

Panorama 생성과 전이 차트를 통한

* Introduction.

인간의 시각 능력은 주변에 제한된 CL. (시야)

따라서 사람은 시야들을 합쳐 환경 정보를 인식한다. (시야는 연속적이고 달연을)

이런 현실에서 입력 이미지를 한꺼번에 넘는 다른 model과 다른 이미지 차트가 해당하는 곳을 생성하고 싶다는 블루스 (Local view)

이미지를 동일한 subprocess로 나누어 생성하는 방식

latent distribution manifold), coordinate manifold의 징후를 누워도

latent dim G는 각 patch 구조를 고려하고 이를 통한 patch를 생성한다.

D는 patch가 구조적으로 좋고, 시각적으로 좋으며, 각 patch의 가로세로를 예측하는 역할을 한다.

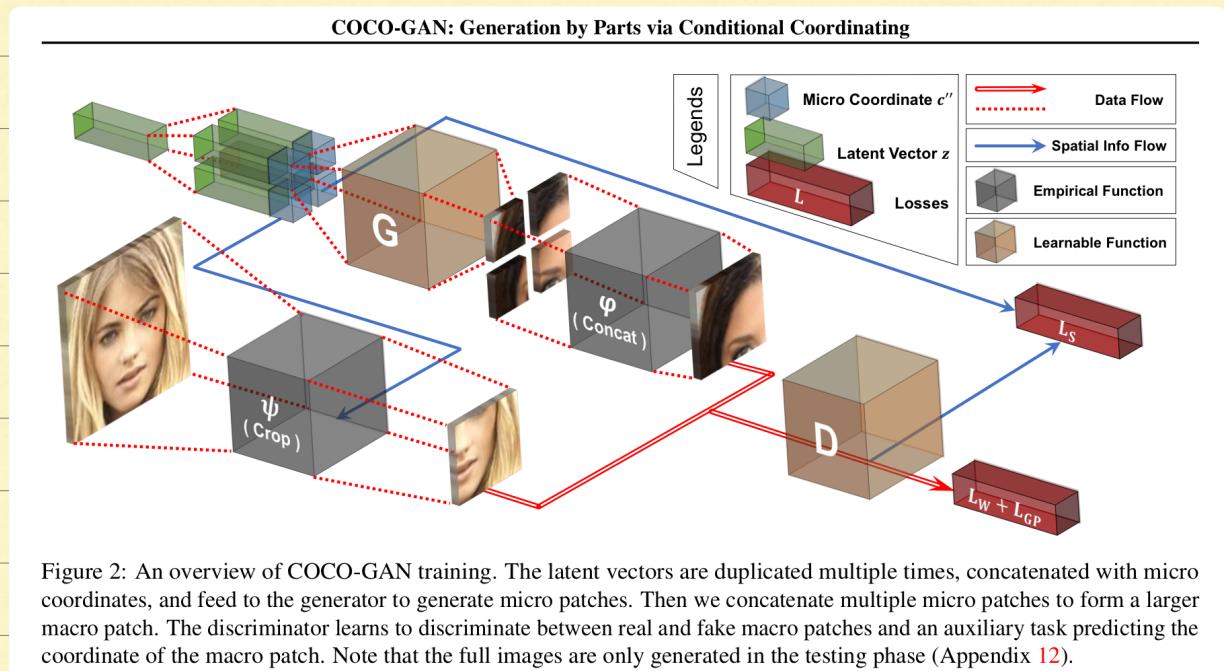


Figure 2: An overview of COCO-GAN training. The latent vectors are duplicated multiple times, concatenated with micro coordinates, and feed to the generator to generate micro patches. Then we concatenate multiple micro patches to form a larger macro patch. The discriminator learns to discriminate between real and fake macro patches and an auxiliary task predicting the coordinate of the macro patch. Note that the full images are only generated in the testing phase (Appendix 12).

G가 다양한 patch를 생성하도록 한다.

16x4 patch는 128x128 1024x1024 가로 만들 수 있다.

실제 train set은 156x256 이지만 284x384로 생성하는 수 있다.

↳ beyond-boundary generation

장점.

