensitive to low-frequency 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).

Table 2. Ablation study

Aggregated ablation scores (DISTS)↓				
Algorithm	Grayscale	Depth (Gui et al., 2024)	Edge (Xie & 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

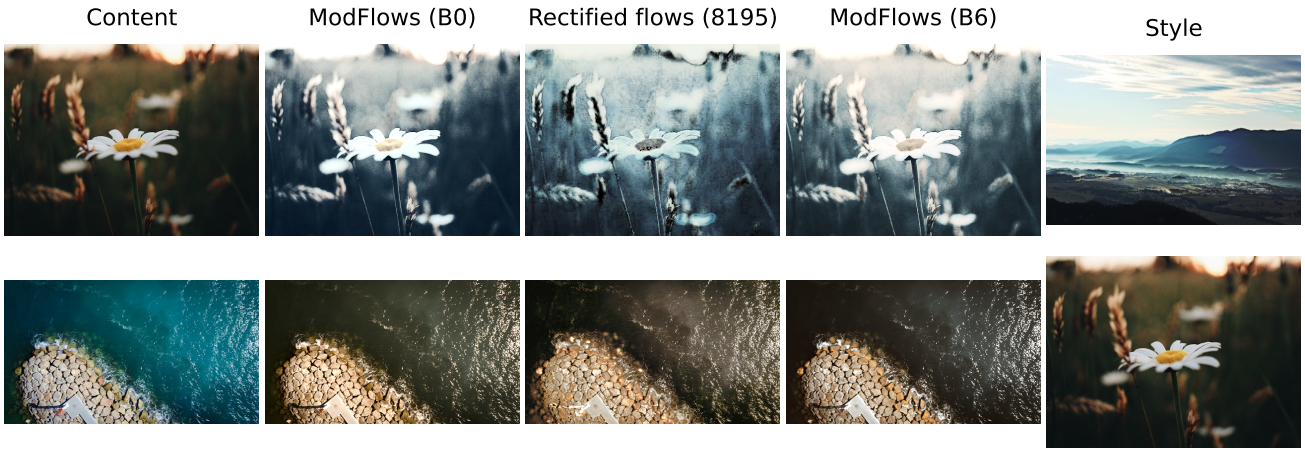

Figure 4. Ablation study. ModFlows models reach a better trade-off between style and content similarity when compared to dataset models used in their training.

