UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Multimodal and Generative Representation Learning

by Naiyuan Liu

Thesis submitted in fulfilment of the requirements for the degree of

Master of Research

under the supervision of Prof. Yi Yang and Dr. Linchao Zhu

University of Technology Sydney
Faculty of Engineering and Information Technology

February 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Naiyuan Liu, declare that this thesis, is submitted in fulfilment of the requirements for the award of Master of Research, in the School of Computer Science at the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE:

Production Note:
Signature removed prior to publication.

DATE: 7, February, 2023

ABSTRACT

Multimodal and Generative Representaion Learning

by

Naiyuan Liu

Representation learning is fundamental for most vision and language tasks. The quality of the designed or learned representation of the input text or visual signals determines the success or failure of relevant tasks. In this thesis, I present novel representation learning methods based on deep neural networks for two different tasks: Natural Language Queries (NLQ) and Face Swapping. 1) Natural Language Queries is a multimodal information retrieval task between video and text. Given an egocentric video clip and a text query, the goal of NLQ is to locate a temporal moment of the video clip where the answer to the query can be obtained. How to learn a combined representation by two different modality features: video and text, is the major problem of this task. To address this challenge, we propose a multi-scale cross-modal transformer and a video frame-level contrastive loss to fully uncover the correlation between video and text. 2) Face Swapping is a generative task. Given a target image and a source image, Face Swapping aims to swap the identity of the target image to the identity of the source image, while the other attributes of the target image (background, expression, et al.) should be preserved. The primary issue of this task is the modulation of identity representation which is solved by our ID modulation block. Furthermore, we utilize the rich and diverse representation priors learned in a pre-trained face GAN to obtain high-fidelity and high-quality face-swapped images. Experiments have shown that our designed representation learning methods for these two tasks bring significant improvement.

Dissertation directed by Professor Yi Yang

The Australian Artificial Intelligence Institute (AAII), School of Computer Science

Acknowledgements

First of all, I would like to thank my principal supervisor, Prof. Yi Yang, for his guidance and patience. He often discussed with me in academic works, cared about my research progress, and guided me when I encountered difficulties. In life, he taught me a lot of life principles, and taught me how to cooperate and get along with others. Besides, I would like to thank my co-supervisor Dr. Linchao Zhu, who has provided me with valuable suggestions and supports throughout these two years. Many thanks to Dr. Xiaohan Wang, Mr. Shuai Zhao and Mr. Tianqi Tang, we did a lot of exciting work together. They helped me a lot in academics and engineering. I want thanks Dr.Wenguan Wang and Dr.Tianfei Zhou for their help and suggestion for my academic work. I also thank all my colleagues in the ReLER Lab at the University of Technology Sydney. It is my pleasure to work with them.

Many thanks to Data to Decision CRC for supporting my research.

Finally, I would like to thank my father Yuntang Liu and my mother Shaoying Kong, for their support and care over these years.

Naiyuan Liu
Sydney, Australia, 2023.

Contents

List of Figures

Chapter 1

Introduction

We have witnessed great progress in computer vision over the decade. AlexNet [37] is the first deep neural network-based method to outperform the hand-crafted feature based methods on the image classification task of ImageNet [9]. AlexNet is built by Convolution blocks; we call this kind of model Convolutional Neural Networks (ConvNets). AlexNet proves the potential deep neural network for dealing with various computer vision tasks. Recently, ConvNets achieve extreme high performance on image classification task [59, 25, 62]. R-CNN [18] proposed by Girshick et al. shows that ConvNets can handle complex computer vision tasks: object detection [51, 17, 50] and segmentation [6, 71]. Vision Transformer (ViT) [13] first adopts Transformer [63] for image classification task and get competitive results compared with ConvNets methods. Transformer based methods [4, 43] also perform well on object detection and segmentation.

Despite the great success of image classification, object detection and segmentation, various computer vision tasks remain challenging. In this thesis, I try to tackle two novel computer vision tasks: Natural Language Queries and Face Swapping. Natural Language Queries is an information retrieval task between two modalities: video and text. Natural Language Queries is proposed by FACEBOOK which is severed for their AR device. The purpose of NLQ is to make their AR device more intelligent. FACEBOOK's vision is to hope that the AR glasses will record all the things the user has seen. When the user asks a question, the AR glasses will give an answer based on the clips it records. This is a new and practical task, so I choose to study it. The critical issue of this task is to align the text information to related video content through representation learning. Face Swapping is a novel generative task. This task aims to generate a new face by replacing the identity of the original face with a specified identity. How to find and modulate the identity representation is the crucial point to generating a face-swapped image. In addition to being used in film and television production, Face Swapping can also generate training data for fake face detection. Forgery detection techniques are used to determine whether the face in video or image is generated by face swapping methods. They all need the fake images generated by current face swapping methods to be training set. A new face swapping method, which can generate realistic faces, can also help the development of face forgery detection.

1.1 Natural Language Queries

Natural Language Queries (NLQ) is a new task proposed by the recently released Ego4D dataset [20]. Given an egocentric video clip and a text query, the goal of this task is to locate a temporal moment of the video clip where the neural network can obtain the answer to the query. NLQ is a complex task that needs to solve three visual tasks simultaneously: video understanding [65, 16, 15], multimodal representation learning [76, 76, 65] and video temporal location [74, 75, 72, 73, 38]. To tackle this task, we propose a multi-scale cross-modal transformer and a video frame-level contrastive loss to fully uncover the correlation between language queries and video clips. Besides, we propose two data augmentation strategies to increase the diversity of training samples. The experimental results demonstrate the effectiveness of our method. We elaborate on the detail of our solution on NLQ in Chapter 2.

1.2 Face Swapping

With the development of Generative Adversarial Network [19], the deep neural network can generate a photorealistic image. Various generative tasks have attracted the attention of the computer vision community; Face Swapping is one of them. Given a target image and a source image, Face Swapping aims to swap the identity of the target image to the identity of the source image while the other attributes of the target image (background, expression et al.) should be preserved. Previous works on Face Swapping [12, 7, 14, 39] perform well on face swapping for low-resolution facial images (lower than 256x256). And recently, Hififace [67] focused on transferring the face shape of the source image to a face-swapped image, which leads to better face-swapped results. In this thesis, we propose a new face swapping framework which utilizes facial Generative Adversarial Network [31, 32, 33] as generative prior. We choose StyleGAN2 [33] which can generate super-resolution (1024x1024) and photorealistic human face photos as our generative prior. The goal of using generative prior is to create high-fidelity and high-resolution face-swapped results by utilizing the knowledge from generative prior. Quantitative and qualitative experiments have shown that our method gets competitive results compared with state-of-the-art face swapping methods. We elaborate on the detail of our solution in Chapter 3.

1.3 Contribution

This thesis is organised as follows. In Chapter 2, we first present the background of Natural Language Queries. We propose a multi-scale cross-modal transformer and a video frame-level contrastive loss to uncover the correlation between video and text fully. We also presented two new data augmentations: variable-length sliding window sampling and video splicing, to collect more samples and avoid overfitting issues. Chapter 3 is about Face Swapping. Besides the background for image synthetic and the survey for the existing Face Swapping methods, we elaborated on our brand new framework for Face Swapping, which utilizes facial generative prior. In Chapter 4, I briefly summarize the thesis and future works for improvement.

Chapter 2

Multimodal Representation Learning for Natural Language Queries

2.1 Introduction

Natural Language Queries is an essential task from Ego4D [20]. Ego4D is a massive-scale egocentric video dataset. Unlike previous video datasets like HowTo100M [45] and MSRVTT [68] which provide third-person perspective video, wearable cameras capture all the videos from Ego4D like GoPro. We denote the video captured by a wearable camera which provides first-person perspective video as egocentric video. Egocentric video understanding has drawn significant attention. In many application scenarios, egocentric video understanding is required, such as augmented reality and service robots or human-computer interaction. These applications all need to handle egocentric video input. Compared with conventional video understanding, the development of egocentric video is relatively slow. The first is the lack of sufficient egocentric video datasets. Since there are not too many egocentric videos on the Internet, it is impossible to collect video data from video websites and label them accordingly, like conventional video datasets. Egocentric video datasets usually require many people to wear wearable devices to record their daily life, which is time-consuming and expensive. Secondly, the scene of the previous egocentric video dataset is limited. Most of the videos in the previous datasets were collected indoors, such as in the kitchen or living room, so the models trained on these datasets could

This chapter is based on our solution (Liu et al. 2022 [42]) for the Ego4D Natural Language Queries (NLQ) Challenge in CVPR 2022. Our solution got the first place award on this challenge.

not handle a variety of complex scenes.

The motivation of the Ego4D NLQ task is to locate a temporal moment which corresponds to a natural language query through an egocentric video. NLQ is a complex task and can be separated into three subtasks: video understanding, multimodal representation learning and video temporal location.

Figure 2.1 is an example to help understand this task. Given a text query: ”Where did I pick the cloth?” and a long egocentric video clip, the goal of the Ego4D NLQ task is to locate the moment span when the man in the video is picking up the blue cloth. There are two challenges to the Ego4D NLQ task: an extremely long duration time and a shortage of videos. First, the total duration of the video clips on Ego4D is extremely long, while the period of moments span represents a tiny percentage of the entire time. For example, the average duration of the video clip is up to 7.5 minutes, while the average time of the span is less than 5 seconds. The second issue is a shortage of videos. The data collection process of Ego4D is as follows. First, volunteers are recruited from all over the world. These volunteers come from all walks of life. These people need to wear a head-mounted camera to record their daily life. The data annotators will split each video into multiple key video clips and then mark each video clip with a text query. Here is how the video clip-text pairs are obtained. Concretely, the Ego4D NLQ training dataset has more than 10000 video clip-text pairs, but there are only about 1200 union video clips which is not enough to learn such a complex task as NLQ.

To alleviate both challenges, we propose a multi-scale cross-modal transformer making the video features interact with text features more adequately. In addition, video frame-level contrastive loss is introduced to enforce our model to focus on video frames that fall into a moment span. To solve the challenge of video shortage, we propose two data augmentation methods: variable-length sliding window sampling

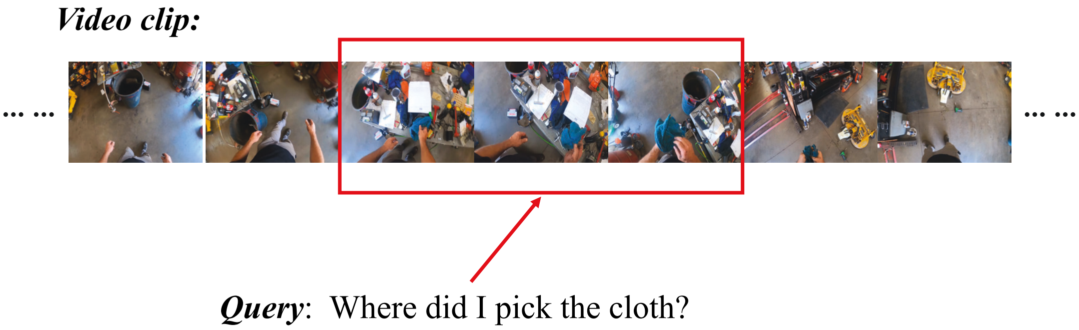


Figure 2.1 : An illustration of Natural Language Queries, the red window is the query corresponding temporal moment span.

(SW) and video splicing (VS), to collect more samples and avoid overfitting issues. Our method outperforms previous state-of-the-art methods and achieves the best performance on the test sets.

