Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Alec Radford, Luke Metz, and Somit Chantal

Presented by Brandon Theodorou

Presentation Outline

- Related Work and Existing Limitations
- Problem Statement and Motivation
- Model Architecture and Novelties
- Experimental Details
- Results

Related Work

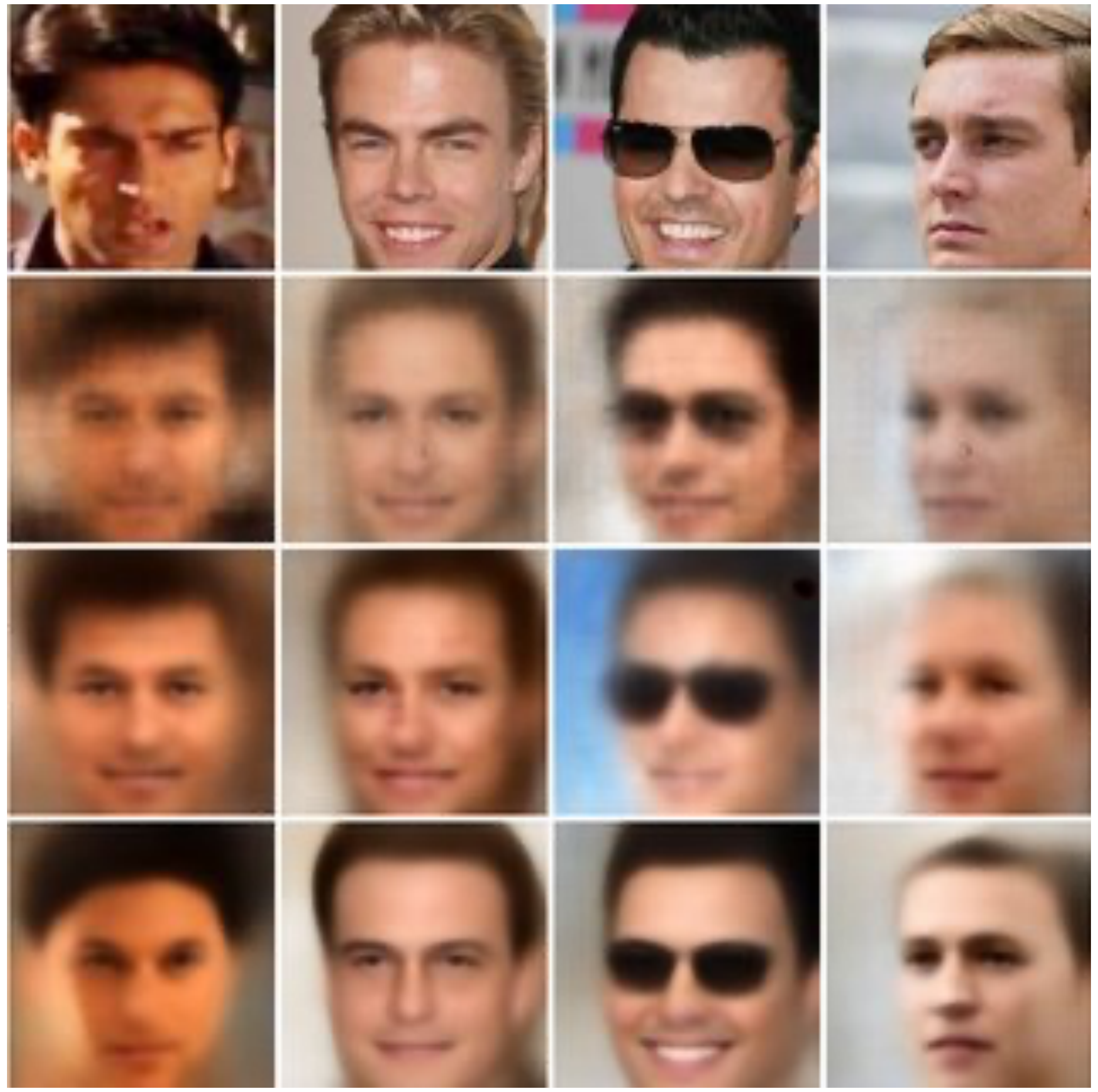
Unsupervised Representation Learning

- Well-studied
- Still relatively unrefined compared to modern results
  - K-Means
  - Autoencoders
  - Ladder Networks
  - Deep Belief Networks

Related Work

Generating Images

- Similarly well explored
- Poor generative output
  - VAEs
  - GANs
  - RNNs with Deconvolutions
- Blurry/Wobbly Images



Related Work

Existing GAN Models

- Exciting architecture and idea
- Generative output quality not yet crisp
  - “Noisy and incomprehensible”
- Unsuccessful using CNNs to model images
  - LAPGAN utilized a different, iterative upscaling approach
  - Problem specifically scaling up CNN architectures from supervised learning literature

