# Technical Report for *Towards a Universal Image Degradation Model via Content-Degradation Disentanglement*

## Contents

# 1 Loss functions

## 1.1 Entropy Regularization Loss

**Similarities and differences to variational methods** Notably, the evidence lower bound (ELBO) used in Variational Autoencoders (VAEs) [1] also optimize the KL-divergence between the posterior latent distribution $q_\theta$ and a distribution $p_\phi$ where each latent entry is mutually independent:

$$\mathcal{L}_{\text{ELBO}} = -\Big[\mathbb{E}_{z\sim p_\phi(z\mid x)} \log q_\theta(x \mid z) - D_{\text{KL}}\big(p_\phi(z \mid x) \parallel q_\theta(z)\big)\Big] \tag{1}$$

However, there are some ground-breaking differences that make our method more suitable for some applications. First, entropy regularization loss provides more flexibility and can be used as a regularization term for other loss functions, as it is effectively a regularization term. In contrast, ELBO loss needs to act as the sole loss for optimizing a model. Second, it is more reasonable to assume the distortion information for a sampled image to also be a fixed vector, instead of an unpredictable quantity. However, VAEs need to assume the latent as a *random variable* for its calculation. Lastly, we believe the distribution of realistic degradations' latents is intractable to model. Instead of having a restrictive assumption on $p$ and $q$, and calculate the divergence term analytically, our method does not constraint the marginal distrobution of $p$ and $q$ and only focus on disentangling individual entries in $\boldsymbol{e}_\cdot$.

In this section, we summarize the rationale behind the loss functions used in our model that are not detailed in the main paper.

## 1.2 Perceptual Loss

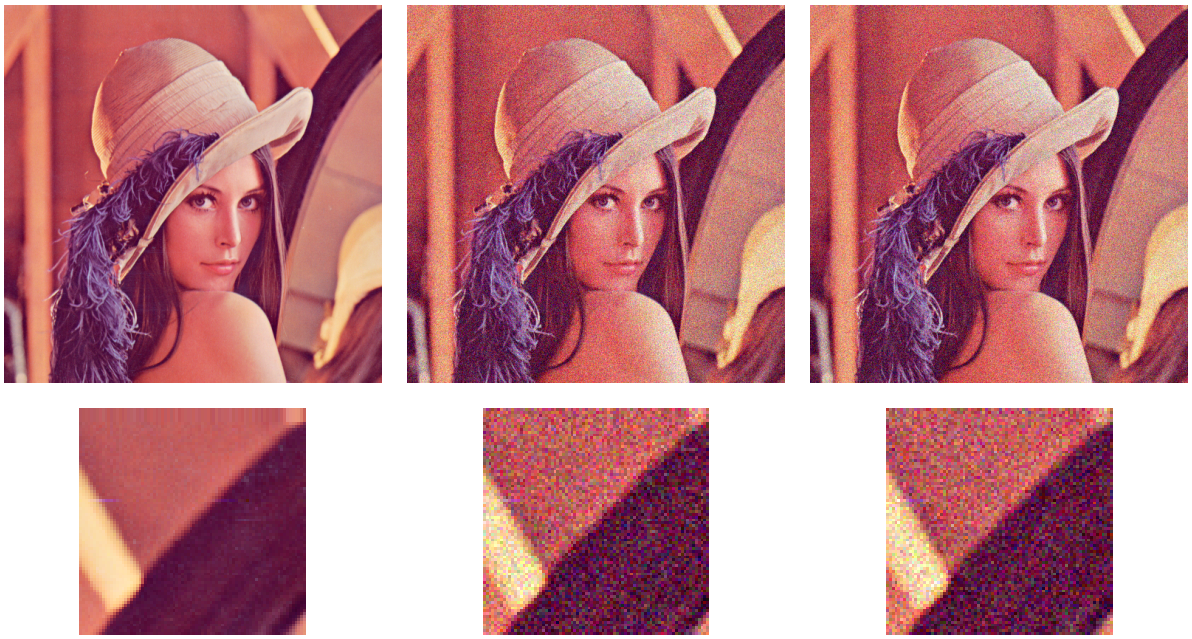
We observed that the random state in stochastic distortions, such as noise has very limited effects at human perception. However, most IQA metrics have strong preferences towards (a) exact reconstruction, and (b) blurry images, instead of images with similar patterns. We choose DISTS [2] as $d$, since it has the least severe undesirable bias. The construction of an IQA that eliminate these bias is reserved for future works. This section details the observations.

Most conventional Full-referenced IQA (FR-IQA) methods are based on measuring the similarity/ error between a target image $\boldsymbol{y}$ and a clean (reference or pristine) image $\boldsymbol{x}$. Many of the metrics are symmetric, meaning they carry out the same operation on $\boldsymbol{x}$ and $\boldsymbol{y}$, making $\text{IQA}(\boldsymbol{x}, \boldsymbol{y}) = \text{IQA}(\boldsymbol{y}, \boldsymbol{x})$ [2], [3], [4], [5], [6], [7], [8], [9], [10]. However, this equality does not hold in general for many other FR-IQAs [11], [12]. Since the objects whose distances are to be compared are symmetric in our application, the asymmetric methods are not suitable for our application of comparing a realistic distorted image with a simulated distorted image.

Since it is impractical to inspect all degradations, we shall focus on the theoretical performance of FR-IQAs in image degradation with stochastic effects, mainly noise. Symmetric FR-IQAs usually involves square-error-like components either directly [4] or through SSIM-like terms [2], [3], [5], [6], [7], [8], [9], [10]. These methods possess a common problem: *When comparing noisy images, they penalize images with similar noise patterns more than a "clean" image*, especially when applied to the spatial domain or an orthogonal transform domain (see Figure 1). Let $d$ denotes the IQAs in question, and assume $d(\boldsymbol{x}, \boldsymbol{y}_1) < d(\boldsymbol{x}, \boldsymbol{y}_2)$ means the model $d$ regard $\boldsymbol{y}_1$ as closer to $\boldsymbol{x}$ than $\boldsymbol{y}_2$.

In this section, we assume the noises to have zero mean. Let $\boldsymbol{y}_1 = \boldsymbol{x} + \boldsymbol{n}_1$ and $\boldsymbol{y}_2 = \boldsymbol{x} + \boldsymbol{n}_2$, where $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ may be dependent on $\boldsymbol{x}$, we further assume noise on the same location is independently generated (conditioning on $\boldsymbol{x}$), *i.e.*, $p\left(n_1^{(i,j)} n_2^{(i,j)} \,\middle|\, \boldsymbol{x}\right) = p\left(n_1^{(i,j)} \,\middle|\, \boldsymbol{x}\right) p\left(n_2^{(i,j)} \,\middle|\, \boldsymbol{x}\right)$.

MSE actually indicates $\boldsymbol{y}_1$ is more similar to $\boldsymbol{x}$ than to $\boldsymbol{y}_2$:



(a) Reference

(b) GN-11

(c) GN-12

SSIM(Ref, GN-11) = 0.59

SSIM(GN-11, GN-12) = 0.45

Figure 1: Indication of the issues with SSIM for capturing the similarity of noises. GN-11 and GN-12 are the same image applied with the same strength of Gaussian noise.

$$\begin{aligned}\mathbb{E}(\mathrm{MSE}(\boldsymbol{y}_1, \boldsymbol{y}_2)) &= \frac{1}{MN}\sum_{i,j}\mathbb{E}\left(y_1^{(i,j)} - y_2^{(i,j)}\right)^2 = \frac{1}{MN}\sum_{i,j}\mathbb{E}\left(n_1^{(i,j)} - n_2^{(i,j)}\right)^2 \\ &= \frac{1}{MN}\sum_{i,j}\mathrm{var}\left(n_1^{(i,j)}\right) + \mathrm{var}\left(n_2^{(i,j)}\right);\end{aligned} \tag{2}$$

$$\mathbb{E}(\mathrm{MSE}(\boldsymbol{y}_1, \boldsymbol{x})) = \frac{1}{MN}\sum_{i,j}\mathbb{E}\left(n_1^{(i,j)}\right)^2 = \frac{1}{MN}\sum_{i,j}\mathrm{var}\left(n_1^{(i,j)}\right). \tag{3}$$

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two image patches; let us consider SSIM [3]

$$\mathrm{SSIM}(\boldsymbol{x}, \boldsymbol{y}) = \frac{2\mu_{\boldsymbol{x}}\mu_{\boldsymbol{y}} + c_1}{\mu_{\boldsymbol{x}}^2 + \mu_{\boldsymbol{y}}^2 + c_1} \cdot \frac{2\sigma_{\boldsymbol{xy}} + c_2}{\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{y}}^2 + c_2}. \tag{4}$$

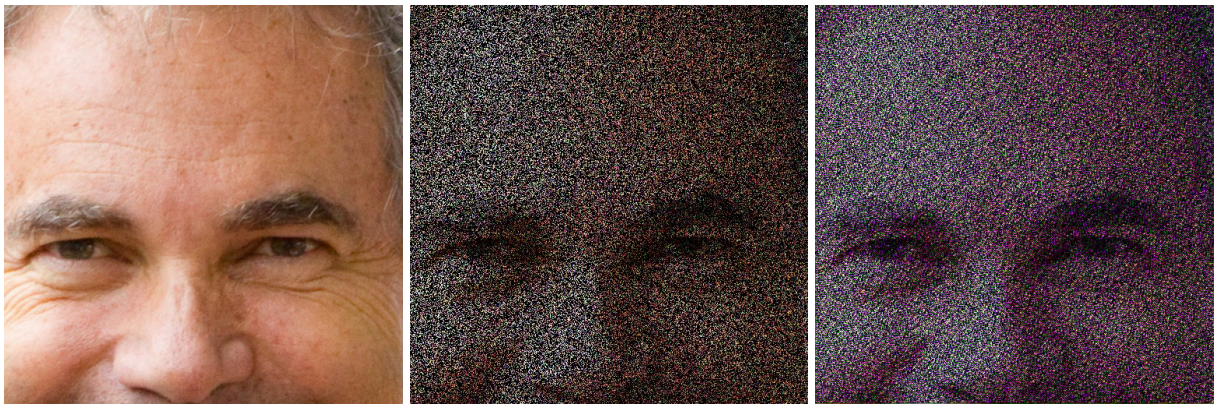
Denoting the second term as $c(\boldsymbol{x}, \boldsymbol{y})$, and assuming the patch is large enough such that we may approximate sample mean by distribution mean, we get

$$c(\boldsymbol{y}_1, \boldsymbol{x}_2) = \frac{2\sigma_{\boldsymbol{x}}^2 + c_2}{2\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{n}_1}^2 + c_2} \tag{5}$$

and

$$c(\boldsymbol{y}_1, \boldsymbol{y}_2) = \frac{2\sigma_{\boldsymbol{x}}^2 + c_2}{2\sigma_{\boldsymbol{x}}^2 + \sigma_{\boldsymbol{n}_1}^2 + \sigma_{\boldsymbol{n}_2}^2 + c_2}. \tag{6}$$

The above derivation shows that SSIM has the same issue as MSE in comparing two noisy images. Due to the embedded similar calculations, MS-SSIM [5], CW-SSIM [6], DISTS [2] (first level), and many other metrics are affected in the same way, but to a different extent.