2.2 Related work

The NLQ task can be treated as a multi-modal retrieval task. There exist similar tasks, including moment retrieval [74, 75, 73], video highlight detection [38] and text-video retrieval [65, 76]. Moment DETR [38] is the SOTA method for moment retrieval. It follows the structure from DETR [4] which is built by transformer encoder blocks and transformer decoder blocks [63]. Moment DETR combines video and text input and sends them into transformer encoder blocks aiming to learn the relationship between these two modalities through the transformer. Moment Queries are learned embedding severed as the input for decoder blocks. Moment Queries are set to decode the fused features from transformer encoder blocks for obtaining the desired moment span. 2D-TAN [75] also gets competitive performance on moment retrieval. The idea of 2D-TAN is to exhaustively enumerate possible time segments, and then take the moment span with the highest correlation with the text as the final result. To be specific, 2D-TAN splits the whole video into time segments with even time duration. These time segments are combined into moment spans with different start times and different end times. 2D-TAN uses ConvNet the calculate the similarity between moment spans and text input. The moment span with the highest score is chosen as the answer for the text input. MS 2D-TAN [74] is a follow-up work of 2D-TAN. 2D-TAN is a time-consuming method because it needs to enumerate all the possible moment spans. The improvement of MS 2D-TAN is to perform exhaustive enumeration only on specified time duration scales, which greatly compresses the time complexity. Many previous works [74, 75, 72, 73, 38] aim to enhance the interactions among multiple knowledge representations from different modalities [70] on these tasks. However, the NLQ task is more challenging due to the long duration of videos and video shortage, and the text input is questions rather than statements. Our idea is to exploit a more efficient way for this task. We also use well-trained video representation [16, 15] and text representation [49, 11] to make the NLQ task easier. SlowFast is a video recognition method that is trained on large-scale video datasets such as Kinetics-400 [34], Kinetics-600 [5], Charades [58] and AVA [21].SlowFast is built by a slow pathway that captures spatial semantics and a fast pathway used to model the temporal relationship. A pre-trained SlowFast model can be severed as a video feature extractor due it can capture the relationship between different video frames. CLIP [49] is the SOTA and the most influential method of image-text retrieval. It demonstrates the benefits of large-scale text-image pre-training. CLIP uses two different feature encoders to get the image features and text features. Contrastive loss [23] is used to make the image and text features belonging to the same pair more similar, and at the same time make the image and text not belonging to a pair farther apart in the feature space.

2.3 Analysis of Moment DETR

Ego4D NLQ task is similar to the classical tasks: moment retrieval and temporal location. All three tasks use textual information to retrieve the period corresponding to the text in the video. We train the state of the art methods: VSLNet [73], 2D TAN [75], Ms 2D-TAN [74], and Moment DETR [38] from moment retrieval and temporal location on the Ego4D NLQ dataset. The result of these methods on the NLQ dataset is shown in the upper part of Table 2.1. But these methods do not perform well. After analysis, we find that Ego4d NLQ differs fundamentally from these two classical tasks. Both classical tasks use the text of declarative sentences to retrieve information from short videos, while Ego4D NLQ uses questions to retrieve information from the long egocentric video. Such an essential difference in tasks leads to the poor performance of these SOTA methods in Ego4D. A declarative sentence usually describes how an event occurs or the appearance of an object. This description-driven task is more about finding a key frame with a high enough similarity with the text to get the best answer. But the question sentence is to ask the cause or result of an event, the answer cannot be obtained directly from the interrogative sentence. It needs to be combined with the entire video to get the optimal result. The answer to this kind of question is usually found before or after the event, so the requirements for capturing the temporal relationship of the video are higher, and the relationship between the video and the text needs to be fully explored. Previous works did not fully uncover the correlation between video and text. That is why these works get unsatisfactory results.

After analysis, we found that the structure of Moment DETR [38] is very suitable for feature fusion between two different modalities. This is because Moment DETR follows the structure of DETR [4]. DETR is a SOTA detection method using Transformer [63] for information fusion. The powerful attention mechanism from

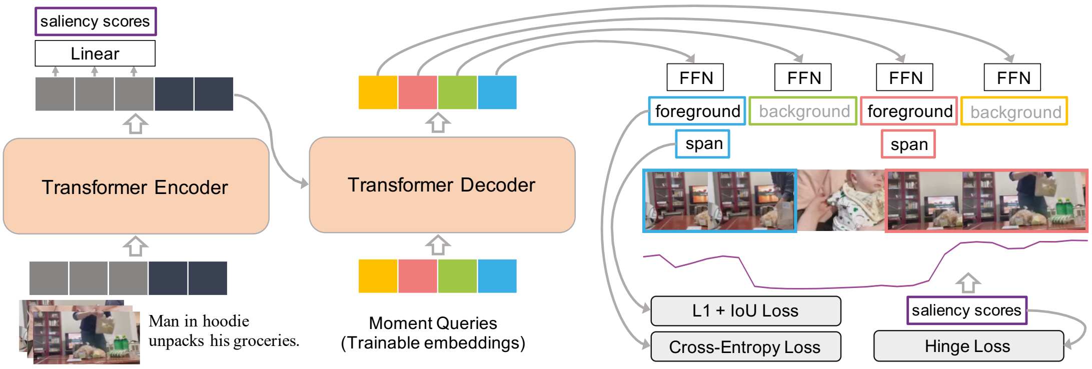


Figure 2.2 : Moment-DETR model overview. Moment DETR follows the pipeline from DETR [4]. The whole structure is built by transformer encoders and decoders. And there three prediction heads including saliency scores, foregroud/backgroud prediction, and span width & coordinate to assist with moment span forecasting.

Transformer uncovers the relationship between different modalities. Therefore, we analyze Moment DETR and propose a new framework that achieves SOTA performance. Next, we frist analyze why Moment DETR performs strugglingly on the Ego4d NLQ task. And we elaborate on our method in section 3.3.

Shortage of feature fusion part from Moment DETR. Figure 2.2 is the overview of Moment DETR. Moment DETR directly concatenates the video and text features along the sequence length dimension and sends the concatenated feature into a transformer encoder. We denote video features as $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}^\top \in \mathbb{R}^{m\times d}$ and text features as $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n\}^\top \in \mathbb{R}^{n\times d}$, where m and n are the sequence length of video feature and text feature. After concatenation, we have concatenated feature $\boldsymbol{F}_c = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_{m+n}\}^\top \in \mathbb{R}^{(m+n)\times d}$. The transformer encoder consists of a self-attention layer and a fully connected feed-forward network. The key component of the transformer encoder is self-attention. The self-attention function is used to reweigh the input feature by comparing every element along the sequence dimension. The self-attention role helps capture the contextual relevance inside the input

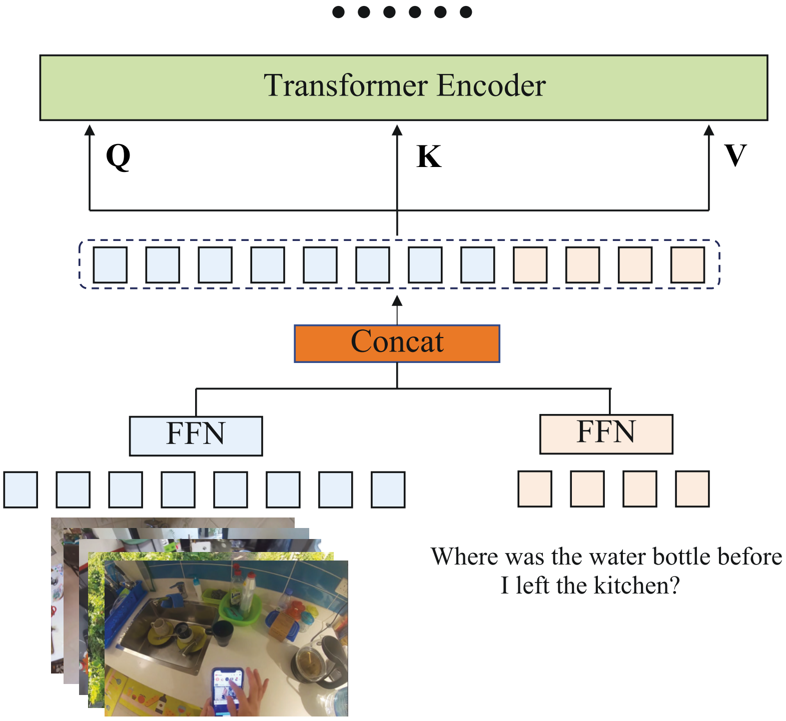


Figure 2.3 : Overview of how Moment DETR [38] does information fusion.

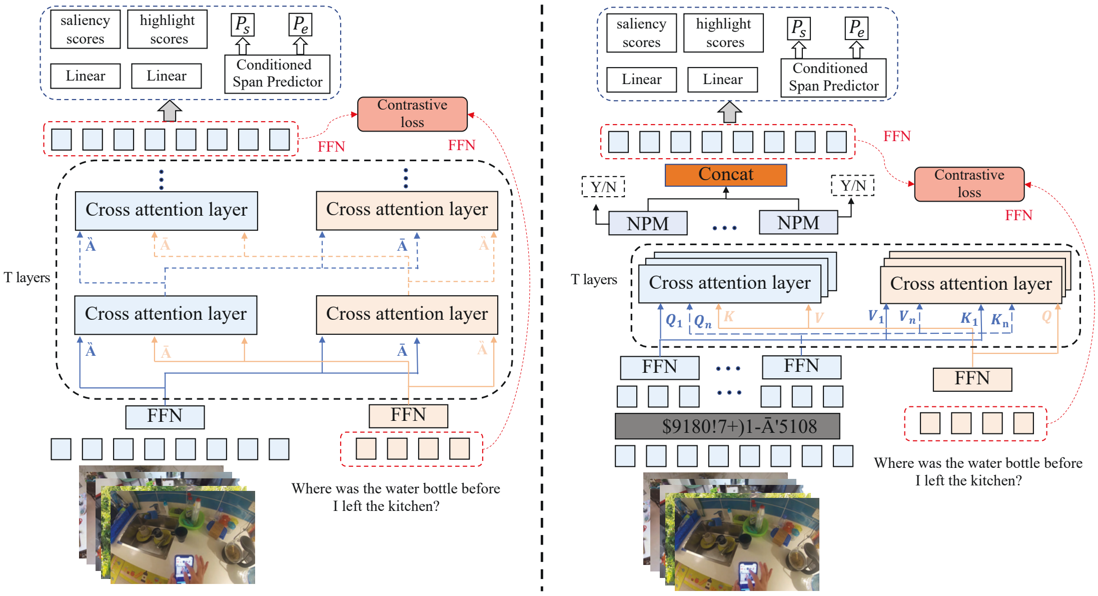
feature. Self-attention has the following two shortcomings on the Ego4D NLQ task. 1) It cannot fully uncover the correlation between the two modalities, w.r.t video and text. 2) It cannot handle extremely long sequence features.

Figure 2.3 is a detail illustration of how Moment DETR does feature fusion. In Moment DETR, queries $Q_c \in \mathbb{R}^{(m+n)\times d}$, keys $K_c \in \mathbb{R}^{(m+n)\times d}$, and values $V_c \in \mathbb{R}^{(m+n)\times d}$ are obtained from the same input F_c by using three different linear layers. The output is calculated as follows:

$$\text{Attention}(Q_c, K_c, V_c) = \text{softmax}(\frac{Q_c K_c^T}{\sqrt{d}})V_c \tag{2.1}$$

F_c consists of both video modality and text modality. We denote f_i and f_j is the i-th element and the j-th element of F_c. In self-attention, each query gives much higher weights for the values from the same modality than that from different modalities. For example, a video query will apply higher attention weights to video values while the attention weights for text values are relatively low. Due to the attention weights for those that belong to different modalities being low, the attention output of f_i is approximately the weighted sum of all f_j, which came from the same modality as f_i. In other words, the attention from Moment DETR is hard to uncover the relationship between video and text. Such an attention way could not meet NLQ's goal, which utilizes the relationship between text and video to retrieve the target moment span. Additionally, time complexity of each self attention layer is $O((m+n)^2 \cdot d)$. The computation cost becomes unaffordable when handling extremely long video or extremely long text queries.

Shortage of the prediction way from Moment DETR. Moment DETR follows the pipeline from DETR [4]. The transformer decoder takes a moment query $M_q \in \mathbb{R}^{N \times d}$ as input and output embedding $O_{dec} \in \mathbb{R}^{N \times d}$ by utilizing the fusion feature from transformer encoder, where N is the number of moment queries. A 3-layers multilayer perceptron network (MLP) with Relu activation is applied on O_{dec} to predict the normalized moment span center and normalized moment span width w.r.t the input video length. The format of original moment span is $[c, w]$ where c and w are center value and width value. Normalized moment span is formulated as $[c/L_v, w/L_v]$ where L_v is the length of input video length. And the unit of c, w, and L_v is second (s). The prediction of the center and the width for the moment span is unconstrained. Such a prediction way could produce an unreasonable or extremely wrong result which both lead to low evaluation performance. Such an unconstrained prediction way which directly predicts the moment span center and moment span width, is hard to get an accurate result. In the next section, we elaborate on our methodology, which solves the above shortages of Moment DETR.



(a) Single-scale Cross-modal Transformer

(b) Multi-scale Cross-modal Transformer

Figure 2.4 : The overall framework of our approach. (a) depicts our single-scale cross-modal transformer. (b) shows the details of our multi-scale cross-modal transformer.

