

# Solving Masked Jigsaw Puzzles with Diffusion Vision Transformers \*

Jinyang Liu  
 Northeastern University  
 Boston MA

liu.jinyan@northeastern.edu

Wondmgezahu Teshome  
 Northeastern University  
 Boston MA

teshome.w@northeastern.edu

Sandesh Ghimire  
 Qualcomm  
 San Diego CA

drsandesghimire@gmail.com

Mario Szaier  
 Northeastern University  
 Boston MA

mszaier@coe.northeastern.edu

Octavia Camps  
 Northeastern University  
 Boston MA

camps@coe.northeastern.edu

## Abstract

Solving image and video jigsaw puzzles poses the challenging task of rearranging image fragments or video frames from unordered sequences to restore meaningful images and video sequences. Existing approaches often hinge on discriminative models tasked with predicting either the absolute positions of puzzle elements or the permutation actions applied to the original data. Unfortunately, these methods face limitations in effectively solving puzzles with a large number of elements. In this paper, we propose JPDVT, an innovative approach that harnesses diffusion transformers to address this challenge. Specifically, we generate positional information for image patches or video frames, conditioned on their underlying visual content. This information is then employed to accurately assemble the puzzle pieces in their correct positions, even in scenarios involving missing pieces. Our method achieves state-of-the-art performance on several datasets.

## 1. Introduction

Image and video jigsaw puzzle solving involves the challenging task of reassembling image fragments or reshuffling video frames from unordered sequences into meaningful images and video sequences. This problem holds significant importance across various domains, ranging from image editing, biology, archaeology, document or photograph restoration, to photo sequencing [3].

\*This work was supported in part by NSF grant 2038493, ONR grant N00014-21-1-2431, NIH grant R01CA240771 from NCI, and U.S. Department of Homeland Security grant 22STESE00001-03-02. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

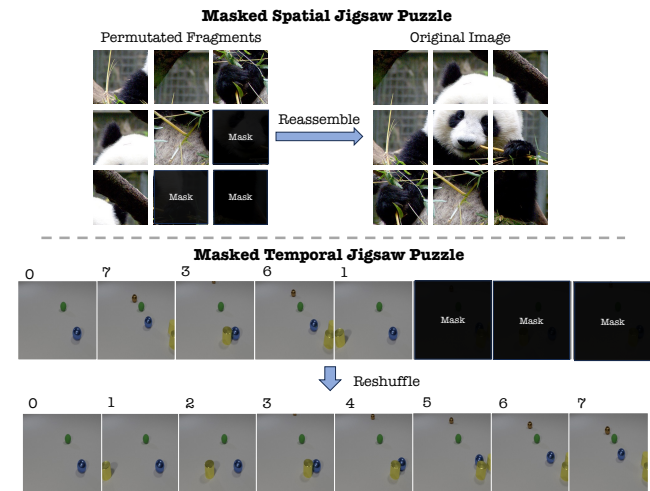


Figure 1. Top: given unordered image fragments, some of them masked, we want to reconstruct the original image. Bottom: given shuffled video frames, some of them masked, we want to reconstruct the original video.

Past methods for solving jigsaw puzzles have commonly relied on discriminative networks, particularly in classifying the positions of image patches or video frames [11, 18]. Typically, these approaches involve feature extraction using a Convolutional Neural Network (CNN) backbone, followed by pairwise feature combinations before classification. The classifier then operates either by classifying permutations applied to the raw data or by directly predicting the positions of unordered elements. The primary emphasis in these methods is on training the feature backbone network to extract highly distinguishable positional features during the jigsaw puzzle-solving process.

However, these approaches face challenges when dealing with a large number of elements and when handling a variable number of elements. Furthermore, they often pri-

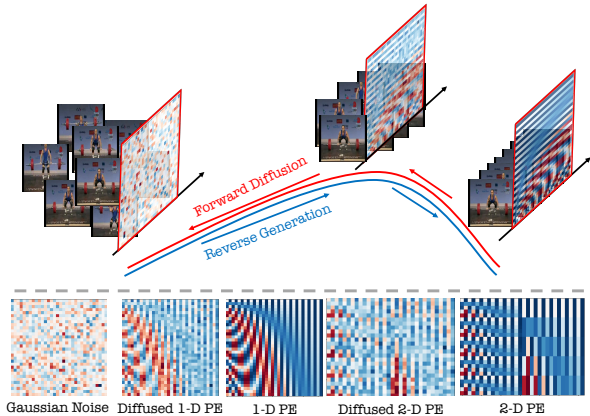


Figure 2. Top. Each piece has a positional encoding and an embedding of its visual content. The forward diffusion step (highlighted in the red box), gradually adds noise to the positional encodings. The reverse generation reconstructs the positional encodings, conditioned by the provided visual content. Bottom. Samples of 1D and 2D positional encodings, for video frames and image tiles, both in their original form and with added noise.

oritize learning order at the expense of capturing the entire structure of the data, making them less suitable for solving jigsaw puzzles when data elements are missing.

This paper introduces **Jigsaw Puzzles with Diffusion Vision Transformers (JPDVT)**, a novel approach to tackle these challenges through the utilization of a conditional diffusion denoising probabilistic model (CDDPM). The CD-DPM is designed to learn the positions of all the elements and generate the missing pieces, guided by the visual context of the available pieces, as illustrated in Figs. 1 and 2. Rigorous testing on diverse image and video datasets validates its superior performance compared to state-of-the-art models, underscoring the effectiveness of our proposed approach. The main contributions of this work are:

1. A simple unified design to solve both image and temporal puzzles with large number of pieces. The proposed model can handle missing data, finding the correct positions of the available pieces and generating plausible reconstructions for the missing ones.
2. State-of-the-art performance on several image and video datasets. Our code and pre-trained models are available at our GitHub project page.<sup>1</sup>

## 2. Related Work

Solving image and video jigsaw puzzles have the common objective of learning (spatial or temporal) positional information from unordered elements.