(a) Clean image (b) Corresponding distorted image (c) Redistorted Image

Figure 2: Examples of color shift observed in result images when the color loss was not applied.

Based on our preliminary experiments, deep features are more robust to the change of random states, and DISTS [2] and LPIPS [4] are the best and second-to-the-best similarity measures among the aforementioned methods in this aspect, even though the spatial-domain global SSIM in DISTS has negative effects.

Although we have selected a similarity measurement after some compromise, selecting a good universal perceptual similarity measurement remains a considerable problem. For example, calculating the $\ell^2$ distance on a transform domain is unlikely to solve the aforementioned problem with degradation similarity in distance measurements, unless some manifolds in the spatial domain collapse in the transform domain. More generally, suppose we have a distance metric $d'$ that satisfies the "equivalence" property to the $\ell^2$ distance $d$, *i.e.*

$$cd'(a,b) \leq d(a,b) \leq C(a,b) \tag{7}$$

for some positive constants $c$ and $C$.

Let $\boldsymbol{y}_1 = \boldsymbol{x} + \boldsymbol{n}_1$, and $\boldsymbol{y}_2 = \boldsymbol{x} + N\boldsymbol{n}_2$, where $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ have zero-mean i.i.d. distributions. Then, we are still expecting

$$d'(\boldsymbol{x}, \boldsymbol{y}_1) > d'(\boldsymbol{y}_1, \boldsymbol{y}_2). \tag{8}$$

However,

$$\begin{aligned} \mathbb{E}[d'(\boldsymbol{x}, \boldsymbol{y}_1)]^2 &\leq C^2 \mathbb{E}[d(\boldsymbol{x}, \boldsymbol{y}_1)]^2 = \frac{C^2}{1+N^2} \mathbb{E}[d(\boldsymbol{y}_1, \boldsymbol{y}_2)]^2 \\ &\leq \frac{C^2}{(1+N^2)c^2} \mathbb{E}[d'(\boldsymbol{y}_1, \boldsymbol{y}_2)]^2. \end{aligned} \tag{9}$$

As long as $1 + N^2 > \frac{C^2}{c^2}$, the desirable condition (8) will be violated.

## 1.3 Color loss

With the other loss functions mentioned in the main task, the model is able to reproduce realistic degradations, but we observed significant color shift in some training examples (Figure 2). To suppress the unwanted color distortions, we add a color loss in a similar manner as proposed in [13]:

$$\mathcal{L}_{\text{color}} = d'(b(\boldsymbol{y}), b(\hat{\boldsymbol{y}})), \tag{10}$$

where $d'$ is another image distance measurement and $b$ is a Gaussian blur operation with very large radius. After applying the low-pass filter, most texture information on $\boldsymbol{y}$ will be lost, and the only thing to be compared is the color information. Since there is almost no stochastic nature in the very

blurry images, the problems of most IQAs in handling stochastic effects (see discussions in the previous subsection) are no longer of concern. Hence, we select $d(\boldsymbol{I}_1, \boldsymbol{I}_2)$ to be the $\ell^1$ distance of $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ after normalizing both images.

## 2 Detailed rationalization of model architecture

To construct an implicit degradations model, we note that all homogeneous degradations can be described by the implicit model:

$$\boldsymbol{y} = f\big(\boldsymbol{x}, \boldsymbol{e}_g, \boldsymbol{n}\big), \tag{11}$$

where $\boldsymbol{x}$ is a pristine image, $\boldsymbol{e}_g$ is the description of degradation type, order and parameters, $\boldsymbol{n}$ is a random state, and $f$ is an operator performing distortions based on the description. We further introduce a degradation map $\boldsymbol{e}_l$ to describe degradations that vary though spatial locations (*inhomogeneous degradations*):

$$\boldsymbol{y} = f\big(\boldsymbol{x}, \boldsymbol{e}_g, \boldsymbol{e}_l, \boldsymbol{n}\big). \tag{12}$$

A network $\hat{f}$ is trained to simulate $f$, and $\boldsymbol{e}_g \coloneqq e_g(\boldsymbol{y}) \in \mathbb{R}^{n_g}$ and $\boldsymbol{e}_l \coloneqq e_l(\boldsymbol{y}) \in \mathbb{R}^{c_l \times \frac{h}{r} \times \frac{w}{r}}$ are predicted by the *homogeneous degradation embedding* (HDE) network $e_g$ and the *inhomogeneous degradation embedding* (IDE) network $e_l$:

$$\hat{\boldsymbol{y}} \coloneqq \hat{f}\big(\boldsymbol{x}, e_g(\boldsymbol{y}), e_l(\boldsymbol{y}), \boldsymbol{n}\big). \tag{13}$$

During the training phase, the distorted image $\boldsymbol{y}$ is used as a ground-truth for distorting its corresponding pristine image $\boldsymbol{x}$. Our disentangle-by-compression technique is able to segregate image degradation information from semantics, which allows us to transfer distortions from $\boldsymbol{y}$ to a pristine image $\boldsymbol{x}$ with different contents during the test phase.

In contrast to existing models, our model has no degradation type- or order-specific structure. With single time of training, out model is applicable to any types and combinations of distortions in the training set. The robust degradation transfer ability allows our model to be applied in GAN inversion-based image restriction tasks.

# 3 More visuals

## 3.1 Degradation encoding and transfer

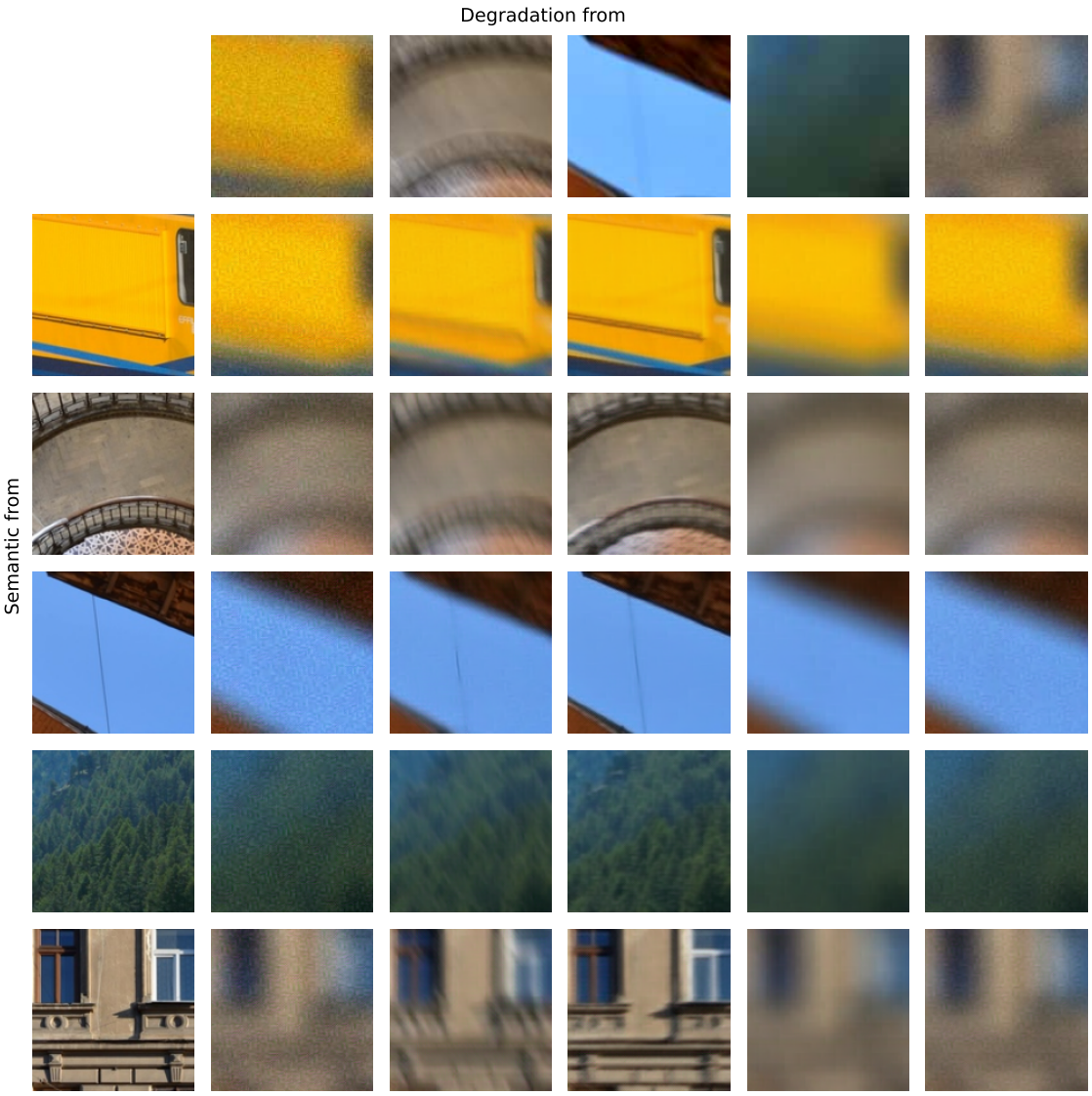



Figure 3: The result of distortion encoding and transfer. In the first row, five distorted image patches ($\boldsymbol{y}^{(j)}$'s) containing different image degradations are shown. In the first column are their corresponding pristine images $\boldsymbol{x}^{(i)}$'s. In each remaining grid is the synthetic distorted image generated by transferring distortions from $\boldsymbol{y}^{(j)}$ to $\boldsymbol{x}^{(i)}$.