2.4 Methodology

We propose a multi-scale cross-modal transformer built by T cross-attention layers, as shown in Figure 2.4. Then, we build a saliency scores predictor [38], a highlight region predictor [38], and a conditioned span predictor [73] upon the backbone. The prediction output from the conditioned span predictor is our final result. Additionally, we utilize pre-extracted video and text features as inputs of the backbone. To improve the generalization ability of our method, two data augmentation methods are adopted during training, including video splicing (VS) and variable-length sliding window sampling (SW).

2.4.1 Input preparation

We use Slowfast features [15] and Omnivore features [16] which are both pre-trained on Kinetic-400 [34] provided by Ego4D developers as video features. Specifically, Slowfast uses window size 32 and temporary stride 16 to extract features (roughly two frames of Slowfast features per second for 30-fps videos). In addition, Omnivore uses window size 32 and temporary stride 6 to extract features (roughly five frames of Omnivore features per second for 30-fps videos). We introduce CLIP [49] feature to improve cross-modal representation learning. For each frame of video features, we randomly select one of its input frames and then feed it to the image encoder (ViT-B/16) of CLIP to get the CLIP visual feature. The video and CLIP visual features are concatenated along channel dimension as the final video input. Text input is obtained by the CLIP text encoder. Instead of taking the EOS token as an aggregate text representation, we reserve text sequence length to use text token-level information. We denote video features as $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\}^\top \in \mathbb{R}^{m \times d}$ and text features as $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n\}^\top \in \mathbb{R}^{n \times d}$, where m and n are the sequence length of video feature and text feature.

2.4.2 Multi-scale Cross-modal Transformer

Cross-attention mechanism. Single-scale cross-modal and multi-scale cross-modal transformer are built by a stack of T cross-attention layers shown in Figure 2.4. We set $T = 3$ by default. The structure of the cross-modal transformer is the same as that of the standard transformer encoder block [63], including a multi-head attention layer and a position-wise fully connected feed-forward network. To fully uncover the correlation between video and text, we use a cross-attention mechanism in a cross-modal transformer instead of the self-attention mechanism used by the standard transformer encoder block. Query, key, and value are all obtained from the same input through three linear layers in the self-attention mechanism.

However, the cross-attention mechanism exchanges the key-value pairs of different input modalities for attention operation, as shown in Figure 2.4 (a). To be specific, video queries $Q_v \in \mathbb{R}^{m\times d}$, video keys $K_v \in \mathbb{R}^{m\times d}$, and video values $V_v \in \mathbb{R}^{m\times d}$ are obtained from video input feature $\boldsymbol{V}$ by three linear layers. We get text queries $Q_t \in \mathbb{R}^{n\times d}$, text keys $K_t \in \mathbb{R}^{n\times d}$, and text values $V_t \in \mathbb{R}^{n\times d}$ from text input feature $\boldsymbol{T}$ by another three linear layers. We build two attention input pairs for attention computation by switching the key-value pairs from video and text: $[Q_v, K_t, V_t]$ and $[Q_t, K_v, V_v]$. These two input pairs are fed into two standard attention blocks separately. As a result, we get the attention that has been language-conditioned in the video stream (Eq. 2.2) and attention that has been video-conditioned in the linguistic stream(Eq. 2.3). Cross attention mechanism ensures that each video feature interacts with text features independently. We call the output feature of the cross-modal transformer from the video stream a cross-modal feature. Cross-modal features are then sent to prediction heads for final forecasting.

$$\text{Attention}(Q_v, K_t, V_t) = \text{softmax}(\frac{Q_v K_t^T}{\sqrt{d}})V_t \tag{2.2}$$

$$\text{Attention}(Q_t, K_v, V_v) = \text{softmax}(\frac{Q_t K_v^T}{\sqrt{d}})V_v \tag{2.3}$$

Moreover, the cross-attention layer is an efficient module. The time complexity of each cross-attention layer is $O(m \cdot n \cdot d)$. The length of video features is far longer than the text features' length. In our case, the average length of the video feature is 600, while the average length of the text feature is 32. Hence, the $O(m \cdot n \cdot d)$ is much lower than $O((m+n)^2 \cdot d)$ which is the time complexity of each self attention layer as we mention in section 2.3. Our method can handle extremely long video and text queries with cross-attention.

Multi-scale mechanism. We build our multi-scale cross-modal transformer by adopting multi-scale split-and-concat strategy from VSLNet-L [72] as shown in Figure 2.4 (b). Here, we summarize the fundamental idea of this strategy below. This strategy equally splits a video into K video segments: $\boldsymbol{V} = [\boldsymbol{V}_1, \ldots, \boldsymbol{V}_K]$. To improve the model's ability to handle video of various lengths, K is a random sample at each data sample step instead of being a fixed hyperparameter. Additionally, random sampled K could provide various data samples during training. Each video segment $\boldsymbol{V}_k = \{\boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2}, \ldots, \boldsymbol{v}_{k,l}\}^\top \in \mathbb{R}^{l \times d}$ along with the whole text query are fed to the cross-modal transformer separately and output cross-modal feature $\boldsymbol{C}_k = \{\boldsymbol{c}_{k,1}, \boldsymbol{c}_{k,2}, \ldots, \boldsymbol{c}_{k,l}\}^\top \in \mathbb{R}^{l \times d}$. Each cross-modal feature $\boldsymbol{C}_k$ is then processed by Nil Prediction Module (NPM) [72] and produces a score $\mathcal{S}^k_{npm}$, which indicates the confidence of video segment $\boldsymbol{V}_k$ overlaps with query corresponding moment span. NPM is built by an attention layer, a linear layer, and a sigmoid activation. The output score $\mathcal{S}^k_{npm}$ from NPM is computed as:

$$\begin{aligned} \boldsymbol{w}_k &= \texttt{SoftMax}(\texttt{Conv1x1}(\boldsymbol{C}_k)) \\ \boldsymbol{A}_k &= \sum\nolimits_{i=1}^{l} w_{k,i} \cdot \boldsymbol{c_{k,i}} \\ \mathcal{S}^k_{npm} &= \sigma(\texttt{FFN}(\boldsymbol{A}_k)) \end{aligned} \tag{2.4}$$

Cross entropy is used as the loss function for Nil Prediction Module:

$$\mathcal{L}_{NPM} = f_{CE}(\mathcal{S}_{npm}, Y_{npm}) \tag{2.5}$$

We denote cross entropy as f_{CE} and Y_{npm} is the ground true label of S_{npm}. Y^k_{npm} is 1 if the k-th video segment has an intersection with the target moment span. Y^k_{npm} is 0, while the k-th video segment does not has any intersection with the target moment span. All features $\boldsymbol{C}_k$ are re-weighted by $\mathcal{S}^k_{npm}$ and produce $\bar{\boldsymbol{C}}_k$:

$$\bar{\mathcal{S}}^k_{npm} = \frac{\mathcal{S}^k_{npm}}{\max(\mathcal{S}_{npm})}, \quad \bar{\boldsymbol{C}}_k = \bar{\mathcal{S}}^k_{npm} \times \boldsymbol{C}_k \tag{2.6}$$

All $\bar{\boldsymbol{C}}_k$ are concatenated into $\bar{\boldsymbol{C}}_{final}$ along the sequence dimension. In the end, we send $\bar{\boldsymbol{C}}_{final}$ to prediction heads. For computational efficiency, the split and concatenation are operated at the feature level instead of the original video.

2.4.3 Video Frame-level Contrastive Loss

The goal of the Ego4D NLQ task is to locate the moment span using text information. The similarity between text features and video frame features belonging to the moment span should be higher than the similarity between text features and video frame features that fall out of the moment span. Therefore, we introduce video frame-level contrastive loss. The similarity calculation function between the video frame feature and text embedding is as follows:

$$F(v, T) = \frac{\sum_{\boldsymbol{t_j} \in T} v \cdot t_j / \tau}{|T|}, \tag{2.7}$$

where $v \in \mathbb{R}^{1 \times d_v}$ is the single video frame from the whole video clip sequence, and $T \in \mathbb{R}^{L_t \times d_t}$ is the whole text embedding sequence ($t_j \in \mathbb{R}^{1 \times d_t}$), τ is the temperature hyper-parameter. We set $\tau = 0.07$ by default. The loss function is formulated as follows:

$$\mathcal{L}_i^{\text{NCE}} = \frac{1}{|\mathcal{P}_i|} \sum_{\boldsymbol{v_i^+} \in \mathcal{P}_i} - \log \frac{\exp(F(\boldsymbol{v_i^+}, \boldsymbol{T_i}))}{\exp(F(\boldsymbol{v_i^+}, \boldsymbol{T_i})) + \sum_{\boldsymbol{v_i^-} \in \mathcal{N}_i} \exp(F(\boldsymbol{v_i^-}, \boldsymbol{T_i}))}, \tag{2.8}$$

Where $\mathcal{P}_i$ and $\mathcal{N}_i$ denote video embedding collections of the positive and negative frames of i^{th} video-text pair. A video frame feature is a positive sample if it falls into the moment span. In contrast, a video frame feature that falls out of the moment span is a negative sample. The total contrastive loss is as follows:

$$\mathcal{L}_{\text{NCE}} = \sum_i \mathcal{L}_i^{\text{NCE}}, \tag{2.9}$$

2.4.4 Prediction heads

To estimate the target moment span, we use the conditioned span predictor and highlight predictor from VSLNet [73], and we also use the saliency predictor following Moment DETR [38].

The conditioned span predictor is constructed with transformer encoder layers and linear layers to predict the start and end boundary of the moment span:

$$\begin{aligned}
\boldsymbol{h}^s &= \texttt{Transformer}(\boldsymbol{C}) \\
\boldsymbol{h}^e &= \texttt{Transformer}(\boldsymbol{h}^s) \\
\boldsymbol{P}_t^s &= \boldsymbol{W}_s([\boldsymbol{h}_t^s; \boldsymbol{C}_t]) + \boldsymbol{b}_s \\
\boldsymbol{P}_t^e &= \boldsymbol{W}_e([\boldsymbol{h}_t^e; \boldsymbol{C}_t]) + \boldsymbol{b}_e
\end{aligned} \tag{2.10}$$

$\boldsymbol{S}_t^s$ and $\boldsymbol{S}_t^e$ is the logit score of being start and end boundary in position t. $\boldsymbol{C}$ denote the cross-modal feature produced by our cross-modal transformer. $P^s = \texttt{Softmax}(S^s) = \{\boldsymbol{p}_1^s, \boldsymbol{p}_2^s, \ldots, \boldsymbol{p}_m^s\}$ and $P^e = \texttt{Softmax}(S^e) = \{\boldsymbol{p}_1^e, \boldsymbol{p}_2^e, \ldots, \boldsymbol{p}_m^e\}$ are the probability of start and end boundary. The loss function for conditioned span predictor is formulated as:

$$\mathcal{L}_{span} = \frac{1}{2}\big[f_{CE}(P_s, Y_s) + f_{CE}(P_e, Y_e)\big] \tag{2.11}$$

Y_s and Y_e are one-hot labels for the start and end boundary. During inference, we compute the joint probability distribution and take $[t_p^s, t_p^e]$ with maximum probability as the final prediction:

$$\begin{gathered}
[t_p^s, t_p^e] = \arg\max_{t_p^s, t_p^e} P_s(t_p^s) P_e(t_p^e) \\
\text{s.t. } 0 \leq t_p^s \leq t_p^e \leq n
\end{gathered} \tag{2.12}$$

The saliency predictor and highlight predictor are both built with two linear layers. The highlight predictor to predict which video frame feature falls into the target moment span belongs to predict region information. Here is the loss function for the highlight predictor:

$$\mathcal{L}_{\text{hl}} = f_{CE}(S_{hl}, Y_{region}) \tag{2.13}$$

S_{hl} is the output of the highlight predictor and Y_{region} 0-1 label. Y_{region}^t is 1 if the t-th video frame falls into the target moment span and 0 otherwise. The saliency loss ensures the saliency score of a video frame that falls into the target moment span should be higher than that falls out of the span. Hinge loss is used to achieve this goal:

$$\mathcal{L}_{sa} = \max(0, \Delta + S_{sa}(t_{\text{out}}) - S_{sa}(t_{\text{in}})). \tag{2.14}$$

We denote the output from the saliency predictor is S_{sa}. t_{in} is a clip that falls into the target moment span, and t_{out} is a clip that falls out of the target moment span. The total loss of our method is shown below:

$$\mathcal{L} = \mathcal{L}_{\text{span}} + \mathcal{L}_{\text{hl}} + \mathcal{L}_{\text{NPM}} + \mathcal{L}_{\text{sa}} + \mathcal{L}_{\text{NCE}}, \tag{2.15}$$

2.4.5 Data augmentation

Even though the Ego4D NLQ training dataset has more than 10000 video clip-text pairs, there are only about 1200 original video which is not enough to learn such a complex task as NLQ. To get more video clip data to facilitate convergence and avoid overfitting issues, we design a new data augmentation method by inserting positive clips into null video clips. Positive clips are sampled from a long video with variable background padding to increase diversity. This approach is a combination of two primary methods: variable-length sliding window sampling strategy (SW) and video splicing strategy (VS), as shown in Figure 2.5.

