LTT-GAN: Looking Through Turbulence by Inverting GANs

Kangfu Mei     Vishal M. Patel
Department of Electrical and Computer Engineering
Johns Hopkins University
https://kfmei.page/LTT-GAN/

Abstract

In many applications of long-range imaging, we are faced with a scenario where a person appearing in the captured imagery is often degraded by atmospheric turbulence. However, restoring such degraded images for face verification is difficult since the degradation causes images to be geometrically distorted and blurry. To mitigate the turbulence effect, in this paper, we propose the first turbulence mitigation method that makes use of visual priors encapsulated by a well-trained GAN. Based on the visual priors, we propose to learn to preserve the identity of restored images on a spatial periodic contextual distance. Such a distance can keep the realism of restored images from the GAN while considering the identity difference at the network learning. In addition, hierarchical pseudo connections are proposed for facilitating the identity-preserving learning by introducing more appearance variance without identity changing. Extensive experiments show that our method significantly outperforms prior art in both the visual quality and face verification accuracy of restored results.

1. Introduction

Generative Adversarial Network (GAN) inversion is a new trending way in image restoration, which has been applied in applications, including but not limited to image super-resolution [7, 30], image colorization [44], blind face restoration [42, 46], and general tasks [14, 33]. Despite photo-realistic results can be restored from well-trained GANs, inverting degraded images is nontrivial. Among major inversion ways [7, 14, 25, 30, 33–35, 41, 42, 46, 51], the learning based method achieves the most compelling efficiency and does not need the awareness of the degradation. Although it is practical, unfaithful realism and unnatural details can be observed in its results, especially in the cases where large degradations exist e.g., atmospheric turbulence.

Learning-based methods, which we call Generative Embedding Network (GEN), usually consist of a learnable latent code predictor and an embedded well-trained GAN [5, 18]. They can transform the predicted latent code from the degraded image into a clear image with realistic details with the help of a well-trained GAN. Among them, the latent code predictor of GEN learns with an additional pixel-wise
loss between the restored image and the target image, e.g., $L_2$, together with the original adversarial loss, for preserving the identity and realism of restored results, respectively. Under such a scenario, the motivation of our method comes from three different aspects theoretically and empirically.

a) **Theoretically, the term of applied adversarial loss and pixel-wise loss is statistically different.** We assume that this difference leads to unfaithful realism of restored results. Thus, we take inspiration from the contextual approach [29] and replace the pixel-wise loss with it, for preserving identity while reducing the statistical difference. By comparing regions with similar semantics only, the contextual loss can bypass the alignment regularization of standard pixel-wise losses. Moreover, such a loss maintains statistical difference of images without explicitly estimating the density of high-dimensional semantic features, which can be approximated to the Kullback–Leibler (KL) divergence between two images [28]. Such a term is actually similar to the adversarial loss that can be approximated to the Jensen–Shannon (JS) divergence between two images [1]. We empirically find that simply replacing the $L_2$ loss with the contextual loss in GEN learning achieves 1.77% improvement on the facial embedding cosine distance.

b) **Empirically, the fine details (e.g., eyes) related to the identity are difficult to restore exactly only if the coarse appearance (e.g., color and pose) is exactly restored.** We argued that this results from the hierarchical generation process used by the embedded GAN, e.g., StyleGAN [18]. It combines interpolated multi-scale coarse features to generate fine details of restored images. However, error always exists in the coarse features, and thus the entangled fine detail cannot be easily learned to be preserved. We further argue that the identical features of each image is redundant, and thus delicately extracting sub-images from the original one does not change its identity. By comparing the statistical context of sub-images with the same identity but different appearances, our new contextual distance can tolerate the differences of coarse appearance and better consider the identity differences between the restored results and the ground truth. Such a modification on the original contextual distance leads to 2.40% improvement.

c) **The network should produce multiple results from a single input. Because multiple clean images can produce the same degraded image.** We further take inspiration from b) and empirically find that gradually changing coarse features (e.g., features in the shallow layer of StyleGAN) can result images with the same identity but different coarse appearance, which can be applied to implicitly correspond to multiple ground truth. Specifically, we connect the hierarchical layers of the embedded StyleGAN with multiple modulation features. By repeating hierarchical layers with different modulation features, the embedded StyleGAN can produce multiple identical images in a single forward pass. Comparing sub-images with more appearance variations on the modified contextual distance with the modified network as the loss further achieves 3.77% improvement.

Overall, our method is based on the recent state-of-the-art GEN method GFPGAN [42], but instead learns by a new contextual manner illustrated in Figure 4, and with a different hierarchical generation manner illustrated in Figure 3. The effectiveness of ours is shown in the turbulence face image mitigation problem, illustrated in Figure 2. Note that this is the first method that can produce sharp-looking face images from turbulence degraded imagery in a resolution of $512 \times 512$. Our method substantially improves upon the previous best method by 4.51 in visual quality metric FID and recognition accuracy Top1 of 11.24%, on synthetic and real-world turbulence degraded face images, respectively.

2. Related Works

**Atmospheric Turbulence Simulation and Mitigation** based on deep neural networks has gained traction in recent years due to its applications in long-range surveillance and biometrics. A detailed discussion regarding turbulence simulation and its effects on images can be found in [13, 36].

