DLFormer: Discrete Latent Transformer for Video Inpainting

Jingjing Ren[1,2]*, Qingqing Zheng[3]†, Yuanyuan Zhao[2], Xuemiao Xu[1]†, Chen Li[2]
[1]School of Computer Science and Engineering, South China University of Technology
[2]WeChat, Tencent Inc. [3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

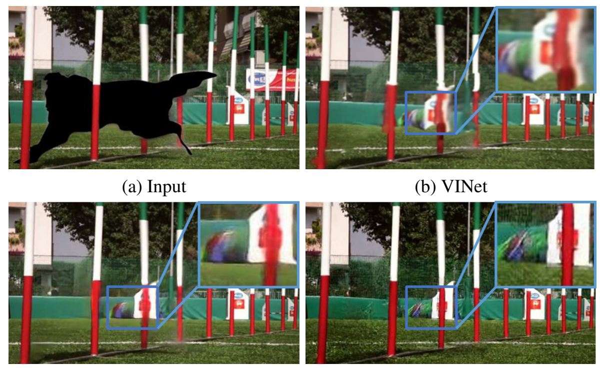
Abstract

*Video inpainting remains a challenging problem to fill with plausible and coherent content in unknown areas in video frames despite the prevalence of data-driven methods. Although various transformer-based architectures yield promising result for this task, they still suffer from hallucinating blurry contents and long-term spatial-temporal inconsistency. While noticing the capability of discrete representation for complex reasoning and predictive learning, we propose a novel Discrete Latent Transformer (*DLFormer*) to reformulate video inpainting tasks into the discrete latent space rather the previous continuous feature space. Specifically, we first learn a unique compact discrete codebook and the corresponding autoencoder to represent the target video. Built upon these representative discrete codes obtained from the entire target video, the subsequent discrete latent transformer is capable to infer proper codes for unknown areas under a self-attention mechanism, and thus produces fine-grained content with long-term spatial-temporal consistency. Moreover, we further explicitly enforce the short-term consistency to relieve temporal visual jitters via a temporal aggregation block among adjacent frames. We conduct comprehensive quantitative and qualitative evaluations to demonstrate that our method significantly outperforms other state-of-the-art approaches in reconstructing visually-plausible and spatial-temporal coherent content with fine-grained details.Code is available at* `https://github.com/JingjingRenabc/dlformer`.

1. Introduction

Video inpainting aims to fill in corrupted regions with meaningful details such that the completed video is consistent both spatially and temporally. It can be applied to various industrial applications, including video restoration [15, 34], unwanted object removal [18, 19] and video retargeting [32].

[1]This work was done while Jingjing Ren was an intern at Tencent.
[2]Qingqing Zheng and Xuemiao Xu are the joint corresponding authors.



(a) Input (b) VINet (c) STTN (d) Ours

Figure 1. Previous methods, like (b) VINet [9] and (c) STTN [34] formulate in a continuous feature space and usually produce artifacts and blurry results around the occluded bars and background. In contrast, our method (d) fills the unknown region with plausible content even in the swift movement case by formulating this problem in a global discrete latent space (please zoom in for better visualization).

Recently, methods [2, 9, 37] have made great progress in this task thanks to the powerful CNN-based deep features extractors. These methods still suffer from limited receptive field along temporal domain and produce blurry and misplacement artifacts in the completed video, as shown in Figure 1 (b). The state-of-the-art methods [12, 16, 34] tend to capture long-term correspondences with attention mechanism, so the available content at distant frames can be globally propagated to the unknown regions. Although these attention-based methods yield promising results, trivially using pair-wise similarity in a continuous feature space, e,g., STTN [34], still suffers from blurry contents (refer to Figure 1 (c)) degrading the visual quality in high frequency areas. It is still challenging to generate plausible and coherent contents with fine-grained details, especially under complex and dynamic scenarios.

To tackle the aforementioned challenges, we propose a novel Discrete Latent Transformer (DLFormer) to model the video inpainting task as a code inference problem in a discrete latent space rather than in the continuous feature space. Benefiting from the Vector Quantized Variational AutoEncoder (VQ-VAE) [20], continuous representation of one image generated by an autoencoder can be quantized into limited discrete codes in latent space, spanned by a codebook to form a quantized feature. Such discrete codes, represented as the indices in the corresponding codebook, can be delivered back to the autoencoder to reconstruct the original image sufficiently. Inspired by this work and in order to capture the fine-grained details, we learn a video-specific and discriminative codebook as well as the corresponding autoencoder to represent the target video in the discrete latent space, which is spanned by a context-rich and efficient codebook. In this way, the obtained codebook naturally captures global discriminative features among the entire video sequence, even for unknown regions.

Based on this discrete latent representation, inpainting unknown regions with plausible content can be regarded as inferring the proper discrete code indices with a certain codebook. By adopting a self-supervised training strategy, the latent code distribution in valid regions can be naturally propagated to unknown regions via the proposed discrete latent transformer. Moreover, to avoid spatial-temporal visual jitters caused by such discrete prediction, we further explicitly enforce short-term consistency with a residual aggregation block before delivering the code inference results back to the autoencoder to generate the final inpainting results.

We extensively evaluate our method in both video restoration and object removal tasks on Youtube-VOS [31] and DAVIS [24] datasets and the experimental results demonstrate the proposed method significantly outperforms the state-of-the-art methods. Thanks to the robust discrete representation, the proposed DLFormer is able to fill visually-plausible and spatial-temporal coherent content with fine-grained details in unknown regions.

We summarize our contributions of this work as follows:

- To the best of our knowledge, we are the first to formulate the video inpainting task as a discrete code inference problem in the discrete latent space. Benefiting from such discrete representation, our method is capable to synthesize more plausible and fine-grained details than previous methods formulating in the continues feature space.