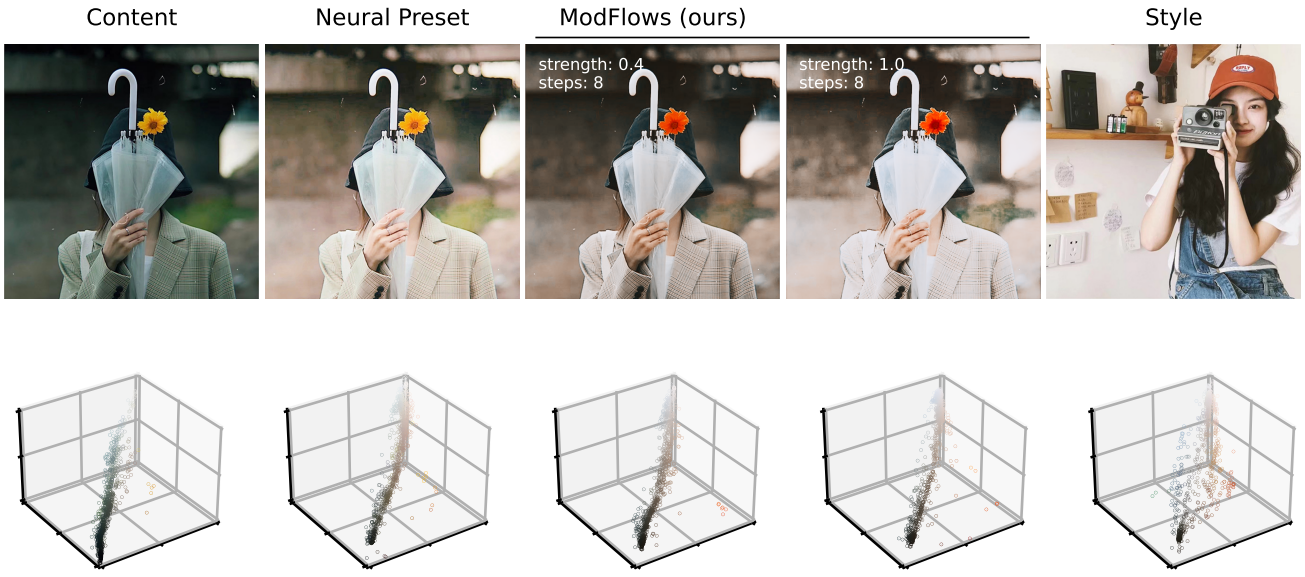

Figure 5. Limitations and algorithm tuning. An example of color switching in two pictures generated with fixed number of steps for ODE solver (steps) and varied percent of interpolation curve passed (strength).

5. Ablation study

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. 4. 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.

6. Limitations and algorithm tuning

The framework of the transport theory gives us an opportunity to design an unsupervised algorithm. In the same time it puts a limitation, that is a greater dependence of the result on the reference image. For instance, depending on the target picture, the method could perform a color replacement, i.e yellow shades may be transformed to red ones, while staying coherent to each other, Fig. 5.

The developed transfer 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. 5 and 7 (in Appendix section).

7. Conclusion

We have introduced a novel approach to color style 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 three-dimensional distributions is a significant departure from existing methods. The existence of an inverse function of the transformation 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 unseen images.

Our proposed approach has shown superior performance in comparison to the available state-of-the-art neural methods. Additionally, it not bound to the specific data and could be robustly reproduced.

References

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

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

Chiu, T.-Y. and Gurari, D. 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*, pp. 2868–2877, 2022.

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

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. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

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

ghoskno. Color-canny controlnet. https://huggingface.co/datasets/ghoskno/laion-art-en-colorcanny, 2023.

Gui, M., Fischer, J. S., Prestel, U., Ma, P., Kotovenko, D., Grebenkova, O., Baumann, S. A., Hu, V. T., and Ommer, B. Depthfm: Fast monocular depth estimation with flow matching, 2024.

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

Hong, K., Jeon, S., Yang, H., Fu, J., and Byun, H. Domain-aware universal style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14609–14617, October 2021.

Huang, J., Wang, H., Weiermann, A., and Ruzhansky, M. Optimal image transport on sparse dictionaries. *arXiv preprint arXiv:2311.01984*, 2023.

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

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

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

Ke, Z., Liu, Y., Zhu, L., Zhao, N., and Lau, R. W. Neural preset for color style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14173–14182, 2023.

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

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

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

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

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

Mahmoud, A. Python implementation of colour transfer algorithm based on linear monge-kantorovitch solution. https://github.com/mahmoudnafifi/colour_transfer_MKL, 2023.

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

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

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

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

Pitie, F., Kokaram, A. C., and Dahyot, R. 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, pp. 1434–1439. IEEE, 2005.

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

Ruderman, D. L., Cronin, T. W., and Chiao, C.-C. 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, 1998.

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

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

Toderici, G., Shi, W., Timofte, R., Theis, L., Balle, J., Agustsson, E., Johnston, N., and Mentzer, F. Workshop and challenge on learned image compression (clic2020), 2020. URL http://www.compression.cc.

Unsplash. Unsplash lite dataset 1.2.2. https://unsplash.com/data, 2023.

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

Xie, S. and Tu, Z. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.

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

A. Appendix / supplemental material

All experiments were conducted on a workstation equipped with two NVIDIA RTX 4090 graphics cards and 256 GB of RAM. Our code is avalable at `https://github.com/maria-larchenko/modflows`.

