Appendix for
Content-Aware Image Color Editing with Auxiliary Color Restoration Tasks

Yixuan Ren[1]*, Jing Shi[2], Zhifei Zhang[2], Yifei Fan[2], Zhe Lin[2], Bo He[1], Abhinav Shrivastava[1]

[1]University of Maryland, College Park [2]Adobe Research

{yxren,bohe,abhinav}@cs.umd.edu, {jingshi,zzhang,yifan,zlin}@adobe.com

A. Preliminary of Diffusion Models

Diffusion models consist of a forward diffusion process progressively adding Gaussian noise to a data point and a reversed denoising process generating a data point from pure random noise:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} \mathcal{N}(x_{t-1}; \sqrt{\alpha_t}x_{t-1}, (1-\alpha_t)I), \quad (1)$$

where α_t are hyperparameters of the noise schedule. The forward process can also be marginalized at each step as

$$q(x_T|x_0) = \mathcal{N}(x_t; \sqrt{\gamma_t}x_0, (1-\gamma_t)I), \quad (2)$$

where $\gamma_t = \prod_t^{t'} \alpha'_t$.

Using the reparameterization trick, the forward diffusion process can be formulated as stepwise operations

$$\begin{aligned} z &:= (x_T \oplus \epsilon_T \oplus \cdots \oplus \epsilon_1) \sim \mathcal{N}(0,1), \\ x_{t-1} &= \mu_T(x_t, t) + \sigma_t \odot \epsilon_t, t = T, ..., 1, \end{aligned} \quad (3)$$

where $\oplus$ denotes concatenation operation, and the Gaussian parameterization provides that

$$\begin{aligned} \mu &= \frac{\sqrt{\gamma_{t-1}}(1-\alpha_t)}{1-\gamma_t}x_0 + \frac{\sqrt{\alpha_t}(1-\gamma_{t-1})}{1-\gamma_t}x_t, \\ \sigma^2 &= \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}. \end{aligned} \quad (4)$$

B. Additional Experiment Results

B.1. Random Sampling Color Editing

Fig. 1 shows another example that our model generates content-aware output based on different patches cropped from the same raw image as the input. Given the same style latent noise z, our model performs consistent color editing on different input patches, as well as adapts to the specific semantics and structures of the input. For z_1 and z_2, the patches with sky only always have the most dramatic and artistic colors of blue and dark green. And when they contain more buildings, street views and persons, the extreme colors are mainly constrained in the sky area, while the other objects show shallower shades of them as being illuminated and reflected under the certain weather conditions. z_3 provides a special case: our model also generates fancy colors on a complicated scene, as long as it looks harmonic (this background shares the similar color across buildings and the ground, and it also depends on the person's clothing). On the contrary, when the patches become simpler without abundant objects in the scene, such a pink filter doesn't fit well and our model also applies less exaggerated colors on them.

B.2. Exemplar-based Color Editing

Fig. 2 display more qualitative results of exemplar-based image color editing. In each subfigure the first row is managed to have identical visual output from three different models as the reference. And then the same style latent vector is applied to other new input images in the below rows.

In subfigure (a), the reference editing style is to enhance the brightness and contrast. But because the reference images have slight green color tone as their content, SpaceEdit applies a green filter to all kinds of other input images, turning other ground (2nd row), wall (3rd), flower (4th row) and sky (5th row) greener. Our joint model faithfully follow the reference editing.

In subfigure (b), the reference editing style is also to emphasize the contrast and lifting the brightness. Our joint model is the only one to successfully preserving and darken the original color: blue sky becomes deeper blue (2nd row), green grass becomes more solid green (3rd row), and *etc*. SpaceEdit learns a not pure transform that contains the red and yellow color in the reference image pairs, and thus turning blue sky and green grass even shallower by combining the complementary colors and cancelling each other out.

*Part of this work was done when Yixuan was an intern at Adobe Research.

Figure 1. Additional qualitative results of random sampling generation on different patches cropped from the same image.

In subfigure (c), the reference editing style is to darken the background while the foreground objects remain the original brighness as highlight. SpaceEdit and the vanilla LDM fail to keep the trees and leaves in a good lighting (2nd and 3rd rows). Besides, they also dye the waterfall into blue. Our joint model follows the reference strictly and produces clear foreground with details visible.

In subfigure (d), the reference pairs largely enhance the raw image with more vivid and vibrant colors, especially orange, as well as address the contrast between warm and cold colors. Our joint model executes the same style to other input images with distinctive promotion. The vanilla LDM has the similar effect but less outstanding. SpaceEdit instead completely misunderstands the target direction and yield unappealing results.

B.3. Language-Guided Color Editing

More language-guided image color editing results are shown in Fig. 3. In the unmasked task, our model generates more solid and brighter colors on the mountain, tree and grass, sky and river, and flower. Please especially note the dark night case (3rd column), where SpaceEdit edits it into black background but still with a shot of white moonlight. Our model instead turning it into dark blue without obvious source of direct light beam, rendering a realistic nature scene. In the masked task, our model's output are also better aligned with both the text prompts and the original texture of the leaves. Our white leaves are more solid, blue leaves are more realistic without reflecting white highlight, and yellow leaves are purer without slight shift to green as of SpaceEdit.

C. Content-Awareness Metrics

To quantitatively measure and compare the content-awareness property across multimodal generative models, we propose a novel framework of metrics based on the correlations between the input image's content and their output edit styles given a certain set of input noise. We name it Content-Awareness Metrics (CAMs), and it has three variants as shown in Fig. 4, *i.e*. *CAM-1*, *CAM-2* and *CAM-3*.