**Spatial Jigsaw Puzzles:** Most existing approaches revolve around the reassembly of square puzzle pieces, with a focus on recreating the original image using non-

overlapping components. Existing state-of-the-art jigsaw puzzle solvers can be broadly categorized into two streams.

The first stream, represented by algorithms like [18–20, 24, 31], leverages feature extraction or image understanding networks for encoding visual content. Subsequently, these methods engage in fragment reassembly, demonstrating effectiveness in handling fragments with large eroded gaps.

The second stream, as exemplified by [3, 14], initiates the process with fragment generation and subsequent reassembly. A neural network serves as a discriminator to verify the correct placement of fragments, yet these methods are generally limited to addressing small eroded gaps.

While most of the prior works use CNNs, owing to their downsampling capabilities that facilitate classification, our paper advocates for using transformer models due to their equivariance to shuffling transformations.

In terms of reassembly strategies, traditional approaches include brute force [20] or greedy methods [3]. However, these methods face scalability challenges with large-scale puzzles due to their exponentially growing complexity. Some studies propose using classifiers to predict permutations [18] or fragment positions [31]. Other sophisticated approaches involve Dijkstra’s algorithm with graph-cut [19], while a recent work [24] employs reinforcement learning as the reassembly strategy. Additionally, [25] utilized a genetic algorithm to tackle the problem. In contrast, our paper introduces the novel use of diffusion generative models for reassembly, leveraging positional information learned from visual patterns encoded by a transformer model. Moreover, our approach extends beyond conventional practices, demonstrating the capability to solve jigsaw puzzles even in the presence of missing elements.

**Temporal Puzzles:** Several recent works [11, 13, 15, 30, 33] propose using sorting temporal sequences as a pretext tasks for self-supervised learning for applications such as action recognition and video anomaly detection. They employ CNNs as video encoders to predict the order of shuffled videos directly. However, these methods, while effective for specific applications, are not explicitly designed to address the broader challenges of solving temporal puzzles, particularly on larger temporal scales.

In contrast, our work draws inspiration from the paradigm set by [1, 4, 6, 16], where the emphasis lies on explicitly reshuffling temporal sequences akin to a photo sequencing problem. Unlike method [6], which learned permutation matrices by solving optimization problems enforcing low-rank structures on shuffled temporal sequences, our approach transcends the limitations of optimization-based techniques. These methods face constraints on sequence length and lose the ability to capture spatial coherence due to significant dimensionality reduction.

This work moves beyond solving purely temporal jigsaw puzzles and tackles a more intricate problem—reshuffling

<sup>1</sup><https://github.com/JinyangMarkLiu/JPDVT>

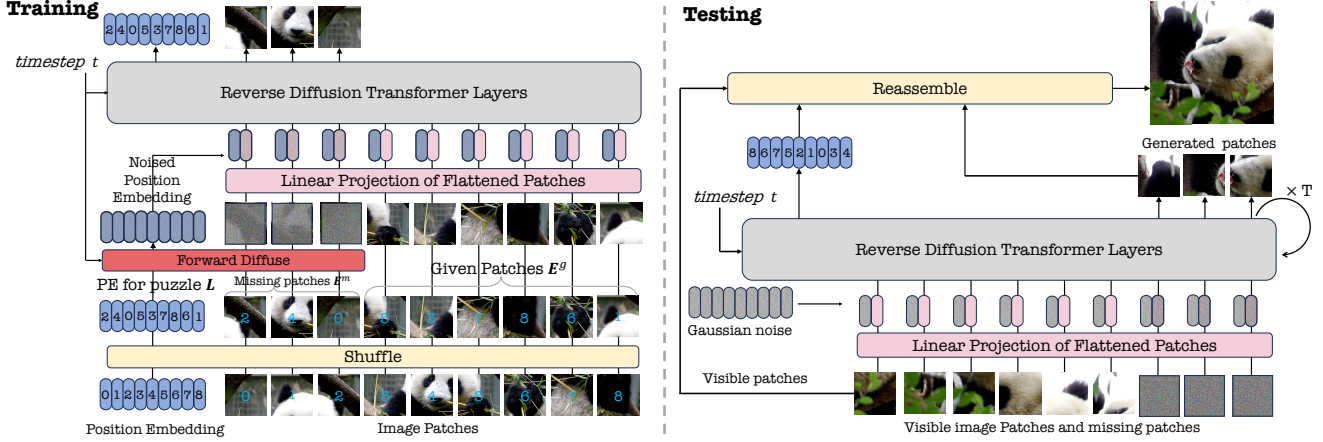


Figure 3. Left: Architecture employed to train the proposed diffusion transformer model. Image patches, accompanied by their corresponding position encoding tokens, undergo permutation in a consistent manner. During forward diffusion, noise is introduced to all positional encoding tokens and a subset of embedded patch tokens. The transformer is trained to execute a reverse diffusion process, mitigating the introduced noise. Right: Inference process. During inference, the observed patches are padded with Gaussian noise to fill the entire image. Linearly embedded patch tokens are then concatenated with positional encoding tokens initialized with Gaussian noise. The transformer model is designed to reconstruct the visual content of the missing patches and to determine the positions for all patches, simultaneously.

temporal sequences with temporal masks. Specifically, it aims to reconstruct a shuffled temporal sequence or video with missing frames, posing a novel and challenging extension to the temporal puzzle-solving paradigm.

### 3. Our Approach

Here, we present JPDVT, which uses a novel perspective on representing video and image data as non-ordered sets, to solve masked jigsaw puzzles. This representation enables JPDVT to harness the inherent properties of vision transformer architectures and the capabilities of conditional generative diffusion models to solve puzzles with missing pieces, as depicted in Fig. 3.