- Based on the aforementioned novel formulation, a discrete latent transformer is proposed to explicitly model the global code distribution among the entire video sequence with a self-attention mechanism. The proposed transformer is allowed to propagate such distribution from valid regions toward corrupted regions regardless of the limited temporal receptive field.

- We further develop a residual temporal aggregation block to relieve temporal visual jitters caused by the discrete prediction across adjacent frames.

2. Related Work

2.1. Video Inpainting

Traditional approaches usually extend from patch-based image inpainting methods [1] for video inpainting. For example, Patwardhan *et al*. [22,23] sampled the nearest neighbor patches to fill unknown regions with a greedy completion scheme under the assumption of the static camera or constrained camera motion. To address the challenge of dynamic camera motion, Wexler *et al*. [30] formulated a global optimization framework where spatial-temporal patches were alternating matched and reconstructed based on local structures. [6, 27] further extend [1, 30] to enhance temporal consistency by introducing flow information. Above traditional methods only local texture and structure information is used which is infeasible to represent complex motion and dynamic content in real world.

Recently, deep learning-based works [9, 10, 35] propose more efficient solutions and achieve great success for video inpainting. These deep video inpainting methods usually fall into three main streams: alignment-based, 3D convolution networks as well as attention-based approaches. Alignment-based methods [4, 12, 32] first align reference frames with either or both optical flow and affine transformation, then borrow information from known regions in the aligned reference frames. However, the above alignment methods are sensitive to motion prediction errors. 3D convolution networks [2, 29] employ 3D convolution to leverage temporal features from nearby frames. Inspired by [13], Zou *et al*. [37] further developed a 3D gated convolution with an embedded temporal shift module to save computation costs. 3D convolution networks are efficient to learn temporal features, but still fail to capture long-range information from distant frames due to the limited receptive field. To better model long-range correspondences, attention-based methods [5, 19] investigate attention mechanism, where similarities between corrupted regions and known regions are calculated as weights to fuse valid information. STTN [34] directly adopts the multi-head transformer architecture [28] and proposes a multi-scale generative model for video inpainting. Based on [34], [15] decouples the attention module along the spatial and temporal dimension to narrow the search space, and thus reduce computational complexity. FuseFormer [16] further splits features in a more fine-grained way compared with [15,34]. All above methods tend to suffer from blurry results, especially in high-frequency regions since they perform similarity evaluation and content generation on appearance features in a continuous space.

2.2. Discrete Representation Learning

The Vector Quantized Variational AutoEncoder (VQ-VAE) [20, 26] are generative models that encode high-

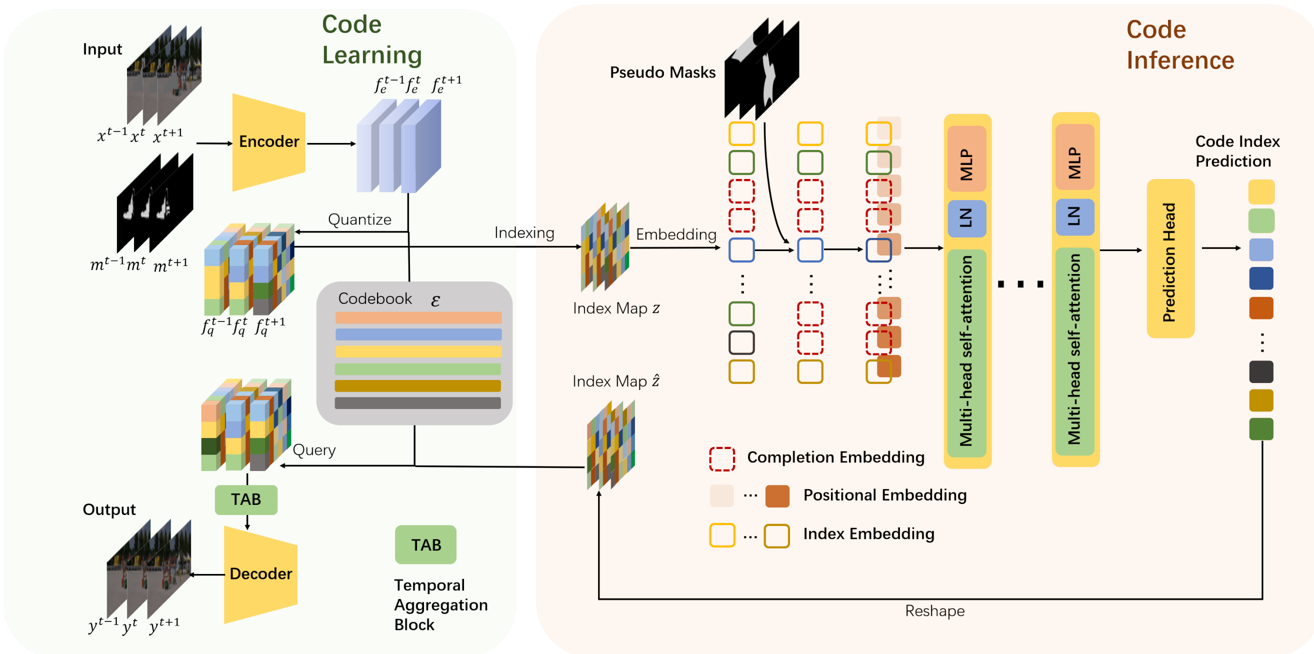


Figure 2. The overview of the proposed video inpainting network. It consists of two components: code learning and code inference. The code learning module learns compact discrete latent codes based on a rich codebook for video representation. With the discrete codes learned, the code inference module subsequently models the video inpainting with the transformer in the discrete latent space.

dimensional inputs into a lower-dimensional discrete latent space, and decode the generated latent representation back to inputs as closely as possible. With the discrete latent representation, they have demonstrated satisfactory reconstruction and generation quality [33, 36]. For example, Kaiser *et al.* [8] employed discrete variables to speed up the decoding process for neural machine translation. Esser *et al.* [3] adapted VQ-VAE by equipping the decoder with a discriminator to enhance details for image generation. Rakhimov *et al.* [25] proposed an autoregressive model to predict new frames in the latent space for video generation. However, to our best knowledge, the discrete representation has not yet been explored for video inpainting.

