Transformer Applications
in ICCV 2021

YeongHyeon Park

Department of Electrical and Computer Engineering

SungKyunKwan University

List of paper

- CrackFormer: Transformer Network for Fine-Grained Crack Detection
  - Segmentation
  - Outdoor environment

- Localize to Binauralize: Audio Spatialization from Visual Sound Source Localization
  - Synthesis
  - Sound source separation / localization

- A Latent Transformer for Disentangled Face Editing in Images and Videos
  - Synthesis
  - Face editing

Name

Purpose

CrackFormer

CrackFormer: Transformer Network for Fine-Grained Crack Detection

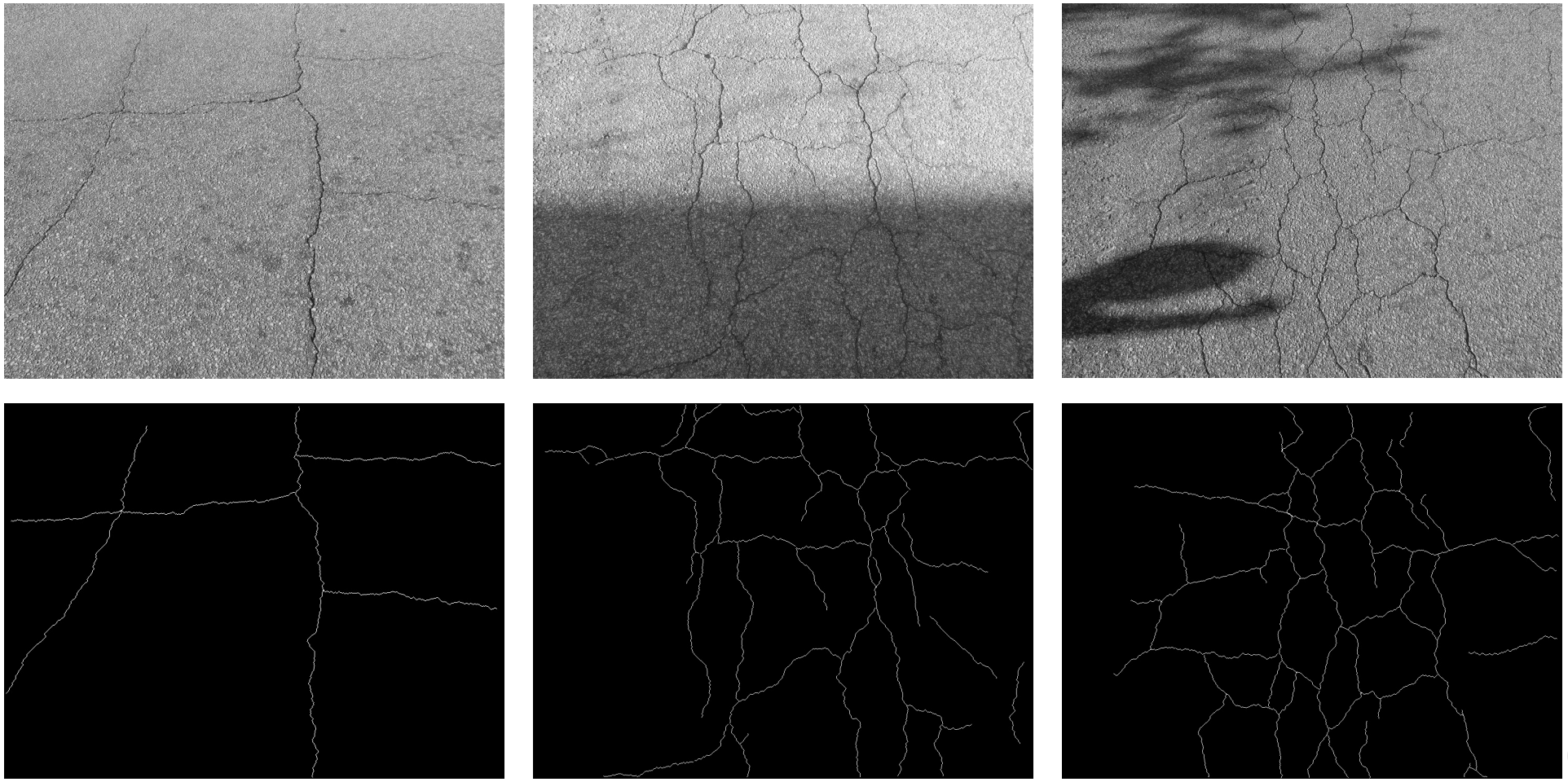
Huajun Liu[1*], Xiangyu Miao[1], Christoph Mertz[2], Chengzhong Xu[3], Hui Kong[3*]
[1]Nanjing University of Science and Technology,[2]Carnegie Mellon University,[3]University of Macau
{liuhj, miaoxy}@njust.edu.cn, cmertz@andrew.cmu.edu, {czxu, huikong}@um.edu.mo

Contributions

- new self-attention block (Self-AB) & scaling attention block (Scal-AB)
  - Self-AB:
  - Scal-AB: suppress other irrelevant features
- integrating the proposed Self-AB and Scal-AB blocks

CrackFormer – Purpose

Easy **Challenge**



CrackFormer – Structure

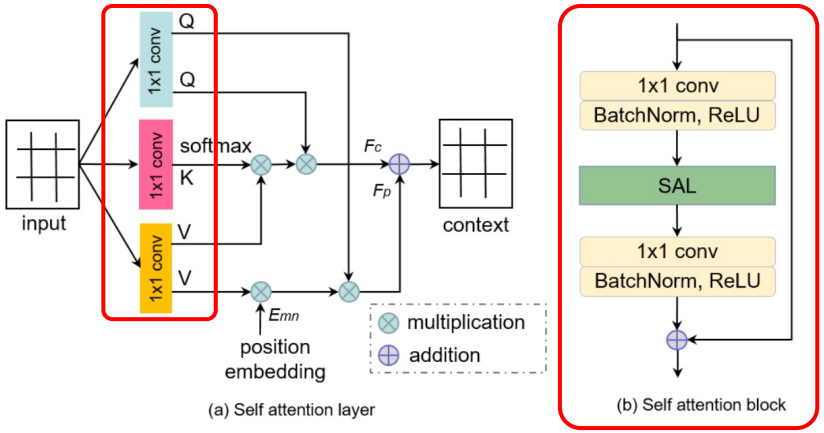


Figure 3. The self-attention block and self-attention layer.

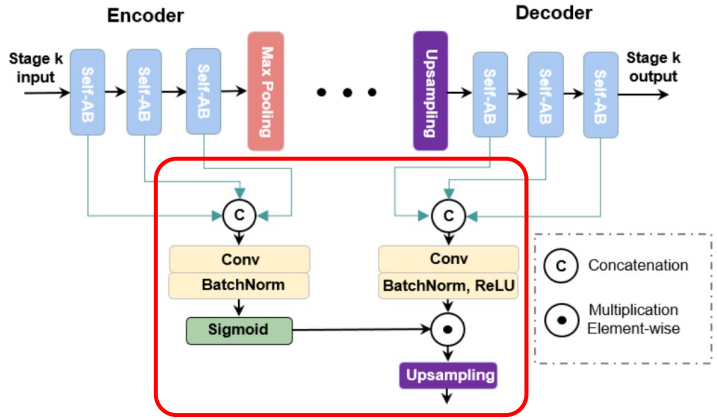


Figure 4. The scaling-attention block.