#### 3.1. Masked Jigsaw Transformer Diffusion

Consider an image or video, denoted as  $\mathbf{X}$ , partitioned into an unordered puzzle comprising individual pieces  $\mathbf{x}_1, \dots, \mathbf{x}_N$  such that  $\mathbf{x}_i \cap \mathbf{x}_j = \phi$  for all  $i$  and  $j$ , and  $\mathbf{X} = \cup_{i=1}^N \mathbf{x}_i$ . Our objective is to reconstruct  $\mathbf{X}$  by establishing a mapping  $f(\mathbf{x}_i) = l_i$  that assigns the correct position to each piece. Specifically, we seek a solution where  $\mathbf{X}$  is the union<sup>2</sup> of the pieces placed at their designated locations, expressed as  $\mathbf{X} = \cup_{i=1}^N (\mathbf{x}_i @ l_i)$ , where  $\mathbf{x}_i @ l_i$  signifies that piece  $\mathbf{x}_i$  should be positioned at location  $l_i$ .

To achieve this goal, we propose representing the solved puzzle of  $\mathbf{X}$  as a set of pairs denoted by  $\mathbf{P} = \{(\mathbf{e}_1, \gamma(l_1)), \dots, (\mathbf{e}_N, \gamma(l_N))\}$ . In this representation,  $\mathbf{e}_i$  is an embedding of the visual content of piece  $\mathbf{x}_i$ , and  $\gamma(l_i)$

<sup>2</sup>In the cases with missing pieces or with eroded pieces, we will allow  $\mathbf{X} \supseteq \cup_{i=1}^M \mathbf{x}_i @ l_i$ .

is a vector containing the positional encoding of its location  $l_i$ , as illustrated in Fig. 2. This representation is similar to the ones used by masked autoencoders (MAE) to learn to inpaint missing data [9]. However, a notable distinction lies in the fact that masked autoencoders possess knowledge of the locations of the given pieces, allowing them to directly work with the pairs  $(\mathbf{e}_i, \gamma(l_i))$ . In our scenario, where the pieces have been shuffled, a separate treatment of the visual content  $\mathbf{e}_i$  and its corresponding location  $\gamma(l_i)$  is necessary.

**Puzzles without Missing Pieces.** Given a complete set of puzzle pieces embeddings,  $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N\}$ , let  $\mathbf{L} = (\gamma(l_1), \gamma(l_2), \dots, \gamma(l_N))^T$  be a matrix with the sought positional encodings of the given pieces. Then, we formulate solving the jigsaw puzzle as a conditional generation problem, where the goal is to learn the joint probability of the locations  $\mathbf{L}$ , given their unsorted visual content  $\mathbf{E}$ :  $\mathbf{L} | \mathbf{E} \sim q(\mathbf{L} | \mathbf{E})$ . We propose to learn this probability by using a CDDPM as follows.

For the forward process, we diffuse the positional encodings, conditioned on the set of unsorted visual contents:

$$q(\mathbf{L}_t | \mathbf{L}_{t-1}, \mathbf{E}) = \mathcal{N}(\mathbf{L}_t | \mathbf{E}; \sqrt{1 - \beta_t} \mathbf{L}_{t-1} | \mathbf{E}, \beta_t \mathbf{I})$$

where noise is gradually added to the positional encodings of the patches (See Fig. 2), and  $\beta_t$  schedules the amount of noise added at each step  $t$ . The distribution of  $\mathbf{L}_t$  given  $\mathbf{L}_0$  can be explicitly expressed by

$$q(\mathbf{L}_t | \mathbf{L}_0, \mathbf{E}) = \mathcal{N}(\mathbf{L}_t | \mathbf{E}; \sqrt{\alpha_t} \mathbf{L}_0 | \mathbf{E}, (1 - \alpha_t) \mathbf{I})$$

where  $\alpha_t = \prod_{s=1}^t (1 - \beta_s)$ .

The reverse diffusion denoising process is given by:

$$p_\theta(\mathbf{L}_{t-1} | \mathbf{L}_t, \mathbf{E}) := \mathcal{N}(\mathbf{L}_{t-1} | t, \mathbf{E}; \mu_\theta(\mathbf{L}_t, t, \mathbf{E}), \sigma_\theta(\mathbf{L}_t, t, \mathbf{E}) \mathbf{I})$$

The conditional diffusion model learns a network  $\epsilon_\theta$  to predict the noise added to the noisy input  $\mathbf{L}_t$  with the loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{L}_0 \sim q(\mathbf{L}_0), \epsilon \sim \mathcal{N}(0, \mathbf{I}), t, \mathbf{E}} [\|\epsilon - \epsilon_\theta(\mathbf{L}_t, t, \mathbf{E})\|_2^2]$$

where  $\mathbf{L}_t = \sqrt{\alpha_t} \mathbf{L}_0 + \sqrt{(1 - \alpha_t)} \epsilon$  for  $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ .

**Puzzles with Missing Pieces.** For jigsaw puzzles with missing pieces,  $\mathbf{E} = \mathbf{E}^g \cup \mathbf{E}^m$ , where  $\mathbf{E}^g$  and  $\mathbf{E}^m$  are the given and missing pieces, respectively. In this case, we generate the positional information for both the given pieces and the missing ones. For the missing pieces, it is possible to fill them up using an inpainting algorithm [28] after having their location, or alternatively, to generate the embeddings of the missing visual content simultaneously with the locations. In the later case, the conditional forward process gradually adds noise to both, the positional encodings and the visual content embeddings of the missing pieces, conditioned on the visual content of the given pieces using  $q(\mathbf{L}_t, \mathbf{E}_t^m | \mathbf{L}_0, \mathbf{E}_0^m, \mathbf{E}^g)$  and the corresponding reverse diffusion process:  $p_\theta(\mathbf{L}_{t-1}, \mathbf{E}_{t-1}^m | \mathbf{L}_t, \mathbf{E}_t^m, \mathbf{E}^g)$ .

### 3.2. Implementation Details

In this section, we give implementation details of our models, which are rooted in the official codebase of diffusion models from [17]. Code will be provided.