3. Method

Video inpainting aims to fill in spatial-temporal holes with visually contents of spatial-temporal consistency. Given a corrupted video sequence $\mathbf{X} = \{x^1, x^2, \cdots, x^T\}$ of height H, width W and T frames in RGB space $\mathcal{R}$, with corresponding annotated masks $\mathbf{M} = \{m^1, m^2, \cdots, m^T\}$ of the same resolution, we define a mapping $\mathcal{F}$ that encodes frame x in a RGB space $\mathcal{R}$ to a discrete latent space $z \in \mathcal{Z}$ with

$$\begin{aligned} &\mathcal{F}(x) = z,\ \mathcal{F}^{-1}(z) = \hat{x} \\ &s.t.\ x, \hat{x} \in \mathcal{R}, z \in \mathcal{Z}, \end{aligned} \tag{1}$$

where $\mathcal{F}^{-1}$ maps z back to $\hat{x}$ to reconstruct x. We use a codebook $\mathcal{E} = \{\mathbf{e}_k \in \mathbb{R}^d | k \in \{1, 2, \cdots, K\}\}$ containing K prototype vectors of d-dimension to describe $\mathcal{Z}$. z represents the index of the corresponding prototype vector in $\mathcal{E}$ for each spatial-temporal location. To this end, our goal is to learn $\mathcal{G}$ taking as input z and mask m, outputs index prediction map $\hat{z}$ such that $\mathcal{F}^{-1}(\hat{z})$ generates completed frame $y \in \mathcal{R}$ that is spatial-temporally consistent as

$$y = \mathcal{F}^{-1}(\mathcal{G}(z, m)). \tag{2}$$

As illustrated in Figure 2, the pipeline of our method consists of two components: code learning and code inference. In the code learning stage, we learn mapping $\mathcal{F}$ and its inversion $\mathcal{F}^{-1}$ by learning a context-rich and video-specific codebook $\mathcal{E}$ to construct a discrete latent space $\mathcal{Z}$ and represent frames as z in a latent discrete space as elaborated in Section 3.1. Then in the code inference stage, we obtain mapping $\mathcal{G}$ by formulating a transformer to propagate code constitution from seen regions to unknown regions as described in Section 3.2. Moreover, we further propose a temporal aggregation block (TAB) to leverage temporal information and explicitly enhance short-term temporal consistency as elaborated in Section 3.3.

3.1. Video-specific Discrete Code Learning

To leverage the highly effective transformer architecture for code index prediction, we train a variational autoencoder module to learn discrete codes for video representation, which can significantly compress the feature description length as well as relive the difficulty of content generation in unknown regions. Similar to VQ-VAE [20], the variational autoencoder module consists of an encoder E, which encodes the video frames into the continuous representation f_e, a codebook $\mathcal{E}$ that is used to quantize the continuous representation into the discrete space, and a decoder decoding the resulting discrete representation back to the RGB space. However, we can not directly utilize the VQ-VAE since that there is no ground truth for missing regions. Therefore, we extend the VQ-VAE to learn the discrete latent representation for the corrupted video sequence.

Each corrupted RGB input frame $x^t \in \mathbb{R}^{3\times H\times W}$ is sent to the encoder E to learn a more compact representation $E(x^t) = f_e^t \in \mathbb{R}^{d\times h\times w}$, where h and w denote the height and width, respectively, t denotes the t^{th} frame, and d denotes the dimension for each pixel in the feature maps. Instead of working in a continuous feature space, we quantize feature on each spatial-temporal location into a discrete latent space using the codebook $\mathcal{E}$. Specifically, we transfer f_e^t into the discrete feature $f_q^t \in \mathbb{R}^{d\times h\times w}$ by element-wise mapping f_e^t to its nearest prototype vector $\mathbf{e}_k$ in the codebook with

$$(f_q)_i^t = \underset{\mathbf{e}_k\in\mathcal{E}}{\arg\min}\, \|(f_e)_i^t - \mathbf{e}_k\|, \tag{3}$$

where $i \in \{1, 2, \cdots, (h \times w)\}$ indicates the spatial index. We obtain discrete representation z defined in Equation (1) by replacing the feature on each location with the corresponding index number in $\mathcal{E}$ with

$$z_i^t = k,\ s.t.\ (f_q)_i^t = \mathbf{e}_k. \tag{4}$$

Subsequently, a decoder D takes as input the quantized feature f_q^t produced by retrieving prototype vectors in $\mathcal{E}$ according to z^t, and decodes f_q^t back to input RGB space with $\hat{x}^t = D(f_q^t)$ as mapping $\mathcal{F}^{-1}$ does in Equation (1). In this way, we can represent frames as discrete index map z^t where each element corresponds to a index of prototype vectors in $\mathcal{E}$.

