

# GANViz: A Visual Analytics Approach to Understand the Adversarial Game

# Generative Adversarial Networks



Generator

Discriminator



Real Money



Counterfeiter prints fake money. It is labeled as fake for police training. Sometimes, the counterfeiter attempts to fool the police by labeling the fake money as real.

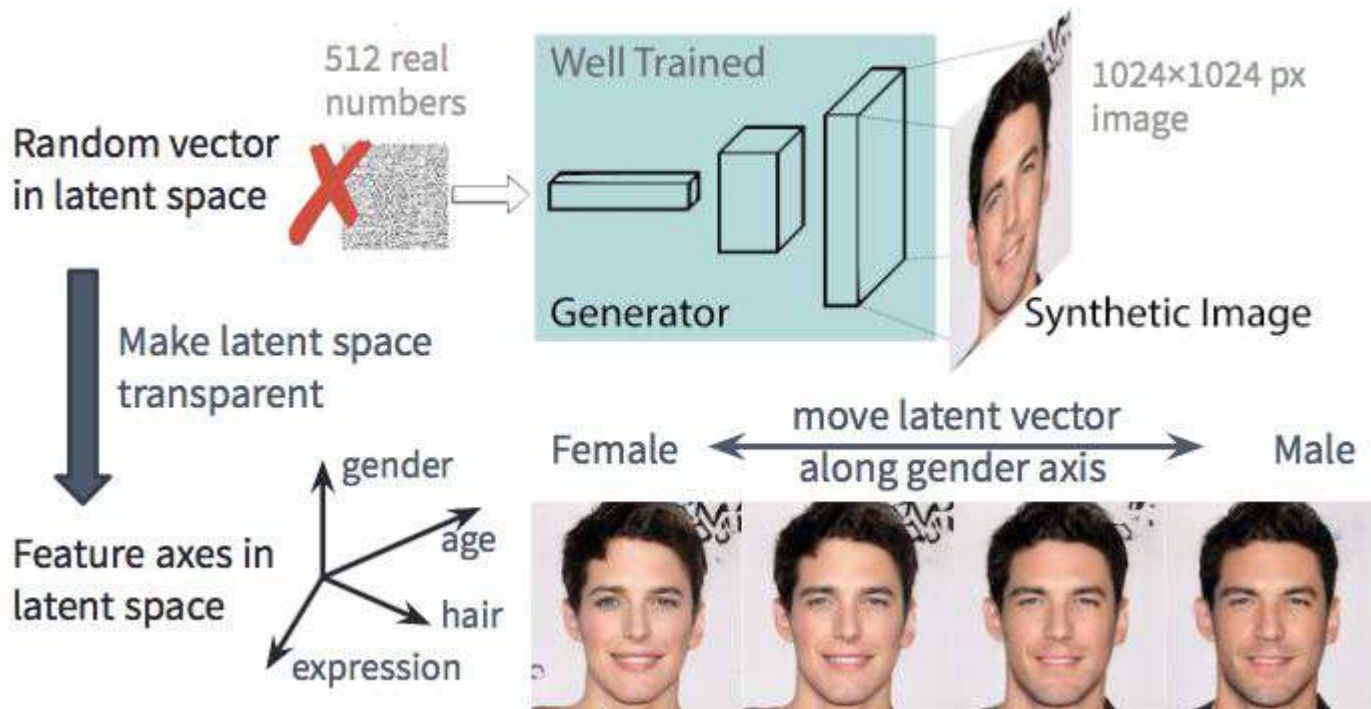


Fake Money



The police are trained to distinguish between. Sometimes, the police give feedback to the counterfeiter about why the money is fake.

# Latent Space

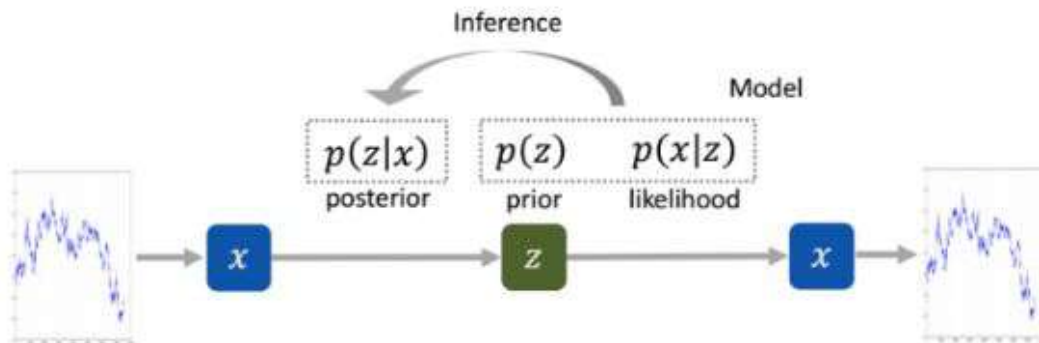


# Latent Space

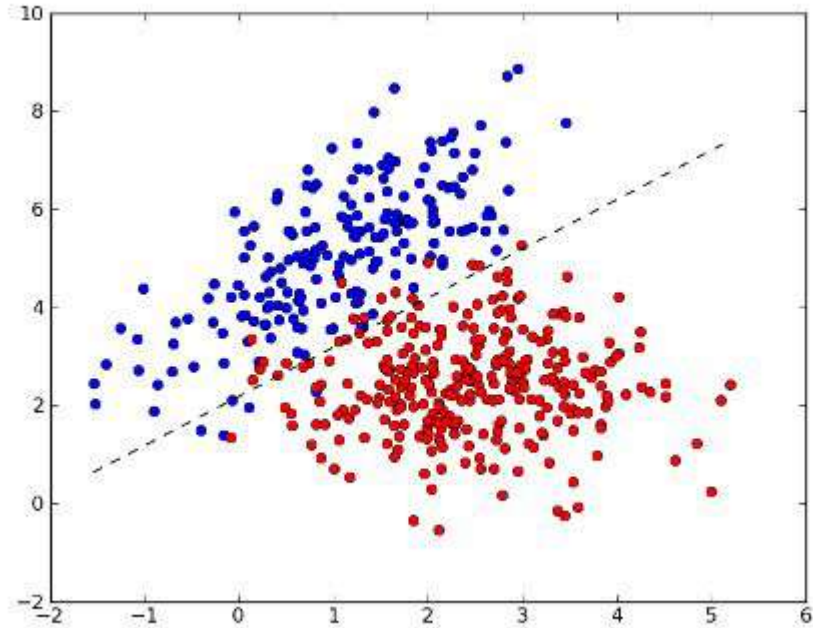
It refers to an *abstract multi-dimensional space* containing feature values.

Some examples of latent space are:

- 1) Word Embedding Space - word vectors where words similar in meaning have vectors that lie close to each other in space (cosine-similarity or euclidean-distance).
- 2) Image Feature Space - CNNs in the final layers encode higher-level features in the input image that allows it to effectively detect.
- 3) VAE & GANS - aim to obtain a latent space/distribution that closely approximates the real latent space/distribution of the observed data.



# Binary Classification



# Binary Classification

$$Loss = (Y)(-\log(Y_{pred})) + (1 - Y)(-\log(1 - Y_{pred}))$$

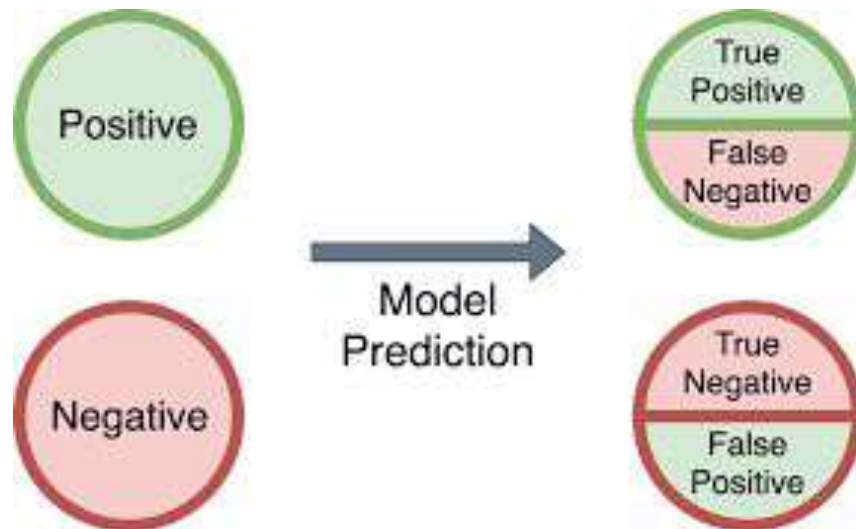
Remains when  $Y = 1$

Removed when  $Y = 0$

Remains when  $Y = 0$

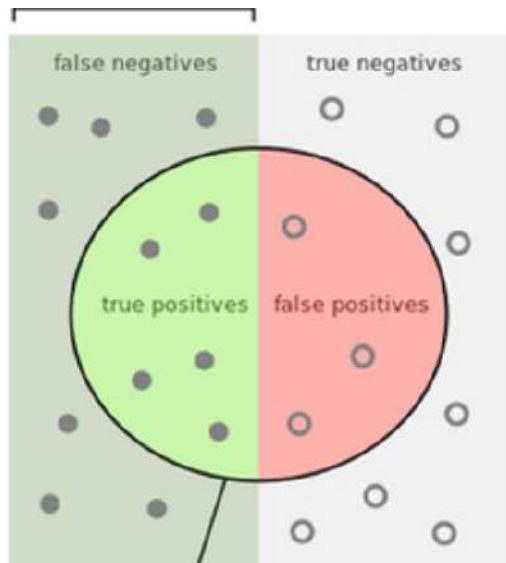
Removed when  $Y = 1$

# Metrics





# Metrics



How many selected items are relevant?

