Towards Robust Blind Face Restoration with Codebook Lookup Transformer

Anonymous Author(s)
Affiliation
Address
email

Abstract

Blind face restoration is a highly ill-posed problem that often requires explicit auxiliary guidance to improve the mapping from a degraded input to a desired restored output. In this paper, we demonstrate that the uncertainty and ambiguity of the mapping can be largely reduced by casting face restoration as a code prediction task in a small, finite proxy feature space. Under this paradigm, we propose a Transformer-based prediction network, named *CodeFormer*, to exploit global contexts of the input for *code prediction*, enabling the discovery of a natural face that closely approximates the target high-quality image even when the input is severely degraded. To further enhance identity preservation, we propose a controllable feature transformation module to control information flow from the input image. Such a design allows a flexible trade-off between fidelity and quality so that one could reduce the reliance on the input image in case of heavy degradation. Equipped with the proposed components, our *CodeFormer* suggests superior robustness against degradation, outperforming state of the arts in both quality and fidelity. Extensive analysis is also conducted to verify the effectiveness of our method.

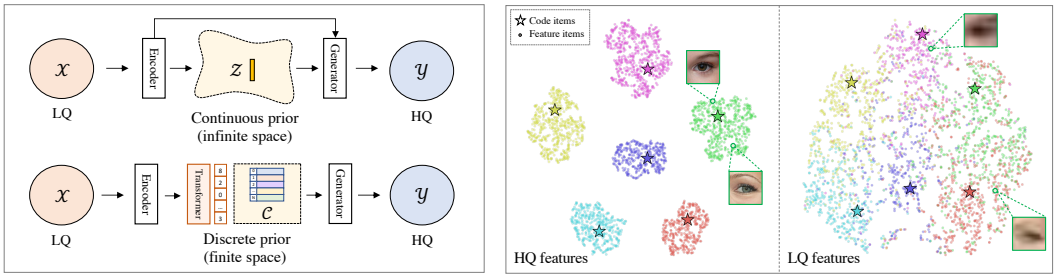
1 Introduction

Face images captured in the wild often suffer from various degradation, such as compression, blur, and noise. Restoring such images is highly ill-posed as the information loss induced by the degradation leads to infinite plausible high-quality (HQ) outputs given a low-quality (LQ) input. The ill-posedness is further elevated in blind restoration, where the specific degradation is unknown. Despite the progress made with the emerge of deep learning, learning a LQ-HQ mapping without additional guidance in the huge image space is still intractable, leading to the suboptimal restoration quality of earlier approaches. To improve output quality, auxiliary information that 1) reduces the uncertainty of LQ-HQ mapping and 2) complements high-quality details is indispensable.

Various priors have been used to mitigate the ill-posedness of the problem, including geometric priors [4, 5, 27, 40], reference priors [21–23], and generative priors [2, 34, 39]. While improved textures and details are observed, these approaches often suffer from *limited expressiveness* and *high sensitivity to degradation*. Such priors provide insufficient guidance for restoration, and the networks essentially resort to the information in the input LQ images, which is usually highly corrupted. As a result, the output quality is deteriorated by the degradation of the input image, and the mapping uncertainty still exists.

Different from the aforementioned approaches, this work casts face restoration as a code prediction task in a small discrete space, and exploits the learnable and discrete codebook prior to improve

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Frameworks of continuous prior (top) and our discrete prior (bottom)

(b) Distributions of HQ (left) / LQ (right) features and the codebook items

(c) Input (d) PULSE (continuous) (e) GFP-GAN (continuous) (f) Nearest Neighbor (discrete) (g) CodeFormer (discrete) (h) Code GT (discrete) (i) Ground Truth

Figure 1: An illustration of motivation. (a) Restoration frameworks of StyleGAN2 based continuous prior (top) and our discrete codebook prior (bottom). (b) t-SNE [32] visualization for HQ/LQ face features and codebook items. (c) LQ input. (d-e) Results of existing methods with continuous prior (PULSE [24] and GFP-GAN [34]). (f-g) Results of discrete prior (Nearest Neighbor [9, 31] and our CodeFormer). (h) Reconstruction results from the code sequence ground truth. (i) Ground truth. As shown, (d) without skip connection shows the low fidelity. (e) with skip connection alleviates identity issues but introduces notable artifacts. (f) utilizing nearest neighbor matching for code lookup can maintain more accurate facial structure than the ones with continuous priors, but details such as glasses are not correctly restored and some artifacts are left behind. (g) employing Transformer for code retrieval, the proposed CodeFormer generates best results with both high quality and fidelity.

feature expressiveness as well as the robustness against degradation. Such an approach has been extensively deployed in the task of image synthesis [3, 9, 31], but its adoption in restoration remains non-trivial. As depicted in Fig. 1(b)(right), even when the encoder is trained with degradation, the features extracted from the corrupted images are not well clustered, and hence nearest-neighbor feature matching is unrealiable in such case.