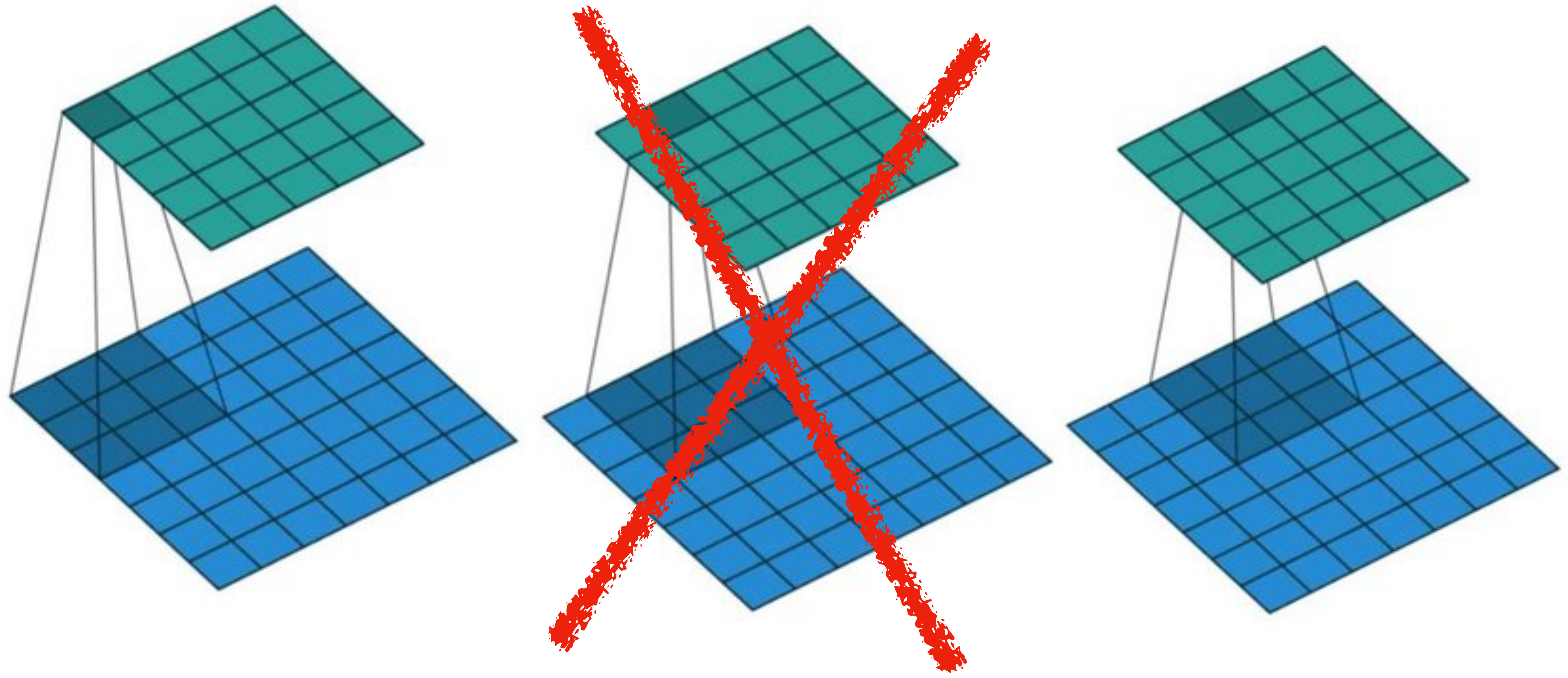
Problem Statement

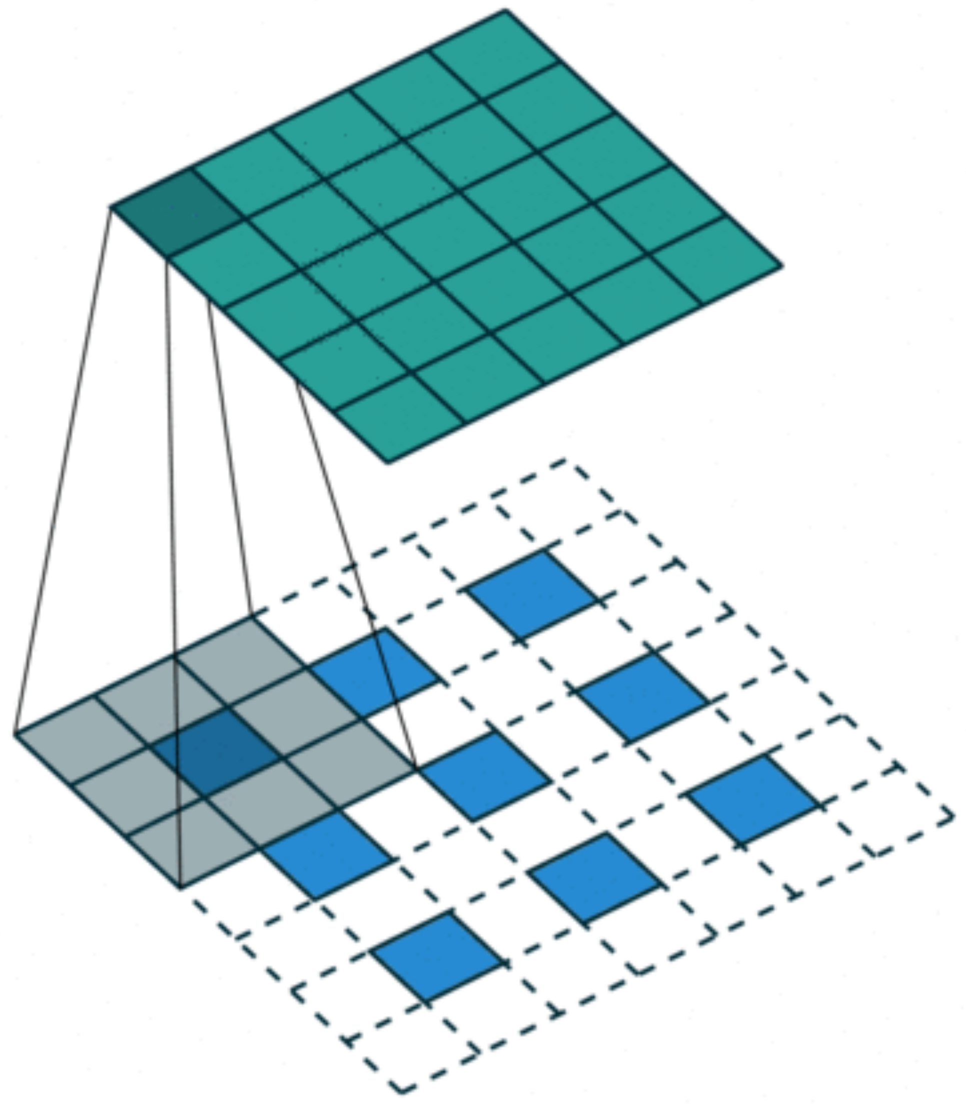
Develop an architecture to utilize CNNs within GANs in order to stabilize training and unsupervisedly learn a strong image representation