Precision =

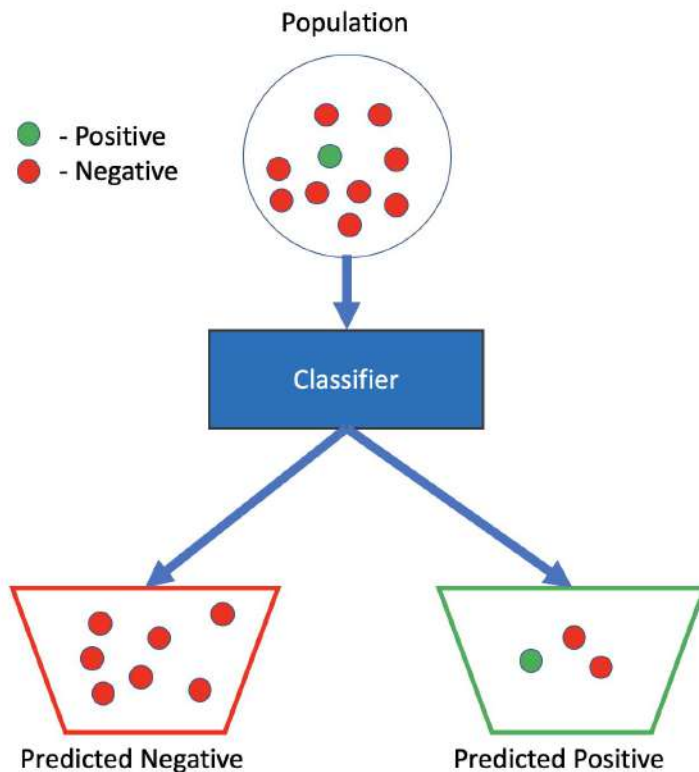


How many relevant items are selected?

Recall =



# Metrics



		Real	
		Positive	Negative
Predicted	Positive	1	2
	Negative	0	7

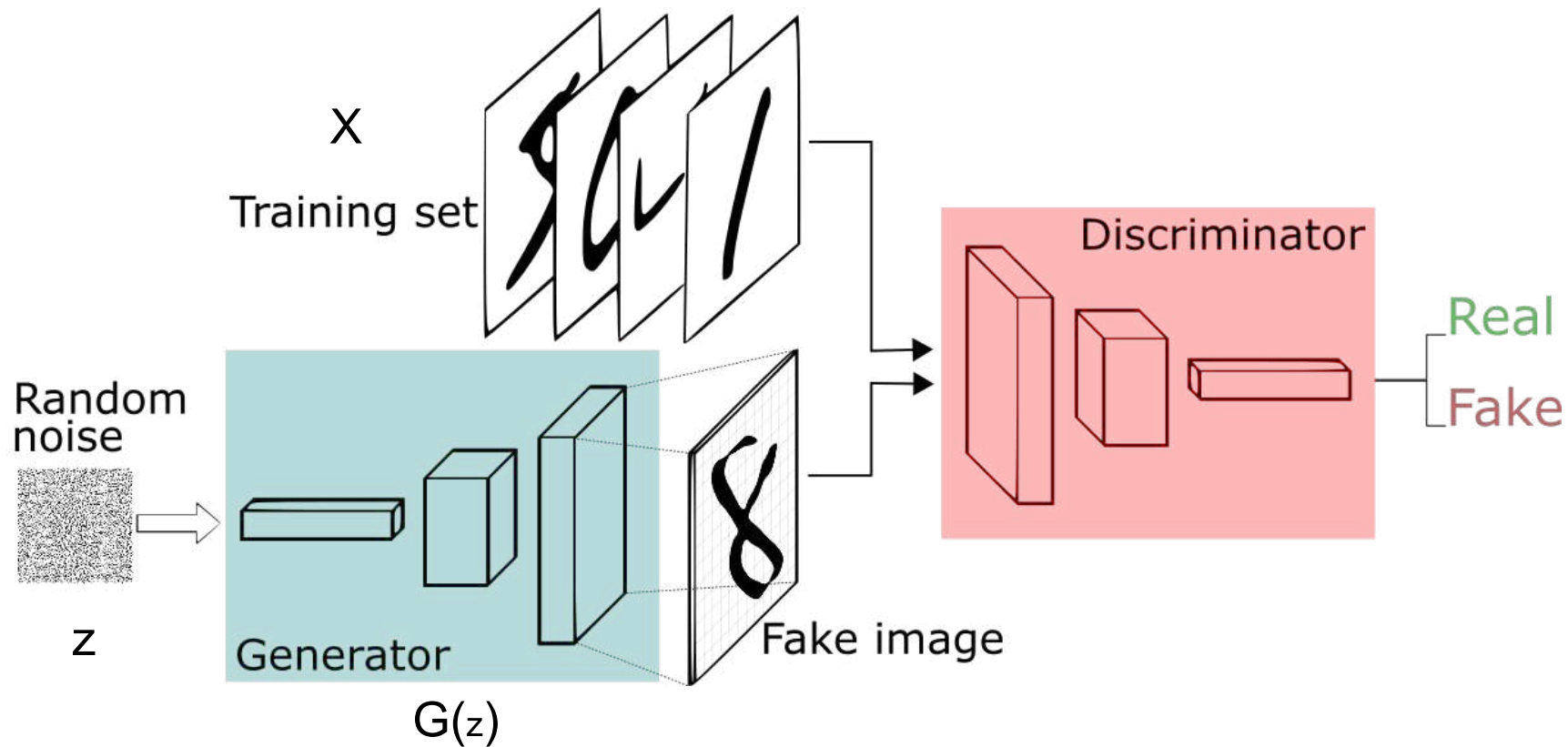
$$\text{precision} = \frac{tp}{tp + fp} = \frac{1}{3} = 33\%$$

$$\text{recall} = \frac{tp}{tp + fn} = \frac{1}{1} = 100\%$$

$$\text{specificity} = \frac{tn}{tn + fp} = \frac{7}{9} = 78\%$$

$$\text{sensitivity} = \text{recall} = 100\%$$

# GAN: Generative Adversarial Network



# GAN: Generative Adversarial Network

$D$  = Discriminator

$G$  = Generator

$\theta_d$  = Parameters of discriminators

$\theta_g$  = Parameters of generator

$P_z(z)$  = Input noise distribution

$P_{data}(x)$  = Original data distribution

$P_g(x)$  = Generated distribution

# GAN: Discriminator

From binary classification:

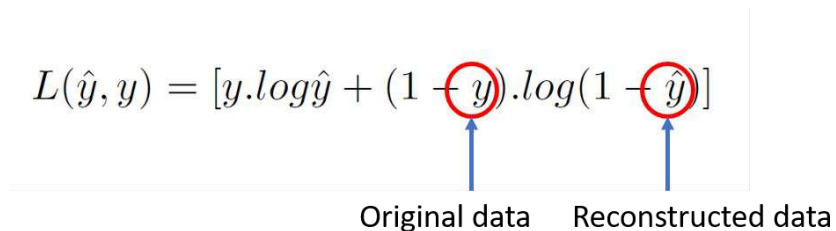
$$L(\hat{y}, y) = [y.\log\hat{y} + (1 - y).\log(1 - \hat{y})]$$

Original data

Reconstructed data

# GAN: Discriminator

From binary classification:

$$L(\hat{y}, y) = [y \cdot \log \hat{y} + (1 - y) \cdot \log(1 - \hat{y})]$$


Original data      Reconstructed data

During training, we have  $y=1$  for real data and  $D(x)$  for fake data.  
So, we obtain:

$$L(D(x), 1) = \log(D(x))$$

# GAN: Discriminator

From binary classification:

$$L(\hat{y}, y) = [y \cdot \log \hat{y} + (1 - y) \cdot \log(1 - \hat{y})]$$

Original data      Reconstructed data

In addition, for data coming from generator we have  $y=0$  for fake data and  $D(G(z))$  for “real” data. So, we obtain:

$$L(D(G(z)), 0) = \log(1 - D(G(z)))$$

## GAN: Discriminator Loss

$$L^{(D)} = \max[\log(D(x)) + \log(1 - D(G(z)))]$$



## GAN: Generator - Discriminator Loss

$$L^{(D)} = \max[\log(D(x)) + \log(1 - D(G(z)))]$$

$$L^{(G)} = \min[\log(D(x)) + \log(1 - D(G(z)))]$$

# GAN: Loss

$$L^{(D)} = \max[\log(D(x)) + \log(1 - D(G(z)))]$$

Recognize images better      Recognize images generated better

$$L^{(G)} = \min[\log(D(x)) + \log(1 - D(G(z)))]$$

Optimize the Generator to fool the discriminator

$$L = \min_G \max_D [\log(D(x)) + \log(1 - D(G(z)))]$$

# GAN: Generative Adversarial Network

$D$  = Discriminator

$G$  = Generator

$\theta_d$  = Parameters of discriminators

$\theta_g$  = Parameters of generator

$P_z(z)$  = Input noise distribution

$P_{data}(x)$  = Original data distribution

$P_g(x)$  = Generated distribution

$$\min_G \max_D V(D, G)$$

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

# GAN: Generative Adversarial Network

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

$$\mathbb{E}_{x \sim p(x)} [f(x)] = \int p(x) f(x) dx$$

$$\mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] = \int p_z(z) \log(1 - D(G(z))) dz$$

As we know, we want to approximate  $P(z)$  to  $P_G(x)$ , and  $G(z)$  to  $x$ . Then, we can re formulate the equation:

$$\mathbb{E}_{z \sim p(z)} = \int p_G(x) \log(1 - D(x)) dx$$

$$\arg \min_G \max_D V(D, G) = \int_x p_{data}(x) \log D(x) dx + \int_x p_G(x) \log(1 - D(x)) dx$$

# GAN: Optimizing the Discriminator

$$\begin{aligned}\arg \min_G \max_D V(D, G) &= \int_x p_{data}(x) \log D(x) dx + \int_x p_G(x) \log(1 - D(x)) dx \\ &= \int [P_{data}(x) \log D(x) + P_G(x) \log(1 - D(x))] dx\end{aligned}$$

Simplifying the equation:

$$\underbrace{P_{data}(x)}_a \underbrace{\log D(x)}_D + \underbrace{P_G(x)}_b \underbrace{\log(1 - D(x))}_D$$

We have:  $f(D) = a \log(D) + b \log(1-D)$ .  $f'(D) = 0$  (to minimize), so we obtain:

$$D = \frac{a}{a+b} = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$$

# GAN: Optimizing the Generator

$$\arg \min_G \max_D V(D, G) = \int_x p_{data}(x) \log D(x) dx + \int_x p_G(x) \log(1 - D(x)) dx$$

