CVPR SEATTLE WASHINGTON
JUNE 16-18 2020

Attentive Normalization for Conditional Image generation

Yi Wang[1], Ying-Cong Chen[1], Xiangyu Zhang[2], Jian Sun[2], Jiaya Jia[1]

[1] The Chinese University of Hong Kong

[2] MEGVII Technology

MEGVII 旷视

What is Conditional Image generation?

panda

Input Class Label

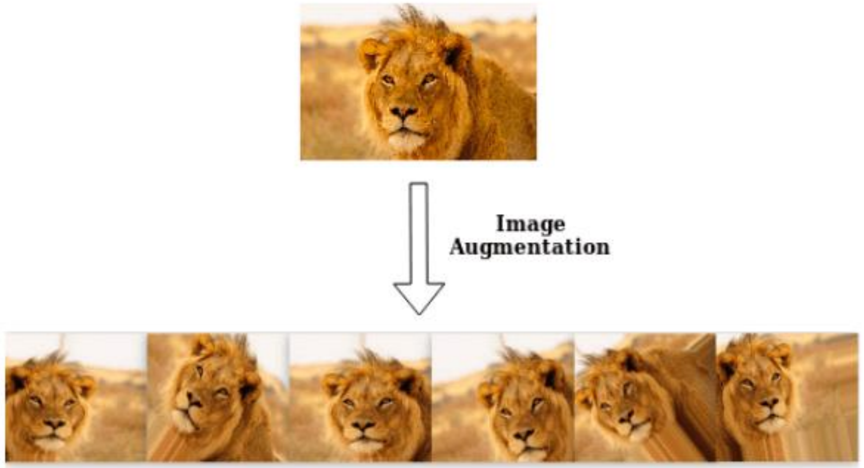
Generation

Generative model

The *Applications* of Conditional Image generation

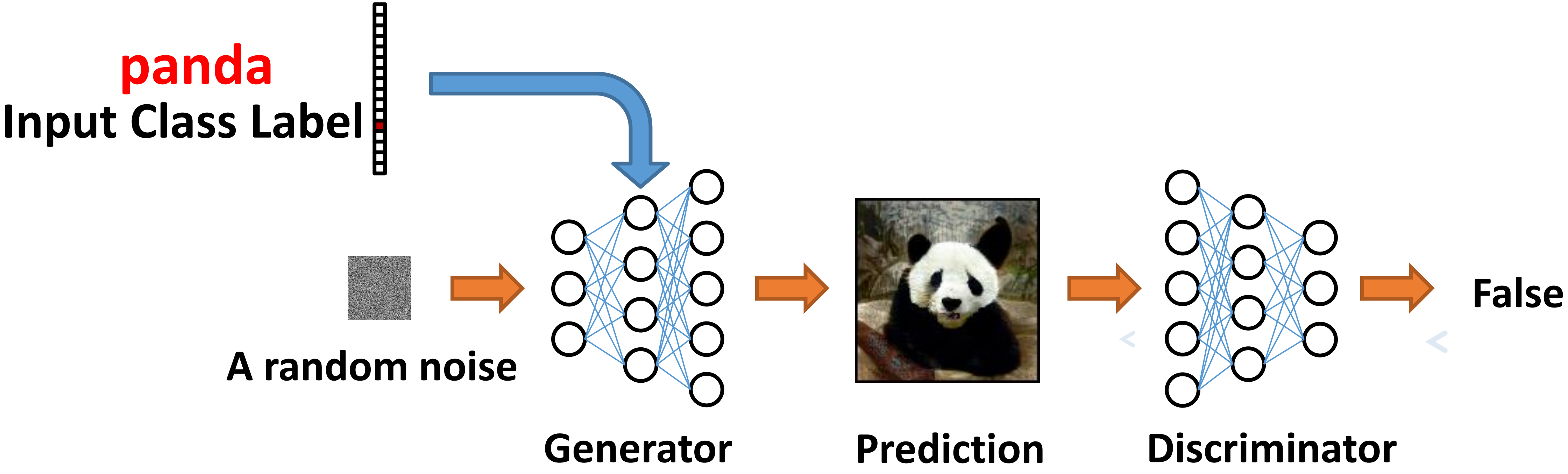
Applications:

- **Image creation and editing.**
- **Data augmentation.**
- **Others.**



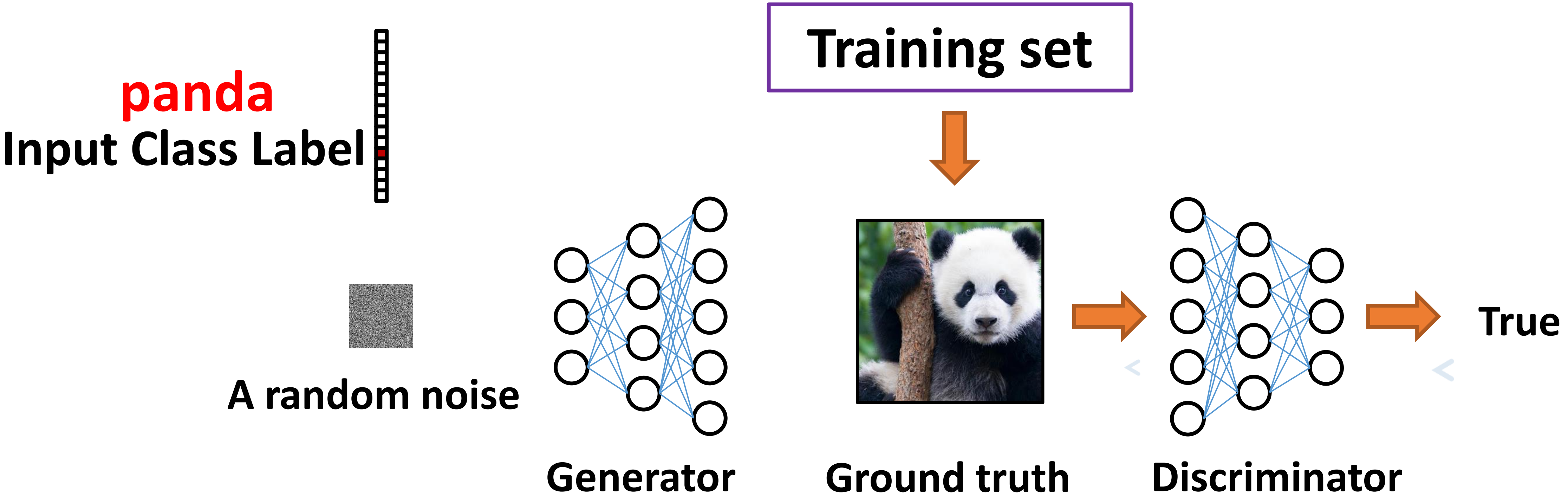
How to generate images from a class label

- Class-conditional image generation using GAN (Generative adversarial networks)



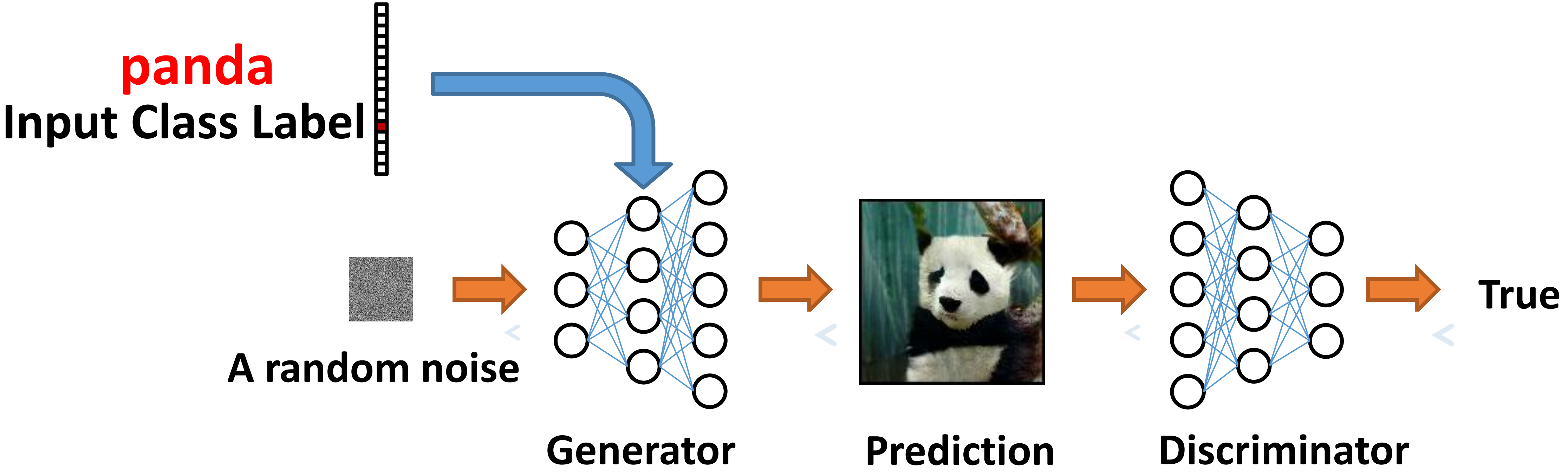
How to generate images from a class label

