Shortcut-V2V: Compression Framework for Video-to-Video Translation based on Temporal Redundancy Reduction

Chaeyeon Chung*[1], Yeojeong Park*[1,2], Seunghwan Choi[1], Munkhsoyol Ganbat[1], and Jaegul Choo[1]

[1]KAIST AI

[2]KT Research & Development Center, KT Corporation

{cy_chung, indigopyj, shadow2496, soyol, jchoo}@kaist.ac.kr

Abstract

Video-to-video translation aims to generate video frames of a target domain from an input video. Despite its usefulness, the existing networks require enormous computations, necessitating their model compression for wide use. While there exist compression methods that improve computational efficiency in various image/video tasks, a generally-applicable compression method for video-to-video translation has not been studied much. In response, we present Shortcut-V2V, a general-purpose compression framework for video-to-video translation. Shortcut-V2V avoids full inference for every neighboring video frame by approximating the intermediate features of a current frame from those of the previous frame. Moreover, in our framework, a newly-proposed block called AdaBD adaptively blends and deforms features of neighboring frames, which makes more accurate predictions of the intermediate features possible. We conduct quantitative and qualitative evaluations using well-known video-to-video translation models on various tasks to demonstrate the general applicability of our framework. The results show that Shortcut-V2V achieves comparable performance compared to the original video-to-video translation model while saving 3.2-5.7× computational cost and 7.8-44× memory at test time. Our code and videos are available at https://shortcut-v2v.github.io/.

1. Introduction

Video-to-video translation is a task of generating temporally consistent and realistic video frames of a target domain from a given input video. Recent studies on video-to-video translation present promising performance in various domains such as inter-modality translation between labels and videos [34, 33, 18], and intra-modality translation between driving scene videos [32] or face videos [1].

Despite enhanced usefulness, video-to-video translation networks usually require substantial computational cost and memory usage, which limits their applicability. For instance, multiply–accumulates (MACs) of a widely-used video translation model, vid2vid [34], is 2066.69G, while the basic convolutional neural networks, ResNet v2 50 [10] and Inception v3 [28], are 4.12G and 6G, respectively. Furthermore, temporally redundant computations for adjacent video frames also harm the cost efficiency of a video-to-video translation network. Performing full inference for every neighboring video frame that contains common visual features inevitably entails redundant operations [20, 19].

In this regard, Fast-Vid2Vid [40] proposes a compression framework for vid2vid [34] based on spatial input compression and temporal redundancy reduction. However, it cannot be applied to other video-to-video translation models since it is designed specifically for vid2vid. Moreover, Fast-Vid2Vid does not support real-time inference since it requires future frames to infer a current one. Alternatively, one can apply model compression approaches for image-to-image translation [13, 22, 12] directly to video-to-video translation, considering video frames as separate images. However, these approaches are not designed to consider the correlation among adjacent video frames during the compression. This may result in unrealistic output quality in video-to-video translation, where the inherent temporal coherence of an input video needs to be preserved in the outputs. Also, frame-by-frame inference without temporal redundancy reduction involves unnecessary computations, resulting in computational inefficiency.

In this paper, we propose *Shortcut-V2V*, a general-purpose framework for improving the computational efficiency of video-to-video translation based on temporal redundancy reduction. Shortcut-V2V allows the original video-to-video translation model to avoid temporally re-

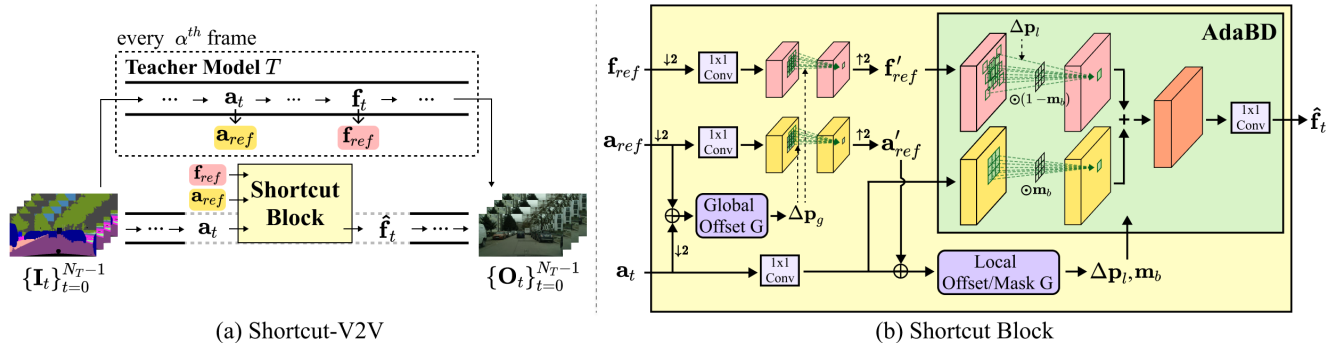---

*indicates equal contributions.

Figure 1: Overview of Shortcut-V2V. (a) is an overall framework of Shorcut-V2V, and (b) shows a detailed architecture of Shortcut block. $\uparrow 2$ and $\downarrow 2$ refer to upsampling and downsampling by a factor of 2, respectively. G in Offset G and Offset/Mask G indicates a generator.

dundant computations by approximating the decoding layer features of the current frame with largely reduced computations. To enable lightweight estimation, our framework leverages features from the previous frame (*i.e.*, reference features), which have high visual similarity with the current frame. We also exploit current frame features from the encoding layer to handle newly-appeared regions in the current frame. Specifically, we first globally align the previous frame features with the current frame features, and our novel *Adaptive Blending and Deformation block (AdaBD)* in Shortcut-V2V blends features of neighboring frames while performing detailed deformation. AdaBD adaptively integrates the features regarding their redundancy in a lightweight manner.