Replacing  $D(x)$  obtained from last slide:

$$V(D, G) = \int_x p_{data}(x) \log\left(\frac{p_{data}(x)}{p_{data}(x) + p_G(x)}\right) dx + \int_x p_G(x) \log\left(\frac{p_G(x)}{p_{data}(x) + p_G(x)}\right) dx$$

$$V(D, G) = V(D, G) + (\log(2) - \log(2)) p_{data}(x) + (\log(2) - \log(2)) p_G(x)$$

$$V(D, G) = \int_x p_{data}(x) \log\left(\frac{2p_{data}(x)}{p_{data}(x) + p_G(x)}\right) dx + \int_x p_G(x) \log\left(\frac{2p_G(x)}{p_{data}(x) + p_G(x)}\right) dx$$

$$V(D, G) = -\log 4 + KL(p_{data} || \frac{p_G(x) + p_{data}(x)}{2}) + KL(p_G || \frac{p_G(x) + p_{data}(x)}{2})$$

# GAN: Optimizing the Generator

$$\arg \min_G \max_D V(D, G) = \int_x p_{data}(x) \log D(x) dx + \int_x p_G(x) \log(1 - D(x)) dx$$

We want to find data distribution  $P_{data}(x)$  so, we have the distribution  $G(z)$  parameterized by  $\theta$ , so we have  $P_G(G(z), \theta)$ , taking  $G(z) = x$ . We compute the Likelihood (the likelihood is equal to the probability density at a particular outcome  $x$  when the true value of the parameter is  $\theta$ : if we have  $m$  samples from  $P_{data}(x)$ , the likelihood will be equal to:

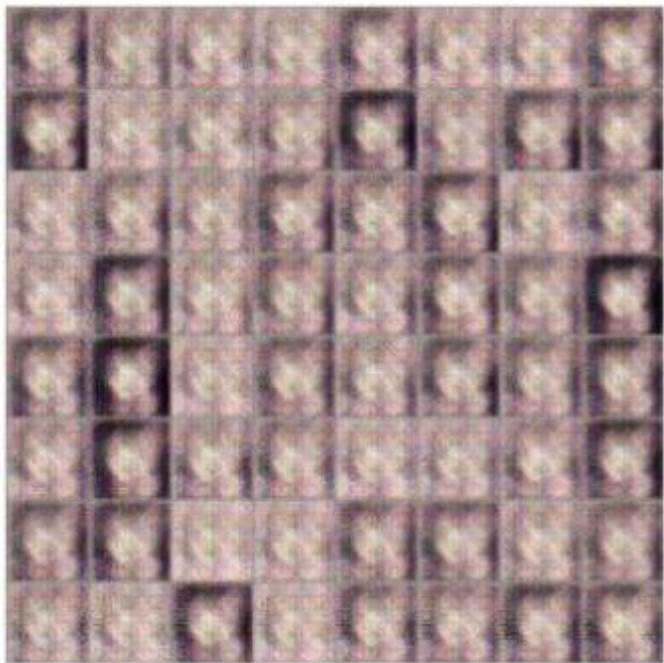
$$L = \prod_{i=1}^{\bar{m}} P_G(x^i; \theta)$$

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^m P_G(x^i; \theta) = \arg \max_{\theta} \log \prod_{i=1}^m P_G(x^i; \theta)$$



# Examples: Anime face generation

100 updates



1000 updates





# Examples: Anime face generation



5000 updates



10,000 updates

# Examples: Anime face generation



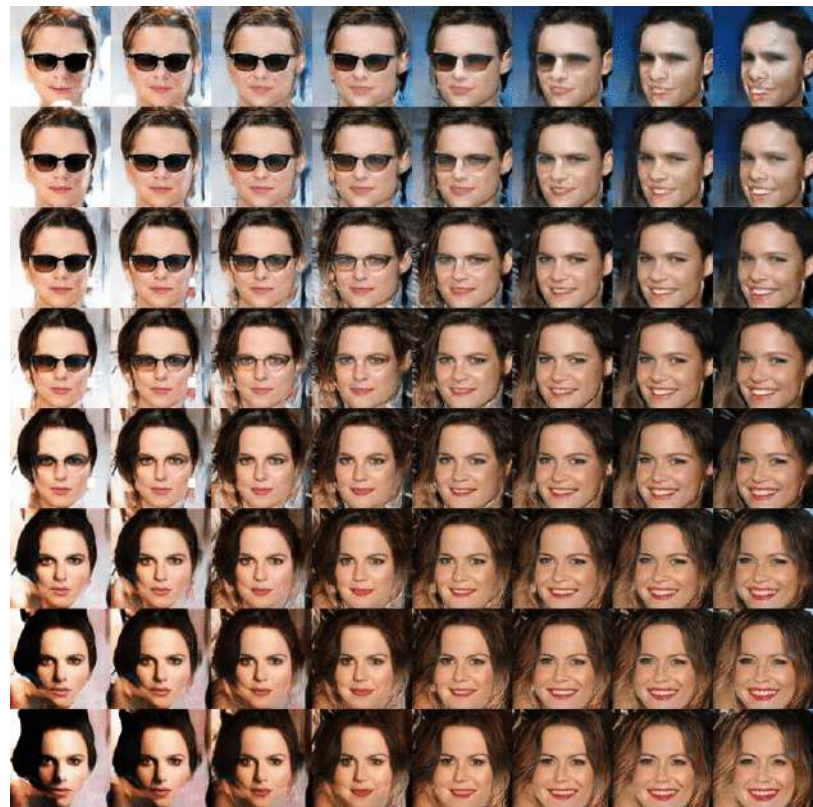
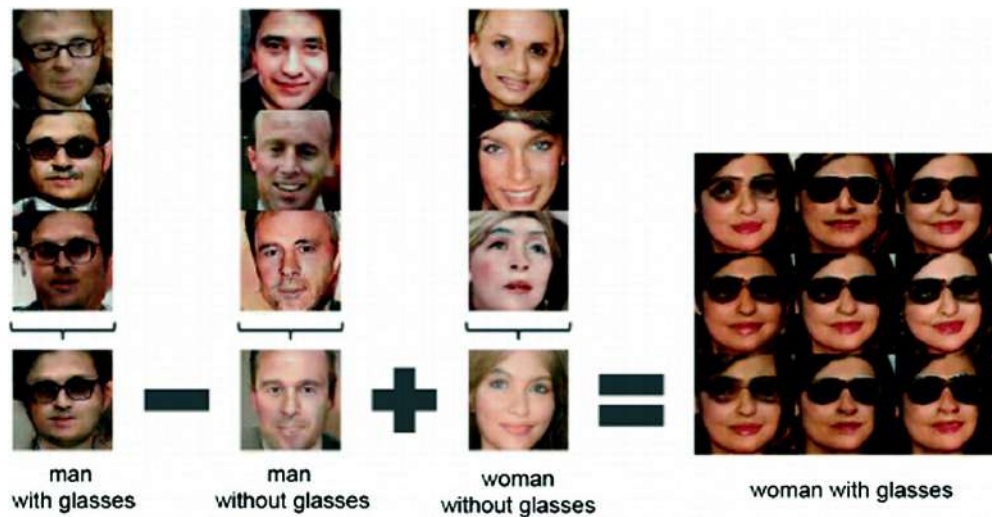
20,000 updates



50,000 updates



# Examples: Arithmetic of latent spaces in faces



# Examples



# Examples

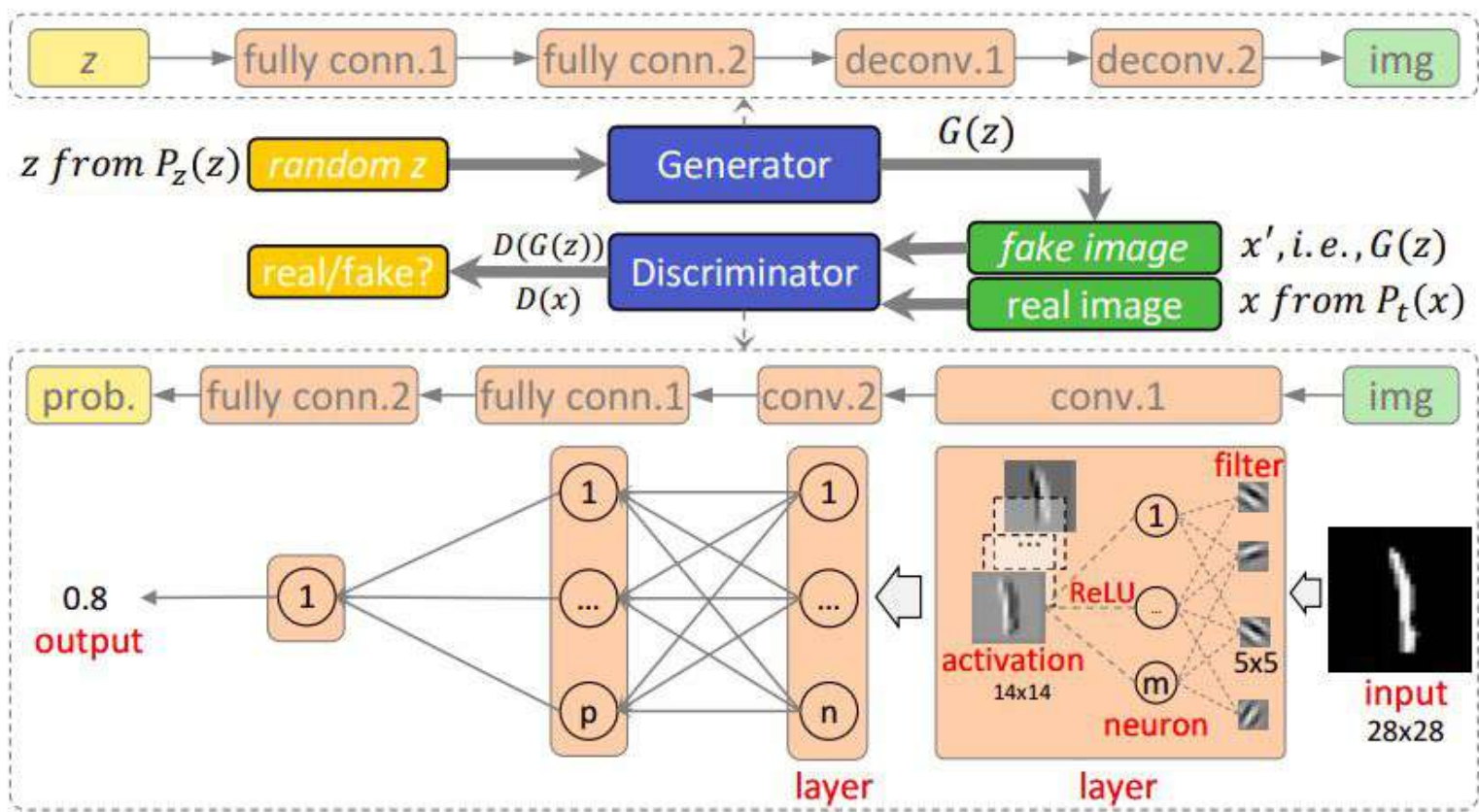
<https://www.thispersondoesnotexist.com/>

GANViz

# Understanding GANs

- What features does D learn to differentiate real/fake data?
- How do the patterns evolve over time?
- Is G learning data distributions only in a local feature space?