For training a deep network for restoring images degraded by turbulence, Chak et al. [6] and Lau et al. [22] proposed random distortions and blur-based on handcrafted rules to synthesize pair-wise turbulence degraded images. This data augmentation/synthesis method has been successfully used in many restoration works [21], [49], [32]. More recent works from Chimitt et al. [9] and its improved version by Mao et al. [27] follow the split-step propagation method to simulate the effect of turbulence on images [4, 15]. It can model the turbulence caused by wavefront distortion, and it statistically fits the theoretical predictions of turbulence. In this paper, we use the simulation method from Mao et al. [27] to synthesize turbulence degraded images.

**GAN Inversion for Restoration** with well-trained GANs can provide photo-realistic priors, and thus it is a new promising avenue for image restoration. According to the recent survey of GAN inversion [45], there are two major ways for embedding a well-trained GAN into the restoration framework. Typical methods [10, 14, 26, 30] belong to the first approach in which the optimal latent code of GANs are retrieved with some iterative optimization algorithms. Different modifications on the latent space definition or optimization procedure are proposed. Methods such as [2, 7, 46, 51] fall in the second approach, in which an additional encoder is employed to predict the optimal point, which usually requires pair-wise data to train the additional encoder with pixel-wise loss. Our method falls in the second category, but our learning objective differs from these typical methods and can better consider the identity of restored results for restoration learning.
3. Proposed Method

3.1. Preliminaries

Following [21, 32, 48, 49], the degradation due to atmospheric turbulence can be expressed as follows

\[ \tilde{I}_k = D_k(H_k(I)) + n_k, \]  

where \( \tilde{I}_k \) is the single turbulence image, \( D_k(*) \) and \( H_k(*) \) are the deformation operator and turbulence-induced point spread function, respectively, \( I \) is the clean image, and \( n_k \) is the noise. Both degradations in Equation (1) can be combined into a single degradation \( T(\cdot) \) and as a result, we obtain the following more general observation model

\[ \tilde{I} = T(I). \]  

Turbulence mitigation entails finding an inverse function that can map arbitrary turbulence degraded image \( \tilde{I} \) into image \( G(\tilde{I}) \) that is close to the clear image \( I \), as shown in Figure 2. However, this is an ill-posed problem due to the large variations in degradation \( T(\cdot) \). In other words, each turbulence degraded image \( \tilde{I} \) can correspond to multiple clear images. Here, we define these possible “ground truths” as ground truth and restored results, respectively. According to the derivation [47], minimizing Equation (3) is equivalent to minimize the Jensen-Shannon divergence between the conditional empirical distribution \( C_{\tilde{I}} \) and the averaged conditional distribution \( C_I \) of \( I/\tilde{I} \) as

\[ \mathcal{L}_{MSE}(G) = \frac{1}{n} \sum_{i=1}^{n} ||G(\tilde{I}) - I_i||^2, I_i \in I. \]  

The averaged loss term will perform differently from the original one depending on the variation level of degradations. Thus, the difference is usually ignored in tasks with limited variance, e.g., super-resolution.

In contrast, GANs are able to generate realistic looking images due to the use of adversarial loss, e.g., the logistic loss function with a discriminator \( D(\cdot) \). When learning a GEN, which fine-tunes the network with a well-trained GAN, the applied well-trained discriminator is optimal. Thus, the adversarial learning of the network \( G(\cdot) \) is equivalent to minimize the Jensen-Shannon divergence between the two image distributions according to [1] as

\[ \mathcal{L}(D, G) = 2JS(P_T || P_g) - 2 \log 2, \]

\[ JS(P_T || P_g) = \mathbb{E}_{x \sim P_T} \left[ \log \frac{P_T(x)}{P_g(x)} \right] + \mathbb{E}_{x \sim P_g} \left[ \log \frac{P_g(x)}{P_T(x)} \right], \]  

where \( P_T \) and \( P_g \) are the probability distributions of the ground truth and restored results, respectively. According to our motivation a), the difference between Equation (3) and Equation (5) shows inconsistency in the gradient orientation of the network learning. Therefore, it is difficult to achieve a good trade-off between realism and identity preservation in the conventional way of learning GENs, especially in the restoration tasks with large variations in the degradation (i.e. turbulence in this paper).

**Contextual Distance** From the derivation in [28] corresponding to the contextual loss [29], which was originally proposed for misaligned data, the contextual distance in perceptual features \( X \) and \( Y \) of images can be approximated as

\[ KL(P_X || P_Y) = \int P_X \log \frac{P_X}{P_Y}, \]  

where \( P_X \) and \( P_Y \) are the density of points estimated on the perceptual features, with multivariate kernel density estimation. Due to the similarity in terms of the adversarial and contextual losses, we empirically find that replacing \( L_{2} \) loss with the contextual loss in training a GEN can significantly accelerate the convergence. However, there has been no demonstration that such a contextual loss is capable of preserving identity as well as realism of restored results.

