GANs for Flow Prediction

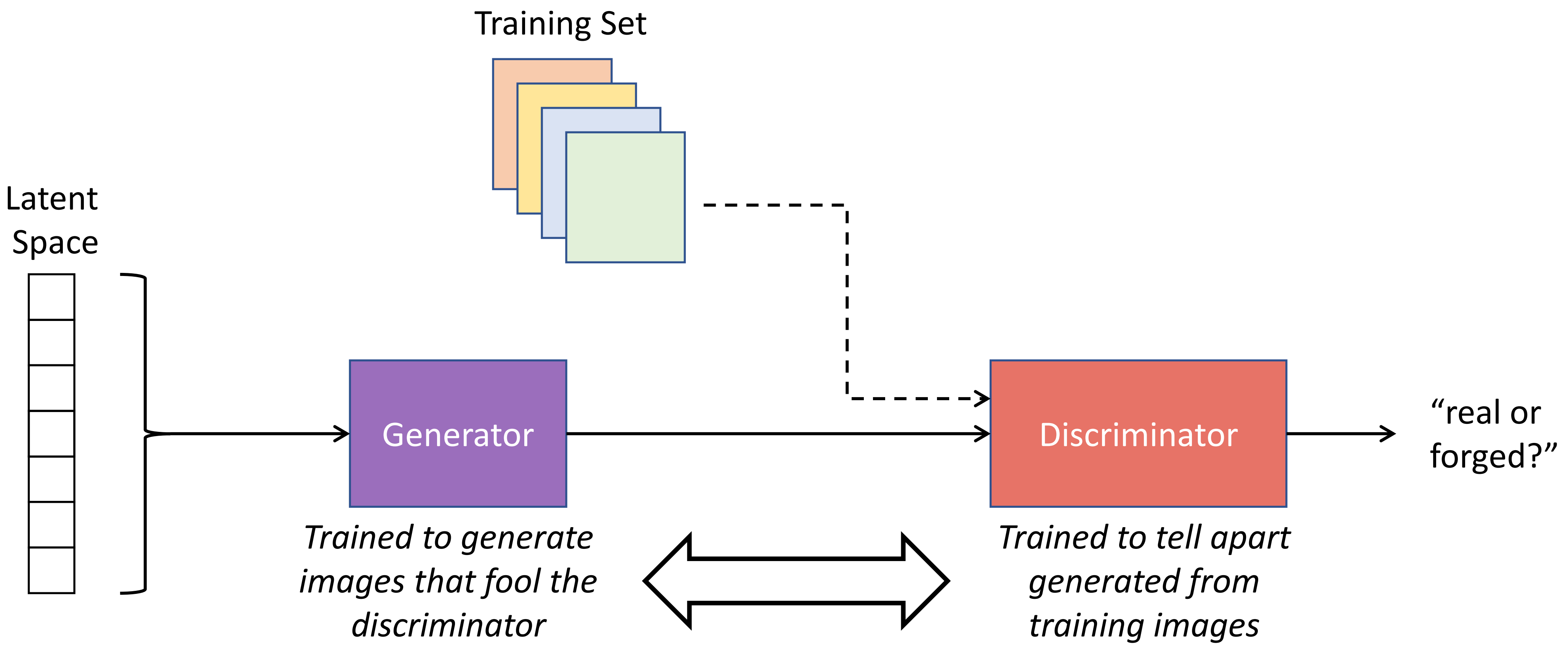
Outline

1. GAN Approach
2. GAN Architectures
3. Image translation: GAN vs. classical NN
4. Conclusion

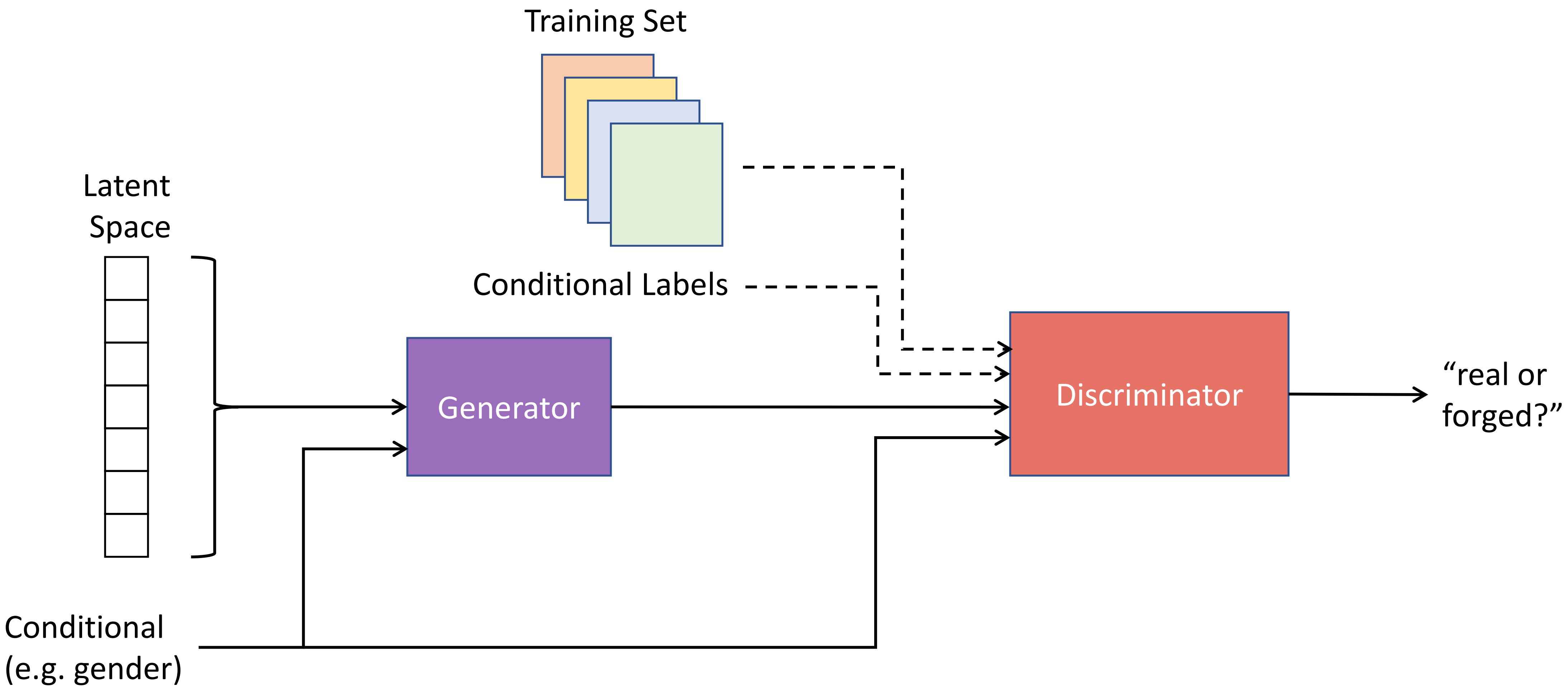
GAN Approach

- Task (example): generate images that look like faces
- Problem: hard to define explicitly what that means
  (“we know a face when we see it”)
  → Unclear what the cost function should be
- GAN method:
  Don’t predefine criteria to optimize for, instead use a second NN to tell you if a generated image is similar to a training set (of faces)
- Training the discriminator in step with the generator is a way of obtaining the “right” cost function for generating images that appear similar to those in a training set
- Can be applied to image-to-image translation by supplying image as conditional input (cGAN) e.g. generate faces based on sketch

Classical GAN Architecture



Conditional GAN (cGAN) Architecture



Translational cGAN Architecture (deterministically)