# GANviz





# GANViz: Data Types

- Values of the D and G losses.
- Probability values (likelihood to be real) for each real/fake image.
  - Predictions of the D-Net taking 256 real and 256 z vectors transformed to  $G(z)$ .
- Activation (from all layers) in the D-Net (ReLU values)
- Parameters of D-Net like filters and weights in Fully-Connected layers.

# GANViz: Metrics

- Area Under Curve (AUC): Performance of the D-Net measured by ROC-curves (Receiver Operating Characteristics) TP/(TP+FN) vs FP/(FP+TN)
- Diversity: (Structure Similarity SSIM):

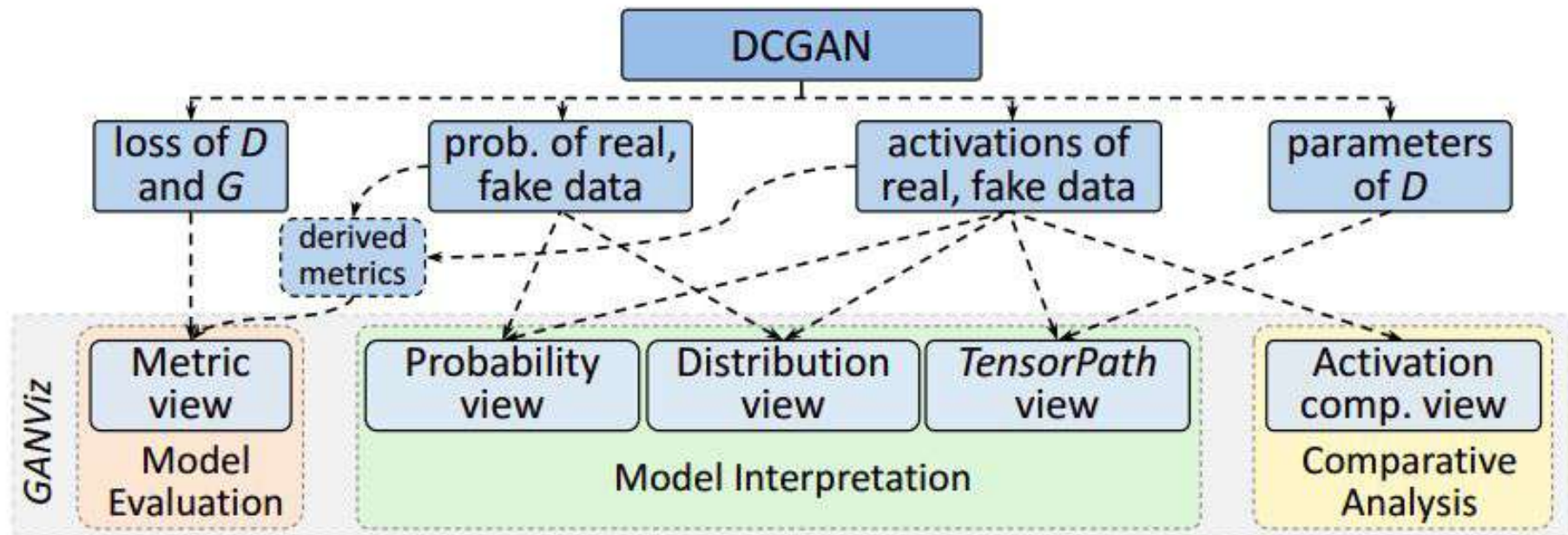
$$Diversity = 1 / \left( \frac{1}{n(n-1)/2} \sum_{i=1}^n \sum_{j=i+1}^n (SSIM(img_i, img_j)) \right).$$

- Dist-Diff (Distribution Difference): KL divergence

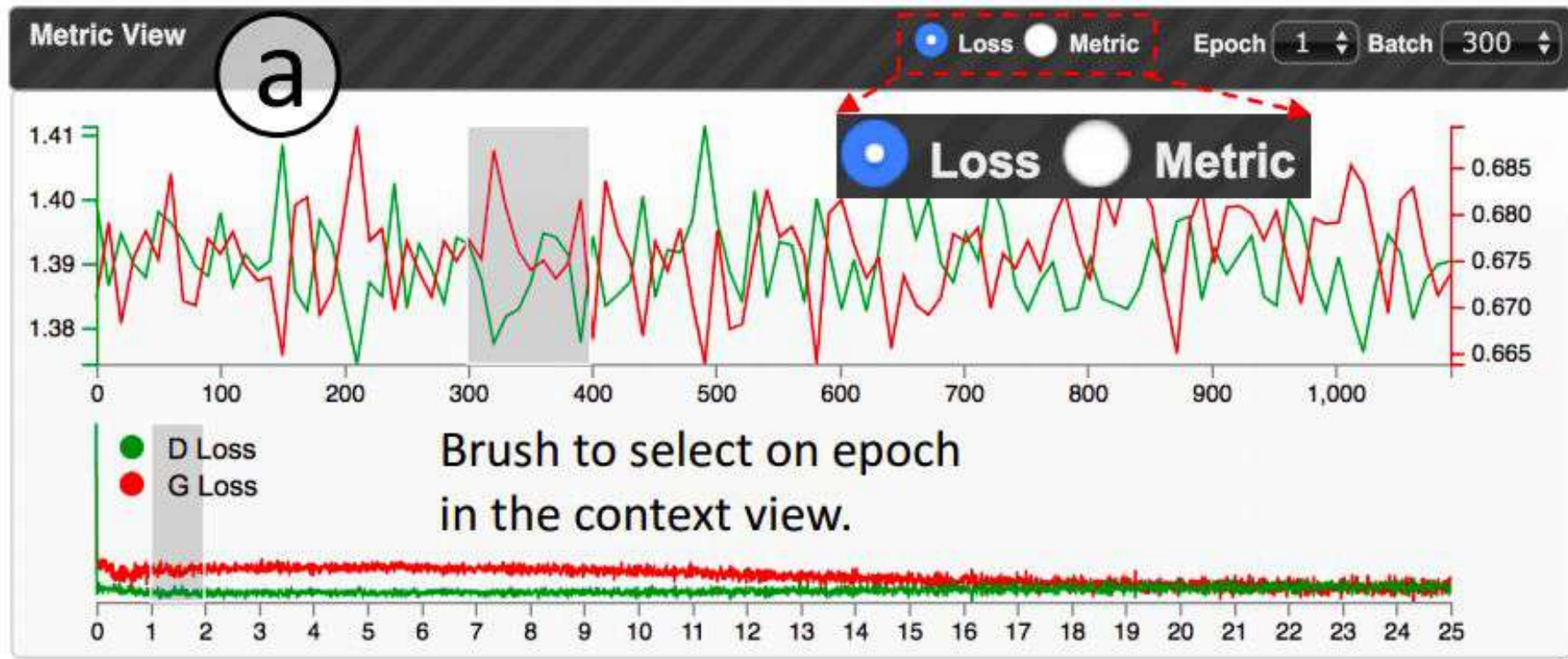
$$Dist - Diff = KL(GMM(PCA(p_{data}(x))) || GMM(PCA(p_G(x))))$$