In this way, our model significantly improves the test-time efficiency of the original network while preserving its original performance. Shortcut-V2V is easily applicable to a pretrained video-to-video translation model to save computational cost and memory usage. Our framework is also suitable for real-time inference since we do not require future frames for the current frame inference. To the best of our knowledge, this is the first attempt at a general-purpose model compression approach for video-to-video translation. We demonstrate the effectiveness of our approach using well-known video-to-video translation models, Unsupervised RecycleGAN [32] (Unsup) and vid2vid [34]. Shortcut-V2V reduces 3.2-5.7$\times$ computational cost and 7.8-44$\times$ memory usage while achieving comparable performance to the original model. Since there is no existing general-purpose compression method, we compare our method with Fast-Vid2Vid [40] and the compression methods for image-to-image translation. Our model presents superiority over the existing approaches in both quantitative and qualitative evaluations. Our contributions are summarized as follows:

- We introduce a novel, general-purpose model com-

pression framework for video-to-video translation, Shortcut-V2V, that enables the original network to avoid temporally redundant computations.

- We present AdaBD that exploits features from neighboring frames via adaptive blending and deformation in a lightweight manner.

- Our framework saves up to 5.7$\times$ MACs and 44$\times$ parameters across various video-to-video translation tasks, achieving comparable performance to the original networks.

2. Related Work

2.1. Video-to-Video Translation

Recent video-to-video translation networks are generally classified into pix2pixHD-based [35] and CycleGAN-based [38] generator. vid2vid [34] proposes a pix2pixHD-based sequential generation framework that synthesizes a current output given the previous outputs as additional guidance. As the following work, few-shot vid2vid [33] achieves few-shot generalization of vid2vid based on attention modules, and world-consistent vid2vid [18] is proposed to improve long-term temporal consistency of vid2vid.

While pix2pixHD-based models require paired annotated videos, RecycleGAN [1] and MocycleGAN [4] propose CycleGAN-based video translation models. They exploit spatio-temporal consistency losses to generate realistic videos using unpaired datasets. STC-V2V [21] leverages optical flow for semantic/temporal consistency to improve the output quality of the existing models. Unsupervised RecycleGAN [32] achieves state-of-the-art performance among CycleGAN-based frameworks with a pseudo-supervision by the synthetic flow. Although the existing video-to-video translation networks achieve decent performance, they commonly demand a non-trivial amount of

computational costs and memory usage. Also, frame-by-frame inference necessarily causes redundant operations due to temporal redundancy among adjacent frames.

2.2. Model Compression

Model compression for video-related tasks has been actively proposed in various domains [19, 20, 16, 9, 27, 8, 14, 29], such as object detection, action recognition, semantic segmentation, and super-resolution. Several studies [19, 9, 20] exploit temporal redundancy among video frames to improve efficiency during training or inference. For instance, Habibian *et al*. [9] distill only the residual between adjacent frames from a teacher model to a student to speed up the inference. Also, Fast-Vid2Vid [40] firstly proposes a compression framework for video-to-video translation based on spatial and temporal compression. However, Fast-Vid2Vid focuses on vid2vid [34], limiting its application to other video-to-video translation networks. Also, temporal redundancy reduction via motion compensation in Fast-Vid2Vid requires the future frame to infer the current frame, which is not suitable for real-time inference.

Alternatively, model-agnostic compression methods for image-to-image translation can also be applied to video-to-video translation models. The existing approaches for image synthesis mainly tackle channel pruning [13], knowledge distillation [12, 22], NAS [13], etc. For instance, CAT [12] compresses the teacher network with one-step pruning to satisfy the target computation budget, while OMGD [22] conducts a single-stage online distillation in which the teacher generator supports the student generator to be refined progressively. However, image-based compression methods cannot consider temporal coherence among neighboring video frames, which may induce unrealistic results in video translation tasks. Also, performing full model inference for each video frame still poses computational inefficiency due to the temporal redundancy.

2.3. Deformable Convolution

Deformable convolution [6] is originally proposed to enhance the transformation capability of a convolutional layer by adding the estimated offsets to a regular convolutional kernel in vision tasks such as object detection or semantic segmentation. Besides its original application, recent studies [30, 7, 36, 2, 11, 15] demonstrate that deformable convolution is also capable of aligning adjacent video frames. TDAN [30] utilizes deformable convolution to capture implicit motion cues between consecutive frames by dynamically predicted offsets in video super-resolution. In addition, EDVR [36] stacks several deformable convolution blocks to estimate large and complex motions in video restoration. In this paper, we also leverage a deformable convolution to adaptively align adjacent video frames in a lightweight manner.

Algorithm 1 Shortcut-V2V Inference

1: **Input:** Input video $\{\mathbf{I}_t\}_{t=0}^{N_T-1}$ of length N_T, teacher model T, layer index of encoder l_e and decoder l_d, Shortcut block S, max interval α
2: **Output:** Output video $\{\mathbf{O}_t\}_{t=0}^{N_T-1}$
3: **for** $t = 0$ to $N_T - 1$ **do**
4:     $\mathbf{a}_t = T_{[:l_e]}(\mathbf{I}_t)$
5:     **if** $t\%\alpha = 0$ **then**
6:         $\mathbf{f}_t = T_{[l_e+1:l_d-1]}(\mathbf{a}_t)$
7:         $\mathbf{a}_{\text{ref}}, \mathbf{f}_{\text{ref}} \leftarrow \mathbf{a}_t, \mathbf{f}_t$ ▷ Update the reference features
8:     **else**
9:         $\mathbf{f}_t = S(\mathbf{f}_{\text{ref}}, \mathbf{a}_{\text{ref}}, \mathbf{a}_t)$
10:    **end if**
11:    $\mathbf{O}_t = T_{[l_d:]}(\mathbf{f}_t)$
12: **end for**

---

3. Shortcut-V2V

In this paper, we propose Shortcut-V2V, a general compression framework to improve the test-time efficiency in video-to-video translation. As illustrated in Fig. 1(a), given $\{\mathbf{I}_t\}_{t=0}^{N_T-1}$ as input video frames, we first use full teacher model T to synthesize the output of the first frame. Then, for the next frames, our newly-proposed Shortcut block efficiently approximates $\mathbf{f}_t$, the features from the l_d-th decoding layer of the teacher model. This is achieved by leveraging the l_e-th encoding layer features $\mathbf{a}_t$ along with reference features, $\mathbf{a}_{ref}$ and $\mathbf{f}_{ref}$, from the previous frame. Here, l_d and l_e correspond to layer indices of the teacher model. Lastly, predicted features $\hat{\mathbf{f}}_t$ are injected into the following layers of the teacher model to synthesize the final output $\hat{\mathbf{O}}_t$. To avoid error accumulation, we conduct full teacher inference and update the reference features at every max interval α. We provide the detailed inference process of Shortcut-V2V in Algorithm 1.