In this study, we propose a robust Transformer-based code prediction network, named *CodeFormer*, to exploit global compositions and long-range dependencies for enhanced predictions. Specifically, taking the input features, a Transformer module is employed to predict the code sequence as a discrete image representation in the learned codebook space that perceptually approximates to the image space, as shown in Fig. 1(h). Furthermore, we introduce a *controllable feature transformation* module with an adjustable coefficient to control the relative importance of the input. Such a design allows a flexible trade-off between *fidelity* and *quality*, where one could reduce the reliance on the input image to produce high-quality outputs in case of heavy degradation.

Our approach is different from existing works in the following aspects: (1) Most existing works [9, 31] adopting a vector-quantized codebook employs nearest-neighbor matching, which is infeasible when the input image is severely corrupted. Tailored for restoration, we propose a Transformer-based prediction module to incorporate global information. Comparing the results depicted in Fig. 1(f)-(g), the proposed *CodeFormer* is able to recover more details than the nearest-neighbor matching, such as the glasses and the eyebrows. (2) In contrast to generative priors [9, 34, 39] (Fig. 1(a)), the combinations of codebook items form a discrete representation space with only finite cardinality. Through mapping the input image to a much smaller proxy space, the uncertainty of the restoration is attenuated, promoting robustness against degradation. With reduced uncertainty, the features possess greater expressiveness, reducing the reliance on the input image and also improving the robustness against degradation, as shown in Fig. 1(c)-(i). (3) Unlike the hand-crafted reference priors [21–23], our dictionary (*i.e.*, codebook) automatically learns optimal elements for face restoration, providing additional flexibility and circumventing the laborious dictionary design.

Equipped with our components, our proposed *CodeFormer* demonstrates superior performance in existing datasets and also our newly introduced *WIDER-Test dataset* that consists of 970 severely

degraded faces collected from the WIDER-Face dataset [38]. In addition to face restoration, our method also demonstrates its effectiveness on other challenging tasks such as face inpainting, where long-range clues from other regions are required. Systematic studies and experiments are conducted to demonstrate the merits of our method over previous works.

2 Related Work

Blind Face Restoration. Existing blind face restoration methods can be categorized into 1) geometric prior-based, 2) reference prior-based, and 3) generative prior-based methods. Since face is highly structured, geometric priors of faces are exploited for blind face restoration. Some methods introduce facial landmarks [5], face parsing maps [4, 27, 37], facial component heatmaps [40] or 3D shapes [26, 14] in their designs. However, such prior information cannot be accurately acquired from degraded faces. Moreover, geometric priors are unable to provide rich details for high-quality face restoration.

Reference-based approaches have been proposed to circumvent the above limitations. These methods generally require the references to possess same identity as the input degraded face. For example, Li *et al.* [23] propose a guided face restoration network that consists of a warping subnetwork and a reconstruction subnetwork, and a high-quality guided image of the same identity as input is used for better restoring the facial details. However, such references are not always available. To alleviate this issue, DFDNet [21] pre-constructs dictionaries composed of high-quality facial component features. However, the component-specific dictionary features are insufficient for high-quality face restoration, especially for regions out of the dictionary scope (*e.g.*, skin, hair). Moreover, the large size ($\sim$3.2G) of the dictionary limits its efficiency.

The recent pre-trained GANs have provided a new insight for blind face restoration. The generative priors are obtained by projecting the input image into a latent feature space. Different strategies such as iterative optimization [24] and multiple-code optimization [10] are proposed for effective GAN inversion. However, preserving high fidelity of the restored faces is challenging when one maps the degraded faces into the low-dimensional latent space. To relieve this issue, GLEAN [2], GPEN [39], and GFPGAN [34] embed the generative prior into encoder-decoder network structures, with additional structural information as guidance. Despite the improvement of fidelity, these methods still cannot bypass the intrinsic limitations of generative priors for faithful restoration.

Dictionary Learning. Sparse dictionary learning in traditional methods aims to learn an over-complete dictionary as a set of basis vectors such that input signals can be expressed as sparse linear combinations of tokens in the dictionary. Sparse representation with learned dictionaries has demonstrated its superiority in image restoration tasks, such as super-resolution [11, 29, 30, 36] and denoising [8]. Their high-level insight of learning a dictionary to store HQ texture clues has inspired deep learning-based image restoration [16] and synthesis methods [9, 31]. Jo and Kim [16] construct a look-up table (LUT) by transferring the network output values to a LUT, so that only a simple value retrieval is needed during inference. However, storing HQ textures in the image domain usually requires a huge LUT, limiting its practicality. Different from LUT, VQVAE [31] automatically learns a highly compressed codebook via a vector quantized autoencoder model. VQGAN [9] further adopts the adversarial loss and perceptual loss to maintain perceptual quality at a high compression rate, significantly reducing the codebook size without sacrificing its expressiveness. Inspired by the learnable discrete codebook, this paper investigates a discrete space for blind face restoration, and we tackle the challenges specific in restoration.