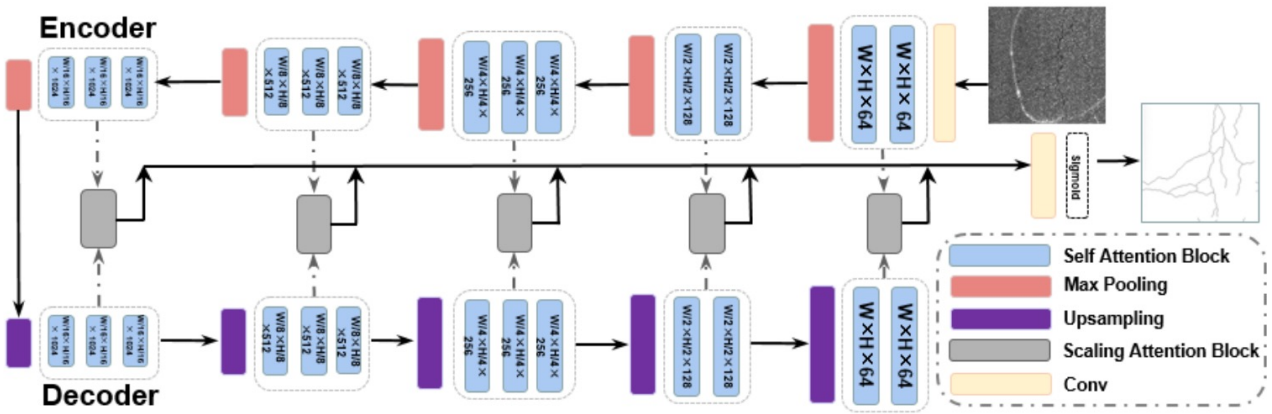
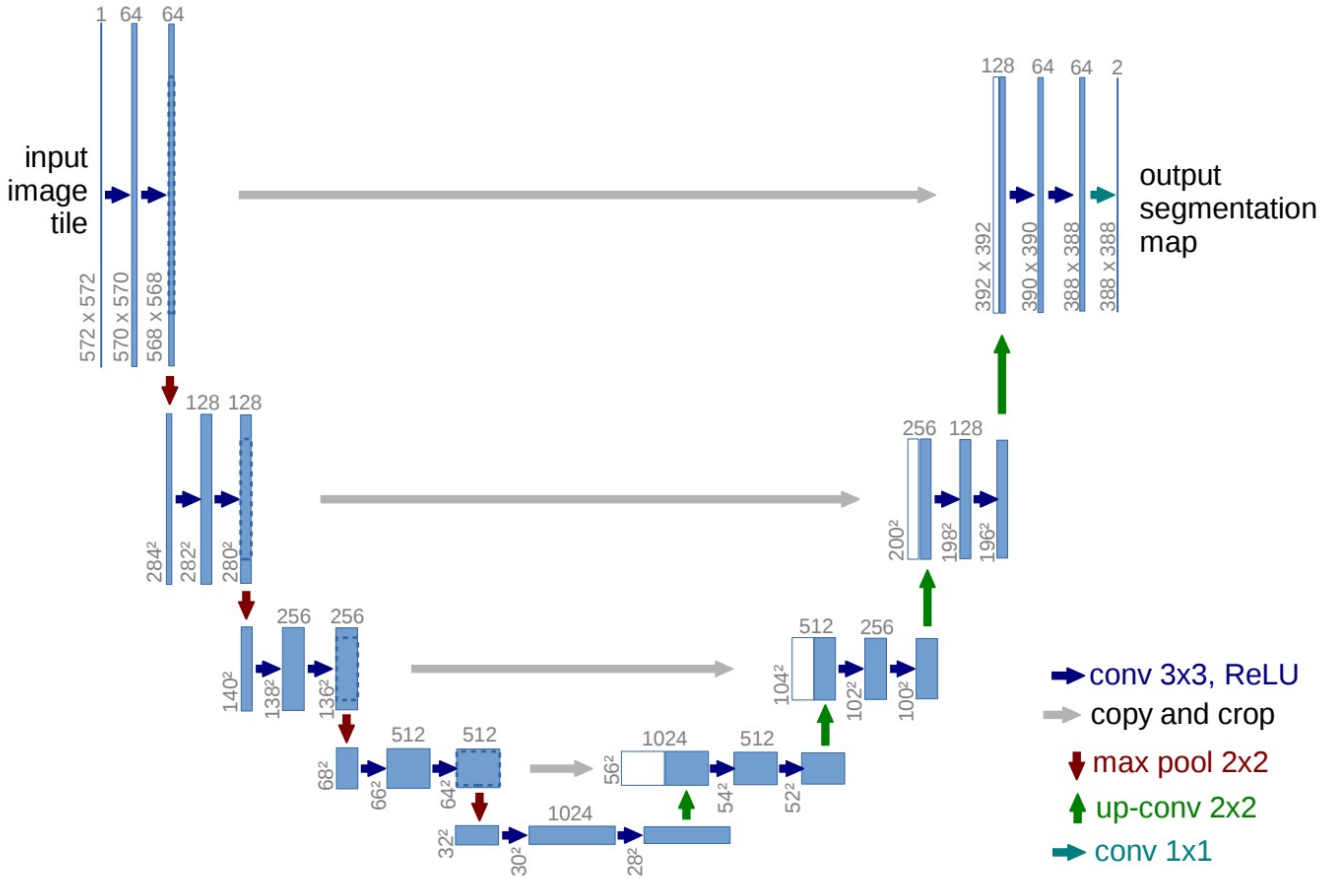


Figure 2. The structure of Crack Transformer network.

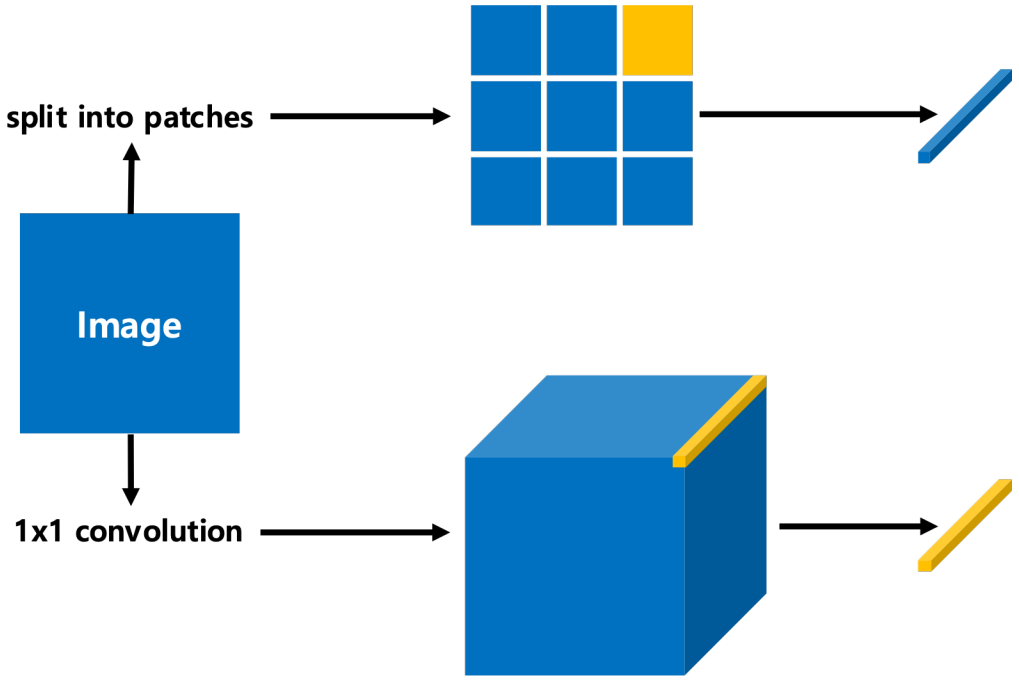
CrackFormer – Reference Model (U-Net)