1. panorama 생성 가능

2. part별 생기는 별도로 안내 기능이 필요

3. latent vector를 24-288는 extra prediction branch를 더함으로써, patch가 다른 실체에 따라 사용

↳ = patch-guided generation

대부분 생성기 동작하고 enc-dec가 있는 구조가 모두 적용 가능

* COCO-GAN

Network는

Grt D, I2I2. fine-grain coordinate G_c , coarse-grain coordinate D.

image는

full size, macro size (x', s'), micro size (s'')로 이루어짐.

$C'' \rightarrow$ micro coordinate 조건을 통해 공간 기여도

$$S'' = G_c(z, C'') \text{ 은 } \text{상수}$$

Ψ 는 image merging function.

L_{overlapping}이 필요없음.

D의 훈련은 위해 Ψ 의 Ψ 도 학습되야 한다.

patch size가 이전이나 같은 이미지와 같은 이미지와 다른 patch를 합침에 D의 넓는다.

Loss는 Wasserstein loss L_w , patch gradient loss L_{gp} , spatial consistency loss L_s 를 쓴다.

$$\begin{cases} L_w + \lambda L_{gp} + \alpha L_s & \text{for } D \\ -L_w + \alpha L_s & \text{for } G \end{cases}$$

- Spatial Coordinate System.

Coordinate system은 G를 위한 micro et D를 위한 macro인 걸까.

merging system 위의 것처럼, \tilde{c} macro coordinate $C_{(i,j)}$ 는

micro coordinate $C''_{(i,j)} = [C''_{(i:i+N, j:j+M)}]$ 과 관련 있다.

G는 micro condition c'' 의 다른 s'' 를 만들고자.

같은 latent z를 넣고, C'' 의 다른 다른 이미지를 만들고자 한다.

C'' 의 조건은 s'' 들이 서로 연결된 patch으로 출력되게.

그리고 그 $\psi(s'')$ 로 합쳐진다.

이제 합쳐진 s'' 에 대한 c'' 를 넣어라. (macro)

real로부터는 바로 x' 를 얻는다.

ψ 는 concat 하는 역할을 한다.

infer에서는 micro는 전부 이미지가 대상인 coordinate를 추출할 수 있다.

COCO-GAN: Generation by Parts via Conditional Coordinating

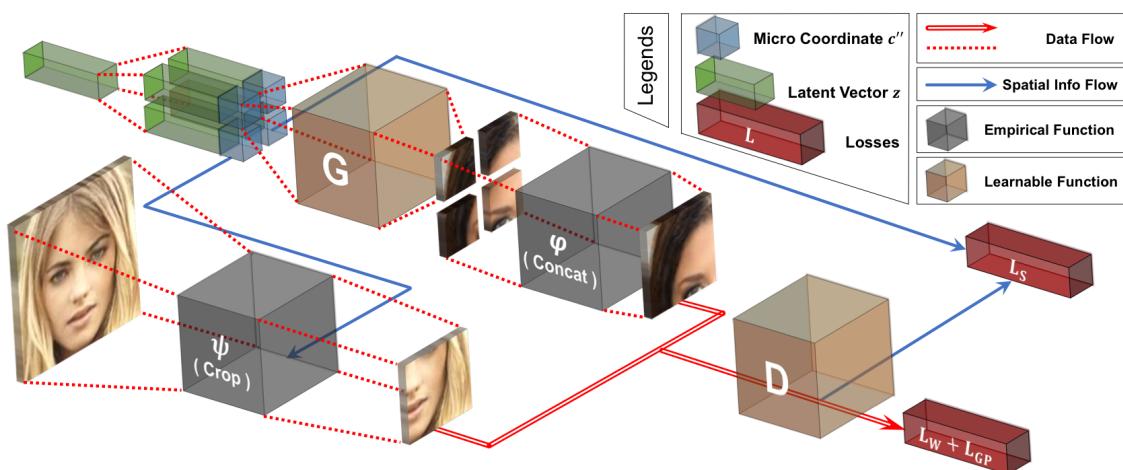


Figure 2: An overview of COCO-GAN training. The latent vectors are duplicated multiple times, concatenated with micro coordinates, and feed to the generator to generate micro patches. Then we concatenate multiple micro patches to form a larger macro patch. The discriminator learns to discriminate between real and fake macro patches and an auxiliary task predicting the coordinate of the macro patch. Note that the full images are only generated in the testing phase (Appendix 12).

- Loss function.