3 Methodology

The main focus of this work is to exploit a discrete representation space that reduces the uncertainty of restoration mapping and complements high-quality details for the degraded inputs. Since local textures and details are lost and corrupted in low-quality inputs, we employ a Transformer module that considers global composition of natural faces for accurate code prediction, enabling high-quality restoration. The overall framework is illustrated in Fig. 2.

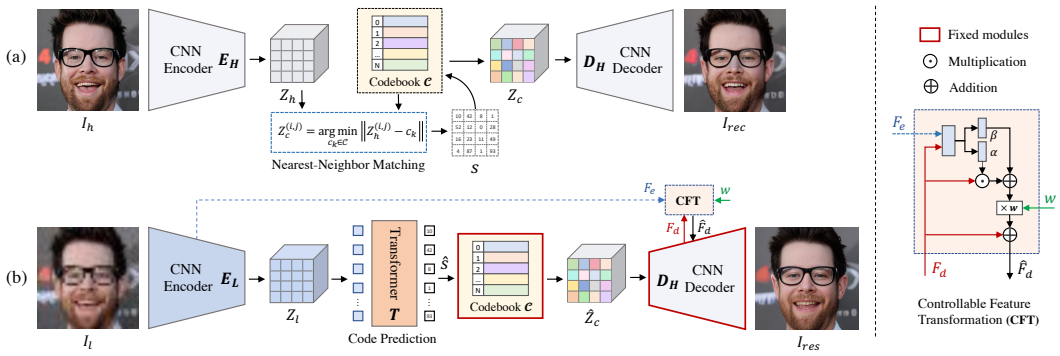


Figure 2: Framework of CodeFormer. (a) We first learn a discrete codebook and a decoder to store high-quality visual parts of face images, through autoencoder learning of self-reconstruction. (b) With the codebook and fixed decoder, we then employ a Transformer for code sequence prediction, modeling the global face composition from the low-quality inputs. Besides, a controllable feature transformation module is used to adjust the relative importance of the inputs. Note that this connection is optional - it can be disabled to avoid adverse effects when the inputs are severely degraded, and one can adjust a scalar weight w to trade between *quality* and *fidelity*.

We first incorporate the idea of quantized vector [9, 31] and pre-train an autoencoder through reconstruction to obtain a discrete codebook and the corresponding decoder (Sec. 3.1). The prior captured by the codebook and decoder is then used for restoration. Based on this prior, we then employ a Transformer for accurate code prediction, through modeling the global face composition from the low-quality inputs (Sec. 3.2). In addition, a controllable feature transformation module is introduced for further enhancing fidelity without compromising quality (Sec. 3.3). The training of our method is divided into three stages accordingly.

3.1 Codebook Learning (Stage I)

To reduce uncertainty of the LQ-HQ mapping and complement high-quality details for restoration, we employ a learnable and discrete codebook prior, which improves feature expressiveness as well as robustness against degradation.

As shown in Fig. 2(a), the HQ face image $I_h \in \mathbb{R}^{H\times W\times 3}$ is first embeded as a compressed feature $Z_h \in \mathbb{R}^{m\times n\times d}$ by an encoder E_H. Following VQVAE [31] and VQGAN [9], we replace each "pixel" in Z_h with the nearest item in the codebook $\mathcal{C} = \{c_k \in \mathbb{R}^d\}_{k=0}^N$ to obtain the quantized feature $Z_c \in \mathbb{R}^{m\times n\times d}$ and the corresponding code entries sequence $s \in \{0, \cdots, N-1\}^{m\times n}$:

$$Z_c^{(i,j)} = \arg\min_{c_k\in\mathcal{C}} \|Z_h^{(i,j)} - c_k\|_2; \quad s^{(i,j)} = \arg\min_k \|Z_h^{(i,j)} - c_k\|_2, \tag{1}$$

The decoder D_H then reconstructs the high-quality face image I_{rec} given Z_c. Thus, a new latent discrete representation s in form of a $m \cdot n$ code token sequence that specify the respective code index in the learned codebook, i.e., $Z_c^{(i,j)} = c_k$ when $s^{(i,j)} = k$.

