

Image를 2D array가 아닌 coordinate의 RGB 예측인 INR(Implicit neural representation) 방식

### \* Introduction

이미지를 2D array 대신 pixel coord로 접근.

$F(p) = v$ ,  $p = (x, y)$ ,  $v = (r, g, b) \Rightarrow$  2D Signal의 양자화 crop인 image 대신, continuous를 제시한다.

$F(p)$ 는 정확히 알 수 없기 때문에 근사되어야 하는데, 이를 위해 3D에서는 비전이 너무 많이 드는

INR(Implicit Neural Representation)을 사용합니다.

Decoder 설계에는 두 가지 선택肢이 있다.

1. 다른 network의 디렉션 param을 생성하는 network는 훨씬 더 불안정하고, param이 너무 많이 필요하다.

2. MACs가 너무 많아 들판.

### 해결법

1. FMM(factorized multiplication modulation)

2. multi-scale INR

INR 가진 decoder의 특성

- 이미지 병합 외상 : 훈련 때만 image 'zoom-out' 가능

- Geometric prior : latent의 거리별로 흡수 encoding

- 높은 저해상도 image 출력 : 훈련된 데이터의 낮은 해상도를 높여가며 생성할 수 있다.

- Image interpolation.

- Super-resolution : 훈련 때만 SR 가능

요약하면, hypernetwork를 위한 FMM(INR 생성 및 훈련 인수화), 고해상도를 위한 multi-scale INR,

즉시 사용이 가능한 SR, extrainterpolaion, inference, interpolate, geometric prior.

## \* Related Work

### - INR

좌표 정보로 NN을 augment 하는 것은 CPNN에서 제시되었다.

다른 많은 방법이 있지만, INR이 3D에서 가장 많이 쓰인다.

mesh, voxel, point cloud 외 절감 표현, 3D shape flow, scene, audio 등의 여러 곳에서 쓰인다.

Occupancy network은 3D shape를 초기화된 Voxel의 확률 함수로 modeling 한다.

일반적으로 단일 image-view에서 작동하는 coordinate-based decoder를 사용한다.

논문에서 제시된 방법은 인접 pixel의 계산을 공유하고, surface extract로 예측을 구현하였다.

INR은 위치 좌표를 encoding하는 것이 중요하다.

↳ fourier를 많이 사용한다.

### - Generative model + coordinates

INR과 GAN을 결합한 모델들이 있다.

IM-Net은 autoencoder의 latent의 generative model을 훈련한다.

논문에서는 latent를 coord-based decoder에 넣는 대신, hypernetwork-based generator로 param을 생성해준다.

Coordinate GAN은 DCGAN의 representation의 coord를 concat한다.

SBD는 coord info로 VAE를 augment 한다.

Spatial VAE는 COCO-GAN의 회전 및 변환을 추가하여, patch 생성 후 조합

### - Hypernetworks.

Hypernetwork는 meta-model의 다른 model의 param을 양자화 network다.

이런 parameterization은 meta-model을 통한 weight sharing으로 높은 expressivity와 compression을 제공한다.

factorized multiplicative modulation (FMM)은 squeeze & excitation과 비슷하다.

하지만 FMM은 hidden representation 대신, weight를 modulate 한다.

Hypernetwork는 훈련이 불안정하다고 알려져 있지만, 몇몇 초기 방식들은 이를 위해 초기화 방식들을 제안한다.

하지만 논문에서는 FMM이 INR 내부의 signal propagation을 hypernet의 초기화의 영향을

회피적으로 훈련하기 때문에 필요없다.

Hypernetwork는 NAS, few-shot 등 여러 곳에서도 쓰인다.

#### - Hypernetwork + Generative model.

HyperGAN 등에서는 classification을 위한 GAN model을 구축했다.

HyperVAE는 주어진 sample distribution의 generative model param을 생성함으로써

target distribution의 encoding을 설정한다.

HCNAF는 conditional autoregressive flow model의 param을 위한 hypernetwork

PI-GAN은 INR form으로 3D object를 생성하고, output을 압축하기 위해 mFiLM을 사용한다.

#### - Hypernetwork + INRs

#### - Computationally efficient model

INR-based decoder는 가장 conv가 비해 계산량이 적다.

각 INR layer는 1x1 conv로 볼 수 있다. 다른 점은 weight가 hypernetwork의 G로 생성된다.

INR의 많은 param을 가지기 때문에, low-rank로 분해한다. (근사)

본 논문에서는 특별한 이유이 아니라 low-rank matrix 두개를 만들기 때문이다.  $W = A * B$

SENet은 squeeze and excitation으로 train을 더 안정화시키고 빠르게 수행하게 한다.