Variable-length sliding window sampling strategy. Inspired by MS 2D-TAN [74], we propose a variable-length sliding window sampling strategy to get more positive clips during training, as shown in Figure 2.5 (a). Specifically, we define a length ratio interval [r_s, r_e]. Suppose we sample a video V whose length is l_v. Then we will randomly sample a ratio $\hat{r}$ from the length ratio interval ($r_s \leq \hat{r} \leq r_e$). The sliding window size is equal to $\hat{r} * l_v$. We use this sliding window to generate positive clip V_p from the video V, and we ensure that the generated positive clip contains the whole query corresponding moment span.

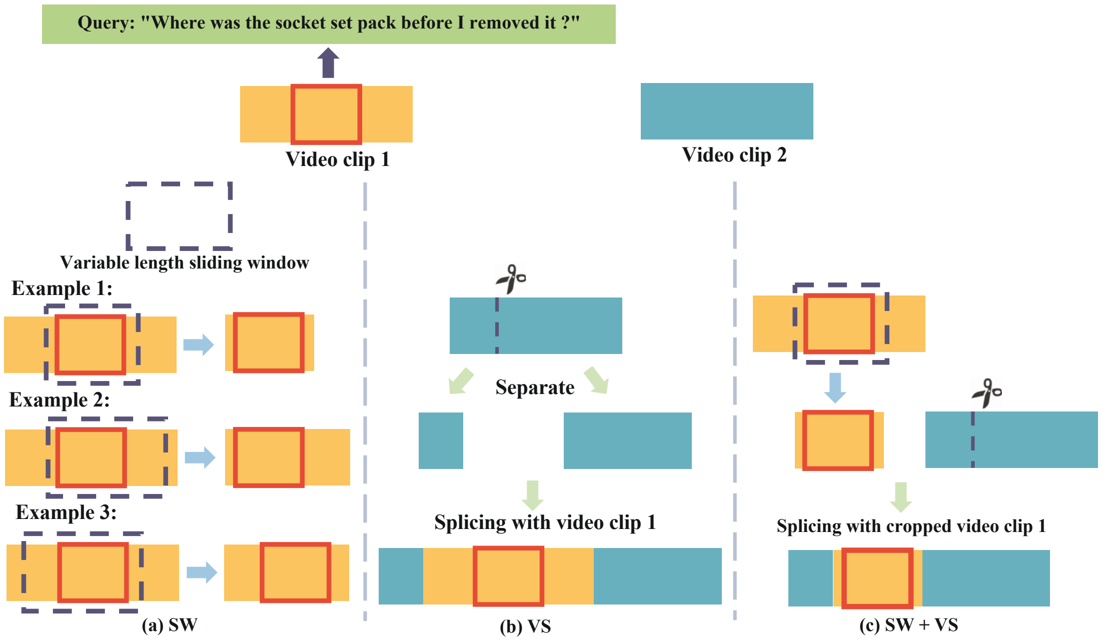

Query: "Where was the socket set pack before I removed it ?"
Video clip 1
Video clip 2
Variable length sliding window
Example 1:
Example 2:
Example 3:
(a) SW
Separate
Splicing with video clip 1
(b) VS
Splicing with cropped video clip 1
(c) SW + VS

Figure 2.5 : Illustration of data augmentation. (a) shows how the variable-length sliding window sampling strategy (SW) works. (b) shows how the video splicing strategy (VS) works. (c) is a combination of these two data augmentations which leads to better performance.

Video splicing strategy. Another data augmentation method is to insert one video clip into a null video clip, as shown in Figure 2.5 (b). Specifically, we sample two videos V_1 and V_2 each time. We randomly select a cut-in position on V_2, divide the video into two parts V_{21} and V_{22}, and place V_{21} and V_{22} on the head and tail of V_1 respectively to generate a new video clip. There is a hyper-parameter called splicing probability P_{vs} to control whether to splice V_2 and V_1 together for this sampling.

Drawback of video splicing strategy. Due to V_1 and V_2 being randomly sampled, the scenes or activity of these two videos could be different. Splicing these two videos could cause the continuity of the video to be broken, especially in the splicing part. And since the data augmentation method is only used in the training phase, there is another problem there will be a domain gap between the training data and the test data. A video usually has some black frames at the beginning and the ending. We call this kind of frame a meaningless blank frame. These empty frames will only appear at the beginning and end of the regular video. When the splicing method is used for training data, there will be a black screen outside the beginning and end of the video. However, the test set does not have this phenomenon, resulting in a domain gap between the training data and the test data, eventually leading to poor performance on the test set.

For the destruction of time continuity, using a video splicing strategy is inevitable to weaken the time continuity. However, the egocentric video has sudden changes in the environment and activities, but the degree of sudden changes is not so obvious. This is due to the nature of the egocentric video. We can assume that egocentric video is how we observe things in our daily lives. For example, I am currently at my desk looking at the computer, but the next step I might pick up the water glass and go to the water dispenser next to the desk and start collecting water. This change is very rapid in our daily life, and it may not take a second. In the egocentric video, such scene and activity changes are commonplace. Although video splicing does cause continuity to be broken, the actual impact is not particularly large. A video stitching strategy can be a trade-off between temporal continuity and data scale.

Combination of these two data augmentation. We combine these two methods as our final data augmentation method, as shown in Figure 2.5 (c). In the experiments section 2.5, we observe that combining these two methods gains better performance than using any of them alone. Similarly, we first sample two videos V_1 and V_2. Moreover, we adopt the variable-length sliding window sampling strategy for V_1 to obtain the positive clip V_{1p}, and then utilize the video slicing strategy for V_{1p} and V_2 to achieve the final video clip. We set the length ratio interval to [0.4,0.8] and set the splicing probability P_{vs} to 0.5. We found this to be the best value for these two hyper-parameters. It is worth noting that these data augmentation

Method	IoU=0.3		IoU=0.5	
	R@1	R@5	R@1	R@5
2D-TAN [75]	5.04	12.89	2.02	5.88
VSLNet [73]	5.45	10.74	3.12	6.63
MS 2D-TAN [74]	7.05	**14.15**	4.75	**9.16**
Moment DETR [38]	4.52	8.03	1.99	3.33
Ours-variant (self-attention)	6.53	11.02	4.05	7.59
Ours-base	7.69	11.51	4.83	7.8
+SW	9.06	11.36	5.68	7.25
+VS	7.38	10.66	4.31	6.84
+SW and VS	9.96	12.55	6.3	8.34
+SW, VS, and Contra	**10.79**	13.19	**6.74**	8.85

Table 2.1 : Performance of different methods on the val set.

strategies are only used during the training stage.

2.5 Experiments

All the experiments run on a single NVIDIA Tesla V100 GPU. Unless otherwise specified, the default video features are Slowfast features. We implement previous state-of-the-art methods: VSLNet [73], 2D TAN [75], Ms 2D-TAN [74], and Moment DETR [38] on the Ego4D NLQ dataset for comparison. We denote our multi-scale cross-modal transformer as Ours-base. We also mark Ours-base with variable-length sliding window sampling (SW), video splicing (VS), and video frame-level contrastive loss (Contra) as Ours-full.

The comparison results are shown in Table 2.1. Ours-full outperforms all state-

Method	IoU=0.3		IoU=0.5	
	R@1	R@5	R@1	R@5
Ours-full-slowfast	10.79	13.19	6.74	8.85
Ours-full-omnivore	10.74	13.47	6.87	8.72
Ensemble	**11.33**	**14.77**	**7.05**	**8.98**

Table 2.2 : Performance of our method with different video input feature on the val set.

of-the-art methods on R1@0.3 and R1@0.5. To verify the effectiveness of the cross-attention mechanism, we replace the cross-attention mechanism on Ours-base with the self-attention mechanism as Moment DETR and denote it as Ours-variant (self-attention). In Moment DETR, features of texts and video are concatenated along the sequence dimension before a self-attention operation. Compared with Ours-base, the performance of the Ours-variant on all the metrics is degraded. It shows that using a cross-attention mechanism to interact video features with textual features explicitly can improve localization performance on Ego4D.

After adding a variable-length sliding window sampling strategy, the performance improved by 1.3% on R1@0.3 without significant improvement in other metrics. When we use the video splicing strategy, the performance has not improved, even worse. When we use the two data augmentation together, the performance on four metrics is boosted by 2.27%, 1.04%, 1.47%, and 0.54% compared with Ours-base. If we add video frame-level contrastive loss to this setting, the performance will reach the highest, and the four indicators are improved by 3.1%, 1.68%, 1.91%, and 1.05% compared to Ours-base, respectively.

As can be seen from Table 2.1, Ours-full can achieve the best performance in

Method	IoU=0.3		IoU=0.5	
	R@1	R@5	R@1	R@5
Ensemble	12.89	15.41	8.14	9.94

Table 2.3 : Performance of our ensemble model on test set.

Participant team	IoU=0.3		IoU=0.5		Mean
	R@1	R@5	R@1	R@5	R@1
Ours	**12.89**	**8.14**	15.41	9.94	**10.51**
EgoVLP	10.46	6.24	16.76	**11.29**	8.35
MSRA AIM3 teams	10.34	6.09	**18.01**	10.71	8.22
Teamretrival	9.94	5.72	17.48	10.21	7.83
Tianti	9.24	5.24	16.36	9.82	7.24
TarHeels	6.42	3.55	10.46	6.39	4.98
Host Team(VSLNet)	5.42	2.75	8.79	5.07	4.08

Table 2.4 : Test set performance compared with other participating teams. Column Mean-R@1 means the average value of Iou=0.3-R@1 and Iou=0.5-R@1.

Slowfast input. In addition, Ego4D provides video features extracted from two models: Slowfast and Omnivore. As shown in Table 2.2, Ours-full has a similar performance on the val set with these two different features as input. However, the ensemble result has improved. That is, Ours-full-omnivore and Ours-full-slowfast are complementary. The ensemble strategy here is straightforward. These two models output top5 results according to their prediction score (the format of the result is (start time, end time, score)), so there are ten results. We sort these ten results according to the score value and take the top5 as the final result. For the final submission, we train Ours-full-slowfast and Ours-full-omnivore on the combination of the train set and val set. The test set performance of our ensemble model achieves the best performance on R1@0.3 and R1@0.5 and competitive result on R5@0.3 and R5@0.5 as shown in Table 2.3.

Table 2.4 summarizes the test set results of all the participating teams of Ego4D Natural Language Queries (NLQ) Challenge in CVPR 2022. We denote our ensemble model as Ours. Our method outperforms all teams on R@1 metrics, including Iou=0.3-Mean, Iou=0.3-R@1, and Iou=0.5-R@1.

Chapter 3

High-fidelity Face Swapping with Generative Facial Prior

3.1 Introduction

Given a target image and a source image, Face Swapping aims to swap the identity of the target image to the identity of the source image while the other attributes of the target image (background, expression et al.) should be preserved. Face swapping has attracted significant attention from the computer vision community. Face Swapping is quite topical, and it can be used for entertainment. But it is likely to be used for evil purposes, such as using this method to forge a fake video to defraud. However, instead of knowing nothing about this technology, it is better to explore the mystery and find a way to distinguish the authenticity of that face-swapped video. Forgery detection techniques are used to determine whether the face in video or image is generated by face swapping methods. They all need the fake images generated by current face swapping methods to be training set. A new face swapping method, which can generate realistic faces, can also help the development of face forgery detection.