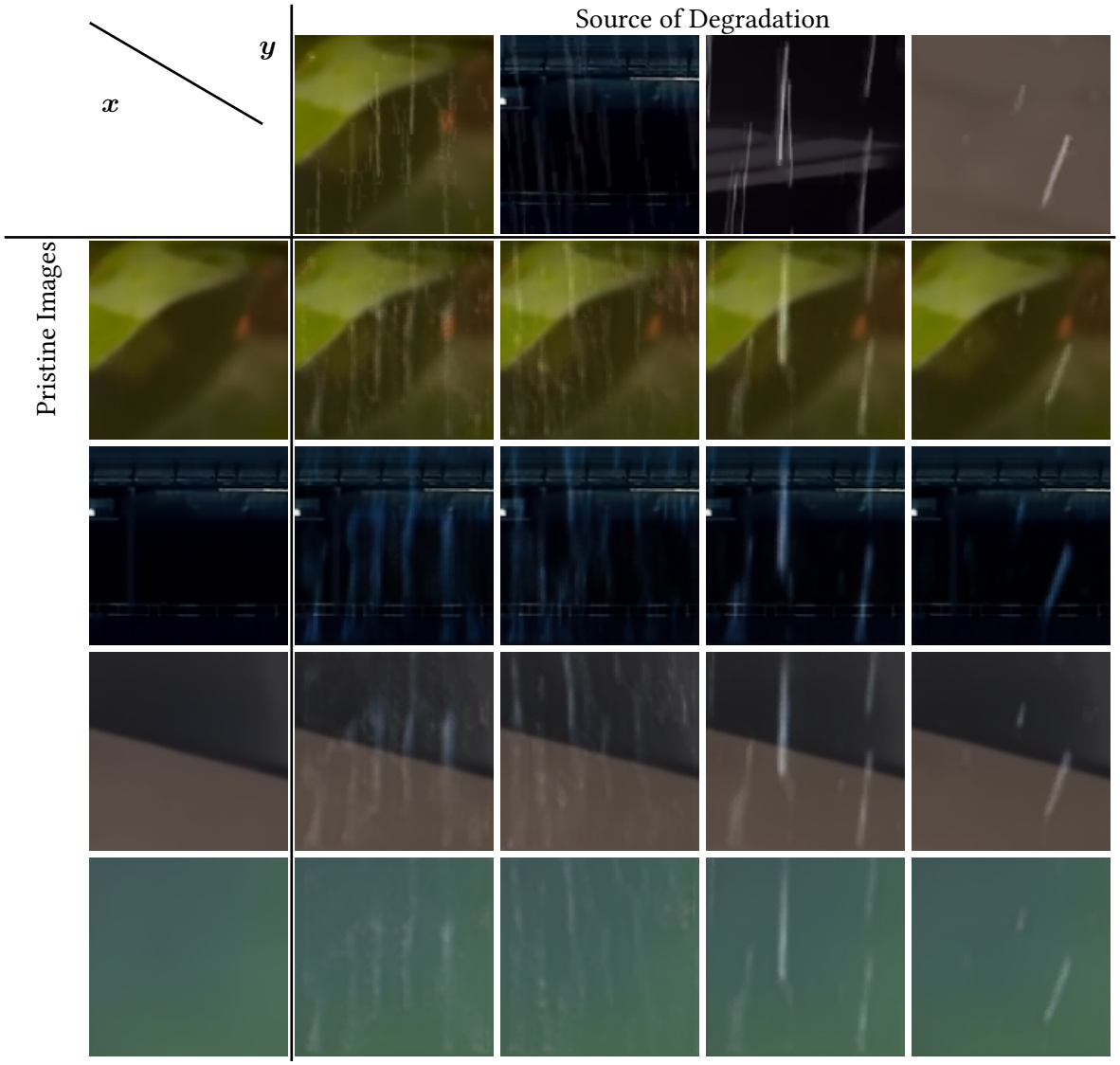



Figure 4: The result of distortion transfer for the realistic rain drop dataset [14]. The meaning of each grid is the same as Figure 3

## 3.2 Extra experiments on real-world datasets


Figure 5: The result of testing our model on reproducing motion blur on the GoPro [15] dataset.

We performed new experiments on the GoPro dataset [15] for real-world motion blur to test its distortion reproduction performance. Our model achieved a MS-SSIM score of 0.94 (see Figure 5 for visual samples).

## 3.3 Disentangling of latent entries

By analyzing variances, we identified five active dimensions in $e_g$ for the model trained on the distorted WQI dataset. To demonstrate disentanglement, we modified each dimension of the degradation embedding for a pristine test image. All active dimensions are shown in Figure 6. Most degradation embedding dimensions control multiple degradations simultaneously, aligning with our understanding of natural biases and masking effects in the dataset.

As an example for why each dimension should not control one distortion, if a dimension controls noise and another controls blur level after noise, in an image with high blur level, the *observed* noise level is always low, which makes the two variable highly dependent to each other. A simulated example when each latent controls a single distortion is shown in Figure 7.

## 3.4 Ablation studies

While the ablation study results for disentangle-by-compression is only obvious through the quantitative results, the one for IDA and IDEN is more visually apparent.

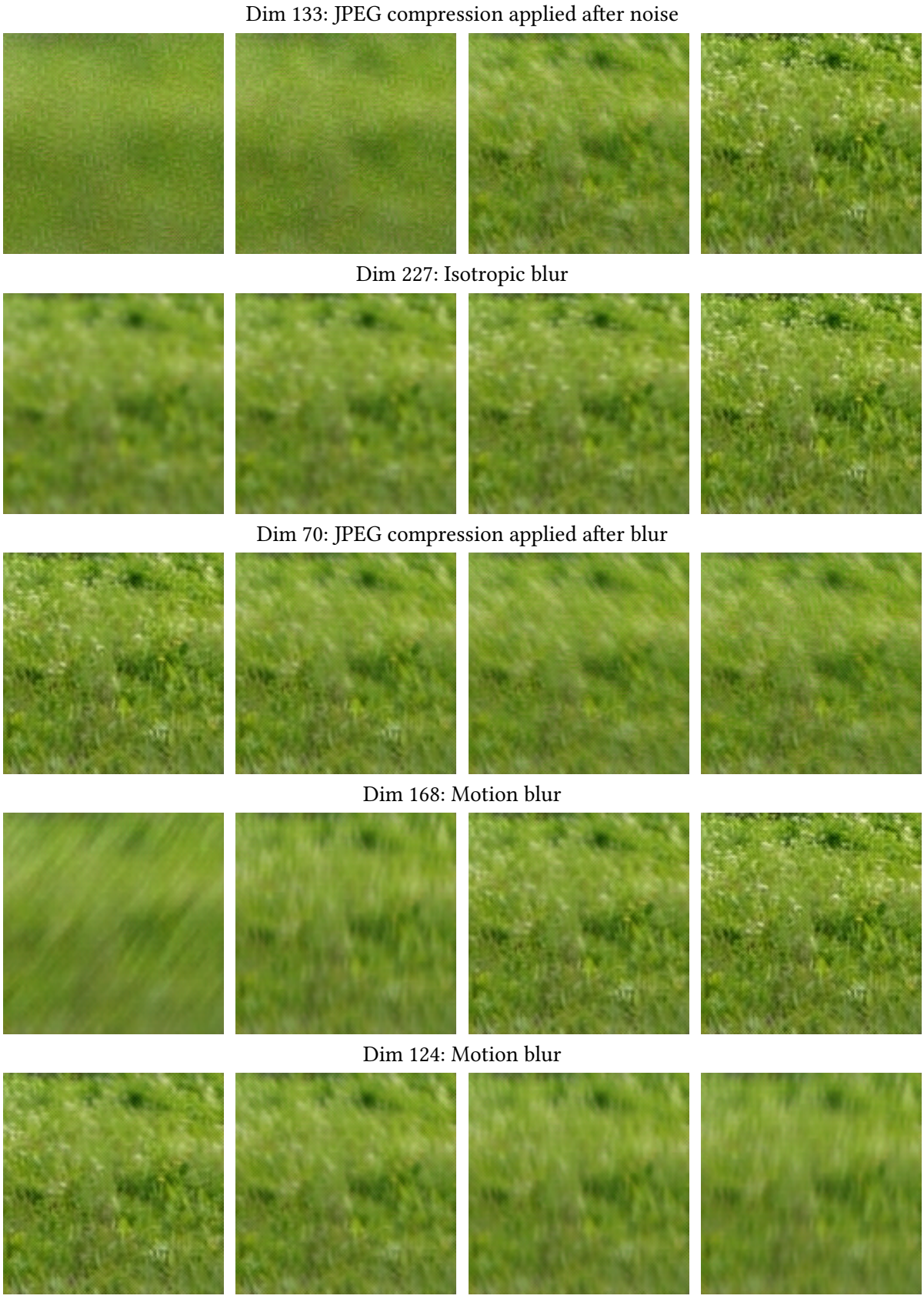



Figure 6: Demonstration of degradation latent disentangling by varying single entry in $\boldsymbol{e}_g$ on a every active dimension. In each row, an active dimension of the degradation embedding is varied by a fixed amount. The dimensions are ordered based on their standard deviations. The degradations a single dimension controls is labeled on the figure.

