Conditional Generation Using Polynomial Expansions

Grigorios G. Chrysos[1], Markos Georgopoulos[2], Yannis Panagakis[3]

[1] Ecole Polytechnique Federale de Lausanne (EPFL), [2] Imperial College London, [3] University of Athens Greece

Motivation

- Polynomial networks (PNs) have demonstrated impressive results in image generation [4, 2].
- However, in conditional image generation we have two (or more) input variables, instead of a polynomial expansion of a single variable.
- Our model, called CoPE, expresses a high-degree, multivariate polynomial for conditional data generation. We exhibit how CoPE can be applied on five diverse conditional generation tasks.

Method

- In conditional generation, we have (at least two) input vectors $\boldsymbol{z}_{\text{I}}, \boldsymbol{z}_{\text{II}} \in \mathbb{R}^d$. We want to learn a function $G(\boldsymbol{z}_{\text{I}}, \boldsymbol{z}_{\text{II}})$ to approximate the target function.
- The typical approach is to concatenate the two vectors either in the input or the feature space. However, this captures only a linear correlation between the two vectors as we argue.
- Instead, we want to use an alternative approximator, i.e. polynomial expansions. We define the recursive form:

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \left(\boldsymbol{U}_{[n,I]}^T \boldsymbol{z}_{\text{I}} + \boldsymbol{U}_{[n,II]}^T \boldsymbol{z}_{\text{II}}\right) * \boldsymbol{x}_{n-1}, \quad (1)$$

for $n = 2, \ldots, N$ with $\boldsymbol{x}_1 = \boldsymbol{U}_{[1,I]}^T \boldsymbol{z}_{\text{I}} + \boldsymbol{U}_{[1,II]}^T \boldsymbol{z}_{\text{II}}$ and $\boldsymbol{x} = \boldsymbol{C}\boldsymbol{x}_N + \boldsymbol{\beta}$. The vector $\boldsymbol{\beta}$ and the matrices $\boldsymbol{C} \in \mathbb{R}^{o \times k}, \boldsymbol{U}_{[n,\phi]} \in \mathbb{R}^{d \times k}$ for $n = 1, \ldots, N$ and $\phi = \{I, II\}$ are learnable.

- The symbol '$*$' refers to an elementwise product.

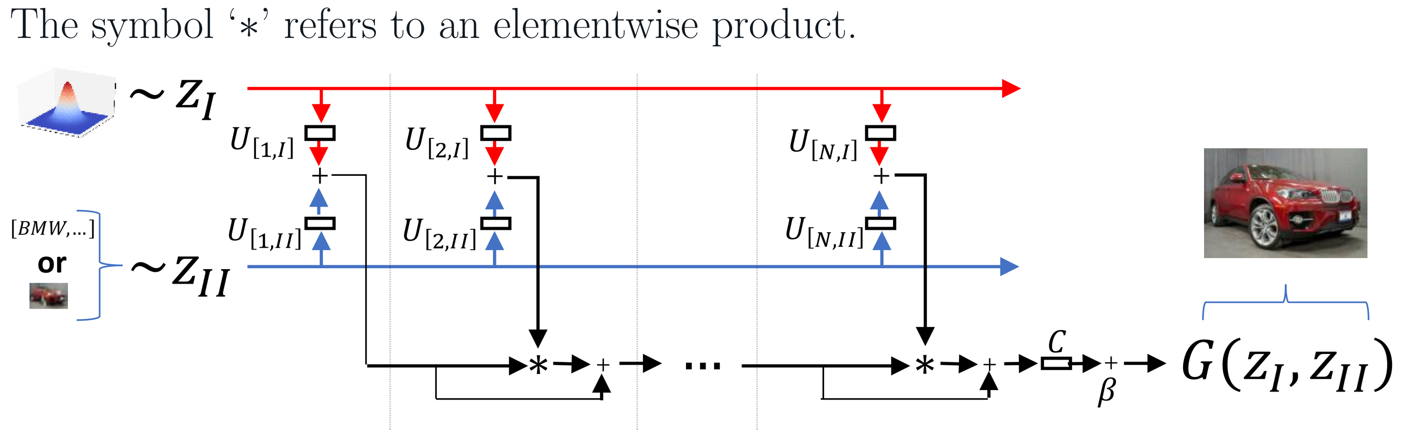


Fig. 1: Schematics of CoPE.