Training Objectives. We pre-train a quantized autoencoder to learn a context-rich codebook using three image-level losses: L1 loss $\mathcal{L}_1$, perceptual loss [17, 41] $\mathcal{L}_{per}$, and adversarial loss [9] $\mathcal{L}_{adv}$:

$$\mathcal{L}_1 = \|I_h - I_{rec}\|_1; \quad \mathcal{L}_{per} = \|\Phi(I_h) - \Phi(I_{rec})\|_2^2; \quad \mathcal{L}_{adv} = [\log D(I_h) + \log(1 - D(I_{rec}))], \tag{2}$$

where Φ denotes the feature extractor of VGG19 [28]. Since, image-level losses are underconstrained when updating the codebook items, we also adopt the intermediate code-level loss [31, 9] $\mathcal{L}_{code}^{feat}$ to reduce the distance between codebook $\mathcal{C}$ and input embeddings Z_h:

$$\mathcal{L}_{code}^{feat} = \|\text{sg}(Z_h) - Z_c\|_2^2 + \beta\|Z_h - \text{sg}(Z_c)\|_2^2, \tag{3}$$

where $\text{sg}(\cdot)$ stands for the stop-gradient operator and $\beta = 0.25$ is a weight trade-off for the update rates of the encoder and codebook. Since the quantization operation in Eq. (1) is non-differentiable, we adopt straight-through gradient estimator [9, 31] to copy the gradients from the decoder to the encoder. The complete objective of codebook learning $\mathcal{L}_{codebook}$ is:

$$\mathcal{L}_{codebook} = \mathcal{L}_1 + \mathcal{L}_{per} + \mathcal{L}_{code}^{feat} + \lambda_{adv} \cdot \mathcal{L}_{adv}, \tag{4}$$

where λ_{adv} is set to 0.8 in our experiments.

Codebook Settings. Our encoder E_H and decoder D_H consist of 12 residual blocks and five resize layers for downsampling and upsampling, respectively. We obtain a large compression ratio of $r = H/n = W/m = 32$. As we will show later, this compression ratio leads to a great robustness against degradation and a tractable computational cost. Although more codebook items could ease reconstruction, the redundant elements could cause ambiguity in subsequent code predictions. Hence, we set the number of elements in the codebook to $N = 1024$, which is sufficient for accurate face reconstruction. Besides, the code dimension d is set to 256.

3.2 Codebook Lookup Transformer Learning (Stage II)

Due to corruptions of textures in LQ faces, the nearest-neighbour (NN) matching in Eq. (1) usually fails in finding accurate codes for face restoration. As depicted in Fig. 1(b), LQ features with diverse degradation could deviate from the correct code and be grouped into nearby clusters, resulting in undesirable restoration results, as shown in Fig. 1(f). To mitigate the problem, we employ a Transformer to model global interrelations for better code prediction.

Built upon the learned model presented in Sec. 3.1, as shown in Fig. 2(b), we append a Transformer [33] module containing nine self-attention blocks to the encoder. We fix the codebook $\mathcal{C}$ and decoder D_H and finetune the encoder E_H for restoration. The finetuned encoder is denoted as E_L. To obtain the LQ embeddings $Z_l \in \mathbb{R}^{m \times n \times d}$ through E_L, we first unfold the embeddings into $m \cdot n$ vectors $Z_l^v \in \mathbb{R}^{(m \cdot n) \times d}$. The s-th self-attention block of Transformer computes the as following:

$$X_{s+1} = \text{Softmax}(Q_s K_s) V_s + X_s, \tag{5}$$

where $X_0 = Z_l^v$. The queries Q, keys K, and values V are obtained from X_s through linear layers. We add a sinusoidal positional embedding [1, 6] $\mathcal{P} \in \mathbb{R}^{(m \cdot n) \times d}$ on the queries Q and the keys K to increase its expressiveness of modeling sequential representation of codebook. Finally, taking the encoding feature Z_l^v as an input, the Transformer predicts the $m \cdot n$ code sequence $\hat{s} \in \{0, \cdots, |N| - 1\}^{m \cdot n}$ in form of the probability of the N-way classification. The predicted code sequence then retrieves the $m \cdot n$ respective code items from the learned codebook, forming the quantized feature $\hat{Z}_c \in \mathbb{R}^{m \times n \times d}$ that produces a high-quality face image through the fixed decoder D_H. Thanks to our large compression ratio (*i.e.*, 32), our Transformer is effective and efficient in modeling global relationships of LQ images.

Training Objectives. We finetune the CNN encoder E_L and Transformer blocks for restoration, while the codebook and decoder are kept fixed. Instead of employing reconstruction loss and adversarial loss in the image-level, only code-level losses are required in this stage: 1) cross-entropy loss $\mathcal{L}_{code}^{token}$ for code token prediction supervision, and 2) L2 loss $\mathcal{L}_{code}^{feat'}$ on code features to force LQ embedding Z_l to approach the HQ embedding Z_c, which eases the difficulty of token prediction learning:

$$\mathcal{L}_{code}^{token} = \sum_{i=0}^{mn-1} -s_i \log(\hat{s}_i); \quad \mathcal{L}_{code}^{feat'} = \|Z_l - \text{sg}(Z_c)\|_2^2, \tag{6}$$

where the ground truth of latent code s_i is obtained from the pre-trained autoencoder in Stage I (Sec. 3.1), thus the HQ embedding Z_c is then retrieved according to the s_i. The final objective of Transformer learning is:

$$\mathcal{L}_{tf} = \lambda_{token} \cdot \mathcal{L}_{code}^{token} + \mathcal{L}_{code}^{feat'}, \tag{7}$$

where λ_{token} is set to 0.5 in our method. Note that the trained model in this stage has already demonstrated robustness and effectiveness in face restoration, especially for severely degraded faces.