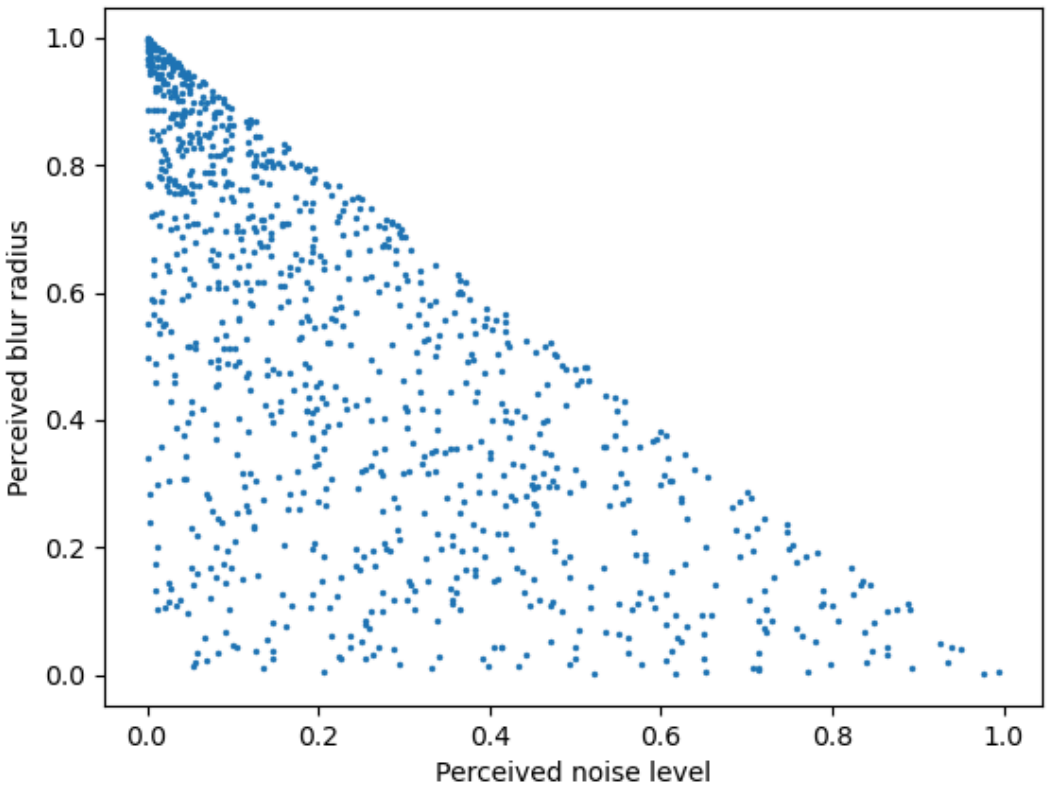



Figure 7: Simulated scenerio if one dimension in $e_g$ controls blur kernel's size and the other one controls the noise level. As indicated in the figure, the observed noise level and blur kernel size are highly dependent on each other.

| Pristine | Source of Homogeneous Deg. | Source of Inhomogeneous Deg. | Full Model | No IDA | No IQA, No IDEN |
|---|---|---|---|---|---|
| Pristine | Source of Homogeneous Deg. | Source of Inhomogeneous Deg. | Full Model | No IDA | No IQA, No IDEN |

Figure 8: The result of ablation studies for IDA and IDEN.

Figure 8 shows some examples in our ablation testset. While the model without IDA and IDEN is totally incapable of capturing inhomogeneous degradation, the model without IDA mainly struggles in reproducing inhomogeneous degradation when combined with a HDE from an image without

local degradation. This indicates that the model without IDA is less capable at disentangling inhomogeneous degradation information from homogeneous.

### 3.5 Comparison with StyTr2


Figure 9: Visual results for StyTr2 [16] (retrained on the distortion dataset) and compared with our model. StyTr2 [16] fails at separating distortion from content, hence, alters skin tone and texture while not transferring distortions.

Extra visual results for the comparison with StyTr2 [16] are shown in Figure 9. The model fails to transfer distortions from the distorted image to the pristine image, and instead alters the skin tone and texture.

### 3.6 Inversion-based image restoration

See Section 5.3 for the more visualizations of GAN inversion-based image restoration and Section 5.4 for the visuals of diffusion inversion-based image restoration.

## 4 Details of the Proposed Dataset

We collected around 300K Wikimedia Quality Images (WQIs) [17], and filtered out non-photographic images. We then applied JPEG compression after zero or more rounds of the following degradations of varying strengths (the probability of applying each degradation and the possible parameters are available in the supplied code):

- Gaussian noise to simulate sensor noise,
- Gaussian blur to simulate out-of-focus,
- Motion blur to simulate camera shake,
- contrast adjustment to simulate exposure change.

These combined into 16 different combinations of degradations. The code for collecting and processing the dataset is available at the `code_appendix/wm/` directory.

# 5 Details of the Experiments

## 5.1 Training Environment

We trained our model on a cluster. Each of its nodes is equipped with

- two Intel Gold 6148 Skylake CPUs,
- 186 G RAM (although we only used a fraction of it), and
- 4 NVIDIA V100 GPUs (16GB GRAM each).

Each of them runs a customized Rocky Linux 8.

## 5.2 Details of the Training Procedure

The training code and parameters are available in the `code_appendix/train/` directory with identifying details redacted. Each model is trained for 1 day for those on FFHQ and 3 days for others.

All other details about training are available in the code appendix.

## 5.3 GAN-inversion-based image restoration

The generation network of StyleGAN contains $N_L$ layers, and each layer contains $N_F$ AdaIN layers [18], each of which accepts a *style vector* $\boldsymbol{s}_i^l \in \mathbb{R}^{512}$ generated by passing $\boldsymbol{w}$ through an affine mapping $\boldsymbol{s}_i^l = A^l(\boldsymbol{w})$, where $\boldsymbol{w} = f(\boldsymbol{z}) \in \mathcal{W} = \mathbb{R}^{512}$. Some models [19], [20], [21], [22], [23] make use of the property, and introduce the $\mathcal{W}^+$ space, where $\boldsymbol{w}^+ \in \mathbb{R}^{N_L \times 512}$ and generate $\boldsymbol{s}_i^l = A^l\left(\left(\boldsymbol{w}^+\right)^l\right)$. Poirier-Ginter and Lalonde [24] further expanded the idea by progressively optimizing on the $\mathcal{W}$, $\mathcal{W}^+$, and $\mathcal{W}^{++}$ space, where $\mathcal{W}^{++} \ni \left(\boldsymbol{w}^{++}\right) \in \mathbb{R}^{N_F \times N_L \times 512}$ and $\boldsymbol{s}_i^l = A^l\left(\left(\boldsymbol{w}^{++}\right)_i^l\right)$.

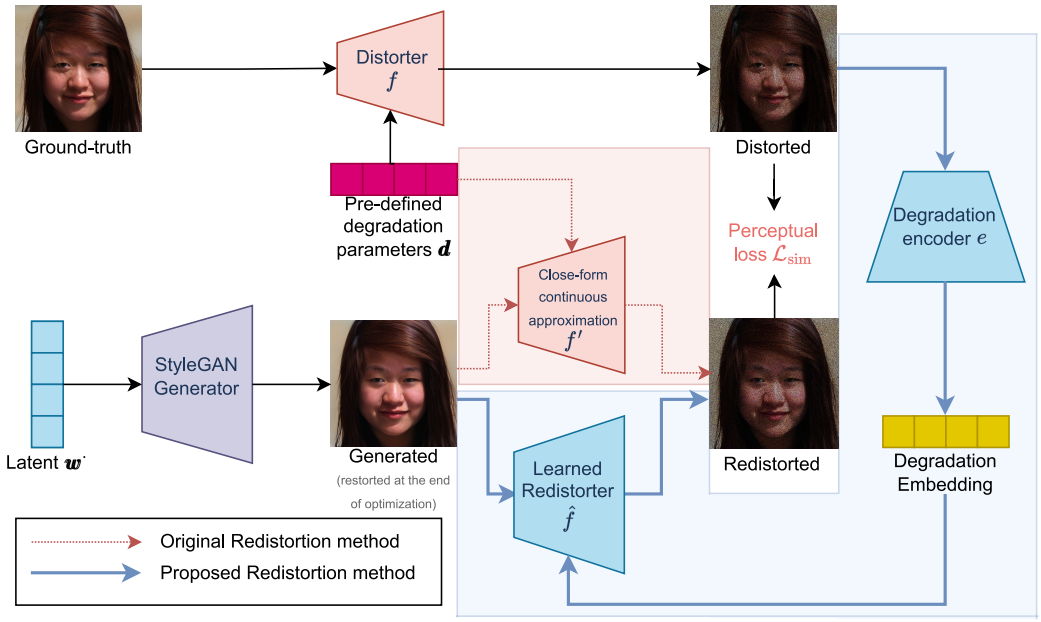



Figure 10: The GAN inversion procedure in [24] and the proposed modifications. In the original design (highlighted in red) uses a closed form approximation of the degradations presented in the distorted image. The degradation parameters (*e.g.* the strength of noise) are known at restoration time. In our modified design (highlighted in blue), the closed-form degradation approximation is replaced by our degradation encoding-decoding structure.

The optimization procedure in [24] is divided into three phases to achieve the final goal

$$\min_{\boldsymbol{w}^{++}} \text{LPIPS}\Big(\boldsymbol{y}, D_{\theta_d}(G(\boldsymbol{w}^{++}))\Big), \tag{14}$$

where $D_{\theta_d}$ is the continuous approximation of the degradation operation defined by the known parameter $\theta_d$. During the first phase (the $\mathcal{W}$ phase), the latent $\boldsymbol{w}^{++} \in \mathbb{R}^{N_F \times N_L \times 512}$ is set to be the expansion of $\boldsymbol{w} \in \mathbb{R}^{512}$, *i.e.*, $(\boldsymbol{w}^{++})_i^l = \boldsymbol{w}$, and in the second phase (the $\mathcal{W}^+$ phase), $\boldsymbol{w}^{++}$ is set to be the expansion of $\boldsymbol{w}^+ \in \mathbb{R}^{N_L \times 512}$, *i.e.*, $(\boldsymbol{w}^{++})_i^l = \boldsymbol{w}_i^+$, for each $l$. In the last phase (the $\mathcal{W}^{++}$ phase), each entry in $\boldsymbol{w}^{++}$ is optimized freely.

