Patchy

Anomaly detection with Generative Adversarial Networks and text patches

Andriy Drozdyuk & Norbert Eke

Romeo and Juliet

by William Shakespeare

O Romeo, Romeo! wherefore art thou Romeo?
Deny thy father and refuse thy name
Or, if thou wilt not, be but sworn my love,
A sudden blow: the great wings beating still
And I’ll no longer be a Capulet

Romeo and Juliet
by William Shakespeare

O Romeo, Romeo! wherefore art thou Romeo?
Deny thy father and refuse thy name
Or, if thou wilt not, be but sworn my love,
A sudden blow: the great wings beating still
And I'll no longer be a Capulet

Leda and the Swan
W. B. *Yeats*

Outline

1. Problem Introduction
    a. Anomaly Detection Task
2. Related Work
    a. Generative Adversarial Networks (GAN)
        i. Description
    b. Unsupervised Anomaly detection (anoGAN)
        i. Description
        ii. Image Patches
    c. FakeGAN
    d. Use of Word Embeddings
    e. Other related work
3. Proposed work
    a. Anomaly detection as a task of text classification using GANs
    b. AnoGAN based approach using Text patches
4. Experimental design
    a. Text classification approach
    b. AnoGAN approach

1. Problem

Anomaly Detection Task

What is it ?

- Identification of out-of-ordinary/ unusual/ unexpected data points
  - E.g.: Outlier detection, fraud detection, malicious intent detection

What are some forms of anomaly detection in NLP?

- Text containing malicious intent:
  - offensive language, hate speech, cyber-bullying, sexual predatory behavior
- Text containing suicidal or depressive behavior

Where can such textual data be found?

- Online chat-room, forums
- Social networking platforms

Anomaly Detection Task

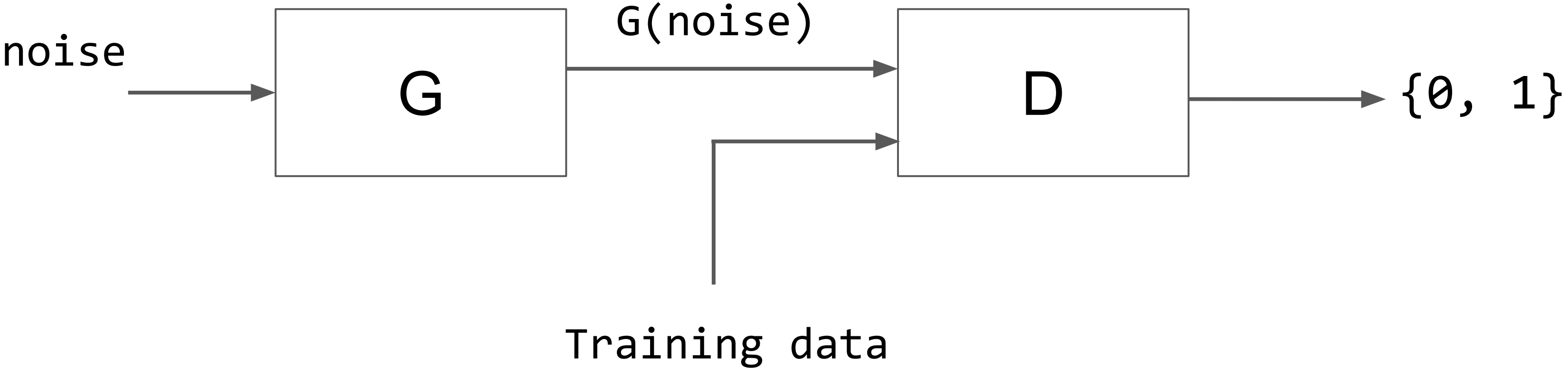
What are the biggest challenges?

- Lack of labelled textual data
  - Normal vs Anomalous
- Lack of negative examples (very unbalanced)
  - Usually only < 10% of the sample size is anomalous
- Such textual data is unusually messy
  - Contains slang, internet acronyms, misspelled words

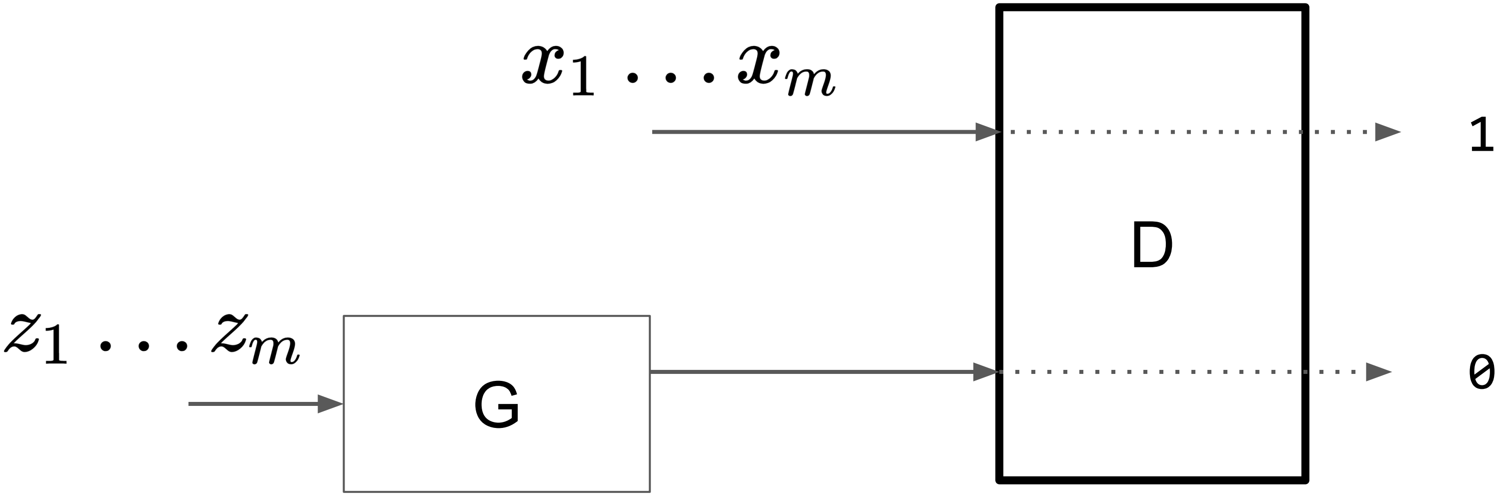
2. Related work

Generative Adversarial Networks (GAN)