The discrete latent codes of video frames can be trained with the whole video sequence via the following loss function:

$$\begin{aligned}\mathcal{L}_{vq} =& \frac{1}{n}\sum \|(x-\hat{x})\odot(\mathbf{1}-m)\|^2 + \gamma_1\|\mathbf{e}_k - sg[E(x)]\| \\ &+ \gamma_2\|E(x) - sg[\mathbf{e}_k]\|,\end{aligned} \tag{5}$$

where n denotes the pixel number in the valid region and sg denotes the stop gradient operation. Here, the first term in $\mathcal{L}_{vq}$ is the reconstruction loss in the valid region. The second term enforce $\mathbf{e}_k$ more representative for current video frames, and the third term is a regularization term to prevent f_e^t from volatility, where γ_1 and γ_2 denotes the penalty weights. Since quantization operation is non-differential, the gradient of the decoder is straightly backward to the encoder as in [3].

Learning effective discrete codes for video frames requires a rich codebook to represent the latent embedding space. A heuristic method is to obtain a fixed codebook via training on a large dataset offline. However, such a codebook may not be representative for the coming videos and thus result in reconstruction of poor perceptual quality. Therefore, we propose a dynamic codebook refining scheme where for each video we maintain a codebook with rich context and video-specific information. To speed up and ease the learning of codebook, we employ a much more general codebank with 8192 prototype vectors pretrained from a large-scale dataset and customize it to a specific video sequence via Equation (5). Specifically, we adopt the model pre-trained on COCO dataset [14] and obtain a rich codebank $\mathcal{B}$, consisting of 8192 prototype vectors of 256-dimension, which is sufficient to describe the latent space for complex scenarios. We select those prototypes ever occurred in f_q to construct our video-specific codebook $\mathcal{E}$ (about $\frac{1}{16}$ of $\mathcal{B}$) , and further refine our codebook $\mathcal{E}$, encoder and decoder. Compared with $\mathcal{B}$, $\mathcal{E}$ pays much more attention on the fine-grained details within the video sequence as well as essentially reduce the difficulty of the code index prediction in the subsequent code inference stages.

3.2. Code Inference with Discrete Latent Transformer

With the code learning module, we are able to represent video frames in terms of codebook index-map $\mathbf{z}$. In this way, video inpainting can be formulated as an indices prediction task given code indices in seen region.

Index maps $z \in \mathbb{R}^{\tau\times h\times w}$ across adjacent τ frames is first flattened and each index is replaced with a specific learnable index embedding to form embedded index feature. In order to distinguish between known regions from unknown regions, we creatively fill unseen regions with a learnable completion embedding indicating that the content is missing and the network need to generate content here. Although transformer is powerful to leverage long distance dependency information, important prior inferred from spatial-temporal location is more or less ignored. To tackle this issue, we encode position information by tagging position embedding onto the index embedding. Since there is usually no ground truth provided for training in real-world scenarios, we therefore propose a self-supervised transformer framework to learn code constituent distribution in valid regions. Specifically, we randomly generate mask m_r to corrupt the valid region and thus form a pseudo unseen region. The corresponding indices in m_r are also replaced with completion flags before training and subsequently provide ground truth to guide transformer to learn code distribution among valid regions.

Let z_{emb} denote the index embedding with completion flag inserted into unseen region, p denotes the position embedding, transformer takes $emb = z_{emb} + p$ as input and learns the global correlation between code indices from the pseudo unknown region and valid region. There are multiple stack of self-attention layers of which the l^{th} layer process its input emb_l as:

$$\begin{aligned} emb_l^{'} &= MSA(LN_1(emb_l)) + emb_l, \\ emb_{l+1} &= MLP(LN_2(emb_l^{'})) + emb_l^{'} \end{aligned} \tag{6}$$

where MSA represents multiple-head self-attention operation, LN_1, LN_2 denotes layer normalization and MLP refers to multi-layer perceptron. Note that we employ Fourier position embedding [7] to preserve the spatial-temporal position structure. A prediction head P realized by one linear layer is used to produce K-way classification scores s for each spatial-temporal location followed by a softmax function layer.

$$(c_i^t)_k = e^{(s_i^t)_k} / \sum_{j=1}^{K} e^{(s_i^t)_j} \tag{7}$$

Finally we impose cross entropy loss between index classification score c and z on known region as,

$$L_{ce} = -\frac{1}{n} \sum_{i}^{hw} \sum_{t}^{\tau} \mathbb{I}_{m_i^t = 0} \sum_{k}^{K} \mathbb{I}_{k = z_i^t} \ln{(c_i^t)_k} \tag{8}$$

where $\mathbb{I}_{(\cdot)}$ is indicator function, which outputs 1 when the condition $(\cdot)$ is satisfied and 0 otherwise. The code structure information is well captured by our latent transformer after learning code distribution in valid region. Therefore, the transformer can predict the indices in unseen regions under the assumption that in a video sequence codes in unseen regions follow a similar distribution as that in valid regions. In the inference phrase, the transformer predicts the indices $\hat{z}$ in unseen region according to the following rule:

$$\hat{z}_i^t = \arg\max_k \, (c_i^t)_k \tag{9}$$

Now we have learned $\mathcal{G}$ in Equation 2, producing code inference result $\hat{z}$. In this way, the hole in the video is filled with discrete indices propagating from the valid region with the transformer. Finally, the corresponding prototype vectors in $\mathcal{E}$ queried by the predicted indices $\hat{z}^t$ are sent to decoder D to reconstruct RGB frames. With the robust discrete latent embedding, our method is able to produce fine-grained details and realistic results. Note that it is much easier to predict the indices among limited discrete codes than continuous vectors for the transformer.