The architecture of our model is mainly inspired by Deformable Convolutional Network (DCN) [6, 39], which we explain in the next section.

3.1. Deformable Convolutional Network

DCN [6, 39] is initially introduced to improve the transformation capability of a convolutional layer in image-based vision tasks, *e.g*., object detection and semantic segmentation. In the standard convolutional layer, a 3×3 kernel with dilation 1 samples points over input features using a sampling position $\mathbf{p}_k \in \{(-1,-1),(-1,0),...,(0,1),(1,1)\}$. Given input feature maps $\mathbf{x}$, DCN predicts additional offsets $\Delta\mathbf{p} \in \mathbb{R}^{2N_{\mathbf{P}} \times H \times W}$ to augment each sampling position along x-axis and y-axis. Here, $N_{\mathbf{P}}$ is the number of sampling positions in a kernel, and H and W are the height and width

of output feature maps, respectively. For further manipulation of input feature amplitudes over the sampled points, DCNv2 [39] introduces a modulated deformable convolution with $\mathbf{m} \in \mathbb{R}^{N_{\mathbf{P}} \times H \times W}$ consisting of learnable modulation scalars. The deformed output feature maps $\mathbf{x}'$ by DCNv2 are defined as:

$$\mathbf{x}' = f_{dc}(\mathbf{w}, \mathbf{x}, \Delta\mathbf{p}, \mathbf{m}), \quad (1)$$

where f_{dc} indicates a deformable convolution. Specifically, a single point $\mathbf{p_o}$ of $\mathbf{x}'$ is obtained as:

$$\mathbf{x}'(\mathbf{p_o}) = \sum_{k=1}^{N_{\mathbf{P}}} \mathbf{w}(\mathbf{p}_k) \cdot \mathbf{x}(\mathbf{p_o} + \mathbf{p}_k + \Delta\mathbf{p}(\mathbf{p}_k)) \cdot \mathbf{m}(\mathbf{p}_k), \quad (2)$$

where $\mathbf{w}(\mathbf{p}_k)$, $\mathbf{p}(\mathbf{p}_k)$, and $\mathbf{m}(\mathbf{p}_k)$ are convolutional layer weights, offsets, and modulation scalars between 0 and 1, respectively, for the k-th sampling position.

Taking advantage of the enhanced transformation capability, we also leverage deformable convolution to align features of adjacent frames only with a few convolution-like operations, instead of using heavy flow estimation networks.

3.2. Shortcut Block

As described in Fig. 1(b), Shortcut block S estimates the current frame features $\hat{\mathbf{f}}_t$ given $\mathbf{a}_t$ and the reference frame features $\mathbf{f}_{ref}$ and $\mathbf{a}_{ref}$ as inputs:

$$\hat{\mathbf{f}}_t = S(\mathbf{f}_{ref}, \mathbf{a}_{ref}, \mathbf{a}_t). \quad (3)$$

Our block effectively obtains rich information from $\mathbf{f}_{ref}$ via coarse-to-fine alignment referring to alignment between $\mathbf{a}_{ref}$ and $\mathbf{a}_t$. Also, during the fine alignment, our newly-proposed AdaBD simultaneously performs adaptive blending of the aligned $\mathbf{f}_{ref}$ and the current frame feature $\mathbf{a}_t$. Here, $\mathbf{a}_t$ supports the synthesis of newly-appeared areas in the current frame.

Coarse-to-Fine Alignment. To handle a wide range of misalignments between the frames, our model aligns $\mathbf{f}_{ref}$ with the current frame in a coarse-to-fine manner. Our global/local alignment module consists of an offset generator to estimate offsets, and deformable convolution layers to deform features based on the predicted offsets. Following TDAN [30], an offset generator estimates sampling offsets given the adjacent frame features. For global alignment, we first downsample the given inputs to enlarge the receptive fields of the corresponding convolutional layers in a lightweight manner. The downsampled $\mathbf{a}_{ref}$ and $\mathbf{a}_t$ are concatenated and fed into a global offset generator to generate global offsets $\Delta\mathbf{p}_g \in \mathbb{R}^{2 \times \frac{H}{2} \times \frac{W}{2}}$. Since we only need to capture coarse movement, $\Delta\mathbf{p}_g$ includes a single offset for each kernel, unlike the original DCN. Each offset is identically applied to all the sampling positions within the

kernel. Then, the deformed features are upsampled back to the original size to obtain $\mathbf{f}'_{ref}$ as follows:

$$\mathbf{f}'_{ref} = (f_{dc}(\mathbf{w}_g, (\mathbf{f}_{ref})^{\downarrow 2}, \Delta\mathbf{p}_g, \mathbf{1}))^{\uparrow 2}, \quad (4)$$

where $\mathbf{w}_g$ denotes weights of global deformable convolution, and $(\cdot)^{\uparrow 2}$ and $(\cdot)^{\downarrow 2}$ refer to upsampling and downsampling by a factor of 2 through bilinear interpolation, respectively. $\mathbf{1}$ indicates a vector filled with 1 so that no modulation is applied here. While local alignment of the coarsely-aligned feature $\mathbf{f}'_{ref}$ follows the process of global alignment, the difference lies in that each sampling point of each kernel has a unique offset, and the alignment operation is conducted in the original resolution. We leverage $\mathbf{a}'_{ref}$ and $\mathbf{a}_t$ to estimate local offsets, where $\mathbf{a}'_{ref}$ is the aligned $\mathbf{a}_{ref}$ which is downsampled, deformed, and upsampled with the same weights $\mathbf{w}_g$ used to synthesize $\mathbf{f}'_{ref}$.