## \* Image meta-generation

### - Model Overview

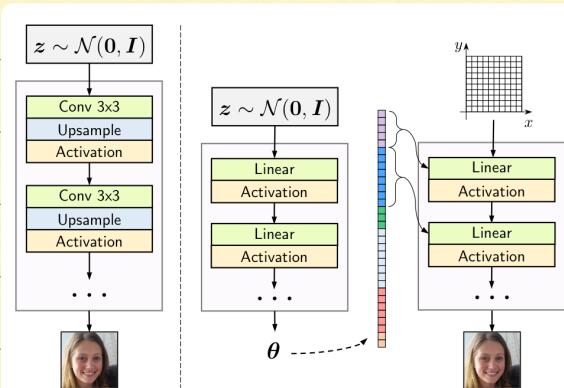


Figure 1: Comparison between a traditional convolutional generator (left) and an INR-based one (right). A traditional generator directly generates a pixel-based image representation given its latent code  $z$ . The INR-based one produces parameters of an MLP. The corresponding pixel-based representation is obtained by evaluating the INR at each coordinate location  $(x, y)$  of a specified grid.

StyleGAN2의 G를 바탕으로 한 나머지는 같다

Hypernetwork-based G는  $z \sim \mathcal{N}(0, I)$ 를 입력으로 받고, INR model  $F_\theta$ 를 위한 param  $\theta$ 를 생성한다.

정확한 image 생성을 위해 dataset 크기에 의해 결정되는 시그마가 정의된다.

모든 coordinate의 param  $F_\theta$ 를 평가한다. (ex)  $256^2 = 65536$

그리고  $(x, y)$ 를 embed 한다. 특히 최근에 개시된 fourier feature를 사용한다.

fourier feature는 linear + sine이다.

$u = \sin(rU_p)$  ,  $p = \phi(y)$ 를  $u$ 로 mapping.

embedding matrix  $U$ 는 G의 의해 예측됨.

내가 만든 image의 가장 적합한 feature  $f$ 를 선택하여 model의 유연성 제공

## - FMM

프로젝터 INR의 param을 생성하고 싶다.

$F_\theta$ 의  $l$ -layer 모드는  $W^l \in \mathbb{R}^{n_{\text{in}} \times n_{\text{out}}}$ ,  $b^l \in \mathbb{R}^{n_{\text{out}}}$

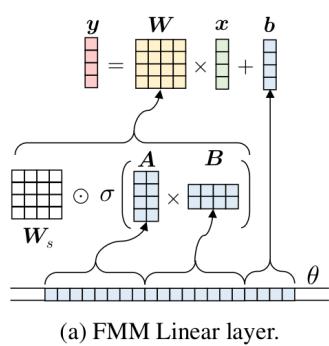
바로 적용시키는 것은  $(\text{layer}) + \text{hidden}$  을 가지고 있다면  $h \times (n_{\text{in}} \times n_{\text{out}} + n_{\text{out}})$  개의 parameter가 필요할 것이다.

$W^l = A^l \times B^l$  은 low-rank로 분해하는 방법은 singular value를 0이 되는 것과 동일한 방법이 사용이 가능하다.

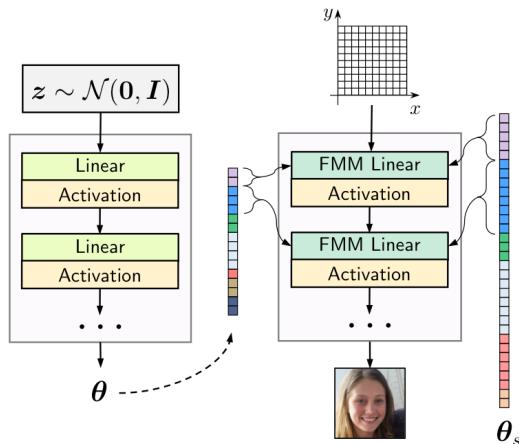
↳ 한방향으로 sensitive하게 만들기

↳ rank가 줄어들 = 방향이 줄어들

또한 full-rank와 shared를 parameterization



(a) FMM Linear layer.



(b) INR-based generator with FMM.

Figure 6: (a) FMM linear layer for inputs  $x$ , shared matrix  $W_s$  and output  $y$ ; (b) INR-based generator with the FMM mechanism: its parameters are split into  $\theta_s$  (shared) and  $\theta$  (predicted by the hypernetwork). This mechanism makes our architecture be similar to StyleGAN2 [40]: the hypernetwork becomes a *mapping network* and the INR  $F_\theta$  becomes the *synthesis network* (decoder).