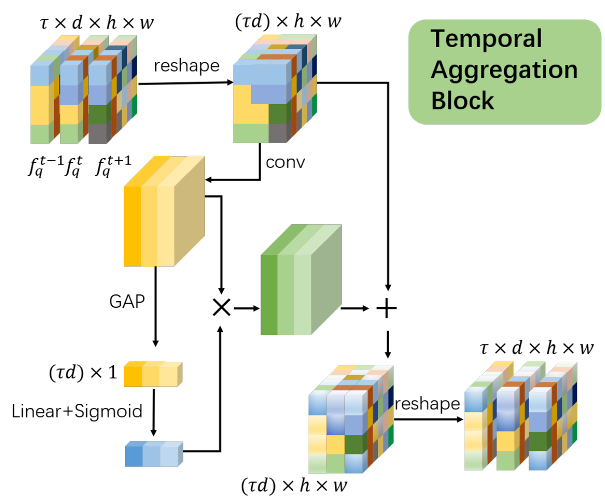


Figure 3. The schematic illustration of our Temporal Aggregation Block (TAB). Temporal information of neighboring frames is aggregated to learn residual for feature refinement.

3.3. Residual Temporal Aggregation

Our latent transformer is trained on a whole video sequence to capture global code distribution in spatial-temporal dimension. Therefore, long-range dependency is implicitly encoded in the sparse codebook and network itself. However, the short-term temporal consistency still remains untackled. Since the predicted discrete code indices may jitter between adjacent frames, reconstructed results could be lack of short-term temporal continuity. To address this issue, we design a Temporal Aggregation Block (TAB) architecture to make up for the discontinuity of discrete latent space. As illustrated in Figure 2 and Figure 3, TAB takes as input the quantized feature f_q, which are queried from codebook according to predicted code index as in Equation (3) from transformer, and outputs residual refined feature. Specifically, the quantized feature $f_q^{t-1}, f_q^t, f_q^{t+1} \in \mathbb{R}^{d \times h \times w}$ is first concatenated and sent into channel attention layer for temporal feature re-weighting, and produce residual refinement to the quantized feature to produce refined feature f_c. The residual is to aggregate temporal information across adjacent frames for feature refinement to better enhance short-term temporal consistency. A total variation loss along temporal dimension setting τ as 3 is used to train our TAB to enhance the visual effect and relieve temporal color discrepancy as following,

$$\begin{aligned} L_{tv} = &\lambda_1 (\| f_c^t - f_c^{t-1} \| + \| f_c^{t+1} - f_c^t \|) \\ &+ \lambda_2 \| f_c - f_q \|, \end{aligned} \tag{10}$$

where the first term is to enhance short-term temporal smoothness and the second term is introduced to avoid trivial solution.

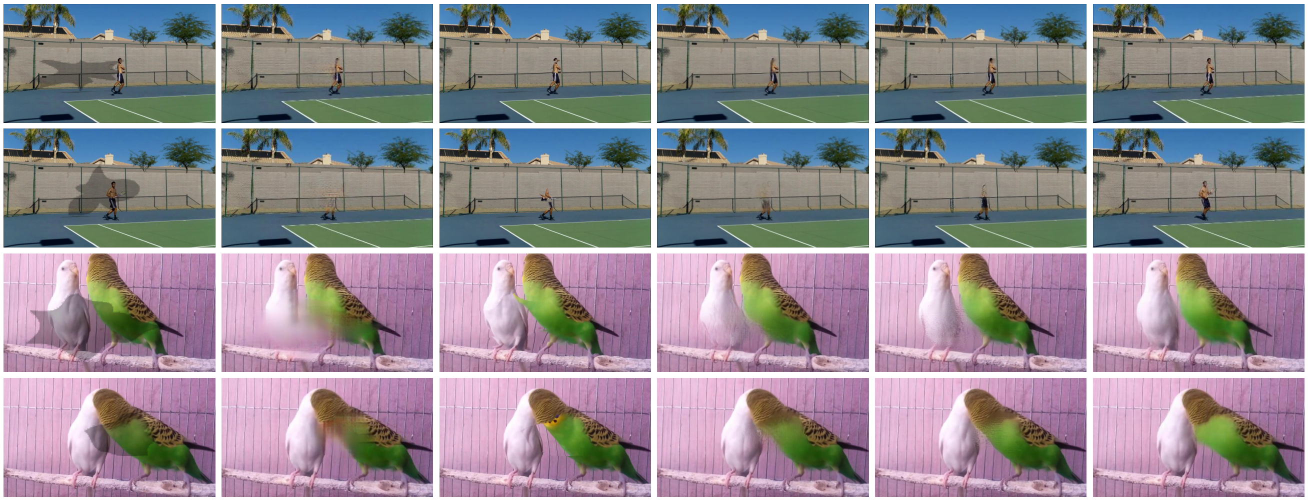


Figure 4. Qualitative comparison of different methods for video restoration. (a) Input masked frames; (b) CPNet [32]; (c) FGVC [4]; (d) STTN [34]; (e) FuseFormer [16]. Please zoom in for better visualization.

4. Experiments

In this section, we first give a necessary description for the implementation details in Section 4.1. Then we conduct comprehensive quantitative and qualitative evaluations to demonstrate the validity and superiority than other state-of-the-art approaches for video restoration and object removal in Section 4.2. We further conduct an ablation study in Section 4.3 to demonstrate the effectiveness of our designed components in our framework.

4.1. Implementation Details

Training details We train the proposed DLFormer with a two-stage learning strategy, namely, the code learning stage and the code inference stage. In the code learning stage, we fine-tune the pre-trained codebook and autoencoder using the valid regions in the target video with Equation (5) to obtain a video-specific codebook and the corresponding autoencoder. In order to limit the searching space of subsequent transformer and reduce the redundant prior knowledge, we further remove the unused prototype vectors in our video-specific codebook. The dimension for each prototype vector in the codebook is experimentally set as 256.