- **Class-conditional image generation using GAN (Generative adversarial networks)**

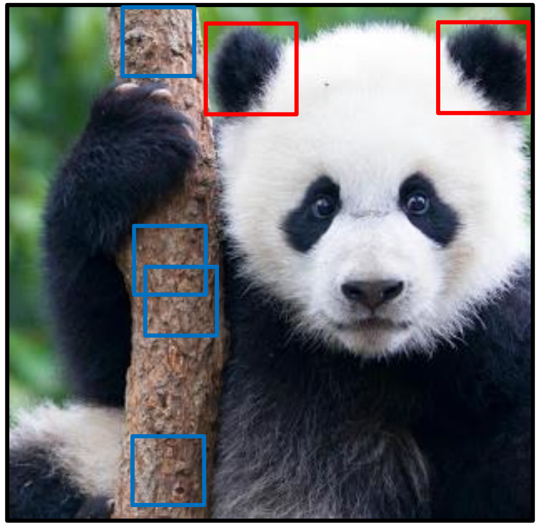


How to generate images from a class label

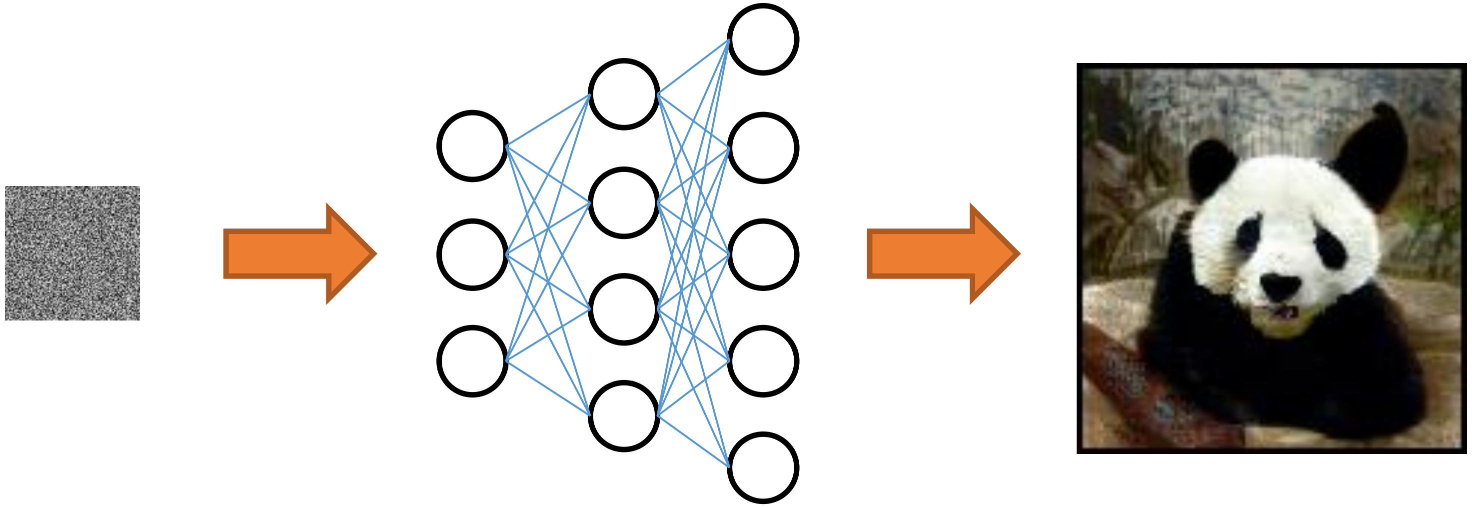
- Class-conditional image generation using GAN (Generative adversarial networks)



Long-range dependency in image generation

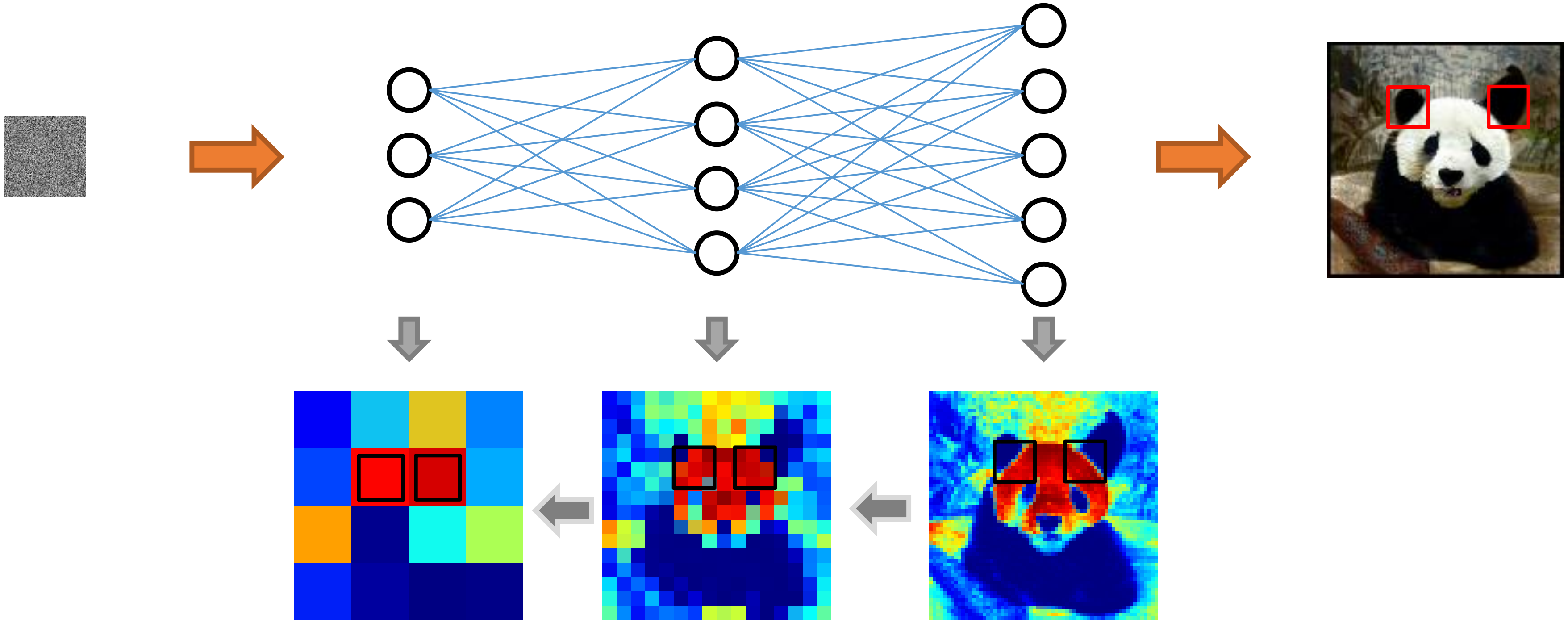


- Standard convolutional neural network:
  - Modelling image contents in a hierarchical manner.