\mathcal{L}_w 는 macro level의 loss입니다.

$$\mathcal{L}_w = \mathbb{E}_{x,c} [D(\psi(x, c))] - \mathbb{E}_{z,c} [D(\psi(G(z, c)))]$$

$$\mathcal{L}_{GP} = \mathbb{E}_{\hat{s}} [(\|\nabla_{\hat{s}} D(\hat{s})\|_2 - 1)^2] \quad , \quad \hat{s}' = \epsilon s' + (1-\epsilon)x'$$

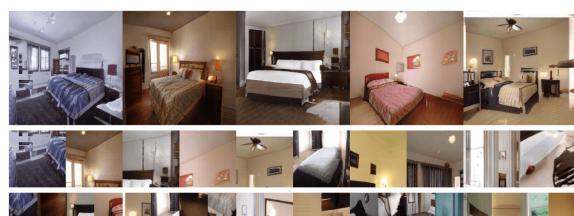
$$\mathcal{L}_s = \mathbb{E}_c [\|c' - A(x')\|_2]$$

↳ macro의 차트 예제. A 는 loss function.

* Experiments.



(a) CelebA (N2,M2,S32) (full image: 128×128).



(b) LSUN bedroom (N2,M2,S64) (full image: 256×256).

Figure 3: COCO-GAN generates visually smooth and globally coherent full images without any post-processing. The three rows from top to bottom show: (a) the generated full images, (b) macro patches, and (c) micro patches. For the first five columns, each column uses the same latent vector, e.g., the leftmost full image (first row), the leftmost micro patch (second row), and the leftmost macro patch (third row) share the same latent vector. Note that the columns are not aligned due to different sizes. More results can be found in the Appendix F.

128x128x3 1/16 scale로 2x 생성됨



(a) CelebA (N4,M4,S16) (full image: 128×128, FID: 9.99).



(b) CelebA (N8,M8,S8) (full image: 128×128, FID: 15.99).

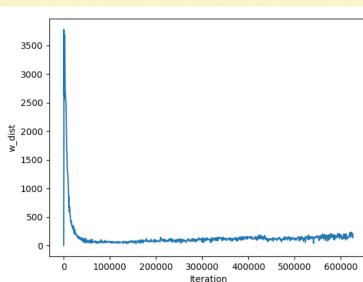


(c) CelebA (N16,M16,S4) (full image: 128×128, FID: 23.90).

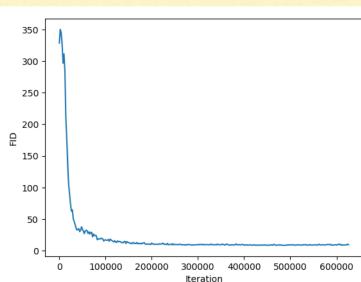
⇒ micro-L 4x4 인 헌법의 미적 질을 증명

Dataset	CelebA 64×64	CelebA 128×128	LSUN Bedroom 64×64	LSUN Bedroom 256×256	CelebA-HQ 1024×1024
DCGAN [26] + TTUR [13]	12.5	-	57.5	-	
WGAN-GP [12] + TTUR [13]	-	-	9.5	-	-
IntroVAE [14]	-	-	-	8.84	-
PGGAN [15]	-	7.30	-	8.34	7.48
Proj. D [22] (our backbone)	-	19.55	-	-	-
Ours (N2,M2,S32)	4.00	5.74	5.20	5.99*	9.49*

⇒ FID 결과는?



(a) Wasserstein distance



(b) FID

⇒ 훈련은 stable이다.

- latent space continuity

full image interpolation & coordinate interpolation 훈련 진행

1. full-image interpolation.

patch 훈련 image) 및 그에 따른 full-image의 interpolation이 이루어지면,

모든 patch의 내용이 함께 변화하는 현상.

(Full image interpolation) 모든 patch를 ("full" 영역)에 적용한 image가 생기는.



Figure 5: The results of full-images interpolation between two latent vectors show that all micro patches are changed synchronously in response to the change of the latent vector. More interpolation results are available in Appendix G.

2. Coordinates interpolation.

각각의 latent vector coordinate interpolation.