Model Architecture

Eliminate Pooling Layers

- Uses strided convolutions in place of pooling layers
  - Allows the network to learn its own upscaling and downscaling algorithms

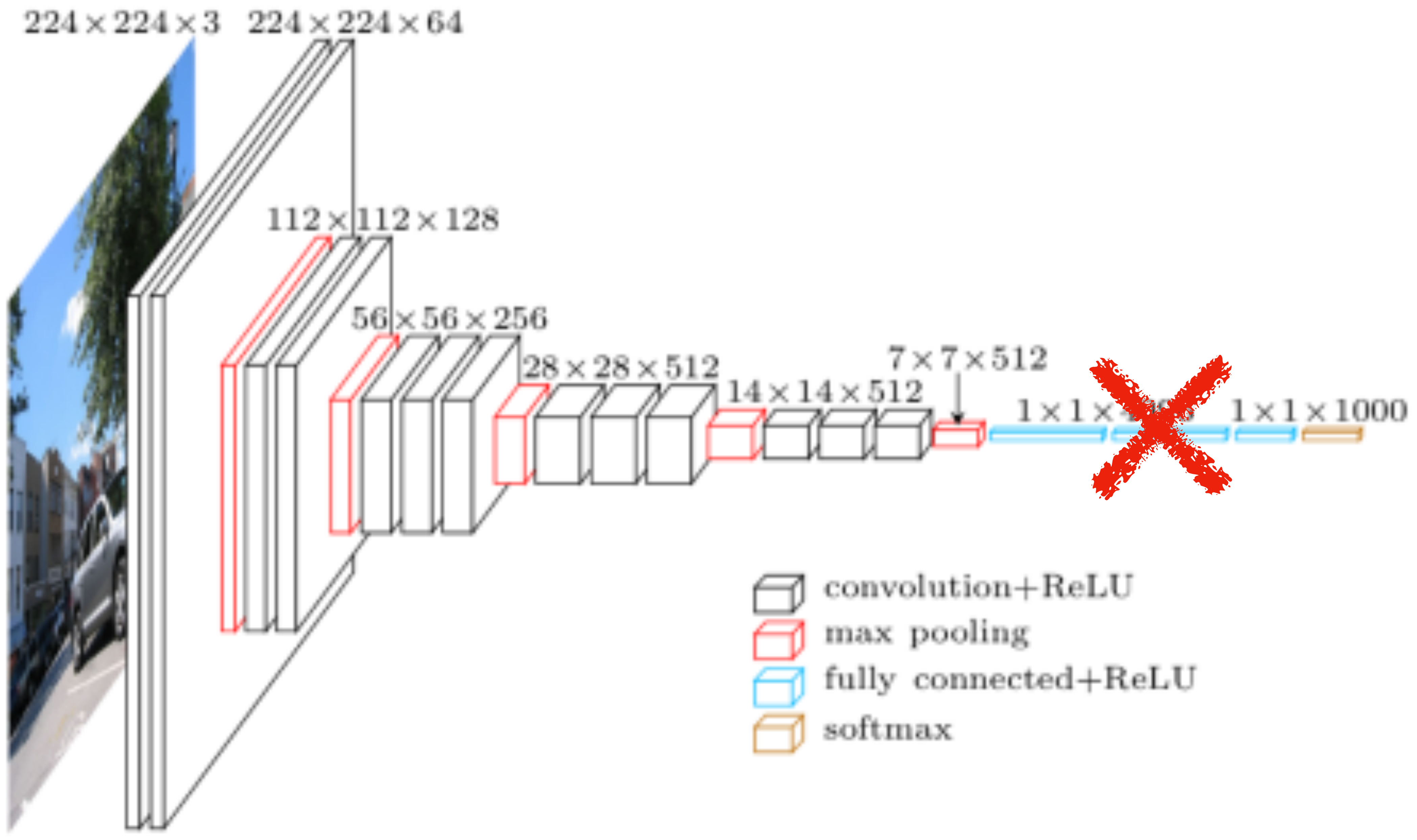




Model Architecture

Remove Fully Connected Layers

- Eliminate fully connected head on top of convolutional layers
  - Extreme is global average pooling
    - Helped stability but hurt convergence rate
  - Final convolution layer instead fed into a single sigmoid layer
- Only other fully connected layer is initial generation layer matrix multiplication to reshape noise



Model Architecture

Using Batch Normalization

- Normalizes the input to each layer to have zero mean and unit variance
- Stabilizes training and improves gradient flow for deep networks
- Helps with mode collapse
- Applied to all but last generator and first discriminator layers

