Large Scale Adversarial Representation Learning

Jeff Donahue and Karen Simonyan
DeepMind

Compiled by Hongming Shan

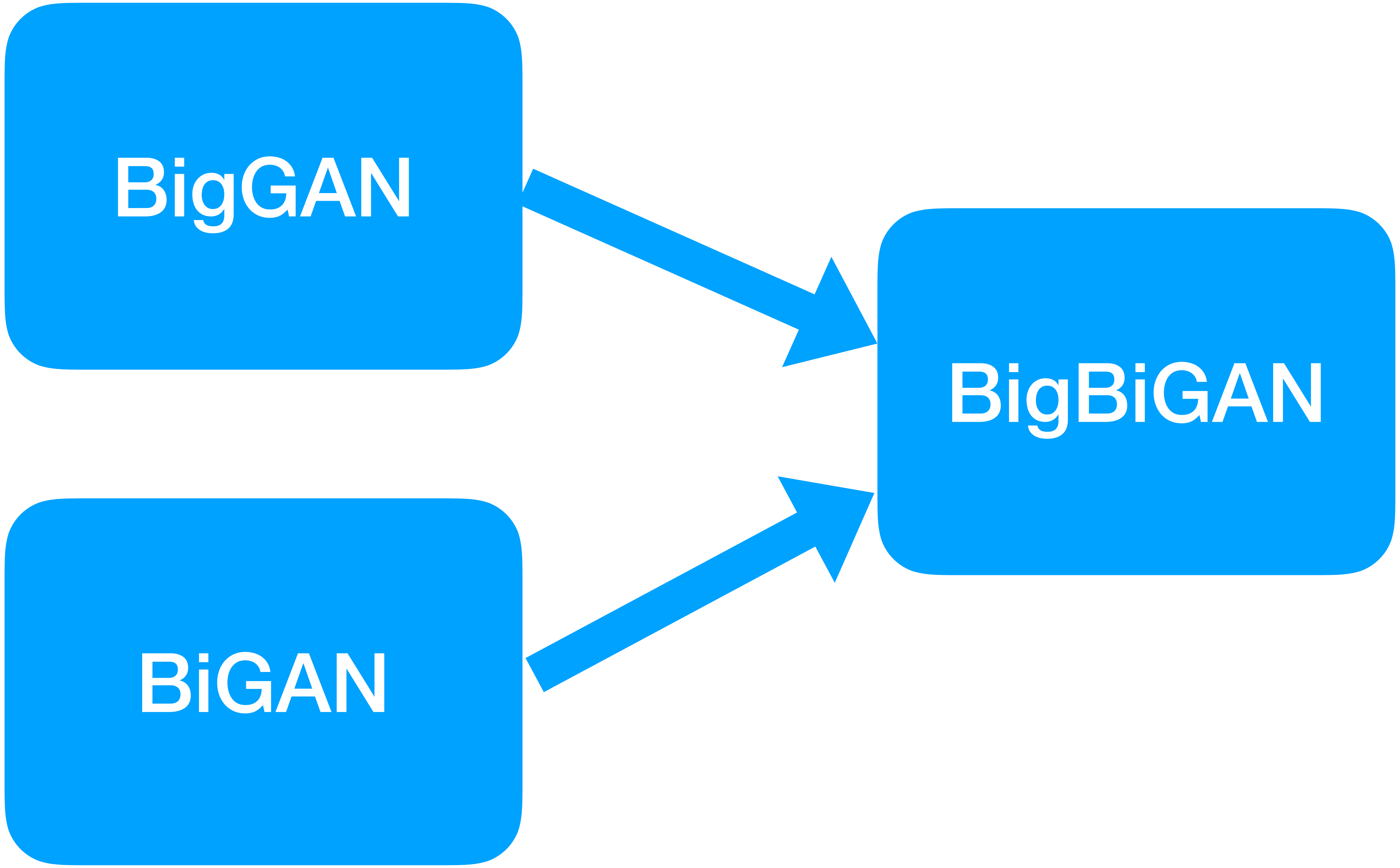
BigBiGAN

- Adversarially trained generative models (GANs) have recently achieved compelling image synthesis results. But despite early successes in **using GANs for unsupervised representation learning**, they have since been **superseded** by approaches based on self-supervision.

- In this work we show that progress in image generation quality translates to substantially improved representation learning performance.

- Our approach, **BigBiGAN**, builds upon the state-of-the-art **BigGAN** model, extending it to representation learning by **adding an encoder** and **modifying the discriminator**.

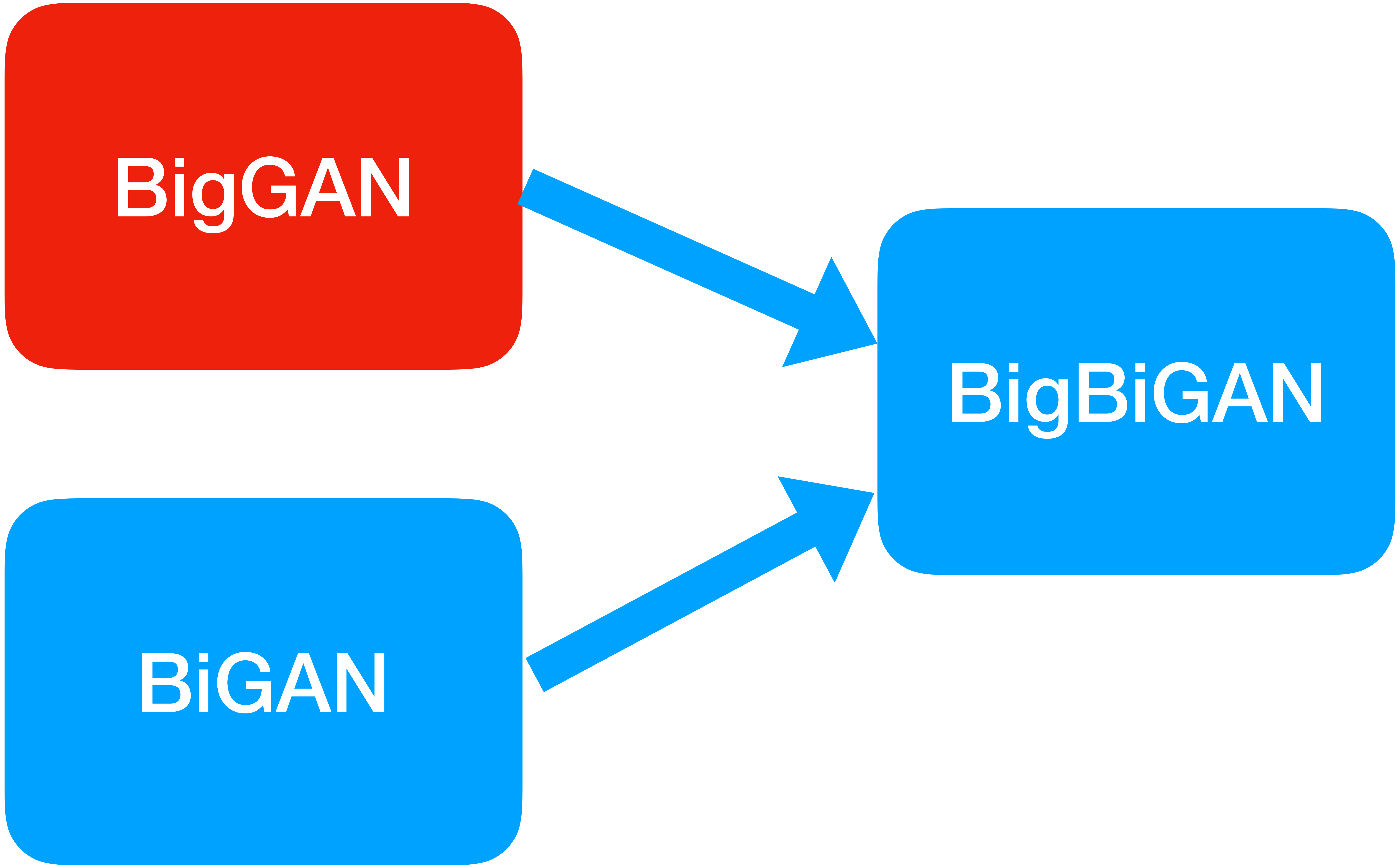
- We extensively evaluate the representation learning and generation capabilities of these BigBiGAN models, demonstrating that these generation-based models achieve the state of the art in unsupervised representation learning on ImageNet, as well as in unconditional image generation.