**Solving image jigsaw puzzles.** We implemented our diffusion transformers by modifying ViTs based on [2], following the approach in [21] for incorporating timesteps. Adaptive normalization layers are introduced before and after the multi-head attention and MLP modules to accommodate timesteps. To avoid the model reducing to a mere classification task, we anchor the positional embedding of the initial patch and represent subsequent patches using relative positional embeddings. This approach enhances the model’s ability to capture spatial dependencies and ensures resilience against overreliance on absolute positional information. We utilized a linear noise schedule with  $T = 1000$  diffusion steps.

**Solving temporal jigsaw puzzles.** For low-resolution videos, with missing frames, we applied diffusion directly on the video pixels. For high-resolution videos, we followed [22] and first compressed each of the frames into a latent space, and subsequently applied diffusion process on these latent variables. For the reverse diffusion process, we modified the video U-Net based on [23]. In our implementation, we only use the 2D spatial convolutional layers without the temporal convolutional layer, and in the 2D+1D attention layers, we removed the temporal positional embedding. We used a linear noise schedule, and we fixed diffusion steps to  $T = 1000$ .

**Positional Encoding.** We use the positional encoding:

$$\gamma_{1D}(l_n) = \begin{cases} \gamma(l_n, 2i) = \sin(l_n/1000^{(2i/16)}) \\ \gamma(l_n, 2i+1) = \cos(l_n/1000^{(2i/16)}) \end{cases}$$

For temporal puzzles we use a 1-D positional encoding with a dimension of 16 to represent each video frame. To align with the input feature map’s shape, the positional encoding undergoes an initial transformation through an MLP, ensuring it matches the size of the input feature map. Subsequently, the positional encoding is concatenated with the feature maps. For image puzzles, we employ a 2-D positional encoding with 32 dimensions to encode the position of an image patch, with the first half representing the x-coordinate and the second half representing the y-coordinate:  $\gamma_{2D}(l_x, l_y) = [\gamma_{1D}(l_x), \gamma_{1D}(l_y)]$ . This positional encoding is incorporated into image tokens following an MLP layer.

## 4. Experiments

Here, we describe the experiments we ran to validate the effectiveness of the proposed approach.

### 4.1. Datasets

We evaluated our approach on five datasets, consisting of both images and videos with different levels of resolution and complexity:

**ImageNet-1k** [5] contains 1,281,167 training images, and 50,000 validation images categorized into 1000 classes.

**JPwLEG-3** [24] contains 12,000 images collected from MET [35] dataset. Each image is segmented into 9 fragments of dimensions  $96 \times 96$ , with gaps of 48 pixels between them. 9,000 images were allocated for training, 2,000 for testing, and an additional 1,000 for validation purposes.

**MovingMNIST** [27] comprises 10,000 video sequences, each containing 20 frames. Two digits move independently within each video, which has low resolution of  $64 \times 64$  pixels. We used 9,000 videos for training and tested our approach on the remaining 1,000 videos.

**CLEVRER** [34] is a diagnostic video dataset with 20,000 videos that simulate collisions among objects of different colors and shapes with  $480 \times 320$  pixels. The dataset is split into 10,000 videos for training, 5,000 for validation, and 5,000 for testing.

**QST**(Quick Sky Time) [36] contains 1,167 video clips extracted from 216 time-lapse 4K videos collected from YouTube, totaling 285,446 frames. The resolution of each frame exceeds  $1,024 \times 1,024$ . We downsample the frames to 256, and train on 1,000 videos, test on 167 videos.

**UCF101** [26] is an action recognition dataset, consisting of 13,320 video clips representing 101 different human actions. Each frame has a resolution of  $320 \times 240$  pixels, with 9,537 videos for training and 3,783 videos for testing.