Generative Adversarial Network (GAN) such as StyleGAN [32], BigGAN [3] and PGGAN [31] can produce photorealistic facial image. The image produced from pre-trained facial GAN has detailed texture and super-resolution. Inspired by GFP-GAN [66], we proposed a new framework for Face Swapping by utilizing generative facial prior. Generative facial prior means the well-trained facial generative model which can produce the photorealistic facial image. With the help of generative fa-

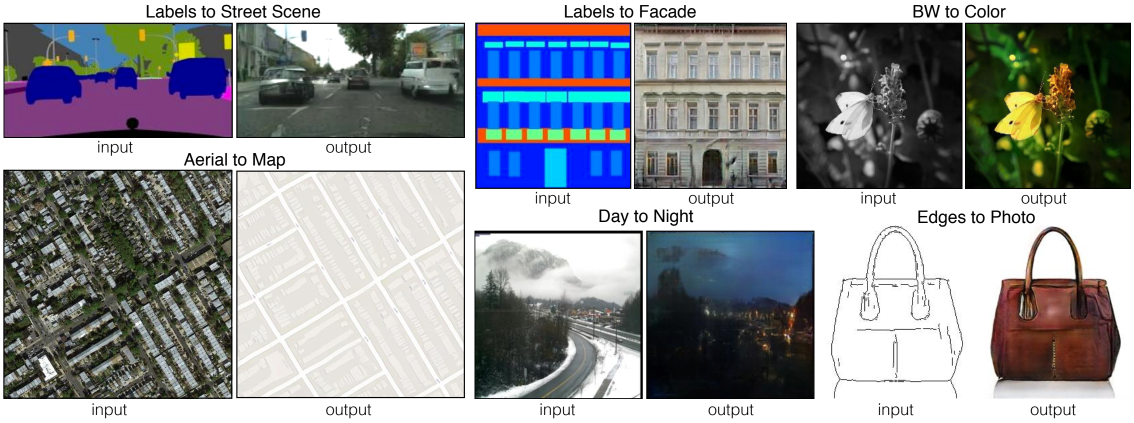


Figure 3.1 : This is the legend provided by the original pix2pix paper. Pix2pix show that conditional GAN has the ability to complete translation between various domains

cial prior, the model can focus on the identity swapping between the source person and the target person. Our framework consists of a pre-trained facial generative prior and an ID injection module. These two modules are connected by Channel-Split SFT (CS-SFT). We proposed an ID modulation block for identity swapping in ID injection module. Quantitative and qualitative experiments have shown that our method gets competitive results compared with state-of-the-art face swapping methods.

3.2 Literature review

3.2.1 image-to-image translation

Face Swapping is a sub filed of image-to-image translation [28, 78, 8, 48]. CycleGAN [78] uses two generator and discriminator pairs to achieve the image-to-image translation between two different domains. Pix2pix [28] proposes a conditional GAN to utilize additional condition input to control the generated output image. SPADE [48] proposes a new spatial modulation module: Spatially-adaptive normalization to achieve condition injection. With spatially-adaptive normalization,

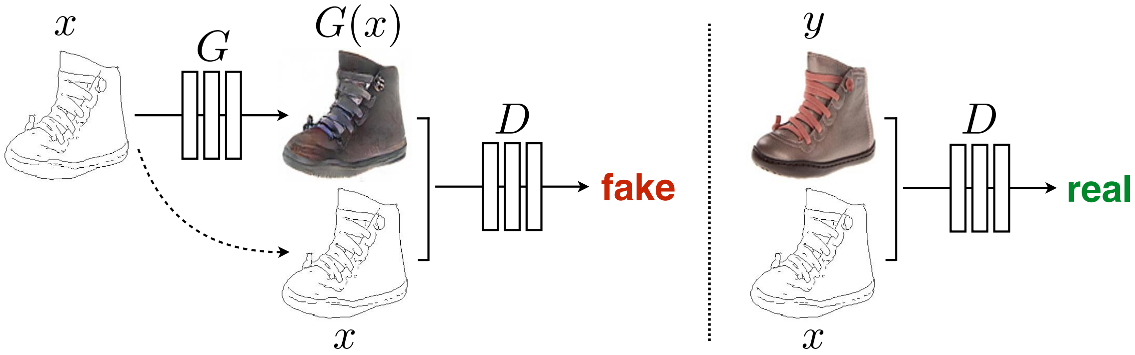


Figure 3.2 : Taking edge to photo as an example, the training purpose of D is to identify whether the input is from real data or fake data.Training purpose of G hopes that the quality of the generated image can make D judge it as coming from real data.

generated fake images can well preserve the semantic information from input. StarGAN [8] is the first method to achieve multiple-domain image-to-image translation within a single model. StarGAN accepts the image and target domain label as generator input and outputs the translated image within the target domain. There are two goals for the discriminator from StarGAN: 1) Determining whether the input image is real or fake. 2) Domain classification. Next, we will elaborate on Pix2pix to help understand the concept of image-to-image translation.

Pix2pix is the most influential paper on image-to-image translation that applies GAN to supervised image-to-image translation. Supervised means that training data is paired, and image-to-image translation means image-to-image mapping. It is the process of producing the required output picture from an input image, which can be seen as a type of image-to-image mapping. For example, image restoration and super-resolution are two populate and practical subtasks of image-to-image translation.

Previous image-to-image translation methods are based on conventional GAN.

Conventional GAN takes a random sampled noise from a predefined distribution as input and outputs a generated output. Due to only random sampled noise being used as input, we could not specify any attribute of the rendered image, including shape, color, and texture. The methods based on conventional GAN add additional constraints on the loss function side to guide image generation. Pix2pix proposed conditional GAN (cGAN) to achieve image-to-image translation. cGAN guides image generation by adding conditional information. Instead of just using random sampled noise as input, cGAN utilizes extra conditional input to steer picture synthesis and gets specified generated images. The form of condition input is unrestricted; it could be edge, segmentation map and labels. As long as a paired dataset is constructed, any type of feature or image can be used as a conditional input. Figure 3.1 demonstrates six subtasks of image-to-image translation by using different types of conditional information, including the processes from label to image generation, image edge to image generation and so on.

The pipeline of the pix2pix is shown in Figure 3.2. As shown in the figure, image generation, which takes an edge photo as input is taken as an example to introduce the workflow of pix2pix. First, the edge map is represented by x, and the ground truth image is denoted as y, which is the corresponding image representation of x. Pix2pix requires paired images (x and y) during training. Edge map x and random noise z are used as the input of generator G. Random noise z is not shown in the figure. Removing z does not have much impact on the generation effect, but if x and z are combined as the input of G, more variety can be obtained. $G(x)$ is the generated fake image. Discriminator D takes fake or real data as input to get the predicted probability value. This value is used to determine the probability that the input belongs to real data distribution. The closer this value is to 1, the discriminator considers that the input has a greater probability of coming from real data. In addition, the real images y and x are also merged based on the channel

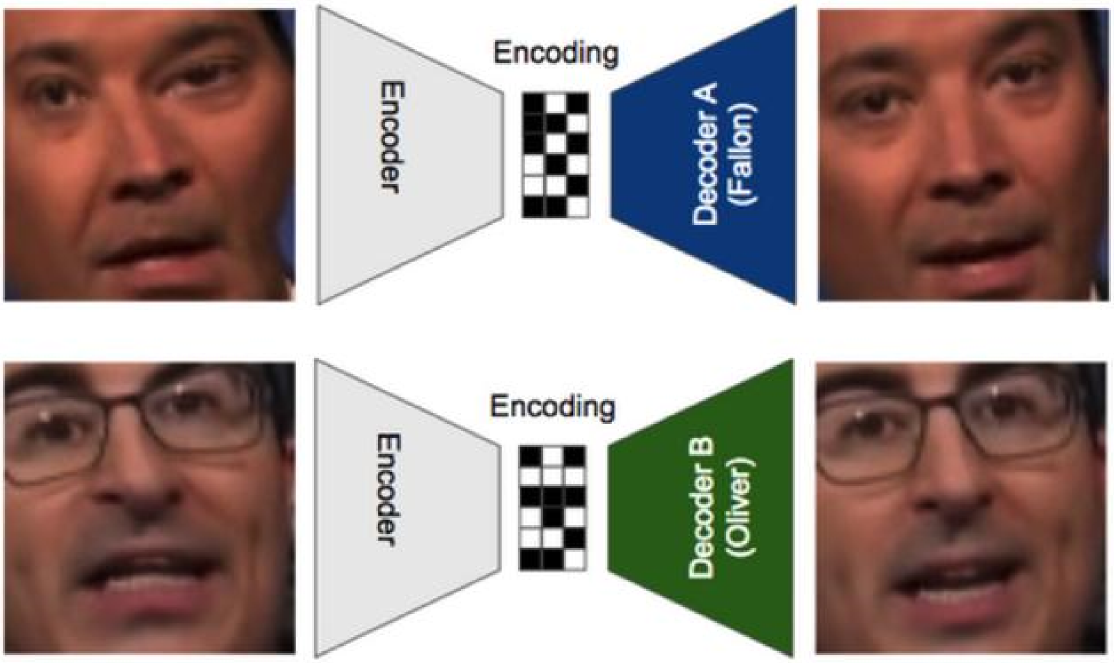


Figure 3.3 : The pipeline and structure of Deepfake

dimension and used as the input of the discriminator D to obtain the probability prediction value. Therefore, the training goal of the discriminator D is to output a small probability value (the minimum value is 0) when the input is a pair of generated fake image and edge map and output an enormous probability value when the input is a pair of real image and edge map. The training purpose of the generator G is to make the generated $G(x)$ and x as the input of the discriminator D, the probability value of the output of the discriminator D is as large as possible, which is equivalent to successfully deceiving the discriminator D.

3.2.2 Existing face swapping method

Given a target image and a source image, Face Swapping aims to swap the identity of the target image to the identity of the source image while the other attributes of the target image (background, expression et al.) should be preserved. There are two types of face swapping methods: subject-specific and subject-agnostic.

Subject-specific means that the model should train in the image set, which includes a large number of images from both the target person and source person. There are many excellent works [12, 55, 54, 40, 29, 69, 36, 77] focus on improving the performance and visual effect about the subject-specific method.

DeepFake [36] is the first to propose the technology of Face Swapping. The pipeline and structure of Deepfake are shown in Figure 3.3. The architecture of DeepFakes consists of an encoder to compress a human face into a latent space, as well as two decoders, A and B. Decoder A is supposed to restore the human character A (Fallon) and decoder B is used to restore the human nature B (Oliver).

After training, when Jimmy's face image is fed to the combination of encoder and decoder B, we can get a new face image replacing Jimmy's identity with Oliver's. However, Subject-specific can not apply widely. For example, a decoder trained for LeBron James can not achieve the goal of swapping faces with Kobe Bean Bryant's identity. In other words, a person for a trained decoder. To achieve the purpose of face swapping with three different people's identities, three decoders need to be prepared separately. It also required many images of the specific person to train the face swapping decoder.

Subject-agnostic.This type of method aims to swap the identity of the face image with an arbitrary identity by a single model without any finetuning operation on the specific person. The face area of the source picture is represented as a vector in FSNet [46], which is merged with a non-face target image to obtain the swapped face image. IPGAN [2] separate identities and face characteristics into distinct vectorized representations. FSGAN [47], FaceShifter [39], Hififace [67], MegaFace [79], and Simswap [7] produce state-of-the-art outcomes by their remarkable performance, based on earlier efforts. The whole pipeline of FaceShifter is shown in Figure 3.4. FaceShifter called the model in the first stage AEI-Net and the model in the second

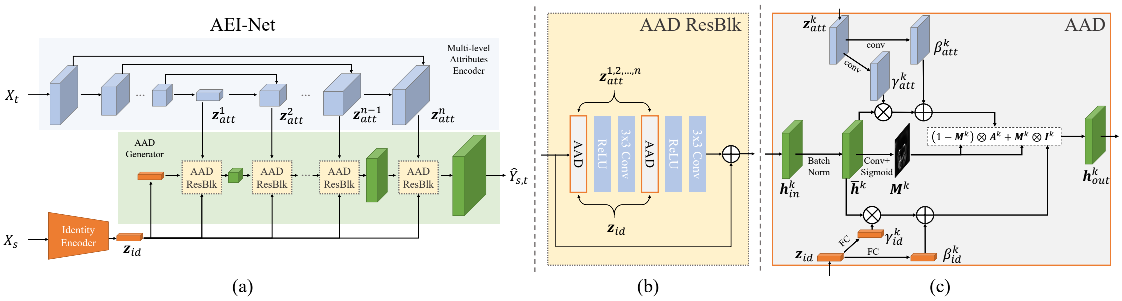


Figure 3.4 : The overview of AEI-Net which is proposed by Faceshifter [39] . AEI-Net consists of an Identity Encoder, a U-NET spatial feature encoder and four AAD ResBlks. The key compent of AAD ResBlks is Adaptive Instance Normalization [27] and Spatially-adaptive Normalization [48].