BigGAN

BiGAN

BigBiGAN



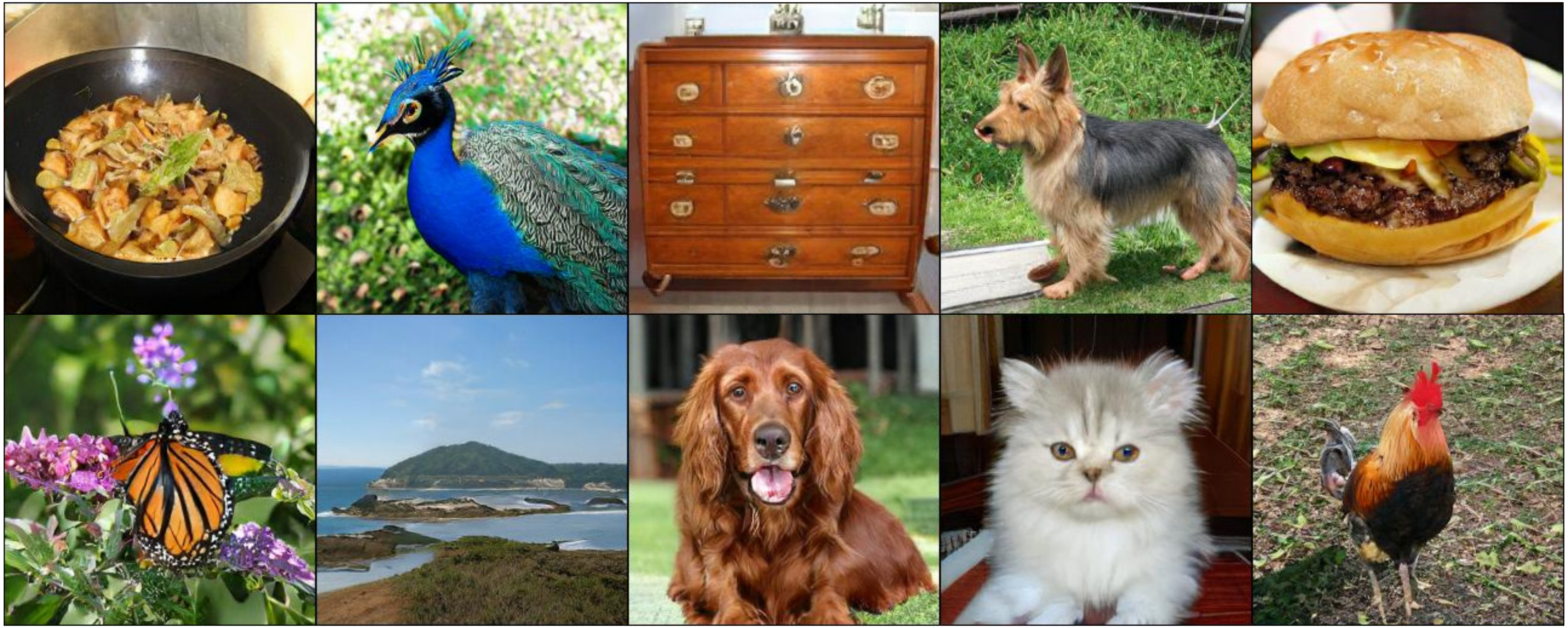
BigGAN

BiGAN

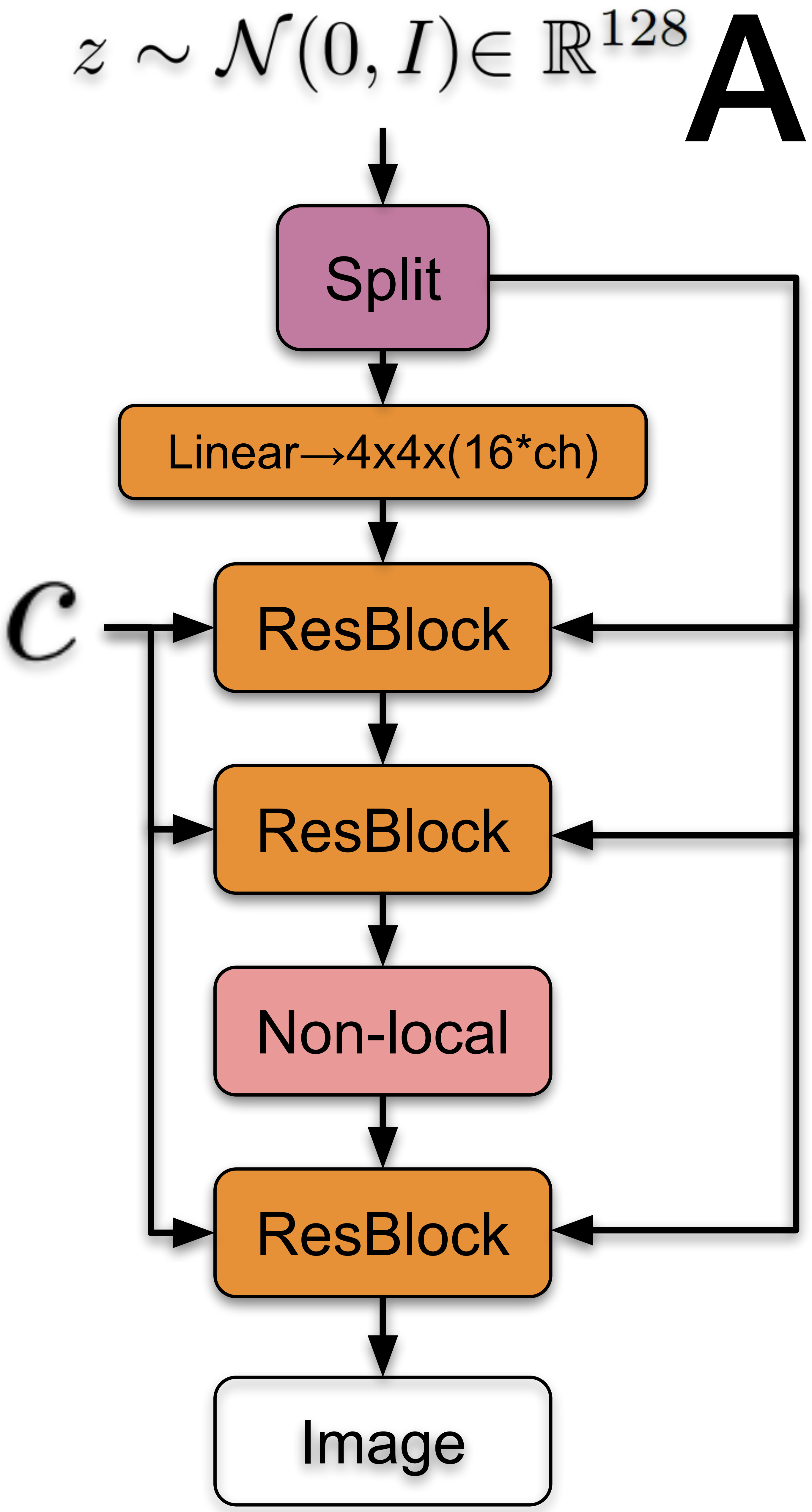
BigBiGAN

BigGAN

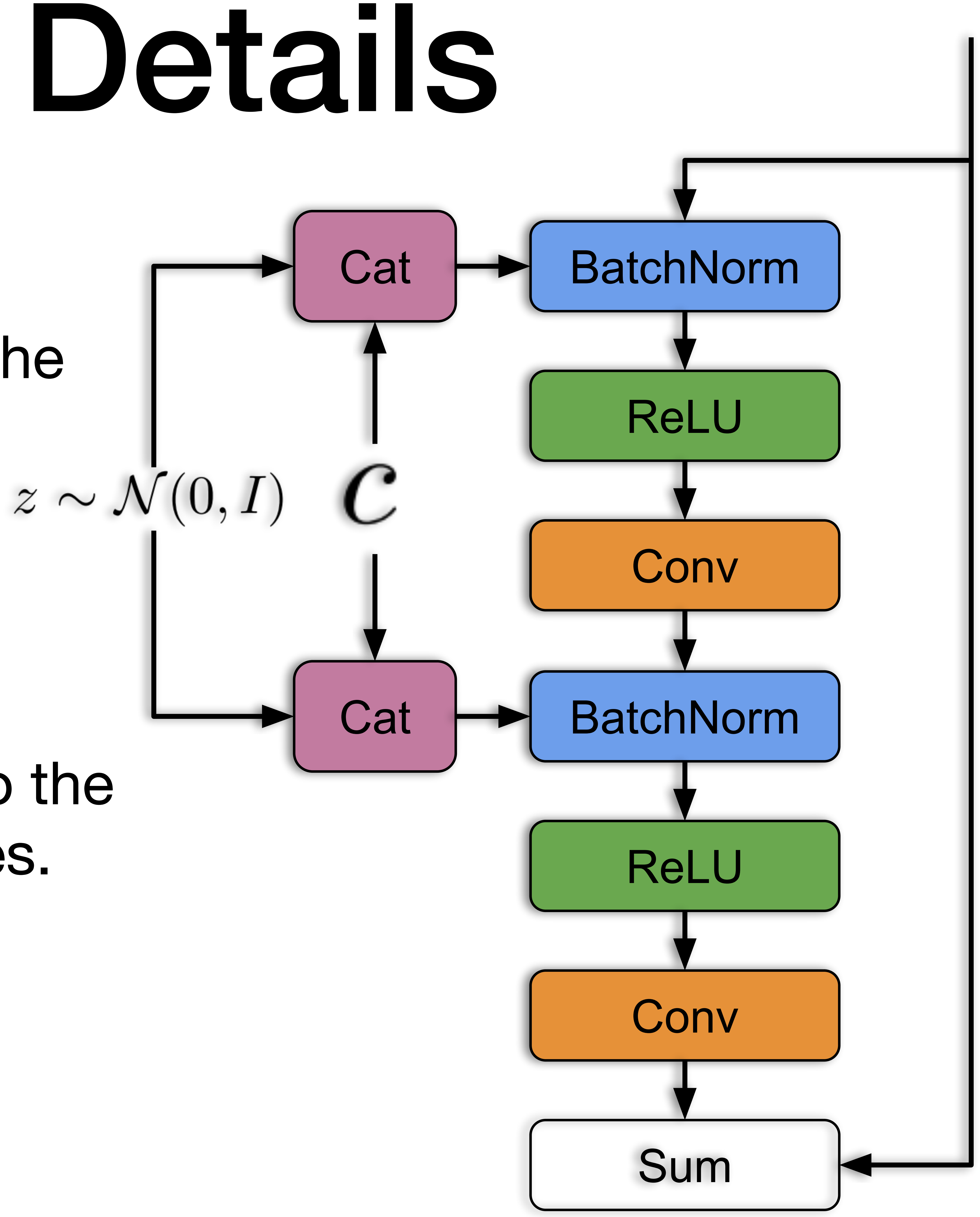
- Presented on Oct. 10, 2018!