### 3.2. Spatial Periodic Contextual Distance

In our method we treat each image as a collection of multiple sub-images extracted spatial periodically, and we then consider the contextual distance between the two sub-image collections as the identity preserving loss. Figure 4 illustrates this procedure with a toy example. As illustrated, \( H \times W \) number of sub-images \( \{x_1, x_2, \ldots, x_{HW}\} \) in the dimension of \( r \times r \) contain different parts of the original image \( X \) in the dimension of \( rH \times rW \). Though these sub-images are significantly different in appearance (no overlapping pixels), they share the same identity property, i.e., each sub-image can denote the same waifu character as the original image denotes. We empirically find that the contextual distance built upon these sub-images helps GENs to learn better preserve the identity information.

Different from the internal patch [52] statistics with sliding windows or contextual feature points [28] with pre-trained convolution layers, each sub-image in our distance measurement is extracted with a defined operation \( PK \). For an image of size \( rH \times rW \times C, PK \) unshuffles (i.e., a twice-inverse transformation of pixel shuffling [38], see the supplement for code reference) the image into \( H \times W \) number of images in a dimension of \( r \times r \times C \) with rate \( r \), which
can be mathematically denoted as
\[
PK(I)_{x \times y, r_1, r_2, c} = I_{[x+r_1], [y+r_2], c}. \tag{7}
\]
Each sub-image is then flattened into a vector, which can be seen as a high-dimensional identity representations of the original image. We can then estimate the contextual distance between the two representations in a global context.

For contextual distance, we follow Mechrez et al. [29] and formally define it on two collections of sub-images
\[
PK(X) = \{x_i\} \text{ and } PK(Y) = \{y_j\}, \text{ e.g., restored results and ground truths, as follows}
\]
\[
SPCX(PK(X), PK(Y)) = -\frac{1}{N} \sum_i \log \sum_j A_{ij}, \tag{8}
\]
where \(A_{ij}\) can be seen as the second log term of a multi-variate kernel density estimation kernel. In practice, \(A_{ij}\) is implemented to be close to a delta function with normalized cosine distance \(d(x_i, y_j)\) and bandwidth \(h\) as
\[
A_{ij} = \frac{\exp \left(1 - \tilde{d}_{ij} / h\right)}{\sum_i \exp \left(1 - \tilde{d}_{ij} / h\right)} = \begin{cases} 1 & \text{if } \tilde{d}_{ij} \ll \tilde{d}_l \forall l \neq j \\ 0 & \text{otherwise} \end{cases}
\tag{9}
\]
Minimizing such a distance approximates minimizing the divergence of \(KL(\mathbb{P}_X \| \mathbb{P}_Y)\) according to Mechrez et al. [28], and it is similar to the term of adversarial loss defines in Equation (5). In practical implementation, for the face dataset FFHQ in the dimension of \(512 \times 512\), we empirically find that \(r = 32\) leads to the best learning performance.

### 3.3. Hierarchical Pseudo Connections

In Section 3.2 we discussed the spatial periodic contextual distance between two images for considering their identity difference. However, the existing multiple ground truths issue mentioned in Equation (4) still affects the identity preserving capability of the network, although it can learn with a better objective. Therefore, in this section, we leveraged the hierarchical generation property of GANs, e.g., StyleGAN [18], which allows high-resolution photo-realistic images to be generated by combining multi-scale features. By delicately organizing the features in different scales, we enable well-trained GANs to produce multiple results in a single forward pass. Thus, with more samples in the same identity but different appearances, our proposed spatial periodic contextual distance can better estimate the distributions and their difference, without network architecture modification on the original embedded GANs.

As Figure 3 illustrated, based on the network architecture of GFPGAN [42], the major difference of our method comes from the proposed Hierarchical Pseudo Connections (HPC), which connects the latent code predictor (\(F_{\text{Encoder}}\) and \(F_{\text{Decoder}}\)) with multiple style blocks for feature modulation. Here feature modulation [18] is applied as the expanded latent code space for GAN inversion. Our \(g\)-th HPC transforms feature \(M\) from \(F_{\text{Decoder}}\) into affine transformation parameters \(\{\alpha, \beta\}\), then each group of affine transformation parameters modulates the feature of StyleGAN as
\[
F_{\text{StyleGAN}} = \alpha_i \odot F_{\text{StyleGAN}} + \beta_i, i \in 2^g. \tag{10}
\]
Here we expand each style layer in the original embedded StyleGAN into a group, i.e., grouping style layers with the same parameters but different modulation features. In Figure 3, the actual style layer is illustrated with solid lines, and the pseudo style layer is illustrated in dotted lines. The pseudo style layer is defined as the real style layer pro-
cesses feature with different modulation features instead of the original modulation features. Benefited by the hierarchical generation manner, HPC allows the well-trained GAN to generate multiple possible results with similar coarse appearance but different fine details. Therefore, the applied HPC with \(2^g\) number of style layers in the final group, comes from groups consisting of \(\{2^3, 2^2, \ldots, 2^{g-1}\}\) number of style layers respectively, which transforms \(G(\cdot)\) as
\[
\hat{I} = \{I_1, I_2, \ldots, I_{2^g}\} = G(\hat{I}). \tag{11}
\]
Assume that the ill-posed nature of problem produces \(n\) possible solutions in the image set \(I\). Though \(2^g << n\), we can significantly reduces their difference with the transformed Equation \((\cdot)\) as
\[
\mathcal{L}(G) = SPCX(\mathcal{PK}(I), \mathcal{PK}(\hat{I}), r). \tag{12}
\]
We then stack all of the generated results for training the network and take the averaged image as the restoration result. A generated example from our method with \(g = 3\) is shown in Figure 5, where 8 similar images are generated.