[학습] 단계에서 모든 sample이 대해 공유하는  $W_s \in \mathbb{R}^{n_{out} \times n_{in}}$  은 정의 (learnable)

Hypernet works:  $A \in \mathbb{R}^{n_{out} \times r}$ ,  $B \in \mathbb{R}^{r \times n_{in}}$  을 생성한다.  $W_h^l = A^l \times B^l$

$$\text{최종 } W^l = W_s^l \circ \sigma(W_h^l)$$

Sigmoid를 통과하는 것이 activation을 bound해주기 때문에 좋다. (stable)

### - Multi-scale INR

Traditional INR은 scaling rate는 같은 계산이 너무 많이 필요하다.

↳ (1) 대량의 1D input 들이 있다.

↳ Batch size가 너무 커서 large hidden layer를 놓고 싶은

Multi-scale INR:  $F_\theta$ 를  $k$ 개로 split

각 block은 각각 input에 대해 작동하고, 마지막은 target에 대해 작동한다.

low-resolution 계산으로 다음 해상도로 복구된다.

낮은 해상도에서 다양한 뉴런을 사용

↑ 각 pixel은 이전 layer로부터 필요한 context를 받음

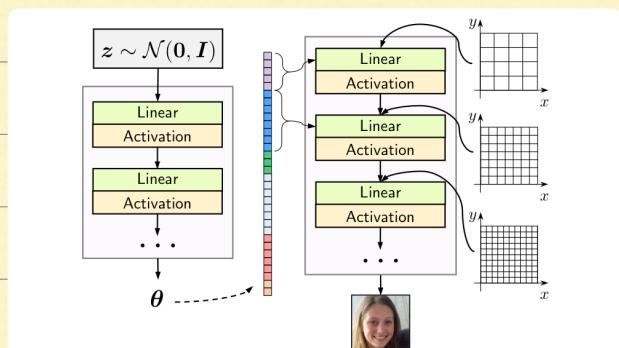


Figure 7: Multi-Scale INR-based GAN (without FMM).  
Each block operates on a different resolution, determined by the granularity of an input grid. We increase the granularity with depth: this allows to share computation between neighbouring pixels *and* condition them on a common context. We depict the multi-scale mechanism without FMM not to clutter the illustration. In practice, we use both FMM and the multi-scale architecture for our INR-GAN.

64<sup>1</sup> 블록 MLP로 3<sup>1</sup> block은 2x4x4x layer로 2x2x2로 학습, fourier features는 3<sup>1</sup> block의 MLP로 concat FID.

모든 high-res 이미지에 대해 nearest neighbor interpolation을, upsampling을 사용한다.

## \* Experiments

- standard GAN training

Baseline:

1. Non-hypernetwork based G:  $w = G(z)$  condition으로 학습.

↳ 공유되는 모든 INR param

↳  $f_\theta$ 의 다른 param을 생성하는 대신, 조건으로  $v = f_\theta(x, y, w)$

2. hypernetwork based G:  $F_\theta$ 를 위한 theta를 생성하지만, factorization X

Baseline의 Fourier PEel, FMM, INR은 포함된다.

G는 4-layer MLP,  $z \sim \mathcal{N}(0, I^5)$

DE styleGANv2

## \* Result.

Decoder type	LSUN 128 <sup>2</sup>			LSUN 256 <sup>2</sup>			FFHQ 1024 <sup>2</sup>		
	GMACs	#params	FID	GMACs	#params	FID	GMACs	#params	FID
Latent-code conditioned INR decoder [44, 3]	30.09	7.1M	229.9	120.33	7.1M	253.3	1925.22	7.1M	O/M
+ Hypernetwork-based decoder [51]	23.54	2055.5M	28.83	88.01	2055.5M	O/M	1377.52	2055.5M	O/M
+ Fourier embeddings from [74, 81] (ours)	28.02	2312.02M	23.07	105.34	2312.02M	O/M	1651.52	2312.02M	O/M
+ Factorized Multiplicative Modulation (ours)	25.87	108.2M	11.51	103.19	108.2M	15.68	1649.37	108.2M	O/M
+ Multi-Scale INR (ours)	21.58	107.03M	5.69	38.76	107.03M	6.27	47.35	117.3M	16.32
StyleGAN2 generator [39]	-	-	-	84.36	30.03M	2.65	143.18	30.37M	4.41