In brief, we first generate output images given a set of input images and a set of style latents. *CAM-1* calculates the diversity of the output images given one certain style latent. *CAM-2* calculates the correlation between the input images' contents and the output images' styles given the same style latent, and *CAM-3* calculates the correlation between the input images' contents and the distributions of output images' styles given the same set of style latents.

C.1. Column Diversity

The currently widely used diversity in the generative area is calculated over the set of various output images given one certain input image and different random noise for styles. Calculating the mean pairwise distance among these output images indicate the variation range across the learned style latent space. Inspired by but on the contrary to this, we focus on how the model performs given the same random noise and various input images. This reflects how the model output adapts to different input images according to their specific content and semantics.

Fig. 4 visualizes the matrix of input images, random latent noise and output images. The conventional diversity is measured along a row horizontally, while our column diversity is measured along a column vertically. Formally, given a set of input images $y_{i=1,\dots,M}$ and a set of input noise $z_{j=1,\dots,N}$, the output images are generated as $x_{ij} = \mathrm{Gen}(y_i, z_j)$, where Gen is the generative model. Then the conventional row diversity is typically calculated over the output images $x_{1j} = \mathrm{Gen}(y_1, z_j)$ given a fixed

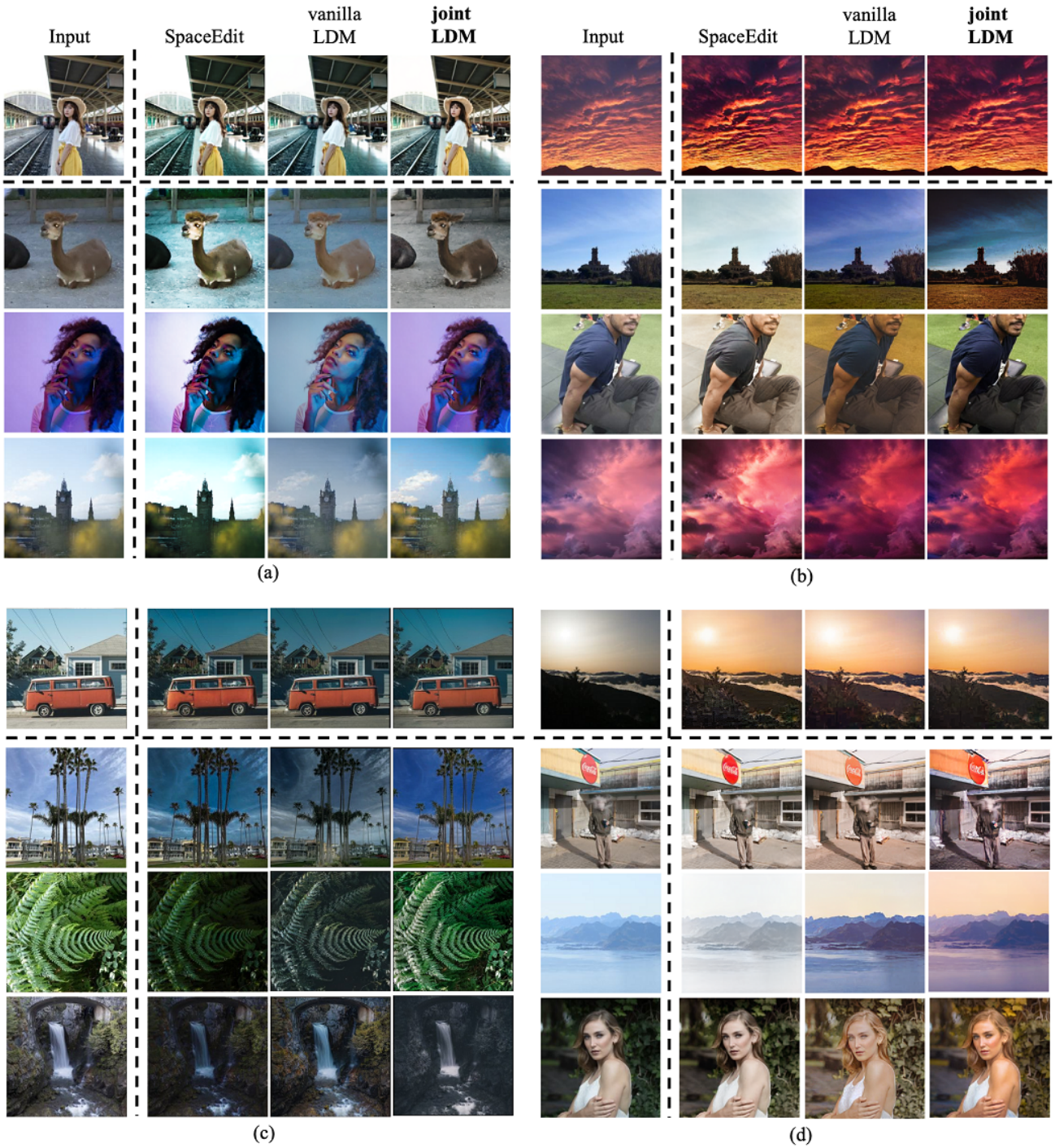

Figure 2. Additional qualitative results of exemplar-based color editing.

input image x_1 as

$$\text{Diversity}_{\text{row}} = \frac{2}{N(N+1)} \sum_{j,k}^{N} \text{LPIPS}(x_{1j}, x_{1k}), \quad (5)$$

$$\text{or Diversity}_{\text{row}} = \sigma\{x_{1j}\}|_{j=1}^{N},$$

where LPIPS represents the distances between the feature maps of two images calculated by a pretrained neural network, and σ refers to the standard deviation in the pixel space.