Architectural Details



- (a) A typical architectural layout for G ; details are in the following tables.
- (b) A Residual Block in G .
- c is concatenated with a chunk of z and projected to the BatchNorm gains and biases.
- Different resolutions used different structures



Architectures

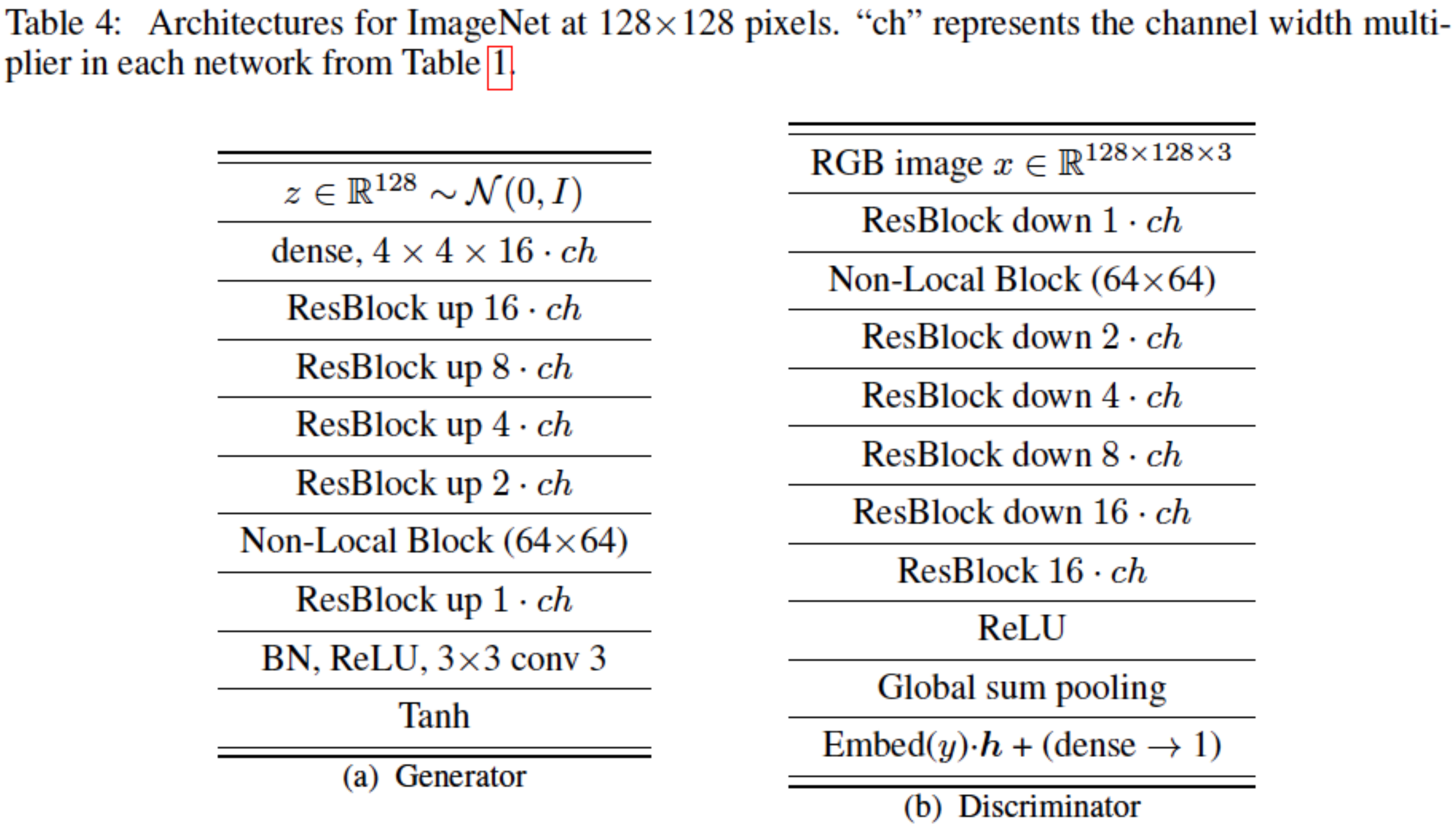


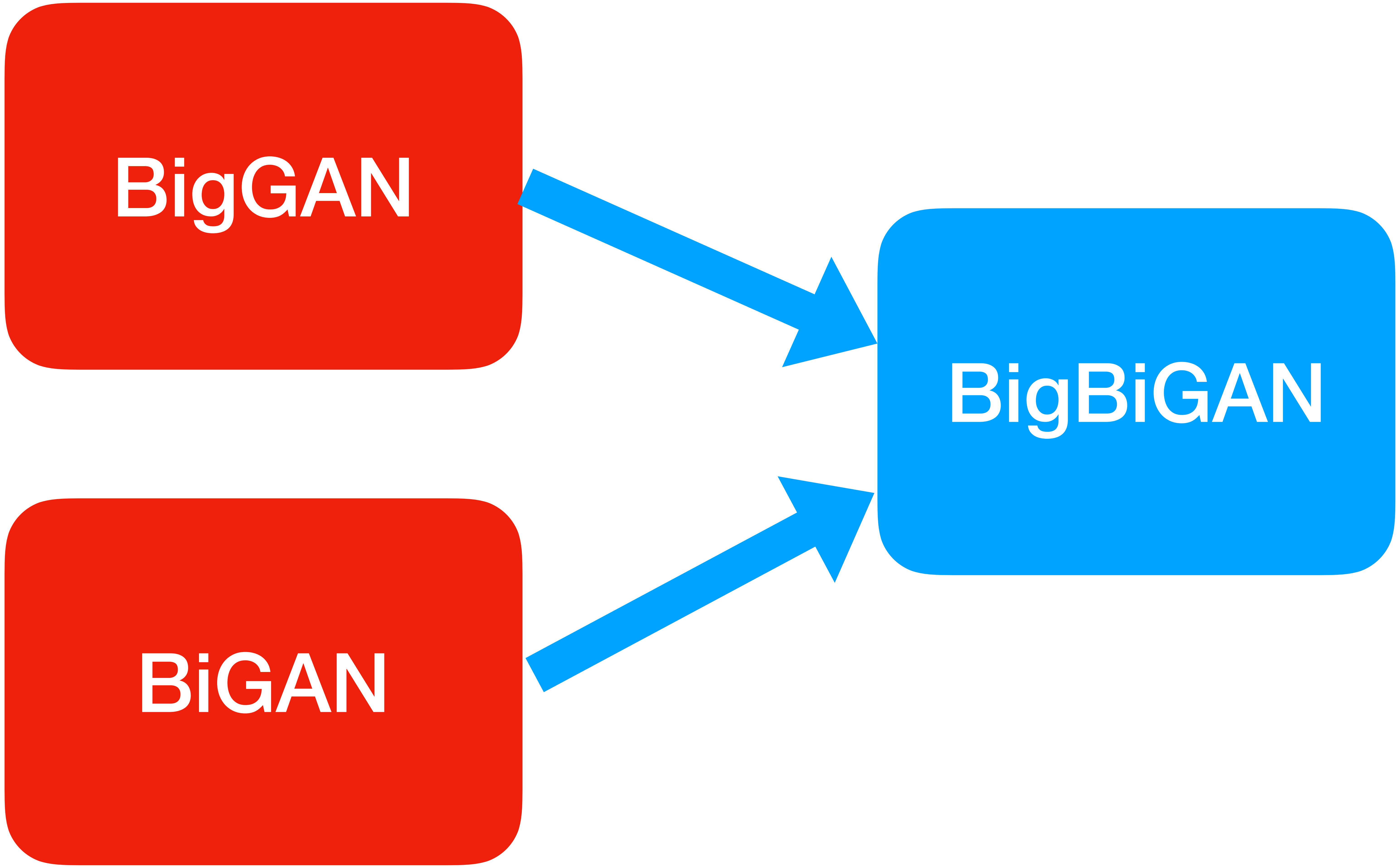
Table 4: Architectures for ImageNet at 128×128 pixels. "ch" represents the channel width multiplier in each network from Table 1.