We estimate the offsets for the decoding layer features $\mathbf{f}$ using the encoding layer features $\mathbf{a}$ under the assumption that $\mathbf{a}_t$ and $\mathbf{f}_t$ have the same structural information. In video-to-video translation, input and output frames share the same underlying structure. Thus, it is natural for the network to learn to maintain the structural information of an input frame throughout the encoding and decoding process. More details are described in our supplementary materials.

Adaptive Blending and Deformation. During the local alignment, we also take advantage of the current frame features $\mathbf{a}_t$ from the encoding layer to handle the regions with large motion differences and new objects. To achieve this in a cost-efficient way, we introduce AdaBD, which simultaneously aligns $\mathbf{f}'_{ref}$ and blends it with $\mathbf{a}_t$ in an adaptive manner, as illustrated in Fig. 1(b) AdaBD.

First, our local offset/mask generator predicts a blending mask $\mathbf{m}_b \in \mathbb{R}^{N_{\mathbf{P}} \times H \times W}$ in addition to the local offsets $\Delta\mathbf{p}_l \in \mathbb{R}^{2N_{\mathbf{P}} \times H \times W}$. A learnable mask $\mathbf{m}_b$ is composed of modulation scalars ranging from 0 to 1, each of which indicates the blending ratio of the current features $\mathbf{a}_t$ to the aligned reference features. While DCNv2 [39] originally introduces the modulation scalars to control feature amplitudes of a single input, we leverage the scalars to adaptively blend the features from two adjacent frames considering their redundant areas.

We apply deformable convolution by adding local offsets $\Delta\mathbf{p}_l$ to sampling positions of the coarsely-aligned reference features $\mathbf{f}'_{ref}$, while the current frame features $\mathbf{a}_t$ are fed into standard convolutional operations. Concurrently, blending mask $\mathbf{m}_b$ adaptively combines the two feature maps. In detail, an output point $\mathbf{p_o}$ of $\hat{\mathbf{f}}_t$ is calculated as follows:

$$\hat{\mathbf{f}}_t(\mathbf{p_o}) = \sum_{k=1}^{N_{\mathbf{P}}} \mathbf{w}_l(\mathbf{p}_k) \cdot \{\mathbf{a}_t(\mathbf{p_o} + \mathbf{p}_k) \cdot \mathbf{m}_b(\mathbf{p}_k) +$$
$$\mathbf{f}'_{ref}(\mathbf{p_o} + \mathbf{p}_k + \Delta\mathbf{p}_l(\mathbf{p}_k)) \cdot (1 - \mathbf{m}_b(\mathbf{p}_k))\}, \quad (5)$$

where $\mathbf{w}_l$ indicates weights of local deformable convolution.

Intuitively, the higher values of $\mathbf{m}_b$ indicate the regions where current frame features are more required. In other words, Eq. 5 can be rewritten as a summation of standard convolution and deformable convolution:

$$\hat{\mathbf{f}}_t = f_{dc}(\mathbf{w}_l, \mathbf{a}_t, \mathbf{0}, \mathbf{m}_b) + f_{dc}(\mathbf{w}_l, \mathbf{f}'_{ref}, \Delta\mathbf{p}_l, 1 - \mathbf{m}_b), \tag{6}$$

In this equation, the convolutional weights $\mathbf{w}_l$ are shared between $\mathbf{a}_t$ and $\mathbf{f}'_{ref}$. f_{dc} with $\mathbf{0}$ indicates DCN with zero offsets, illustrating standard convolutional operation. To save computational costs, we decrease the channel dimension of all input features and reconstruct the original channel size before injecting the output features into the remaining layers of the teacher network.

3.3. Training Objectives

To train Shortcut-V2V, we mainly leverage alignment loss, distillation loss, and GAN losses widely used in image/video translation networks [35, 34, 38]. First, we adopt alignment loss L_{align} to train the deformation layers in Shortcut-V2V. Since Shortcut-V2V aims to align the reference frame features $\mathbf{f}_{ref}$ with the current frame, L_{align} computes L1 loss between the aligned feature $\mathbf{f}^*_{ref}$ and the current frame features $\mathbf{f}_t$ extracted from the teacher model. To obtain $\mathbf{f}^*_{ref}$, we align $\mathbf{f}_{ref}$ in a coarse-to-fine manner without an intervention of $\mathbf{m}_b$ or blending with the current features. The alignment loss L_{align} is formulated as follows:

$$\mathbf{f}^*_{ref} = f_{dc}(\mathbf{w}_l, \mathbf{f}'_{ref}, \Delta\mathbf{p}_l, 1), \tag{7}$$

$$L_{align} = \left\| \mathbf{f}_t - \mathbf{f}^*_{ref} \right\|_1. \tag{8}$$

Additionally, we employ knowledge distillation losses at the feature and output levels. A feature-level distillation loss L_{feat} is applied between the estimated feature $\hat{\mathbf{f}}_t$ and the ground truth feature $\mathbf{f}_t$, while an output-level distillation loss L_{out} compares the approximated output $\hat{\mathbf{O}}_t$ to the output $\mathbf{O}_t$ generated by the teacher network. The perceptual loss [37] L_{perc} is also incorporated to distill the high-frequency information of the outputs.

Lastly, we utilize a typical GAN loss L_{GAN} and a temporal GAN loss L_{T-GAN}, following the existing video-based frameworks [34, 3]. Temporal GAN loss L_{T-GAN} encourages both temporal consistency and realisticity of the output frames. For the GAN losses, we consider the outputs of the teacher network as real images.