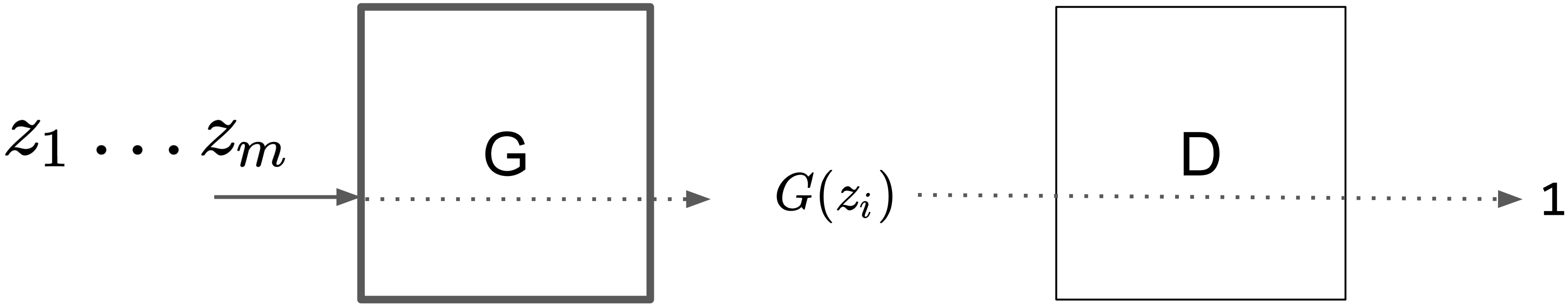
Generative Adversarial Networks, Ian J. Goodfellow et. al.
2014