$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
dense, $4 \times 4 \times 16 \cdot ch$
ResBlock up $16 \cdot ch$
ResBlock up $8 \cdot ch$
ResBlock up $4 \cdot ch$
ResBlock up $2 \cdot ch$
Non-Local Block (64×64)
ResBlock up $1 \cdot ch$
BN, ReLU, 3×3 conv 3
Tanh

(a) Generator

RGB image $x \in \mathbb{R}^{128 \times 128 \times 3}$
ResBlock down $1 \cdot ch$
Non-Local Block (64×64)
ResBlock down $2 \cdot ch$
ResBlock down $4 \cdot ch$
ResBlock down $8 \cdot ch$
ResBlock down $16 \cdot ch$
ResBlock $16 \cdot ch$
ReLU
Global sum pooling
Embed(y)$\cdot \boldsymbol{h}$ + (dense $\rightarrow$ 1)

(b) Discriminator



BigGAN

BiGAN

BigBiGAN

BiGAN

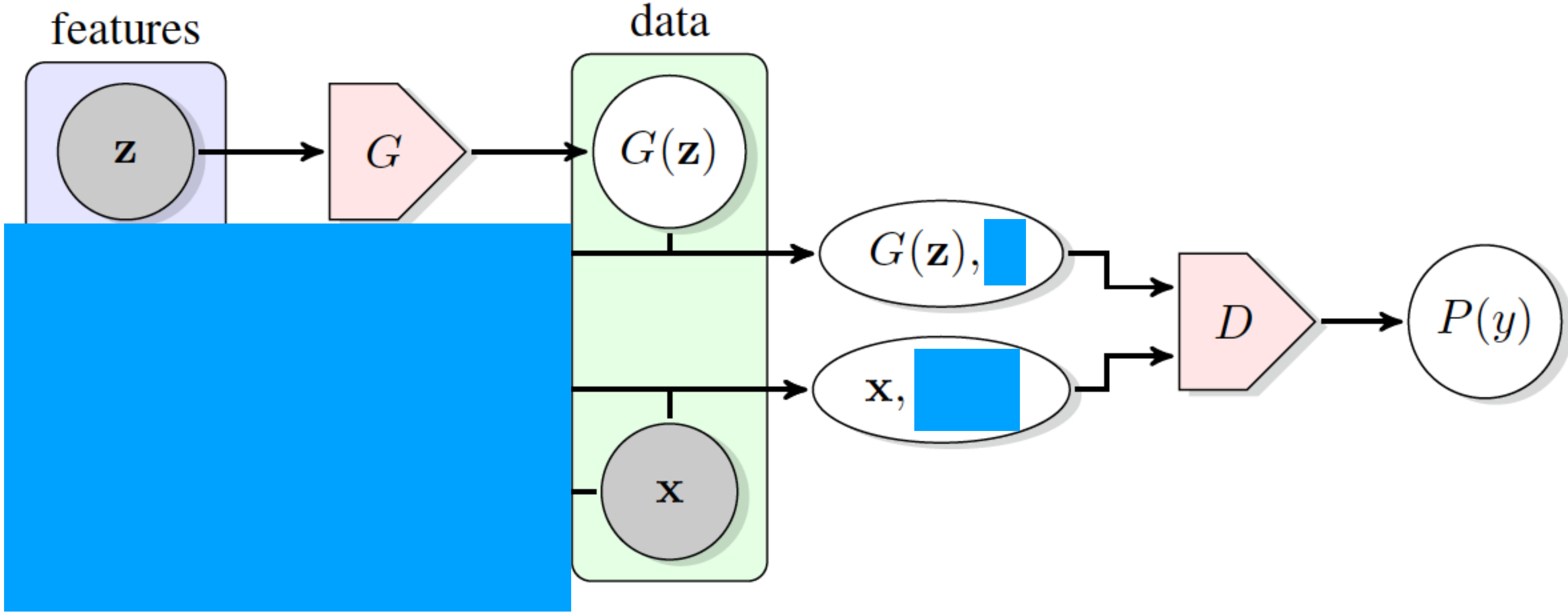
- The ability of the Generative Adversarial Networks (GANs) framework to learn generative models mapping from simple latent distributions to arbitrarily complex data distributions has been demonstrated empirically, with compelling results showing that the latent space of such generators captures **semantic variation** in the data distribution.

- Intuitively, **models trained to predict these semantic latent representations given data** may serve as useful feature representations for auxiliary problems where semantics are relevant.

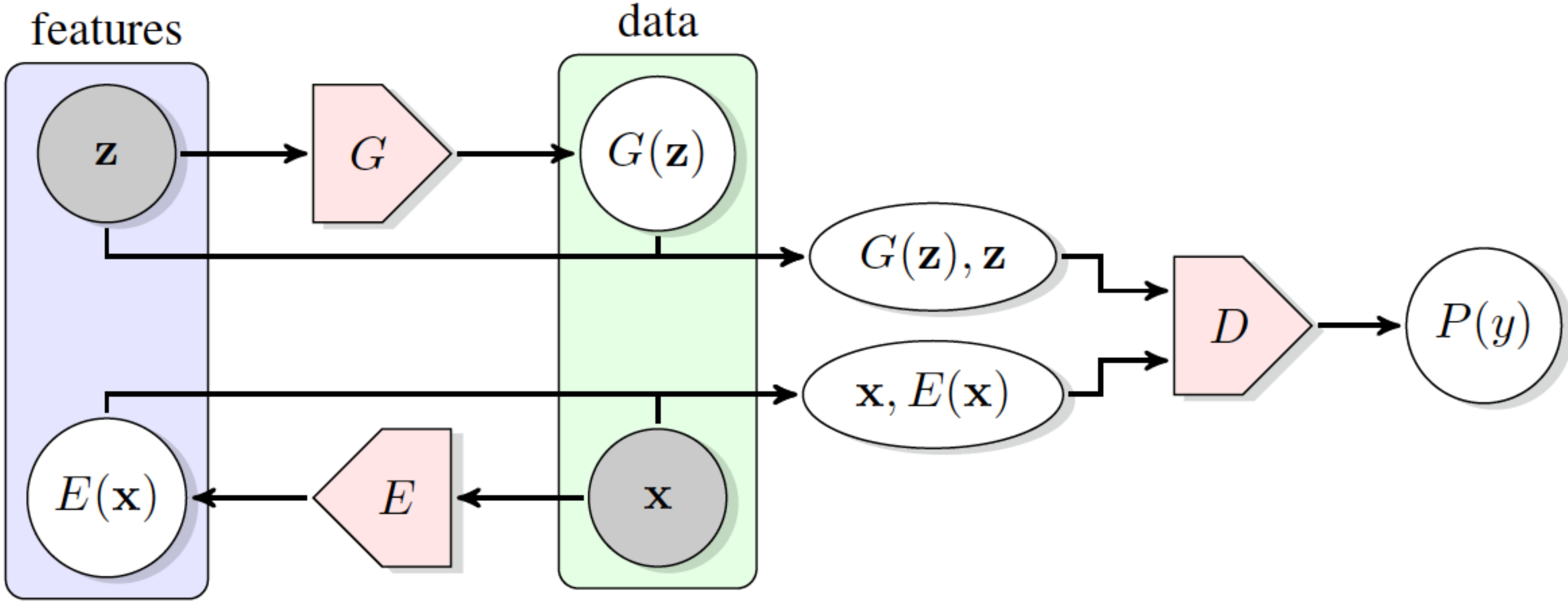
- However, in their existing form, GANs have **no means of** learning the inverse mapping – projecting data back into the latent space.

- We propose **Bidirectional Generative Adversarial Networks (BiGANs)** as a means of learning this **inverse mapping**, and demonstrate that the resulting learned feature representation is useful for auxiliary supervised discrimination tasks, competitive with contemporary approaches to unsupervised and self-supervised feature learning.

Vanilla GAN

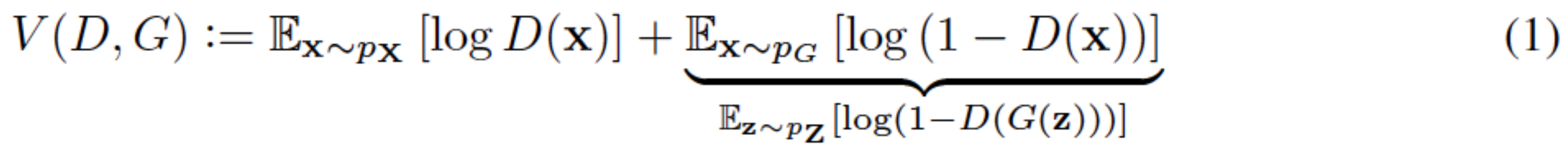


Bidirectional GAN



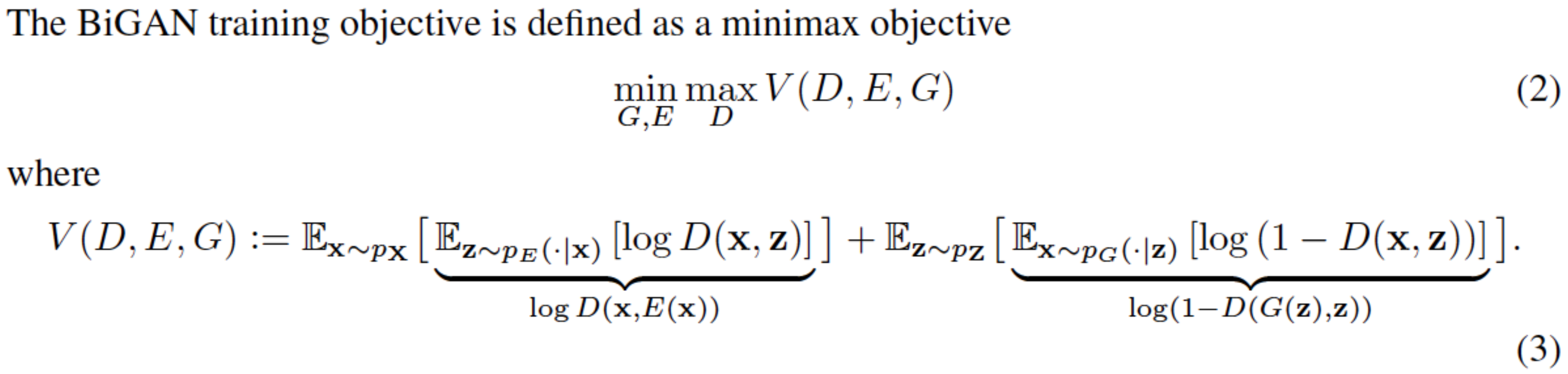
Loss function

- Vanilla GAN



$$V(D,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\left[\log D(\mathbf{x})\right] + \underbrace{\mathbb{E}_{\mathbf{x}\sim p_G}\left[\log\left(1 - D(\mathbf{x})\right)\right]}_{\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}[\log(1-D(G(\mathbf{z})))]} \tag{1}$$