Fig. 2: Source code

Edge-to-image generation

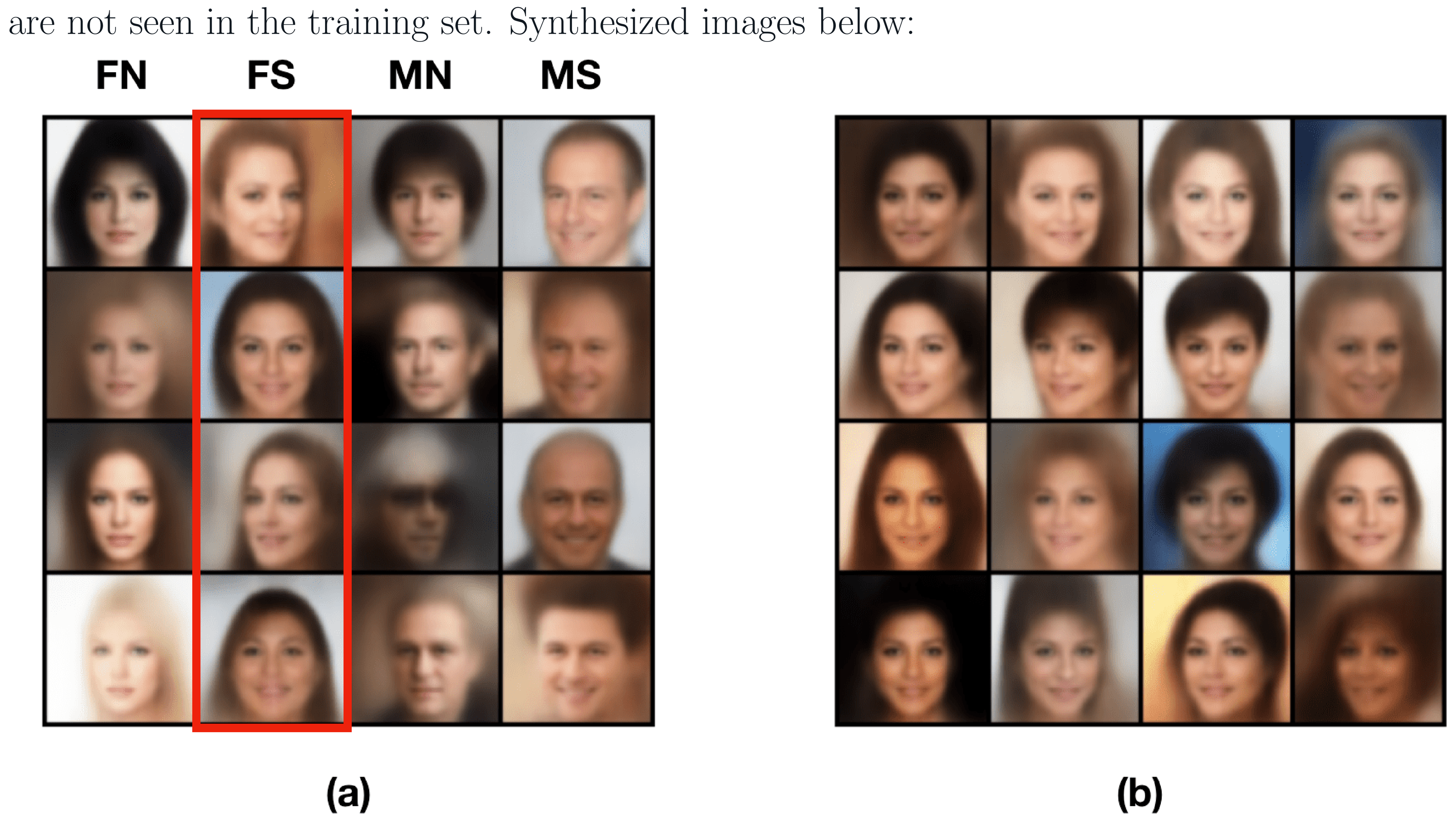
- We train polynomial generators with linear blocks, i.e. ditching the activation functions between the layers, in a GAN setting:

Fig. 3: The first row depicts the conditional input (i.e., the edges). The rows 2-6 depict outputs when we vary $\boldsymbol{z}_I$ (i.e., noise).

- Contrary to previous works, **only the GAN minmax objective is used** without any additional losses and without any additional networks.
- The generator without activation functions between the layers can learn the data distributions.

Generation of unseen attribute combinations

We assess the performance in the multi-label setting of [3], where one (or more) combinations are not seen in the training set. Synthesized images below:



In (a), all combinations are illustrated (the red is the combination missing during training, i.e. Female+Smile), while in (b), only images from the missing combination are visualized.

Class-conditional generation

Quantitative evaluation on class-conditional generation with SNGAN-based [5] generator:

class-conditional generation on CIFAR10		
Model	Inception Score (↑)	Frechet Inception Distance (↓)
SNGAN	8.30 ± 0.11	14.70 ± 0.97
SNGAN-CONC	8.50 ± 0.49	30.65 ± 3.55
SNGAN-ADD	8.65 ± 0.11	15.47 ± 0.74
SNGAN-SPADE	8.69 ± 0.19	21.74 ± 0.73
SNGAN-CoPE	$\mathbf{8.77 \pm 0.12}$	$\mathbf{14.22 \pm 0.66}$
BigGAN [1]	-	14.70

The baselines SNGAN-CONC, SNGAN-ADD are constructed by changing the Hadamard product to concatenation and addition respectively. SNGAN-SPADE adapts SPADE [6] for class-conditional generation. Notice that the proposed SNGAN-CoPE outperforms all the compared methods, even larger models.

Acknowledgements

This project was sponsored by the Department of the Navy, Office of Naval Research(ONR) under a grant number N62909-17-1-2111.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale gan training for high fidelity natural image synthesis". In: *ICLR*. 2019.

[2] Grigorios Chrysos et al. "Π−nets: Deep Polynomial Neural Networks". In: *CVPR*. 2020.

[3] Markos Georgopoulos et al. "Multilinear Latent Conditioning for Generating Unseen Attribute Combinations". In: *ICML*. 2020, pp. 3442–3451.

[4] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *CVPR*. 2019.

[5] Takeru Miyato et al. "Spectral normalization for generative adversarial networks". In: *ICLR*. 2018.

[6] Taesung Park et al. "Semantic image synthesis with spatially-adaptive normalization". In: *CVPR*. 2019, pp. 2337–2346.