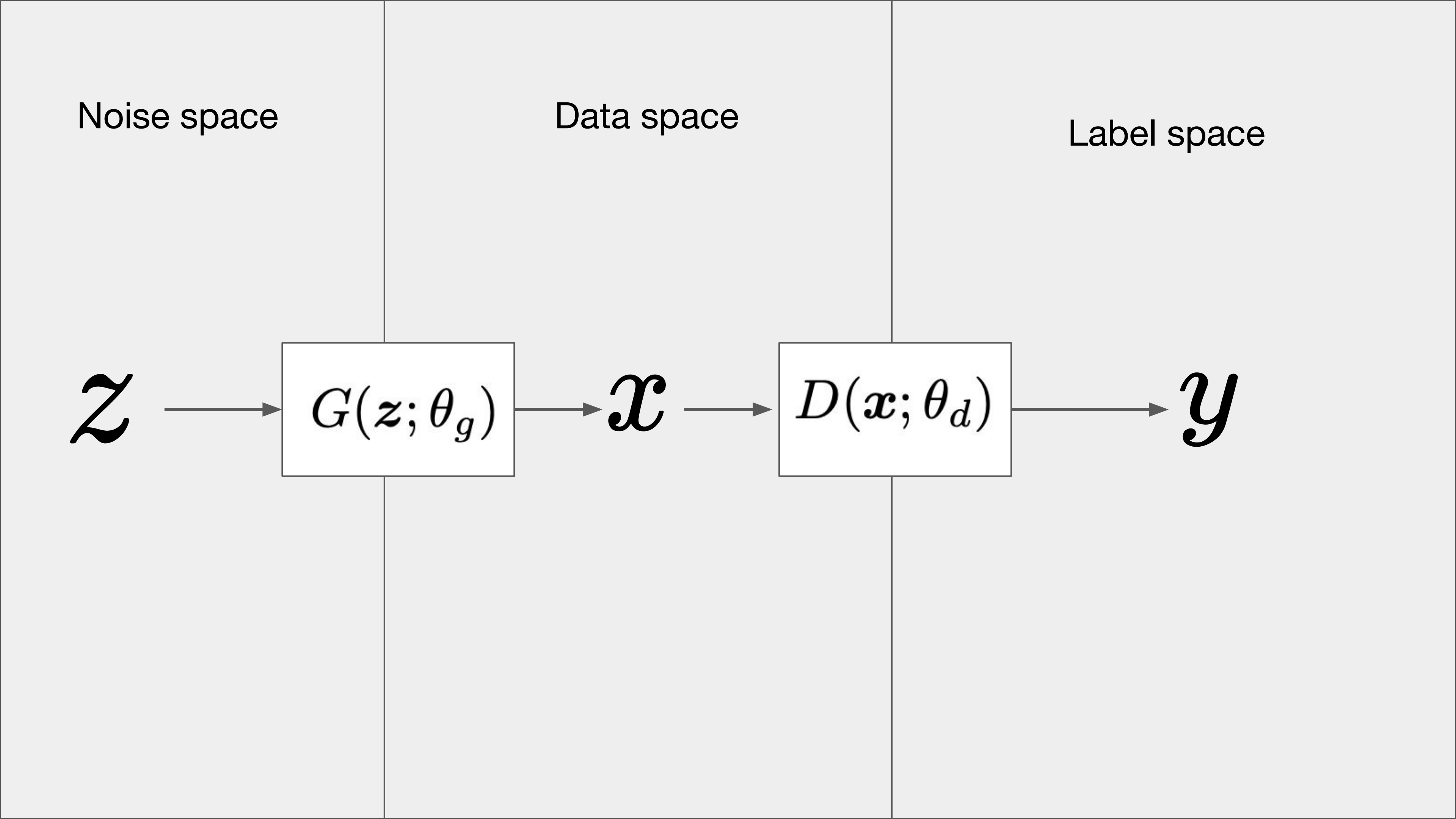

noise
G
G(noise)
D
{0, 1}
Training data

Train D



Train G





Noise space

Data space

Label space

$\boldsymbol{z}$

$G(\boldsymbol{z}; \theta_g)$

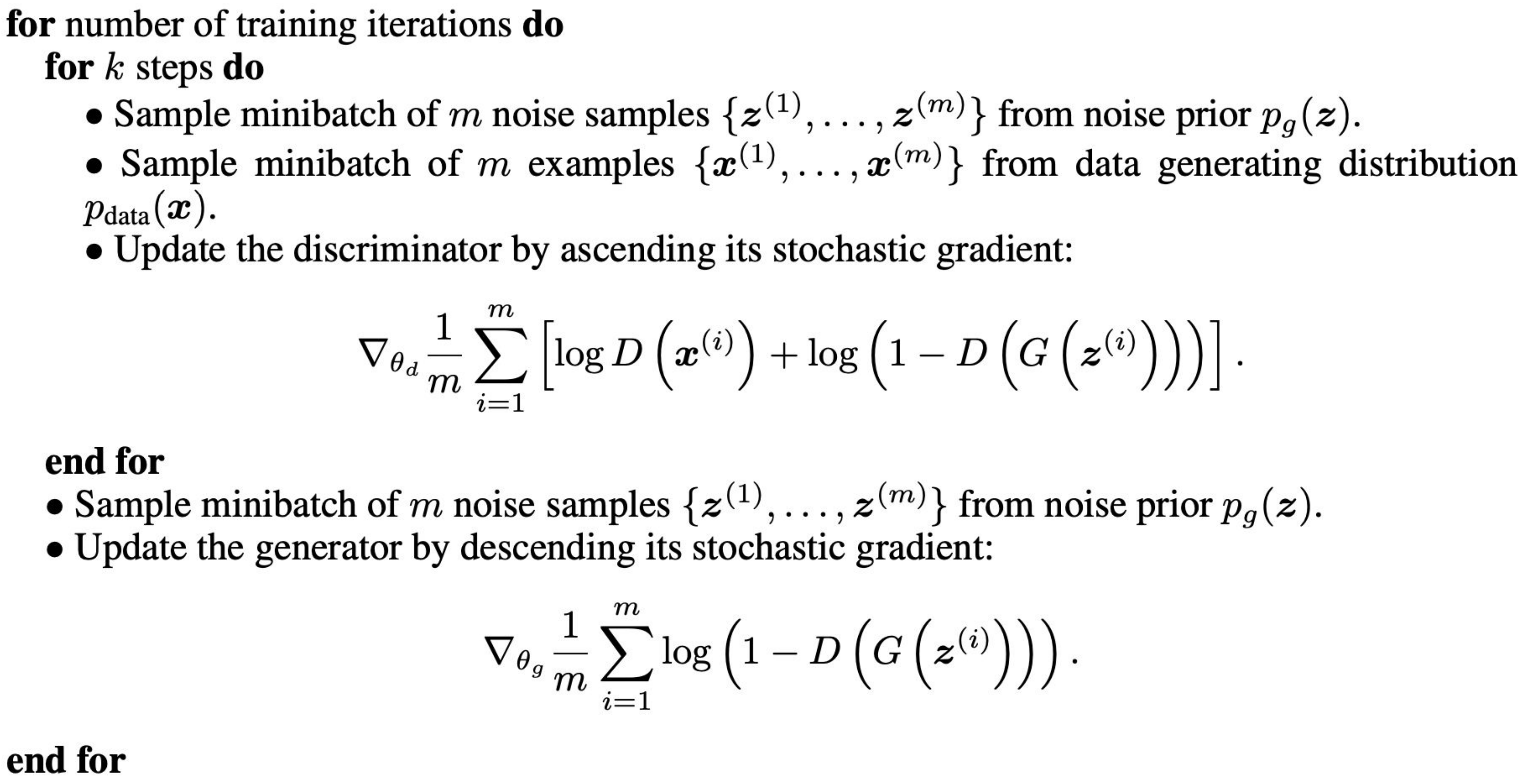
$\boldsymbol{x}$

$D(\boldsymbol{x}; \theta_d)$

$\boldsymbol{y}$

Minimax game value function

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



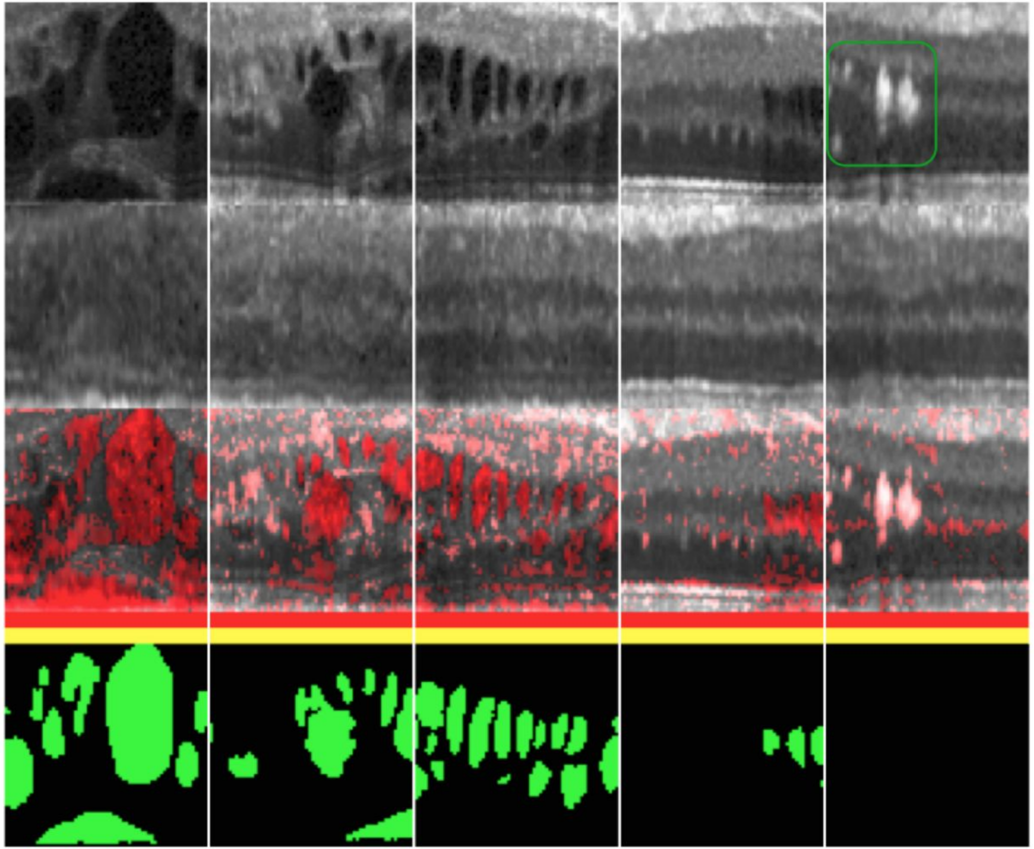
```
for number of training iterations do
    for $k$ steps do
        • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
        • Sample minibatch of $m$ examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
        • Update the discriminator by ascending its stochastic gradient:
            $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$
    end for
    • Sample minibatch of $m$ noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
    • Update the generator by descending its stochastic gradient:
        $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$
end for
```

Unsupervised Anomaly Detection

Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery, 2017, Schlegl et. al.

Anomaly Detection

Identification of unusual observations in the data.



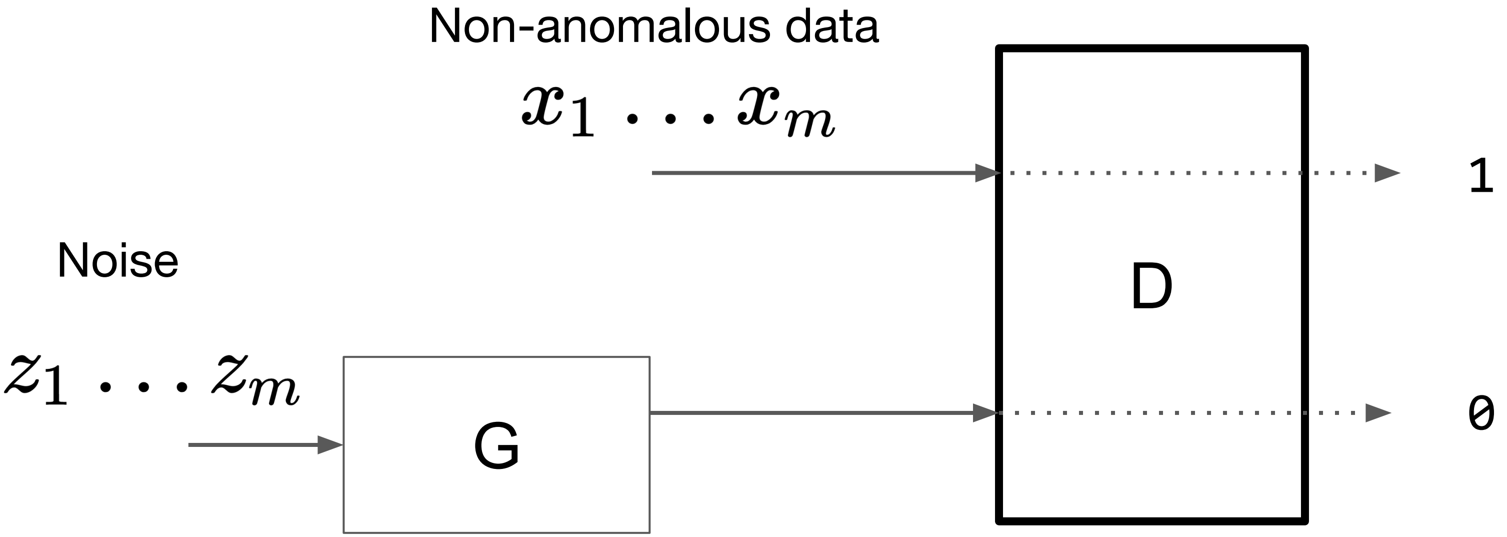
Two phases:

1. Training
2. Anomaly detection

Training

We train GAN as before, on normal (non-anomalous) data only.