- Bidirectional GAN



The BiGAN training objective is defined as a minimax objective

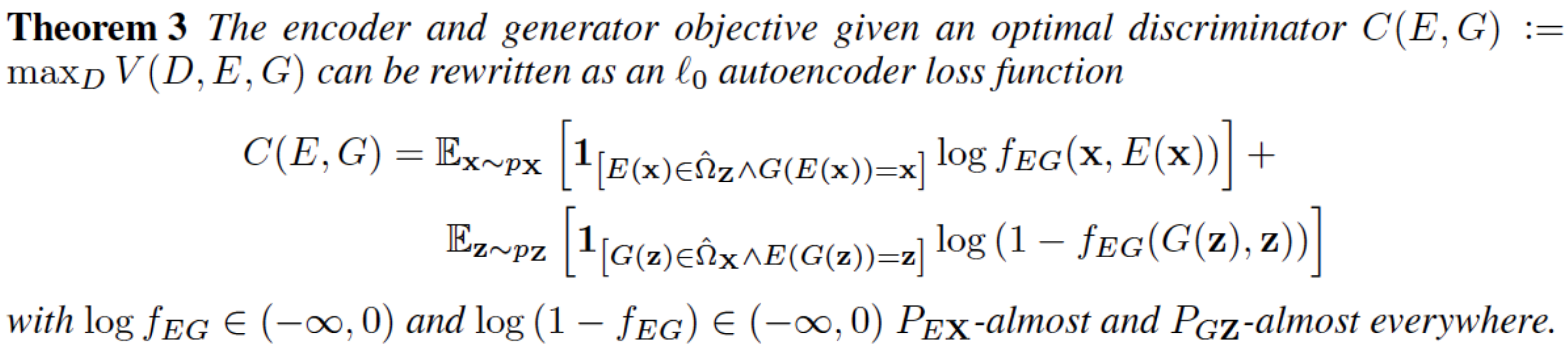
$$\min_{G,E}\max_{D} V(D,E,G) \tag{2}$$

where

-

$$V(D,E,G) := \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{X}}}\Big[\underbrace{\mathbb{E}_{\mathbf{z}\sim p_E(\cdot|\mathbf{x})}\left[\log D(\mathbf{x},\mathbf{z})\right]}_{\log D(\mathbf{x},E(\mathbf{x}))}\Big] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}}\Big[\underbrace{\mathbb{E}_{\mathbf{x}\sim p_G(\cdot|\mathbf{z})}\left[\log\left(1 - D(\mathbf{x},\mathbf{z})\right)\right]}_{\log(1-D(G(\mathbf{z}),\mathbf{z}))}\Big]. \tag{3}$$

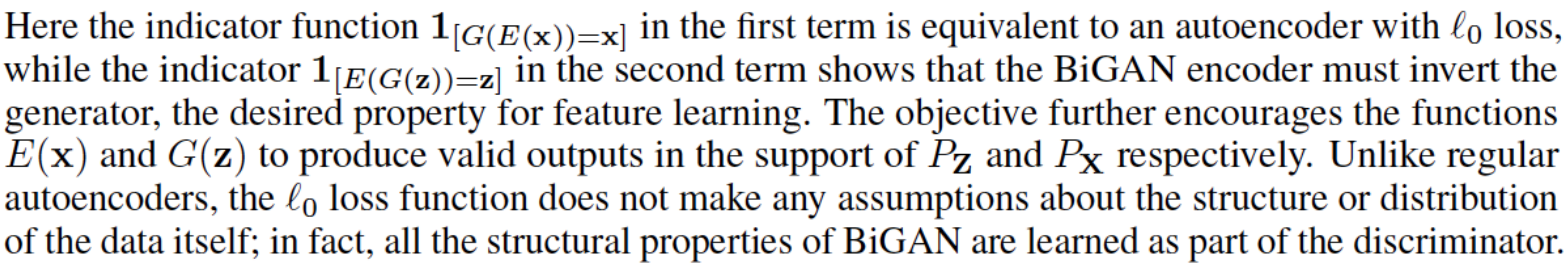
Relationship to Autoencoder



Theorem 3 *The encoder and generator objective given an optimal discriminator* $C(E,G) := \max_D V(D,E,G)$ *can be rewritten as an* ℓ_0 *autoencoder loss function*

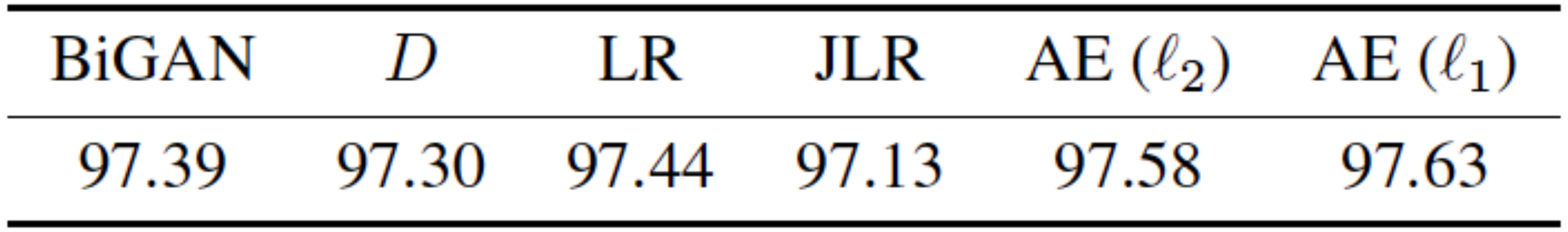
$$C(E,G) = \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\mathbf{1}_{[E(\mathbf{x}) \in \hat{\Omega}_{\mathbf{Z}} \wedge G(E(\mathbf{x})) = \mathbf{x}]} \log f_{EG}(\mathbf{x}, E(\mathbf{x})) \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\mathbf{1}_{[G(\mathbf{z}) \in \hat{\Omega}_{\mathbf{X}} \wedge E(G(\mathbf{z})) = \mathbf{z}]} \log \left(1 - f_{EG}(G(\mathbf{z}), \mathbf{z})\right) \right]$$

with $\log f_{EG} \in (-\infty, 0)$ *and* $\log(1 - f_{EG}) \in (-\infty, 0)$ $P_{E\mathbf{X}}$*-almost and* $P_{G\mathbf{Z}}$*-almost everywhere.*



Here the indicator function $\mathbf{1}_{[G(E(\mathbf{x}))=\mathbf{x}]}$ in the first term is equivalent to an autoencoder with ℓ_0 loss, while the indicator $\mathbf{1}_{[E(G(\mathbf{z}))=\mathbf{z}]}$ in the second term shows that the BiGAN encoder must invert the generator, the desired property for feature learning. The objective further encourages the functions $E(\mathbf{x})$ and $G(\mathbf{z})$ to produce valid outputs in the support of $P_{\mathbf{Z}}$ and $P_{\mathbf{X}}$ respectively. Unlike regular autoencoders, the ℓ_0 loss function does not make any assumptions about the structure or distribution of the data itself; in fact, all the structural properties of BiGAN are learned as part of the discriminator.

Performance of BiGAN



BiGAN	D	LR	JLR	AE (ℓ_2)	AE (ℓ_1)
97.39	97.30	97.44	97.13	97.58	97.63

Table 1: One Nearest Neighbors (1NN) classification accuracy (%) on the permutation-invariant MNIST (LeCun et al., 1998) test set in the feature space learned by BiGAN, Latent Regressor (LR), Joint Latent Regressor (JLR), and an autoencoder (AE) using an ℓ_1 or ℓ_2 distance.