By contrast, our column diversity focuses on the images

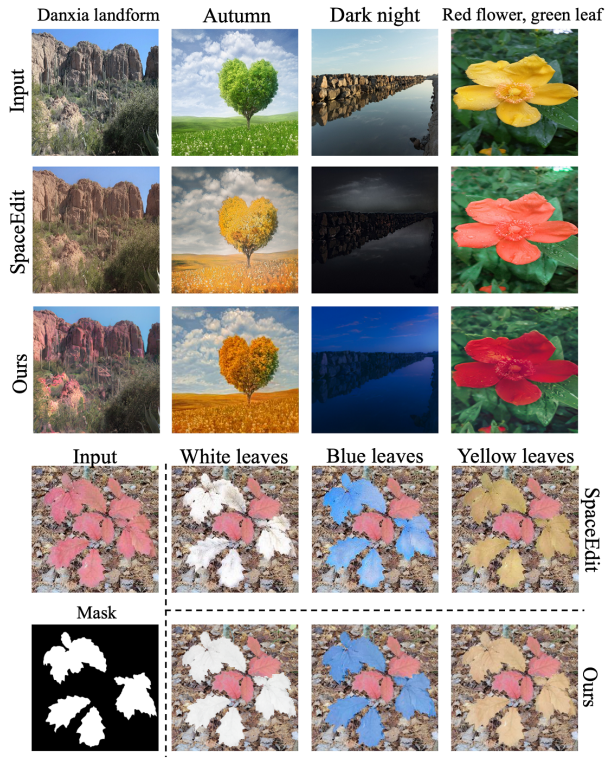

Figure 3. Additional qualitative results of language-guided color editing.

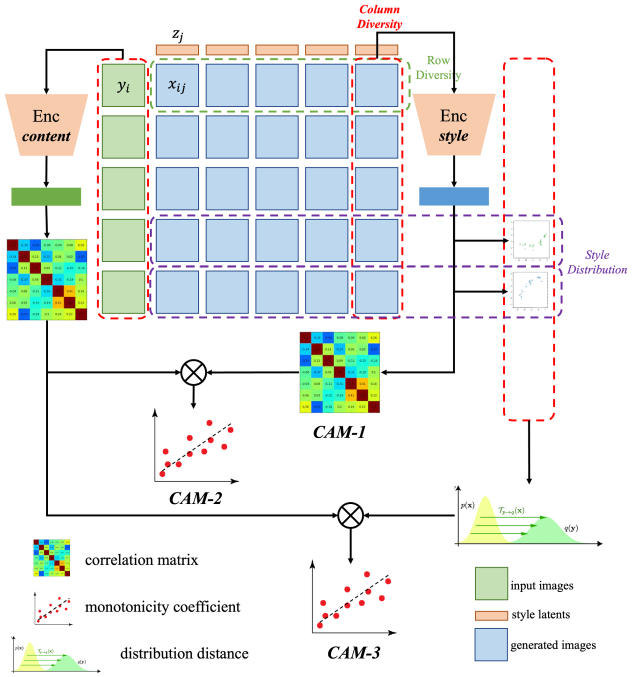

Figure 4. Illustration of our proposed framework of metrics for content-awareness in MMI2IT tasks. Conventional diversity is calculated over each row, measuring the output various given different noise for the same image. Our proposed metrics are calculated over each column to reflect how the same latent noise have adaptive editing style for different input contents. Sec.C in details narrates the whole framework and complete implementations.

generated by different input images with a fixed noise, *i.e.* $x_{i1} = Gen(x_i, z_1), i = 1...M$. It is the first and most basic variant *CAM-1* can be generally formulated as:

$$\begin{aligned} s_{ij} &= \mathrm{Enc}_{\mathrm{style}}(x_{ij}|y_i), \\ \mathrm{CAM\_1} &:= \mathrm{Diversity}_{\mathrm{col}} \\ &= \frac{2}{M(M+1)} \sum_{i,k}^{M} \mathrm{dist}(s_{i1}, s_{k1}), \end{aligned} \tag{6}$$

where $\mathrm{Enc}_{\mathrm{style}}$ is a designated pretrained style encoder, and dist refers to a form of distance measurement such as L1 or L2 norm or cosine distance *etc*.

Note that a core challenge here is to solely represent an image's style properly. In Eq. 5, this problem is smartly bypassed by only calculating the output images from the same input image, which share all the same input content and structure naturally. Unfortunately so far there doesn't exist a widely-acknowledged method or model off the shelf to extract the pure style from an image regardless of its content and structure broadly applicable for most tasks. For our image color editing task in particular, we set up a pipeline to adopt a neural network to serve as $\mathrm{Enc}_{\mathrm{style}}$ and train it on our specific data and labels to serve as an approximation. This is illustrated in details in Sec. C.3.