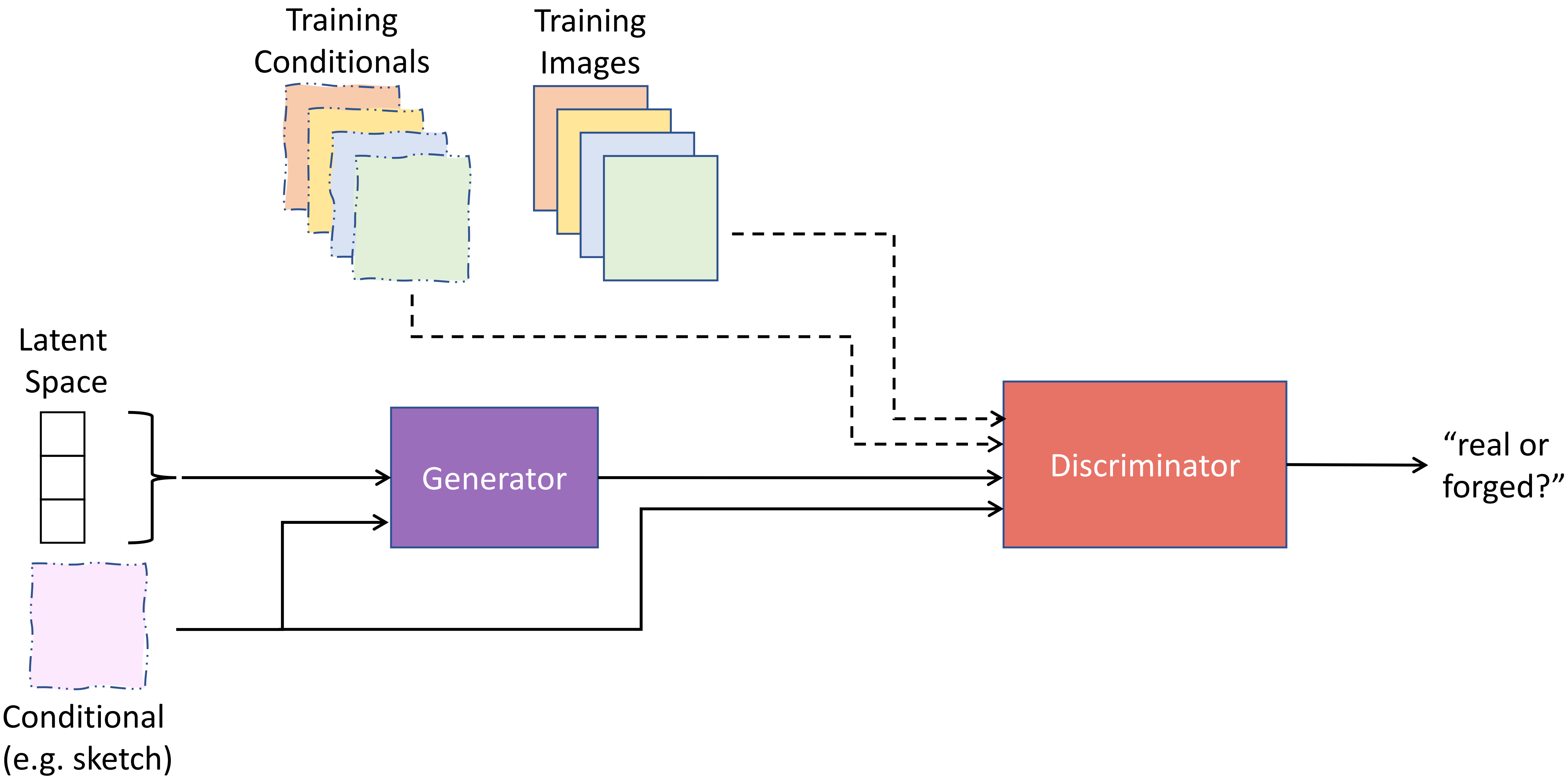


Image Translation Comparison: Classical NN vs. Translational cGAN

Classical NN:

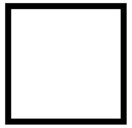
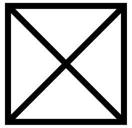
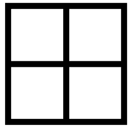
- Trained to perform image transformations which reduce the average training error as measured by predefined cost function
- Generated image compared to ground truth during training

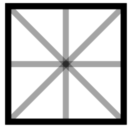
cGAN:

- No cost function explicitly defined, instead cost function is part of training (= discriminator)
- Goal of training: make it as difficult as possible to identify generated image as not from training set
- Generated image compared to image transformation in training set

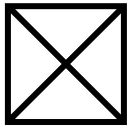
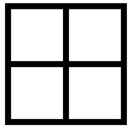
<u>What are the resulting differences in the generated images?</u>

Classical NN vs. Translational cGAN: Simplified Example

- Task: fill in the square: 
- Training data:  

- Standard NN with MSE loss: 

→Clearly not from dataset

- cGAN:  or  (depending on initialization)

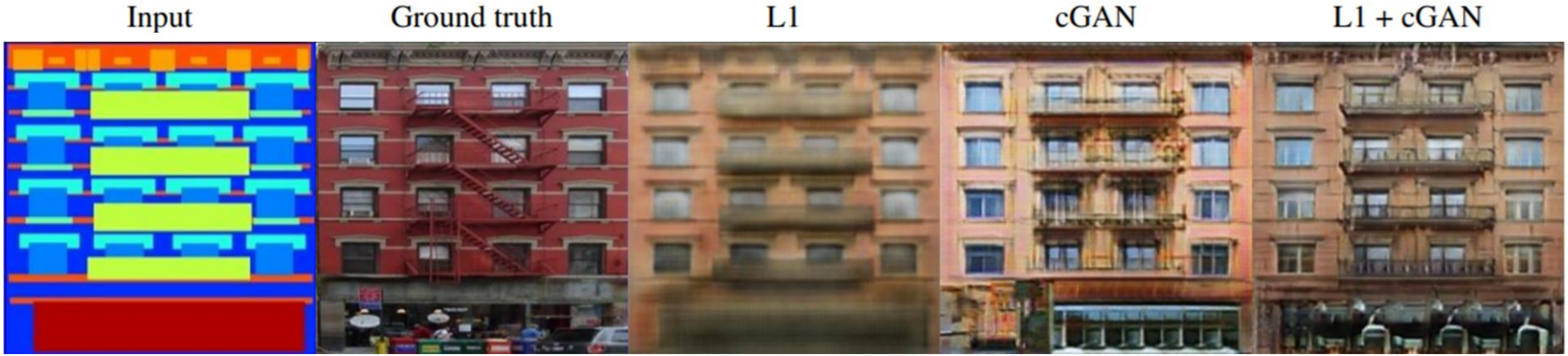
→Plausible result!