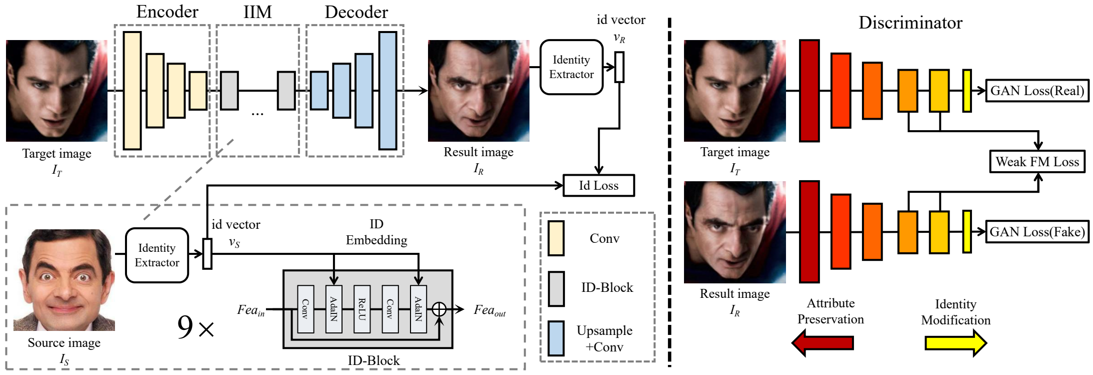


Figure 3.5 : The SimSwap framework. The generator of Simswap is a encoder-decoder structure. ID Injection Module connects the encoder and decoder, the goal of face swapping is achieve by this block. From the target image I_T, the Encoder extracts features Fea_T. The I_S identification information is sent to Fea_T by the ID Injection Module. The Decoder adds the changed features back to the final picture. Simswap employ Identity Loss to urge his network to produce outcomes that are comparable to the source face's identity. Weak feature matching loss help achieve the balance between face-swapped extent and authenticity of the output image.

stage HEAR-NET. In the subject-agnostic method, the key point is to train a model to extract features of the source image, while the feature should be sufficient to express the identity information of the source person. FaceShifter proposed a new way to achieve Subject-agnostic face swapping. Instead of using a random initialization and training the feature extraction model from scratch, to learn what kind of feature can be used to represent the identity of the source person just like FSNet, IPGAN, and FSGAN did. FaceShifter proposed to use a well-trained face recognition model to get the representative feature $\boldsymbol{z}_{id}$ of source person X_s. As shown in the upper part of Figure 3.4(a), AEI-Net employs an Encoder-Decoder model while the feature of the encoder layer while be concatenated with the feature from the corresponding decoder layer as UNET [53] did. FaceShifter apply such a model to target person image X_t and defines the output feature of each decoder layer as $\boldsymbol{z}_{att}^{k}$. All the $\boldsymbol{z}_{id}$ along with $\boldsymbol{z}_{att}^{k}$ are fed to AAD Generator while the $\boldsymbol{z}_{id}$ and $\boldsymbol{z}_{att}^{k}$ do fusion operation by using in AAD Reslk which is shown in Figure 3.4(b). The key component of AAD ResBlk is AAD which is illustrated in Figure 3.4(c).

The idea of using an identity encoder to extract identity feature proposed by FaceShifter is very influential, and later work like Simswap [7] and Hififace [67] all have used this idea, and the effect is excellent.

Simswap. The framework of Simswap is shown in Figure 3.5. Simswap proposed an ID-Block to inject identity information. The key component of ID-Block is the AdaIN [27] and Resblock design [25]. Different from Faceshifter, which uses $\boldsymbol{z}_{att}$ to preserve the attribute from the target image, Simswap modified the Feature Matching Loss from [30] to Weak Feature Matching Loss, which is shown below:

$$L_{wFM}(D) = \sum_{i=m}^{M} \frac{1}{N_i} \| D^{(i)}(I_R) - D^{(i)}(I_T) \|_1 \tag{3.1}$$

$\boldsymbol{I}_R$ means the swapped image, $\boldsymbol{D}^i$ means the feature of i-th layer from discriminator, and $\boldsymbol{I}_T$ means the target image.

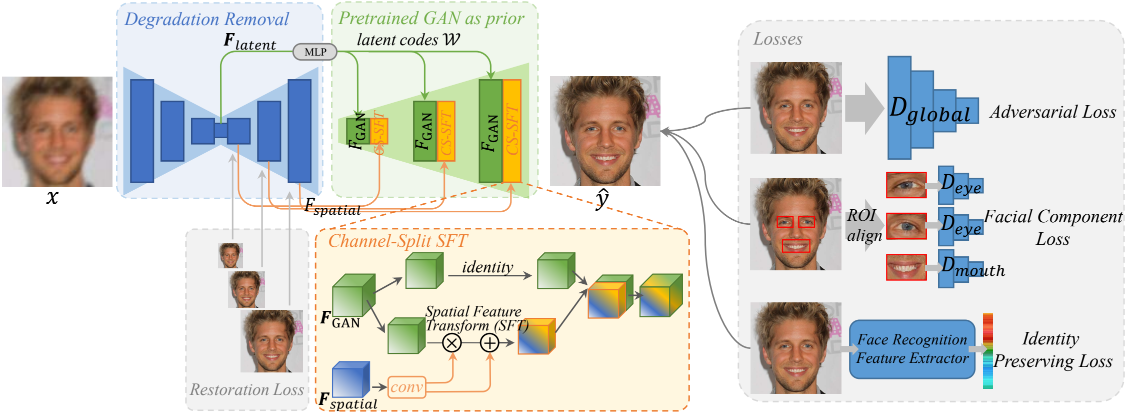


Figure 3.6 : **Framework of GFP-GAN**. It consists of a degradation removal module (which is a U-Net structure) and a pretrained face GAN(StyleGAN2) as facial prior. The corresponding layers between U-Net and StyleGAN2 are connected through Channel-Split Spatial Feature Transform (CS-SFT) and latent code mapping.

3.2.3 Generative Priors for Image Generation

Using pre-trained GANs as generative prior is a new method previously exploited by GAN inversion. Powerful GAN model like PGGAN [31], StyleGAN [32] and StyleGAN2 [33] can produce photorealistic and high resolution (up to 1024*1024) human face picture. Previous methods [1, 22, 52, 44] achieve the goal of GAN inversion by modulating the latent code of pre-trained GANs. Abdal et al. [1] and Gu et al. [22] project the original image to latent space by repeat optimization, which is super slow and not practical. Pixel2Style2Pixel (pSp) [52] adopt a ResNet [25] with Feature pyramid networks (FPN) [41] to extract the multi-scale features of desired image. pSp feeds the multi-scale extracted features as latent code of generative priors and gets the desired output. The above methods utilize the generative prior by a channel-wise operation. These methods can get the rough outline of the original image, but they are not good in detail and texture reconstruction.

GFPGAN [66] is a beautiful work which applies generative prior information in Blind Face Restoration. The framework of GFPGAN is shown in Figure 3.6.

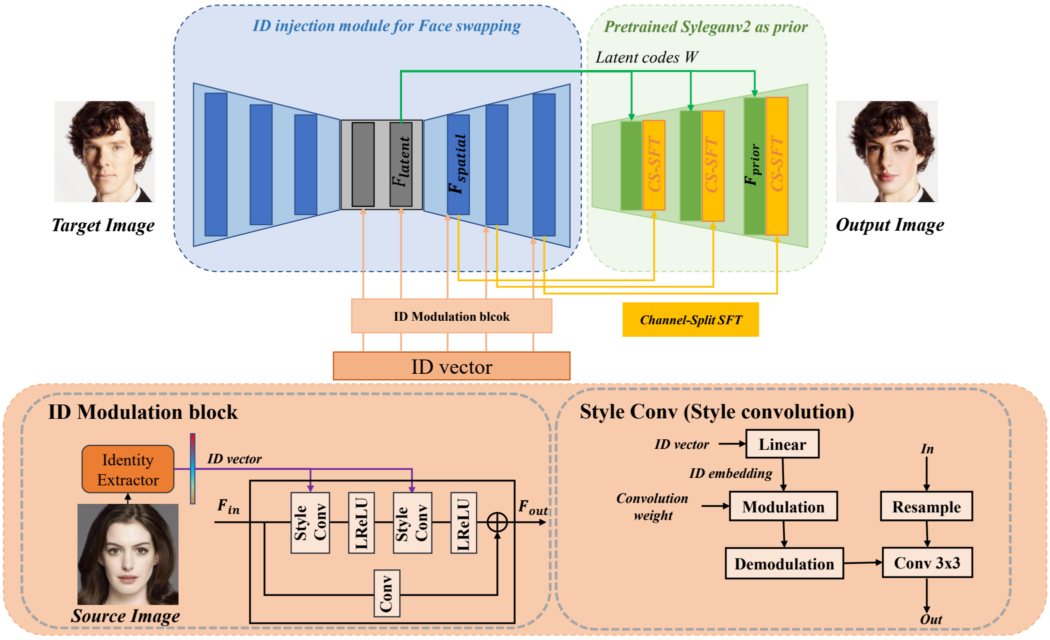


Figure 3.7 : Overview of Ours framework. This framwork is modified on GFPGAN. The biggest difference between GFPGAN and ours framework is the ID injection module. We modify the original U-NET of GFPGAN by removing the skip connection, resulting in an Encoder-Decoder structure. We use ID modulation blocks to achieve the face swapping goal.

Previous works utilize generative prior by only exploiting the mapping between image and latent code. GFPGAN proposed to use a U-Net network as a degradation removal module that can represent an image by hierarchical information, that is, the multiple-resolution feature map generated by each layer of the decoder in the U-Net. And GFPGAN proposes a Channel-Split Spatial Feature Transform (CS-SFT) to guide the generative prior generation by the hierarchical information.

In this section, we introduce the background of Image Synthesis. We also elaborate on the existing Face swapping methods and the ability of Generative Facial GAN Prior. In the next section, we will explain how to utilize Generative Facial GAN Prior in Face Swapping task to obtain high-fidelity Face Swapping results.

3.3 Methodology

3.3.1 Network design

Inspired by GFPGAN [66], we proposed a new framework to achieve face swapping goal via generative prior, shown in Figure 3.7. Given a target image and a source image, Face Swapping aims to exchange the identity of the target image to the identity of the source image while the other attributes of the target image (background, expression et al.) should be preserved.

The overall framework of our method consists of an ID injection module, a pre-trained StyleGAN2 as generative prior and discriminators. The central part of this framework is the ID injection module. GFPGAN used a U-NET structure where the feature on the encoder side will directly feed to the corresponding decoder layer via the skip connection. Such a skip connection design is helpful for the model to capture all the spatial information of the input image and make the reconstruction. Such a design works very well when only small-scale or local editing is performed on the input image. This is, no large-scale geometric editing is involved. Since the blind face restoration task does not include large-scale changes in geometry, the output result maintains the same geometric shape as the input. Using U-Net here is beneficial to produce good results. However, the U-Net structure is not helpful for face swapping tasks. The face swapping task aims to switch the target image's identity to the source image's identity. This process is equivalent to replacing the target face's geometry with the source face's geometry. Using the U-Net structure in the face swapping task will cause the model's output to be more biased towards the input image, damaging the ability of face swapping. Therefore, we removed the skip connection in U-Net. That is, there is no direct information interaction between the encoder and the decoder. We called the U-Net without skip connection *Enc-Dec*. We get $\boldsymbol{F}_{latent}$ and $\boldsymbol{F}_{spatial}$ by feeding input image $\boldsymbol{x}$ to *Enc-Dec* as shown in Eq. 3.2.

$$\boldsymbol{F}_{latent}, \boldsymbol{F}_{spatial} = \texttt{Enc-Dec}(\boldsymbol{x}). \tag{3.2}$$

The latent features $\boldsymbol{F}_{latent}$ is the bottleneck feature in which will then be mapped to the same size as the latent code of StyleGAN2 through multi-layer perceptron layers (MLP), and then the mapped latent features are fed to StyleGAN2 as latent code(Eq. 3.3). $\boldsymbol{F}_{latent}$ are the feature from each decoder layer.