C.2. Correlation between Content and Style

As a univariate criterion, the column diversity or *CAM-1* doesn't incorporate the relationship between the output styles and the input content. While its zero value represents no content-awareness, a big value either just indicates the output styles are completely haphazard and even harms the physical meaning of the same input noise vector. So we further measure the correlation between the input image content and the output editing style. Since these two representations are not in the same latent space, we alternatively simplify the objective to first calculating the pairwise distances among themselves respectively, and then measuring the correlation between the two sets of distances. Following the same notations in Eq.6, *CAM-2* is calculated as

$$\begin{aligned} c_i &= \mathrm{Enc}_{\mathrm{content}}(y_i), \\ \mathrm{CAM\_2} &:= \mathrm{Corr}_{i=1}^{M}(s_{i1}, c_i), \\ &:= \mathrm{Corr}_{i,k=1}^{M}(\mathrm{dist}(s_{i1}, s_{k1}), \mathrm{dist}(c_i, c_k)), \end{aligned} \tag{7}$$

where $\mathrm{Enc}_{\mathrm{style}}$ is a designated pretrained content encoder, and Corr refers to a form of correlation between two sets of univariate samples. Similarly, it is also a challenge to define and prepare a proper and accurate content encoder for many

or one task. Its details are described in Sec. C.3.

Furthermore, we regard the output editing styles $\{s_{ij}\}|_{j=1}^{N}$ of a single input image y_i given a set of input random noise $\{z_j\}|_{j=1}^{N}$ as a distribution with multiple sampling. The correlation between these distributions in the form of their pairwise divergences and the input content in the form of their pairwise distances then reflects the content-awareness in a deeper level. Formally, the *CAM-3* metric in Fig. 4 is calculated as

$$\begin{aligned} \text{CAM\_3} &:= \text{Corr}_{i=1}^{M}(\{s_{ij}\}|_{j=1}^{N}, c_i), \\ &:= \text{Corr}_{i,k=1}^{N}(\text{Div}(\{s_{ij}\}|_{j=1}^{N}, \{s_{kj}\}|_{j=1}^{N}), \\ &\quad \text{dist}(c_i, c_k)), \end{aligned} \tag{8}$$

where Div refers to a form of divergence between two distributions.

In addition, please be advised that it is still not always the bigger the better for *CAM-2* and *CAM-3*, because there also exist some commonly welcomed color editing styles shared across different input content clusters. A too big value *i.e.* an over high correlation with exclusive editing styles for certain input content may imply the model overfitting on the ground truth and even mode collapse. A comparable value to the ground truth target images would be a proper reference.

C.3. Content and Style Encoders

In practice, it is challenging to acquire good representations for the content of an input image and the color editing style of a pair of input and output images. Their definitions are ambiguous and include various aspects comprehensively, and also depend on specific tasks and objectives. Inspired by FID [2], we hire a pretrained Inception-v3 [3] model for its last layer's output before Softmax, i.e. the same feature vector of the FC layer of dimension 2048 used in FID to use as our content representation.

On the other hand, it's not trivial to extract the pure editing style representation completely out of the content and structure even given a pair of the input and output images. For example, the direct difference between individual representations of the input and output images still contains strong information about the main content and structure of the images (*e.g.* plain difference in RGB space, or LPIPS as of difference between a set of feature maps). More fundamentally, this is because the definition of styles remain ambiguous, lacking details and highly dependent on specific tasks, and the labels are limited and supervised models are so affected. We cannot leverage any existing generative models such as StyleGAN either as we're setting an objective criterion to fairly assess them

To address this problem, inspired by [4] that trains a model specifically for painting art style classification and proposes *art-FID*, we train a Inception-v3 model from scratch on our own labels for the image color editing task. These labels are the ground truth color adjustment slider values that user applied in the Lightroom software. The model takes in two concatenated images as the input and output pair to predict the ground truth slider values, and we then use the same last feature vector of 2048 dimensions to represent the color editing style. This aligns with the standard FID feature vector well in shape and value range when calculating their distances and correlations, although not that they are still two different latent space and cannot be operated together natively.

C.4. Implementations

We use L2 distance for all $\text{dist}(\cdot,\cdot)$ between two vectors u and v in Eqs. 6, 7 and 8:

$$\text{dist}(u, v) = \|u - v\|_2. \tag{9}$$

We choose Wasserstein distance to represent the divergence between two distributions $\{u_i\}_i$ and $\{v_i\}_i$ in Eq. 8, assuming they are the distributions of two random variables μ and υ:

$$\begin{aligned} \text{Div}(\{u_i\}_i, \{v_i\}_i) &:= \mathcal{W}_p(\mu, \upsilon) \\ &= \left(\int_0^1 |F_\mu^{-1}(\alpha) - F_\upsilon^{-1}(\alpha)|^p d\alpha \right)^{\frac{1}{p}}, \end{aligned} \tag{10}$$

where F^{-1} is the inverse cumulative distribution function (CDF), and we choose $p = 2$ as the power or moment. Because both our content and style vectos are of dimension $d = 2048 > 1$, we in practice calculate the sliced Wasserstein distance [1] by randomly projecting the samples onto a unit sphere first:

$$\begin{aligned} \text{Div}(\{u_i\}_i, \{v_i\}_i) &:= \mathcal{SW}_p(\mu, \upsilon) \\ &= \left(\int_{\mathbb{S}^{d-1}} \mathcal{W}_p^p(\mu^\theta, \upsilon^\theta) d\theta \right)^{\frac{1}{p}} \\ &= \left(\int_{\mathbb{S}^{d-1}} \int_0^1 |F_{\mu^\theta}^{-1}(\alpha) - F_{\upsilon^\theta}^{-1}(\alpha)|^p d\alpha d\theta \right)^{\frac{1}{p}}, \end{aligned} \tag{11}$$

where $\mathbb{S}^{d-1}$ is the unit sphere of dimension $d - 1$.