1. real vs image ratio

2. FID 차이

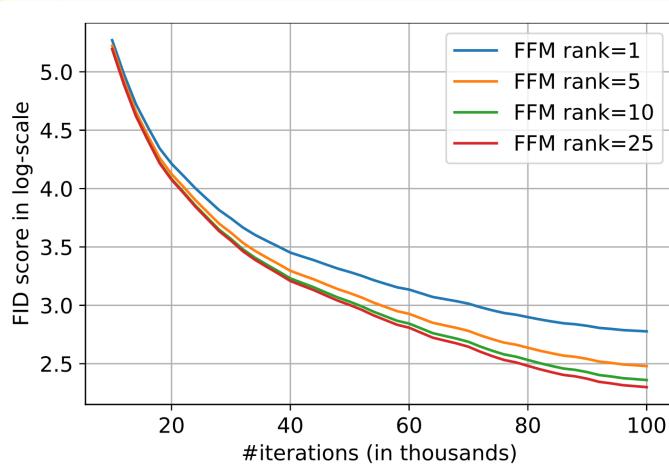
3. parameter 차이

46. infer, FID = 15

Graph paramol 높았을 때. direct inference는 좋음

- Ablating FMM

rank 10 이상은 overparametrized (overparametrize)



- Additional ablation.

FMM의 σ가 높을 때 영향을 미친다 → 표 2

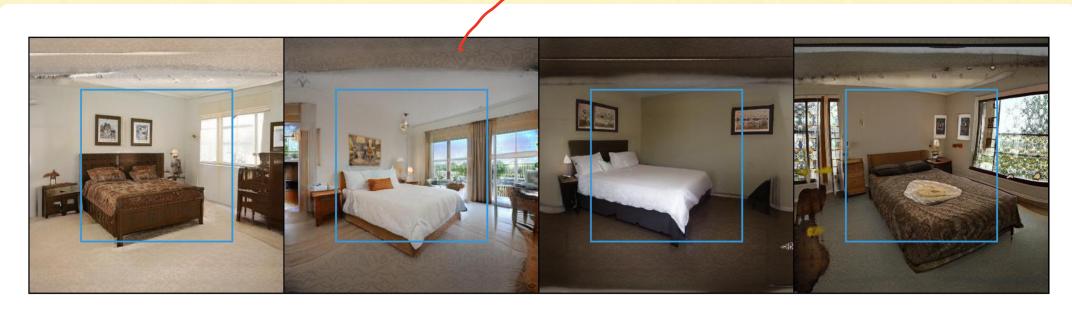
- Multi class

Table 4: FID & IS at 300k iterations on multi-class datasets.

Decoder type	LSUN-10		MiniImageNet	
	FID ↓	IS ↑	FID ↓	IS ↑
Basic INR decoder	216.8	1.0	271.5	1.03
+ Hypernetwork-based decoder	OOM	OOM	112.9	8.76
+ Fourier embeddings	OOM	OOM	102.8	9.85
+ FMM	23.78	2.48	84.66	9.32
+ Multi-scale INR	12.47	3.02	59.63	11.29
StyleGAN2	8.99	3.18	52.94	12.32
Validation set	0.42	9.93	0.39	61.79

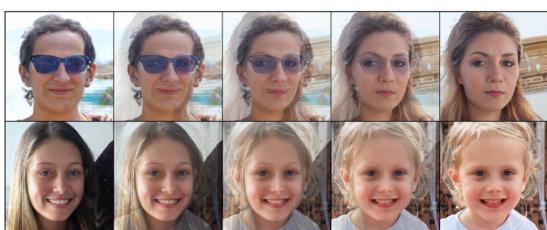
~ Exploring properties.

- Extrapolating



⇒ 그림에 대한 일반화가 잘됨!

- interpolating



(a) Image interpolation in the pixel-based form.



(b) Image interpolation in the INR-based form.

⇒ 합리적!

- keypoint prediction.

- Accelerating infer

Traditional Grid은 많은 계산을耗费하는 low-res infer로 인해 느려

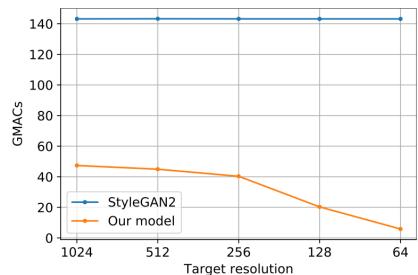


Figure 9: **Accelerated low-resolution image generation.**  
We measure a decoder’s efficiency in terms of #MACs on generating an image of lower resolution compared to what it has been trained on. Since INR can do this by evaluating on a sparser grid, this allows it to save a lot of computation. Traditional convolutional decoders require performing a full inference first and then downsampling the produced image.

- Out-of-box SR



\* Additional potential

- Ability to backpropagate through pixel position

- Faster infer speed.

- parallel pixel context.

\* Limitation.

high-freq coordinate artifact 142