3.3 Controllable Feature Transformation (Stage III)

Despite our Stage II model has maintained a good image fidelity, we also investigate a flexible tradeoff between *fidelity* and *quality* for face restoration. Thus, we propose the controllable feature transformation (CFT) module to control information flow from the LQ encoder E_L, which enhances the fidelity without compromising the restoration quality. Specifically, as shown in Fig. 2, the LQ

features F_e are used to slightly tune the decoder features F_d through feature transformation [35] with the predicted affine parameters of α and β. An adjustable coefficient $w \in [0, 1]$ is then used to control the relative importance of the input:

$$\hat{F}_d = F_d + (\alpha \odot F_d + \beta) \times w. \tag{8}$$

We adopt this module at multiple scales $s \in \{32, 64, 128, 256\}$ between encoder and decoder. Such a design allows a flexible trade-off between fidelity and quality, where one could use a small w to reduce the reliance on the input image, thus producing high-quality outputs in case of heavy degradation. On the contrary, a large w will introduce more input information to enhance the fidelity in case of mild degradation. Unless otherwise stated, we set the $w = 0$ by default in our experiments.

Training Objectives. To train the controllable modules and finetune the encoder E_L in this stage, we keep the code-level losses of $\mathcal{L}_{tf}$ in Stage II, and also add same image-level losses in Stage I except that I_{rec} is replaced by I_{res}. The complete loss for this stage is the sum of above losses weighted with their original weight factors. We set the w to 1 during training of this stage, which then allows network to achieve continuous transitions of results by adjusting w in $[0, 1]$ during inference.

4 Experiments

4.1 Datasets

Training Dataset. We train our models on the FFHQ dataset [19], which contains 70,000 high-quality (HQ) images, and all images are resized to 512×512 for training. To form training pairs, we synthesize LQ images I_l from the HQ counterparts I_h with the following degradation model [21, 34, 39]:

$$I_l = \{[(I_h \otimes k_\sigma)_{\downarrow_r} + n_\delta]_{\text{JPEG}_q}\}_{\uparrow_r}, \tag{9}$$

where the HQ image I_h is first convolved with a Gaussian kernel k_σ, followed by a downsampling of scale r. After that, additive Gaussian noise n_δ is added to the images, and then JPEG compression with quality factor q is applied. Finally, the LQ images are resize back to 512×512. We randomly sample σ, r, δ, and q from $[1, 15]$, $[1, 30]$, $[0, 20]$, and $[30, 90]$, respectively.

Testing Datasets. We evaluate our method on a synthetic dataset CelebA-Test and three real-world datasets: LFW-Test, WebPhoto-Test, and our proposed WIDER-Test. CelebA-Test contains 3,000 images selected from the CelebA-HQ dataset [18], and the LQ images are synthesized under the same degradation range as our training settings. The three real-world datasets respectively contain three different degrees of degradation, *i.e.*, mild (LFW-Test), medium (WebPhoto-Test), and heavy (WIDER-Test). LFW-Test consists of the first image of each person whose name starting with `A` in LFW dataset [15], containing 431 images. WebPhoto-Test [34] consists of 407 low-quality faces collected from the Internet. Our WIDER-Test comprises 970 severely degraded face images from the WIDER Face dataset [38], providing a more challenging dataset for blind face restoration.

4.2 Experimental Settings and Metrics

Settings. We represent an input image of $512 \times 512 \times 3$ as 16×16 code sequences. For all stages of training, we use the Adam [20] optimizer with a batch size of 16. We set the learning rate to 8×10^{-5} for stages I and II, and adopt a smaller learning rate of 2×10^{-5} for stage III. The three stages are trained with 1.5M, 200K, and 20K iterations, respectively. Our method is implemented with the PyTorch framework and trained using four NVIDIA Tesla V100 GPUs.

Metrics. For the evaluation on real-world datasets, we employ the widely-used non-reference perceptual metrics: FID [13] and NIQE [25]. For the evaluation on CelebA-Test with ground truth, we adopt PSNR, SSIM, and LPIPS [41] as our metrics. We also evaluate the identity preservation using the cosine similarity of features from ArcFace network [7], denoted as IDS.

4.3 Comparisons with State-of-the-art Methods

We compare our approach with current state-of-the-art methods, including PULSE [24], DFDNet [21], PSFRGAN [4], GLEAN [2], GFP-GAN [34], and GPEN [39]. We conduct extensive comparisons on synthetic and real-world datasets. More results are shown in the supplementary material.