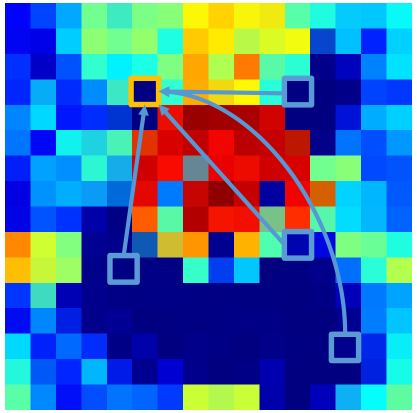


Long-range dependency in image generation

- **Standard convolutional neural network:**
  - **Long-range dependency is conduct in a Markov chain.**



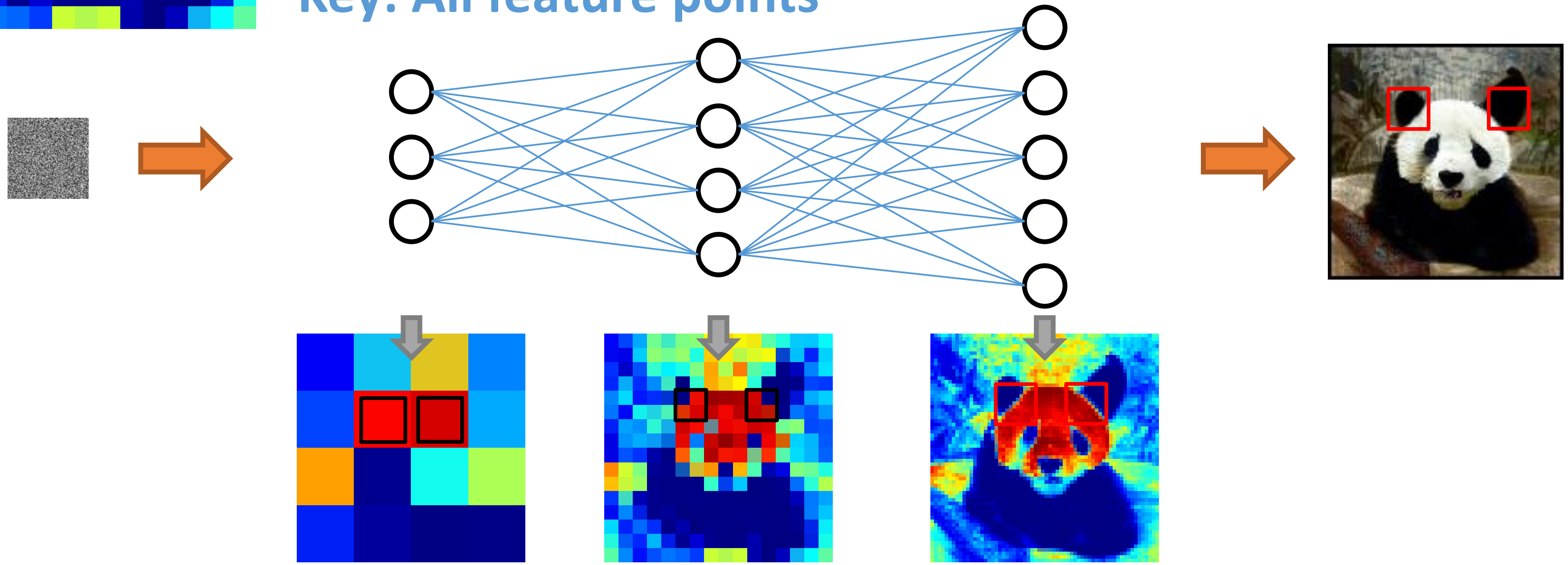
Prior work: self-attention



Self attention: reconstructing each feature point using the weighted sum of all feature points[1].

Query: every feature point

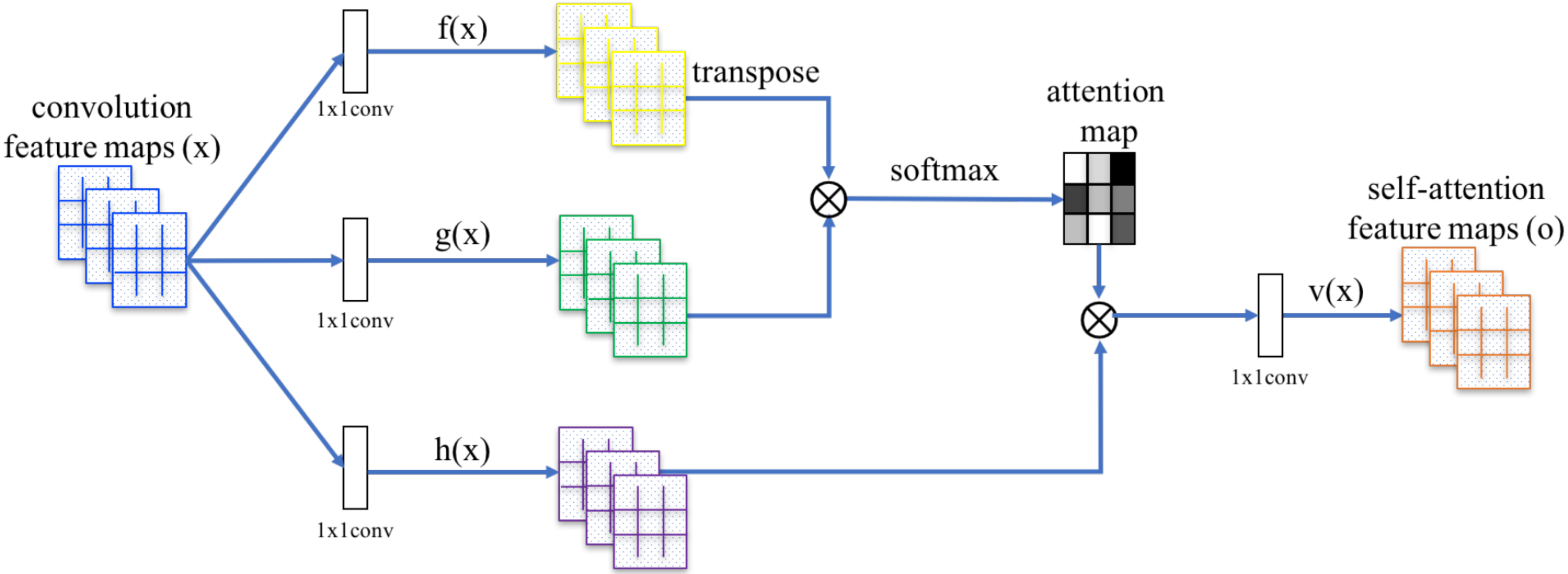
Key: All feature points



[1] Zhang, Han, et al. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Prior work: self-attention

Self attention: reconstructing each feature point using the weighted sum of all feature points[1].



[1] Zhang, Han, et al. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Our method: Attentive Normalization

Key observations

- For a feature map, different locations may correspond to semantics with varying mean and variance.
- Normalizing the whole feature map tends to deteriorate the learned semantics of the intermediate features spatially [2].

(-0.08, 4.77) (0.00, 0.35) (0.01, 0.44) (-0.01, 1.09)

Mean

Standard deviation

[2] Park, Taesung, et al. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019.

Our method: Attentive Normalization

Core idea:

We normalize the input feature maps spatially according to the semantic layouts predicted from them.