The overall objective function L_{total} is as follows:

$$L_{total} = \lambda_{align}L_{align} + \lambda_{feat}L_{feat} + \lambda_{out}L_{out}$$
$$+ \lambda_{perc}L_{perc} + \lambda_{GAN}L_{GAN} + \lambda_{T-GAN}L_{T-GAN}, \tag{9}$$

Model	Dataset	Method	FVD	MACs (G)	Param. (M)
Unsup	V2C	Original	1.253	84.61	7.84
		CAT	1.310	22.83	1.48
		OMGD	1.199	25.85	1.21
		Ours	**1.180**	**18.19**	**0.32**
	L2V	Original	1.663	42.36	7.84
		CAT	1.957	12.02	1.71
		OMGD	2.467	12.93	1.21
		Ours	**1.754**	**9.09**	**0.32**
vid2vid	E2F	Original	0.186	2066.69	365
		CAT	0.288	380.55	24.67
		OMGD	0.264	485.84	35.81
		Fast-Vid2Vid	0.223	767.63	63.17
		Ours	**0.209**	**359.99**	**8.29**
	L2C	Original	0.146	1254.17	411.34
		CAT	0.304	399.36	71.11
		OMGD	0.346	391.59	**47.83**
		Fast-Vid2Vid	0.310	455.29	71.8
		Ours	**0.165**	**389.16**	52.58

Table 1: Quantitative comparison with baselines.

Dataset	Method	mIoU	AC	MP
V2L	Original	12.2	16.0	62.8
	CAT	5.99	8.82	44.2
	OMGD	9.76	13.0	58.7
	Ours	**11.5**	**15.2**	**61.4**
L2V	Original	10.0	15.6	47.3
	CAT	4.10	7.81	27.6
	OMGD	3.42	6.87	29.5
	Ours	**9.24**	**14.9**	**43.5**

Table 2: Comparison of segmentation score for V2L and FCN-score for L2V on Unsup.

where λ_{align}, λ_{feat}, λ_{out}, λ_{perc}, λ_{GAN}, and λ_{T-GAN} are hyperparameters to control relative significance among the losses. More details on training objectives are described in the supplementary materials.

4. Experiments

4.1. Experimental Settings

Models and Datasets. To demonstrate the generality of our model, we conduct experiments on various tasks with widely-used video-to-video translation models, Unsupervised RecycleGAN [32] and vid2vid [34].

Unsupervised RecycleGAN (Unsup) is the state-of-art video-to-video translation model among CycleGAN-based [38] ones. We conduct experiments for Unsup on Viper→Cityscapes (V2C), which involves translating inter-modality from game driving scenes (Viper [23]) to real-world driving scenes (Cityscapes [5]). The Viper dataset consists of driving scenes collected from the GTA-V game engine, with 56 training videos and 21 test videos.
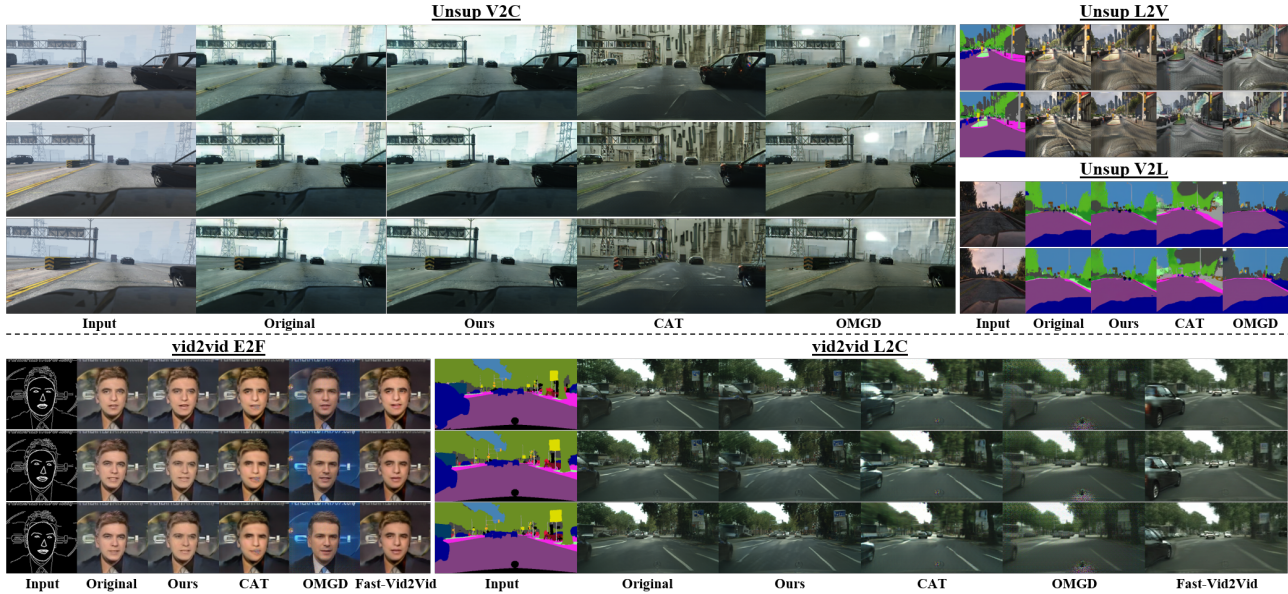
Figure 2: Qualitative comparison. The upper rows represent the results of Unsup V2C, L2V, and V2L. The bottom rows show the results of vid2vid E2F and L2C.

Cityscapes is a dataset composed of 2,975 training videos and 500 validation videos. The input images are resized into 256×512 for the experiments. We also tackle the translation of the videos from the Viper dataset to their corresponding segmentation label maps, Viper→Label (V2L), and vice versa, Label→Viper (L2V). We resize the images and the labels into 256×256 following the previous work [1, 32].

vid2vid is a widely-used pix2pixHD-based video-to-video translation network that serves as a base architecture for various recent video-to-video translation models [33, 18]. Following vid2vid, we evaluate Shortcut-V2V on Edge→Face (E2F) and Label→Cityscapes (L2C). E2F translates edge maps into facial videos from the Face-Forensics [24], containing 704 videos for training and 150 videos for validation with various lengths. The images are cropped and resized to 512×512 for the experiments. The edge maps are extracted using the estimated facial landmarks and Canny edge detector. Also, L2C synthesizes videos of driving scenes from segmentation label maps using the Cityscapes. We generate segmentation maps using pretrained networks following the previous studies [34, 40]. The images and the labels are resized into 256×512.