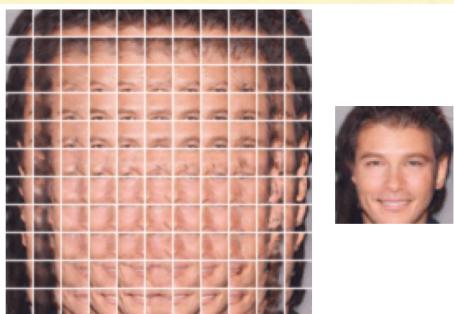


Figure 6: An example of spatial coordinates interpolation showing the spatial continuity of the micro patches. The spatial coordinates are interpolated between range $[-1, 1]$ of the micro coordinate with a fixed latent vector. More examples are shown in Appendix I.

COCO-GAN의 coord는 이산적이기 때문에 늘 시야의 미간의 흔적은 알지 못하지만

직접 빈대는 번영하여 진현된다.

↳ 고정점에서는 물체의 인라인에 잘 표시할 수도 있다.

↳ 미간을 생생하게 표현하는 뜻인듯.

- Beyond-boundary Generation.

COCO-GAN의 경우 넓은 training image보다 큰 이미지를 만들 수도 있다. (여기서 원본 색깔은 이미지)

넓은 GAN은 latent dimension extrapolation을 가능화한, COCO-GAN은 coordinate manifold을 확장화.

하지만 patch는 연속성은 보장되지 않기 때문에 이를 위해 보정이 필요하다.

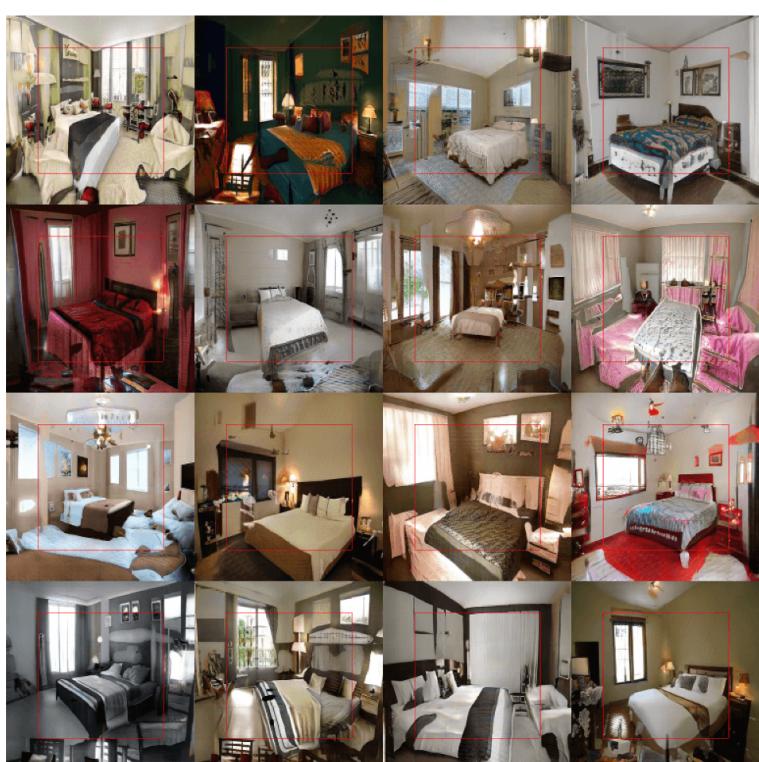
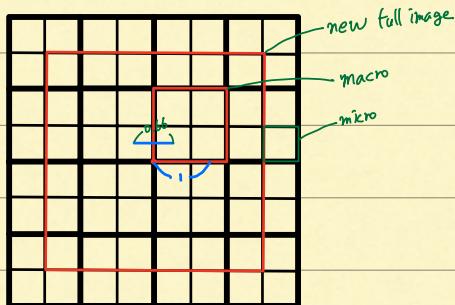


Figure 7: “Beyond-Boundary Generation” generates additional contents by extrapolating the learned coordinate manifold. Note that the generated samples are 384×384 pixels, whereas **all** of the training samples are of a smaller 256×256 resolution. The **red** box annotates the 256×256 region for regular generation without extrapolation.

넓고의 이미지는 $[-1, 1]$ 사이에 생성되는지만, 노동으로 extrapolation을 허용, 주변부 생성이 가능하다

하지만 artifact가 있으며, 가변로는 이미지를 구해, 추가적인 훈련을 시킨다.



$$\therefore c'' \in [-1.66, 1.66], c' \in [-2, 2]$$

또한 tanh activation 때문에 A 는 $[-1, 1]$ 범위 내에서 훈련 시키지 않는다.