Classical NN vs. Translational cGAN: Realistic Example

- Task: Reconstruct facade from schematic
- Result not fully determined from training data → NN must fill in lack of information
- What is at the base of the building? Storefront, Garage, advertisement?
- Classical NN produces “average of possibilities”

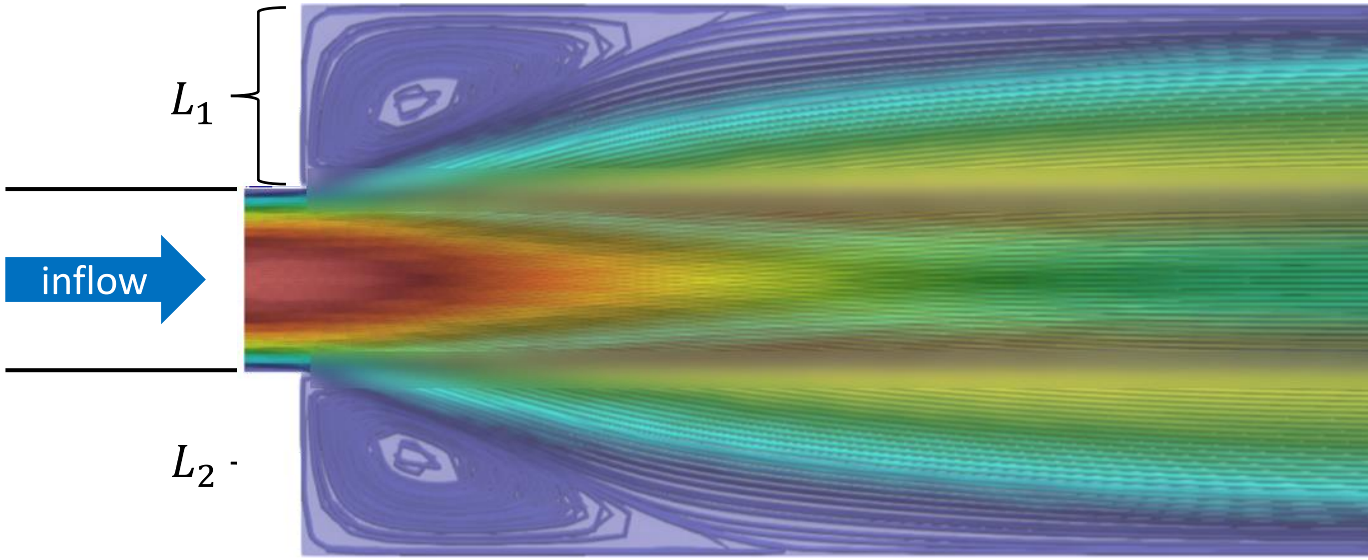
→ Minimizes cost function, but easy to detect as generated

- cGAN produces something concrete and plausible, but not necessarily closer to truth



Source: P. Isola, Image-to-Image Translation with Conditional Adversarial Networks, Berkeley AI Research (BAIR)

Classical NN vs. Translational cGAN: Flow Example



- Whether flow attaches at top or bottom depends on L_1/L_2
- NN may be trained to learn this relation
- For $L_1 = L_2$, the outcome is essentially random

→ A NN trained with MSE loss would most likely produce a superposition of both cases to minimize error

- A trained discriminator network could easily identify this as a “forgery”

→ Adversarially trained generator would likely learn to pick and choose one

Conclusion

- GANs help us find a suitable loss function (discriminator) on which to train the generator so that it produces images that are difficult to detect as not from the training set
- In the setting of flow prediction, this means optimizing our reduced order model for generating images that look like they could plausibly be real flow / CFD results
- If we can **explicitly define** what we are trying to optimize for (e.g. mean deviation from ground truth or adherence to PDE), GANs provide **no obvious advantage**
- That being said, how exactly adversarially trained generators compare to standard NN in terms of training, accuracy and generalization power remains to be seen