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

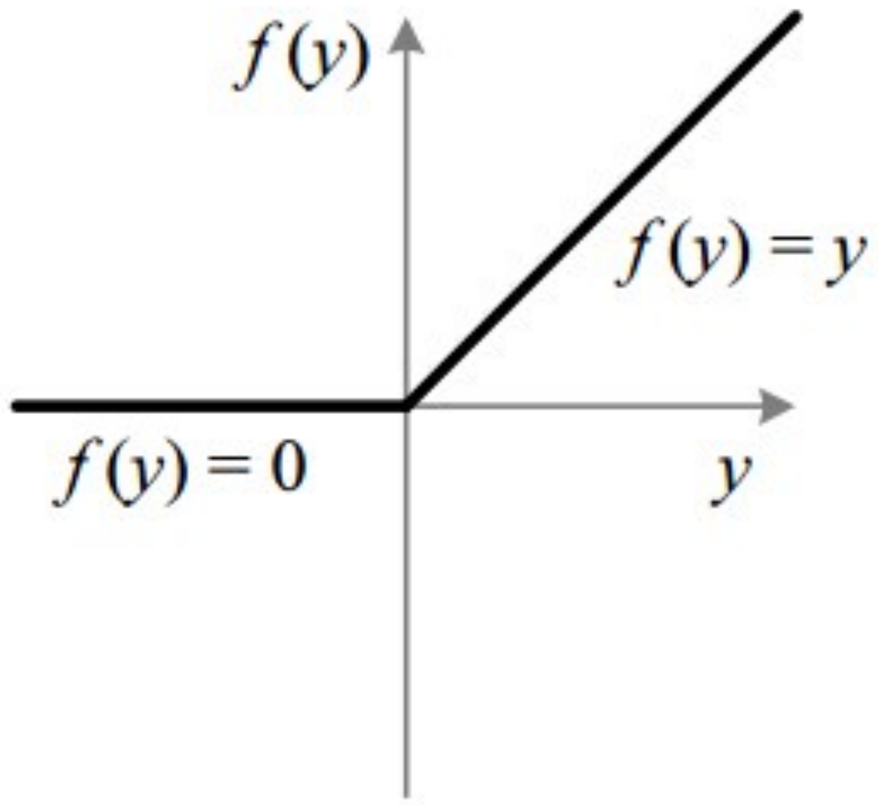
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

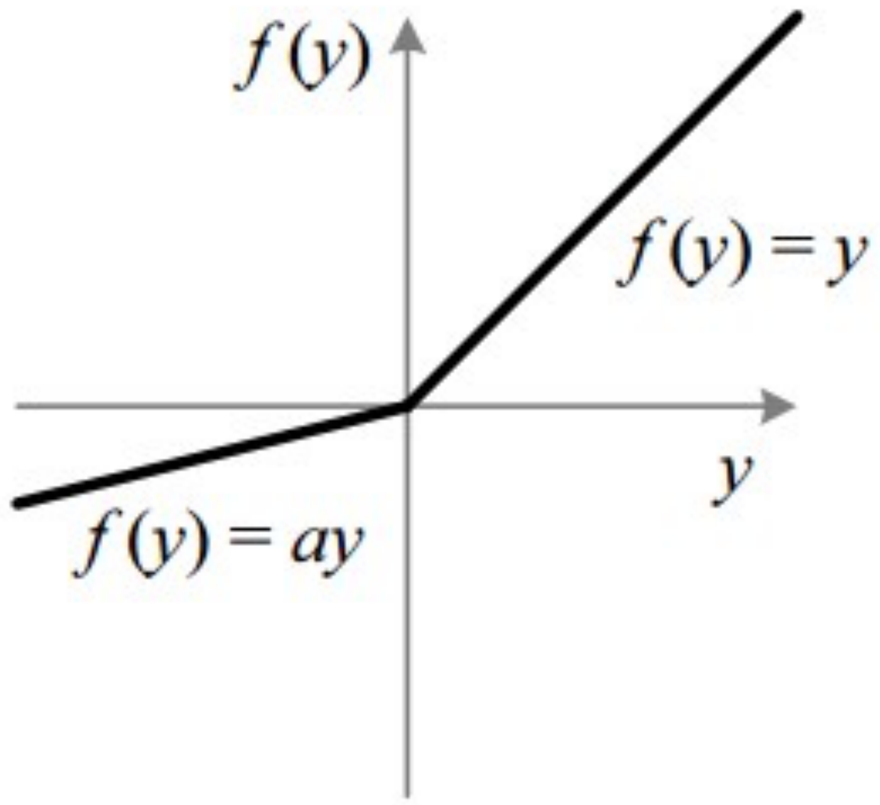
$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

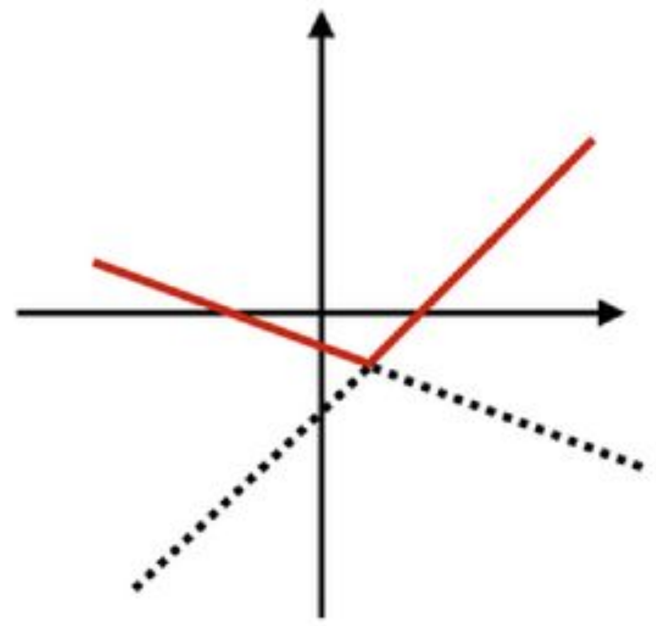
Model Architecture

Activation Functions

- Tanh for generator output
- ReLU in generator otherwise
- LeakyReLU in discriminator
- Original GAN used Maxout