Gelu 기능은 layer만 훈련시키면 됨으로 고려

▷ 디 훈련시키면 artifact가 생김.

▷ 더욱 연구

- Panorama & Partial Generation.

COCO-GAN 경우 카메라를 학습한 대로 cycle topology of panorama image를 생성할 수 있습니다.



Figure 8: The generated panorama is cyclic in the horizontal direction since COCO-GAN is trained with a cylindrical coordinate system. Here, we paste the same generated panorama twice (from 360° to 720°) to better illustrate the cyclic property of the generated panorama. More generation results are provided in Appendix H.

\Rightarrow 수평 방향 카메라 $[-1, 1]$ 의 $[cos\theta, sin\theta]$ 를 바탕으로 하면

\Rightarrow Matterport 3D 이미지 학습했다.

\Rightarrow URDF 사용 가능

- Patch-guided image generation.

DOI latent을 추적하는 network Q를 달아서, 원래 patch를 생성할 수도 있고,

전체가 아니라 생성이 학습되었던, 보호하고 일관된 이미지를 생성할 수 있다.

↳ 이전의 복원 및 inpainting의 활용 가능

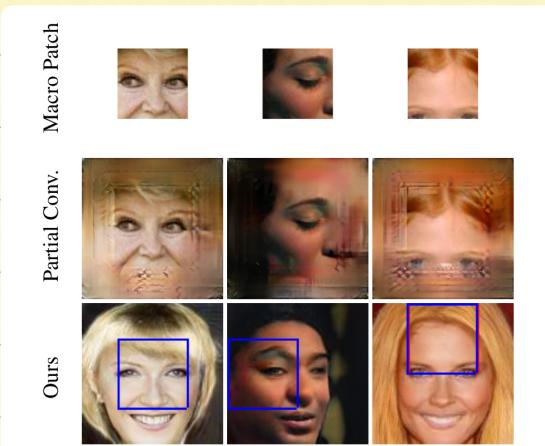


Figure 9: Patch-guided image generation loosely retains the local structures from the original image and make the full image still globally coherent. The quality outperforms the partial convolution [17]. The blue boxes visualize the predicted spatial coordinates $A(x')$, while the red boxes indicate the ground truth coordinates c' . Note that the generated images are **not** expected to be identical to the original real images. More examples are provided in Appendix K.

- Computation friendly Generation

이 super resolution은 0.021을 기준 bottleneck을 찾는다

(COCO-GAN이라는 부분적으로 높아 때문에 더 많은 계산이 필요하다.

- Ablation study

Model	best FID (150 epochs)
COCO-GAN (cont. sampling)	6.13
COCO-GAN + optimal D	4.05
COCO-GAN + optimal G	6.12
Multiple G	7.26
COCO-GAN (N2,M2,S16)	4.87

Table 2: The ablation study shows that COCO-GAN (N2,M2,S16) can converge well with little trade-off in convergence speed on CelebA 64×64 dataset.

Optimal은 전부나 이전거를 선택

Optimal G,D의 경우 전 훈련에 대해서 좋은 성능.

당연히 제일단 COCO-GAN의 성능이 좋다.

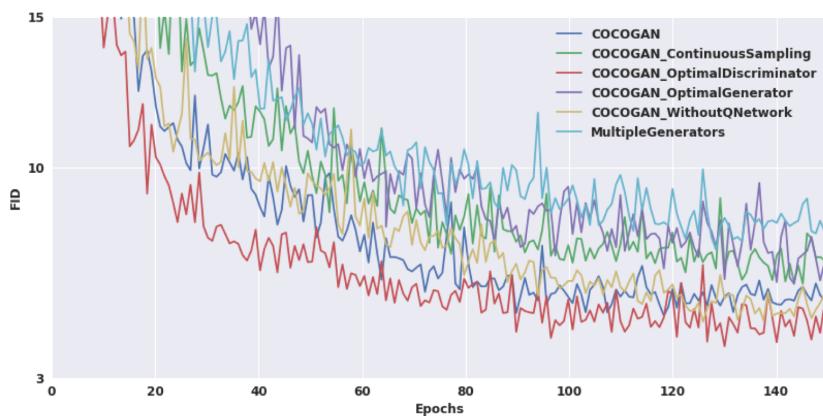


Figure 26: FID score curves of different variants of COCO-GAN in CelebA 64×64 setting. Combined with Figure 27, the results do not show significant differences in quality between COCO-GAN variants. Therefore, COCO-GAN does not pay significant trade-off for the conditional coordinate property.