Olaf Ronneberger et al. (University of Freiburg, Germany), U-Net: Convolutional Networks for Biomedical Image Segmentation, MICCAI 2015

CrackFormer – For Multi-Head Attention

where $\otimes$ is a matrix multiplication operation. Let h be the number of head, d_u be intra-depth dimension, r be the receptive field size, d_k and d_v be the dimension of tensor K and V, respectively. Then we have $\boldsymbol{Q} \in \mathbb{R}^{d_k \times h \times WH}$, $\boldsymbol{K} \in \mathbb{R}^{d_k \times d_u \times WH}$, and $\boldsymbol{V} \in \mathbb{R}^{d_v \times d_u \times WH}$. Let σ denote the operation of applying softmax normalization on the tensor. This attention operation can be interpreted as first aggregating the pixel features in $\boldsymbol{V}$ into global context vectors using the weights in $\sigma\left(\boldsymbol{K}^{\top}\right)$, and then redistributing the global context vectors back to individual pixels using the weights in $\boldsymbol{Q}$. We notice its similarity to the one used in Bello [3], but it does not use batch normalization on queries and values. Softmax normalizing on the keys constrains the output features to be convex combinations of the global context vectors.



CrackFormer – Qualitative Comparison

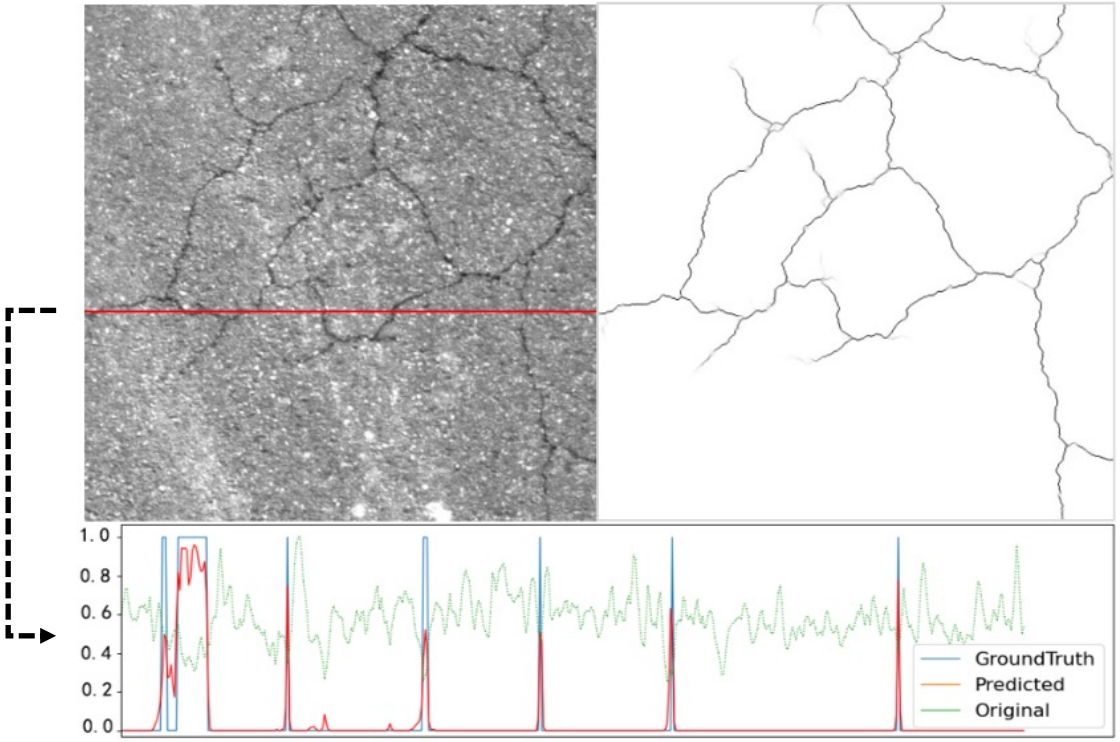


Figure 1. Crack prediction from our CrackFormer model (Best viewed in color). The upper left is a classical crack image. The upper right is the predicted result. The bottom shows a profile slice with normalized grey scale, its ground truth and corresponding crack predicted probabilities.

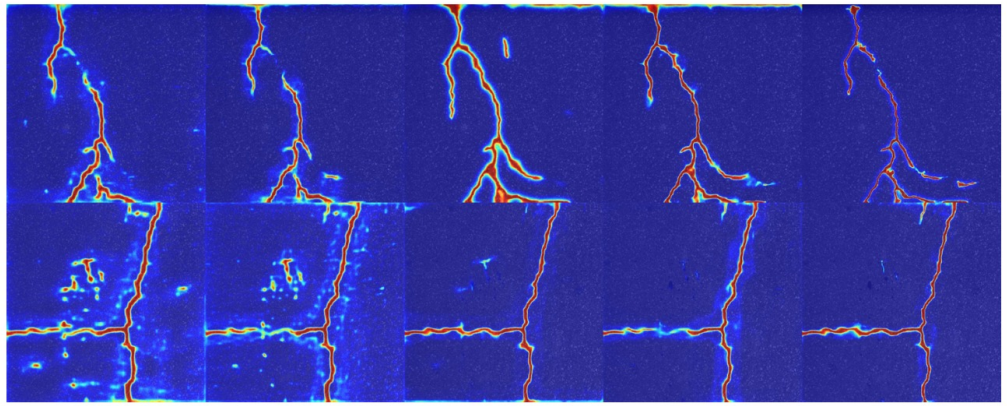


Figure 5. From left to right: the scaling-attention maps from stage 1 to stage 5, respectively.