Figure 5. Turbulence degraded images with the same identity but different coarse appearance are restored with HPC \(g = 3\). Here the lighting in the hair part is changing in each image.

3.4. Model Objective

The final learning objective for training our method combines both the proposed spatial periodic contextual loss, adversarial loss, perceptual loss, and identity preserving loss.

Reconstruction Related. For network \(G(\cdot)\) with \(2^g\) number of final hierarchical pseudo connections we have
\[
\mathcal{L}_{rec}(G) = \frac{1}{2^g} \sum_{i}^{2^g} SPCX(\mathcal{PK}(I), \mathcal{PK}(\hat{I}_i))
- \lambda_{adv} E_{G(\hat{I}_i)} \text{softplus}(D(G(\hat{I}_i)))
+ \lambda_{per} ||\phi(I) - \phi(G(\hat{I}_i))||
+ \lambda_{id} ||\eta(I) - \eta(G(\hat{I}_i))||, \tag{13}
\]
where \(\phi(*)\) is the pre-trained VGG-19 network [40] that is applied with its \{conv1, \ldots, conv5\} layers before activation [43], \(\eta(*)\) is the pre-trained ArcFace network [11] without the final logistic layer. \(\lambda_{adv}, \lambda_{per}, \text{and } \lambda_{id}\) are the loss weights of adversarial loss, perceptual loss, and identity preserving loss, respectively. In our implementation, we empirically set them as \(\{1 : 0.1 : 10\}\).

Adversarial Related. The discriminator \(D(*)\) is trained similar to StyleGAN2 [19] except that multiple results are applied and the adversarial losses are averaged.

4. Experiments

In this section, we present the experimental details for evaluating our method and its settings, as well as the comparison results with the state-of-the-art methods.

4.1. Testing and Training Settings

Synthesized Testing Benchmark. For the reference-based evaluation, we synthesize turbulence-clean image pairs using TurbulenceSim_P2S [27] which is the current state-of-the-art turbulence simulation works. The synthesis is conducted on the selected first 100 images of CelebAHQ [17], named CelebAHQ100, and the parameters of TurbulenceSim_P2S are carefully selected to match the real-world turbulence images. Specifically, we set \(D, r_0\), and \(corr\) as \(\{5, 1.25, -0.01\}\) for the CelebAHQ100 simulation. The synthesized image pairs are provided in the supplementary document in 8-bit sRGB format with a resolution of \(512 \times 512\).

Real-world Testing Benchmark. For evaluating the performance of different methods on real-world turbulence degraded images without pixel-wise corresponding ground truths, we use face recognition accuracy based on indoor reference clear images. The authors of [48] provided us high-quality raw real-world turbulence degraded faces which are taken at 300 meters from the camera in a hot day. In addition to those images, we received the corresponding indoor face images without turbulence for reference. We crop and wrap faces with the pre-trained RetinaFace [12] network. The final dataset contains images from 89 separate individuals each having 3 turbulence degraded images in different poses. We call this data TubFace89 and show its sampled images in Figure 6.

Figure 6. Visualization of applied real-world turbulence images. These images are captured by cameras put in left, center, right positions, and the last image is captured indoor without turbulence.

Training Dataset and Data Augmentation We applied FFHQ [18] with 70,000 high-quality images in a resolution of \(512 \times 512\) as the training dataset. To find the best way to boost the turbulence mitigation capability of networks, we conduct experiments with three existing data augmentation methods, i.e., Blind Face Restoration (BFR) [23,24,42,46],
Turbulence Image  TDRN arXiv20  ATFaceGAN arXiv20  PSFRGAN CVPR21  GFPGAN CVPR21  GFPGAN* (ElasticAug)  LTT-GAN Ours  Ground Truth Image

Figure 7. Visualization results of restored synthesized turbulence face images. Compared with results contain unnature artifacts, our results achieve the best visual quality and identity similarity of the ground truth. See the supplement for more such comparisons.

TurbulenceSim, BFR, and our new method, called ElasticAug for simulation turbulence effects for training.

Among them, TurbulenceSim P2S [27], and our new method, ElasticAug for simulation turbulence models for turbulence simulation. Since the real-world turbulence images usually suffer strong blur degradation, we empirically find that the degradation procedure used in BFR can also produce similar degraded results to real-world turbulence degradation. Moreover, in this paper, we introduce a new simulation method, called ElasticAug, which is based on BFR augmentation, and it combines the blur augmentation with Elastic transformation [39], which randomly moves pixels using local displacement fields.

Here we directly apply GFPGAN [42] as the baseline and train the network on three data augmentation methods. Figure 8 shows their restored results. One can find that the network trained on ElasticAug achieves the best visual quality of restored results. Although BFR produces a similar degradation procedure of turbulence, it cannot estimate the spatial deformation, and hence its result contains strong ring artifacts. Although TurbulenceSim_P2S produces a comparable clear result, the network trained on its simulated images generates the result with strange contour artifacts, and thus the data augmentation is only used for evaluation. Therefore, our proposed ElasticAug is availed as the default data augmentation method in the following experiments.