In the subsequent code inference stage, we fix the autoencoder and codebook obtained in the code learning stage, and only train the discrete latent transformer for inferring proper code indices in unknown regions. By randomly generating pseudo masks in seen regions and giving a completion signal, we train our transformer via Equation (8) with a self-attention mechanism. Specifically, 12 self-attention layers each with 16 heads are stacked. We use Adam [11] optimizer for the first stage and AdamW [17] for the second stage with a learning rate 1.8×10^{-5}.

Method	Youtube-VOS			DAVIS		
	PSNR ↑	SSIM ↑	VFID ↓	PSNR ↑	SSIM ↑	VFID ↓
VINet [9]	29.72	0.953	0.111	32.38	0.967	0.105
FFVI [2]	33.39	0.968	0.119	31.13	0.972	0.087
CPNet [12]	30.21	0.957	0.117	29.57	0.955	0.147
STTN [34]	33.67	0.965	0.087	33.07	0.976	0.071
FuseFormer [16]	33.26	0.968	0.089	33.45	**0.979**	0.074
DLFomer (ours)	**33.95**	**0.970**	**0.082**	**34.22**	0.977	**0.062**

Table 1. Quantitative comparison with state-of-the-art methods for video restoration on Youtube-VOS and DAVIS datasets.

Datasets and evaluation metrics Following [16, 34], we fairly evaluate our method on the two most popular datasets, namely Youtube-VOS [31] and DAVIS [24]. Youtube-VOS contains 541 video sequences for test with various dynamic scenes. We perform the video restoration task on Youtube VOS and DAVIS, and generate various types of unknown masks, including moving masks, randomly corrupted masks and object removal masks. We perform the object removal task on DAVIS dataset, which consists of 150 high-quality videos, and we select 90 videos for test following [16, 34]. For quantitative comparison, we not only employ the two widely-used metrics, structure similarity measure (SSIM) and peak signal-to-noise ratio (PSNR), to assess overall reconstruction, but also adopt the video-based Frechet inception distance (VFID) to measure the spatial-temporal consistency and perceptual quality.

4.2. Comparison with Existing Methods

Comparisons in video restoration We quantitatively compare our method in the video restoration task with existing competitive methods VINet [9], FFVI [2], CPNet [12], STTN [34] and FuseFormer [16] on Youtube-VOS and DAVIS dataset. As shown in Table 1, our method gener-

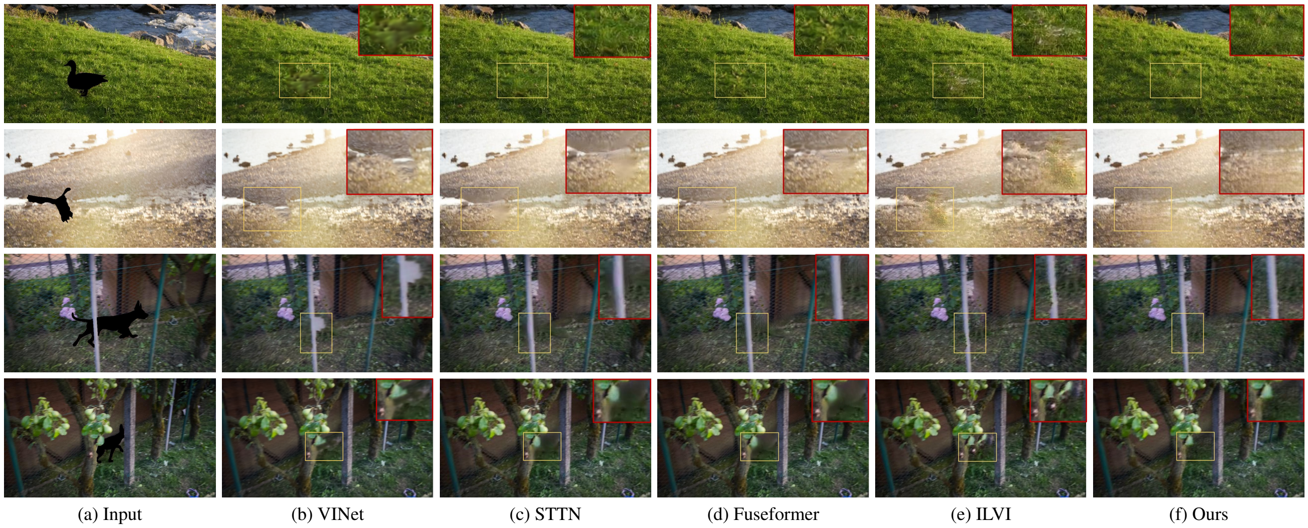


Figure 5. Qualitative comparison of different methods for object removal. (a) Input object-masked frames; (b) VINet [9]; (c) STTN [34]; (d) FuseFormer [16]; (e) ILVL [21]. Please zoom in for better visualization.