Evaluation Metrics. Primarily, we adopt the Fréchet video distance (FVD) score [31] to evaluate the performance of Shortcut-V2V quantitatively. The FVD score measures the Fréchet Inception distance between the distribution of video-level features extracted from generated and real videos. The lower the FVD score is, the better visual quality and temporal consistency the generated video frames have. For V2L and L2V, we follow the measure-

ment of the evaluation metrics in the teacher model, Unsup [32]. We measure the segmentation scores, mean intersection over union (mIoU), mean pixel accuracy (MP), and average class accuracy (AC), to validate the performance of V2L. L2V is evaluated using FCN-score [17]. For the evaluation, we first estimate label maps from the generated videos using the FCN model pretrained with the Viper dataset. Then, we measure how accurately the estimated label maps are mapped to the ground truth segmentation labels. Higher FCN-scores refer to better output quality.

Implementation Details. We attach our Shortcut block to the fixed teacher networks implemented based on the official codes and pretrained by the authors, except for Unsup V2C. The teacher network of Unsup V2C is trained from scratch in the same way the original paper described [32].

Also, standard convolutional kernels in the Shortcut block are replaced with HetConv [26] to further enhance computational efficiency without a performance drop. For the convenience of implementation and training stability, we intend $\mathbf{a}$ and $\mathbf{f}$ to have the same spatial size. Furthermore, we set the max interval α for each dataset considering the factors that reflect motion differences between frames, such as the frame per second (FPS) of training videos. Lower FPS usually results in larger motion differences between the adjacent frames, requiring a shorter max interval α and vice versa. Specifically, we set α as 3 on V2C, V2L, L2V, and L2C, since the FPS of Viper [23] and Cityscapes [5] is 15 and 17, respectively. Also, α of E2F is set as 6, where the FPS of FaceForensics [24] is 30. Additional details are included in the supplementary materials.
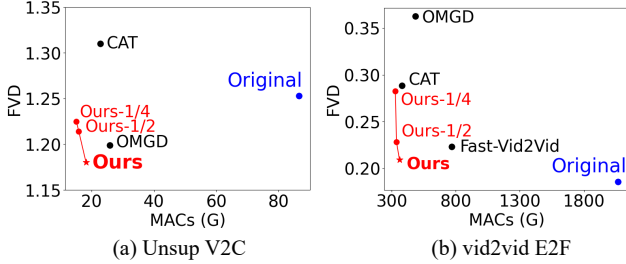
Figure 3: Performance-efficiency trade-off of the original model, Shortcut-V2V, and the existing compression methods including OMGD [22], CAT [12], and Fast-Vid2Vid [40]. We measure the FVD score and MACs, where the lower FVD score indicates better quality. Red points and stars denote ours with various model sizes.

Configurations	Unsup V2C	vid2vid E2F
(a) w/o reference features	1.256	0.398
(b) w/o current features	1.249	0.218
(c) Single-stage alignment	1.195	0.213
(d) w/o adaptive blending	1.208	0.244
(e) Ours	**1.180**	**0.209**

Table 3: An ablation study on Unsup V2C and vid2vid E2F. We measure the FVD scores.

4.2. Comparison to Baselines

To demonstrate the effectiveness of our framework, we conduct qualitative and quantitative evaluations compared to the original model and other baselines. Since this is the first work that tackles a generally applicable compression framework for video-to-video translation, we compare Shortcut-V2V to the existing compression methods for image-to-image translation, CAT [12] and OMGD [22], regarding video frames as individual images. In the case of vid2vid, we additionally conduct a comparison to Fast-Vid2Vid [40], which is the compression method designed specifically for vid2vid. For a fair comparison, we compress the student networks of the baselines to have similar or higher MACs compared to our model.

Qualitative Evaluation. According to Fig. 2, our method presents outputs of comparable visual quality to the original model with much fewer computations. In contrast, CAT on Unsup V2C generates undesirable buildings in the sky, and OMGD on Unsup V2C struggles with noticeable artifacts. For Unsup L2V, CAT and OMGD generate unrealistic textures for the terrain in the middle or the vegetation on the right. Moreover, in V2L, CAT estimates inappropriate labels on the sky and OMGD on the vegetation.

For vid2vid E2F, CAT shows unwanted artifacts on the mouth, and OMGD presents inconsistent outputs. For vid2vid L2C, the outputs of CAT are blurry, especially for the trees in the background, and OMGD generates artifacts at the bottom of the images. Although Fast-Vid2Vid shows reasonable image quality, the output frames are inaccurately aligned with the input (*e.g.*, a head pose of a person) and suffer from ghost effects (*e.g.*, the black car on the left) due to motion compensation using interpolation.

Quantitative Evaluation. As shown in Table 1 and Table 2, our framework successfully improves the computational efficiency of the original network without a significant performance drop. In the case of Unsup, Shortcut-V2V reduces the MACs by $4.7\times$ and the number of parameters by $24.5\times$. In addition, our approach saves vid2vid's MACs by 3.2 and $5.7\times$ and the number of parameters by 7.8 and $44\times$ on each task. Fig. 3 visualizes the performance-efficiency trade-off of our framework and other baselines.

According to Table 1, we also outperform other compression methods for image-to-image translation, CAT [12] and OMGD [22], even with fewer computations. Image-based approaches cannot consider temporal coherence among the frames during the compression, leading to a loss of quality. Meanwhile, we effectively preserve the original performance by exploiting rich information in the previous features during inference. Shortcut-V2V even shows superiority over Fast-Vid2Vid [40], which is a compression method specifically designed for vid2vid. Table 2 also demonstrates that ours on Unsup surpasses the existing compression models by a large margin.