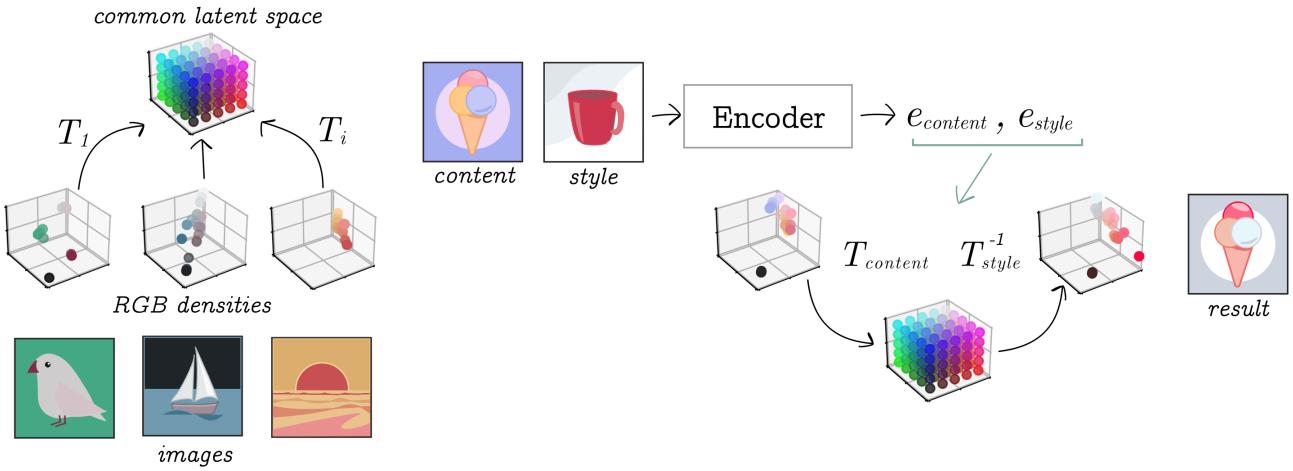

Figure 6. General scheme of our color style transfer approach. First, we generate a set of 5,826 flow-image pairs. Each flow T is trained to map an RGB density of an image into a uniform cube. Note that a rectified flow T is bijective. Then, we train an encoder model to produce weights e for T. Color style transfer is a composition of two flows applied to the content image: A content forward flow and a style inverse flow.

Table 3. Aggregated score and style distances given for more widespread SSIM and OpenCV Canny edge detection. For CV2 Canny low threshold is 100, high threshold is 200. Here, DAST_d refers to DAST with a vanilla decoder, whereas DAST_da represents DAST using a decoder with an adversarial loss (Hong et al., 2021).

Aggregated scores (SSIM)↓			
Algorithm	Grayscale	Depth	Edge
ModFlows (ours)	**0.131**	**0.177**	0.422
MKL	0.146	0.189	**0.321**
CT	0.170	0.197	0.330
WCT2	0.178	0.194	0.468
PhotoNAS	0.298	0.237	0.500
PhotoWCT2	0.349	0.193	0.334
DAST_d	0.375	0.218	0.494
DAST_da	0.377	0.236	0.492
Deep Preset	0.384	0.389	0.413

Style distance↓	
Algorithm	mean ± std
DAST_d	**0.112 ± 0.039**
ModFlows (ours)	0.123 ± 0.049
DAST_da	0.127 ± 0.042
PhotoWCT2	0.129 ± 0.055
MKL	0.145 ± 0.060
WCT2	0.163 ± 0.065
CT	0.166 ± 0.064
PhotoNAS	0.183 ± 0.069
Deep Preset	0.384 ± 0.171