To mitigate its drawback of the necessity of knowing degradations during restoration, we introduce our model as the redegradator, replacing the known differentiable approximation of the degradation process used in [24] (see Figure 10). We retrained our model on the FFHQ dataset [25], which is the training set of StyleGAN [26] that is used in [24]. We then used the trained degradation model to replace $D_{\theta_d}$, and modified the objective as

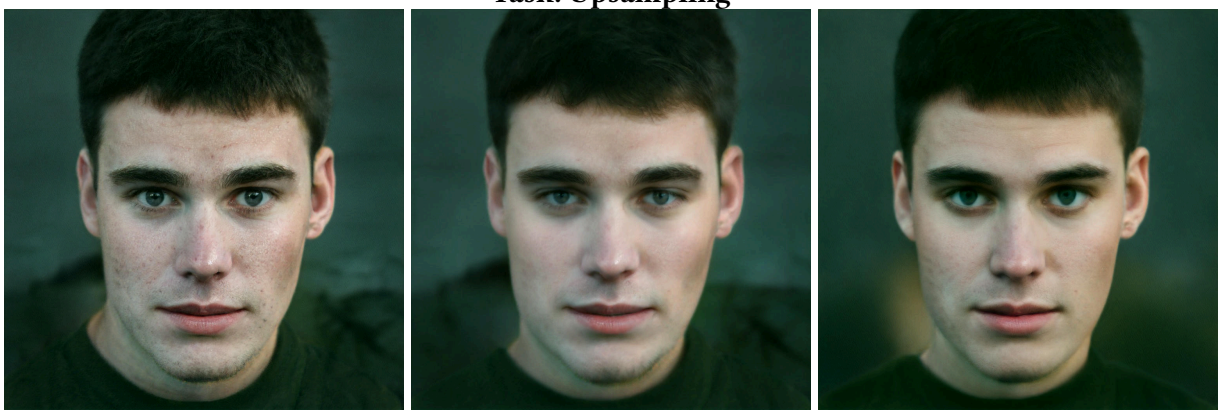
$$\min_{\boldsymbol{w}^{++}, \boldsymbol{e}_g, \boldsymbol{e}_l, \boldsymbol{n}} \text{LPIPS}\Big(\boldsymbol{y}, \hat{f}\big(G(\boldsymbol{w}^{++}), \boldsymbol{e}_g, \boldsymbol{e}_l, \boldsymbol{n}\big)\Big), \tag{15}$$

where $\boldsymbol{e}$ is the degradation embedding, $\boldsymbol{n}$ is the random state, and $\hat{f}$ is the redegradation module. Inspired by [24]'s design choice, we initialize $\boldsymbol{n} \sim \mathcal{N}(0, I)$, as expected by $\hat{f}$, and initialize $\boldsymbol{e} = e(\boldsymbol{y})$. In addition, we further introduce a $\mathcal{V}$ phase before the $\mathcal{W}$ phase, in which only the $\boldsymbol{w}$ latent to the StyleGAN generator is optimized, and $\boldsymbol{e}$ and $\boldsymbol{n}$ are frozen. In the subsequent phases, $\boldsymbol{e}$ and $\boldsymbol{n}$ are optimized with $\boldsymbol{w}$, $\boldsymbol{w}^+$ and $\boldsymbol{w}^{++}$. In practice, we initialize $\boldsymbol{e}_d = e(\boldsymbol{y})$ at the early stages of [24]'s optimization, and optimize $\boldsymbol{e}_d$ along with $\boldsymbol{w}^+$ and $\boldsymbol{w}^{++}$ in later optimization stages.

### 5.3.1 Visualizations

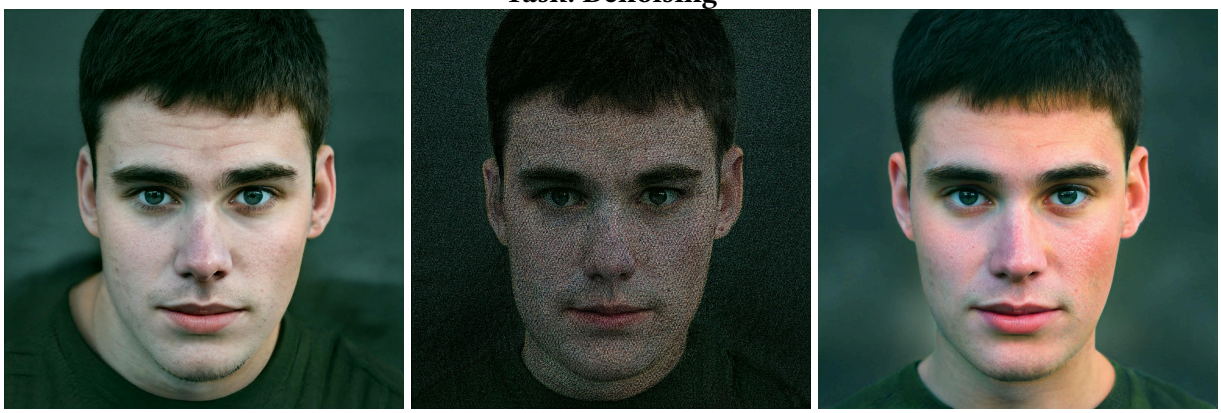
In Figure 25, we show examples for all tasks generated by RSG [24], the naive blind GAN inversion-based image restoration algorithm, and our proposed solutions. For single tasks, only level "L" is shown. It can be clearly seen from the examples that our model is able to restore a realistic face for most tasks, similar to [24]. RSG [24] also struggles on examples that our model struggles, suggesting the imperfectness does not come from our modifications. However, the naive blind method fails, especially at tasks involving noise and inpainting, generating significantly annoying and sometimes terrifying artifacts.

**Task: Upsampling**



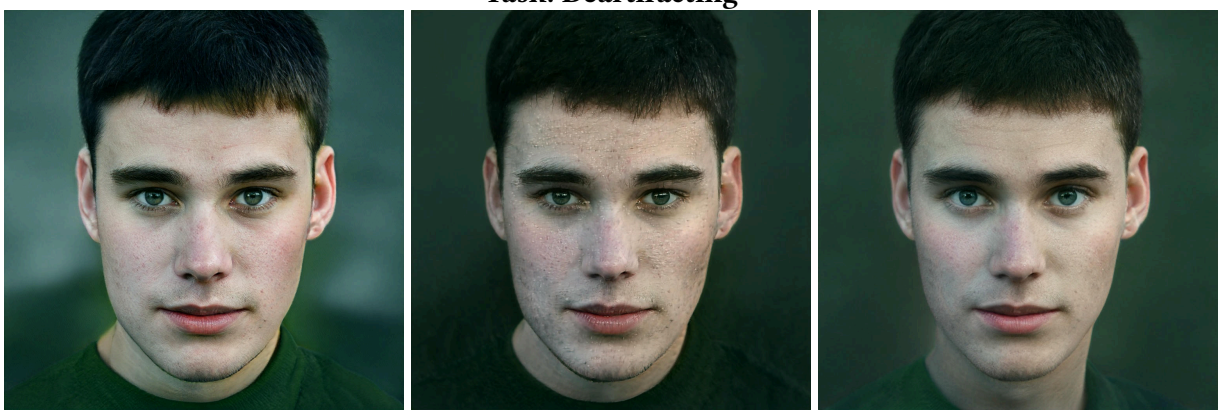
RSG DIR Ours

**Task: Denoising**



RSG DIR Ours

**Task: Deartifacting**



RSG DIR Ours

**Task: Inpainting**

RSG DIR Ours

**Task: NA**

RSG DIR Ours

**Task: AP**

RSG DIR Ours

**Task: UA**

RSG DIR Ours

**Task: NP**

RSG DIR Ours

**Task: UN**

RSG DIR Ours

**Task: UP**

RSG DIR Ours

**Task: UNP**

RSG DIR Ours

**Task: UAP**

RSG DIR Ours

**Task: UNA**

RSG DIR Ours

**Task: NAP**

RSG DIR Ours

**Task: UNAP**

RSG DIR Ours

Figure 25: Restoration resules from different algorithms.

### 5.3.2 Complete Quantitative Results

Table 1 shows the quantitative comparison of our algorithm against the naive blind GAN inversion-based image restoration algorithm (RSG [24] without degradation information and without our augmentation). The performance of non-blind GAN inversion-based image restoration algorithms (PULSE [19], L-BRGM [21], and RSG [24]) is also listed for reference. and the non-blind image restoration algorithms.

There are two important details that should be noted for interpreting the results:

1. The main contribution of PULSE [19], L-BRGM [21], and RSG [24] was to introduce optimization methods for GAN inversion-based image restoration. They are not designed to extract degradations. Instead, the re-degradation process was purely based on the recorded degradation parameters (when generating the test set). Hence, our augmented methods are inherently handicapped in comparison to these methods. Since **we did not attempted to improve the optimization method developed by RSG [24]**, a slight performance drop is expected from RSG, since our augmented method has **less available information** during restoration time (we are still able to outperform PULSE [19] and L-BRGM [21] on nearly all tasks). However, plugging in our model to RSG [24] allows us to apply GAN inversion-based image restoration to **more practical scenarios**.
2. While RSG augmented without our model appears to have slightly better performance as indicated in some categories' accuracy and fidelity scores, their results involves significantly artifacts in the restored images (as indicated by the bad realism score and visuals in Section 5.3.1), making them unusable in real-world applications.