### 4.2. Metrics

**Image Puzzle Solving Accuracy:** In assessing image jigsaw puzzle solutions, we gauge solving accuracy both at



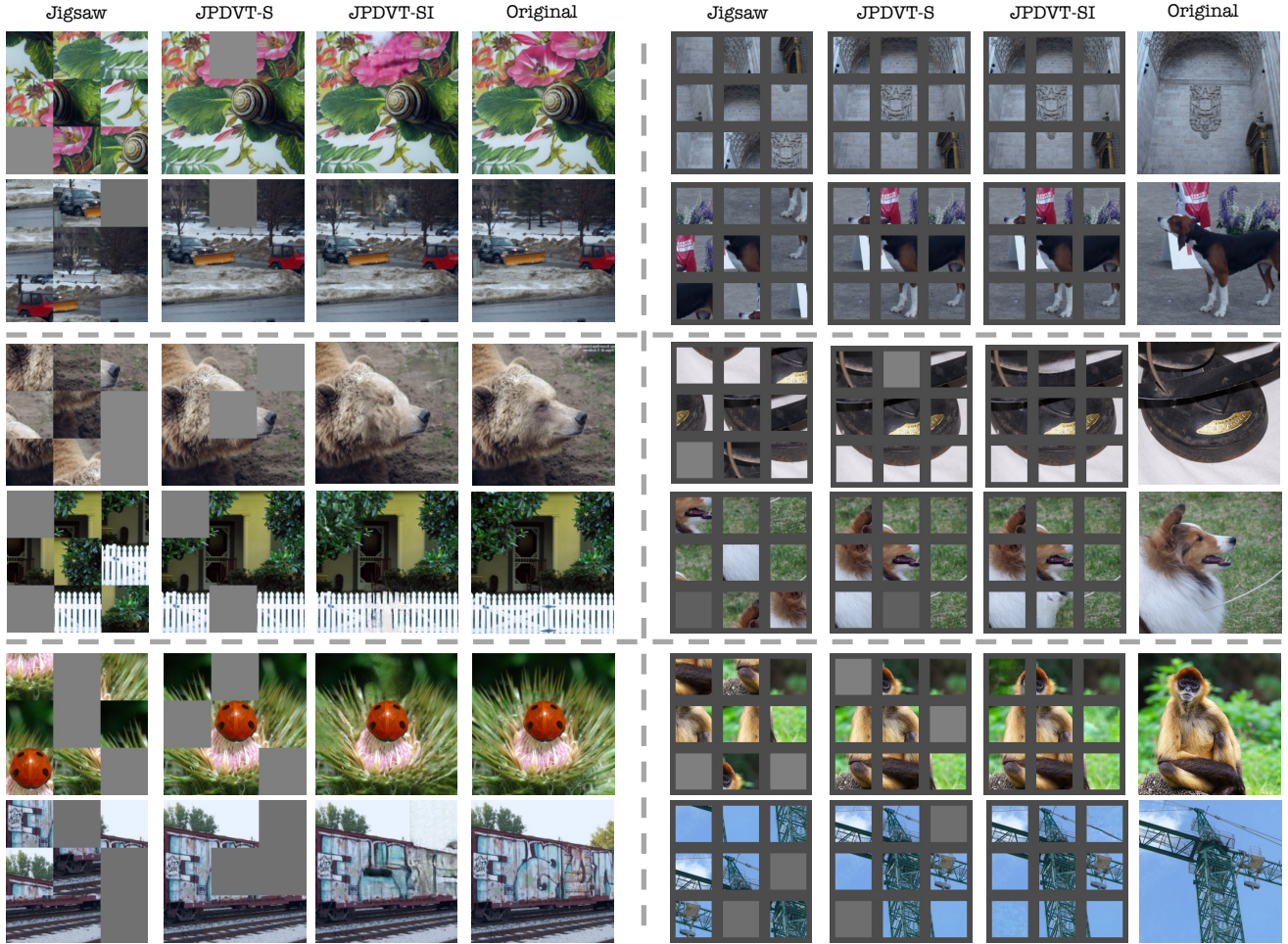


Figure 4. Qualitative results for puzzles for Imagenet-1K dataset with 1, 2, and 3 missing pieces. Left: without cropping. Right: with cropping. First column shows the input puzzle, second column shows the sorted result (denoted as JPDVT-S), third column shows the sorted and inpainted result (denoted as JPDVT-SI), and last column shows the ground truth images.

the puzzle and piece levels. The puzzle-wise accuracy metric denotes the percentage of flawlessly reassembled image puzzles—essentially, the fraction of puzzles with every single piece correctly positioned, totaling 100%. On the other hand, the piece-wise accuracy metric quantifies the ratio of correctly placed pieces to the overall number of pieces in the dataset. This comprehensive evaluation framework provides a nuanced understanding of the reconstruction quality, capturing both holistic puzzle-level performance and the precision of individual pieces within the puzzle assembly.

**Video Puzzle Solving Accuracy** Following [6], we measure video jigsaw puzzle solving accuracy with the average *Normalized Kendall distance*, based on 1,000 testing samples from each dataset. The *Kendall distance*, also called *bubble-sort distance*, scores the number of pairwise disagreements between the candidate sorting and the ground truth, which is defined as the number of elements in the set

constructed from two sequences  $\sigma_1$  and  $\sigma_2$  as follows:

$$\mathcal{G}(\sigma_1, \sigma_2) = \{(i, j) | \sigma_1(i) < \sigma_2(j), \sigma_2(i) > \sigma_1(j)\}$$

where  $\sigma_1(i)$  and  $\sigma_2(i)$  are the rankings of the element  $i$  in the  $\sigma_1$  and  $\sigma_2$  sequences. The normalized distance is the Kendall distance, normalized by the biggest possible distance. The lower the distance is, the more similar the two sequences are, with the worst possible score being 1.

### 4.3. Image Jigsaw Puzzles Experiments

We conducted experiments on solving image jigsaw puzzles using the *ImageNet-1k* dataset, *JPwLEG-3* [24], and the dataset used in [3]. Our models were trained separately with varying hyperparameters tailored to accommodate different numbers of puzzle pieces. To assess JPDVT’s performance, we conducted comparative evaluations against state-of-the-art methods [3, 19, 24, 25, 31].