# GANViz: Analytics System

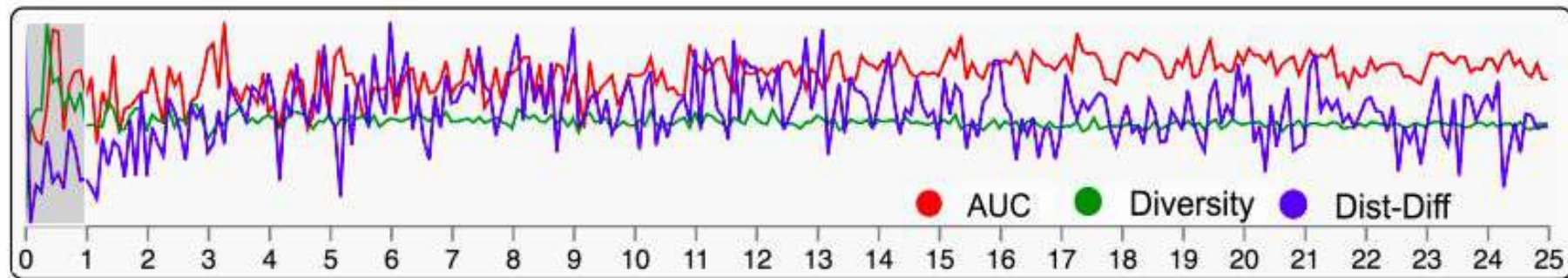
# GANviz: Analytics System



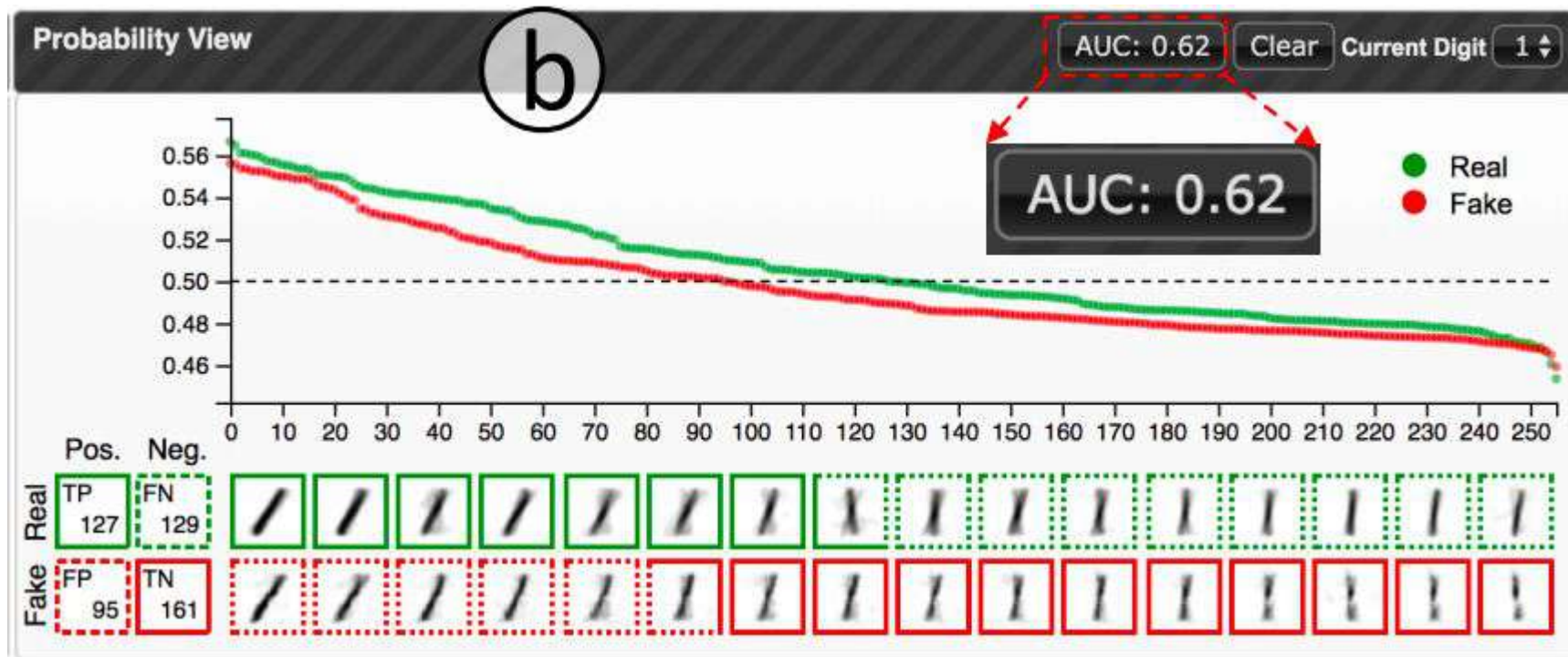
# GANViz: Metric View



# GANViz: Metric View



# GANViz: Probability View





# GANViz: Probability View

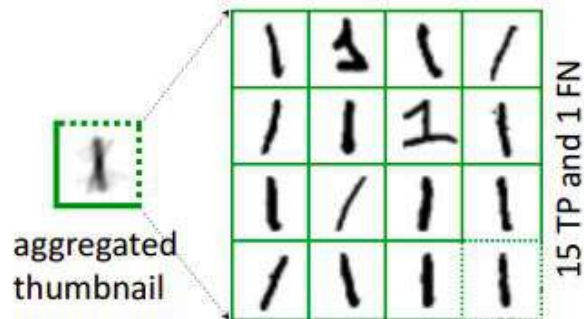
	positive	negative
real	TP	FN
fake	FP	TN



	Pos.	Neg.
Real	TP 127	FN 129
Fake	FP 95	TN 161

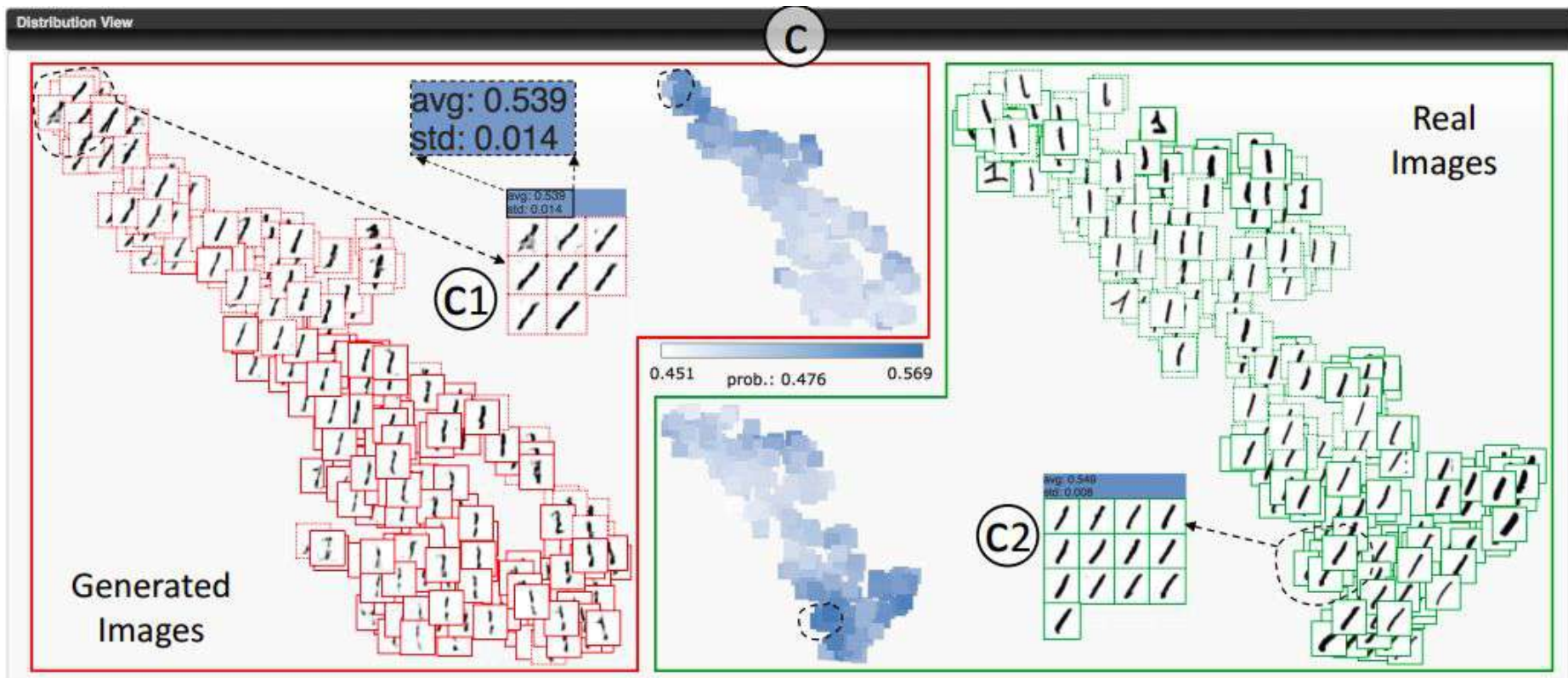
129 (out of 256) real images are considered as fake by  $D$