Implementation
Our implementation follows the settings of conventional generative embedding networks [42], i.e., the learning rate was set to $2 \times 10^{-3}$ and decayed at 600k and 700k iterations by a rate of 0.5, with a total of 800k iterations. The whole training is conducted in a mini-batch size of 12 using 4 NVIDIA A100 GPUs. For the $F_{Encoder}$ and $F_{Decoder}$, we initialized their parameters with Xavier initialization. For the GAN module, we apply StyleGAN2 [19] with its well-trained parameters on FFHQ generation, and its parameters are frozen at the network training. For the discriminator module, we apply the discriminator of well-trained StyleGAN2 on FFHQ generation, and its parameters are optimized with the same learning rate as the GAN module. Here we empirically set $r = 32$ and $g = 3$.

4.2. Comparisons with SOTA Methods
We conduct comparisons with several state-of-the-art turbulence mitigation methods i.e., TDRN [48] and ATFaceGAN [21]. For fair comparisons, we finetune them on FFHQ with their officially released codes. We also compare with the recent SOTA blind face restoration methods, i.e., PSFRGAN [8] and GFPGAN [42] trained on FFHQ.

Synthesized Turbulence. In Figure 7, we compare visual restored results on the synthesized turbulence degraded images. One can find that our method best restores the facial details with photorealism. Among the compared methods, though the results from GFPGAN trained on ElasticAug seem to contain comparable details, their details are less similar to the ground truth. In comparison, our results best preserve the identity related details in the restored images.
Figure 9. Visualization results of compared methods on real-world turbulence images. The proposed method achieves the best visual quality and maintains the identity in the restored images with fewer unfaithful details. See the supplement for more such comparisons.

| Method          | LPIPS | FID | NIQE | Deg. (%) | PSNR | SSIM | Imgs/Sec | Params (M) |
|-----------------|-------|-----|------|----------|------|------|----------|------------|
| BFR             |       |     |      |          |      |      |          |            |
| PSFRGAN [8] [CVPR21] | ✔ | - | 0.4070 | 122.39 | 4.051 | 22.30 | 19.96 | 0.5451 | 3.70 | 184.2 |
| GFPGAN [42] [CVPR21] | ✔ | - | 0.3800 | 113.42 | 5.515 | 26.55 | 20.31 | 0.5797 | 20.14 | 615.4 |
| TDRN [48] [arXiv20] | - | ✔ | 0.5869 | 190.17 | 13.55 | 18.54 | 21.63 | 0.6611 | 3.41 | 8      |
| ATFaceGAN [21] [TBIOM21] | - | ✔ | 0.5868 | 181.10 | 12.71 | 15.55 | 21.77 | 0.6633 | 7.32 | 68.70 |
| GFPGAN* | - | ✔ | 0.3288 | 90.23 | 4.435 | 40.53 | 20.49 | 0.5790 | 20.14 | 615.4 |
| LTT-GAN (Ours) | - | ✔ | 0.2906 | 85.72 | 4.285 | 49.88 | 20.96 | 0.6042 | 14.12 | 741.4 |

Table 1. Performance comparisons against the state-of-the-art methods and ours on the synthesized turbulence face images. achieves the best performance in visual quality, perceptual metrics, and identity-preserving metrics, i.e., FID, LPIPS, and Deg.

In Table 1, we present quantitative performance comparisons in the widely used visual quality metrics, i.e., LPIPS [50], FID [16], NIQE [31], and pixel-wise metrics, i.e., PSNR and SSIM. Since the turbulence mitigation task is highly correlated to face recognition, we further employ the identity metric, i.e., Deg, which is the cosine distance of facial features on the pretrained ArcFace-Resnet18 [11]. From the results, we can notice that our method achieves the best performance in the LPIPS, FID, and Deg metrics, and the comparable performance in NIQE. Regarding facial identity preserving performance, our method outperforms the second best method in 9.53%. Note that as recent restoration works [3, 42] suggested, the pixel-wise metrics (e.g., PSNR and SSIM) are not strongly correlated to visual quality of images, and thus our method is not good at them does not mean our results are worse than others.

In Figure 9, we further present the turbulence mitigation results of compared methods in real-world conditions, which are more severely degraded than the synthesized ones. However, our method can still achieve its superiority in visual quality comparisons, with fewer unnatural artifacts compared with both the turbulence mitigation methods and BFR methods. We argue that such superiority comes from the learning objective difference, in which our objective bypasses the uncertainty of datas, and thus it can better preserve the identity features of restored images.

In Table 2 and Table 3, we show the pre-trained ArcFace-Resnet18 face recognition accuracy of the restored results, in the center pose, and center, left, right poses, respectively. From the comparisons, one can easily find that our method significantly outperforms the other methods in recognition accuracy Top1 and Top3, in both the center pose and all poses. Compared with the second best methods, ours outperform it over 7.86% and 11.24%. Though GFPGAN partly achieved better Deg than ours, we find that it tends to produce more unfaithful details that effects Deg, but unfaithful details is not identical and lead to a worse Top1.