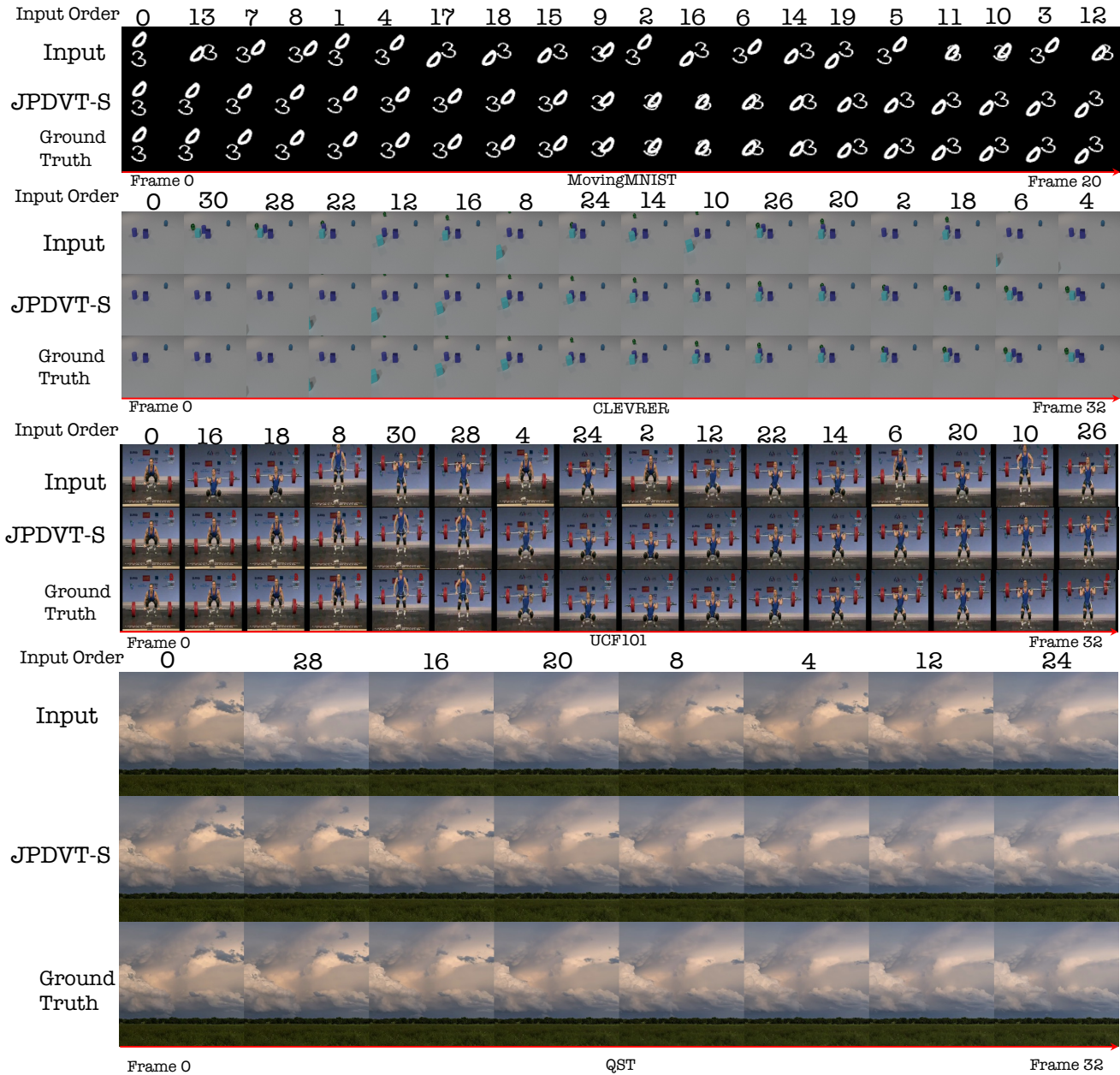


Figure 5. Video reshuffle results are showcased on various datasets, including 20 frames on *MovingMNIST*, 32 frames on *CLEVRER*, 32 frames on *UCF101*, and 32 frames on *QST*. Due to space limitations, only a subset of frames is presented in this illustration.

The experiments on *ImageNet-1k* follow the data pre-processing procedure from [31]. During training, each image underwent an initial resizing to  $255 \times 255$  before being divided into nine patches, each of size  $85 \times 85$ . Subsequently, every patch was randomly cropped to  $64 \times 64$ . A total of  $9! = 362,880$  permutations were applied to these patches, and the models were trained to generate permutation indices based on the image patches. Data augmentation solely involved horizontal flipping. During testing, in con-

trast to random cropping, we used the cropped patch from the center. The experiments on *JPwLEG-3* follow the experiment setting in [24]. In the experiments with a large number of pieces, we follow the experiment setting with the largest number of pieces in [3].

Fig. 4 presents qualitative results for puzzles with 1, 2, and 3 missing pieces, showcasing flawless reshuffling. Quantitative results for puzzles without missing pieces are provided in tables 1, 3 and 4. JPDVT achieved a 68.7%



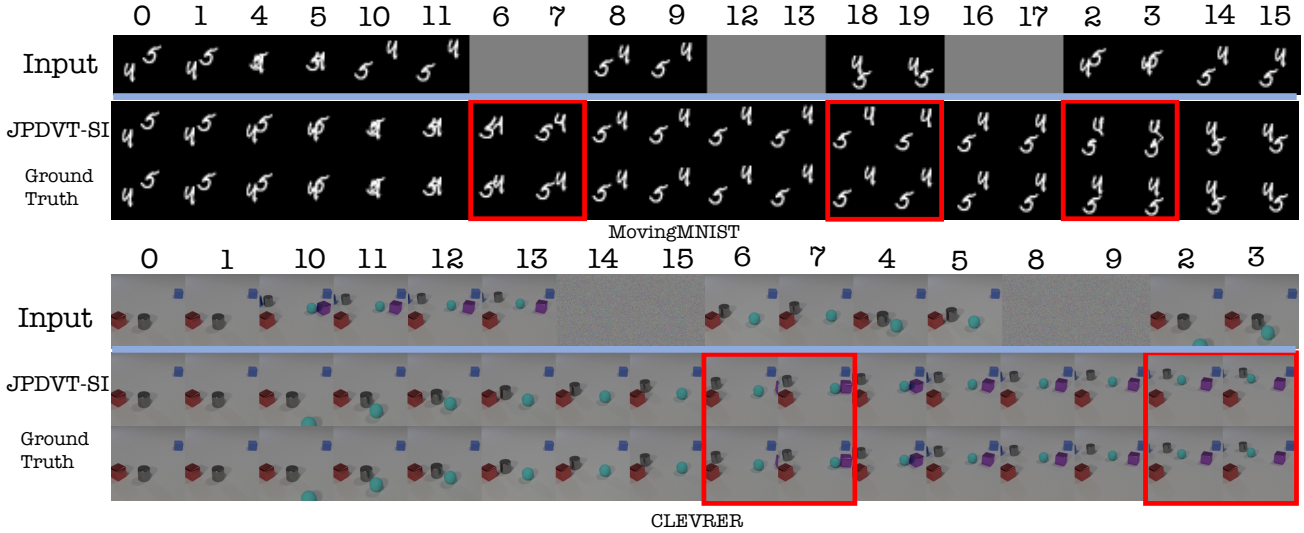


Figure 6. Video reshuffle results with masks: Our diffusion model adeptly captures and reconstructs crucial latent components essential for the sequence reordering process.

Table 1. Results of  $3 \times 3$  image puzzle solving on *ImageNet-1k* val split. Puzzle-level shows the percentages of flawlessly re-assembled puzzles. Fragment-level shows the fraction of correctly placed pieces of all the pieces in the dataset.

Method	Puzzle-level	Piece-level
SAJ[31]	47.3%	NA
Deepzzle [19]	48%	78%
JPDVT(Ours)	<b>68.7%</b>	<b>83.3%</b>

Table 2. Results on  $3 \times 3$  puzzle recognition on *ImageNet-1K* val split on our proposed model on different number of missing fragments with and without eroded gaps.