$$\begin{aligned} \mathcal{W} &= \texttt{MLP}(\boldsymbol{F}_{latent}), \\ \boldsymbol{F}_{\mathrm{GAN}} &= \texttt{StyleGAN}(\mathcal{W}). \end{aligned} \tag{3.3}$$

To perform id information injection, we modulate the features from the bottleneck and decoder of the ID injection module by our proposed Id Modulation Blocks. The detailed design of ID Modulation block is shown in the lower part of Figure 3.7. ID Modulation block is inspired by Residual Block [25] and we replace the convolution and batch normalization of the residual branch with our proposed style convolution. Inspired by Faceshifter [39] and Simswap [7], we used a face recognition network to extract the id embedding to represent the id information of the source image. The face recognition network we used here is Arcface [10]. Arcface's network design intends to obtain a unique id embedding for each independent person, and the id embedding of two different people should be entirely dissimilar. That is, this embedding theoretically only contains a unique id feature of a person, and other attributes of the current person, such as the current background, the lighting environment and the orientation of the face are not included. Such information is precisely what the face swapping framework need. In Faceshifter and Simswap, they both use Adaptive Instance Normalization (AdaIN) [27] to fuse id embedding with a network feature. AdaIN is formulated by:

$$AdaIN(F, id_S) = \sigma_S \frac{F - \mu(F)}{\sigma(F)} + \mu_S \tag{3.4}$$

Here, F means the input feature and id_S is obtained by sending the source image into the face recognition network, and $\mu(F)$ and $\sigma(F)$ means the channel-wise mean and standard deviation of F. σ_S and μ_S obtained from id_S via fully connection layers. However, as pointed out by StyelGAN2, the instance normalization in AdaIN causes water droplet-like artifacts. We use style modulation proposed in StyleGAN2 to integrate id information instead of AadIN. The style modulation is formulated in Eq. 3.5 and Eq. 3.6, where i, j and k enumerate the convolution footprint of input feature,the output feature and spatial dimension. m^s is generated from id embedding id_S via fully connected layer and s is the scale of input feature. Due to the characteristics of StyleGAN2, each layer of StyleGAN2 outputs an intermediate result, so we used id modulation blocks for the bottleneck in the id injection module and each layer of the decoder to ensure that the id information is fully integrated.

$$w'_{ijk} = s_i \cdot w_{ijk} \cdot m^s_j, \tag{3.5}$$

$$w''_{ijk} = w'_{ijk} \Big/ \sqrt{\sum_{i,k} {w'_{ijk}}^2 + \epsilon}, \tag{3.6}$$

To utilize the information from the ID injection module, we adopt the Channel-Split Spatial Feature Transform (CS-SFT) from GFPGAN [66]. GFPGAN find out that only using latent code to control image generation may not achieve satisfactory results. CS-SFT is proposed to fuse the multi-resolution spatial with StyleGAN2 features to modulate the generation. The full name of CS-SFT is Channel-Split Spatial Feature Transform. Spatial Feature Transform is inspired by SPADE [48], formulated by:

$$\begin{aligned} \boldsymbol{\alpha}, \boldsymbol{\beta} &= \texttt{Conv}(\boldsymbol{F}_{spatial}), \\ \boldsymbol{F}_{output} &= \texttt{SFT}(\boldsymbol{F}_{\mathrm{GAN}} | \boldsymbol{\alpha}, \boldsymbol{\beta}) = \boldsymbol{\alpha} \odot \boldsymbol{F}_{\mathrm{GAN}} + \boldsymbol{\beta}. \end{aligned} \tag{3.7}$$

GFPGAN points out that preforming modulation on GAN features by leaving half GAN features to directly pass through can achieve better fidelity and realness. CS-SFT is formulated by:

$$\begin{aligned}\boldsymbol{F}_{output} &= \texttt{CS-SFT}(\boldsymbol{F}_{\mathrm{GAN}}|\boldsymbol{\alpha},\boldsymbol{\beta}) \\ &= \texttt{Concat}[\texttt{Identity}(\boldsymbol{F}_{\mathrm{GAN}}^{split0}),\boldsymbol{\alpha}\odot\boldsymbol{F}_{\mathrm{GAN}}^{split1}+\boldsymbol{\beta}],\end{aligned}$$

Following GFPGAN [66], we introduce a global discriminator D and three facial component discriminators. Facial component discriminators including left eye discriminator D_{leye}, right eye discriminator D_{reye} and mouth discriminator D_{mouth}. The global discriminator is used to supervise the overall image realism. But the authenticity of the facial component is also critical for face realism. Only using a global discriminator cannot achieve part-specific supervision. The introduction of the facial component discriminator is to ensure the authenticity of the face. During training, we crop each facial component by ROI align [24] and then feed to the corresponding facial component discriminator.

3.3.2 Loss function

The loss function of our method consists of reconstruction loss, identity loss, adversarial loss and weak feature matching loss.

Reconstruction loss. Face Swapping aims to swap the target image's identity with the source image's identity. If the target image and source image came from the same person, the output image of our framework should be the same as the target image. We use L1 loss which captures the pixel-wise difference between the output image and target image as our reconstruction loss, which is formulated as:

$$L_{Recon} = \|G(I_{T,1}, I_{T,2}) - I_{T,1}\|_1 \tag{3.8}$$

Here $G(\cdot,\cdot)$ is the generator of our method. It takes two images as input. $I_{T,1}$ and $I_{T,2}$ are two different images from the same person.

Identity loss. After feeding a source image and a target image to the generator, the identity of the output image should be the same as the source image, while the background from the target image should persevere. We use Arcface [10], which is a face recognition network to extract the id vector of source image V_S and the id vector of output image V_O. Cosine similarity between these two vector is used as identity loss. The higher the cosine similarity of two id vectors, the closer they are. The id loss is shown in Eq. 3.9, where $cos(\cdot, \cdot)$ is the function to calculate the cosine similarity of two different id vector.

$$L_{ID} = 1 - cos(V_O, V_S) \tag{3.9}$$

Adversarial loss. We adopt adversarial loss to encourage generator output photorealistic face-swapped images. The adversarial loss is formulated as follows:

$$\begin{aligned} \mathcal{L}_{adv} = \mathbb{E}_{\hat{y}}[\log(1 - D(\hat{y}))] + \\ \mathbb{E}_{\hat{y}_{\texttt{leye}}}[\log(1 - D_{\texttt{leye}}(\hat{y}_{\texttt{leye}}))] + \\ \mathbb{E}_{\hat{y}_{\texttt{reye}}}[\log(1 - D_{\texttt{reye}}(\hat{y}_{\texttt{reye}}))] + \\ \mathbb{E}_{\hat{y}_{\texttt{mouth}}}[\log(1 - D_{\texttt{mouth}}(\hat{y}_{\texttt{mouth}}))] \end{aligned} \tag{3.10}$$

where $\hat{y}$ is the whole output image from generator and $\{\hat{\mathtt{y}}_{\texttt{leye}}, \hat{\mathtt{y}}_{\texttt{reye}}, \hat{\mathtt{y}}_{\texttt{mouth}}\}$ are the crop regions of output image which are belong to facial component collection $\{\texttt{left eye}, \texttt{right eye}, \texttt{mouth}\}$. D is the global discriminator and $\{D_{\texttt{leye}}, D_{\texttt{reye}}, D_{\texttt{mouth}}\}$ are the facial component discriminators.

Weak Feature Matching loss. Simswap [7] proposes weak feature matching loss to help achieve the balance between face-swapped extent and authenticity of the output image. Weak feature matching loss aims to ensure the authenticity of face-swapped images by closer the feature map distance between the output image and the target image on the discriminator side. Here is the formulation of weak feature matching loss:

$$L_{wkFM}(D) = \sum_{i=m}^{M} \frac{1}{M-m} \|D^{(i)}(I_O) - D^{(i)}(I_T)\|_1 \tag{3.11}$$

M is the total layer number of the discriminator, and $D^{(i)}$ is the i-th layer of the discriminator. Weak feature matching loss forces the output image and target image to have similar features from layer m to layer M of global discriminator D.

The overall loss function is a weighted sum of the above four losses:

$$L = \lambda_{ID} L_{ID} + \lambda_{Recon} L_{Recon} + \lambda_{adv} L_{adv} + \lambda_{wkFM} L_{wkFM} \tag{3.12}$$

We set the loss weights as follows: $\lambda_{ID} = 10$, $\lambda_{Recon} = 10$, $\lambda_{adv} = 1$ and $\lambda_{wkFM} = 5$. λ_{ID}, λ_{Recon}, λ_{adv} and λ_{wkFM} are the weights of reconstruction loss, identity loss, adversarial loss and weak feature matching loss respectively.

3.4 Experiment

Implementation detail. We use StyleGAN2 (512x512) trained on the FFHQ dataset as our generative prior. Our method can generate a high-fidelity face-swapped image with 512x512 resolution with this setting. We trained the overall model on both VGGFace2-HQ [64] and FFHQ [32] for 200 epochs on four NVIDIA RTX 2080Ti 12G. We use Adam [35] as our optimizer with $\beta_1 = 0$ and $\beta_2 = 0.999$. The learning rate is set to 0.0002 without decay.

Quantitative comparison on FaceForensics++. We compare our method with state-of-the-art face swapping method: Simswap [7], FaceSwap [14], FaceShifter [39] and Hififace [67]. Following Hififace [67], we evaluate these methods on FaceForensics++ [55] dataset by measuring ID retrieval, pose error and face shape error. ID retrieval measure the degree of similarity between the face-swapped image and the source image by using facial recognition network [10]. Pose error estimate how well

Method	ID↑	Pose↓	Shape↓
FaceSwap [14]	54.19	2.51	0.610
FaceShifter [39]	97.38	2.96	0.511
SimSwap [7]	92.83	**1.53**	0.540
Hififace [67]	**98.48**	2.63	**0.437**
Ours	<u>97.83</u>	<u>2.50</u>	<u>0.504</u>

Table 3.1 : Quantitative comparison on FaceForensics++.

the face-swapped image maintains the face orientation of the target image by utilizing a head pose estimator [56]. Face shape error calculate the face shape distance between swapped image and the source image facial by using a 3DMM method [57], which is a 3D facial reconstruction network. A good face swapping method should get a high ID retrieval value, low pose error and low face shape error. The quantitative comparison is shown in Table. 3.1, our method ranks second on all three metrics.

Qualitative results. We collect a celebrity set from 16 stars, including American, European and Asian. We randomly combine the photos from the celebrity set and form the source-target pairs; the high-fidelity face-swapped results from celebrity source-target pairs are shown in Figure 3.8 and Figure 3.9. All the target images, source images and face-swapped results are high resolution (512x512). From Figure 3.8 and Figure 3.9, we can see that our model performs well in the following three scenarios: 1) Face swapping between different races. 2) Face swapping between different genders. 3) Face swapping involving profile faces. The qualitative results prove that our method can generate high-fidelity and high-quality face swapped results while perverse the attribute from the target image.