4.3. Ablations and Discussions

In this section, we conduct experiments on different settings of our method to discuss their effectiveness on the turbulence mitigation capability, includes Baseline (L2 for identity) vs. Baseline with Brute-Force L2 (larger weights of L2) vs. Baseline with Brute-Force ID (larger weights of ID) vs. Baseline with Contextual loss [29] (CX) vs. LTT-GAN with different rates of SPCX vs. LTT-GAN with SPCX(r=32)
Figure 10. Visualization results of our method on the synthesized CelebAHQ100, as we vary the usage of SPCX and HPC. Compared with the baseline, or the baseline combines with the contextual loss, our method significantly reduce the unnatural artifacts with better details.

Table 2. A verification accuracy comparison against the state-of-the-art methods and ours on the real-world turbulence degraded face images collected at 300 meters.

| Method                | Top1 ↑ | Top3 ↑ | Top5 ↑ | Deg. (%) ↑ |
|-----------------------|--------|--------|--------|------------|
| Turbulence Images     | 14.61  | 26.97  | 32.58  | 13.59      |
| TDRN [48] [arXiv20]  | 13.37  | 28.99  | 30.11  | 10.41      |
| ATFaceGAN [21] [TBIOM21] | 23.60  | 38.20  | 51.69  | 17.26      |
| PSFRGAN [8] [CVPR21]  | 39.33  | 56.18  | 61.80  | 27.71      |
| GFPGAN [42] [CVPR21]  | 49.44  | 68.54  | 79.78  | 32.08      |
| GFPGAN*               | 34.83  | 52.81  | 64.04  | 35.11      |
| LTT-GAN (Ours)        | 57.30  | 71.91  | 82.02  | 35.27      |

Table 3. A verification accuracy comparison against the state-of-the-art methods and ours on the real-world turbulence degraded face images collected at 300 meters in center, left, and right poses.

| Method                | Top1 ↑ | Top3 ↑ | Top5 ↑ | Deg. (%) ↑ |
|-----------------------|--------|--------|--------|------------|
| Turbulence Images     | 14.61  | 44.94  | 50.56  | 11.11      |
| TDRN [48] [arXiv20]  | 19.10  | 42.11  | 53.68  | 12.46      |
| ATFaceGAN [21] [TBIOM21] | 23.60  | 64.04  | 69.66  | 13.48      |
| PSFRGAN [8] [CVPR21]  | 41.57  | 73.03  | 84.27  | 20.30      |
| GFPGAN [42] [CVPR21]  | 48.31  | 73.03  | 84.27  | 20.30      |
| GFPGAN*               | 37.08  | 78.65  | 86.52  | 27.46      |
| LTT-GAN (Ours)        | 59.55  | 87.64  | 93.26  | 26.57      |

Table 4. A performance comparison against several different settings of our method on the synthesized CelebAHQ100.

| Method                | LPIPS ↓ | FID ↓ | Deg. (%) ↑ |
|-----------------------|---------|-------|------------|
| no./StyleG no./StyleD no./ID | 0.3765  | 134.15| 23.64      |
| no./StyleG no./StyleD no./StyleG | 0.3733  | 114.56| 33.02      |
| Baseline              | 0.3469  | 85.36 | 36.80      |
| w./Brute-Force L_2    | 0.3357  | 86.62 | 36.70      |
| w./Brute-Force ID     | 0.3510  | 95.72 | 37.77      |
| w./CX                 | 0.3358  | 85.08 | 38.57      |
| w./SPCX(r=8)          | 0.3360  | 87.94 | 38.10      |
| w./SPCX(r=16)         | 0.3328  | 82.75 | 38.67      |
| w./SPCX(r=32)         | 0.3370  | 85.90 | 39.20      |
| w./SPCX(r=64)         | 0.3403  | 91.54 | 38.90      |
| w./L_2 w./HPC(g=3)    | 0.3429  | 87.43 | 38.75      |
| w./SPCX(r=32) w./HPC(g=3) | 0.3273  | 92.80 | 40.03      |
| w./SPCX(r=32) w./HPC(g=4) | 0.3324  | 94.22 | 40.74      |
| w./SPCX(r=32) w./HPC(g=5) | 0.3395  | 102.64| 40.57      |

5. Conclusion

We have presented LTT-GAN, a new generative embedding network that can achieve the best trade-off between the identity and realism of restored turbulence images. Compared to previous approaches, LTT-GAN achieves the goal in a similar way of the adversarial loss, but instead, is implemented with a new contextual distance. The new objective can better consider the identity difference than previously applied losses. To further boost its performance, we modify the embedded well-trained GAN to allow multiple results to be acquired from a single input, which can introduce more appearance variance without identity changing in a single forward pass. On the difficult turbulence mitigation problem, LTT-GAN is the first GAN inversion method, and it provided significant performance improvements over the existing state-of-the-art methods in both the synthesized and real-world turbulence degraded images.
References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017. 2, 3

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019. 2

[3] Yochai Blau, Roey Mechez, Radu Timofte, Tomer Michaeli, and Lihl Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 7