Train D



Non-anomalous data

$x_1 \ldots x_m$

Noise

$z_1 \ldots z_m$

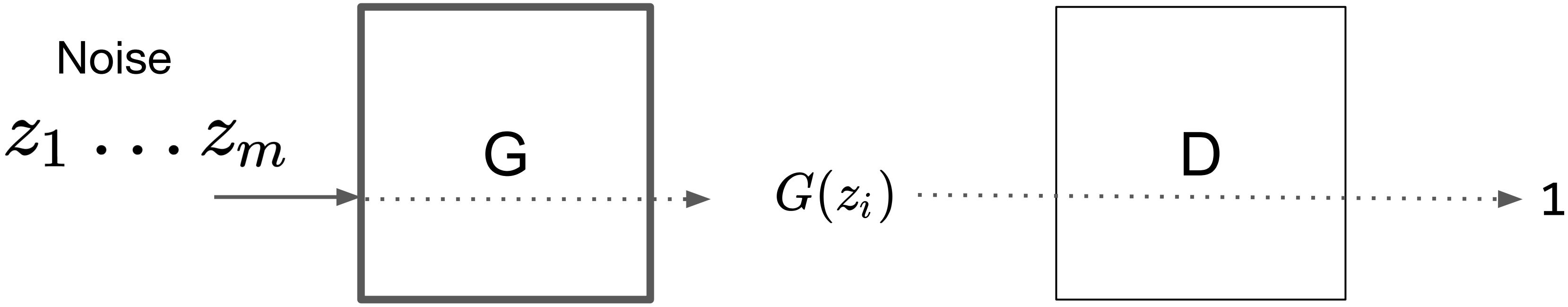
G

D

1

0

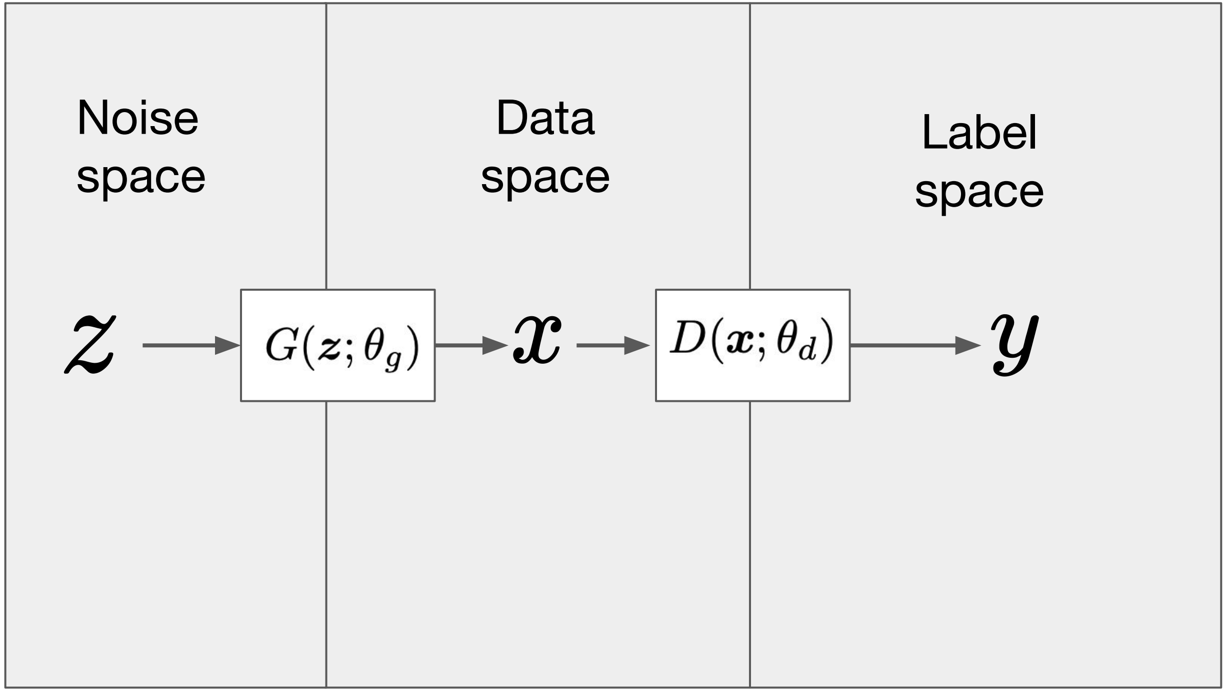
Train G



Anomaly detection

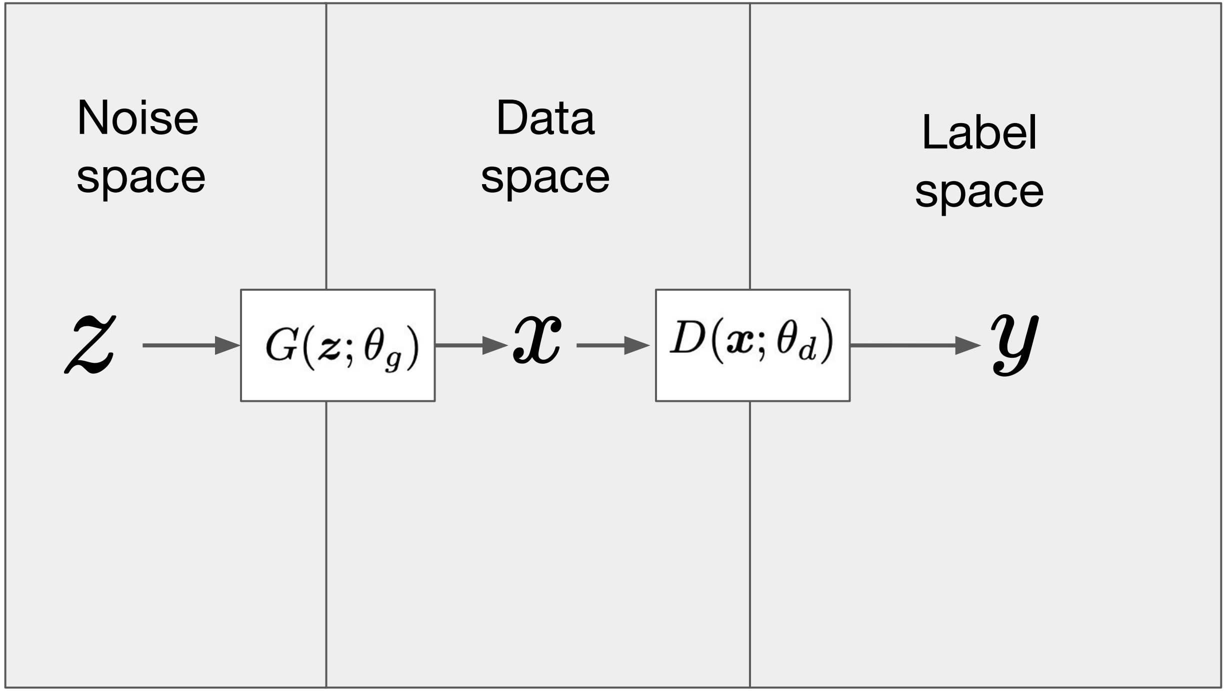
We are given some data and we use our trained GAN to determine if it is anomalous.