Figure 3: Qualitative comparison on the CelebA-Test. Our CodeFormer produces high-quality faces with faithful details. (**Zoom in for best view.**)

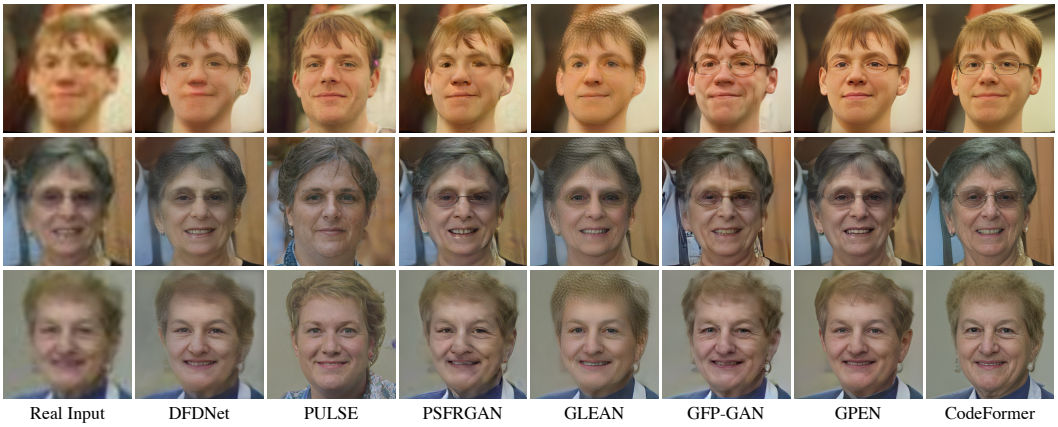


Figure 4: Qualitative comparison on real-world faces. Our CodeFormer is able to restore high-quality faces, showing robustness to the heavy degradation. (**Zoom in for best view.**)

Evaluation on Synthetic Dataset. We first show the quantitative comparison on the CelebA-Test in Table 1. In terms of the image quality metrics LPIPS, FID, and NIQE, CodeFormer achieves the best scores than existing methods. Besides, it also faithfully preserves the identities, reflected by the highest IDS score and PSNR. Additionally, we present the qualitative comparison in Fig. 3. The compared methods fail to produce pleasant restoration results. For example, DFDNet [21], PSFRGAN [4], GFP-GAN [34], and GPEN [39] introduce artifacts and GLEAN [2] produces over-smoothed results. Moreover, all other methods are unable to preserve the identity. Thanks to the effective code prediction in the discrete and finite space, CodeFormer not only produces high-quality faces but also preserves the identifies well, even when the input faces are highly degraded.

Table 1: Quantitative comparison on the **CelebA-Test**. Red and blue indicate the best and the second best performance, respectively. The result of Code GT is the upper bound of CodeFormer.

Methods	LPIPS↓	FID↓	NIQE ↓	IDS↑	PSNR↑	SSIM↑
Input	0.712	295.73	18.67	0.32	21.53	0.623
PULSE [24]	0.406	72.94	5.71	0.30	21.38	0.571
DFDNet [21]	0.466	85.15	7.24	0.42	21.24	0.562
PSFRGAN [4]	0.395	62.05	6.59	0.43	20.91	0.549
GLEAN [2]	0.371	59.87	5.17	0.51	21.59	0.598
GFP-GAN [34]	0.391	58.36	4.87	0.42	20.37	0.545
GPEN [39]	0.349	59.70	4.71	0.54	21.26	0.565
CodeFormer (ours)	0.307	57.01	4.47	0.59	21.82	0.612
Code GT	0.124	54.31	4.33	0.89	25.43	0.749
GT	0	51.40	3.84	1	∞	1

Table 2: Quantitative comparison on the *real-world* **LFW-Test**, **WebPhoto-Test**, and **WIDER-Test**. Red and blue indicate the best and the second best performance, respectively.

Dataset	**LFW-Test**		**WebPhoto-Test**		**WIDER-Test**	
Degradation	mild		medium		heavy	
Methods	FID↓	NIQE ↓	FID↓	NIQE ↓	FID↓	NIQE ↓
Input	151.82	12.84	170.11	12.76	202.06	13.40
PULSE [24]	73.11	5.23	86.45	5.15	73.59	3.72
DFDNet [21]	77.32	4.83	100.68	5.29	57.84	5.50
PSFRGAN [4]	70.24	5.08	88.45	5.58	51.16	3.85
GLEAN [2]	71.29	4.25	105.63	5.18	47.11	5.22
GFP-GAN [34]	70.05	4.29	87.35	4.14	40.59	3.74
GPEN [39]	70.78	4.01	81.77	4.29	46.99	3.93
CodeFormer (ours)	70.18	4.12	86.03	4.61	40.32	3.90

Evaluation on Real-world Datasets. As presented in Table 2, our CodeFormer achieves comparable perceptual quality with the compared methods on the real-world testing datasets with mild and medium degradation and the best scores on the testing dataset with heavy degradation. Although PULSE [24] also obtains good perceptual scores, it cannot preserve the identity of input images, as

the FID scores and visual results respectively suggested in Table 1 and Fig. 1(d). From the visual comparisons in Fig. 4, we see that our method generalizes well to real-world face restoration and produces most visually pleasing results. Notably, CodeFormer successfully preserves the identity and produces natural results with rich details.