We select the F-test algorithm for linear regression to calculate $\text{Corr}(\cdot,\cdot)$ in Eqs. 7 and 8. In particular, we first calculate the Pearson correlation coefficient:

$$\begin{aligned} \rho_{X,Y} &= \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \\ &= \frac{\mathbb{E}(X - \mu_X)(Y - \mu Y)}{\sigma_X \sigma_Y}, \end{aligned} \tag{12}$$

Table 1. The quantitative results of the proposed Content-Awareness Metrics (CAMs). In the 3rd *Style* column, "random" refers to random perturbations added to the image data, and "big" and "small" refers to the scale of the perturbation added. "same" refers to a same deterministic perturbation added to all data samples. The validation section shows our proposed metrics are consistent with the original data and additional perturbations. The model section shows that our best model with joint auxiliary tasks outperforms other competitors on these metrics.

Experiments	Content	Style	CAM-1↑	CAM-2		CAM-3	
				F-stat↑	p-val↓	F-stat↑	p-val↓
Validations	raw content	target content	-	21686.	0.	-	-
	raw	raw	-	13.093	0.0141	-	-
	raw	target	-	**34.665**	0.0162	-	-
	target	target	-	29.950	0.0194	-	-
	target	raw	-	14.843	0.0134	-	-
	raw	raw + random big	-	18.253	0.014	-	-
	raw	target + random big	-	22.178	0.0125	-	-
	raw	raw + random small	-	13.504	0.0165	-	-
	raw	target + random small	-	31.490	0.0126	-	-
	raw	raw + same	-	16.050	0.0135	-	-
	raw	target + same	-	27.228	0.0136	-	-
Models	raw	SpaceEdit (60k)	0.2467	15.093	0.0181	2.4619	0.0243
	raw	SpaceEdit	0.2816	16.257	0.0122	2.9985	0.0253
	raw	Ours LDM baseline	0.3571	21.553	0.0148	4.9446	0.0204
	raw	**Ours LDM Joint L1 [main]**	**0.5767**	**28.343**	0.0142	**7.1684**	0.0278

and then convert it into F-stat and p-val:

$$\begin{aligned} \mathrm{Corr}(X,Y) := F_{X,Y} &= \frac{\rho^2_{X,Y}}{1-\rho^2_{X,Y}} \cdot \frac{n-k-1}{k}, \\ p_{X,Y} &= 1 - F_{(k,n-k-1)}(|F_{X,Y}|), \end{aligned} \tag{13}$$

where n is the number of samples, k is the number of features (1 here as our X and Y are both distances), and $F_{(k,n-k-1)}$ is the CDF of the F-distribution with $(k, n-k-1)$ degrees of freedom. Here n is the number of samples for X, *i.e.* the number of pairs for distances used, and k is the dimension of Y, *i.e.* 1 as it is distance. Although the main purpose of this metric is to measure linear correlation, we notice that it also performs well to distinguish the property of monotonically increasing for a mapping $f : X \to Y$ relatively, that is, to reflect how the distance between two output styles are correlated with the distance between their corresponding input content.

C.5. Experiments and Results

We randomly sample 5000 pairs of raw images and their output to execute the above pipelines as one run, and conduct 16 individual runs for average. The quantitative results are listed in Tab. 1.

We first design and perform some preliminary experiments to validate the consistency and rationality of our proposed metrics. We calculate the metrics for the ground truth raw and target pairs, as well as exchanging or duplicating them, and applying various types of perturbations with different scales on them. It shows that our metrics are well consistent with the hypotheses that in the original data from community users, the edited images follow some patterns according to their original content and semantics, while the raw images have quite arbitrary shooting environment of color tones and lighting *etc*. And different scales of perturbations lead to corresponding scales of impacts on our metrics' values.

Then the comparison among our proposed models, its baseline models and previous SOTA methods demonstrates that our joint model outperforms all other competitors. Our ablation models also have intermediate values between the previous SOTA and our final best model.

D. Latent Noise Space Analyses

D.1. Purity of Color Style Latent

To examine what is learned in our color editing style latent, we conduct an experiment based on the paired image color editing transfer task, but both the source and target reference images input to the DPM-Encoder are the edited image, and the acquired color editing style is applied on a

Figure 5. Visualizations of the purity of our learned color style latent. (c) = (b) $\ominus$ (a) $\oplus$ (a), proving the reconstruction ability of our model. (d) = (b) $\ominus$ (b) $\oplus$ (a), indicating that our model has learned the ideal identity transform from identical source and target images.

new raw image. Ideally, it is expected not to transfer any color style information as the reference pair contains only identity transform.

Fig. 5 displays the results. Here columns (a) and (b) are the original raw and edited images from the dataset, and column (c) is a reconstruction of (b) by applying the inverted color style from the pair of (a) and (b) back to (a) again. It shows that our model is able to produce faithful reconstruction on both content structure and color style. Then we invert (b) over itself as instructed above, to acquire a corresponding style latent. And we apply this style to (a) again to generate column (d). It shows that (d) is identical to (a). This demonstrates that our model has learned a well pure identity transform when the source and target images are identical in the reference pair, which is independent to their concrete styles and content *etc*.