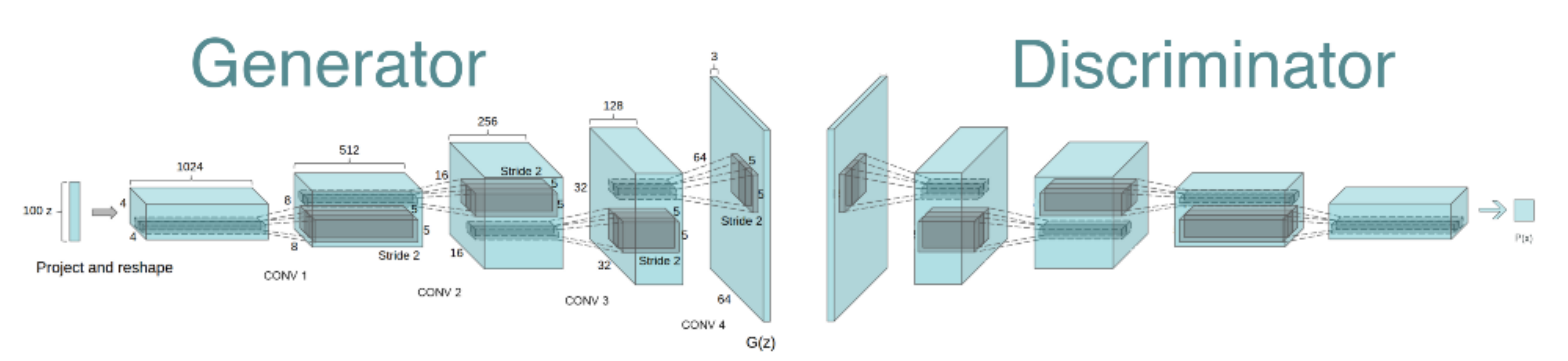
Maxout (n=2)

$$g(z) = max(w_1 x + b_1, w_2 x + b_2, \ldots, w_i x + b_i)$$

Model Architecture

Overview

- All CNNs
    - No pooling, no fully connected layers
- Utilize batch normalization
- ReLU (+Tanh) for the generator, LeakyReLU for the discriminator
- Simple changes but the result of lots of experimentation



Experimental Details

Datasets

- 3 Main Datasets
  - LSUN
  - ImageNet-1k
  - Faces
- Only preprocessing is scaling to range [-1,1]
- No Data Augmentation

Experimental Details

Training Specifications

- Mini-Batch SGD
- LeakyReLU slope of 0.2
- Adam Optimizer
- Learning Rate of 0.0002
- Momentum β_1 = 0.5

Results

Analysis of Possible Memorization

- Analysis of limited training
  - 1 Epoch, Small LR
- Hashing model
  - Deduplication and analysis

Results

Classifying CIFAR-10 Using DCGAN Features

- Use DCGAN as feature extractor with linear model on top for supervised learning task
- Discriminator feature maps max pooled to same 4x4 size, concatenated, and flattened to 28672 dimensional vector
- Beat strong K-means benchmark

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% ($\pm$0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% ($\pm$0.7%)	3200
View Invariant K-means	81.9%	72.6% ($\pm$0.7%)	6400
Exemplar CNN	84.3%	77.4% ($\pm$0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% ($\pm$0.4%)	512

Results

Classifying SVHN Numbers Using DCGAN Features

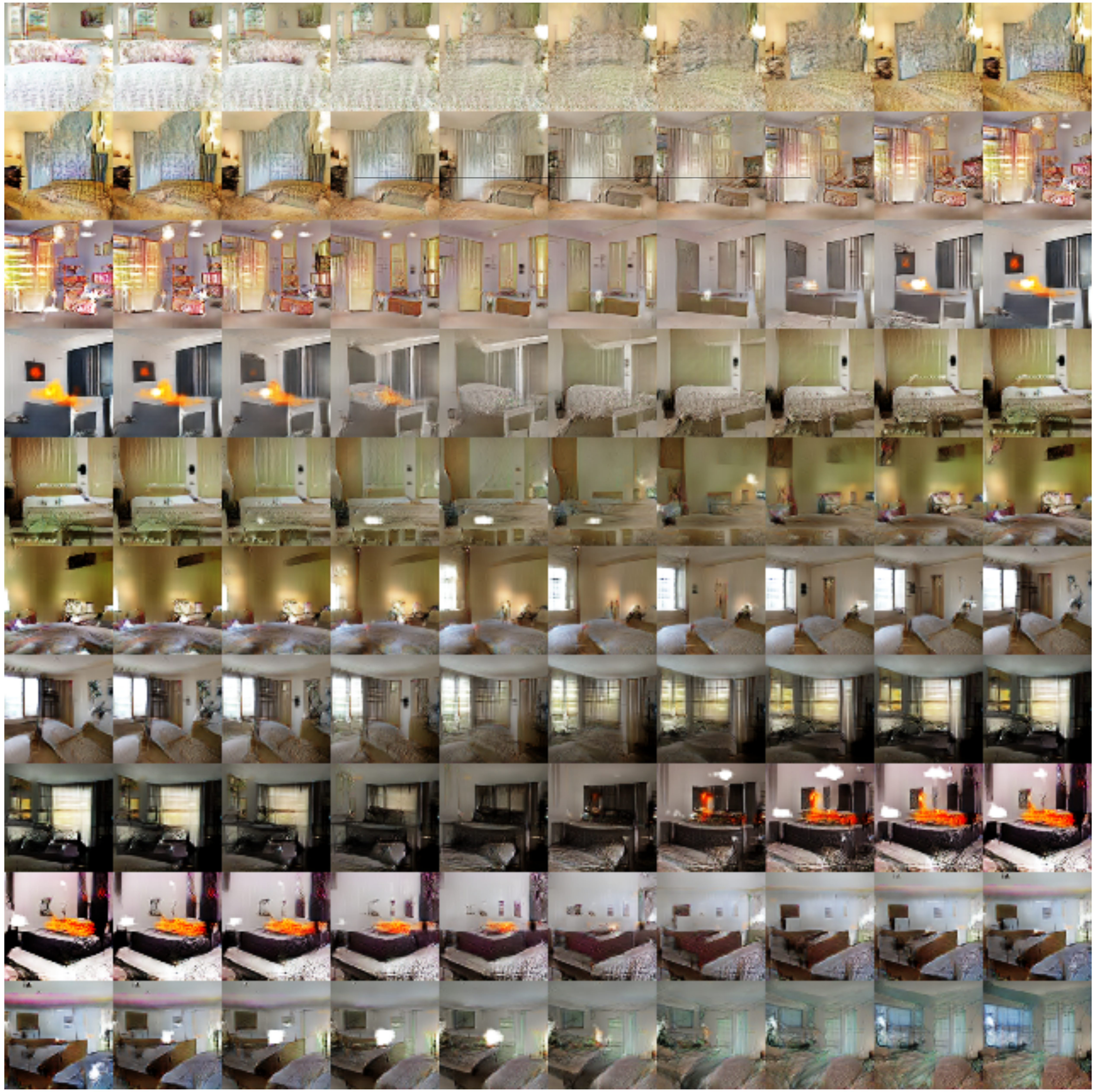
- Same setup as CIFAR-10
- Here achieves state of the art
- Makes sure architecture is not the key by supervisedly training CNN with the same architecture