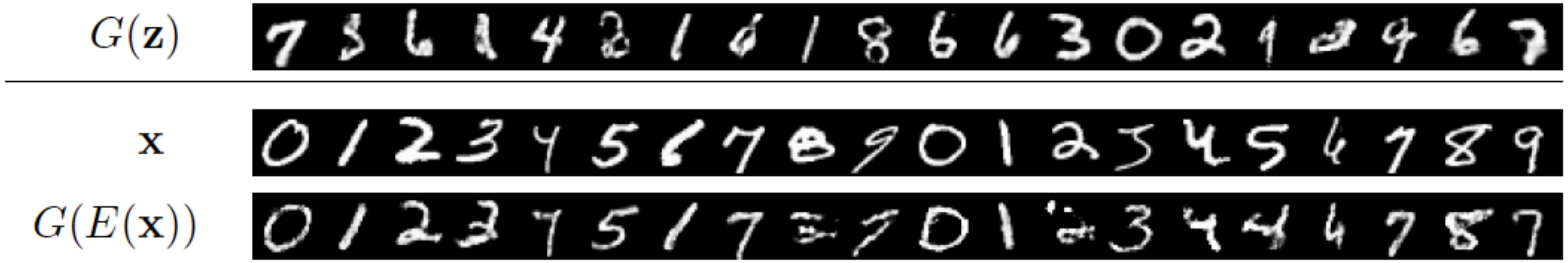
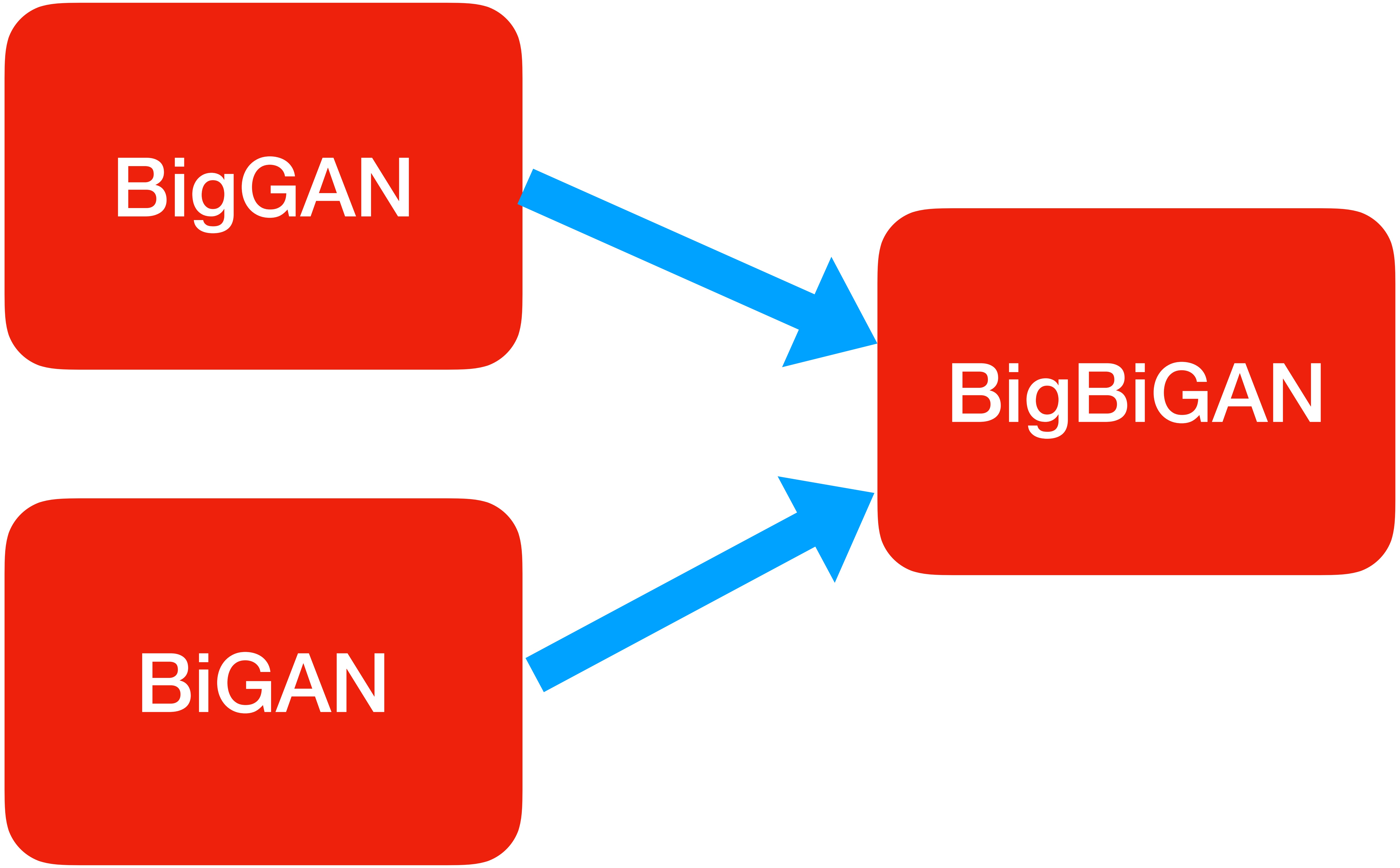


Figure 2: Qualitative results for permutation-invariant MNIST BiGAN training, including generator samples $G(\mathbf{z})$, real data $\mathbf{x}$, and corresponding reconstructions $G(E(\mathbf{x}))$.