Problem: given a query x is it anomalous?



Solution

1. Pick a random z
2. Calculate loss.
3. Backpropagate to update z



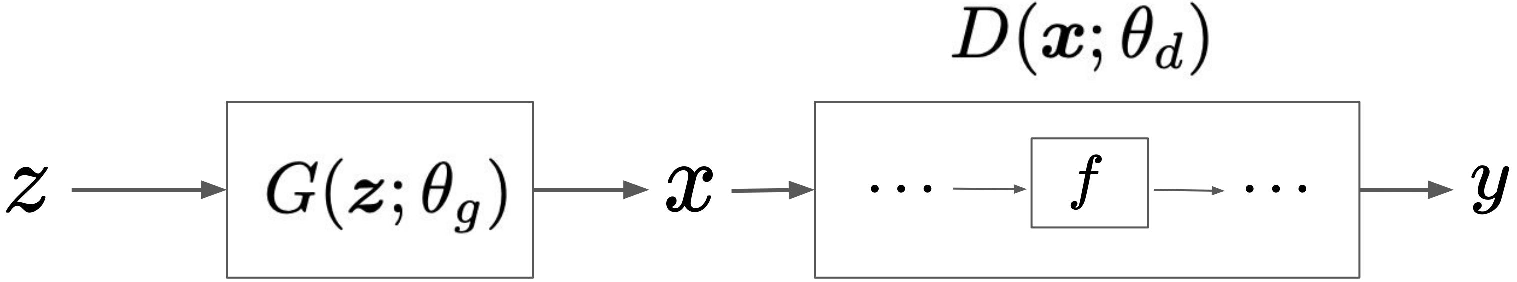
Example

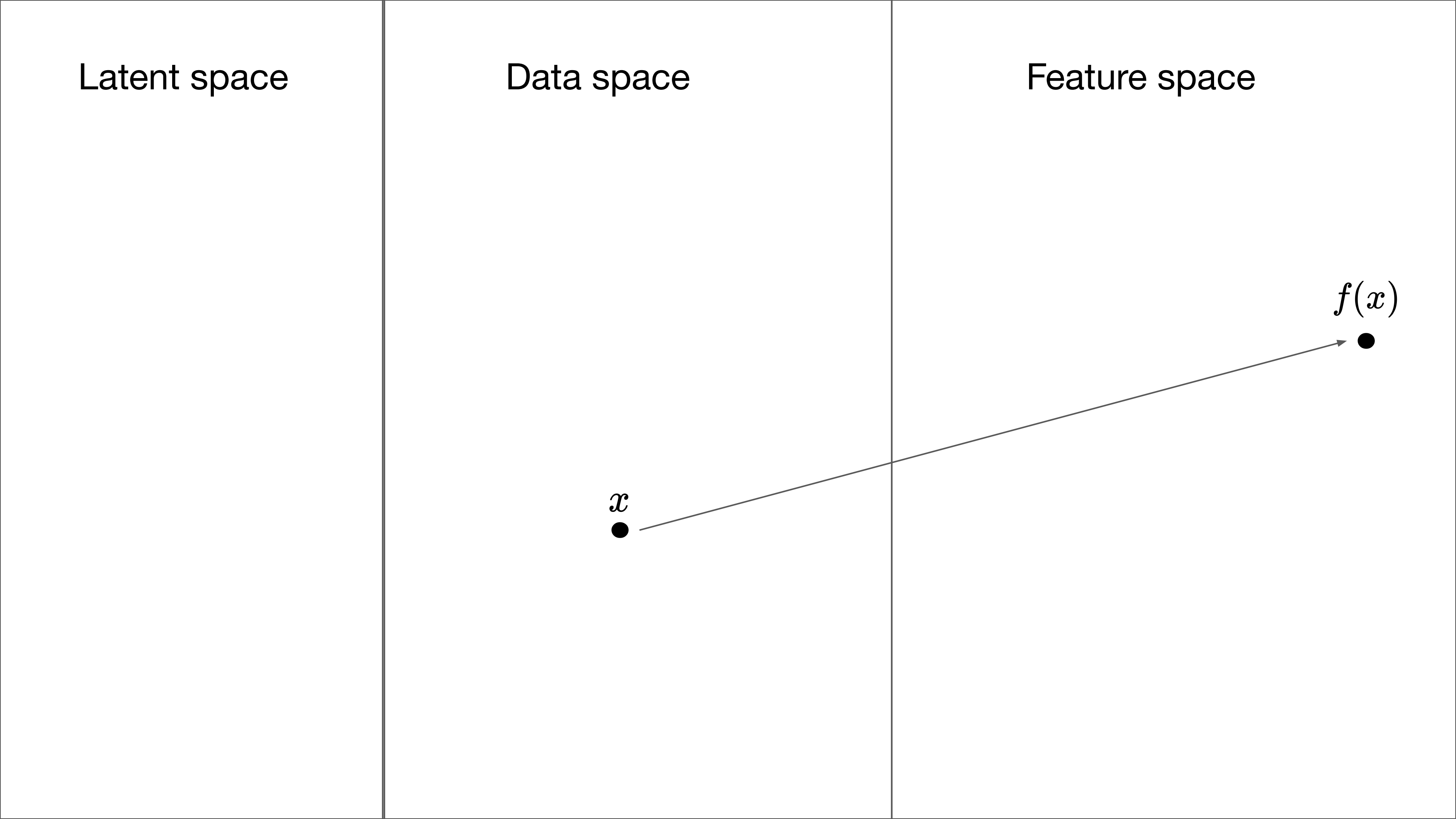
Latent space

Data space

Feature space

Feature representation

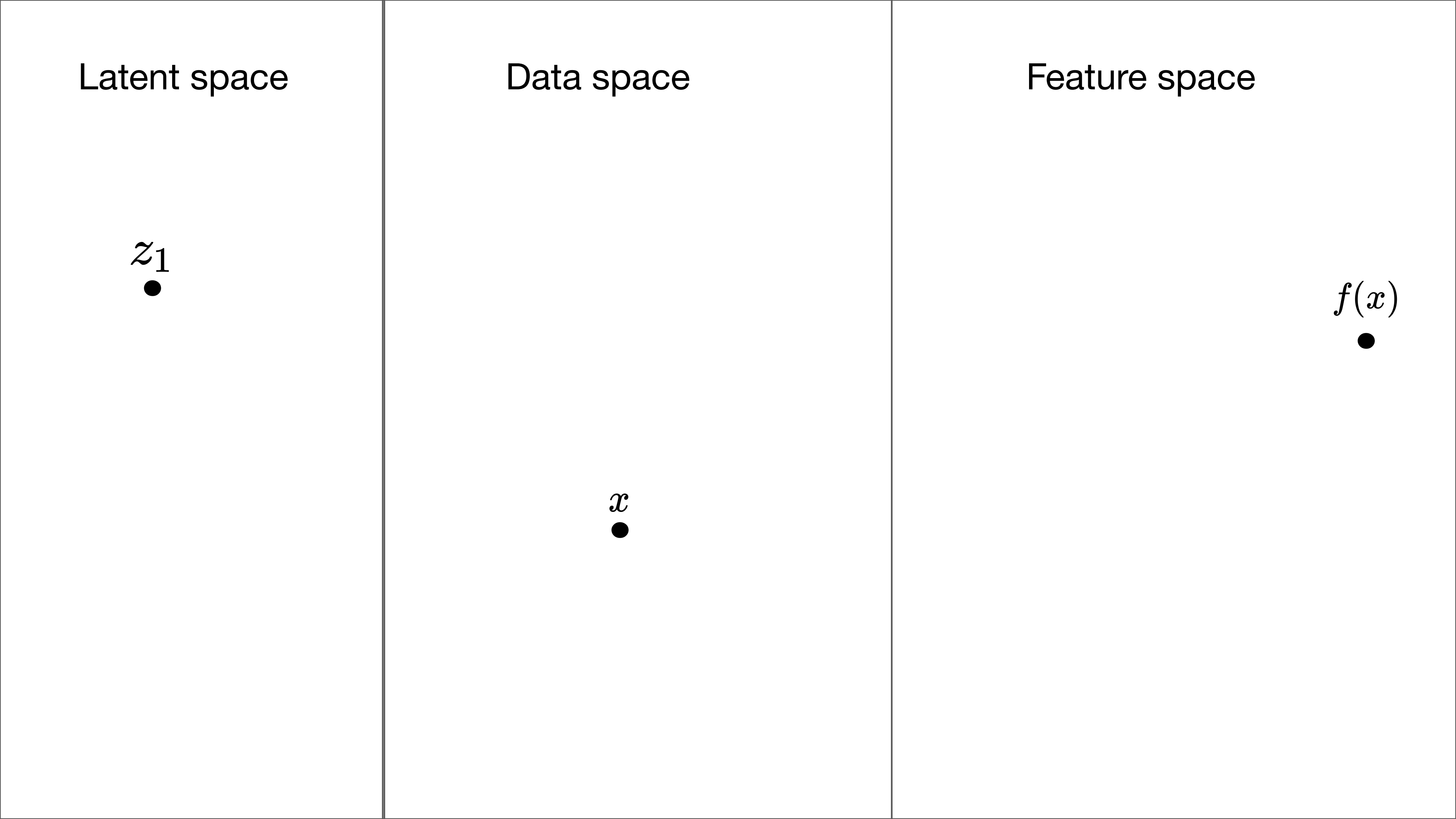




Latent space

Data space