| | Accur. (LPIPS) ↓ | | | | | Fidelity (LPIPS) ↓ | | | | | Realism (pFID) ↓ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Non-Blind | | | Blind | | Non-Blind | | | Blind | | Non-Blind | | | Blind | |
| | PUL | LBR | RSG | Augmented RSG | | PUL | LBR | RSG | Augmented RSG | | PUL | LBR | RSG | Augmented RSG | |
| | | | | w/o Ours | w/ Ours | | | | w/o Ours | w/ Ours | | | | w/o Ours | w/ Ours |
| **Upsampling (U)** | | | | | | | | | | | | | | | |
| XS | .493 | .407 | .414 | **.378** | .433 | .432 | .295 | .313 | **.258** | .356 | 44.5 | 23.6 | 17.0 | **12.8** | 20.8 |
| S | .492 | .412 | .449 | **.413** | .448 | .353 | .140 | .239 | **.178** | .279 | 34.3 | 25.5 | 22.0 | 22.5 | **16.9** |
| M | .495 | .458 | .472 | **.451** | .459 | .261 | .124 | .172 | **.109** | .202 | 29.3 | 35.4 | 22.3 | 42.0 | **28.8** |
| L | .501 | .487 | .490 | .487 | **.477** | .185 | .129 | .127 | **.073** | .121 | 21.9 | 26.0 | 20.9 | 45.5 | **41.0** |
| XL | .512 | .506 | .514 | .524 | **.512** | .083 | .095 | .090 | .070 | **.065** | 24.9 | 21.3 | 21.3 | **31.4** | 66.0 |
| **Denoising (N)** | | | | | | | | | | | | | | | |
| XS | .501 | .440 | .425 | .616 | **.452** | .275 | .152 | .156 | .292 | **.186** | 56.1 | 27.2 | 18.5 | 169.1 | **17.0** |
| S | .499 | .450 | .434 | .658 | **.484** | .252 | .138 | .140 | .306 | **.172** | 53.7 | 28.6 | 19.1 | 201.3 | **19.2** |
| M | .500 | .465 | .446 | .697 | **.482** | .224 | .155 | .130 | .307 | **.168** | 54.5 | 22.1 | 19.8 | 235.6 | **19.8** |
| L | .501 | .481 | .457 | .729 | **.504** | .185 | .138 | .110 | .325 | **.152** | 56.4 | 24.6 | 19.2 | 274.1 | **20.4** |
| XL | .504 | .511 | .474 | .742 | **.549** | .134 | .110 | .084 | .401 | **.127** | 49.4 | 25.1 | 17.9 | 318.6 | **25.2** |
| **Deartifacting (A)** | | | | | | | | | | | | | | | |
| XS | .498 | .442 | .432 | **.424** | .442 | .404 | .341 | .349 | **.340** | .366 | 52.3 | 26.3 | 14.8 | 25.7 | **16.4** |
| S | .497 | .448 | .437 | **.436** | .449 | .398 | .352 | .350 | **.345** | .369 | 49.6 | 22.4 | 15.4 | 32.3 | **16.6** |
| M | .498 | .461 | .445 | **.451** | .457 | .413 | .357 | .357 | **.352** | .377 | 33.2 | 24.1 | 15.4 | 41.7 | **16.9** |
| L | .500 | .475 | .460 | .472 | **.470** | .395 | .367 | .374 | **.361** | .389 | 46.9 | 25.2 | 16.0 | 57.5 | **18.6** |
| XL | .508 | .503 | .490 | .501 | **.495** | .427 | .418 | .412 | **.385** | .414 | 30.8 | 22.1 | 18.7 | 81.0 | **22.8** |
| **Inpainting (P)** | | | | | | | | | | | | | | | |
| XS | .498 | .409 | .378 | **.345** | .423 | .464 | .374 | .348 | **.314** | .391 | 46.9 | 24.4 | 12.9 | **10.9** | 17.2 |
| S | .501 | .425 | .387 | **.386** | .422 | .356 | .287 | .264 | **.254** | .295 | 42.3 | 27.2 | 14.2 | **15.0** | 17.5 |
| M | .509 | .438 | .396 | **.425** | .427 | .283 | .227 | .206 | **.207** | .231 | 38.5 | 30.1 | 14.5 | 22.0 | **17.4** |
| L | .513 | .452 | .409 | .464 | **.439** | .231 | .184 | .163 | **.176** | .183 | 32.6 | 33.1 | 15.3 | 29.8 | **18.2** |
| XL | .524 | .460 | .422 | .496 | **.446** | .187 | .157 | .132 | .152 | **.151** | 36.2 | 25.2 | 15.9 | 39.0 | **18.6** |
| **2 degradations** | | | | | | | | | | | | | | | |
| NA | .517 | .485 | .459 | .680 | **.513** | .328 | .301 | .290 | .424 | **.334** | 43.4 | 24.2 | 17.3 | 228.8 | **28.1** |
| AP | .511 | .478 | .457 | .506 | **.463** | .270 | .231 | .204 | .232 | **.219** | 29.7 | 17.6 | 17.0 | 45.1 | **16.9** |
| UA | .510 | .518 | .508 | .519 | **.512** | .307 | .348 | .287 | **.274** | .275 | 23.7 | 20.5 | 19.7 | 47.8 | **30.0** |
| NP | .511 | .480 | .458 | .713 | **.485** | .125 | .079 | .062 | .187 | **.081** | 47.0 | 20.9 | 19.2 | 221.9 | **20.6** |
| UN | .501 | .519 | .511 | .700 | **.564** | .178 | .149 | .153 | .273 | **.177** | 33.4 | 26.2 | 21.1 | 211.2 | **41.4** |
| UP | .510 | .478 | .485 | .541 | **.478** | .140 | .061 | .089 | .130 | **.097** | 23.9 | 35.3 | 20.7 | 57.7 | **38.8** |
| **3 degradations** | | | | | | | | | | | | | | | |
| UNP | .510 | .526 | .507 | .715 | **.590** | .086 | .062 | .051 | .156 | **.082** | 28.5 | 22.1 | 20.1 | 215.6 | **60.9** |
| UAP | .525 | .523 | .513 | .588 | **.526** | .205 | .154 | .119 | .169 | **.115** | 23.0 | 18.4 | 20.5 | 80.5 | **34.8** |
| UNA | .521 | .535 | .533 | .669 | **.569** | .265 | .310 | .290 | .407 | **.298** | 26.2 | 20.7 | 22.8 | 175.5 | **34.5** |
| NAP | .526 | .502 | .470 | .686 | **.508** | .210 | .197 | .160 | .282 | **.182** | 38.7 | 18.4 | 18.5 | 213.7 | **25.3** |
| **4 degradations** | | | | | | | | | | | | | | | |
| UNAP | .533 | .546 | .525 | .666 | **.560** | .192 | .177 | .131 | .251 | **.141** | 25.5 | 21.1 | 21.8 | 147.4 | **31.7** |

Table 1: Quantitative comparison of our algorithm against non-blind and the naive blind algorithm (DIR) on the restoration of images degraded by single/multiple distortions of different levels. Blind methods that perform the best in each category are shown in **bold fonts**.

### 5.4 Diffusion inversion-based image restoration


Figure 26: The result of plugging in our model to Diffusion Posterior Sampling (DPS) [27].

Due to the limited computational resources, we were not able to conduct experiments on DPS [27] on large-scale datasets for quantitative comparison. However, we conducted qualitative experiments on ImageNetV2 [28] (results are shown in Figure 26). The dataset is constructed to avoid testing images' presence in the training set of the inverted diffusion model.

## 6 Proofs

### 6.1 Regarding inhomogeneous degradation-aware layer

**Proposition.** Given a latent vector $\boldsymbol{F}_{\text{in}} \in \mathbb{R}^{c\times 2h\times 2w}$. For any combination of four $2\times 2$ depthwise convolution layers $\{C_k : k \in \mathbb{N}_4\}$, with stride 2 and spatially-varying kernel $W^{(k)} \in \mathbb{R}^{h\times w\times c\times 2\times 2}$, suppose there exists some $u^{(k)}_{pqr}$'s such that $W^{(k)}_{ijpqr} = u^{(k)}_{pqr} W^{(1)}_{ijpqr}$ for every $i \in \{2,3,4\}$. Then, there exists an $\boldsymbol{e} \in \mathbb{R}^{4c,h/2,w/2}$ such that

$$\left(\text{IDA}\ (\boldsymbol{F}_{\text{in}}, \boldsymbol{e})\right)_{p,2i+a,2j+b} = \left(C_{2a+b-2}(\boldsymbol{F}_{\text{in}})\right)_{p,i,j}, \tag{16}$$

for $a, b \in \{1,2\}$.

*Proof* For the convenience of notations, all matrices and tensors are indexed from 0. See Figure 27 for the visualization of the proof.

$$\text{IDA}(\boldsymbol{F}_{\text{in}}, \boldsymbol{e}) = \text{DConv}_2(\text{ConvDS}_1(\text{DConv}_1(\boldsymbol{e})) \odot \text{ConvDS}_2(\boldsymbol{F}_{\text{in}})) \tag{17}$$

Let the weight $W_{pqij} \in \mathbb{R}^{c_i\times c_o\times k\times k}$ for $\text{ConvDS}_2$ be

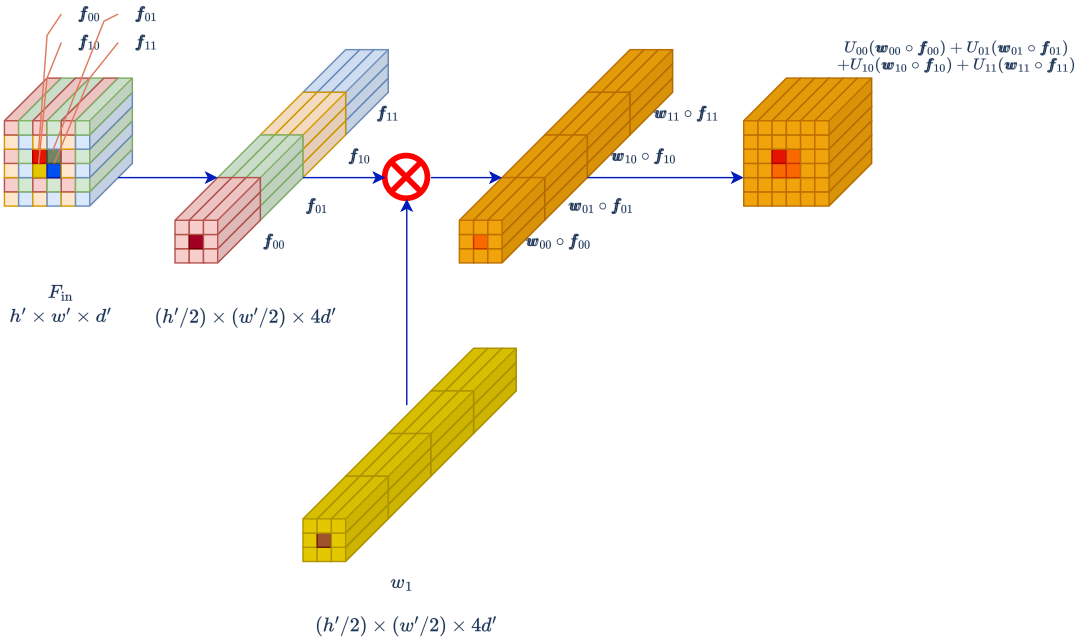



Figure 27: The rationalization of the proof of the proposition.

$$\begin{aligned} W_{p,4p,0,0} &= 1, \\ W_{p,4p+1,0,1} &= 1, \\ W_{p,4p+2,1,0} &= 1, \\ W_{p,4p+3,1,1} &= 1, \end{aligned} \tag{18}$$

where

$$c_o = 4c_i \tag{19}$$

. Let all other entries in $W$ be zero.

then,

$$f^{\text{out}}_{p,2i+k_1,2j+k_2} = f^{\text{interim}}_{4p+2k_1+k_2,i,j}. \tag{20}$$

Let $w_1 := \text{ConvDS}_1(\text{DConv}_1(\boldsymbol{e}))$. If $\text{ConvDS}_1$ and $\text{DConv}_1$ cancel each other, then we have $e = w$. Suppose we can construct $\boldsymbol{e}$ and $\text{ConvDS}_1$ and $\text{DConv}_1$ such that

$$w_{4p+2q+r,i,j} := W^{(0)}_{ijpqr}, \quad \text{where} \quad q, r \in \{0, 1\}. \tag{21}$$