4.4 Ablation Studies

Exp.	Networks			Code Lookup		Metrics	
	Codebook	Transformer	Fix Decoder	NN	Code Pred.	LPIPS↓	IDS↑
(a)			✓			0.420	0.47
(b)	✓		✓	✓		0.397	0.51
(c)	✓		✓		✓	0.351	0.55
(e)	✓	✓			✓	0.379	0.52
(f) (ours, w=1)	✓	✓	✓		✓	0.393	**0.61**
(g) (ours, w=0)	✓	✓	✓		✓	**0.307**	0.59

Table 3: Ablation studies of variant networks and code lookup methods on the CelebA-Test. Removing 'Codebook' means the network is a general encoder-decoder structure. 'w' is an adjustable coefficient of CFT modules that controls the information flow from encoder.

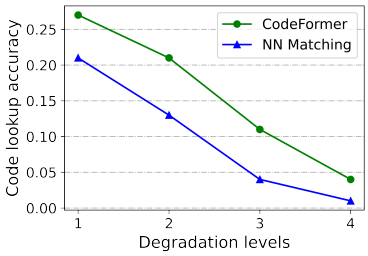


Figure 5: Curve comparison on code sequence prediction accuracy.

Effectiveness of Codebook Space. We first investigate the effectiveness of the codebook space. As shown in Exp. (a) of Table 3, removing the codebook (*i.e.*, directly feeding the encoder features to the decoder) results in worse LPIPS and IDS scores. The results suggest that the discrete space of codebook is the key to guarantee the robustness of our model.

Superiority of Transformer-based Codebook Prediction. To verify the superiority of our Transformer-based codebook prediction, we compare it with two different solutions, *i.e.*, nearest-neighbour (NN) matching and a CNN-based code prediction module, denoted as Code Pred., (*i.e.*, encoder E_L followed by a Linear prediction layer). As shown in Table 3, the comparison of Exp. (b) and Exp. (c) indicates that adopting a code prediction module is more effective than optimizing the features for NN-matching. However, the local nature of convolution operation of CNNs restricts its modeling capability for long code sequence prediction. In comparison to the CNN-based method, *i.e.*, Exp. (c), our Transformer-based solution produces higher-fidelity results. The superiority of our Transformer-based module is also demonstrated in Fig. 5, Fig. 6, and Fig. 8.

Figure 6: Qualitative comparisons of different codebook lookup methods.

Importance of Fixed Decoder. Unlike DFDNet [21], we adopt a compact codebook $\mathcal{C} \in \mathbb{R}^{N \times d}$ with N=1024, d=256. We hypothesize that the efficacy of our method is brought by the codebook coupled with the pre-trained decoder, instead of the codebook alone. To verify our hypothesis, we conduct two studies: 1) fixing both the codebook and decoder, *i.e.*, Exp. (g), and 2) fine-tuning the decoder, *i.e.*, Exp. (e), in the training process of stage II. As shown in Table 3, fine-tuning decoder deteriorates the performance. This is because fine-tuning the decoder breaks the tight connections between the codebook and the decoder, resulting in suboptimal performance. Therefore, we keep the decoder fixed throughout training.

Flexibility of Controllable Feature Transformation Module. Considering the diverse degradation in real-world face images, we provide a controllable featrue transformation module (CFT) to allow a flexible trade-off between fidelity and quality. As shown in Fig. 7, a smaller w tends to produce a high-quality output while a larger w improves the fidelity of output. Such an appealing flexibility improves the adaptiveness of our method for different scenarios, but is rarely touched in previous works. In addition, as shown in Table 3, Exp. (f), *i.e.*, setting the coefficient w to 1 increases the identity score but decreases the quality score. In this work, we prioritize quality, and hence the

Figure 7: CFT module is capable to generate continuous transitions between image quality and fidelity.

coefficient w is set to 0 by default. Note that one could adjust w without retraining to suit the requirement of the tasks.

4.5 Extension

The proposed Codeformer can be easily extended to face inpainting, and it shows great performance even in large mask ratios. To build training pairs, we use a publicly available script [39] to randomly draw irregular polyline masks for generating masked faces. We compare our method with two state-of-the-art face inpainting methods of CTSDG [12] and GPEN [39], as well as Nearest-Neighbor (NN) matching for code retrieval. As shown in Fig. 8, CTSDG and GPEN show less effectiveness for masked faces with the large masks. The NN matching based method reconstructs the face structures, but it also fails in restoring complete visual parts such as the glasses and the eyes. In contrast, CodeFormer generates high-quality faces without strokes and artifacts.