[4] Jeremy P. Bos and Michael C. Roggemann. Technique for simulating anisoplanatic image formation over long horizontal paths. *Optical Engineering*, 51(10):101704, 2012. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 1

[6] Wai Ho Chak, Chun Pong Lau, and Lok Ming Lui. Subsampled turbulence removal network. *arXiv preprint arXiv:1807.04418*, 2018. 2

[7] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14245–14254, 2021. 1, 2

[8] Chaofeng Chen, Xiaoming Li, Lingbo Yang, Xianhui Lin, Lei Zhang, and Kwan-Yee K. Wong. Progressive semantic-aware style transformation for blind face restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11896–11905, 2021. 6, 7, 8

[9] Nicholas Chimit and Stanley H. Chan. Simulating anisoplanatic turbulence by sampling correlated Zernike coefficients. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, 2020. 2

[10] Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE Transactions on Neural Networks and Learning Systems*, 30(7):1967–1974, 2018. 2

[11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 5, 7

[12] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019. 5

[13] Joseph W. Goodman. *Statistical optics*. John Wiley & Sons, 2015. 2

[14] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3012–3021, 2020. 1, 2

[15] Russell C. Hardie, Jonathan D. Power, Daniel A. LeMaster, Douglas R. Droge, Szymon Gladysz, and Santasri Bose-Pillai. Simulation of anisoplanatic imaging through optical turbulence using numerical wave propagation with new validation analysis. *Optical Engineering*, 56(7):071502, 2017. 2

[16] 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. 7

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 5

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 2, 4, 5

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 5, 6

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 11

[21] Chun Pong Lau, Carlos D. Castillo, and Rama Chellappa. ATFaceGAN: Single Face Semantic Aware Image Restoration and Recognition From Atmospheric Turbism. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2):240–251, 2021. 1, 2, 3, 6, 7, 8

[22] Chun Pong Lau and Lok Ming Lui. Subsampled turbulence removal network. *Mathematics, Computation and Geometry of Data*, 1(1):1–33, 2021. 2

[23] Xiaoming Li, Chaofeng Chen, Shangchen Zhou, Xianhui Lin, Wangmeng Zuo, and Lei Zhang. Blind face restoration via deep multi-scale component dictionaries. In *Proceedings of the European Conference on Computer Vision*, pages 399–415, 2020. 5

[24] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. In *Proceedings of the European Conference on Computer Vision*, pages 272–289, 2018. 5

[25] Ji Lin, Richard Zhang, Frieder Ganz, Song Han, and Jun-Yan Zhu. Anycost gans for interactive image synthesis and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14986–14996, 2021. 1

[26] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 9651–9660, 2019. 5

[27] Zhiyuan Mao, Nicholas Chimit, and Stanley H Chan. Accelerating atmospheric turbulence simulation via learned phase-to-space transform. In *Proceedings of the IEEE/CVF In-
ternational Conference on Computer Vision, pages 14759–14768, 2021. 2, 5, 6

[28] Roey Mechrez, Itamar Talmi, Firas Shama, and Lihi Zelnik-Manor. Maintaining natural image statistics with the contextual loss. In Asian Conference on Computer Vision, pages 427–443, 2018. 2, 3, 4

[29] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision, pages 768–783, 2018. 2, 3, 4, 7

[30] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2437–2445, 2020. 1, 2

[31] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a “completely blind” image quality analyzer. IEEE Signal processing letters, 20(3):209–212, 2012. 7

[32] Nithin Gopalakrishnan Nair and Vishal M. Patel. Confidence Guided Network For Atmospheric Turbulence Mitigation. In 2021 IEEE International Conference on Image Processing, pages 1359–1363, 2021. 2, 3

[33] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. 1

[34] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14104–14113, 2020. 1

[35] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2287–2296, 2021. 1, 2, 3

[36] Michael C. Roggemann and Byron M. Welsh. Imaging through turbulence. CRC press, 2018. 2

[37] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1532–1540, 2021. 11

[38] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1874–1883, 2016. 3

[39] Patrice Y. Simard, David Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In ICDAR, volume 3, 2003. 6

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 3

[41] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. ACM Transactions on Graphics, 40(4):1–14, 2021. 1

[42] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards Real-World Blind Face Restoration with Generative Facial Prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9168–9178, 2021. 1, 2, 4, 5, 6, 7, 8

[43] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision Workshops, pages 0–0, 2018. 5

[44] Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. Towards Vivid and Diverse Image Colorization with Generative Color Prior. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14377–14386, 2021. 1

[45] Wei-hao Xia, Yulin Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. arXiv preprint arXiv:2101.05278, 2021. 2

[46] Tao Yang, Peiran Ren, Xuan-song Xie, and Lei Zhang. GAN Prior Embedded Network for Blind Face Restoration in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 672–681, 2021. 1, 2, 5

[47] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. IEEE Transactions on Multimedia, 21(12):3106–3121, 2019. 3

[48] Rajeev Yasarla and Vishal M. Patel. Learning to Restore A Single Face Image Degraded by Atmospheric Turbulence using CNNs. arXiv preprint arXiv:2007.08404, 2020. 3, 5, 6, 7, 8, 11