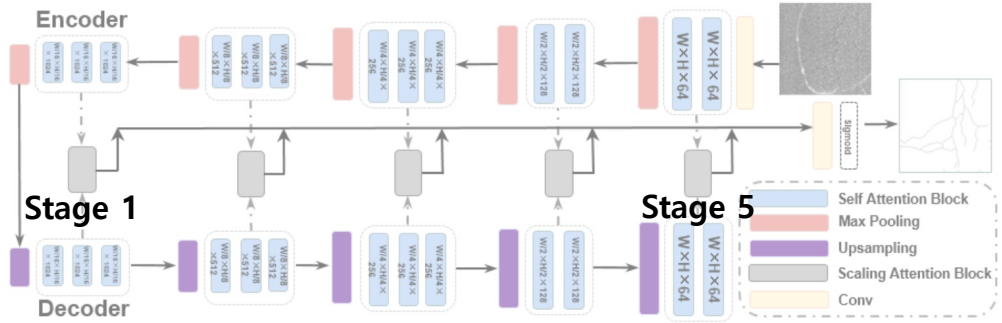


Figure 2. The structure of Crack Transformer network.

Optimal Dataset Scale (**ODS**)
- F-measure with single threshold for a whole image

Optimal Image Scale (**OIS**)
- F-measure with a threshold for each image

Average Precision (**AP**)
- no detailed descriptions

CrackFormer – Qualitative Comparison

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.662	0.673	0.683	-	-
FPHBN [35]	0.517	0.579	-	-	-
SRN [16]	0.774	0.781	0.779	451.3G	28.5M
HED [34]	0.816	0.820	0.831	146.9G	14.7M
SegNet [1]	0.844	0.851	0.862	311.3G	29.5M
U-Net [27]	0.847	0.832	0.869	400.0G	31.0M
RCF [20]	0.857	0.863	0.861	187.9G	14.8M
DeepCrack [37]	0.852	0.864	0.875	1001.7G	30.9M
CrackFormer	**0.881**	**0.883**	**0.896**	**123.0G**	**7.35M**

Table 1. Performance on the CrackTree260.

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.459	0.521	0.495	-	-
U-Net [27]	0.672	0.703	0.740	218.6G	31.0M
SRN [16]	0.755	0.789	0.795	246.6G	28.5M
SegNet [1]	0.761	0.780	0.780	170.1G	29.5M
HED [34]	0.763	0.798	0.829	80.3G	14.7M
RCF [20]	0.788	0.816	0.829	102.7G	14.8M
DeepCrack [37]	0.853	0.867	0.877	547.4G	30.9M
CrackFormer	**0.871**	**0.879**	**0.883**	**67.2G**	**7.35M**

Table 2. Performance on the CrackLS315.

Model	ODS↑	OIS↑	AP↑	FLOPs↓	mPara↓
SE [8]	0.557	0.623	0.605	-	-
HED [34]	0.719	0.763	0.758	80.3G	14.7M
SRN [16]	0.735	0.776	0.741	246.6G	28.5M
U-Net [27]	0.757	0.776	0.809	218.6G	31.0M
RCF [20]	0.789	0.829	0.820	102.7G	14.8M
SegNet [1]	0.794	0.815	0.787	170.1G	29.5M
DeepCrack [37]	0.856	0.875	0.888	547.4G	30.9M
CrackFormer	**0.877**	**0.885**	**0.894**	**67.2G**	**7.35M**

Table 3. Performance on the Stone331.

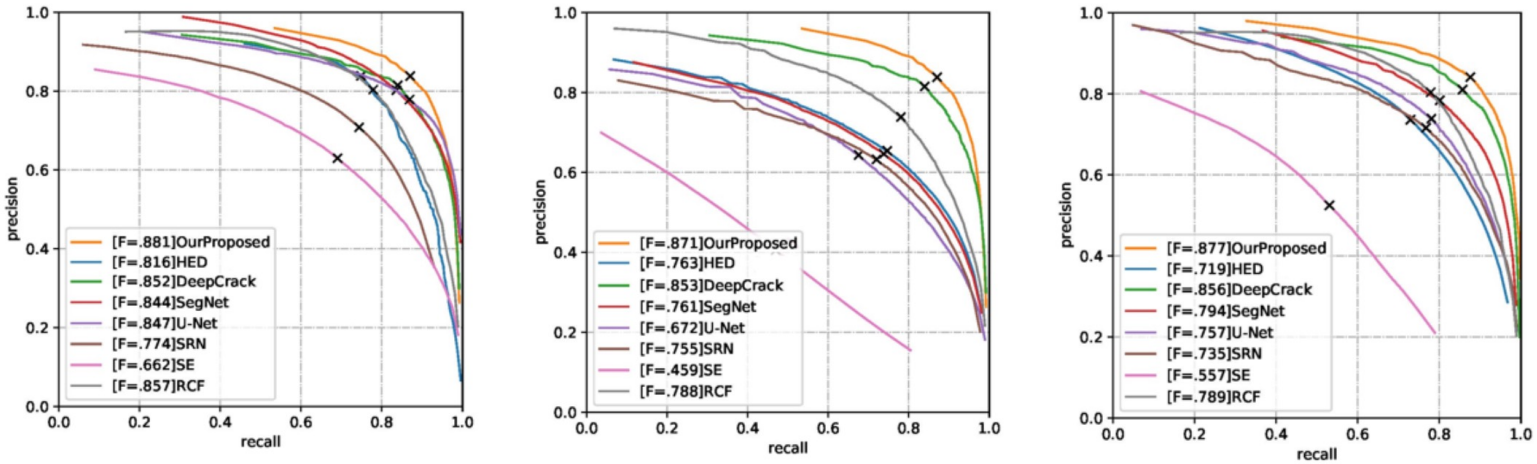


Figure 9. The precision-recall curves on the CrackTree260, CrackLS315 and Stone331, respectively.

Name

Purpose

Localize to Binauralize: L2BNet

Localize to Binauralize: Audio Spatialization from Visual Sound Source Localization

Kranthi Kumar Rachavarapu Aakanksha Vignesh Sundaresha Rajagopalan A. N.
Indian Institute of Technology Madras, India
{kranthi.rachavarapu, aakankshajha30, vigneshsundaresh}@gmail.com raju@ee.iitm.ac.in

Contributions

- end-to-end model to convert monaural audio stereo audio
  - but needs video input
- weakly-supervised learning
  - training with only few GT in learning stage-1
  - generate pseudo-GT and use for training in stage-2