161 (out of 256) fake images are considered as fake by  $D$

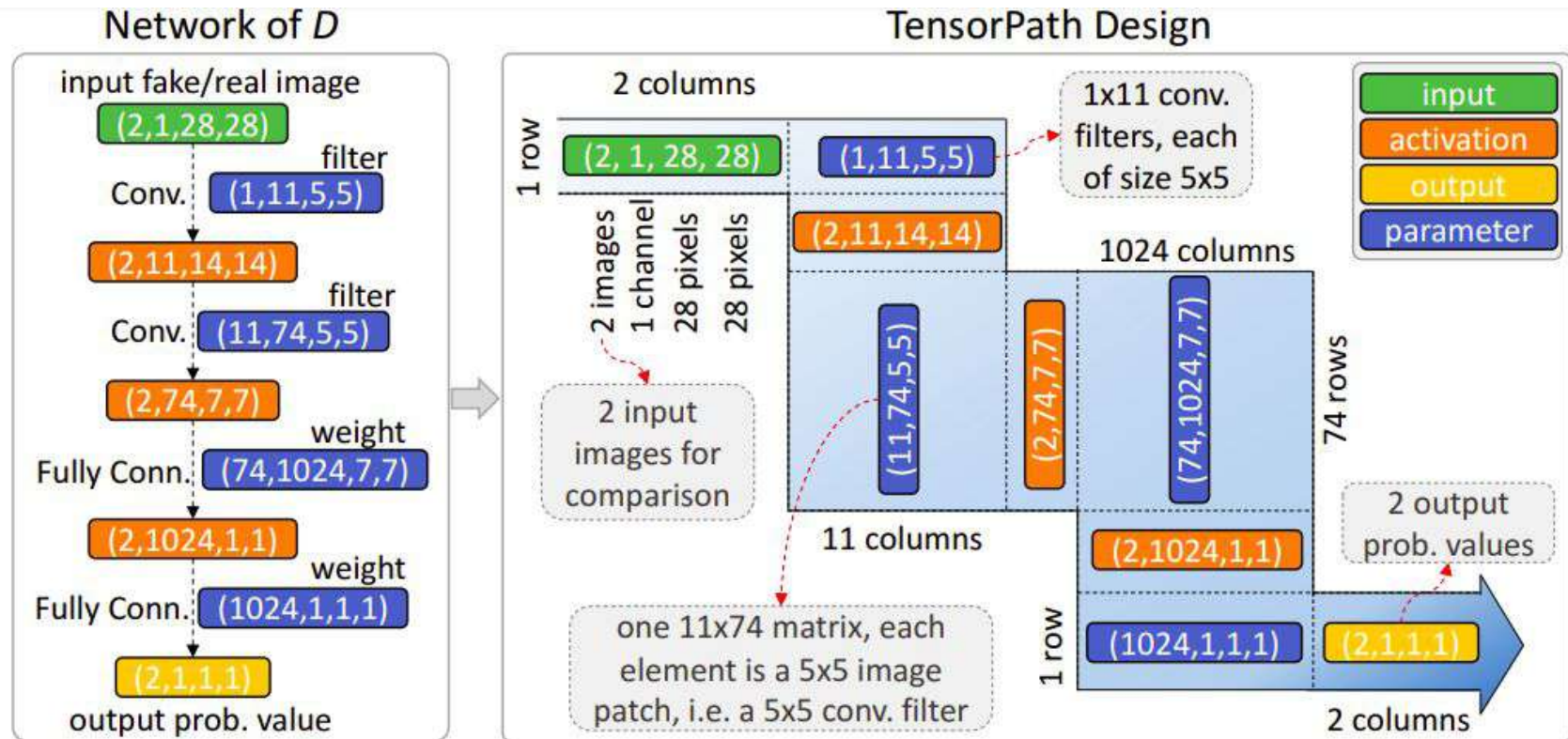




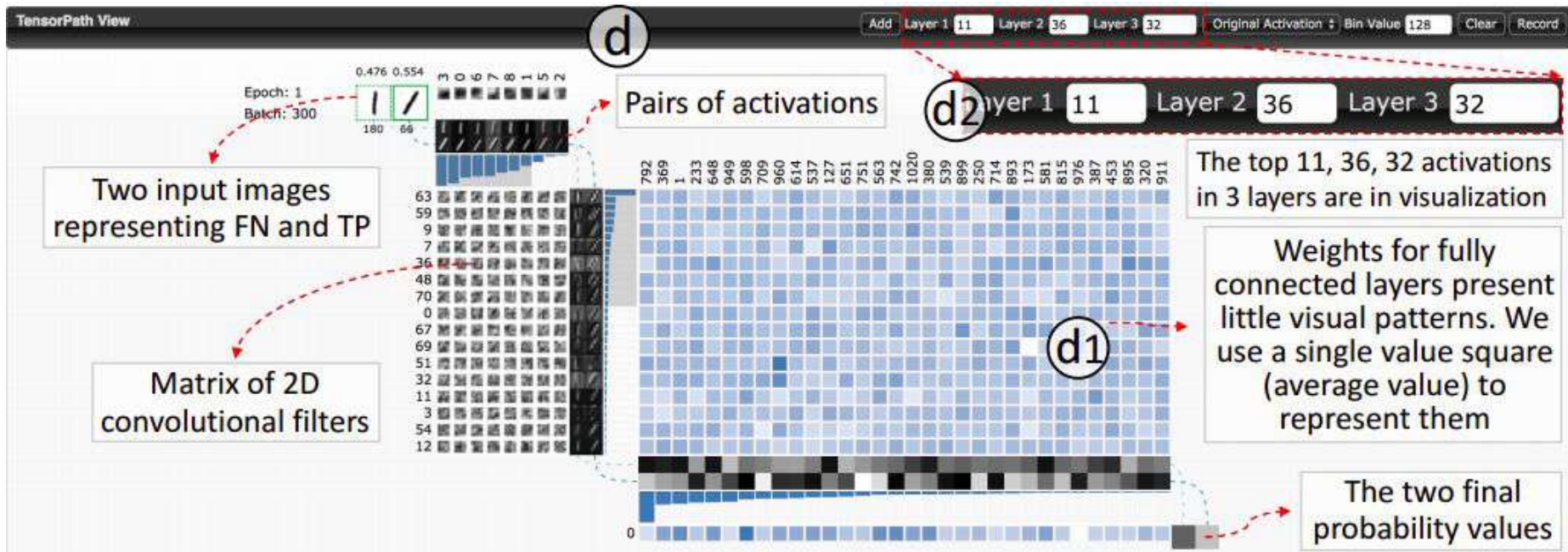
# GANViz: Distribution View



# GANViz: TensorPath View

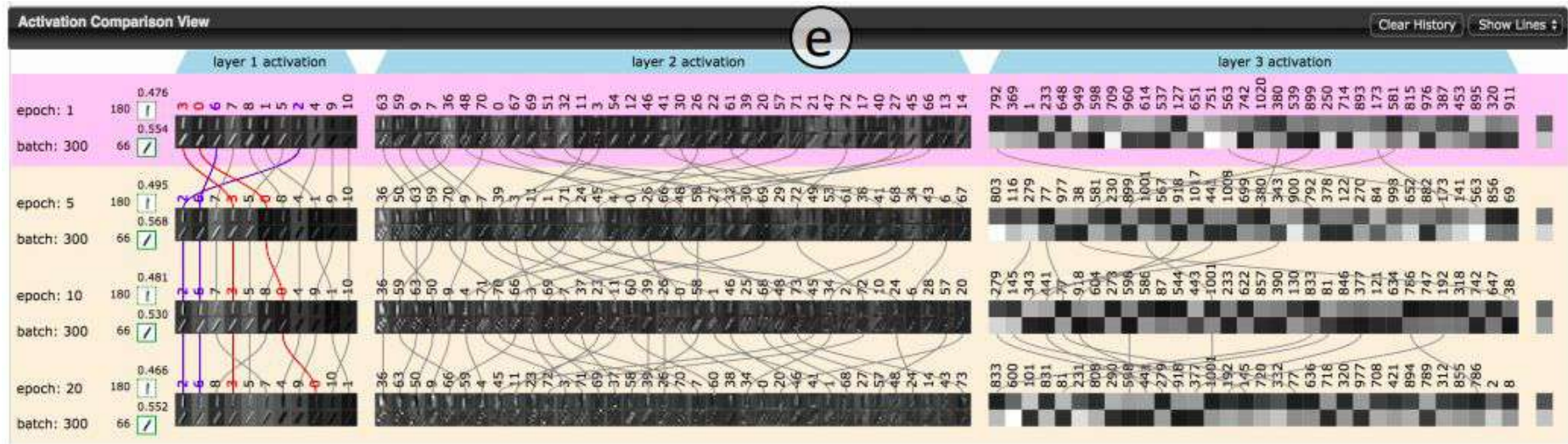


# GANViz: TensorPath View



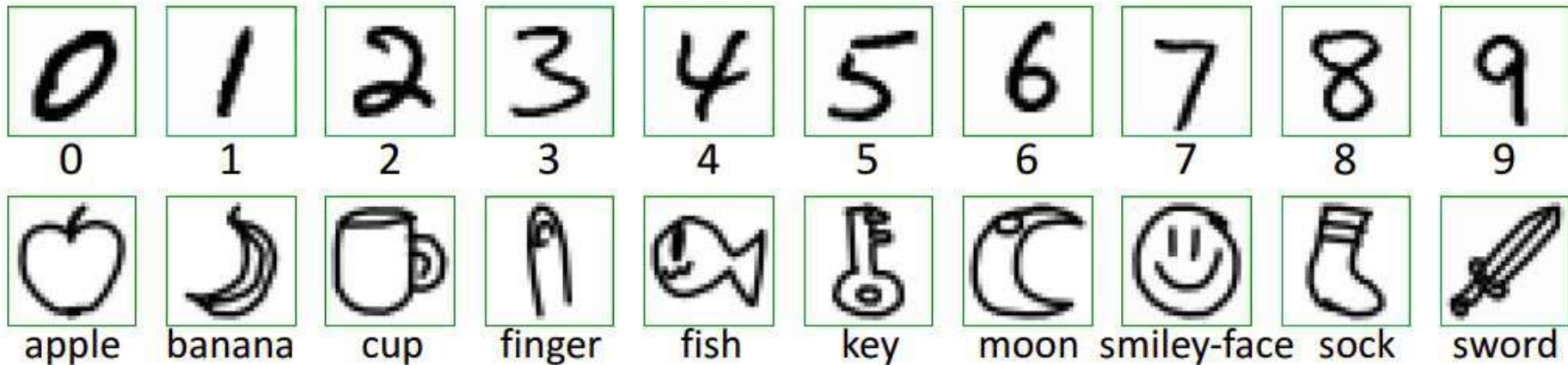


# GANViz: Activation Comparison View



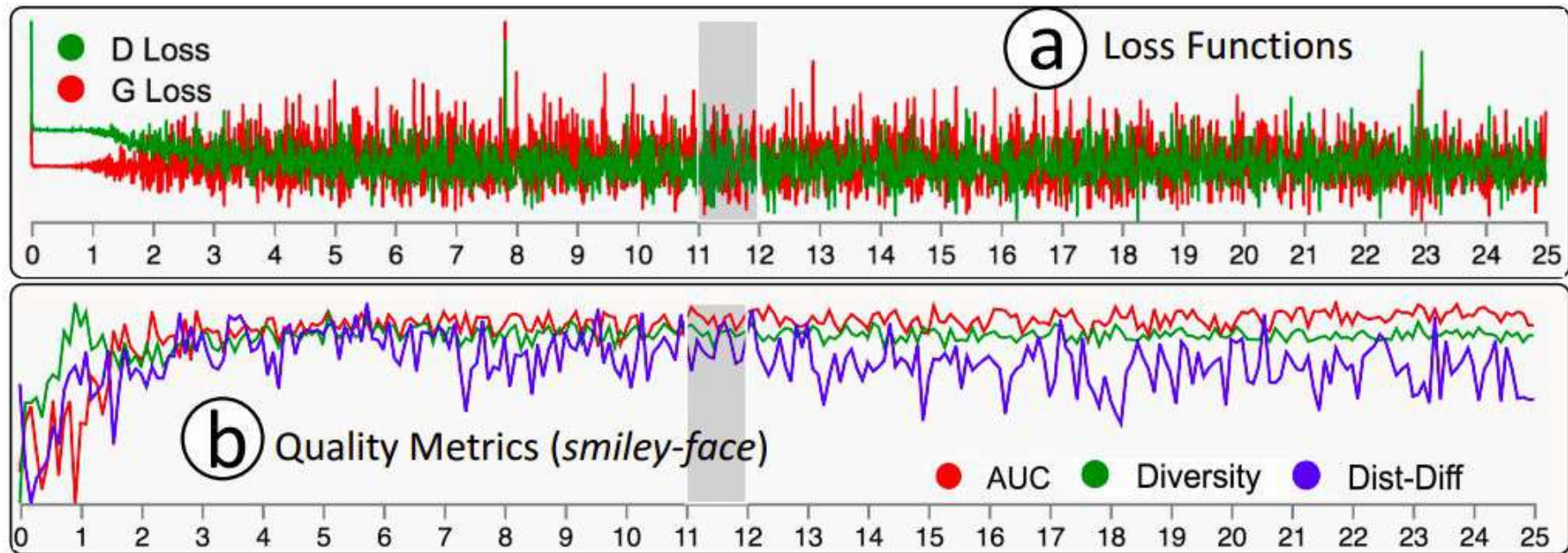