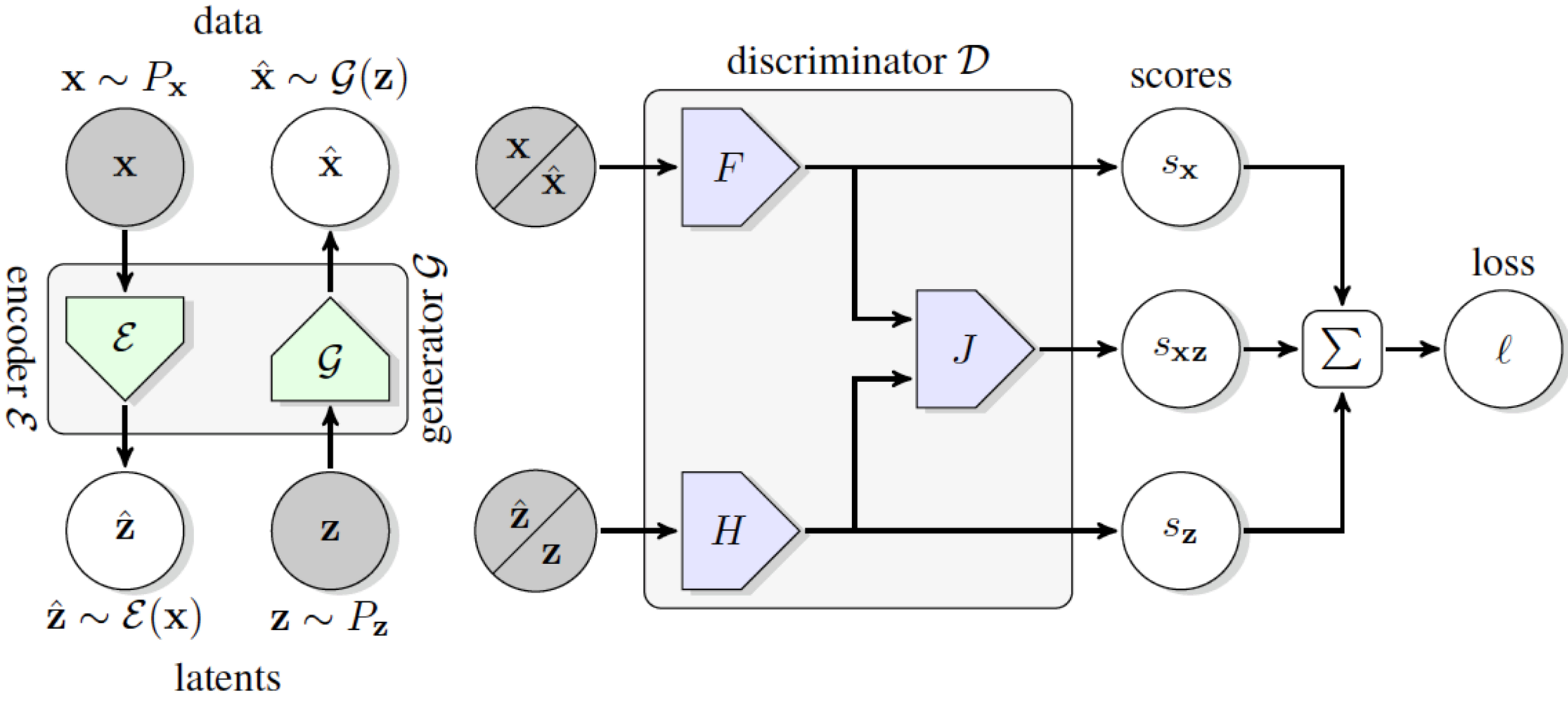


BigGAN

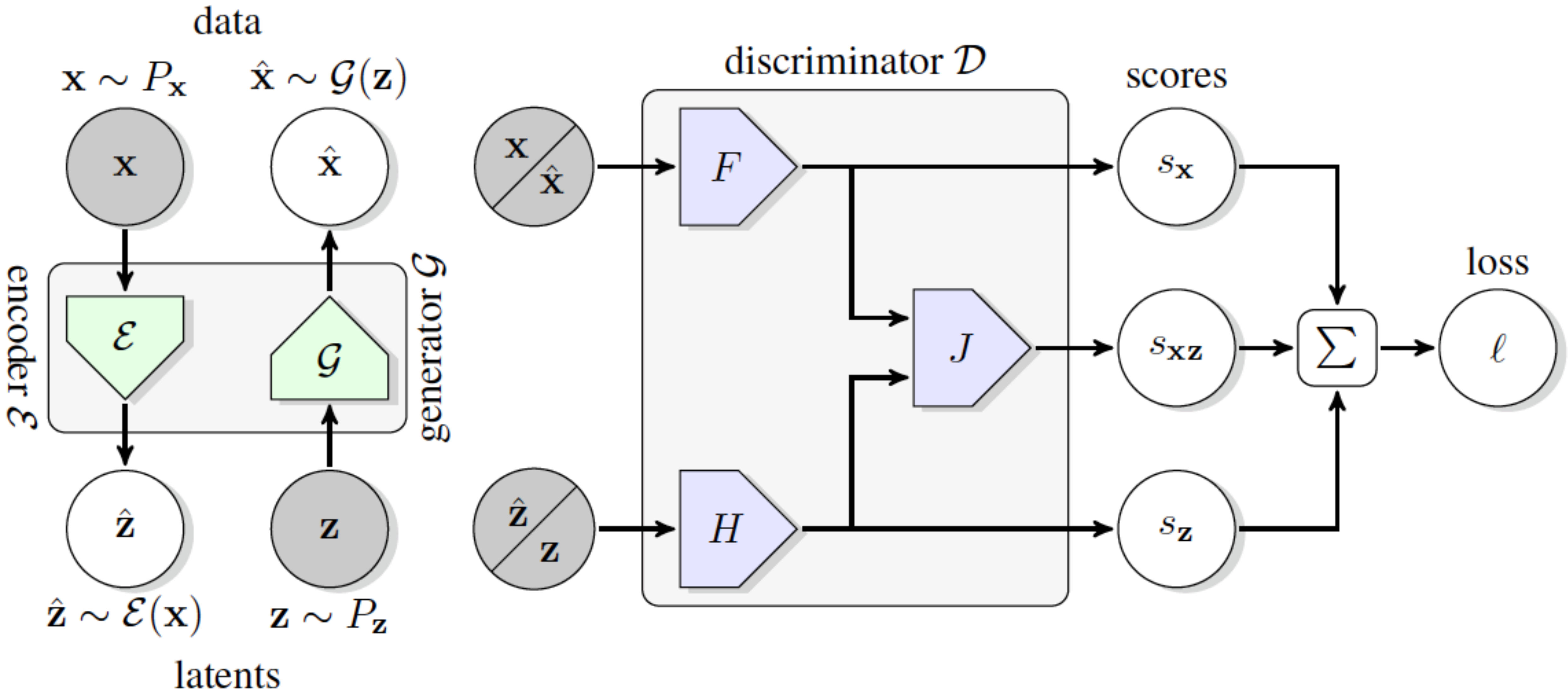
BiGAN

BigBiGAN

BigBiGAN


data
$\mathbf{x} \sim P_{\mathbf{x}}$
$\hat{\mathbf{x}} \sim \mathcal{G}(\mathbf{z})$
$\mathbf{x}$
$\hat{\mathbf{x}}$
encoder $\mathcal{E}$
$\mathcal{E}$
$\mathcal{G}$
generator $\mathcal{G}$
$\hat{\mathbf{z}}$
$\mathbf{z}$
$\hat{\mathbf{z}} \sim \mathcal{E}(\mathbf{x})$
$\mathbf{z} \sim P_{\mathbf{z}}$
latents
discriminator $\mathcal{D}$
$\mathbf{x}$ / $\hat{\mathbf{x}}$
$\hat{\mathbf{z}}$ / $\mathbf{z}$
F
J
H
scores
$s_{\mathbf{x}}$
$s_{\mathbf{xz}}$
$s_{\mathbf{z}}$
$\sum$
loss
ℓ

Loss function


data
$\mathbf{x} \sim P_\mathbf{x}$
$\hat{\mathbf{x}} \sim \mathcal{G}(\mathbf{z})$
$\mathbf{x}$
$\hat{\mathbf{x}}$
encoder $\mathcal{E}$
$\mathcal{E}$
$\mathcal{G}$
generator $\mathcal{G}$
$\hat{\mathbf{z}}$
$\mathbf{z}$
$\hat{\mathbf{z}} \sim \mathcal{E}(\mathbf{x})$
$\mathbf{z} \sim P_\mathbf{z}$
latents
$\mathbf{x}$ / $\hat{\mathbf{x}}$
$\hat{\mathbf{z}}$ / $\mathbf{z}$
discriminator $\mathcal{D}$
F
J
H
scores
$s_\mathbf{x}$
$s_{\mathbf{xz}}$
$s_\mathbf{z}$
$\sum$
loss
ℓ

$$
\begin{aligned}
s_\mathbf{x}(\mathbf{x}) &= \theta_\mathbf{x}^\intercal F_\Theta(\mathbf{x}) \\
s_\mathbf{z}(\mathbf{z}) &= \theta_\mathbf{z}^\intercal H_\Theta(\mathbf{z}) \\
s_{\mathbf{xz}}(\mathbf{x},\mathbf{z}) &= \theta_{\mathbf{xz}}^\intercal J_\Theta(F_\Theta(\mathbf{x}), H_\Theta(\mathbf{z})) \\
\ell_{\mathcal{EG}}(\mathbf{x},\mathbf{z},y) &= y\left(s_\mathbf{x}(\mathbf{x}) + s_\mathbf{z}(\mathbf{z}) + s_{\mathbf{xz}}(\mathbf{x},\mathbf{z})\right) && y \in \{-1,+1\} \\
\mathcal{L}_{\mathcal{EG}}(P_\mathbf{x},P_\mathbf{z}) &= \mathbb{E}_{\mathbf{x}\sim P_\mathbf{x},\hat{\mathbf{z}}\sim\mathcal{E}_\Phi(\mathbf{x})}\left[\ell_{\mathcal{EG}}(\mathbf{x},\hat{\mathbf{z}},+1)\right] + \mathbb{E}_{\mathbf{z}\sim P_\mathbf{z},\hat{\mathbf{x}}\sim\mathcal{G}_\Phi(\mathbf{z})}\left[\ell_{\mathcal{EG}}(\hat{\mathbf{x}},\mathbf{z},-1)\right] \\
\ell_{\mathcal{D}}(\mathbf{x},\mathbf{z},y) &= h(ys_\mathbf{x}(\mathbf{x})) + h(ys_\mathbf{z}(\mathbf{z})) + h(ys_{\mathbf{xz}}(\mathbf{x},\mathbf{z})) && y \in \{-1,+1\} \\
\mathcal{L}_{\mathcal{D}}(P_\mathbf{x},P_\mathbf{z}) &= \mathbb{E}_{\mathbf{x}\sim P_\mathbf{x},\hat{\mathbf{z}}\sim\mathcal{E}_\Phi(\mathbf{x})}\left[\ell_{\mathcal{D}}(\mathbf{x},\hat{\mathbf{z}},+1)\right] + \mathbb{E}_{\mathbf{z}\sim P_\mathbf{z},\hat{\mathbf{x}}\sim\mathcal{G}_\Phi(\mathbf{z})}\left[\ell_{\mathcal{D}}(\hat{\mathbf{x}},\mathbf{z},-1)\right]
\end{aligned}
$$

	Encoder ($\mathcal{E}$)						Gen. ($\mathcal{G}$)		Loss $\mathcal{L}_*$				Results		
	A.	D.	C.	R.	Var.	η	C.	R.	$s_{\mathbf{xz}}$	$s_{\mathbf{x}}$	$s_{\mathbf{z}}$	$P_{\mathbf{z}}$	IS (↑)	FID (↓)	Cls. (↑)
Base	S	50	1	128	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	22.66 ± 0.18	31.19 ± 0.37	48.10 ± 0.13
Deterministic $\mathcal{E}$	S	50	1	128	(-)	1	96	128	✓	✓	✓	$\mathcal{N}$	22.79 ± 0.27	31.31 ± 0.30	46.97 ± 0.35
Uniform $P_{\mathbf{z}}$	S	50	1	128	(-)	1	96	128	✓	✓	✓	($\mathcal{U}$)	22.83 ± 0.24	31.52 ± 0.28	45.11 ± 0.93
$\mathbf{x}$ Unary Only	S	50	1	128	✓	1	96	128	✓	✓	(-)	$\mathcal{N}$	23.19 ± 0.28	31.99 ± 0.30	47.74 ± 0.20
$\mathbf{z}$ Unary Only	S	50	1	128	✓	1	96	128	✓	(-)	✓	$\mathcal{N}$	19.52 ± 0.39	39.48 ± 1.00	47.78 ± 0.28
No Unaries (BiGAN)	S	50	1	128	✓	1	96	128	✓	(-)	(-)	$\mathcal{N}$	19.70 ± 0.30	42.92 ± 0.92	46.71 ± 0.88
Small $\mathcal{G}$ (32)	S	50	1	128	✓	1	(32)	128	✓	✓	✓	$\mathcal{N}$	3.28 ± 0.18	247.30 ± 10.31	43.59 ± 0.34
Small $\mathcal{G}$ (64)	S	50	1	128	✓	1	(64)	128	✓	✓	✓	$\mathcal{N}$	19.96 ± 0.15	38.93 ± 0.39	47.54 ± 0.33
No $\mathcal{E}$ (GAN) *	(-)						96	128	(-)	✓	(-)	$\mathcal{N}$	23.56 ± 0.37	30.91 ± 0.23	-
High Res $\mathcal{E}$ (256)	S	50	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.45 ± 0.14	27.86 ± 0.13	50.80 ± 0.30
Low Res $\mathcal{G}$ (64)	S	50	1	(256)	✓	1	96	(64)	✓	✓	✓	$\mathcal{N}$	19.40 ± 0.19	15.82 ± 0.06	47.51 ± 0.09
High Res $\mathcal{G}$ (256)	S	50	1	(256)	✓	1	96	(256)	✓	✓	✓	$\mathcal{N}$	24.70	38.58	51.49
ResNet-101	S	(101)	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.29	28.01	51.21
ResNet ×2	S	50	(2)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.68	27.81	52.66
RevNet	(V)	50	1	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.33 ± 0.09	27.78 ± 0.06	49.42 ± 0.18
RevNet ×2	(V)	50	(2)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.21	27.96	54.40
RevNet ×4	(V)	50	(4)	(256)	✓	1	96	128	✓	✓	✓	$\mathcal{N}$	23.23	28.15	57.15
ResNet (↑ $\mathcal{E}$ LR)	S	50	1	(256)	✓	(10)	96	128	✓	✓	✓	$\mathcal{N}$	23.27 ± 0.22	28.51 ± 0.44	53.70 ± 0.15
RevNet ×4 (↑ $\mathcal{E}$ LR)	(V)	50	(4)	(256)	✓	(10)	96	128	✓	✓	✓	$\mathcal{N}$	23.08	28.54	60.15