L2BNet – Structure

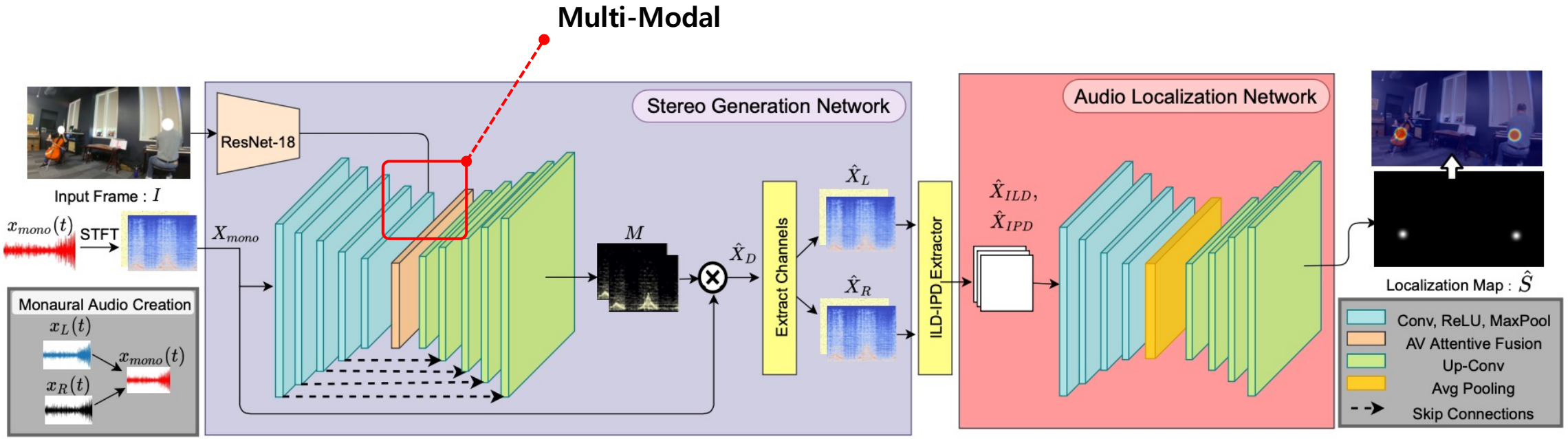


Figure 2. Architecture of our proposed L2BNet comprising of a Stereo Generation Network and an Audio Localization network with an ILD-IPD Extractor.

L2BNet – Structure Detail

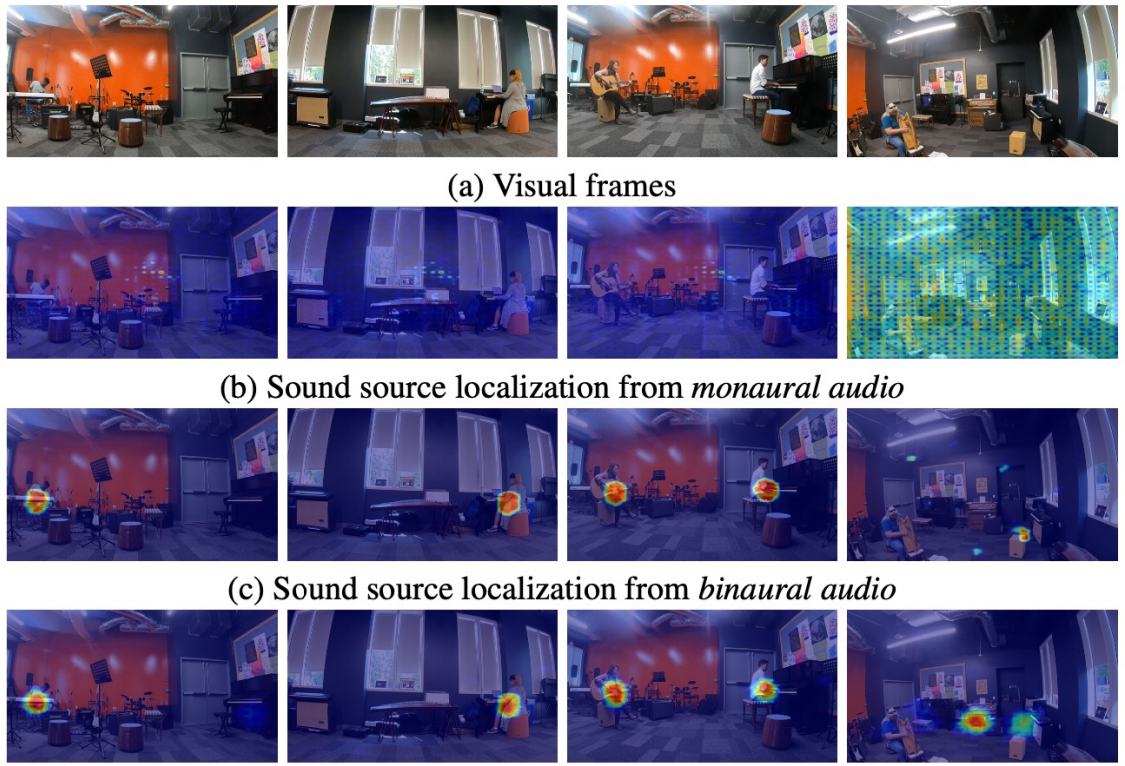
Subnetwork	Name	Type	K	S	Out
Audio Subnetwork	Encoder	Conv-2D	4	2	64
		BatchNorm-2D	-	-	64
		Conv-2D	4	2	128
		BatchNorm-2D	-	-	128
		Conv-2D	4	2	256
		BatchNorm-2D	-	-	256
		Conv-2D	4	2	512
		BatchNorm-2D	-	-	512
		Conv-2D	4	2	512
		BatchNorm-2D	-	-	512
	Decoder	ConvTranspose-2D	4	2	512
		BatchNorm-2D	-	-	512
		ConvTranspose-2D	4	2	256
		BatchNorm-2D	-	-	256
		ConvTranspose-2D	4	2	128
		BatchNorm-2D	-	-	128
		ConvTranspose-2D	4	2	64
		BatchNorm-2D	-	-	64
		ConvTranspose-2D	4	2	2
		BatchNorm-2D	-	-	2
Visual Subnework	Pretrained ResNet-18				
Attention	Query	Conv-2D	3	1	512
	Key	Conv-2D	3	1	512
	Value	Conv-2D	3	1	512

Table S1. Architecture summary of Stereo-Generation Network. K stands for kernel size, S for stride, n_f for number of input feature channels and *Out* for number of channels in convolutional layers. All the layers use *Leaky-ReLU* activation.

Name	Type	K	S	Out
Encoder	Conv-2D	4	2	64
	BatchNorm-2D	-	-	64
	Conv-2D	4	2	128
	BatchNorm-2D	-	-	128
	Conv-2D	4	2	256
	BatchNorm-2D	-	-	256
	Conv-2D	4	2	512
	BatchNorm-2D	-	-	512
	Conv-2D	4	2	1024
	BatchNorm-2D	-	-	1024
	AveragePool-2D			
Decoder	ConvTranspose-2D	4	2	512
	BatchNorm-2D	-	-	512
	ConvTranspose-2D	4	2	256
	BatchNorm-2D	-	-	256
	ConvTranspose-2D	4	2	128
	BatchNorm-2D	-	-	128
	ConvTranspose-2D	4	2	32
	BatchNorm-2D	-	-	32
	ConvTranspose-2D	4	2	4
	BatchNorm-2D	-	-	4
	ConvTranspose-2D	4	2	1
	BatchNorm-2D	-	-	1

Table S2. Architecture summary of Audio-Localization Network. K stands for kernel size, S for stride, n_f for number of input feature channels and *Out* for number of channels in convolutional layers. All the layers use *Leaky-ReLU* activation.

L2BNet – Qualitative Comparison



(a) Visual frames

(b) Sound source localization from *monaural audio*

(c) Sound source localization from *binaural audio*

(d) Sound source localization from *ILD and ITD features*

Figure 4. Visual comparisons of Audio-based Visual Sound Source Localization task using various input audio forms.

Table 2. Quantitative comparisons of the proposed *Weakly and Semi-Supervised* approach with various baseline methods on FAIR-Play and YT-Music using *STFT* and *Envelope* Distance. $F/S/W$ indicate Full/Semi/Weak supervision.