Then, let $V \in \mathbb{R}^{c_o \times c_i \times 3 \times 3}$, the weight for $\text{DConv}_2$, be defined as:

$$V(p, 4p + 2k_1 + k_2, a, b) = u^{(2a+b)}_{p,k_1,k_2}, \text{ for } \quad k_1, k_2, a, b \in \{0, 1\}, \tag{22}$$

where $u^{(0)}_{p,k_1,k_2} = 1$ for $k_1, k_2 \in \{0, 1\}$, and $u^{(k)}_{p,k_1,k_2}$'s are defined as in the main paper for $k = 1, 2, 3$. Then

$$
\begin{aligned}
f^{\text{out}}_{p,2i+a,2j+b} &= \sum_{k_1,k_2} u^{(2a+b)}_{p,k_1,k_2} w_{4p+2k_1+k_2,i,j} f^{\text{interim}}_{4p+2k_1+k_2,i,j} \\
&= \sum_{k_1,k_2} u^{(2a+b)}_{p,k_1,k_2} w_{4p+2k_1+k_2,i,j} f^{\text{in}}_{p,2i+k_1,2j+k_2}
\end{aligned} \tag{23}
$$

Let $w_{4p+2k_1+k_2,i,j} = W^{(0)}_{ijpk_1k_2}$ Then,

$$
\begin{aligned}
f^{\text{out}}_{p,2i+a,2j+b} &= \sum_{k_1,k_2} u^{(2a+b)}_{p,k_1,k_2} W^{(0)}_{ijpk_1k_2} f^{\text{in}}_{p,2i+k_1,2j+k_2} \\
&= \sum_{k_1,k_2} W^{(2a+b)}_{ijpk_1k_2} f^{\text{in}}_{p,2i+k_1,2j+k_2} \\
&= \left(C_{2a+b}(f^{\text{in}})\right)_{p,i,j}
\end{aligned} \tag{24}
$$

What remains to be shown is that $\text{ConvDS}_1$ and $\text{DConv}_1$ can be constructed to cancel each other.

## 6.2 Regarding disentangle-by-compression

In the main paper, we claim that the disentangle-by-compression technique is able to achieve three effects:

1. separating degradation from distorted images contents,
2. segregating inhomogeneous from homogeneous distortions, and
3. disentangling individual degradation components.

We have fully demonstrated the third effect in the main paper. The remainder of this section is dedicated to proving first and second effects.

### 6.2.1 Separating degradation from images contents

We make the following assumptions:

- **A-1** With information about the degradation process and the clean image, the degraded image can be exactly reconstructed, albeit the difference of random state difference:

$$H(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{d}) = 0. \tag{25}$$

- **A-2** The distortion process is independent of the clean image:

$$I(\boldsymbol{x};\boldsymbol{d}) = 0. \tag{26}$$

- **A-3** The distortion can be inferred from the degraded image:

$$H(\boldsymbol{d}|\boldsymbol{y}) = 0. \tag{27}$$

- **A-4** When an appropriate perceptual similarity loss $\mathcal{L}_{\text{sim}} = d(\boldsymbol{y},\hat{\boldsymbol{y}})$ is used, the reconstructed distorted image $\hat{\boldsymbol{y}}$ is assumed to be similar enough to $\boldsymbol{y}$, *i.e.*, $\boldsymbol{y} \overset{\mu}{=} \hat{\boldsymbol{y}}$ (which means $H(\boldsymbol{y}|\hat{\boldsymbol{y}}) = H(\hat{\boldsymbol{y}}|\boldsymbol{y}) = 0$).

**Proposition** With the assumptions (**A-1**) to (**A-4**), the optimal solution for minimizing $H(\boldsymbol{e})$ in our model architecture is to minimize $I(\boldsymbol{e};\boldsymbol{x})$.

*Proof*

Since (25) and (27), we have[1]

$$H(\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{y}) + I(\boldsymbol{d};\boldsymbol{y}). \tag{28}$$

Using (26), we have[2]

[1] $I(\boldsymbol{x};\boldsymbol{d} \mid \boldsymbol{y}) = H(\boldsymbol{d} \mid \boldsymbol{y}) - H(\boldsymbol{d} \mid \boldsymbol{x},\boldsymbol{y}) = 0$; $I(\boldsymbol{x};\boldsymbol{d};\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{d}) - I(\boldsymbol{x};\boldsymbol{d} \mid \boldsymbol{y}) = 0$.

[2] $H(\boldsymbol{d}) = H(\boldsymbol{d} \mid \boldsymbol{y}) + I(\boldsymbol{d};\boldsymbol{y}) = I(\boldsymbol{d};\boldsymbol{y})$.

$$H(\boldsymbol{y}) = I(\boldsymbol{x}; \boldsymbol{y}) + H(\boldsymbol{d}). \tag{29}$$

However, by the generation process of $\hat{\boldsymbol{y}}$ ($\hat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{e})$) (albeit the difference of random state), we have

$$H(\hat{\boldsymbol{y}}) = I(\boldsymbol{x}; \hat{\boldsymbol{y}}) + I(\boldsymbol{e}; \hat{\boldsymbol{y}}) \tag{30}$$

Applying **A-4**, we have

$$H(\boldsymbol{d}) = I(\boldsymbol{d}; \boldsymbol{y}) = I(\boldsymbol{d}; \hat{\boldsymbol{y}}) = I(\boldsymbol{d}; \boldsymbol{e}) \tag{31}$$

Due to the generation process outlined in the main text, we know that

$$H(\boldsymbol{e} \mid \boldsymbol{x}, \boldsymbol{d}) = H(\boldsymbol{e} \mid \boldsymbol{y}) = 0. \tag{32}$$

Combining A-4, we have

$$\begin{aligned} H(\boldsymbol{e}) &= I(\boldsymbol{e}; \boldsymbol{x}) + H(\boldsymbol{e} \mid \boldsymbol{x}) \\ &= I(\boldsymbol{e}; \boldsymbol{x}) + I(\boldsymbol{d}; \boldsymbol{e} \mid \boldsymbol{x}) + H(\boldsymbol{e} \mid \boldsymbol{x}, \boldsymbol{d}) \\ &= I(\boldsymbol{e}; \boldsymbol{x}) + I(\boldsymbol{d}; \boldsymbol{e} \mid \boldsymbol{x}) = I(\boldsymbol{e}; \boldsymbol{x}) + I(\boldsymbol{d}; \boldsymbol{e} \mid \boldsymbol{x}) + I(\boldsymbol{d}; \boldsymbol{x}) + H(\boldsymbol{d} \mid \boldsymbol{x}, \boldsymbol{e}) \\ &= I(\boldsymbol{e}; \boldsymbol{x}) + H(\boldsymbol{d}). \end{aligned} \tag{33}$$

Since $H(\boldsymbol{d})$ is a fixed (but unknown) constant, by minimizing $H(\boldsymbol{e})$, we are minimizing $I(\boldsymbol{e}; \boldsymbol{x})$, which encourages $\boldsymbol{e}$ to contain no information in $\boldsymbol{x}$, disentangling the degradation from image content.

*Q.E.D.*

#### 6.2.2 Segregating inhomogeneous and homogeneous distortions

We further add one more assumption: (**A-5**) there exists two kinds of degradations. The homogeneous one $\boldsymbol{d}_g$ is the same for all geometry location, and the inhomogeneous one $\boldsymbol{d}_l^{(i)}$ is different for different geometry location $(i)$, meaning that $\boldsymbol{d}_l^{(i)}$ is independent of $\boldsymbol{d}_l^{(j)}$ for $i \neq j$.

**Proposition** Let $\boldsymbol{e}_g$ and $\boldsymbol{e}_l^{(i)}$ denote the properly-learned degradation representations at location $(i)$. With the assumptions (**A-1**) to (**A-5**), the optimal solution for $\mathcal{L} = kH(\boldsymbol{e}_g) + \sum_i H\left(\boldsymbol{e}_l^{(i)}\right)$, where $1 < k < n$, satisfies the following properties:

1. Each $\boldsymbol{e}_l^{(i)}$ only contains degradation information about the corresponding geometry location, and $\boldsymbol{e}_g$ does not contain any location-specific information: $\left\{\boldsymbol{e}_1^{(1)}, ..., \boldsymbol{e}_n^{(n)}, \boldsymbol{e}_g\right\}$ are jointly independent.
2. Each $\boldsymbol{e}_l^{(i)}$ and $\boldsymbol{e}_g$ contains just enough information to reconstruct the corresponding degradation:

$$H(\boldsymbol{d}_g) = H(\boldsymbol{e}_g), \qquad H\left(\boldsymbol{d}_l^{(i)}\right) = H\left(\boldsymbol{e}_l^{(i)}\right). \tag{34}$$

*Proof* Based on (**A-1**), (**A-3**) and (**A_4**) and out network architecture, we know that:

- For the whole image, $H\left(\boldsymbol{d}_g, \boldsymbol{d}_l^{(1)}, ..., \boldsymbol{d}_l^{(n)} \mid \boldsymbol{e}_g, \boldsymbol{e}_l^{(1)}, ..., \boldsymbol{e}_l^{(n)}\right) = 0$
- A spatial location $(i)$, $H\left(\boldsymbol{d}_g, \boldsymbol{d}_l^{(i)} \mid \boldsymbol{e}_g, \boldsymbol{e}_l^{(i)}\right) = 0$

Hence, we have the following inequalities:

$$\begin{aligned} H(\boldsymbol{e}_g) + \sum_{i=1}^{n} H\left(\boldsymbol{e}_l^{(i)}\right) &\geq H\left(\boldsymbol{e}_g, \boldsymbol{e}_l^{(1)}, ..., \boldsymbol{e}_l^{(n)}\right) \\ &\geq H\left(\boldsymbol{d}_g, \boldsymbol{d}_l^{(1)}, ..., \boldsymbol{d}_l^{(n)}\right) = H(\boldsymbol{d}_g) + \sum_{i=1}^{n} H\left(\boldsymbol{d}_l^{(i)}\right) \end{aligned} \tag{35}$$

$$H(\boldsymbol{e}_g) + H\left(\boldsymbol{e}_l^{(i)}\right) \geq H\left(\boldsymbol{e}_g, \boldsymbol{e}_l^{(i)}\right) \geq H\left(\boldsymbol{d}_g, \boldsymbol{d}_l^{(i)}\right) = H(\boldsymbol{d}_g) + H\left(\boldsymbol{d}_l^{(i)}\right) \tag{36}$$

The first inequality in (35) becomes an equality iff $\left\{\boldsymbol{e}_1^{(1)}, ..., \boldsymbol{e}_n^{(n)}, \boldsymbol{e}_g\right\}$ are jointly independent [29]. Similarly, the first inequality in (36) becomes an equality iff $\boldsymbol{e}_g$ and $\boldsymbol{e}_l^{(i)}$ are independent.