Regional normalization

Semantic layout learning

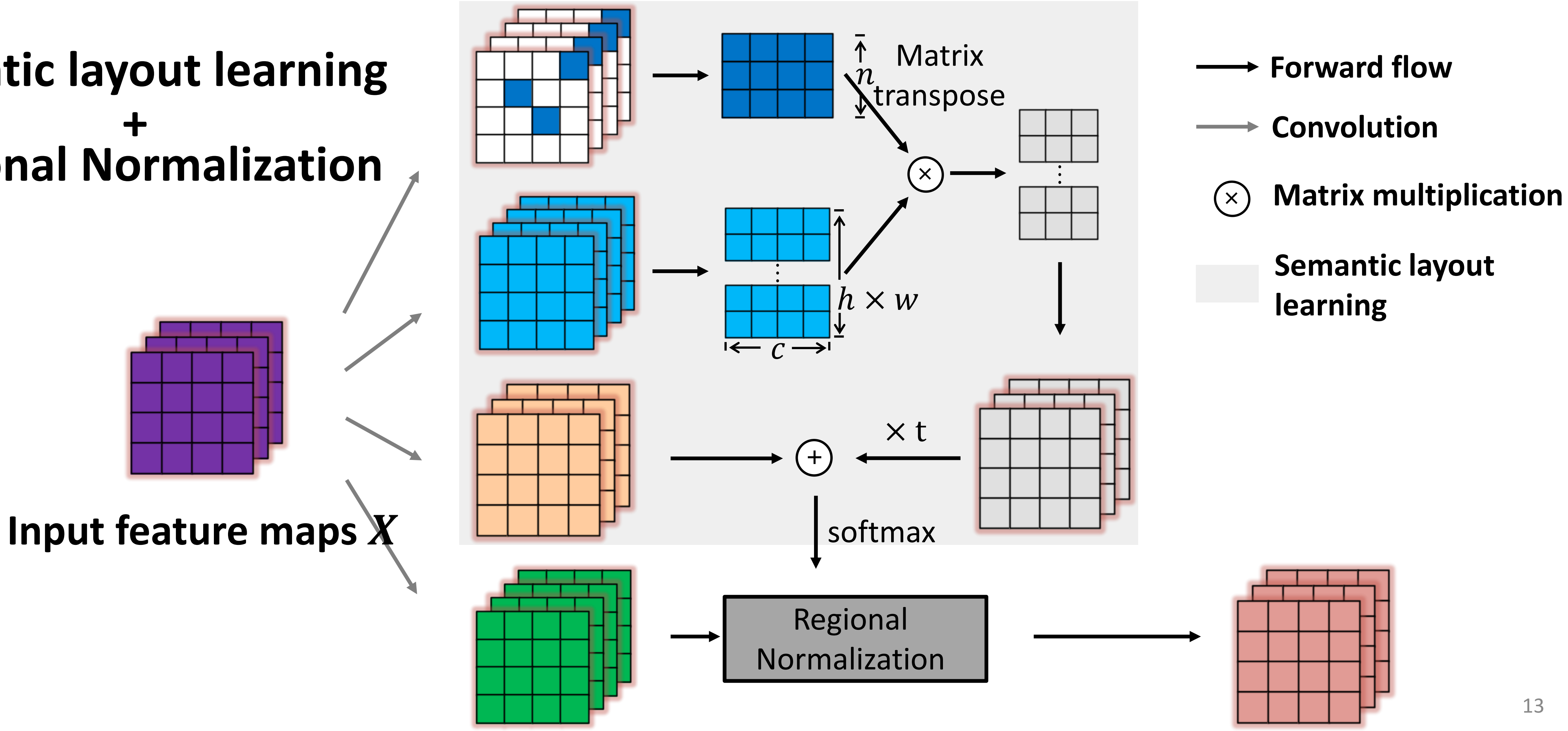
Empirical observations to backup our method:

- A feature map can be viewed as a composition of multiple semantic entities[3,4].
- The deep layers in a neural network capture high-level semantics of the input images[5].

[3] Greff, Klaus, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. In *NeurIPS*, 2017.
[4] Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *NeurIPS*, 2017.
[5] Le, Quoc V. Building high-level features using large scale unsupervised learning. In *ICASSP*, 2013.

Our method: Attentive Normalization

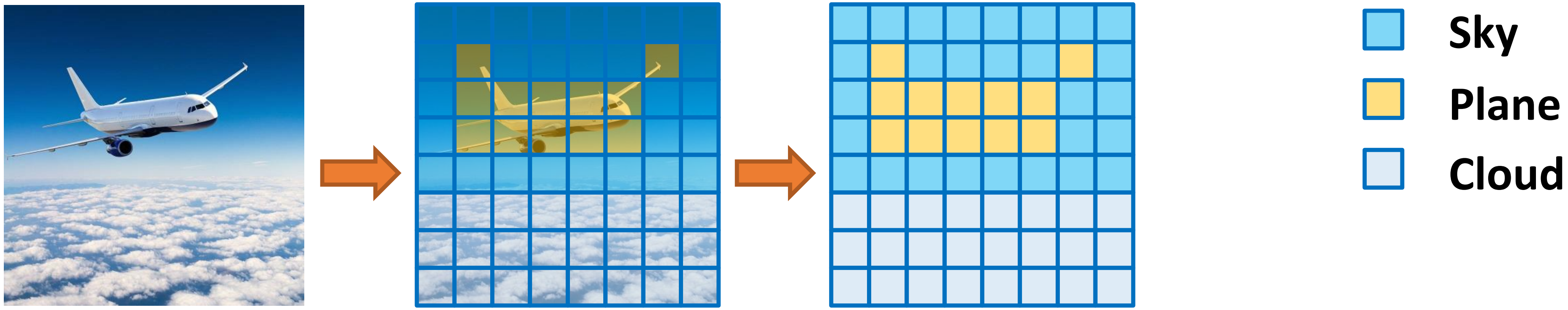
Semantic layout learning
+
Regional Normalization



Semantic layout learning

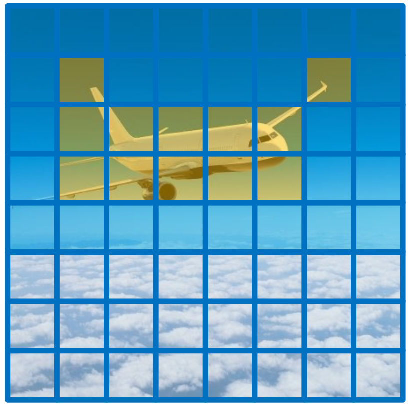
An image is composed of n semantic entities.

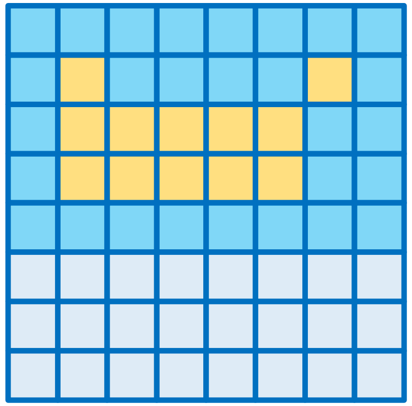
Each feature point of the image, it is determined by at least one entity.



Semantic layout learning

How to group feature points of an image according to their correlation to the semantic entities.





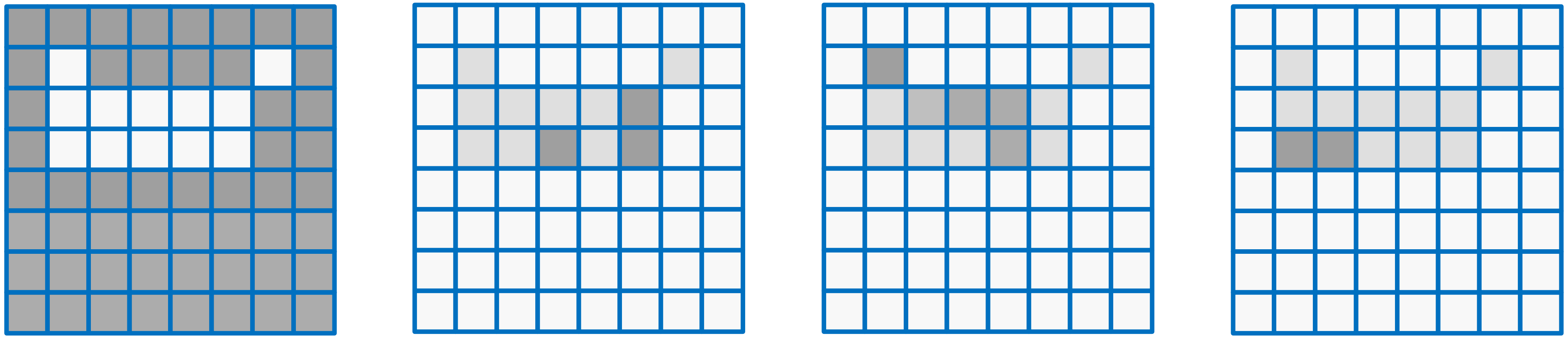
Sky

Plane

Cloud

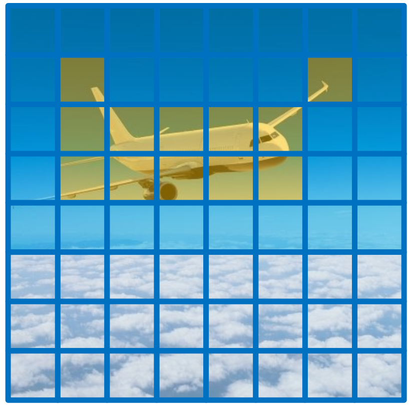
Learned entities

- Background
- Round thing
- Square thing
- Flat thing



Semantic layout learning

How to get these semantic entities?