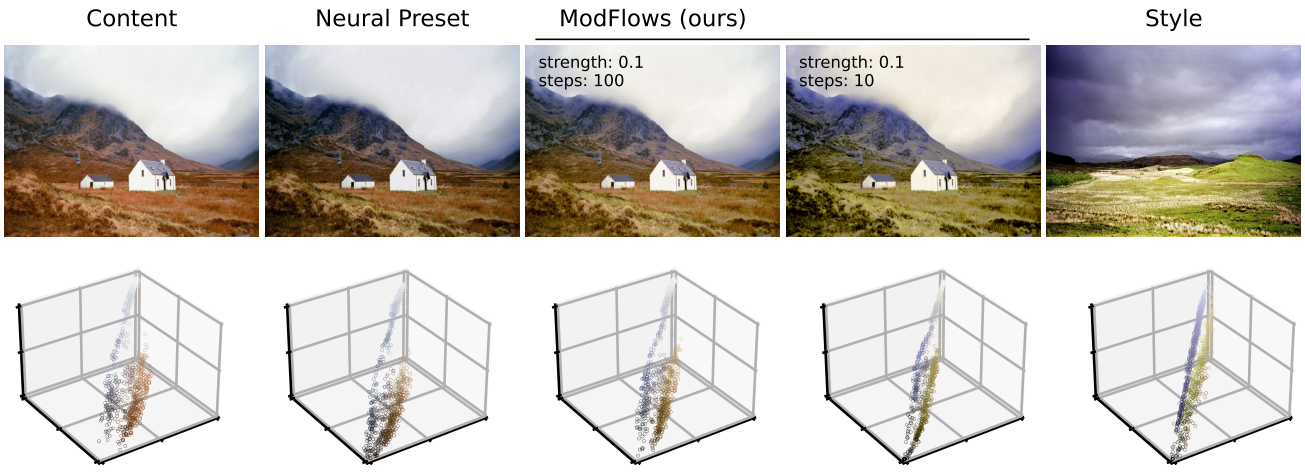

Figure 7. Algorithm tuning. Variation of a number of steps for ODE solver (steps) and a percent of interpolation curve passed (strength) results in different amount of changes for a distribution. In this example, increasing the strength or decreasing the number of steps further leads to the appearance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