				Fair-Play		YT-Music	
	F	S	W	STFT	ENV	STFT	ENV
Mono				1.195	0.156	3.075	0.241
Mono2Binaural [5]	✓			0.951	0.141	1.346	0.179
Sep-Stereo [32]	✓			0.879	0.135	1.051	0.145
Ours-SG (10 %)		✓		1.188	0.156	2.156	0.203
Ours-SG (30 %)		✓		1.109	0.151	1.855	0.192
L2BNet-WSS (10 %)		✓	✓	1.121	0.151	1.908	0.195
L2BNet-WSS (30 %)		✓	✓	1.028	0.148	1.816	0.189

https://drive.google.com/drive/folders/1a5BV0U3RaQJS5wXyR7pzIAPMKOsGQz_q

Name

Purpose

Latent Transformer

A Latent Transformer for Disentangled Face Editing in Images and Videos

Xu Yao[1,2], Alasdair Newson[1], Yann Gousseau[1], Pierre Hellier[2]
[1] LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[2] InterDigital R&I, 975 avenue des Champs Blancs, Cesson-Sévigné, France
{xu.yao,anewson,yann.gousseau}@telecom-paris.fr, Pierre.Hellier@InterDigital.com

Contributions

- disentangled and more controllable manipulations
- identity preservation
- efficient sequential attribute editing
- generalized and stable face editing on HD videos

Latent Transformer – Concept

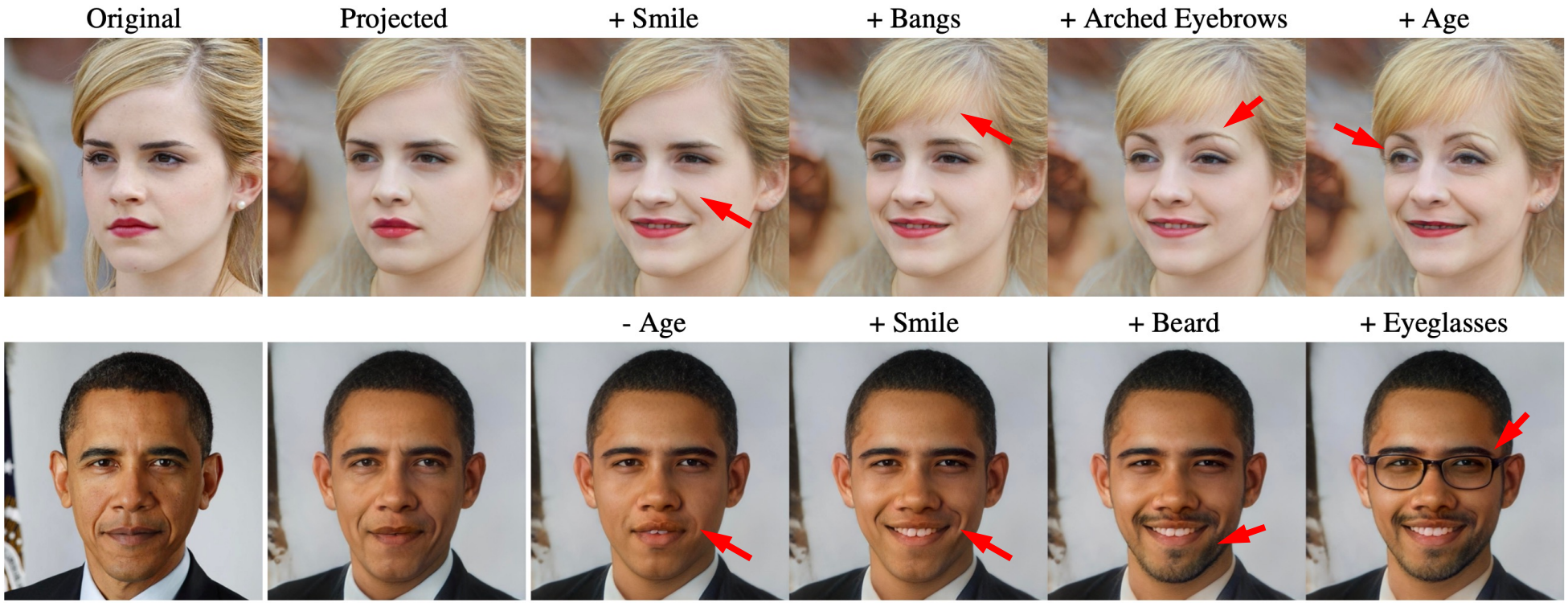


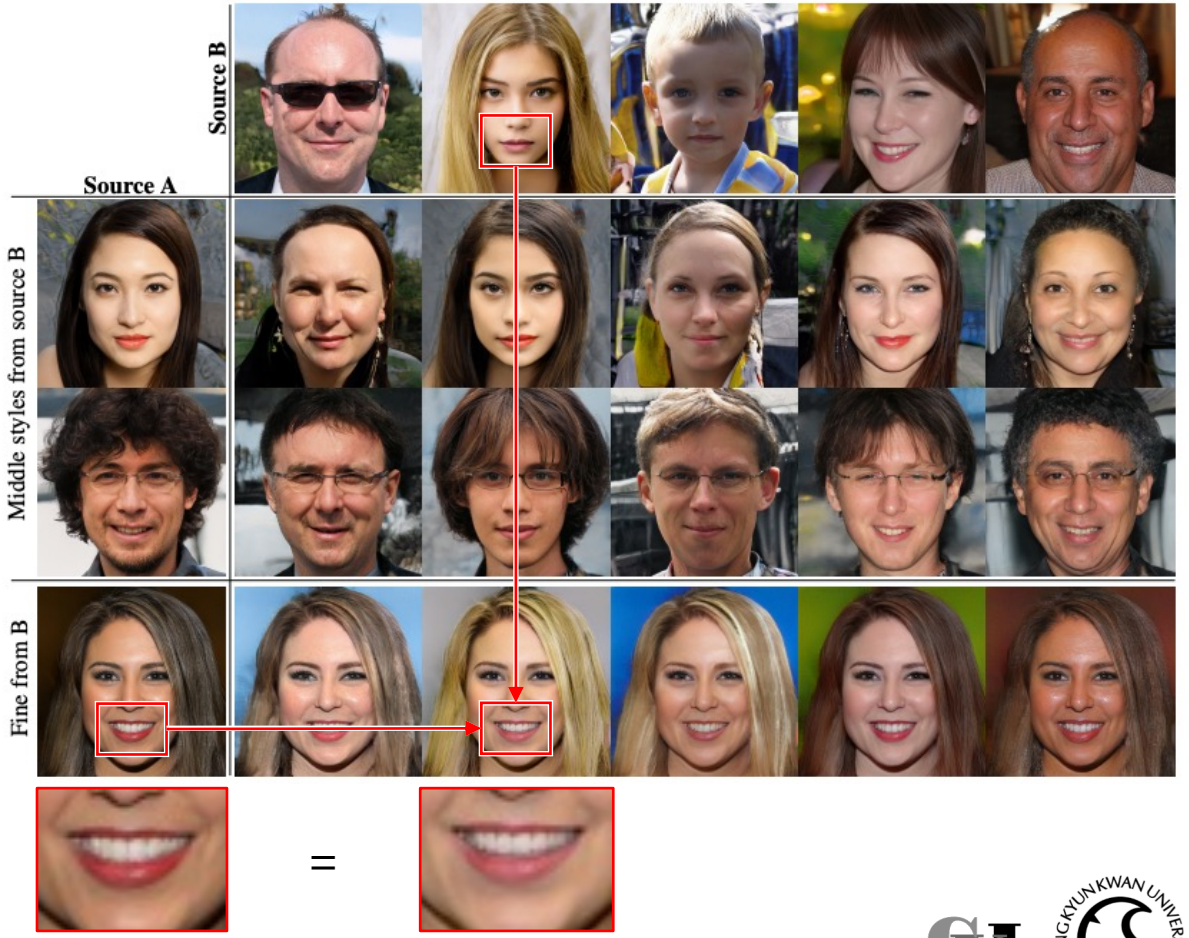
Figure 1: We project real images to the latent space of a StyleGAN generator and achieve sequential disentangled attribute editing on the encoded latent codes. From the original and the projected image, we can edit sequentially a list of attributes such as: 'smile', 'bangs', 'arched eyebrows', 'age', 'beard' and 'eyeglasses'. All results are obtained at resolution 1024×1024.