[49] Rajeev Yasarla and Vishal M. Patel. Learning to Restore Images Degraded by Atmospheric Turbulence Using Uncertainty. In IEEE International Conference on Image Processing, pages 1694–1698, 2021. 3

[50] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 586–595, 2018. 7

[51] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In Proceedings of the European Conference on Computer Vision, pages 597–613, 2016. 1, 2

[52] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In International Conference on Computer Vision, pages 479–486, 2011. 3
A. Limitations

**Training Data Bias.** The applied training dataset follows the most common settings of the face generation tasks, *i.e.*, FFHQ. However, we find that the dataset has data bias towards dark-skinned faces. The network learns with the bias tends to light the face images when the degradation is severe, *e.g.*, real-world turbulence degradation. In the following figure, we show our turbulence mitigation results in both the synthesized and real-world turbulence images, respectively. From the results, we suggest that the way of solving the issue is including more diverse images at training, or applying a controllable synthetic dataset.

![Image](image1.png)

**Spatial Inductive Bias.** As recent works suggested, most state-of-the-art generative networks, *e.g.*, StyleGAN, have strong spatial inductive bias. Specifically, the fine details appear to be fixed in pixel coordinates. We empirically find that such a spatial inductive bias also affects the restored results of GAN inversion. In the following figure, we show our turbulence mitigation results as well as the baseline, *i.e.*, GFPGAN in three different poses. Thanks to the proposed spatial periodic contextual distance, which can bypass the spatial pixel difference, our results are significantly more robust than GFPGAN with less unnatural artifacts. Though these unnatural artifacts led by the bias have limited effects on face recognition accuracy. However, our results in left and right poses are less realistic than the results in center poses.

![Image](image2.png)

B. Uncertainty Visualization

Identifying the uncertainty of restored results can help both the manual inspection and automated methods. Conventional restoration methods tend to produce artifacts in uncertain areas, which can help to identify. However, in GAN inversion, the uncertainty area of their results remains sharp, and hence it is hard to identify, but the imperceptible uncertainty is extremely dangerous in practical applications. With the help of HPC and SPCX, our method now can show the uncertainty map of its prediction using the variance of its pseudo results, as shown in Figure 13. By simply measuring the variance of multiple pseudo result images, we can treat the uncertainty as the variance of each pixel, in a single forward pass. It does not require additional network architecture modification, *e.g.*, Monte Carlo dropouts [20] used in TDRN [48], nor multiple forward passes of Style-Mixing used in recent GAN inversion methods [35, 37].

![Image](image3.png)

C. Code

Algorithm 1 provides reference NumPy code for \( \mathcal{P} \mathcal{K} \) operation and spatial periodic contextual distance measurement. A comparable code for vanilla contextual distance is provided for reference too. A complete code release will be made available upon publication.

D. Additional Results

Figure 14 presents the additional results of compared methods and ours on the synthetic CelebAHQ100 dataset. Figure 15 presents the additional results of compared method and ours on the real-world turbulence face TubFace89 dataset.
Figure 14. Additional visualizations of restored synthetic turbulence face images. Compared with results contain unnature artifacts, our results achieve the best visual quality and identity similarity of the ground truth. (200% Zoom is recommended to see their difference)
Figure 15. Additional visualizations of restored real-world turbulence face images. Compared with results contain unnature artifacts, our results achieve the best visual quality and identity similarity of the reference images. (200% Zoom is recommended to see their difference)
import numpy as np
from VGG import vgg19_feature_extractor

def pk_operation(img, r):
    H, W, C = img.shape
    img = img.reshape(H//r, r, W//r, r, C)
    img = img.transpose(0, 2, 1, 3, 4)
    img = img.reshape(H//r * W//r, r * r * C)
    return img

# vanilla contextual distance
def CX(X, Y, band_width=0.2):
    perceptual_x = vgg19_feature_extractor(X)
    perceptual_y = vgg19_feature_extractor(Y)

    C = perceptual_x.size(0)
    perceptual_x = perceptual_x.reshape(C, -1)
    perceptual_y = perceptual_y.reshape(C, -1)
    cos_dis = np.matmul(perceptual_x.transpose(0, 1), perceptual_y)
    w = np.exp((1 - cos_dis) / band_width)
    cx = w / np.sum(w, axis=2, keepdims=True)
    cx = np.mean(np.max(cx, axis=1)[0], dim=1)
    cx = np.mean(-np.log(cx + 1e-5))
    return cx

# spatial periodic contextual distance
def SPCX(X, Y, band_width=0.2):
    sub_x = pk_operation(X)
    sub_y = pk_operation(Y)

    # ===== different from CX =====
    cos_dis = np.matmul(sub_x, sub_y.transpose(0, 1))
    w = np.exp((1 - cos_dis) / band_width)
    cx = w / np.sum(w, axis=2, keepdims=True)
    cx = np.mean(np.max(cx, axis=1)[0], dim=1)
    cx = np.mean(-np.log(cx + 1e-5))
    return cx

Algorithm 1. Reference numpy code for PK operation and spatial periodic contextual distance measurement, as well as the vanilla contextual distance measurement for reference. Here we highlighted the difference between SPCX and CX that denotes context difference.