✗ Appendix

A. COCO-GAN during Testing Phase

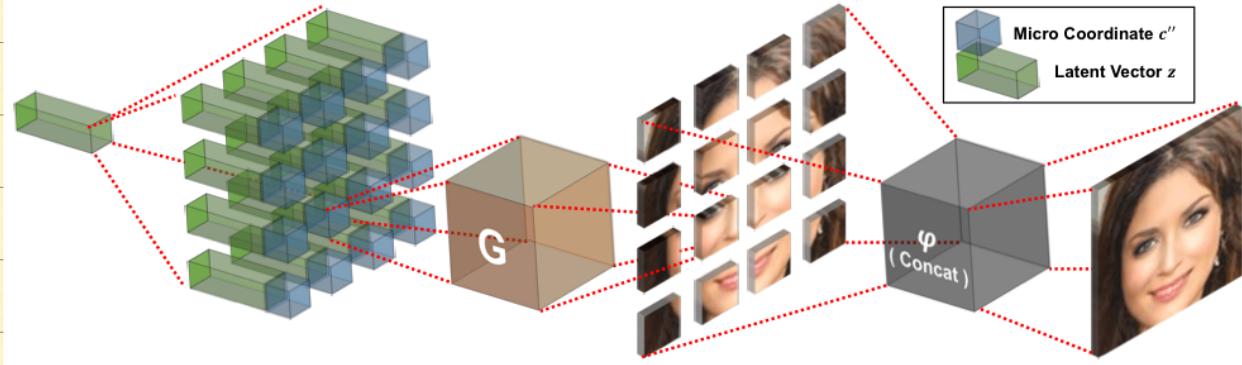


Figure 12: An overview of COCO-GAN during testing phase. The micro patches generated by G are directly combined into a full image as the final output.

B. Symbols

Group	Symbol	Name	Description	Usage
Model	G	Generator	Generates micro patches.	$s'' = G(z, c'')$
	D	Discriminator	Discriminates macro patches.	$D(\varphi(G(z, C'')))$
	A	Spatial prediction head	Predicts coordinate of a given macro patch.	$\hat{c}' = A(x')$
	Q	[†] Content prediction head	Predicts latent vector of a given macro patch.	$z_{est} = Q(s')$
Heuristic Function	φ	Merging function	Merges multiple s'' to form a s' or s .	$s' = \varphi(G(z, C''))$
	ψ	Cropping function	Crops x' from x . Corresponding to φ .	$x' = \psi(x, c')$
Variable	z	Latent vector	Latent variable shared among s'' generation.	$s'' = G(z, c'')$
	z_{est}	[†] Predicted z	Predicted z of a given macro patch.	$L_Q = \mathbb{E}[\ z - z_{est}\ _1]$
	c'	Macro coordinate	Coordinate for macro patches on D side.	$L_S = \mathbb{E}[\ c' - \hat{c}'\ _2]$
	c''	Micro coordinate	Coordinate for micro patches on G side.	$s'' = G(z, c'')$
	\hat{c}'	Predicted c'	Coordinate predicted by A with a given x' .	$L_S = \mathbb{E}[\ c' - \hat{c}'\ _2]$
	C''	Matrix of c''	The matrix of c'' used to generate S'' .	$s' = \varphi(G(z, C''))$
Data	x	Real full image	Full resolution data, never directly used.	$x' = \psi(x, c')$
	x'	Real macro patch	A macro patch of x which D trains on.	$adv_{x'} = D(\psi(x, c'))$
	s'	Generated macro patch	Composed by s'' generated with C'' .	$adv_{s'} = D(s')$
	s''	Generated micro patch	Smallest data unit generated by G .	$s'' = G(z, c'')$
	S''	Matrix of s''	Matrix of s'' generated by C'' .	$S'' = G(z, C'')$
	\hat{s}'	Interpolated macro patch	Interpolation between random x' and s' .	$\hat{s}' = \epsilon s' + (1 - \epsilon) x', \text{ which } \epsilon \sim [0, 1]$
Loss	L_W	WGAN loss	The patch-level WGAN loss.	$L_W = \mathbb{E}[D(x')] - \mathbb{E}[D(s')]$
	L_{GP}	Gradient penalty loss	The gradient penalty loss to stabilize training.	$L_{GP} = \mathbb{E}[(\ \nabla_{\hat{s}'} D(\hat{s}')\ _2 - 1)^2]$
	L_S	Spatial consistency loss	Consistency loss of coordinates.	$L_S = \mathbb{E}[\ c' - A(x')\ _2]$
	L_Q	[†] Content consistency loss	Consistency loss of latent vectors.	$L_Q = \mathbb{E}[\ z - Q(s')\ _1]$
Hyper-parameter	α	Weight of L_S	Controls the strength of L_S (we use 100).	$\begin{cases} L_W + \lambda L_{GP} + \alpha L_S, & \text{for } D, \\ -L_W + \alpha L_S, & \text{for } G. \end{cases}$
Testing Only	s	Generated full image	Composed by s'' generated with C''_{Full} .	$s = \varphi(G(z, C''_{Full}))$
	C''_{Full}	Matrix of c'' for testing	The matrix of c'' used during testing.	$s = \varphi(G(z, C''_{Full}))$