% Missing	With gap		Without gap	
	puzzle	piece	puzzle	piece
0	<b>68.7%</b>	<b>83.3%</b>	<b>99.4%</b>	<b>99.6%</b>
11.1	41.5%	72.0%	96.9%	98.8%
22.2	21.4%	61.8%	82.8%	95.0%
33.3	14.9 %	54.1 %	56.1%	76.4%

success rate on *imagenet-1k* dataset and 71.3% on *JPwLEG-3* dataset for puzzles without missing pieces—an improvement of 20.7% and 11.6%, respectively, over prior state-of-the-art performances at the puzzle level. The results in table 3 show the capability of our model to handle a large number of puzzle pieces. Unlike previous methods, our approach handles puzzles with missing fragments, as demonstrated in Table 2. The performance gradually decreases with an increasing number of random missing pieces, indicating greater difficulty with eroded pieces compared to those without erosion. This suggests the architecture effectively learns to align piece boundaries, a crucial aspect in solving the puzzle.

Table 3. Results of 150 pieces puzzle solving with 7% erosion.

Method	Piece-level	Puzzle-level
SJP-ED[3]	66.7%	10%
JPDVT(Ours)	<b>75.9%</b>	<b>45%</b>

Table 4. Comparison to the SOTA of  $3 \times 3$  puzzle recognition on the *JPwLEG-3* [24] dataset.

Method	Venue	Piece-level
Siamese [24]	AAAI-2023	59.7%
Puzzlet [25]	ICASSP-2023	58.2%
SJP-ED [3]	CVPR-2020	55.2%
Deepzzle [19]	TIP-2020	52.3%
JPDVT(Ours)	-	<b>71.3%</b>

Table 5. Normalized Kendall Distance for Temporal Puzzles without missing data measures the number of pairwise disagreements between two sequences.

Datasets	Method	Normalized Kendall Distance x $10^3$				
Size, #Pieces		1, 20	2, 10	4, 5		
	STP	1.4	0.04	0.01		
	VCOP	-	-	13.7		
	JPDVT(Ours)	<b>0.7</b>	<b>0.04</b>	<b>0</b>		
MMNIST	J-VAD	173	0.46	0		
	JPDVT(Ours)	<b>0.7</b>	<b>0.04</b>	<b>0</b>		
	Size, #Pieces		1, 32	2, 16	4, 8	8, 4
		STP	-	0.15	0	0
		VCOP	-	-	-	4.6
J-VAD		268.2	58.2	0.04	0	
JPDVT(Ours)	<b>2.6</b>	<b>0.17</b>	<b>0.04</b>	<b>0</b>		
CLEVRER	VCOP	-	-	-	14.5	
	J-VAD	-	164.3	3.8	0.5	
	JPDVT(Ours)	<b>26.2</b>	<b>4.2</b>	<b>1.2</b>	<b>0.3</b>	
UCF	STP	-	0	0	0	
	VCOP	-	-	-	14.5	
JPDVT(Ours)	<b>7.0</b>	<b>3.0</b>	<b>0.19</b>	<b>0.04</b>		

#### 4.4. Temporal Jigsaw Puzzles Experiments

We constructed temporal puzzles from videos by segmenting them, where each piece is a clip comprising one or more frames in the correct temporal order. Specifically, in the case of MovingMNIST, we built puzzles with: 20 pieces, where each piece is a single frame; 10 pieces, each comprising a pair of consecutive frames; and 5 pieces, with each piece containing four consecutive frames. Similarly, for the CLEVRER, UCF, and QST datasets, our evaluation included puzzles with 32 pieces (single frames), 16 pieces (two frames), 8 pieces (four frames), and 4 pieces (eight frames). Qualitative examples in Fig. 5 and 6 demonstrate instances where our approach adeptly resolved the puzzles without and with missing data, respectively.

We compared the performance of our algorithm solving temporal puzzles, without missing data, against the state-of-the-art methods STP (Space-Time Puzzle) [6], VCOP (Video Clip-Order Prediction) [33], and J-VAD (Joint Video Anomaly Detection) [30], using four video datasets, whenever possible, as described below. We trained these models following the processes detailed in their respective papers, with adjustments made to network sizes to accommodate varying dataset dimensions. To ensure fair comparisons, we fixed the first frame as an anchor frame, and fully randomly shuffle the remaining frames with no mask. Detailed training hyperparameters are available in the appendix.

STP adapts its inputs based on video characteristics. For dynamic textures, exemplified by the QST dataset, it utilizes the PCA decomposition of pixels. However, for more structured videos, such as MovingMNIST and CLEVRER, STP requires feature correspondences across the input frames. Thus, for MovingMNIST and CLEVRER we ran experiments by providing STP ground-truth object coordinates, giving this method an unfair advantage. Regrettably, providing feature correspondences for the UCF dataset was impractical due to its complexity. Additionally, the computational demands of STP rendered unfeasible to run for puzzles with 32 pieces, since its inference time for this setting takes over 30 minutes/video. In contrast, our approach takes approximately 10 seconds/video.

VCOP and J-VAD, on the other hand, have faster inference times, approximately 1 sec/video. However, VCOP predicts the permutation of videos by classifying over a vast number of possibilities. For instance, with 10 clips, the model must classify among  $10! = 3,628,800$  classes. As the number of video clips for ordering increases, training the model becomes impractical due to the exponential growth in classes. J-VAD, employs a 3D-CNN model coupled with a classifier to directly predict the position of each input frame or clip. However, its performance experiences a significant decline when tested on 16 and 32 clips. This phenomenon aligns with findings in [30], where performance deterioration is reported for frame numbers larger than 9,

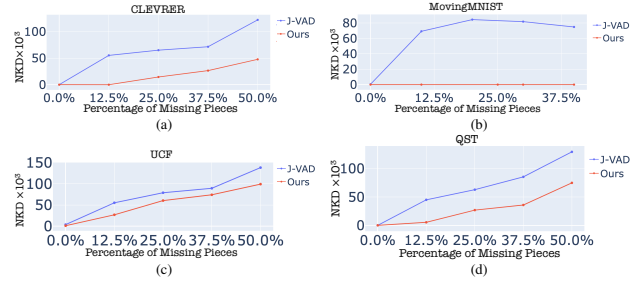


Figure 7. Normalized Kendall distances vs. missing pieces. (a) CLEVRER dataset with 8 pieces, each containing 4 frames. (b) MovingMNIST dataset with 8 pieces, each with 4 frames. (c) UCF dataset with 8 pieces, each with 4 frames. (d) QST dataset with 8 pieces, each with 4 frames.

consistent with our observations.