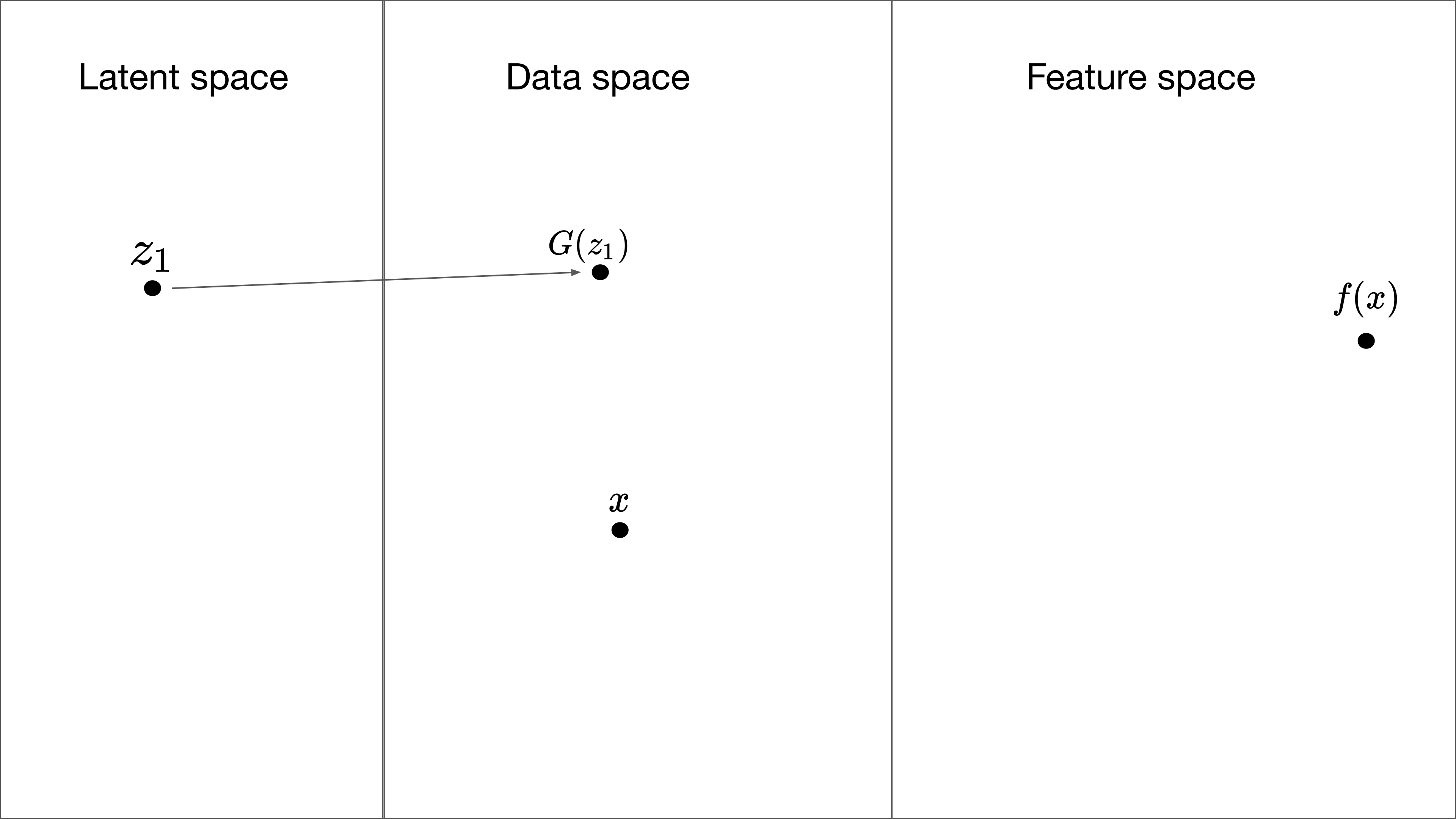
Feature space



Latent space

Data space

Feature space



Latent space

Data space

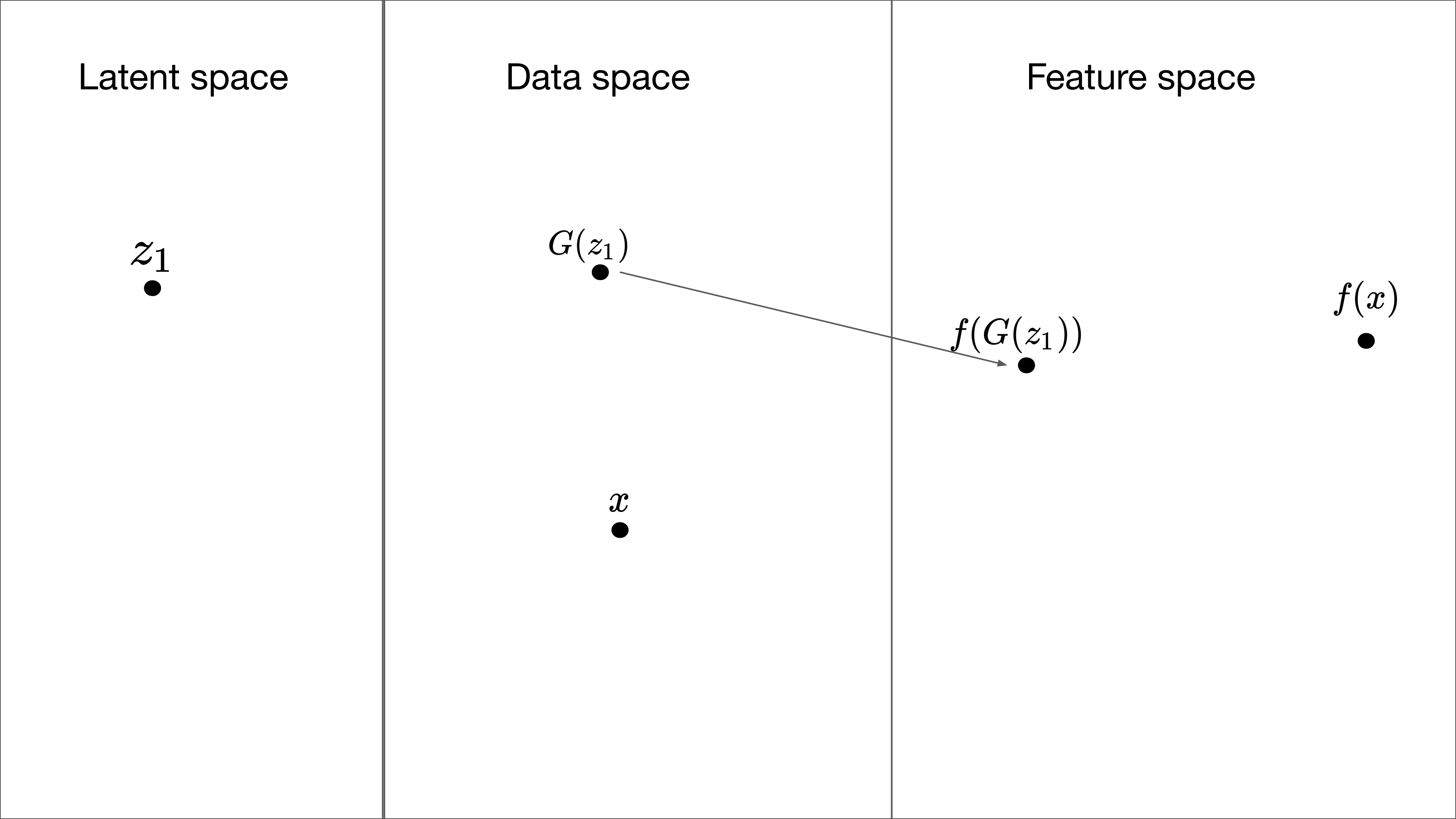
Feature space

z_1

$G(z_1)$

x

$f(x)$



Latent space

Data space

Feature space

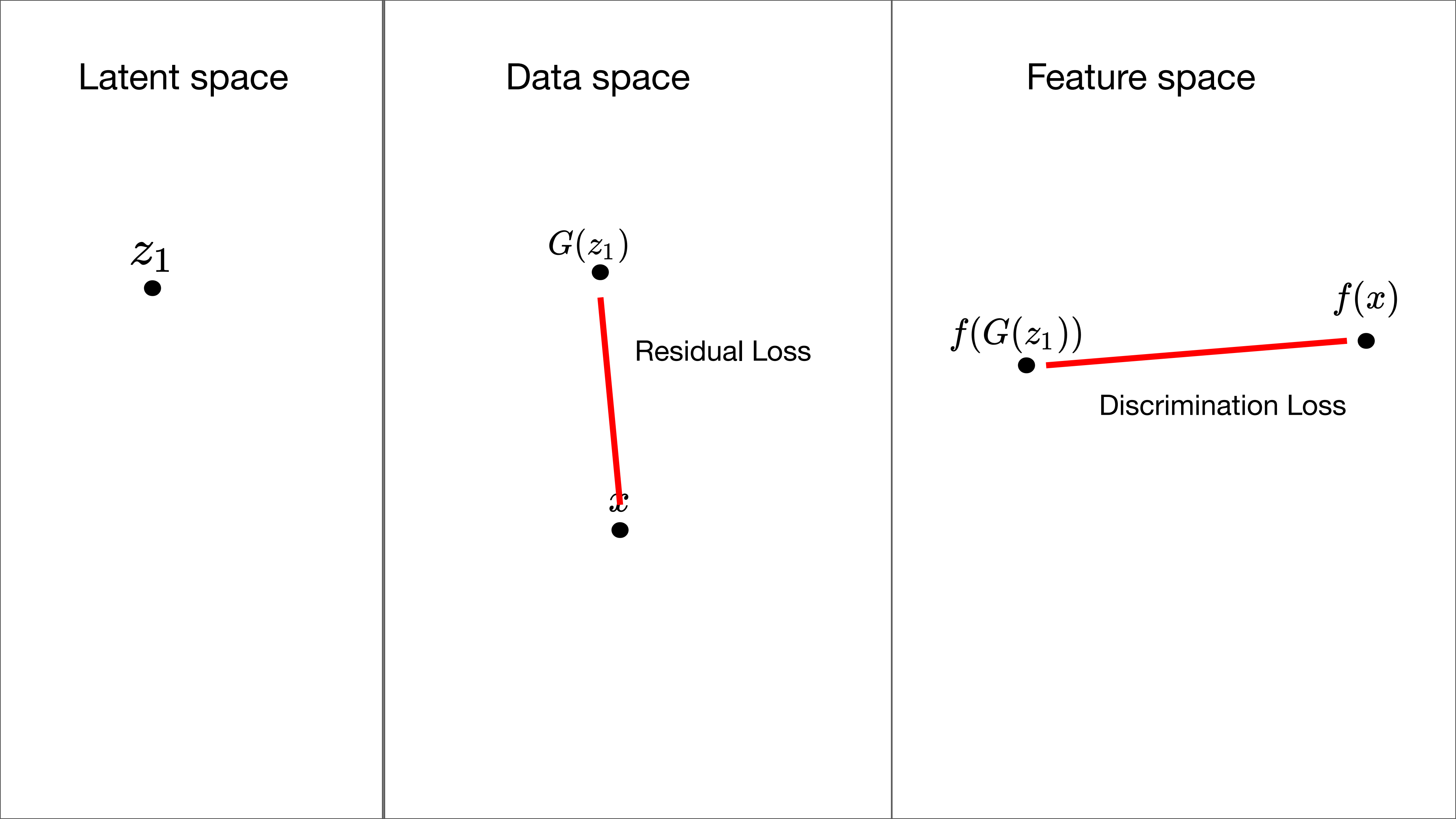
z_1

$G(z_1)$

x

$f(G(z_1))$

$f(x)$



Latent space

z_1

Data space

$G(z_1)$

Residual Loss

x

Feature space

$f(x)$

$f(G(z_1))$

Discrimination Loss

Residual loss

Measures dissimilarity between query image `x` and generated image `G(z)`

$$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum |\mathbf{x} - G(\mathbf{z}_\gamma)|.$$

Discrimination loss

Measures the dissimilarity in features extracted by the discriminator.