[†] Only used in “Patch-Guided Image Generation” application.

- Architecture

Backbone은 projection GAN과 같음.

모든 convolutional spectral Norm을 사용함.

- Conditional BN

coordinate와 latent를 위한 $\text{MLP}_\mu(z, c)$, $\text{MLP}_\sigma(z, c)$ 를 생성함.

$$\alpha_k = ((\hat{\omega}_k - \mu_k) / \sigma_k) \approx r + \beta$$

- hyperparam.

$$\lambda=10, d=100, \text{Adam}=(0, 0.999)$$

$$\text{lr} = \text{Two Time-scale Update Rule. (TTUR)} \Rightarrow G: 0.0001, D: 0.0004$$

- Coordinate setup

$$C', C'' = [-1, 1], z = [-1, 1]$$

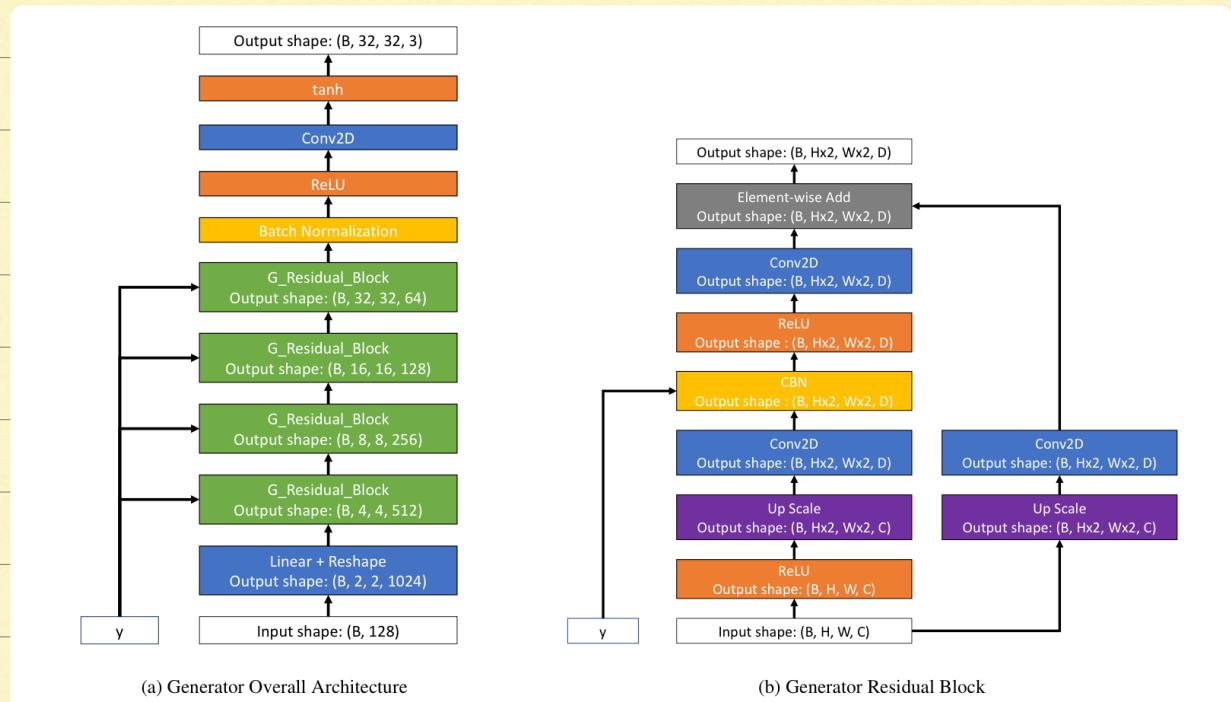


Figure 13: The detailed generator architecture of COCO-GAN for generating micro patches with a size of 32×32 pixels.

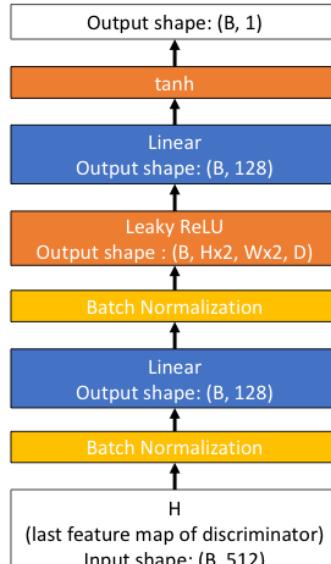
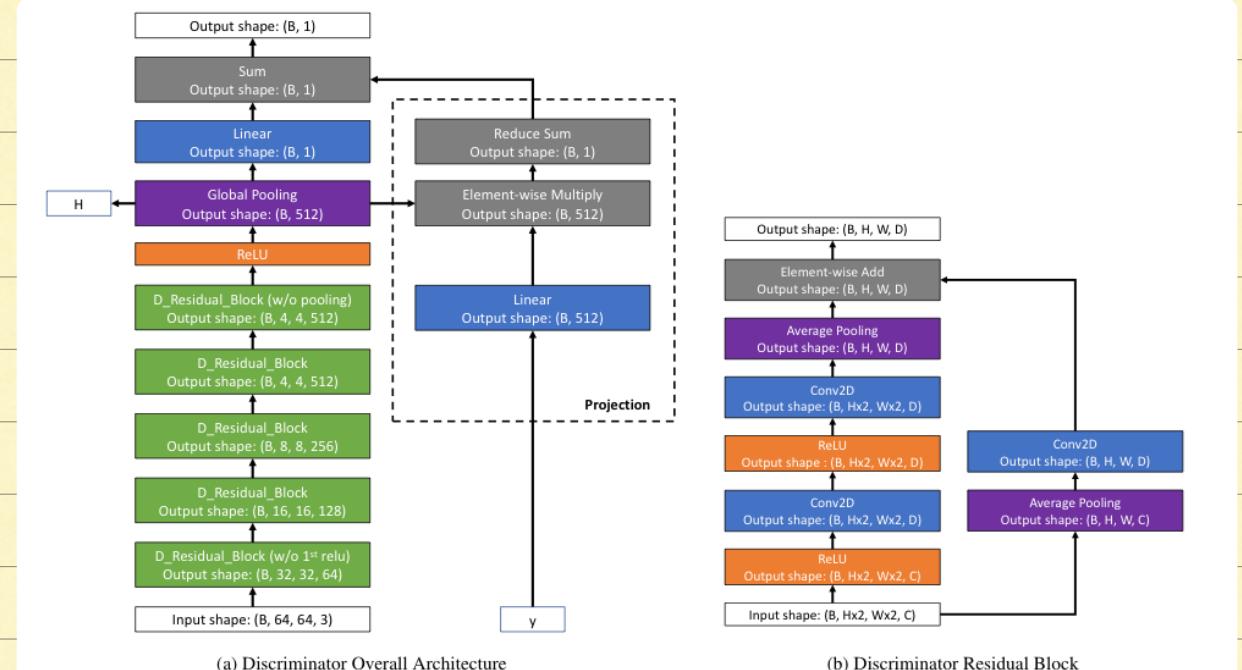


Figure 14: The detailed discriminator architecture of COCO-GAN for discriminate macro patches with a size of 64×64 pixels. Both the content vector prediction head (Q) and the spatial condition prediction head use the same structure shown in (c).