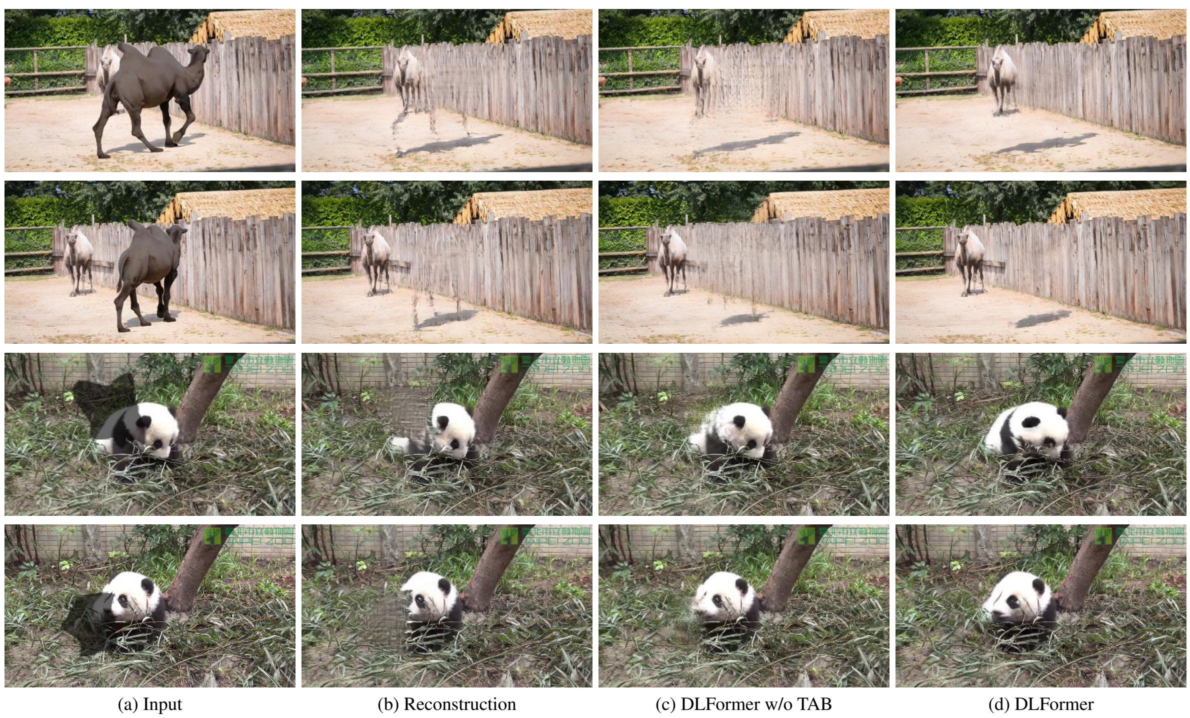


Figure 6. Visual comparison of completed results of our methods and basic networks (a) Input masked frames; (b) reconstruction results; (c) completed results without temporal aggregation block; (d) results of our full pipeline. Please zoom in for better visualization.

ates results with almost the best performance in terms of all the three indicators. Considering our improvement over STTN [34] and FuseFormer [16], especially around the regions with high-frequency textures, is difficult to measure with these indicators, we further present more qualitative results in Figure 4. The results in column (b) and (c) are produced by alignment-based methods and result in misplacement and blurry artifacts. Transformer-based methods, STTN [34] and FuseFormer [16], in (d) and (e) give better results but still fail to generate visually plausible content, especially for the sportsman in the first two cases. As shown in (f), our method recovers fine-grained details and consistent structure in the body of sportsman and the feather of birds which convincingly demonstrates the discrete code distribution is fully learned and properly propagated through the proposed discrete latent transformer.

	PSNR ↑	SSIM ↑	VFID ↓
Reconstruction	33.07	0.959	0.124
DLFormer w/o TAB	33.82	0.968	0.086
DLFomer (ours)	**33.95**	**0.970**	**0.082**

Table 2. Ablation experimental results on Youtube-VOS dataset.

4.3. Ablation Analysis

Comparisons in object removal For object removal task, we present qualitative results in Figure 5. The results from column (b) to (d) give blurry texture and obvious spatial artifacts around high-frequency areas, such as the grass, sand beach and leaves. Although ILVI [21] outputs sharper results, the spatial-temporal distortion still remains around the railing and leaf regions. Comparatively, our method generates more consistent results both spatially and temporally, thanks to our novel framework as well as the specially designed residual temporal aggregation block for relieving visual jitters.

User study We perform a user study to compare our results on both video restoration and object removal tasks with state-of-the-art methods FuseFormer [16], STTN [34] and VINet [9]. 32 volunteers are invited to rate the visual quality (from 1 to 10, the higher the better) for both image frames and videos randomly sampled from Youtube-VOS and DAVIS for evaluating the inpainted details and spatial-temporal consistency, respectively. The results of the user study are presented in Figure 6. Our method achieves the highest scores on both frame and video quality, indicating our method generates more temporal-spatial consistent contents in unknown regions.

Effectiveness of discrete video representation The foundation of our work is the obtained discrete codebook and the corresponding autoencoder can represent the target video sufficiently. To measure the effectiveness of this representation, we directly deliver the quantized feature from encoder to decoder, without the code inference stage, to reconstruct the target video. As shown in Figure 6 (b), the known regions are vividly reconstructed, indicating our codebook captures the discriminative part of the target video and the discrete latent space is sufficient to represent it. In unknown regions, not surprisingly, the results are filled with visible artifacts due to the lacking of critical code inference.

Effectiveness of discrete latent transformer DLFormer w/o TAB refers to the results generated with a completed code map after code inference stage but without the temporal aggregation block. As shown in Figure 6 (c), the unknown region is properly recovered with overall reasonable content, such as the plank behind the masked camel and the part of the giant panda, indicating that discrete latent transformer effectively learned code distribution from the known region and properly predicts reasonable discrete code. Although the quantitative results in Table 2 show that DLFormer w/o TAB achieves much better performance compared with the aforementioned reconstruction results, there is still flicking artifacts across neighboring frames in terms of short-term temporal consistency.