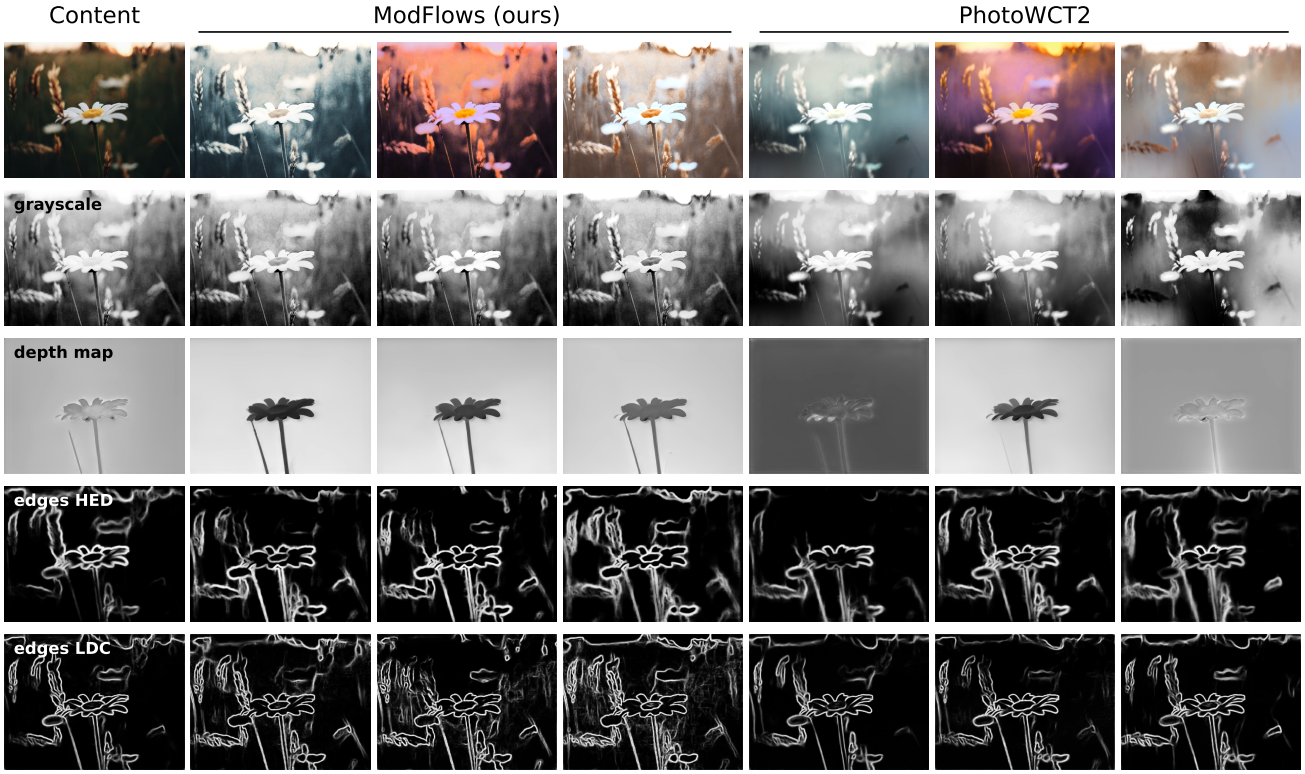

Figure 8. Colorless content metrics. Depth maps are given by (Gui et al., 2024) model. Edges detection by HED model (Xie & Tu, 2015) grasp mostly the main objects of a scene, while canny LDC (Soria et al., 2022) images are capturing the too detailed edges. Both of them are not sensitive to low-frequency artifacts. Absence of such artifacts should be additionally detected.

Figure 9. Search for similar color styles in the Unsplash Lite dataset (25k of images) based on the ModFlows (B6) model output.

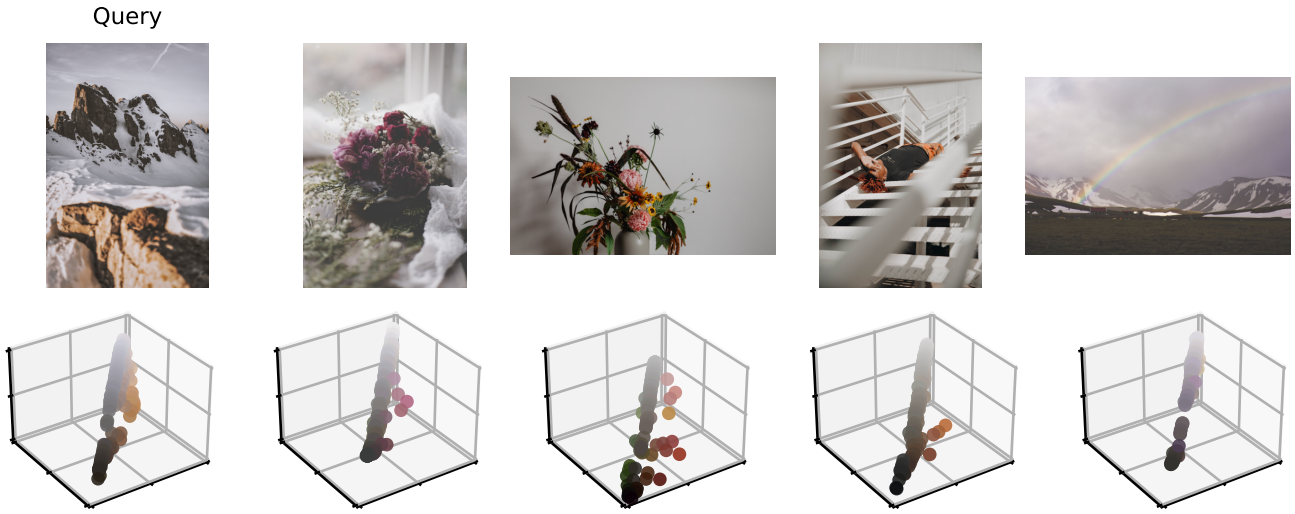

Figure 10. Search for similar color styles in the Unsplash Lite dataset (25k of images) based on image statistics, specifically flattened vectors representing the first and second centered moments of the color distribution.