Latent Transformer – Backbone Model (StyleGAN)

Coarse
(4 x 4)

Resolution

Fine
(1024 x 1024)



Tero Kerras et al. (NVIDIA), A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR 2019

Latent Transformer – Pipeline

Named as transformer but not a transformer.
- No attention mechanism
- No reflection of sequential/positional info.

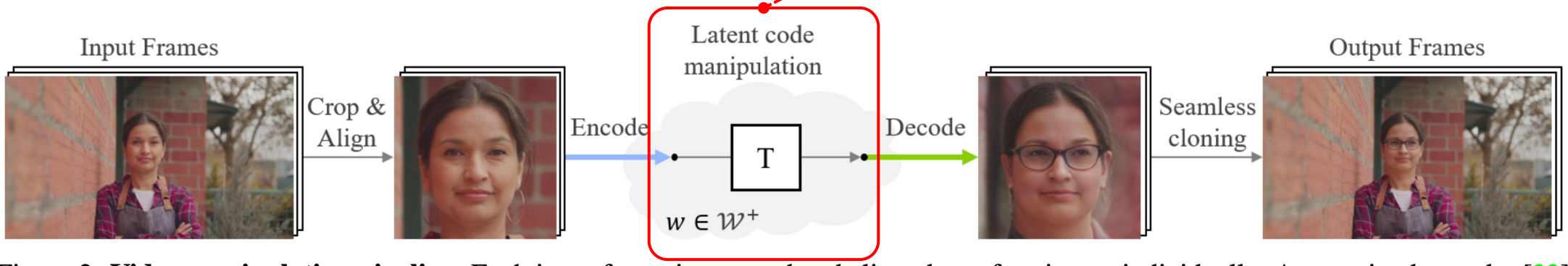
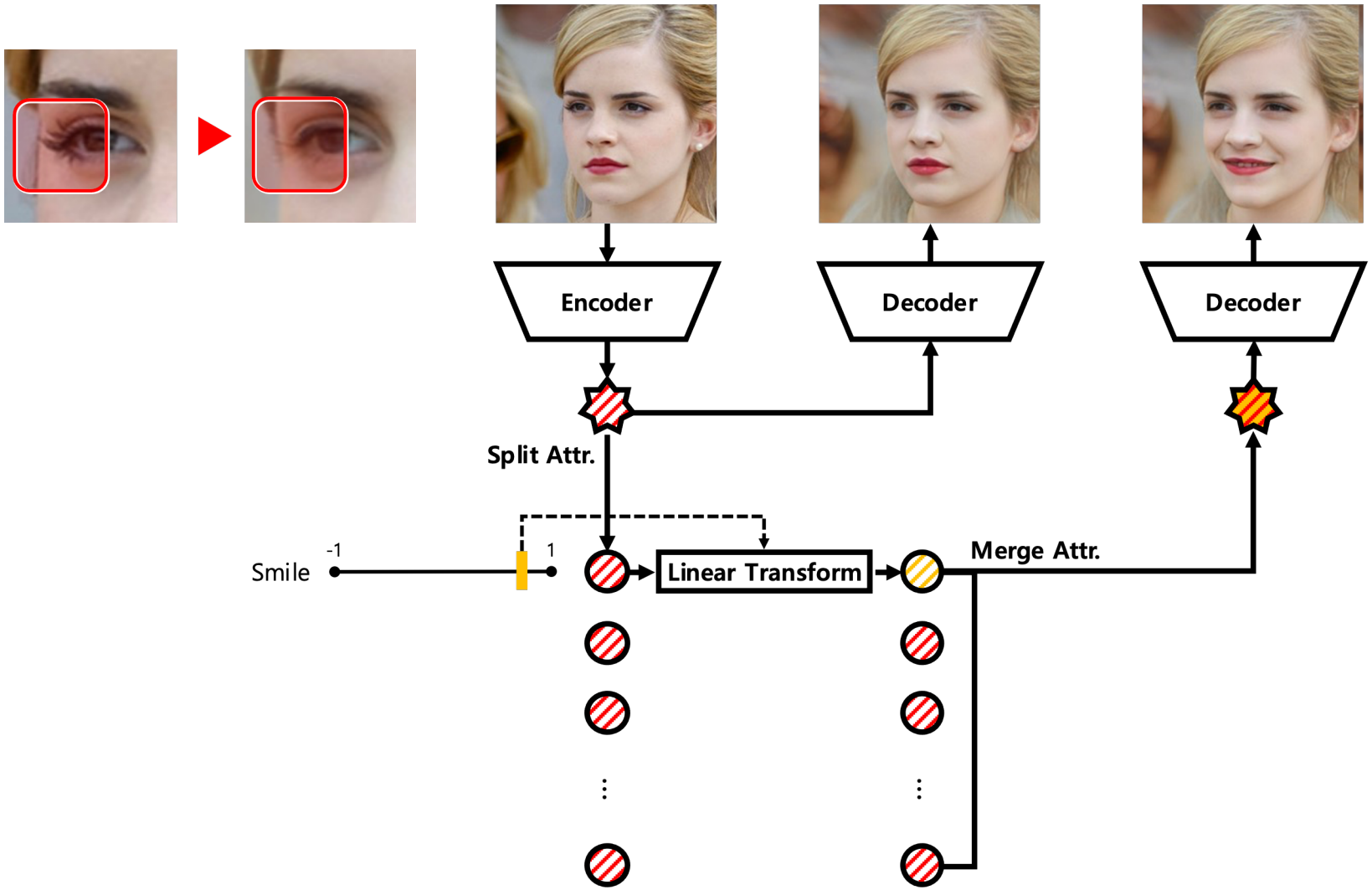


Figure 2: **Video manipulation pipeline.** Each input frame is cropped and aligned to a face image individually. A pretrained encoder [33] is used to encode the face images to the latent space $\mathcal{W}^+$ of StyleGAN [21]. The obtained latent codes are processed by the proposed latent transformer **T** to realize the attribute editing. The manipulated latent codes are further decoded by StyleGAN to generate the manipulated face images, which are blended with the original input frames to get the output frames.