Model	error rate
KNN	77.93%
TSVM	66.55%
M1+KNN	65.63%
M1+TSVM	54.33%
M1+M2	36.02%
SWWAE without dropout	27.83%
SWWAE with dropout	23.56%
DCGAN (ours) + L2-SVM	22.48%
Supervised CNN with the same architecture	28.87% (validation)

Results

Exploring the Latent Space

- Pick two random points in the latent space, generate outputs along the connecting line
- Checks for memorization
- Evaluates quality of representation

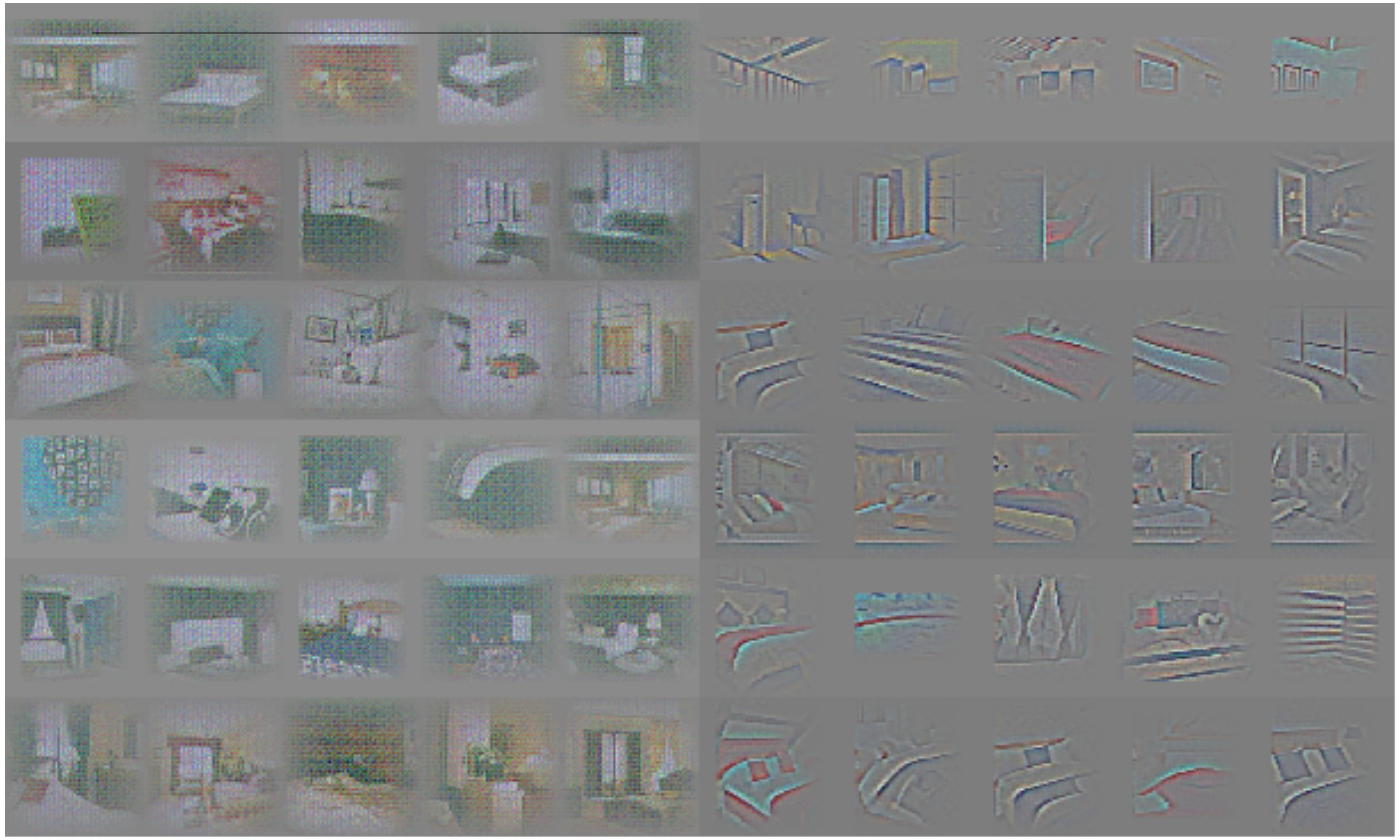


Results

Visualizing Discriminator Features

- Use guided back propagation to find exemplar activations of learned features