# GANViz: Case Studies

# GANViz: Case Studies



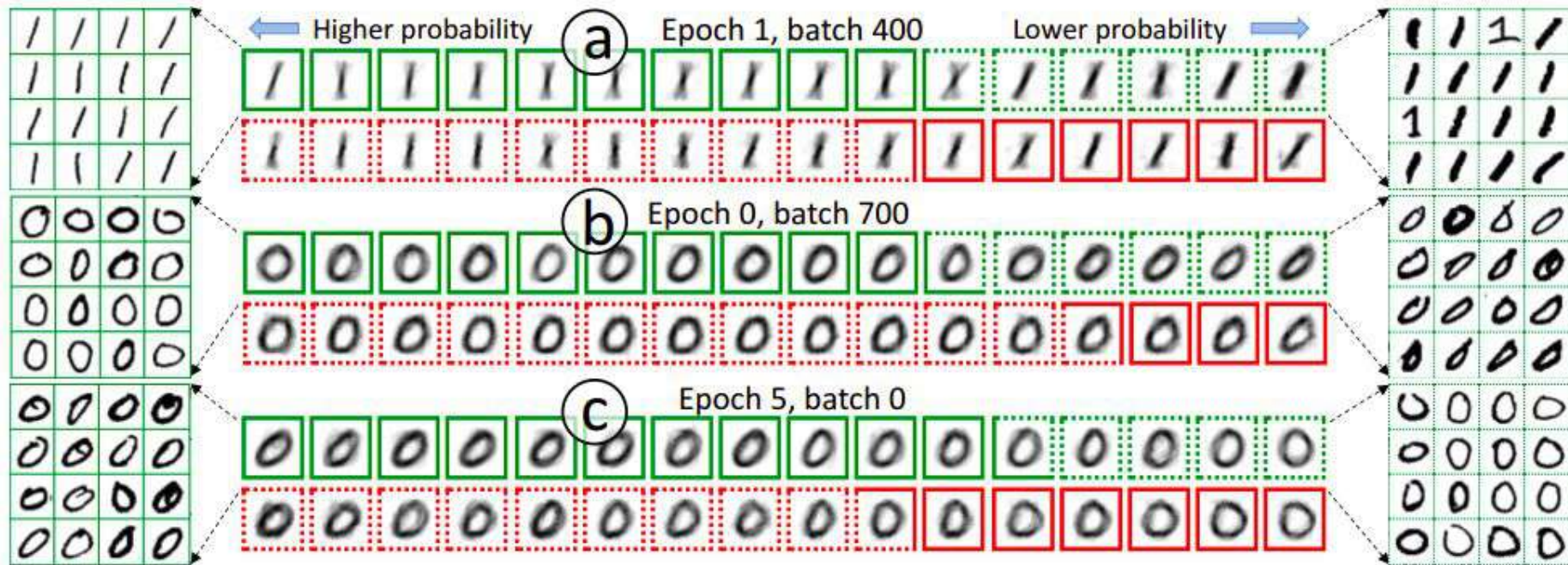
MNIST & QuickDraw databases

# GANViz: Model Evaluation



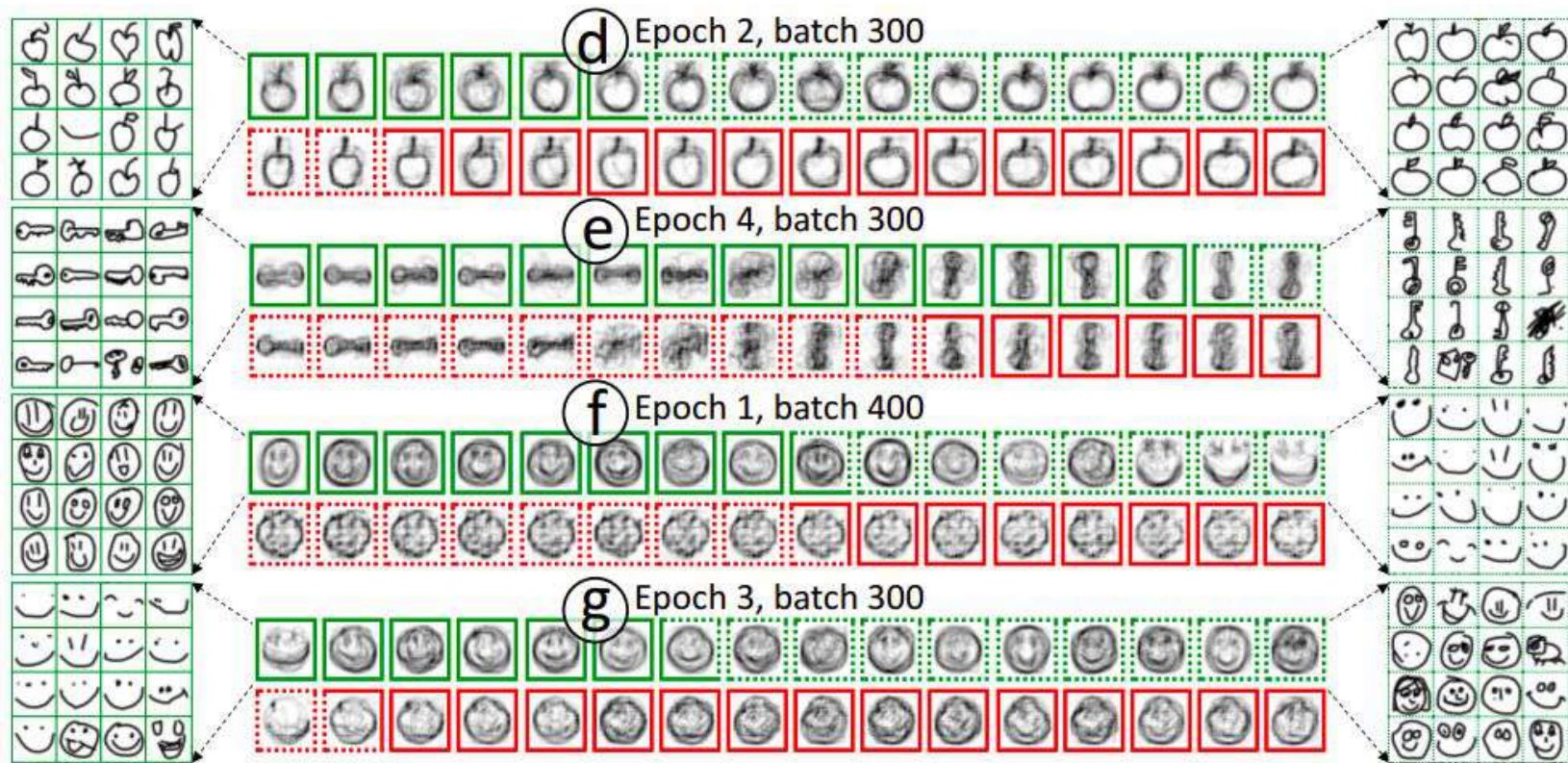


# GANViz: Model Interpretation

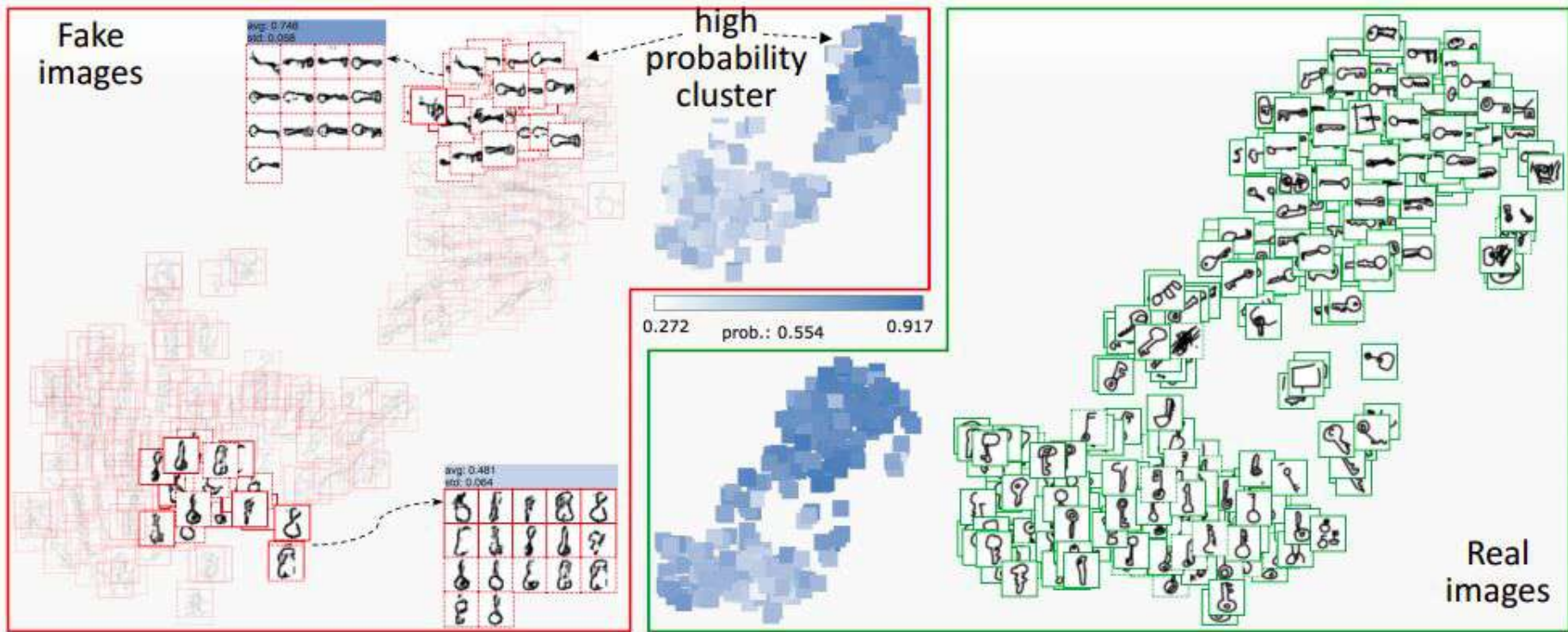




# GANViz: Model Interpretation

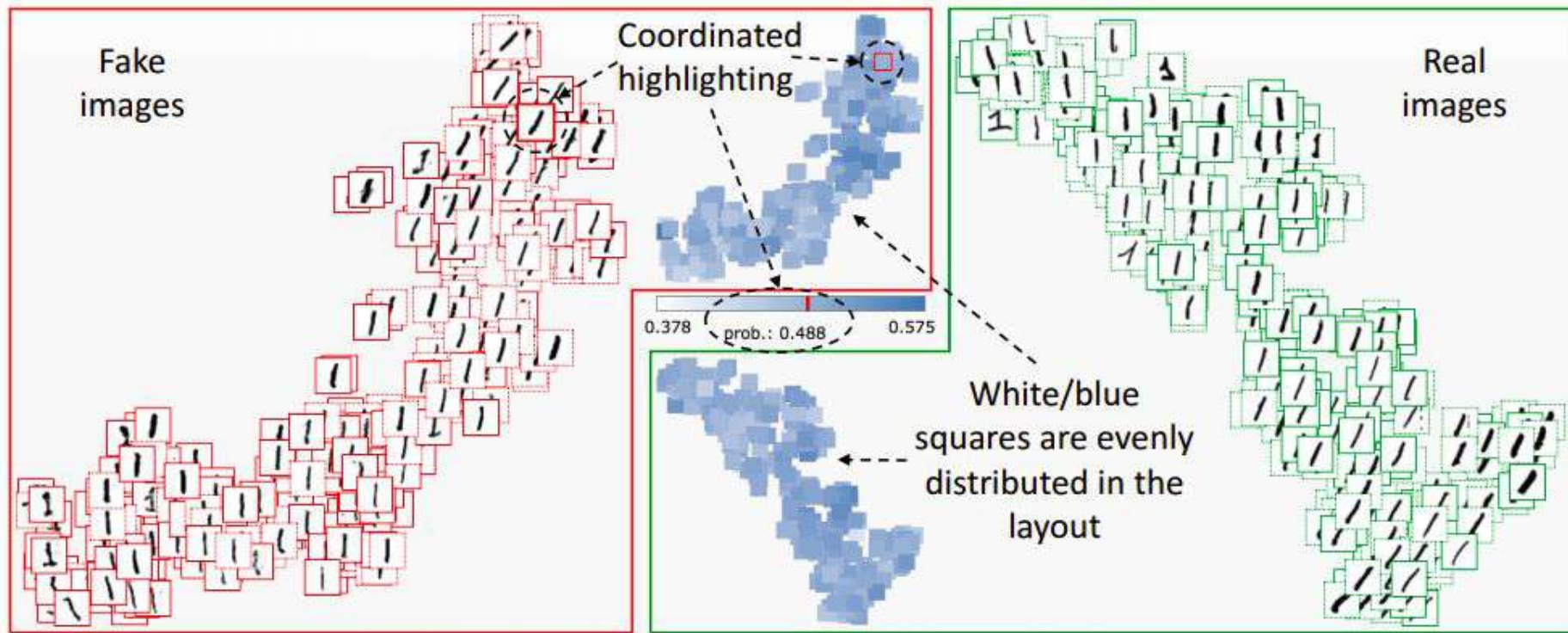


# GANViz: Model Interpretation

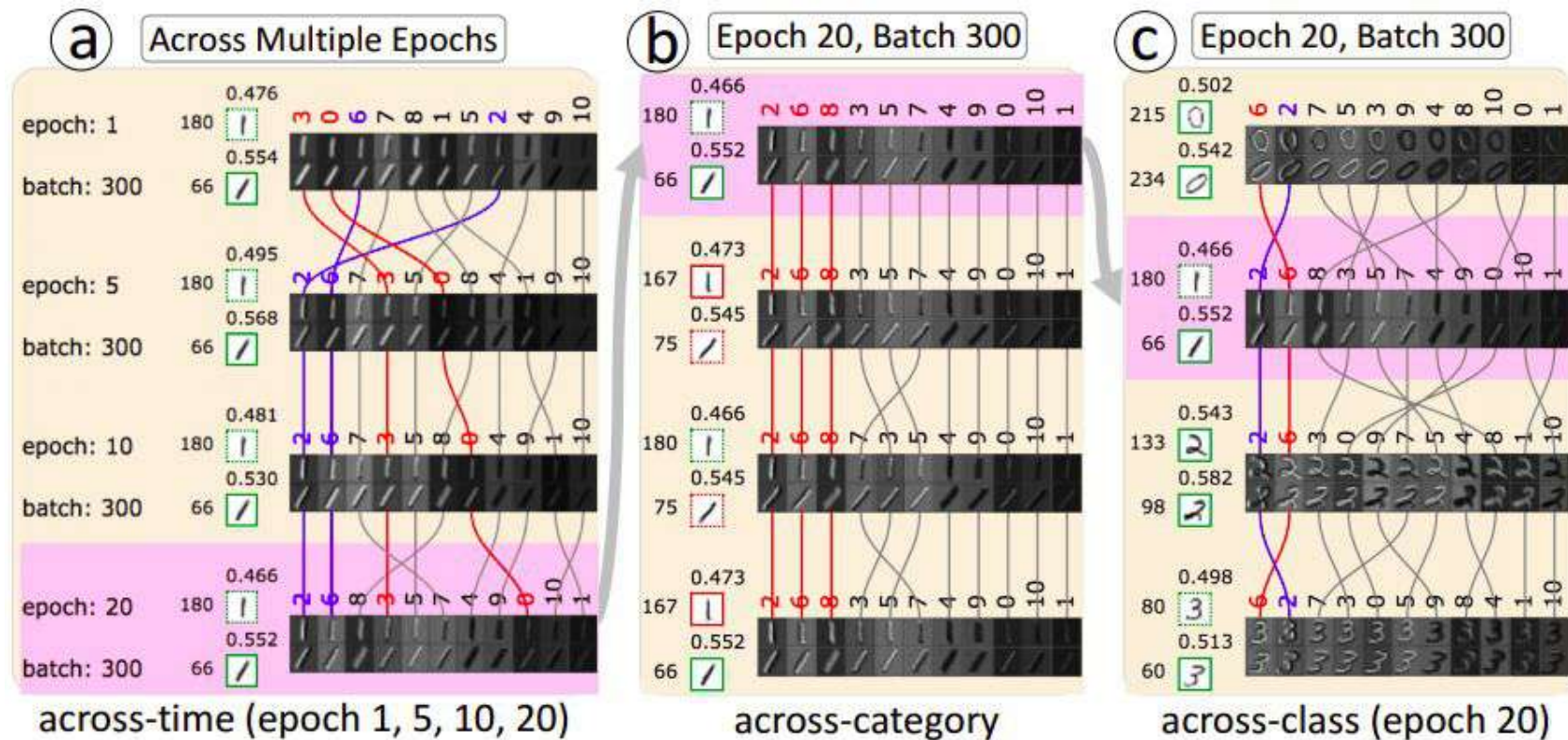