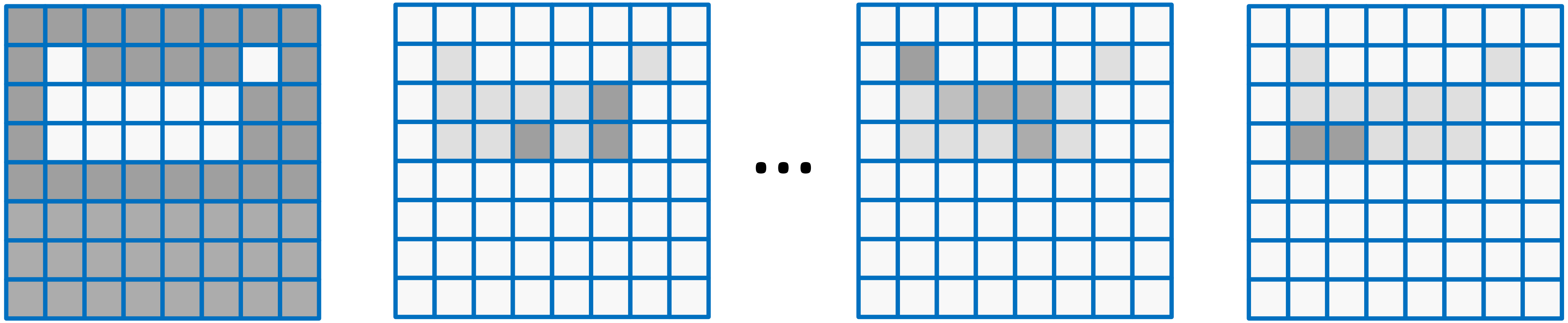


Learned entities

- Background
- Round thing
- …
- Square thing
- Flat thing

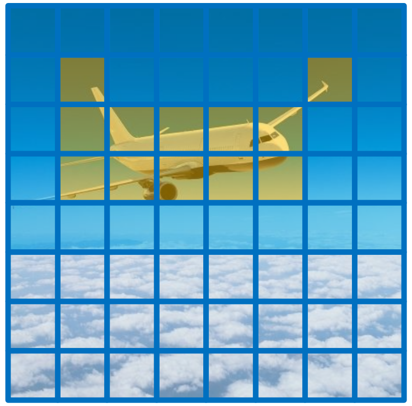
Implementations

- A convolutional layer with n filters



Semantic layout learning

How to get these semantic entities?



Learned entities
- Background
- Round thing
- …
- Square thing
- Flat thing

Implementations

- **A convolutional layer with n filters**

To encourage semantic entities to approach diverse patterns

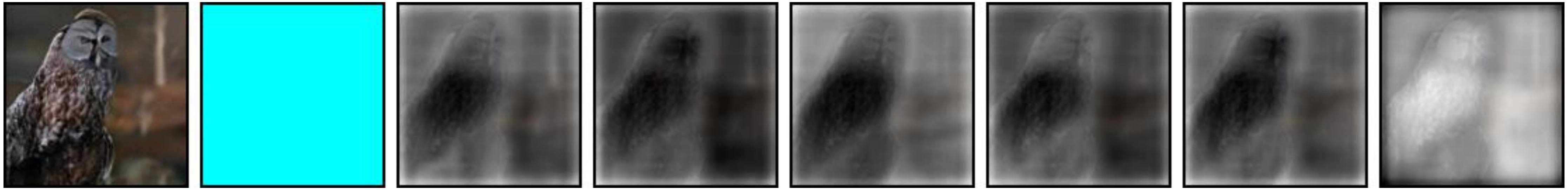
$$\mathcal{L}_o = \lambda_o ||\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} - \boldsymbol{I}||_F^2$$

where $\boldsymbol{W} \in \mathcal{R}^{n \times c}$ is a weight matrix constituted by these n entities (each row is the spanned weight in the row-vector form).

Challenges in optimizing SSL

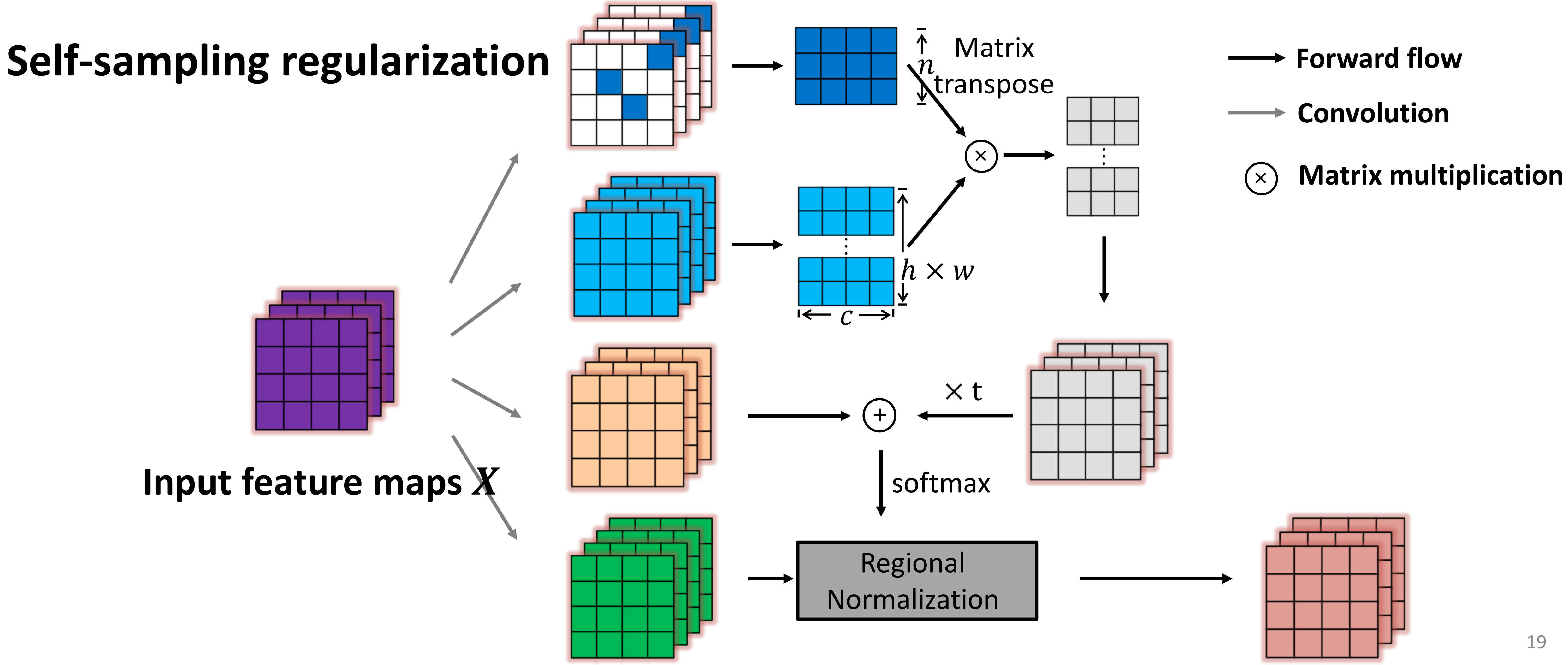
Trivial solutions for learning entities directly

- It tends to group all feature points with a single semantic entity.
- No protocols are set to ban useless semantic entities.