- Can see bedroom features corresponding to LSUN dataset



Random filters Trained filters

Results

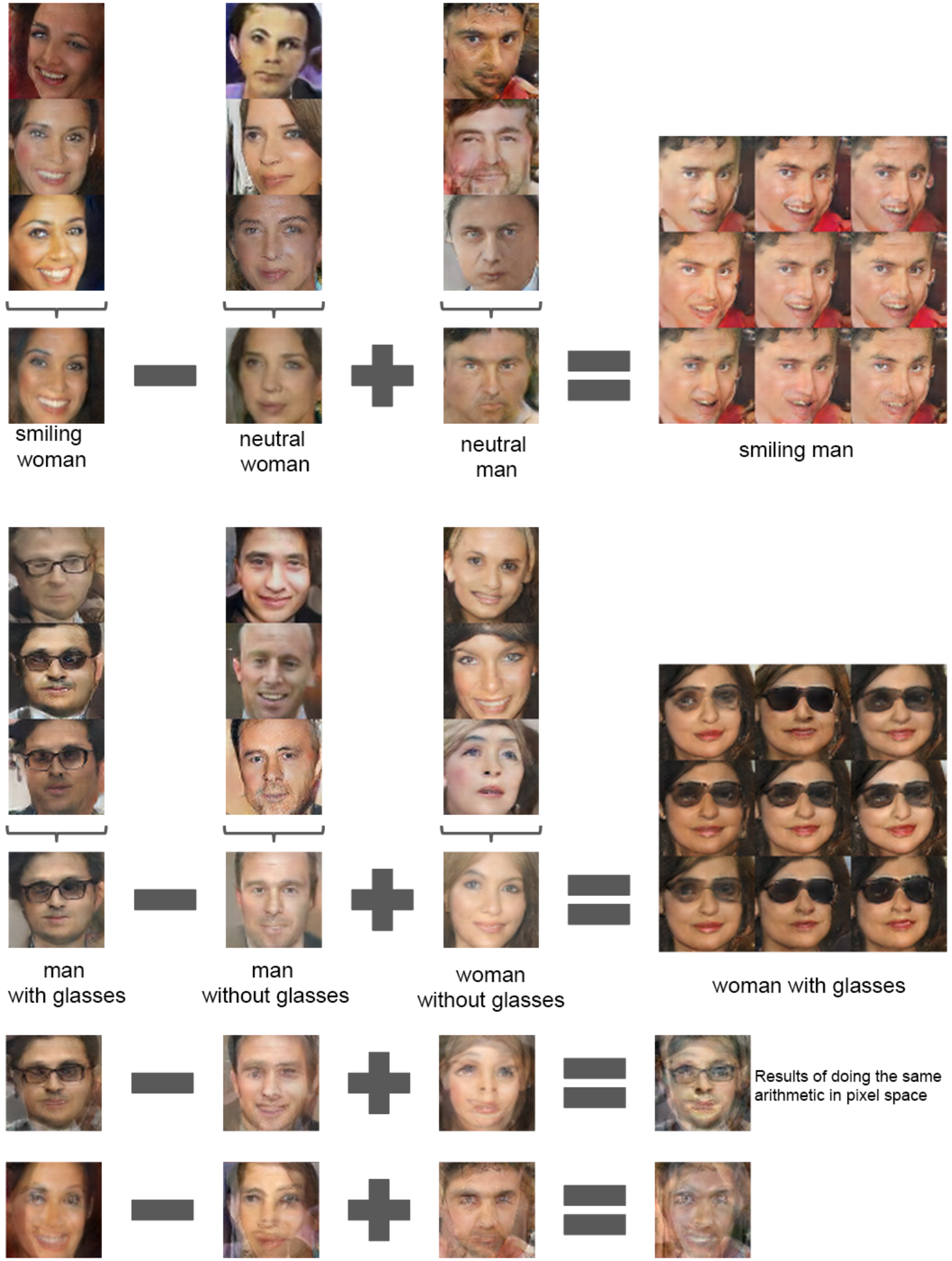
Removing Features in Generations

- Using manual analysis and logistic regression, identify window features
- During forward pass, dropped all positive values for these features and replaced with noise
- Images do not have windows but remain semantically sound

Results

Performing Vector Arithmetic

- Vector manipulation similar to Word2Vec
- Use average of multiple images rather than single image for stability