Latent Transformer – Pipeline Detail



Latent Transformer – Qualitative Comparison

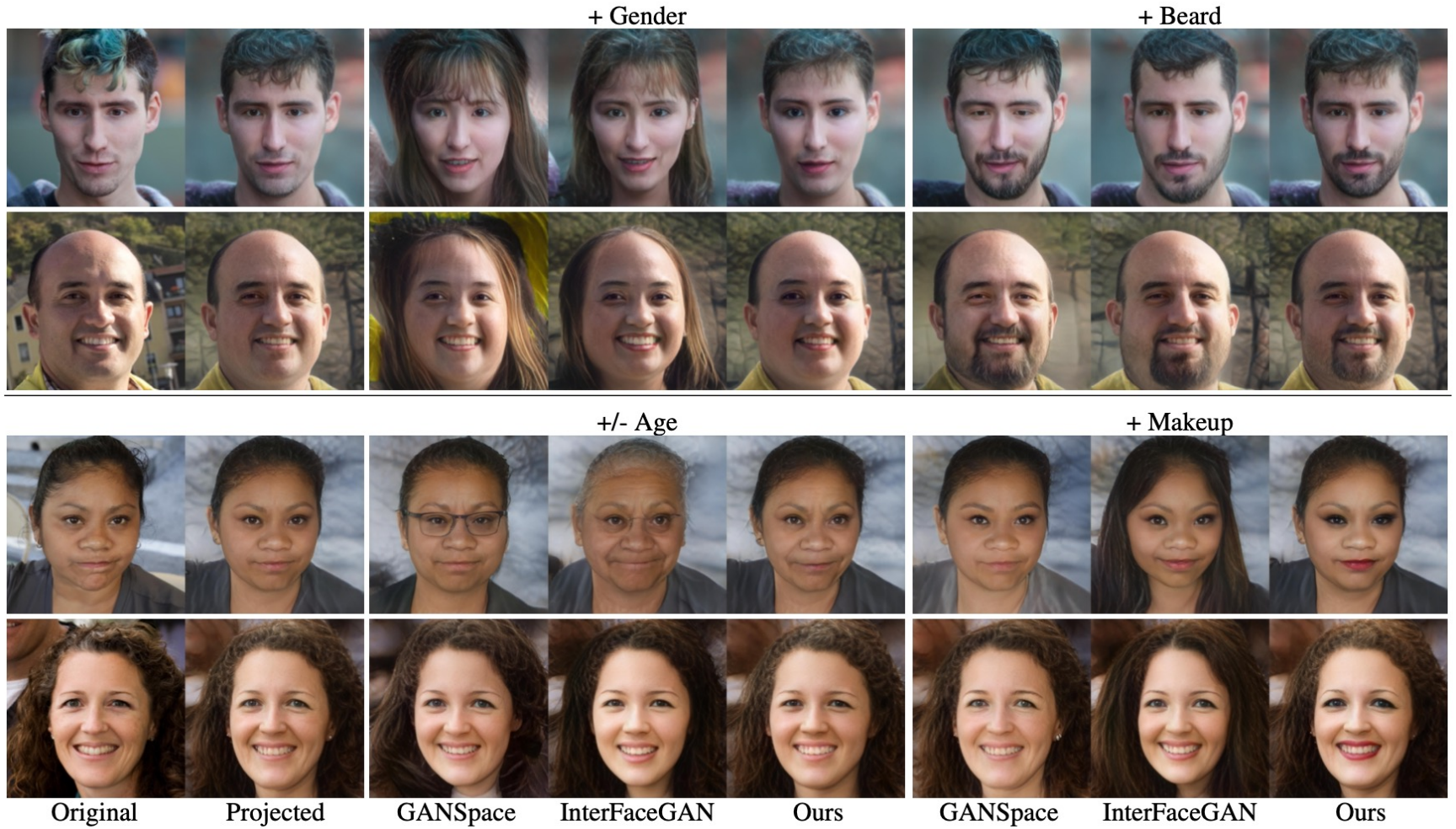


Figure 3: **Disentangled facial attribute editing on real images.** The first two columns show the original image and the projected image reconstructed with the encoded latent code in StyleGAN. From the 3rd column in each subfigure, from left to right are the manipulation result of GANSpace [18], that of InterFaceGAN [34] and ours. Compared to recent approaches, our method achieves a controllable, disentangled and realistic editing, where the person's identity is preserved.

Latent Transformer – Quantitative Comparison

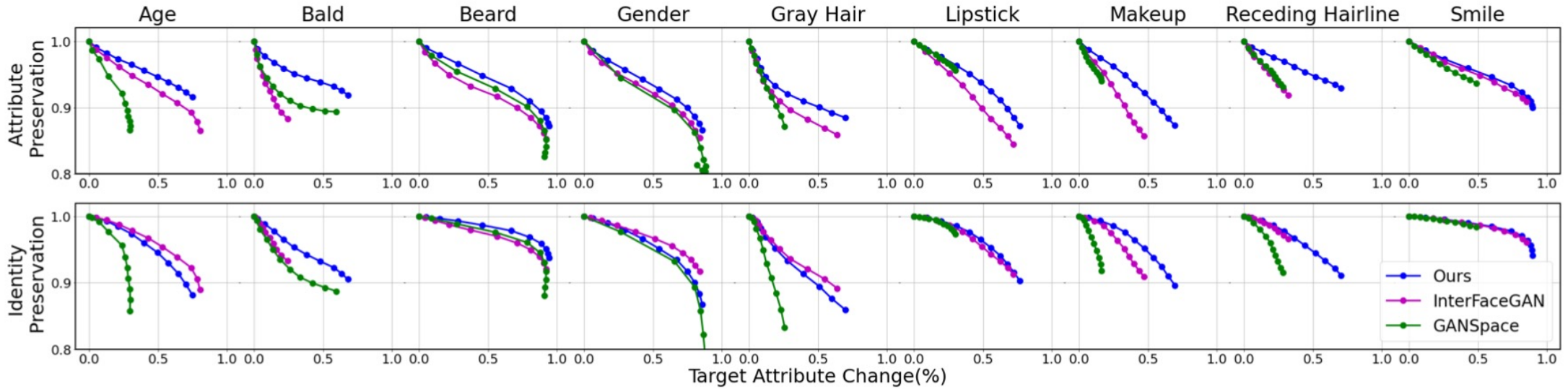


Figure 4: **Attribute and identity preservation vs. target attribute change.** For each method, we edit each target attribute with 10 different scaling factors ($\{0.2 \cdot d, 0.4 \cdot d, ..., 2 \cdot d\}$, d is the magnitude of change suggested in each method), and generate the modified images. Attribute preservation rate and identity preservation score are measured on the output images. In the figure, each point corresponds to a scaling factor, where the position x indicates the target attribute change rate (the fraction of the samples with target attribute successfully changed among all the manipulations). In the upper sub-figure, the position y indicates the average attribute preservation rate on the other attributes. In the bottom sub-figure, the position y indicates the average identity preservation score. Ideally, we want higher attribute and identity preservation for the same amount of change on the target attribute (higher curve is better).