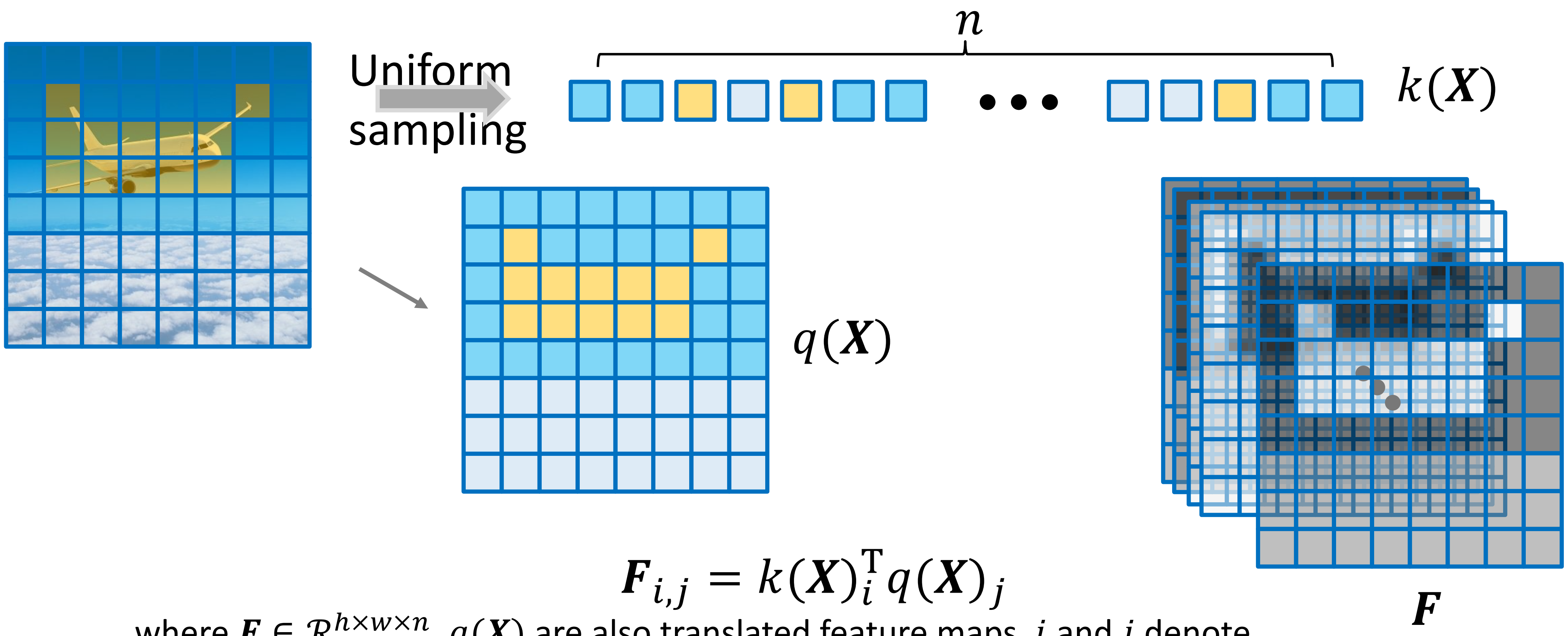
Generated The highlighted regions indicated by the learned semantic layouts

Our method: Attentive Normalization



Self-Sampling Regularization (SSR)

Regularizing semantics learning with a self-sampling branch

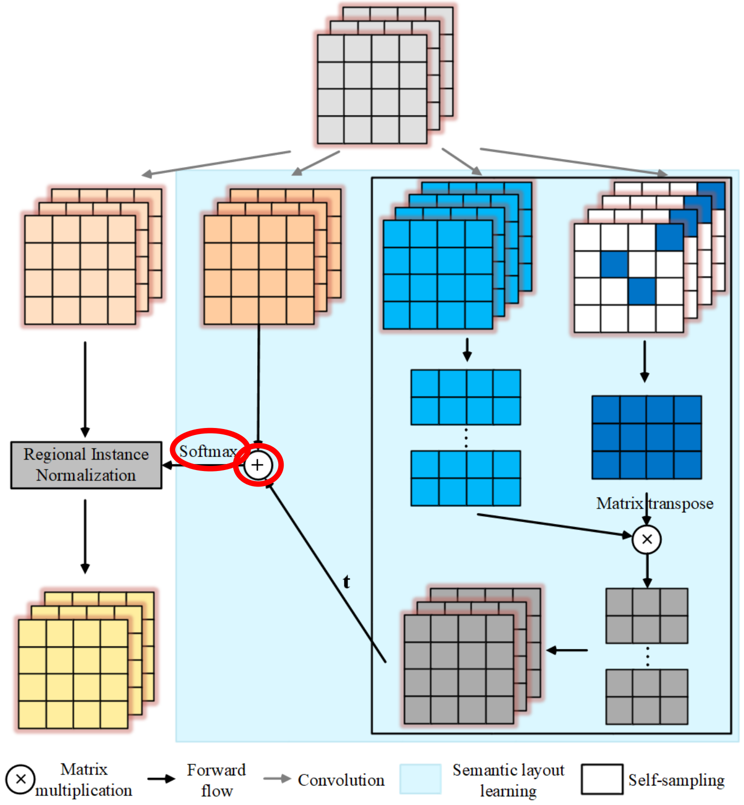


$$\boldsymbol{F}_{i,j} = k(\boldsymbol{X})_i^{\mathrm{T}} q(\boldsymbol{X})_j$$

where $\boldsymbol{F} \in \mathcal{R}^{h \times w \times n}$, $q(\boldsymbol{X})$ are also translated feature maps. i and j denote pixel location. We set $\#\{i\} = n$ and $\#\{j\} = h \times w$.

Soft Semantic Layout Computation

How to get semantic layout



$$\boldsymbol{S}^{\text{raw}} = t\boldsymbol{F} + f(\boldsymbol{X})$$

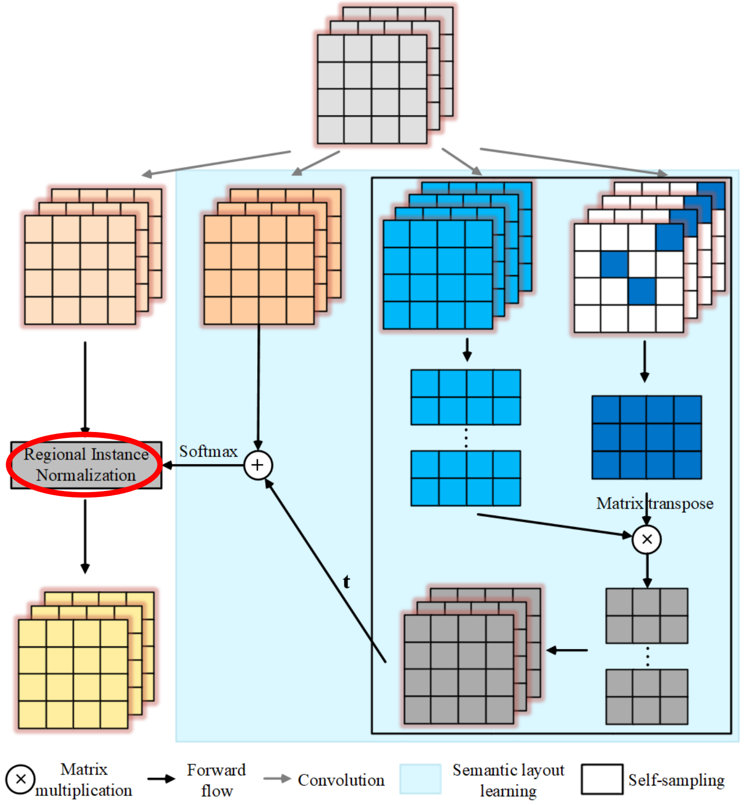
where $t \in \mathcal{R}^{1\times1\times n}$ is a learnable vector initialized as 0.1.

$$\boldsymbol{S}_k = \frac{\exp(\tau \boldsymbol{S}_k^{\text{raw}})}{\sum_{i=1}^{n} \exp(\tau \boldsymbol{S}_i^{\text{raw}})}$$

where i and k index the feature channels. Each $\boldsymbol{S}_k$ is a soft mask, indicating the probability of every pixel belonging to class k. τ is the coefficient to control the smoothness of the predicted semantic layout with default value set to 0.1.