Table 2 provides the quantitative results of the comparisons against the baselines. Our approach achieves State-Of-The-Art performance across the board and second best for the QST dataset. While STP has comparable performance in some settings and best performance for QST, as noted above, it uses different types of inputs depending on the video characteristics and in this case ground-truth feature correspondences, while our approach always works directly on the raw data. In addition, while our approach is an order of magnitude slower than VCOP and J-VAD, it can handle all the testing settings while the others cannot.

Finally, JPDVT can handle temporal puzzles with missing fragments, as shown in Fig. 6. As seen in Fig. 7, JPDVT’s performance gradually decreases with an increasing number of random missing pieces. This capability allows the use of JPDVT to increase the temporal resolution of videos, as illustrated in the supplemental material.

## 5. Conclusion

Solving puzzles is a fundamental task involving the re-assembly of disordered fragments, a classic problem in pattern recognition that traces its roots back to 1964 [8]. In this paper, we present JPDVT, a novel method for solving image and temporal puzzles in an unified framework. Our approach works by describing the underlying data as an unordered set of pairs, associating the (2D or 1D) positional encoding of a puzzle piece with an embedding of its visual content. This representation enables the formulation of the puzzle-solving problem as a Conditional Diffusion Denoising process, where the unknown positional encodings of the shuffled puzzle pieces are recovered through a reverse denoising process, conditioned on the visual content embeddings of the given elements. JPDVT achieves state-of-the-art performance in quantitative comparisons and produces high-quality qualitative examples, while solving image and video puzzles. Notably, it can handle missing data and solve larger puzzles than previous approaches.



## References

- [1] Tali Basha, Yael Moses, and Shai Avidan. Photo sequencing. In *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12*, pages 654–667. Springer, 2012. [2](#)
- [2] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k. *arXiv preprint arXiv:2205.01580*, 2022. [4](#)
- [3] D. Bridger and *et al.* Solving jigsaw puzzles with eroded boundaries. In *CVPR*, 2020. [1](#), [2](#), [5](#), [6](#), [7](#)
- [4] Tali Dekel, Yael Moses, and Shai Avidan. Photo sequencing. *International journal of computer vision*, 110:275–289, 2014. [2](#)
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. [4](#)
- [6] Caglayan Dicle, Burak Yilmaz, Octavia Camps, and Mario Sznajder. Solving temporal puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5896–5905, 2016. [2](#), [5](#), [8](#)
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. [1](#)
- [8] Herbert Freeman and L Garder. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers*, (2):118–127, 1964. [8](#)
- [9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. [3](#)
- [10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. [1](#)
- [11] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI conference on artificial intelligence*, pages 8545–8552, 2019. [1](#), [2](#)
- [12] Kyungmin Kim, Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Zhicheng Yan, Peter Vajda, and Seon Joo Kim. Rethinking the self-attention in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3071–3075, 2021. [2](#)
- [13] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE international conference on computer vision*, pages 667–676, 2017. [2](#)
- [14] Ru Li, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Jigsawgan: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks. *IEEE Transactions on Image Processing*, 31:513–524, 2021. [2](#)
- [15] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 527–544. Springer, 2016. [2](#)
- [16] Yael Moses, Shai Avidan, *et al.* Space-time tradeoffs in photo sequencing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 977–984, 2013. [2](#)
- [17] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. [4](#)
- [18] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016. [1](#), [2](#)
- [19] M. Paumard and *et al.* Deepzlle: Solving visual jigsaw puzzles with deep learning and shortest path optimization. *IEEE Trans. on Image Proc.*, 29:3569–3581, 2020. [2](#), [5](#), [7](#)
- [20] Marie-Morgane Paumard, David Picard, and Hedi Tabia. Image reassembly combining deep learning and shortest path problem. In *Proceedings of the European conference on computer vision (ECCV)*, pages 153–167, 2018. [2](#)
- [21] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. [4](#)
- [22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. [4](#)
- [23] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, *et al.* Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2022. [4](#), [5](#)
- [24] X. Song and *et al.* Siamese-discriminant deep reinforcement learning for solving jigsaw puzzles with large eroded gaps. In *AAAI*, 2023. [2](#), [4](#), [5](#), [6](#), [7](#)
- [25] X. Song and *et al.* Solving jigsaw puzzle of large eroded gaps using puzzlet discriminant network. In *ICASSP*, 2023. [2](#), [5](#), [7](#)
- [26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. [4](#)
- [27] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015. [4](#)
- [28] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with

- fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 4
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1
- [30] Guodong Wang, Yunhong Wang, Jie Qin, Dongming Zhang, Xiuguo Bao, and Di Huang. Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In *European Conference on Computer Vision*, pages 494–511. Springer, 2022. 2, 8, 5
- [31] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1910–1919, 2019. 2, 5, 6, 7
- [32] Jie Xiao, Xueyang Fu, Man Zhou, Hongjian Liu, and Zheng-Jun Zha. Random shuffle transformer for image restoration. In *International Conference on Machine Learning*, pages 38039–38058. PMLR, 2023. 1
- [33] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10334–10343, 2019. 2, 8, 5
- [34] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B Tenenbaum. Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*, 2019. 4
- [35] Nikolaos-Antonios Ypsilantis, Noa Garcia, Guangxing Han, Sarah Ibrahimi, Nanne Van Noord, and Giorgos Tolias. The met dataset: Instance-level recognition for artworks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 4
- [36] Jiangning Zhang, Chao Xu, Liang Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 300–315. Springer, 2020. 4