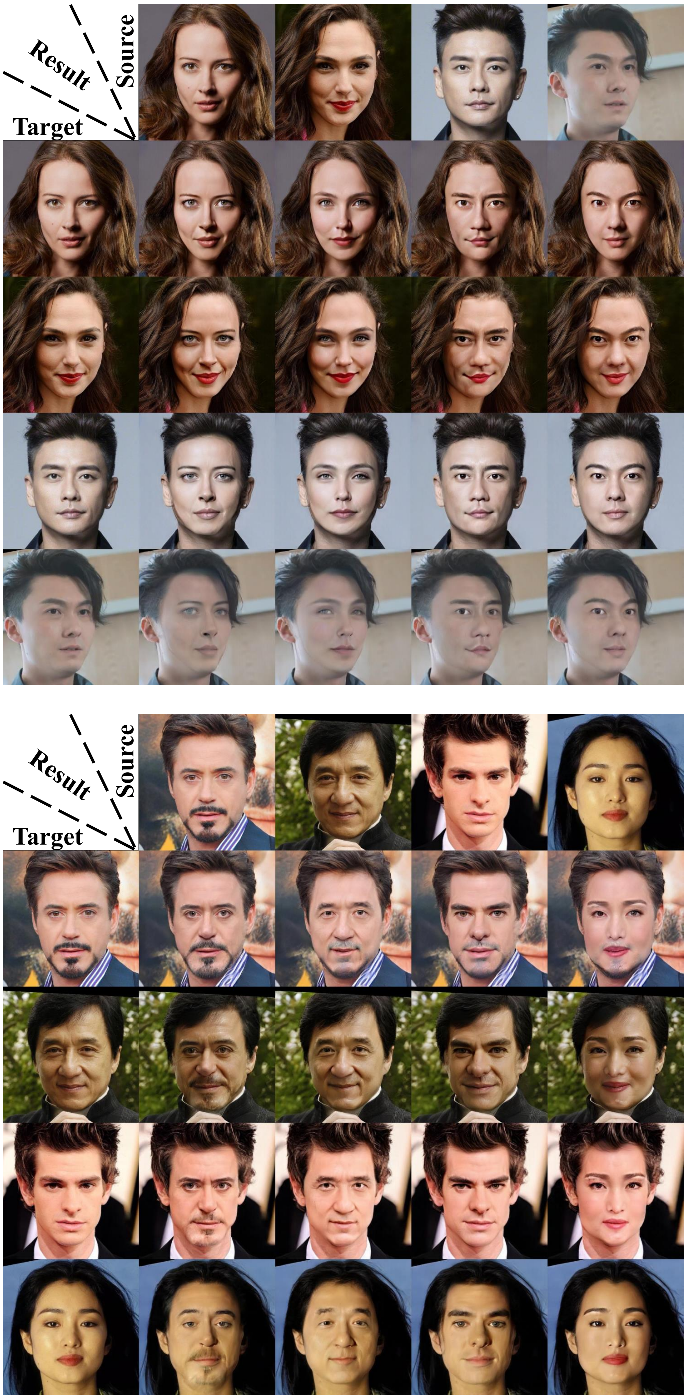


Figure 3.8 : Some HR (512x512) face-swapped results on celebrity photos. Please zoom in for detail. On each sub-figure, the photo from the first row are served as source images, and the photo from the first column are served as target images. The rest images are the face-swapped results generated by source-target image pairs.

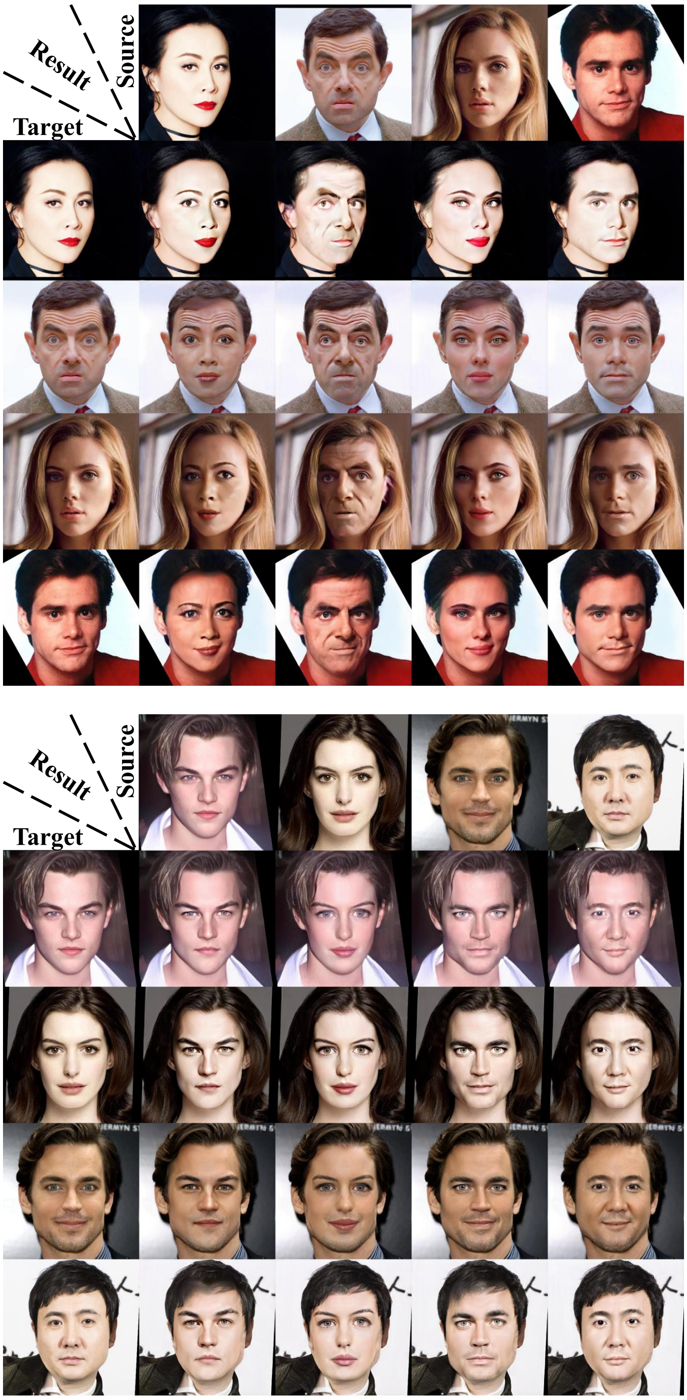


Figure 3.9 : More face swapped results on high quality celebrity photos.

Chapter 4

Future Works

4.1 Conclusion.

In this thesis, I investigate representation learning on two different tasks: Natural Language Queries and Face Swapping. In particular, we focus on Natural Language Queries in Chapter 2. We propose a multi-scale cross-modal transformer and a video frame-level contrastive loss to learn the relation between video and text. Besides, we propose two brand new data augmentation methods to obtain more meaningful training data to further improve performance. Our methods achieve the SOTA performance on Natural Language Queries and won the first place award on Ego4D Natural Language Queries (NLQ) Challenge in CVPR 2022. Chapter 3 is about Face Swapping. We propose to utilize generative facial prior to obtaining high-fidelity Face Swapping results. Generative facial prior means the well-trained facial generative model which can produce the photorealistic facial image. With the help of generative facial prior, the model can focus on the identity swapping between the source person and the target person. Our method gets competitive results compared with SOTA Face Swapping methods.

4.2 Future Directions

Natural Language Queries. In our work, we use cross-attention to fuse video and text input and then hand it over to a deep-learning model to learn the correlation. Such a process is difficult to converge. Using some deep-learning based clustering methods should be able to lower the difficulty of learning. Of course, cross-attention is just a kind of naive attention modification, and trying more attention variants is also a future direction.

Face Swapping. Recently, Diffusion Model [61, 60, 26] has shown its ability on various generative tasks including text to image generation, super-resolution, image translation, and so on. Diffusion Model gradually replaces the dominance of the Generative Adversarial Network (GAN) in generating tasks. Many researchers achieve the SOTA performance on generative tasks by utilizing Diffusion Model. My proposed Face Swapping method is based on Generative Adversarial Network (GAN) and more studies can be conducted by using Diffusion Model paradigm.

Bibliography

[1] R. Abdal, Y. Qin, and P. Wonka, "Image2stylegan: How to embed images into the stylegan latent space?" in *ICCV*, 2019.

[2] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "Towards open-set identity preserving face synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6713–6722.

[3] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv preprint arXiv:1809.11096*, 2018.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision.* Springer, 2020, pp. 213–229.

[5] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," *arXiv preprint arXiv:1808.01340*, 2018.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[7] R. Chen, X. Chen, B. Ni, and Y. Ge, "Simswap: An efficient framework for high fidelity face swapping," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2003–2011.

[8] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation,"

in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8789–8797.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[10] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[12] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (dfdc) preview dataset," *arXiv preprint arXiv:1910.08854*, 2019.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] FaceSwap, "https://github.com/ondyari/faceforensics/tree/master/dataset/faceswapkowalski," *Accessed: 2020-12-20*, 2020.

[15] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[16] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra,

"Omnivore: A single model for many visual modalities," *arXiv preprint arXiv:2201.08377*, 2022.

[17] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 1, pp. 142–158, 2015.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[20] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu *et al.*, "Ego4d: Around the world in 3,000 hours of egocentric video," *arXiv preprint arXiv:2110.07058*, vol. 3, 2021.

[21] C. Gu, C. Sun, D. A. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar *et al.*, "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6047–6056.

[22] J. Gu, Y. Shen, and B. Zhou, "Image processing using multi-code gan prior," *ArXiv*, 2019.

[23] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," 2006.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[27] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[29] L. Jiang, R. Li, W. Wu, C. Qian, and C. C. Loy, “Deeperforensics-1.0: A large-scale dataset for real-world face forgery detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2889–2898.

[30] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European conference on computer vision.* Springer, 2016, pp. 694–711.

[31] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of gans for improved quality, stability, and variation,” *arXiv preprint arXiv:1710.10196*, 2017.

[32] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410.

[33] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8110–8119.

[34] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[38] J. Lei, T. L. Berg, and M. Bansal, "Qvhighlights: Detecting moments and highlights in videos via natural language queries," *arXiv preprint arXiv:2107.09609*, 2021.

[39] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.

[40] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A large-scale challenging dataset for deepfake forensics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3207–3216.

[41] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[42] N. Liu, X. Wang, X. Li, Y. Yang, and Y. Zhuang, "Reler@ zju-alibaba submission to the ego4d natural language queries challenge 2022," *arXiv preprint arXiv:2207.00383*, 2022.

[43] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[44] S. Menon, A. Damian, S. Hu, N. Ravi, and C. Rudin, "Pulse: Self-supervised photo upsampling via latent space exploration of generative models," in *CVPR*, 2020.

[45] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2630–2640.

[46] R. Natsume, T. Yatagawa, and S. Morishima, "Fsnet: An identity-aware generative model for image-based face swapping," in *Asian Conference on Computer Vision.* Springer, 2018, pp. 117–132.

[47] Y. Nirkin, Y. Keller, and T. Hassner, "Fsgan: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7184–7193.

[48] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference*

on Computer Vision and Pattern Recognition, 2019, pp. 2337–2346.

[49] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning.* PMLR, 2021, pp. 8748–8763.

[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[51] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[52] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or, "Encoding in style: a stylegan encoder for image-to-image translation," *Arxiv*, 2020.

[53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention.* Springer, 2015, pp. 234–241.

[54] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *arXiv preprint arXiv:1803.09179*, 2018.

[55] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1–11.

[56] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 2074–2083.

[57] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "Learning to regress 3d face shape and expression from an image without 3d supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7763–7772.

[58] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 510–526.

[59] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[60] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[61] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," *arXiv preprint arXiv:2011.13456*, 2020.

[62] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural*

information processing systems, vol. 30, 2017.

[64] VGGFace2-HQ, "https://github.com/nnnnai/vggface2-hq," *Accessed: 2022-10-15*, 2022.

[65] X. Wang, L. Zhu, and Y. Yang, "T2vlad: global-local sequence alignment for text-video retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5079–5088.

[66] X. Wang, Y. Li, H. Zhang, and Y. Shan, "Towards real-world blind face restoration with generative facial prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9168–9178.

[67] Y. Wang, X. Chen, J. Zhu, W. Chu, Y. Tai, C. Wang, J. Li, Y. Wu, F. Huang, and R. Ji, "Hififace: 3d shape and semantic prior guided high fidelity face swapping," *arXiv preprint arXiv:2106.09965*, 2021.

[68] J. Xu, T. Mei, T. Yao, and Y. Rui, "Msr-vtt: A large video description dataset for bridging video and language," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5288–5296.

[69] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 8261–8265.

[70] Y. Yang, Y. Zhuang, and Y. Pan, "Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies," *Frontiers of Information Technology & Electronic Engineering*, vol. 22, no. 12, pp. 1551–1558, 2021.

[71] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[72] H. Zhang, A. Sun, W. Jing, L. Zhen, J. T. Zhou, and R. S. M. Goh, "Natural language video localization: A revisit in span-based question answering framework," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[73] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, "Span-based localizing network for natural language video localization," *arXiv preprint arXiv:2004.13931*, 2020.

[74] S. Zhang, H. Peng, J. Fu, Y. Lu, and J. Luo, "Multi-scale 2d temporal adjacency networks for moment localization with natural language," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[75] S. Zhang, H. Peng, J. Fu, and J. Luo, "Learning 2d temporal adjacent networks for moment localization with natural language," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 870–12 877.

[76] S. Zhao, L. Zhu, X. Wang, and Y. Yang, "Centerclip: Token clustering for efficient text-video retrieval," *arXiv preprint arXiv:2205.00823*, 2022.

[77] T. Zhou, W. Wang, Z. Liang, and J. Shen, "Face forensics in the wild," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5778–5788.

[78] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[79] Y. Zhu, Q. Li, J. Wang, C.-Z. Xu, and Z. Sun, "One shot face swapping on megapixels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4834–4844.