Table 1: Results for variants of BigBiGAN, given in Inception Score [31] (IS) and Fréchet Inception Distance [15] (FID) of the generated images, and ImageNet top-1 classification accuracy percentage (Cls.) of a supervised logistic regression classifier trained on the encoder features [37], computed on a split of 10K images randomly sampled from the training set, which we refer to as the "$\text{train}_{\text{val}}$" split. The *Encoder ($\mathcal{E}$)* columns specify the $\mathcal{E}$ architecture (A.) as ResNet (S) or RevNet (V), the depth (D., e.g. 50 for ResNet-50), the channel width multiplier (C.), with 1 denoting the original widths from [13], the input image resolution (R.), whether the variance is predicted and a z vector is sampled from the resulting distribution (Var.), and the learning rate multiplier η relative to the $\mathcal{G}$ learning rate. The *Generator ($\mathcal{G}$)* columns specify the BigGAN $\mathcal{G}$ channel multiplier (C.), with 96 corresponding to the original width from [1], and output image resolution (R.). The *Loss* columns specify which terms of the BigBiGAN loss are present in the objective. The P_z column specifies the input distribution as a standard normal $\mathcal{N}(0,1)$ or continuous uniform $\mathcal{U}(-1,1)$. Changes from the *Base* setup in each row are highlighted in blue. Results with margins of error (written as "$\mu \pm \sigma$") are the means and standard deviations over three runs with different random seeds. (Experiments requiring more computation were run only once.) (* Result for vanilla GAN (*No $\mathcal{E}$ (GAN)*) selected with early stopping based on best FID; other results selected with early stopping based on validation classification accuracy (Cls.).)

Method	Architecture	Feature	Top-1	Top-5
BiGAN [4, 38]	AlexNet	conv3	31.0	-
Motion Segmentation (MS) [27, 3]	ResNet-101	AvePool	27.6	48.3
Exemplar (Ex) [5, 3]	ResNet-101	AvePool	31.5	53.1
Relative Position (RP) [2, 3]	ResNet-101	AvePool	36.2	59.2
Colorization (Col) [37, 3]	ResNet-101	AvePool	39.6	62.5
Combination of MS+Ex+RP+Col [3]	ResNet-101	AvePool	-	69.3
CPC [35]	ResNet-101	AvePool	48.7	73.6
Rotation [8, 21]	RevNet-50 ×4	AvePool	55.4	-
Efficient CPC [14]	ResNet-170	AvePool	61.0	83.0
BigBiGAN (ours)	ResNet-50	AvePool	55.4	77.4
	ResNet-50	BN+CReLU	56.6	78.6
	RevNet-50 ×4	AvePool	60.8	81.4
	RevNet-50 ×4	BN+CReLU	61.3	81.9

Table 2: Comparison of BigBiGAN models on the official ImageNet validation set against recent competing approaches with a supervised logistic regression classifier. BigBiGAN results are selected with early stopping based on highest accuracy on our train$_{\text{val}}$ subset of 10K training set images. *ResNet-50* results correspond to row *ResNet* ($\uparrow \mathcal{E}$ *LR)* in Table 1, and *RevNet-50* ×4 corresponds to *RevNet* ×4 *(*$\uparrow \mathcal{E}$ *LR)*.

Method	Steps	IS (↑)	FID vs. Train (↓)	FID vs. Val. (↓)
BigGAN + SL [24]	500K	20.4 (15.4 $\pm$ 7.57)	-	25.3 (71.7 $\pm$ 66.32)
BigGAN + Clustering [24]	500K	22.7 (22.8 $\pm$ 0.42)	-	23.2 (22.7 $\pm$ 0.80)
BigBiGAN + SL (ours)	500K	25.38 (25.33 $\pm$ 0.17)	22.78 (22.63 $\pm$ 0.23)	23.60 (23.56 $\pm$ 0.12)
BigBiGAN High Res $\mathcal{E}$ + SL (ours)	500K	25.43 (25.45 $\pm$ 0.04)	22.34 (22.36 $\pm$ 0.04)	22.94 (23.00 $\pm$ 0.15)
BigBiGAN High Res $\mathcal{E}$ + SL (ours)	1M	27.94 (27.80 $\pm$ 0.21)	20.32 (20.27 $\pm$ 0.09)	21.61 (21.62 $\pm$ 0.09)

Table 3: Comparison of our BigBiGAN for unsupervised (unconditional) generation vs. previously reported results for unsupervised BigGAN from [24]. We specify the "pseudo-labeling" method as *SL* (Single Label) or *Clustering*. For comparison we train BigBiGAN for the same number of steps (500K) as the BigGAN-based approaches from [24], but also present results from additional training to 1M steps in the last row and observe further improvements. All results above include the median m as well as the mean μ and standard deviation σ across three runs, written as "m ($\mu \pm \sigma$)". The BigBiGAN result is selected with early stopping based on best FID vs. Train.

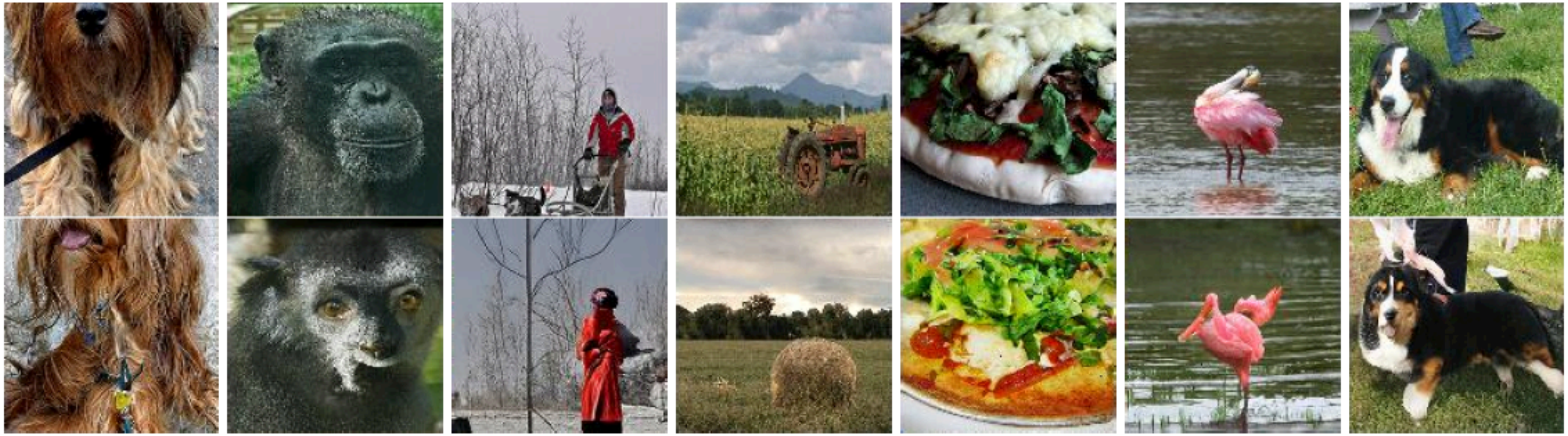


Figure 2: Selected reconstructions from an unsupervised BigBiGAN model (Section 3.3). Top row images are real data $\mathbf{x} \sim P_{\mathbf{x}}$; bottom row images are generated reconstructions of the above image $\mathbf{x}$ computed by $\mathcal{G}(\mathcal{E}(\mathbf{x}))$. Unlike most explicit reconstruction costs (e.g., pixel-wise), the reconstruction cost implicitly minimized by a (Big)BiGAN [4, 7] tends to emphasize more semantic, high-level details. Additional reconstructions are presented in Appendix B.

Conclusion

- We have shown that BigBiGAN, an unsupervised learning approach based purely on generative models, achieves state-of-the-art results in image representation learning on ImageNet.

- Our ablation study lends further credence to the hope that powerful generative models can be beneficial for representation learning, and in turn that learning an inference model can improve large-scale generative models.

- In the future we hope that representation learning can continue to benefit from further advances in generative models and inference models alike, as well as scaling to larger image databases.