Regional Normalization

Regional normalization based on the computed semantic layout



$$\bar{\boldsymbol{X}} = \sum_{i=1}^{n} \left(\frac{\boldsymbol{X} - \mu(\boldsymbol{X}_{\boldsymbol{S}_i})}{\sigma(\boldsymbol{X}_{\boldsymbol{S}_i}) + \epsilon} \times \beta_i + \alpha_i \right) \odot \boldsymbol{S}_i$$

where $\boldsymbol{X}_{\boldsymbol{S}_i} = \boldsymbol{X} \odot \boldsymbol{S}_i$. β_i and α_i are learnable parameter vectors for the affine transformation, initialized to 1 and 0, respectively. $\mu(\cdot)$ and $\sigma(\cdot)$ compute the mean and standard deviation from the instance, respectively.

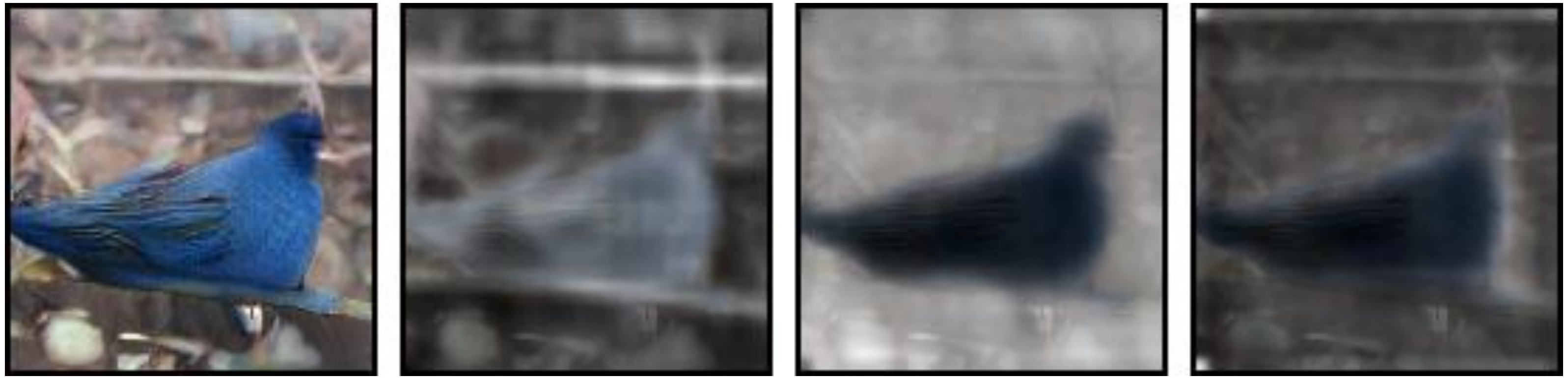
$$AN(\boldsymbol{X}) = \rho \bar{\boldsymbol{X}} + \boldsymbol{X}$$

where ρ is a learnable scalar initialized as 0.

Analysis: learned semantic layouts

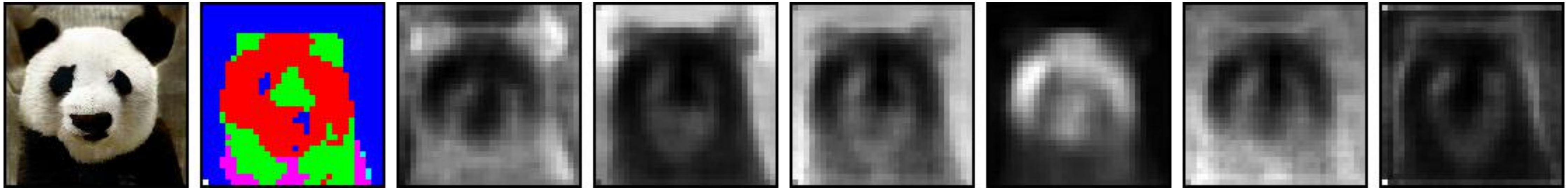
The predicted semantic layout indicates regions with high inner coherence in semantics.

(-0.01, 4.55) (0.03, 0.68) (0.03, 0.80) (-0.03, 1.07)



Mean

Standard deviation



Generated The highlighted regions indicated by the learned semantic layouts

Analysis: complexity analysis

Complexity Analysis

The computational complexity of Attentive Normalization: $O(nN\boxed{HW}C)$
The computational complexity of Self-Attention: $O(N(\boxed{H^2W^2}C + HWC^2))$

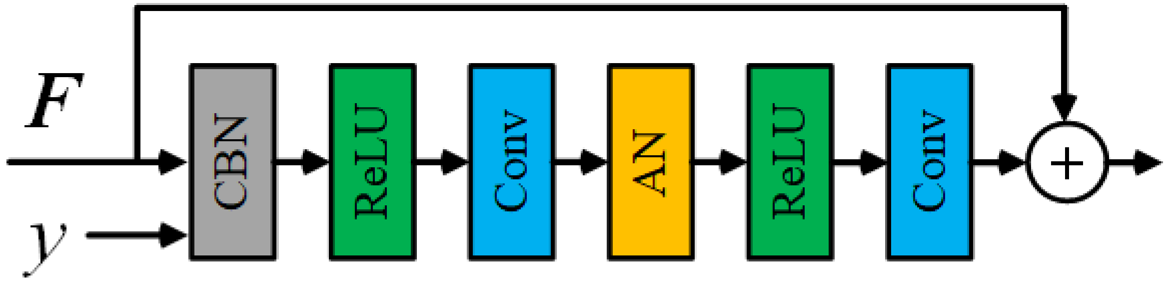
Module (ms)	128 x 128	256 x 256	512 x 512	1024 x 1024
AN (n=16)	**0.73**	**2.24**	**9.46**	**37.68**
Self-attention[1]	5.21	79.42	-	-

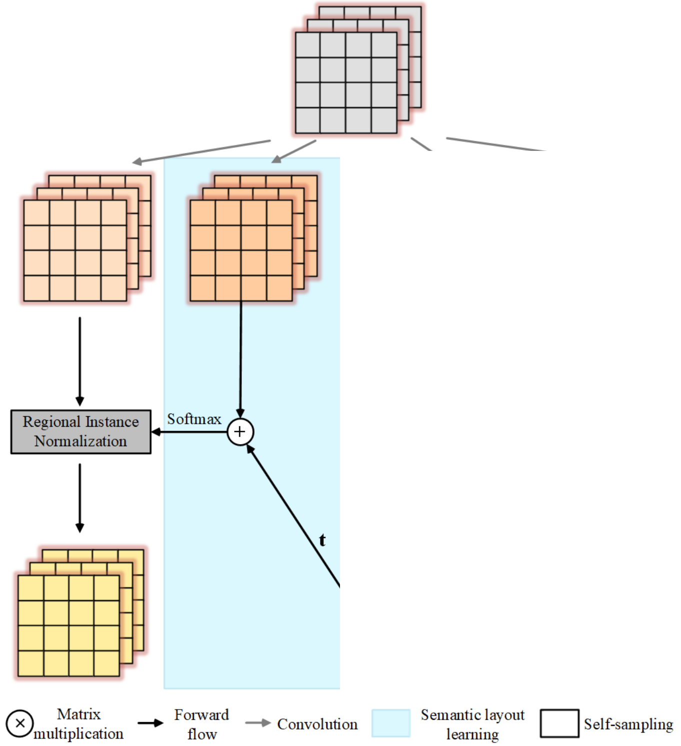
All fed tensors are with the same batch size 1 and channel number 32. Resolutions are different.
'-' stands for evaluation time unmeasurable due to out-of-memory in GPU.
Running environment: Pytorch 1.1.0, 4 CPUs, 1 TiTAN 2080 GPU, 32GB Memory.

[1] Zhang, Han, et al. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

Applications: how to use Attentive Normalization

How to use Attentive Normalization





In the testing phase, we remove the randomness in the self-sampling branch of AN by switching off this branch with $t = 0$.

Applications: class-conditional image generation

Task: it learns to synthesize image distributions by training on the given images. It maps a randomly sampled noise to z an image x via a generator G, conditioning on the image label y.