4.3. Ablation Study

We conduct an ablation study to evaluate the effect of each component of Shortcut-V2V. As described in Table 3, we compare the FVD scores of 5 different configurations, (a) w/o reference features, (b) w/o current features, (c) single-stage alignment, (d) w/o adaptive blending, and (e) ours, on Unsup V2C and vid2vid E2F.

First, while our model originally exploits both the reference features $\mathbf{f}_{ref}$ and the current frame features $\mathbf{a}_t$ to synthesize $\hat{\mathbf{f}}_t$, (a) and (b) are designed to leverage only either of them. To be specific, (a) synthesizes $\hat{\mathbf{f}}_t$ by processing standard convolutions on $\mathbf{a}_t$ without integrating the deformed $\mathbf{f}_{ref}$. Meanwhile, (b) estimates $\hat{\mathbf{f}}_t$ using only $\mathbf{f}_{ref}$ aligned in a coarse-to-fine manner without blending $\mathbf{a}_t$. The results demonstrate that the absence of either $\mathbf{f}_{ref}$ or $\mathbf{a}_t$ leads to performance degradation. Next, (c) performs a single-stage alignment on $\mathbf{f}_{ref}$ to show the necessity of a coarse-to-fine alignment. Specifically, (c) leverages a single offset/mask generator to predict $\Delta\mathbf{p}$ and $\mathbf{m}_b$ which are then used in AdaBD for blending and deformation. The result indicates that coarse-to-fine alignment effectively assists the estimation of $\hat{\mathbf{f}}_t$ by minimizing misalignment between the features from adjacent frames. This makes Shortcut-V2V suitable for

Reference frame   Current frame

$\Delta\mathbf{p}_g$ $\Delta\mathbf{p}_g + \Delta\mathbf{p}_l$ $\mathbf{p}_o$ $\mathbf{m}_b$
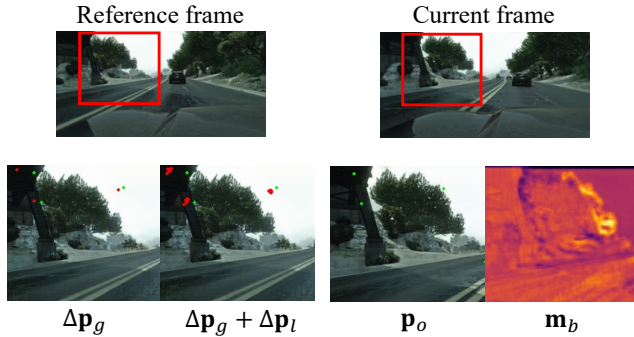
Figure 4: Visualization of global/local offsets and blending masks. The green points denote output points of the current frame, and the red points around them are each output point's sampling positions modified by global offsets $\Delta\mathbf{p}_g$ and local offsets $\Delta\mathbf{p}_l$. The values in a blending mask $\mathbf{m}_b$ are averaged by the kernel size for visualization. The brighter area indicates higher mask values.

generating videos with diverse motion differences. Lastly, (d) blends $\mathbf{f}_{ref}^*$ and $\mathbf{a}_t$ simply by element-wise addition instead of adaptive blending with $\mathbf{m}_b$ in AdaBD. The result demonstrates that $\mathbf{m}_b$ also encourages better estimation of $\hat{\mathbf{f}}_t$ by selectively exploiting features from temporally adjacent features.

4.4. Offset/Mask Visualization

We provide a qualitative analysis of the generated global/local offsets and blending masks. Fig. 4 visualizes the sampling positions (red points) for each output position $\mathbf{p}_o$ of the current frame (green points), where the predicted offsets $\Delta\mathbf{p}_g$ and $\Delta\mathbf{p}_l$ are added to the original sampling positions. According to the result, global offsets $\Delta\mathbf{p}_g$ effectively reflect the global movement of the objects such as the bridge. Also, the summation of the global and local offsets indicates the sampling points refined by local offsets enabling fine alignment. This shows that the estimated offsets effectively support the utilization of common features in the reference frame. For the blending masks $\mathbf{m}_b$, Fig. 4 presents that the regions with significant motion differences (*e.g.*, trees) have large mask values compared to regions with little change (*e.g.*, road, sky). That is, our model relies more on the current features rather than the reference features when the deformation is challenging, which aligns with our intention and leads to robust performance.

4.5. Performance-Efficiency Trade-off

We present a performance-efficiency trade-off of our framework with respect to varying model sizes of Shortcut block and teacher model dependence.
Shortcut Block Size. We construct our models of different sizes by reducing the number of the output chan-

Dependence	Unsup V2C			vid2vid E2F		
	FVD	MACs (G)	Param. (M)	FVD	MACs (G)	Param. (M)
Low	1.221	13.79	0.14	0.277	243.97	2.11
Medium	1.180	18.19	0.32	0.209	359.99	8.29
High	1.166	27.97	1.38	0.193	475.99	32.98

Table 4: Shortcut-V2V performance with different teacher model dependence. High dependency denotes using more teacher network layers during the inference.

nel of 1×1 channel reduction layer by half (Ours-1/2) and a quarter (Ours-1/4). As shown in Fig. 3, despite the performance-efficiency trade-off depending on channel dimension, Ours-1/2 and Ours-1/4 still achieve comparable FVD to the baselines with less computation costs. On a single RTX 3090 GPU, our models with three different sizes on Unsup achieve 99.67, 109.69, and 111.80 FPS with an 11.6-21.2% reduction in inference time compared to the original model's speed of 88.10 FPS. Additionally, Ours, Ours-1/2, and Ours-1/4 on vid2vid contribute to a significant improvement in speed from 5.63 FPS to 12.86, 12.93, and 13.16 FPS, resulting in a 56.2-57.2% reduction in inference time. Our method is capable of a real-time inference, unlike the previous work [40] which requires future frames for the current frame inference due to motion compensation. More details on channel manipulation are described in the supplementary materials.