From (36), we have

$$nH(\boldsymbol{e}_g) + \sum_i H\left(\boldsymbol{e}_l^{(i)}\right) \geq nH(\boldsymbol{d}_g) + \sum_i H\left(\boldsymbol{d}_l^{(i)}\right) \tag{37}$$

Let $A := \frac{k-1}{n-1}$. Then $0 < A < 1$. Multiplying (35) by $(1-A)$ and (37) by $A$, we have

$$(1 + (n-1)A)H(\boldsymbol{e}_g) + \sum_i H\left(\boldsymbol{e}_l^{(i)}\right) \geq (1 + (n-1)A)H(\boldsymbol{d}_g) + \sum_i H\left(\boldsymbol{d}_l^{(i)}\right), \tag{38}$$

which is equivalent to

$$\mathcal{L} = kH(\boldsymbol{e}_g) + \sum_i H\left(\boldsymbol{e}_l^{(i)}\right) \geq kH(\boldsymbol{d}_g) + \sum_i H\left(\boldsymbol{d}_l^{(i)}\right). \tag{39}$$

We note that the value on the RHS is realizable, bacause $\boldsymbol{e}_g = \boldsymbol{d}_g$ and $\boldsymbol{e}_l^{(i)} = \boldsymbol{d}_l^{(i)}$ realizes the value.

Now we examine the properties of the optimal solution: Since the equality in (39) holds, all inequalities in (35) and (36) hold as equalities. Hence, we have[3]

$$H(\boldsymbol{e}_g) + \sum_{i=1}^{n} H\left(\boldsymbol{e}_l^{(i)}\right) = H\left(\boldsymbol{e}_g, \boldsymbol{e}_l^{(1)}, ..., \boldsymbol{e}_l^{(n)}\right), \tag{40}$$

meaning $\left\{\boldsymbol{e}_1^{(1)}, ..., \boldsymbol{e}_n^{(n)}, \boldsymbol{e}_g\right\}$ being jointly independent (for an optimal solution). With such optimal solution, we also know that (By substracting (35) from (37))

$$(n-1)H(\boldsymbol{e}_g) = (n-1)H(\boldsymbol{d}_g). \tag{41}$$

Hence, $H(\boldsymbol{e}_g) = H(\boldsymbol{d}_g)$ and $H\left(\boldsymbol{e}_l^{(i)}\right) = H\left(\boldsymbol{d}_l^{(i)}\right)$ for all $i$.

*Q.E.D.*

#### 6.2.2.1 Discussion

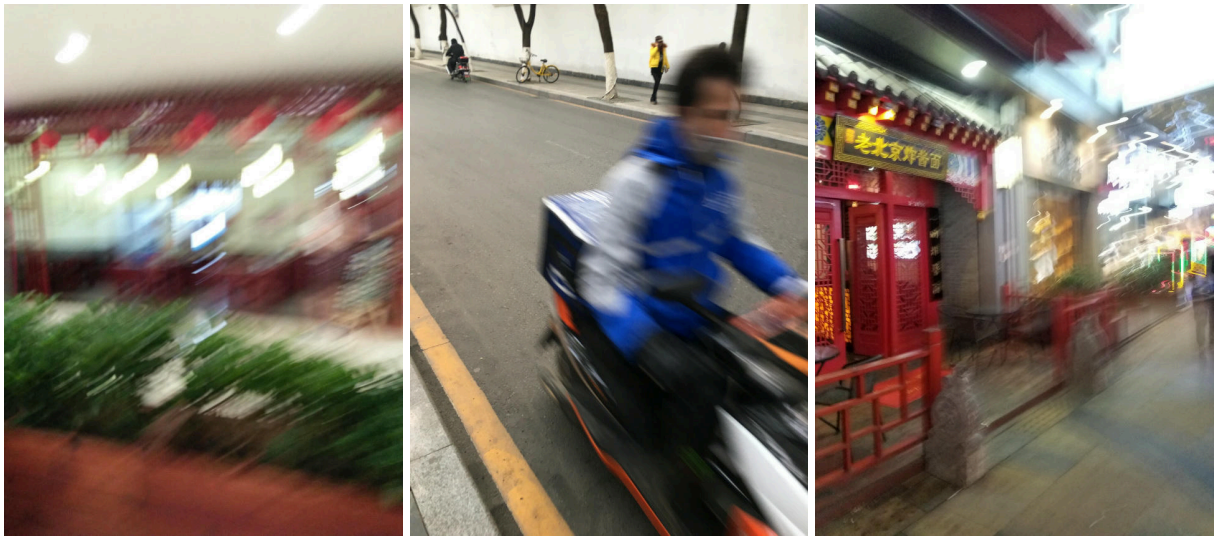
In the proof, we show that as long as the ratio between coefficients $k$ for $H(\boldsymbol{e}_g)$ and $\sum_i H(\boldsymbol{e}_l^{(i)}$ satisfies $1 < k < n$, the optimal solution satisfies the desirable properties. However, in practice, we calculate bits per picture for $\boldsymbol{e}_g$ and bits per pixel for $\boldsymbol{e}_l^{(i)}$, which means $k$ is indeed equivalent to $\frac{\lambda_g}{1/N\lambda_l}$, where $N$ is the number of pixels in the image. Hence, the condition $\lambda_g$ and $\lambda_l$ should satisfy is

$$\frac{1}{N} < \frac{\lambda_g}{\lambda_l} < \frac{n}{N}. \tag{42}$$

While the number of spatial location is ambiguous to define, we can always set $\lambda_g$ close to $\lambda_l$ to satisfy the condition in theory. In practice, $\lambda_g = \frac{1}{20}\lambda_l$ works well for patches sized from 64 to 384.

[3]From now on, $\boldsymbol{e}_g$ and $\boldsymbol{e}_l^{(i)}$ denote the optimal solution, instead of any feasible solution.

# 7 Examples for real-world inhomogeneous degradations



(a) Motion Blur with inhomogeneous strengths

(b) Motion Blur on limited area

(c) Motion Blur with different shape and strength

Figure 28: Examples of some inhomogeneous image distortions sampled from SPAQ [30]. In (a), the blur kernel for motion blur on the right is larger than the left. In (b), only the motorcyclist is feected by motion blur, the background is rather clear. In (c), the left has very limited level of blur but the right hand side has a severe blur with curved trajectory.

Several types of spatially inhomogeneous degradations are observed from authentic images (Figure 28 shows some examples from SPAQ [30]).

# Bibliography


[1] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes." [Online]. Available: https://arxiv.org/abs/1312.6114

[2] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image Quality Assessment: Unifying Structure and Texture Similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 5, pp. 2567–2581, 2022, doi: 10.1109/TPAMI.2020.3045810.

[3] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[4] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2018.

[5] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003, pp. 1398–1402. doi: 10.1109/ACSSC.2003.1292216.

[6] Z. Wang and E. Simoncelli, "Translation insensitive image similarity in complex wavelet domain," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, 2005, p. ii/573–ii/576 Vol. 2. doi: 10.1109/ICASSP.2005.1415469.

[7] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011, doi: 10.1109/TIP.2011.2109730.

[8] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 684–695, 2014, doi: 10.1109/TIP.2013.2293423.

[9] L. Zhang, Y. Shen, and H. Li, "VSI: A Visual Saliency-Induced Index for Perceptual Image Quality Assessment," *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4270–4281, 2014, doi: 10.1109/TIP.2014.2346028.

[10] F. Gao, Y. Wang, P. Li, M. Tan, J. Yu, and Y. Zhu, "DeepSim: Deep similarity for image quality assessment," *Neurocomputing*, vol. 257, pp. 104–114, 2017, doi: https://doi.org/10.1016/j.neucom.2017.01.054.

[11] H. R. Sheikh, A. C. Bovik, and G. de Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2117–2128, 2005, doi: 10.1109/TIP.2005.859389.

[12] H. Sheikh and A. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006, doi: 10.1109/TIP.2005.859378.

[13] M. Fritsche, S. Gu, and R. Timofte, "Frequency Separation for Real-World Super-Resolution," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 3599–3608. doi: 10.1109/ICCVW.2019.00445.

[14] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial Attentive Single-Image Deraining With a High Quality Real Rain Dataset," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12262–12271. doi: 10.1109/CVPR.2019.01255.

[15] S. Nah et al., "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," in *CVPR*, 2017. doi: 10.1109/CVPR.2017.35.

[16] Y. Deng et al., "StyTr2: Image Style Transfer With Transformers," in *CVPR*, Jun. 2022.

[17] Gnangarra, et al., "Commons:Quality images - Wikimedia Commons." [Online]. Available: https://commons.wikimedia.org/wiki/Commons:Quality_images

[18] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1510–1519. doi: 10.1109/ICCV.2017.167.

[19] S. Menon et al., "PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models," in *CVPR*, 2020. doi: 10.1109/CVPR42600.2020.00251.

[20] R. V. Marinescu, D. Moyer, and P. Golland, "Bayesian Image Reconstruction using Deep Generative Models," *CoRR*, 2020, [Online]. Available: https://arxiv.org/abs/2012.04567

[21] A. Conmy, S. Mukherjee, and C.-B. Schönlieb, "StyleGAN-induced data-driven regularization for inverse problems," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 3788–3792.

[22] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4431–4440. doi: 10.1109/ICCV.2019.00453.

[23] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to Edit the Embedded Images?," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8293–8302. doi: 10.1109/CVPR42600.2020.00832.

[24] Y. Poirier-Ginter and J.-F. Lalonde, "Robust Unsupervised StyleGAN Image Restoration," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 22292–22301. doi: 10.1109/CVPR52729.2023.02135.

[25] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[26] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and Improving the Image Quality of StyleGAN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[27] Hyungjin Chung et al., "Diffusion Posterior Sampling for General Noisy Inverse Problems," in *ICLR*, 2023. [Online]. Available: https://openreview.net/forum?id=OnD9zGAGT0k

[28] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet Classifiers Generalize to ImageNet?," in *International Conference on Machine Learning*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:67855879

[29] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM J. Res. Dev.*, vol. 4, no. 1, pp. 66–82, Jan. 1960, doi: 10.1147/rd.41.0066.

[30] Y. Fang, H. Zhu, Y. Zeng, K. Ma, and Z. Wang, "Perceptual Quality Assessment of Smartphone Photography," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3677–3686.