This implies one of the reasons of our model's improved content-awareness: our model's latent style noise is pure on the color editing transforms and differences only, and will not introduce any particular style or content themselves from the source or target reference images. For example, transferring an editing of increasing brightness from a pair of red images to a new raw green image with our model won't bring the base red tone to mess up the original green color. This also contributes to our model's semantic-adaptive harmonic output when sampling from random noise: since our latent style noise space learns the color editing transforms only, it processes the training data from the right perspective of color style differences instead of color styles. Then our model learns the realistic color editing transforms that community users actually performed, based on their frequencies and correlations with various input content. Learning impure editing styles entangled with their own color tones *etc*. will mess up and fail to capture the right patterns in real data.

D.2. Latent Noise Interpolation

We further perform linear interpolation over the learned latent noise space. Our models are based on a standard latent diffusion model, and the size of their latent noise space is $(T+1) \times 3 \times \frac{r}{f} \times \frac{r}{f}$, where $T = 100$ is the number of diffusion steps using a respaced DDPM scheduler, $r = 256$ is the image resolution and $f = 4$ is the compression factor. This space is significantly bigger than previous GAN-based methods such as StyleGANs. We doesn't adopt DDIM schedulers where $\sigma = 0$ and thus no additional noise beside the initial $t = T$ step, to make sure the generative quality without compromisation.

To keep the interpolated noise also standard Gaussian distribution, we calculate it by

$$z_\lambda = \sqrt{\lambda} \cdot z_0 + \sqrt{1-\lambda} \cdot z_1 \sim \mathcal{N}(0,1), \qquad (14)$$

for every timestep t (omitted), where $z_0, z_1 \sim \mathcal{N}(0, 1)$ are the two reference noise vectors at that step. The visual results are shown in Fig. 6. It displays gradual changes from one end to the other, which reveals the arithmetic property of our model and potential benefits for more downstream applications such as image color editing style clustering and retrieval.

E. Visualizations of Auxiliary Color Restoration Tasks

We visualize the output of our proposed auxiliary color restoration tasks during joint training in Figs. 7 and 8, for the luminance input (Y channel, grayscale) and chrominance input (UV channels) respectively. All images are generated with our jointly trained model, *i.e*. the output only depends on the input image type from the same model. Note that we don't measure their performance but only leverage them for assistance. They reveal significantly high diversity and content-awareness, which boost our model in these aspects for our main color editing task.

For example, in Fig. 7 about chrominance, the color of sky vary largely at sunrise or sunset (6th, 11th, 13th and 15th rows), but its color remains within the shadows of blue or green when the lighting is stable (5th, 7th and 12th rows), given all the same other circumstances including the outdoor natural landscape scenes and objects. The color of persons also keeps in a certain range, while the backgrounds and other artificial objects have much more diverse possibilities (3rd, 5th, 7th, 10th, 12th, 15th and 16th rows). For buildings (1st, 3rd and 14th rows) and trees/plants (2nd, 4th and 16th rows), the color diversity is adaptive to their categories and materials in particular.

Similar observations also emerge in Fig. 8 about luminance. The brightness for sunrise or sunset has the broadest range (6th, 11th, 13th and 15th rows), while it gets limited when the semantics imply that the environment lighting is stable (7th, 8th and 12th rows). The 5th row is a complicated case where the sun is covered by foreground persons. Then the outputs are more artistic in sky colors, while still preserving the persons not as exaggerated. For pure person portrait with a close-up shot on face (10th row), it varies in a restrained range to keep the face color reasonable, plus a special artistic case *i.e*. grayscale effect additionally. For buildings it also depends on their specific styles, *e.g*. modern buildings (1st row) differ from old buildings (3rd row) given the similar structures and layouts (buildings around with sky in the top middle).

References

[1] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015. 5

[2] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 5

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 5

[4] Matthias Wright and Björn Ommer. Artfid: Quantitative evaluation of neural style transfer. *GCPR*, 2022. 5