Results

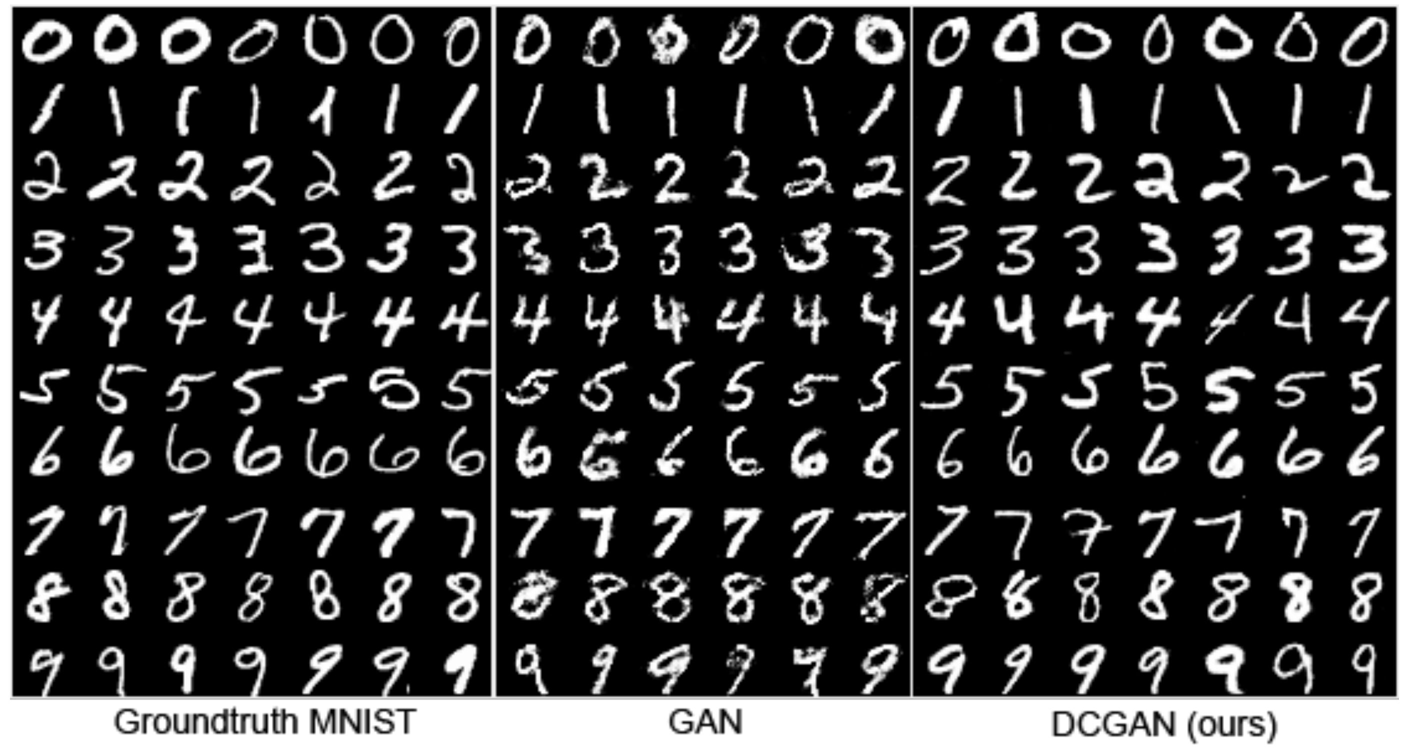
Conditional DCGANs

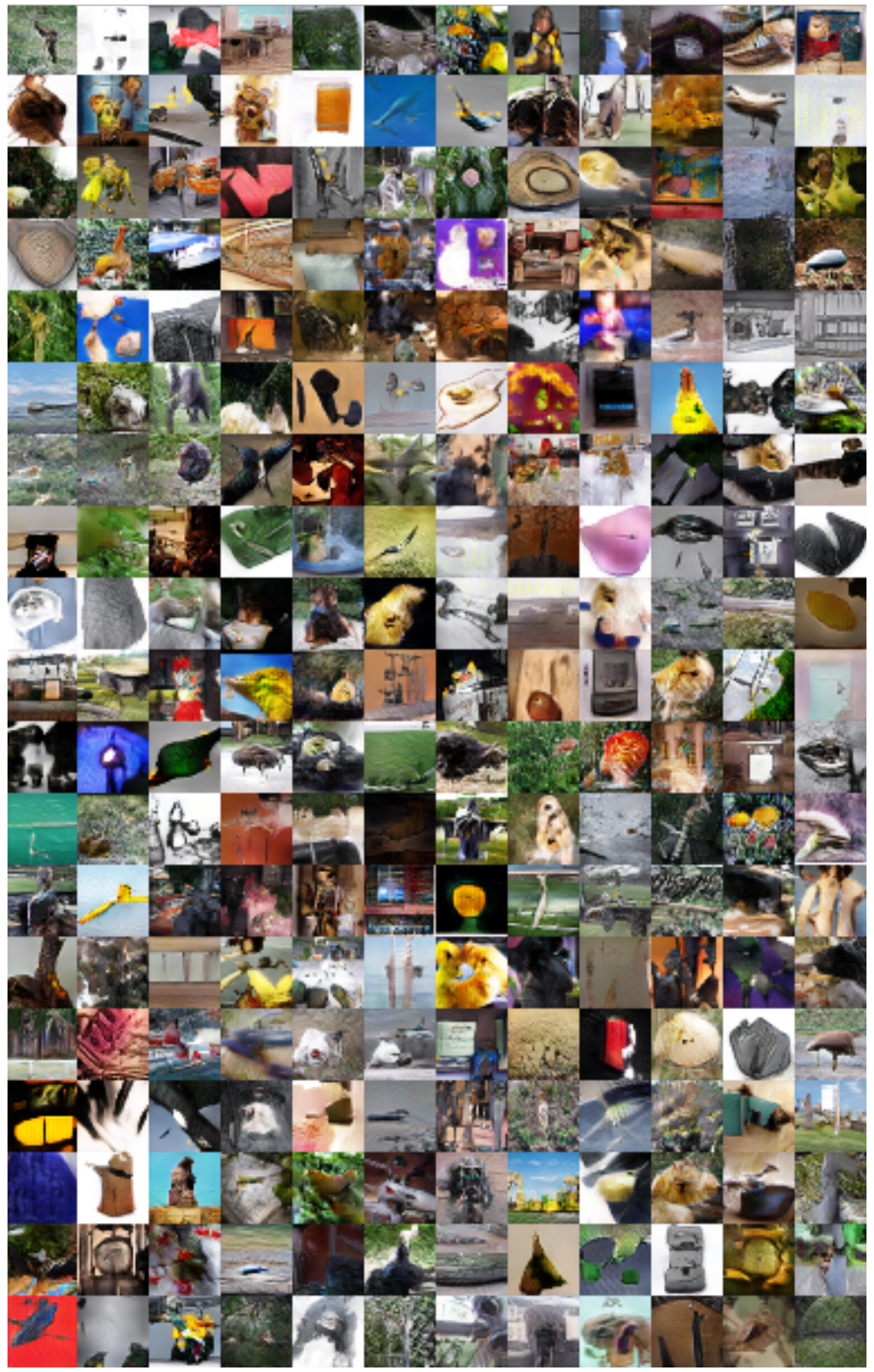
- Trained a conditional version of the model
- Evaluated using a nearest neighbor classifier on the test dataset

Model	Test Error @50K samples	Test Error @10M samples
AlignMNIST	-	1.4%
InfiMNIST	-	2.6%
Real Data	3.1%	-
GAN	6.28%	5.65%
DCGAN (ours)	2.98%	1.48%

Conclusion

- Propose a CNN-only architecture for GANs which offers more stable training
- Learns strong representations and produces strong image generations
- Remaining work to improve generative capacity, handle mode collapse, and apply to other domains





Paper Citation

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.