Teacher Model Dependence. Since Shortcut-V2V leverages a subset of the teacher network during inference, the computational costs and memory usage may vary depending on the amount of teacher network we use. In this regard, we present an analysis of the computational efficiency and performance of our model with respect to various levels of teacher model dependence. We categorize the dependence on the teacher model into three levels, low, medium, and high, where high dependence indicates using more teacher network layers. According to Table 4, our model of higher dependence achieves better performance with larger MACs and the number of parameters. The results also demonstrate that Shortcut-V2V can leverage temporal redundancy in features extracted from various layers of the teacher network. Further details for teacher model dependence and experiments for temporal redundancy are included in the supplementary materials.

5. Discussion and Conclusion

Although Shortcut-V2V significantly improves the test-time efficiency of video-to-video translation, it still poses several limitations. First, a constant max interval may induce unsatisfactory outputs when a degree of temporal redundancy largely varies between frames. Following recent studies [25, 20, 19], applying an adaptive interval based on frame selection algorithms or a learnable policy net-

work could be promising future research. In addition, we need to manually configure various hyperparameters such as channel dimension, max interval, and teacher model dependence, which can be automated by NAS [13]. Lastly, our method still has the potential to further improvement of computational efficiency by compressing the teacher model using other methods before applying our framework.

Despite the limitations, Shortcut-V2V presents a significant improvement in test-time efficiency of video-to-video translation networks based on temporal redundancy reduction, while preserving the original model performance. Extensive experiments with widely-used video-to-video translation models successfully demonstrate the general applicability of our framework. To the best of our knowledge, this is the first work for a general model compression in the domain of video-to-video translation. We hope our work facilitates research on video-to-video translation and extends the range of its application.

References

[1] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. *Proc. of the European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 6

[2] Kelvin C. K. Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Understanding deformable alignment in video super-resolution. *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 3

[3] Ya-Liang Chang, Zhe Yu Liu, Kuan-Ying Lee, and Winston H. Hsu. Free-form video inpainting with 3d gated convolution and temporal patchgan. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[4] Yang Chen, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. Mocycle-gan: Unpaired video-to-video translation. In *ACM Multimedia (ACM MM)*, 2019. 2

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5, 6

[6] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 3

[7] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. Spatio-temporal deformable convolution for compressed video quality enhancement. *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 3

[8] Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation, 2019. 3

[9] Amirhossein Habibian, Haitam Ben Yahia, Davide Abati, Efstratios Gavves, and Fatih Porikli. Delta distillation for efficient video processing, 2022. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[11] Bangrui Jiang, Zhihuai Xie, Zhen Xia, Songnan Li, and Shan Liu. Erdn: Equivalent receptive field deformable network for video deblurring. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2022. 3

[12] Qing Jin, Jian Ren, Oliver J. Woodford, Jiazhuo Wang, Geng Yuan, Yanzhi Wang, and Sergey Tulyakov. Teachers do more than teach: Compressing image-to-image models. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 3, 7

[13] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. GAN compression: Efficient architectures for interactive conditional gans. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 3, 9

[14] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[15] Jiayi Lin, Yan Huang, and Liang Wang. Fdan: Flow-guided deformable alignment network for video super-resolution. *arXiv preprint arXiv:2105.05640*, 2021. 3

[16] Yifan Liu, Chunhua Shen, Changqian Yu, and Jingdong Wang. Efficient semantic video segmentation with per-frame inference. *Proc. of the European Conference on Computer Vision (ECCV)*, 2020. 3

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[18] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. *Proc. of the European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 6

[19] Yue Meng, Rameswar Panda, Chung-Ching Lin, Prasanna Sattigeri, Leonid Karlinsky, Kate Saenko, Aude Oliva, and Rogério Feris. Adafuse: Adaptive temporal fusion network for efficient action recognition. *Proc. the International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 8

[20] Bowen Pan, Rameswar Panda, Camilo Fosco, Chung-Ching Lin, Alex Andonian, Yue Meng, Kate Saenko, Aude Oliva, and Rogerio Feris. Va-red^2: Video adaptive redundancy reduction. *Proc. the International Conference on Learning Representations (ICLR)*, 2021. 1, 3, 8

[21] KwanYong Park, Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In So Kweon. Preserving semantic and temporal consistency for unpaired video-to-video translation. *ACM Multimedia (ACM MM)*, 2019. 2

[22] Yuxi Ren, Jie Wu, Xuefeng Xiao, and Jianchao Yang. Online multi-granularity distillation for gan compression. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 3, 7

[23] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 5, 6

[24] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 6

[25] Jonghyeon Seon, Jaedong Hwang, Jonghwan Mun, and Bohyung Han. Stop or forward: Dynamic layer skipping for efficient action recognition. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 8

[26] Pravendra Singh, Vinay Kumar Verma, Piyush Rai, and Vinay P. Namboodiri. Hetconv: Heterogeneous kernel-based convolutions for deep cnns. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[27] Ximeng Sun, Rameswar Panda, Chun-Fu Chen, Aude Oliva, Rogério Feris, and Kate Saenko. Dynamic network quantization for efficient video inference. *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 3

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[29] Ryan Szeto, Mostafa El-Khamy, Jungwon Lee, and Jason J. Corso. Hypercon: Image-to-video model transfer for video-to-video translation tasks. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3

[30] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. TDAN: temporally deformable alignment network for video super-resolution. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3, 4

[31] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *Proc. the International Conference on Learning Representations Workshop (ICLRW)*, 2019. 6

[32] Kaihong Wang, Kumar Akash, and Teruhisa Misu. Learning temporally and semantically consistent unpaired video-to-video translation through pseudo-supervision from synthetic optical flow. *Proc. the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 1, 2, 5, 6

[33] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2, 6

[34] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Proc. the Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2, 3, 5, 6

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5

[36] Xintao Wang, Kelvin C. K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. EDVR: video restoration with enhanced deformable convolutional networks. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2019. 3

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5

[39] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[40] Long Zhuo, Guangcong Wang, Shikai Li, Wayne Wu, and Ziwei Liu. Fast-vid2vid: Spatial-temporal compression for video-to-video synthesis. *Proc. of the European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 6, 7, 8