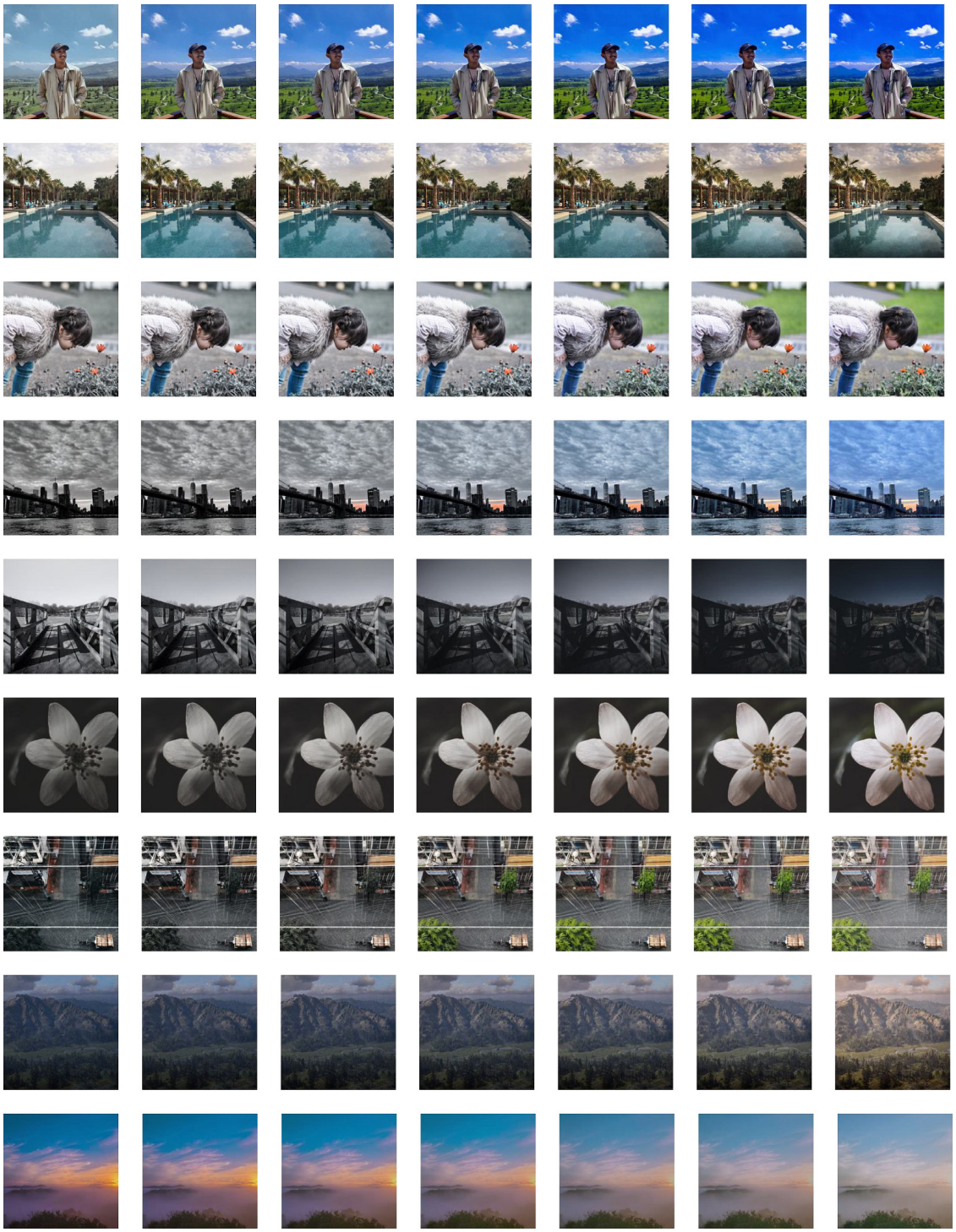

Figure 6. The interpolation results of our model. Two random Gaussian noise vectors are sampled to generate the first and last column, and their interpolated noise are used to generate the intermediate columns. It shows gradual changes from one color style to the other and indicates the arithmetic property of our learned latent noise space.