Network architectures

Generator: $z \rightarrow$ FC $(4 \times 4 \times 1024) \rightarrow$ ResBlock up 1024 $\rightarrow$ ResBlock up 512 $\rightarrow$ ResBlock up 256 (AN) $\rightarrow$ ResBlock up 128 $\rightarrow$ ResBlock up 64 $\rightarrow$ BN $\rightarrow$ ReLU $\rightarrow$ Conv3 $\times$ 3 $\rightarrow$ Tanh

Discriminator: $x \rightarrow$ ResBlock down 64 $\rightarrow$ ResBlock down 128 (AN) $\rightarrow$ ResBlock down 256 $\rightarrow$ concat(Embed(y), h) $\rightarrow$ ResBlock down 512 $\rightarrow$ ResBlock down 1024 $\rightarrow$ ReLU $\rightarrow$ Global sum pooling $\rightarrow$ FC $\rightarrow$ 1

Applications: class-conditional image generation

For the optimization objective, we use hinge adversarial loss.

For generator

$$\mathcal{L}_G = -\mathrm{E}_{z\sim P_z, y\sim P_{\text{data}}} D(G(z,y),y)$$

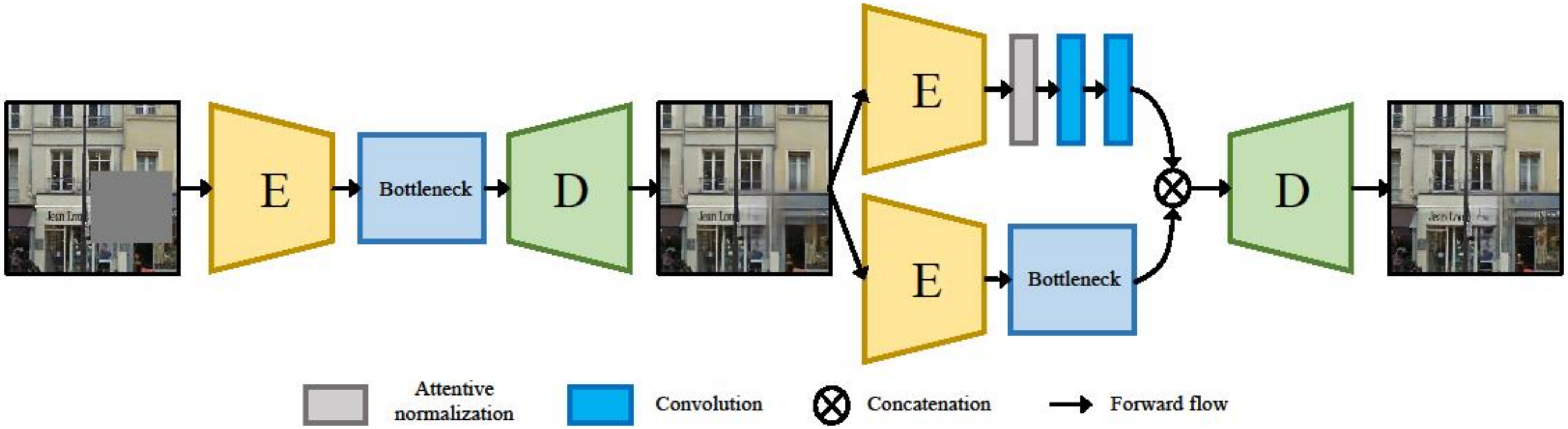
For discriminator,

$$\mathcal{L}_D = -\mathrm{E}_{(x,y)\sim P_{\text{data}}}\left[\min\left(1 - D(x,y)\right)\right] + \mathrm{E}_{z\sim P_z, y\sim P_{\text{data}}}\left[\min\left(1 + D(G(z,y),y)\right)\right]$$

Applications: generative image inpainting

Task: it takes an incomplete image C and a mask M (with missing pixels value 1 and known ones 0) as input and predicts a visually plausible result based on image context. The generated content should be coherent with the given context.

Network architectures



Applications: generative image inpainting

For the optimization objective,

For generator, it uses a reconstruction term and an adversarial term as

$$\mathcal{L}_G = \lambda_{\text{rec}} ||G(C, M) - Y||_1 - \lambda_{\text{adv}} \mathrm{E}_{\hat{C} \sim P_{\hat{C}}} [D(\hat{C})]$$

where Y **is the corresponding ground truth of** C**,** $\hat{C} = G(C, M) \odot M + Y \odot (1 - M)$.

For discriminator,

$$\mathcal{L}_D = -\mathrm{E}_{Y \sim P_{\text{data}}} [D(Y)] + \mathrm{E}_{\hat{C} \sim P_{\hat{C}}} [D(\hat{C})] + \lambda_{\text{gap}} \mathrm{E}_{\tilde{C} \sim P_{\tilde{C}}} [(||\nabla_{\tilde{C}} D(\tilde{C})||_2 - 1)^2]$$

where $\tilde{C} = t\hat{C} + (1 - t)Y, t \in [0,1]$, **and** $\lambda_{\text{gap}} = 10$.

Experimental results

Two tasks

- **Class-conditional image generation**.
  - ImageNet (with 128 x 128 resolution).
- **Generative image inpainting**.
  - Paris Streetview (with 256 x 256 resolution).

Both tasks rely heavily on *distant visual relationship* modelling for generating convincing semantic structures for objects and complex scenes.

Quantitative Results

MEGVII 旷视

Class-conditional image generation On ImageNet (128x128):

	Itr x 1K↓	FID↓	Intra FID↓	IS↑
AC-GAN[6]	/	/	260.0	28.5
SN-GAN[7]	1000	27.62	92.4	36.80
SN-GAN*[1]	1000	22.96	/	42.87
SA-GAN[1]	1000	18.65	83.7	**52.52**
Ours	**880**	**17.84**	**83.4**	46.57

[1] Zhang, Han, et al. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.
[6] Odena, Augustus, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2017.
[7] Miyato, Takeru, and Masanori Koyama. cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*. 2018.

Quantitative Results

MEGVII 旷视

On ImageNet: Intra-FID comparison on type image classes

Class Name (label)	**SN-GAN**	**SA-GAN**	**Ours**
Stone wall (825)	49.3	57.5	**34.16**
Geyser (974)	19.5	21.6	**13.97**
Valley (979)	26.0	39.7	**22.90**
Coral fungus (991)	37.2	38.0	**24.02**
Indigo hunting (14)	66.8	53.0	**42.54**
Redshank (141)	60.1	48.9	**39.06**
Saint Bernard (247)	55.3	**35.7**	39.36
Tiger cat (282)	90.2	88.1	**66.65**

Natural scenes or textures

Objects with complex structural relationship

Quantitative Results

On Paris streetview *test*:

Method	PSNR (dB)↑	SSIM↑	MAE↓
CA	23.78	0.8406	0.0338
Ours	**25.09**	**0.8541**	**0.0334**

Qualitative Results

Class-conditional image generation

Drilling platform (540)

Agaric (992)

Schooner (780)

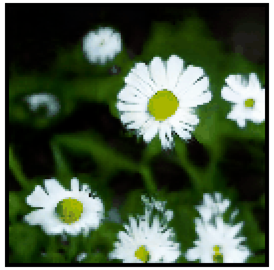
Image inpainting

Categorical interpolation

Blenheim spaniel<->indigo hunting<->schooner

coffee<->owl

Panda<->Drilling platform



Flower<->bird

Ablation studies

Quantitative results of AN module ablation on ImageNet with class-conditional generation

Module	**IS↑**	**FID↓**
Attentive Normalization **w BN**	43.92	19.59
Attentive Normalization **w/o orthogonal reg**	45.99	18.07
Attentive Normalization **w/o SSR**	37.86	23.58
Attentive Normalization (**n=8**)	45.51	19.01
*Attentive Normalization (**n=16**)*	*46.57*	*17.84*
Attentive Normalization (**n=32**)	47.14	17.75

Conclusion

- We propose a novel method to conduct *distant relationship modeling* in *conditional image generation* through *normalization*.
- It may be beneficial to other tasks in future work.

Limitation

- It is possible that some features are *not normalized* when they are not similar to the given entities.
- Though these learned entities are correlated with the high-level understanding, it is still hard to interpret their exact meaning.

Acknowledgement

This work is conducted in the collaboration with Dr. CHEN Ying-cong, Dr. ZHANG Xiangyu, Prof. SUN Jian, and Prof. JIA Jiaya.

Thanks