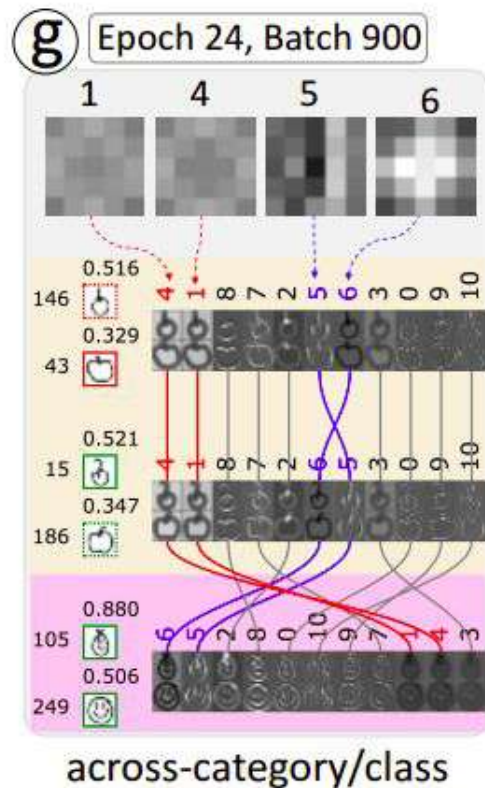
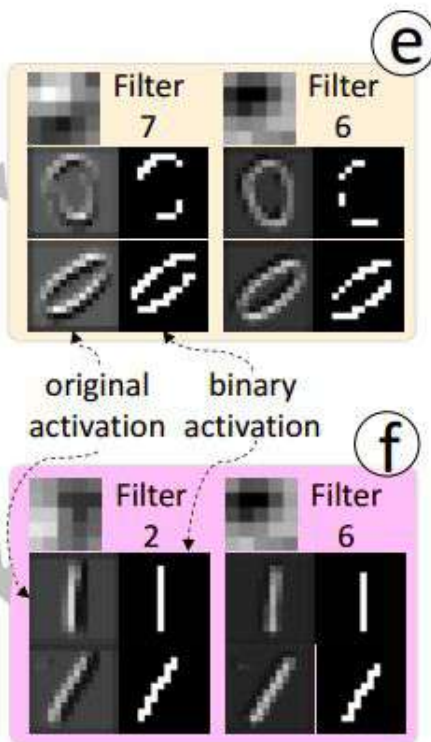
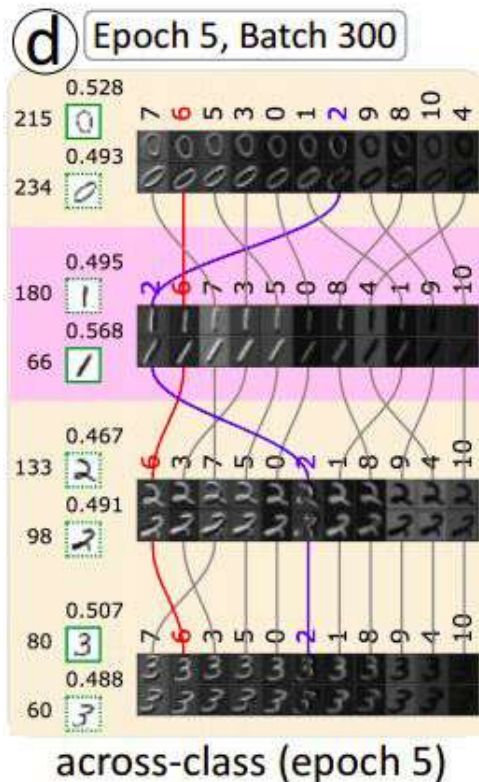
# GANViz: Model Interpretation



# GANViz: Comparative Analysis



# GANViz: Comparative Analysis



# GANViz: Limitations

- Loss function doesn't reflect sufficient details about the model quality.
- Selecting of representative images: This selection depending of the person who selects.
- Low size images.

Thanks