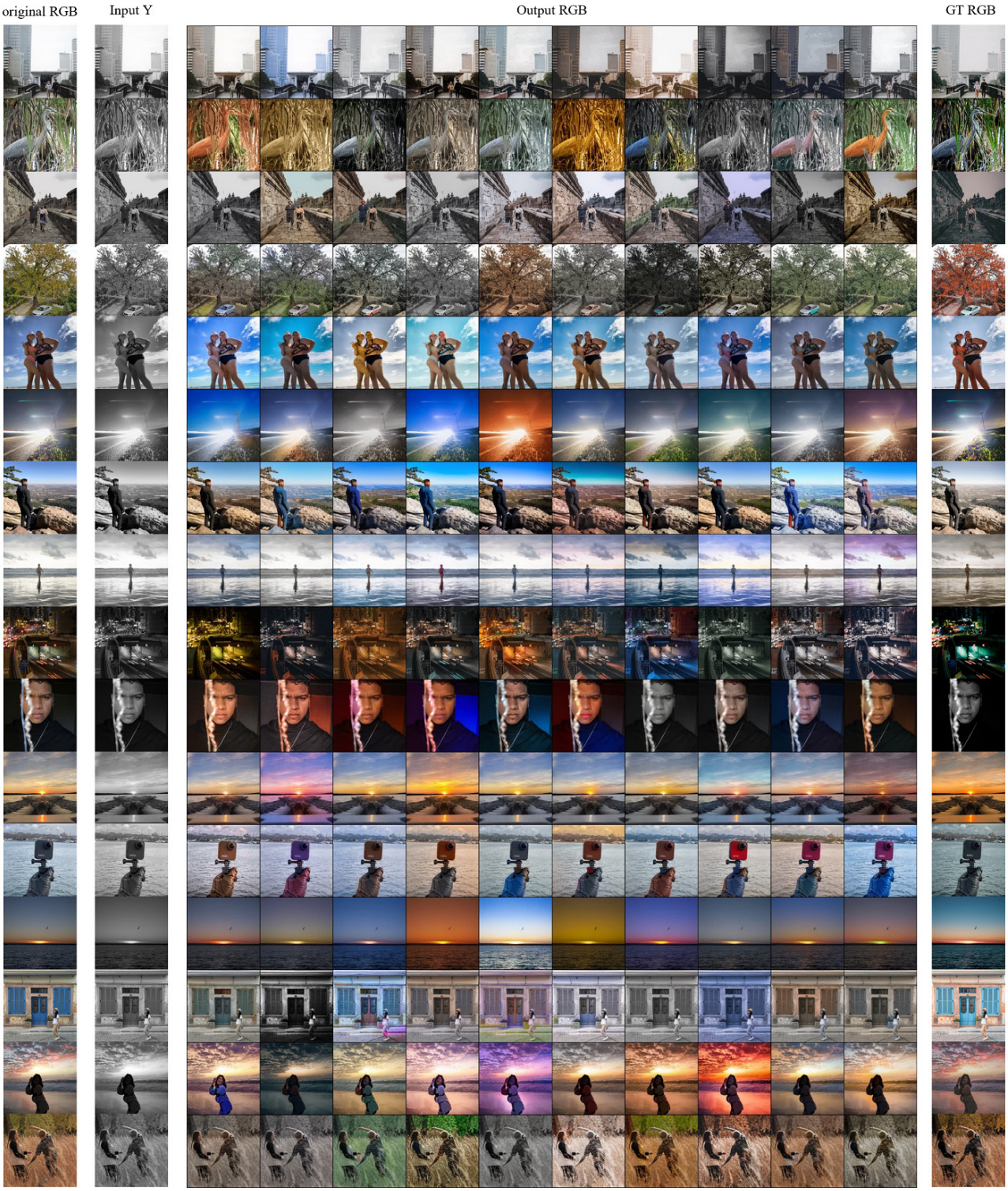

Figure 7. The output of the colorization (chrominance completion) task as one of our proposed auxiliary color restoration tasks. Note that only the 2nd column *Input Y* is the actual input. It shows that our model benefit from this auxiliary task to enhance the capability of producing more diversified and semantic-adaptive color, including the ones similar to the GT. See more details in Sec. E.

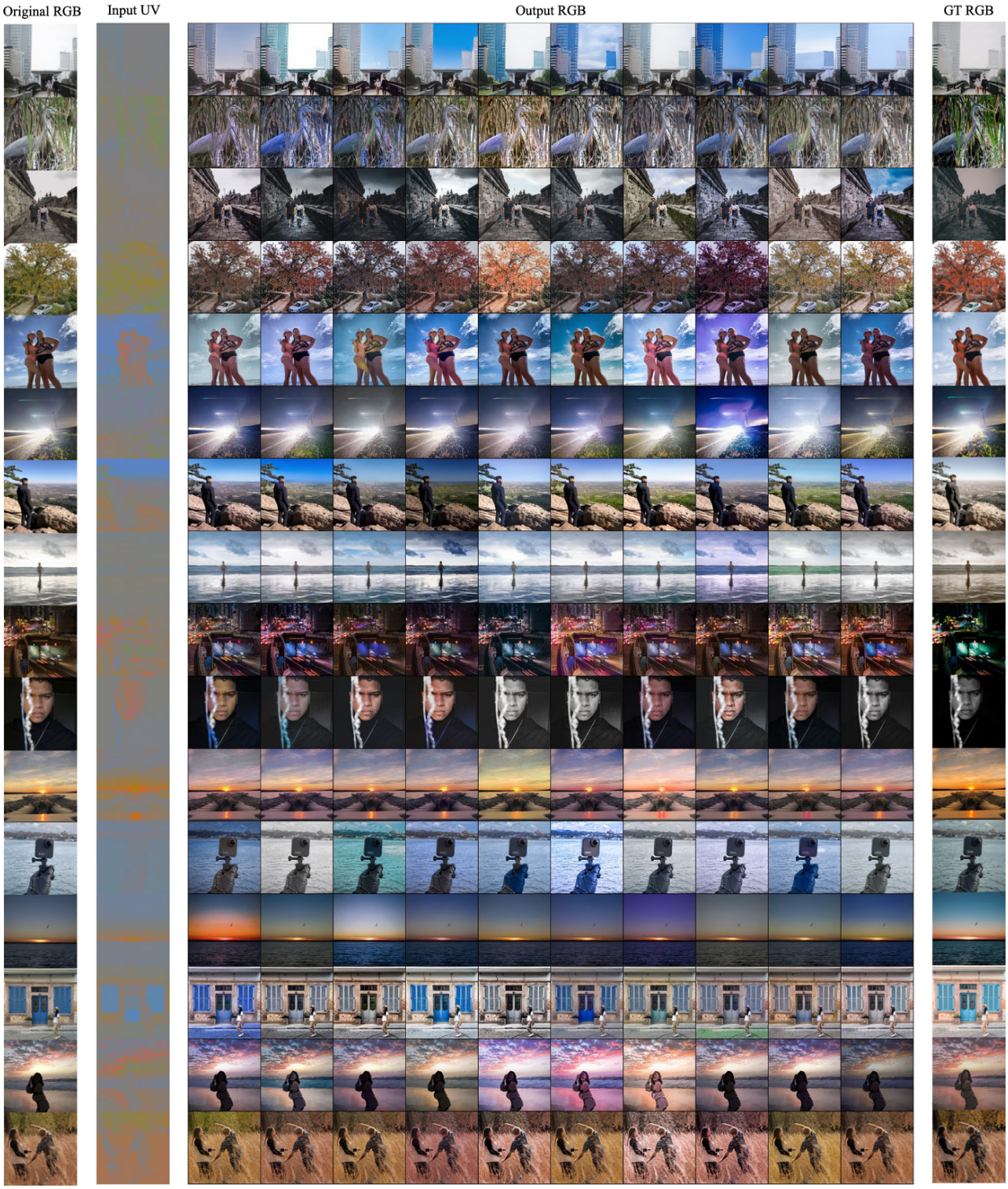

Figure 8. The output of the luminance completion task as one of our proposed auxiliary color restoration tasks. Note that here only the 2nd column *Input UV* is the actual input. It shows that our model benefit from this auxiliary task to enhance the capability of producing more diversified and semantic-adaptive lighting, including the ones similar to the GT. See more details in Sec. E.