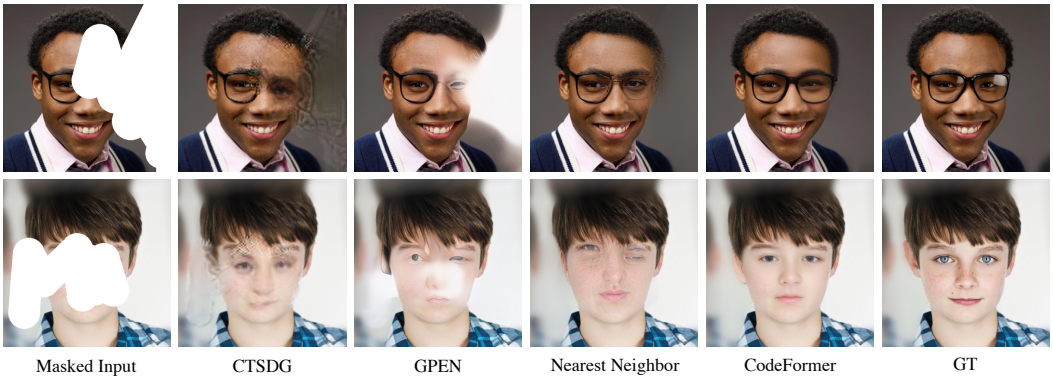


Figure 8: Visual comparison with state-of-the-art face inpainting methods on the challenging cases.

4.6 Limitation

Our method is built on a pre-trained codebook and decoder. Thus, their capability may affect the performance of our method. For example, though the identity issue is significantly relieved by the Transformer's global dependency modeling, it still exists in some rare visual parts such as accessories, where the current codebook space cannot seamlessly replace the image space. Using multiple scales in the codebook space to explore more fine-grained visual representations may be a solution.

5 Conclusion

This paper addresses the fundamental challenges of generating high-quality faces when inputs are severely corrupted. Through learning a small discrete codebook space, we turn face restoration to discrete code prediction, significantly reducing the uncertainty of restoration mapping and easing the learning of restoration. We explore global interaction from degraded inputs and model the code composition of face images via an expressive Transformer for accurate code prediction. Benefiting from these designs, our method shows strong robustness against heavy degradation. To enhance fidelity without compromising the restoration quality, we also propose a controllable feature transformation module that allows a flexible trade-off control between fidelity and quality. Experimental results suggest the superiority and robustness of our method.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[2] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In *CVPR*, 2021.

[3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022.

[4] 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 *CVPR*, 2021.

[5] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *CVPR*, 2018.

[6] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[8] Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, 2006.

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021.

[10] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In *CVPR*, 2020.

[11] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *ICCV*, 2015.

[12] Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *ICCV*, 2021.

[13] 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. In *NeurIPS*, 2017.

[14] Xiaobin Hu, Wenqi Ren, John LaMaster, Xiaochun Cao, Xiaoming Li, Zechao Li, Bjoern Menze, and Wei Liu. Face super-resolution guided by 3d facial priors. In *ECCV*, 2020.

[15] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008.

[16] Younghyun Jo and Seon Joo Kim. Practical single-image super-resolution using look-up table. In *CVPR*, 2021.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[21] Xiaoming Li, Chaofeng Chen, Shangchen Zhou, Xianhui Lin, Wangmeng Zuo, and Lei Zhang. Blind face restoration via deep multi-scale component dictionaries. In *ECCV*, 2020.

[22] Xiaoming Li, Wenyu Li, Dongwei Ren, Hongzhi Zhang, Meng Wang, and Wangmeng Zuo. Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In *CVPR*, 2020.

[23] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. In *ECCV*, 2018.

[24] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *CVPR*, 2020.

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

[26] Wenqi Ren, Jiaolong Yang, Senyou Deng, David Wipf, Xiaochun Cao, and Xin Tong. Face video deblurring using 3d facial priors. In *ICCV*, 2019.

[27] Ziyi Shen, Wei-Sheng Lai, Tingfa Xu, Jan Kautz, and Ming-Hsuan Yang. Deep semantic face deblurring. In *CVPR*, 2018.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[29] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *ICCV*, 2013.

[30] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *ACCV*, 2014.

[31] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *NeurIPS*, 2017.

[32] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *The journal of machine learning research*, 15(1):3221–3245, 2014.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[34] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, 2021.

[35] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, 2018.

[36] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[37] Lingbo Yang, Shanshe Wang, Siwei Ma, Wen Gao, Chang Liu, Pan Wang, and Peiran Ren. Hifacegan: Face renovation via collaborative suppression and replenishment. In *ACM MM*, 2020.

[38] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, 2016.

[39] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *CVPR*, 2021.

[40] Xin Yu, Basura Fernando, Bernard Ghanem, Fatih Porikli, and Richard Hartley. Face super-resolution guided by facial component heatmaps. In *ECCV*, 2018.

[41] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplemental material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data will be released once this paper is accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]