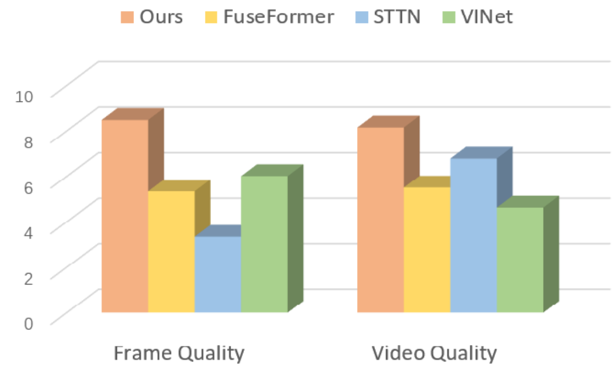


Figure 7. User study results. 32 volunteers are invited to rate the completed video frames in terms of inpainted details and entire video sequence in terms of spatial-temporal consistency. Our method produces results of high image quality as well as pleasing spatial-temporal consistency compared with existing methods.

Effectiveness of TAB After the code inference stage, the resulting index map can be mapped back to discrete codes with the codebook. Such discrete codes are further sent to the subsequent TAB block to refine the short-term temporal information. In addition, a total variation loss is imposed on the refined feature. As presented in Figure 6 (d), results with TAB block are more visually pleasing and consistent across neighboring frames and quantitative results in Table 2 demonstrate the same consequence.

5. Conclusion

We novelly formulate the video inpainting task as a discrete code inference problem in the latent discrete space which is spanned by a context-rich and efficient codebook. We learn a compact video-specific codebook and infer the missing code indices via a discrete latent transformer. While training this transformer in a self-supervision manner, code distribution in known regions can be propagated to unknown regions. A temporal aggregation block across adjacent frames is further proposed to relieve temporal visual jitters caused by the discrete prediction. Our method generates visually-plausible and spatial-temporal coherent content with fine-grained details in unknown regions and outperforms the state-of-the-art methods significantly.

Acknowledgements: The work is supported by Key-Area Research and Development Program of Guangdong Province, China (2020B010165004,2020B010166003); NSFC (61772206, U1611461, 61472145); Guang-Dong Basic and Applied Basic Research Foundation (No.2021A1515110598).

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 2

[2] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9066–9075, 2019. 2, 6

[3] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021. 3, 4

[4] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *European Conference on Computer Vision*, pages 713–729. Springer, 2020. 2, 6

[5] Yuan-Ting Hu, Heng Wang, Nicolas Ballas, Kristen Grauman, and Alexander G Schwing. Proposal-based video completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 38–54. Springer, 2020. 2

[6] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Temporally coherent completion of dynamic video. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016. 2

[7] Andrew Jaegle, Felix Gimeno, Andrew Brock, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver: General percetion with iterative attention. *arXiv preprint arXiv:2103.03206*, 2021. 5

[8] Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. Fast decoding in sequence models using discrete latent variables. In *International Conference on Machine Learning*, pages 2390–2399. PMLR, 2018. 3

[9] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019. 1, 2, 6, 7, 8

[10] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Recurrent temporal aggregation framework for deep video inpainting. *IEEE transactions on pattern analysis and machine intelligence*, 42(5):1038–1052, 2019. 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[12] Sungho Lee, Seoung Wug Oh, DaeYeun Won, and Seon Joo Kim. Copy-and-paste networks for deep video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4413–4421, 2019. 1, 2, 6

[13] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[15] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint arXiv:2104.06637*, 2021. 1, 2

[16] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14040–14049, 2021. 1, 2, 6, 7, 8

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[18] Yasuyuki Matsushita, Eyal Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on pattern analysis and Machine Intelligence*, 28(7):1150–1163, 2006. 1

[19] Seoung Wug Oh, Sungho Lee, Joon-Young Lee, and Seon Joo Kim. Onion-peel networks for deep video completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4403–4412, 2019. 1, 2

[20] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 2, 4

[21] Hao Ouyang, Tengfei Wang, and Qifeng Chen. Internal video inpainting by implicit long-range propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14579–14588, 2021. 7, 8

[22] Kedar A Patwardhan, Guillermo Sapiro, and Marcelo Bertalmio. Video inpainting of occluding and occluded objects. In *IEEE International Conference on Image Processing 2005*, volume 2, pages II–69. IEEE, 2005. 2

[23] Kedar A Patwardhan, Guillermo Sapiro, and Marcelo Bertalmío. Video inpainting under constrained camera motion. *IEEE Transactions on Image Processing*, 16(2):545–553, 2007. 2

[24] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 2, 6

[25] Ruslan Rakhimov, Denis Volkhonskiy, Alexey Artemov, Denis Zorin, and Evgeny Burnaev. Latent video transformer. *arXiv preprint arXiv:2006.10704*, 2020. 3

[26] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, pages 14866–14876, 2019. 2

[27] Michael Strobel, Julia Diebold, and Daniel Cremers. Flow and color inpainting for video completion. In *German Conference on Pattern Recognition*, pages 293–304. Springer, 2014. 2

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[29] Chuan Wang, Haibin Huang, Xiaoguang Han, and Jue Wang. Video inpainting by jointly learning temporal structure and spatial details. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5232–5239, 2019. 2

[30] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Transactions on pattern analysis and machine intelligence*, 29(3):463–476, 2007. 2

[31] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 2, 6

[32] Rui Xu, Xiaoxiao Li, Bolei Zhou, and Chen Change Loy. Deep flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2019. 1, 2, 6

[33] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. 3

[34] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020. 1, 2, 6, 7, 8

[35] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2720–2729, 2019. 2

[36] Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, and Changyou Chen. Feature quantization improves gan training. *arXiv preprint arXiv:2004.02088*, 2020. 3

[37] Xueyan Zou, Linjie Yang, Ding Liu, and Yong Jae Lee. Progressive temporal feature alignment network for video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16448–16457, 2021. 1, 2