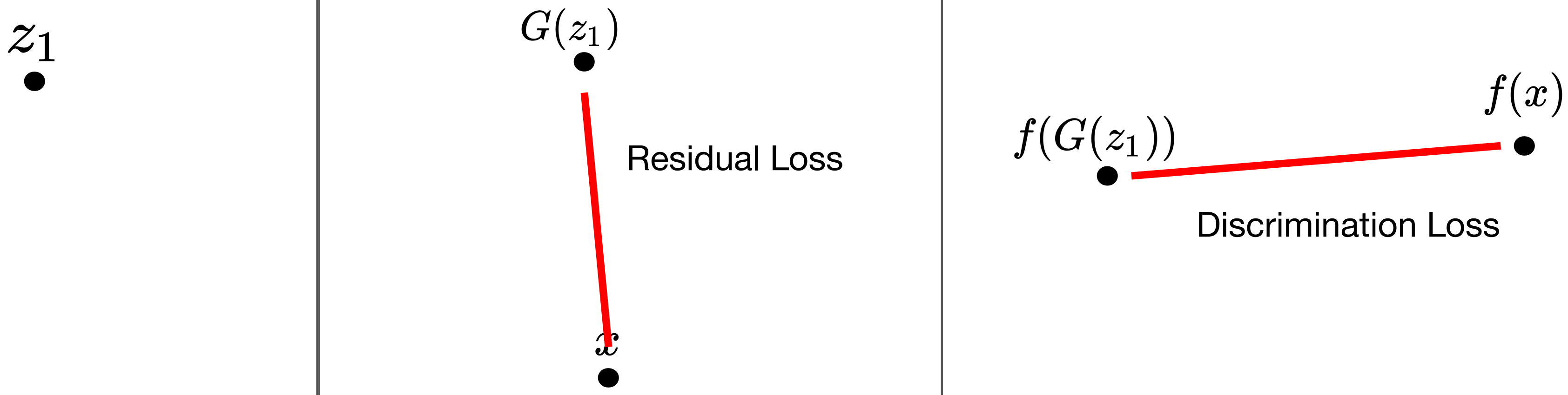
$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum |\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|,$$

where `f(.)` is the output of the intermediate layer in the discriminator.

Latent space

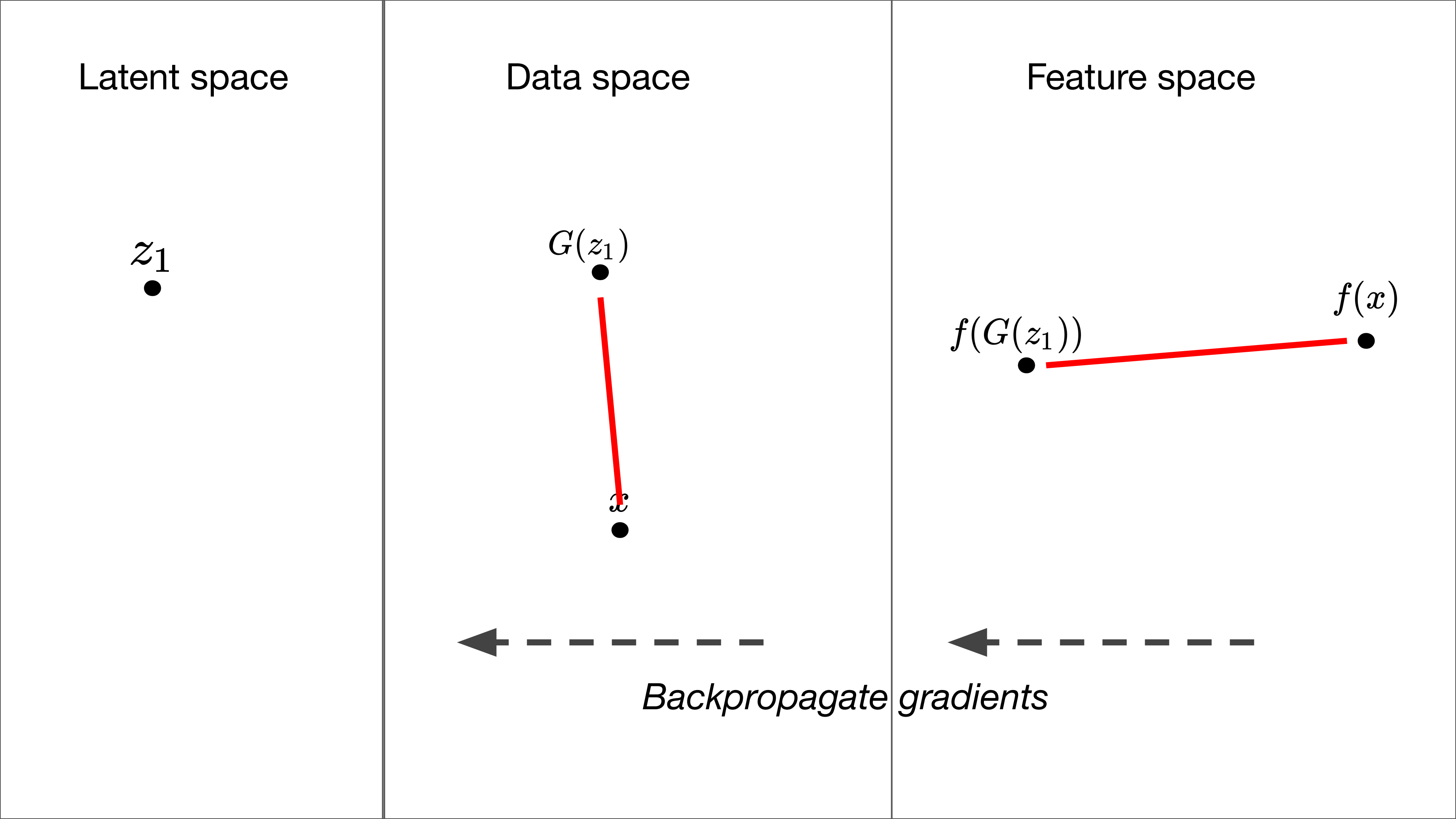
Data space

Feature space



$$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum |\mathbf{x} - G(\mathbf{z}_\gamma)|.$$

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum |\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|,$$

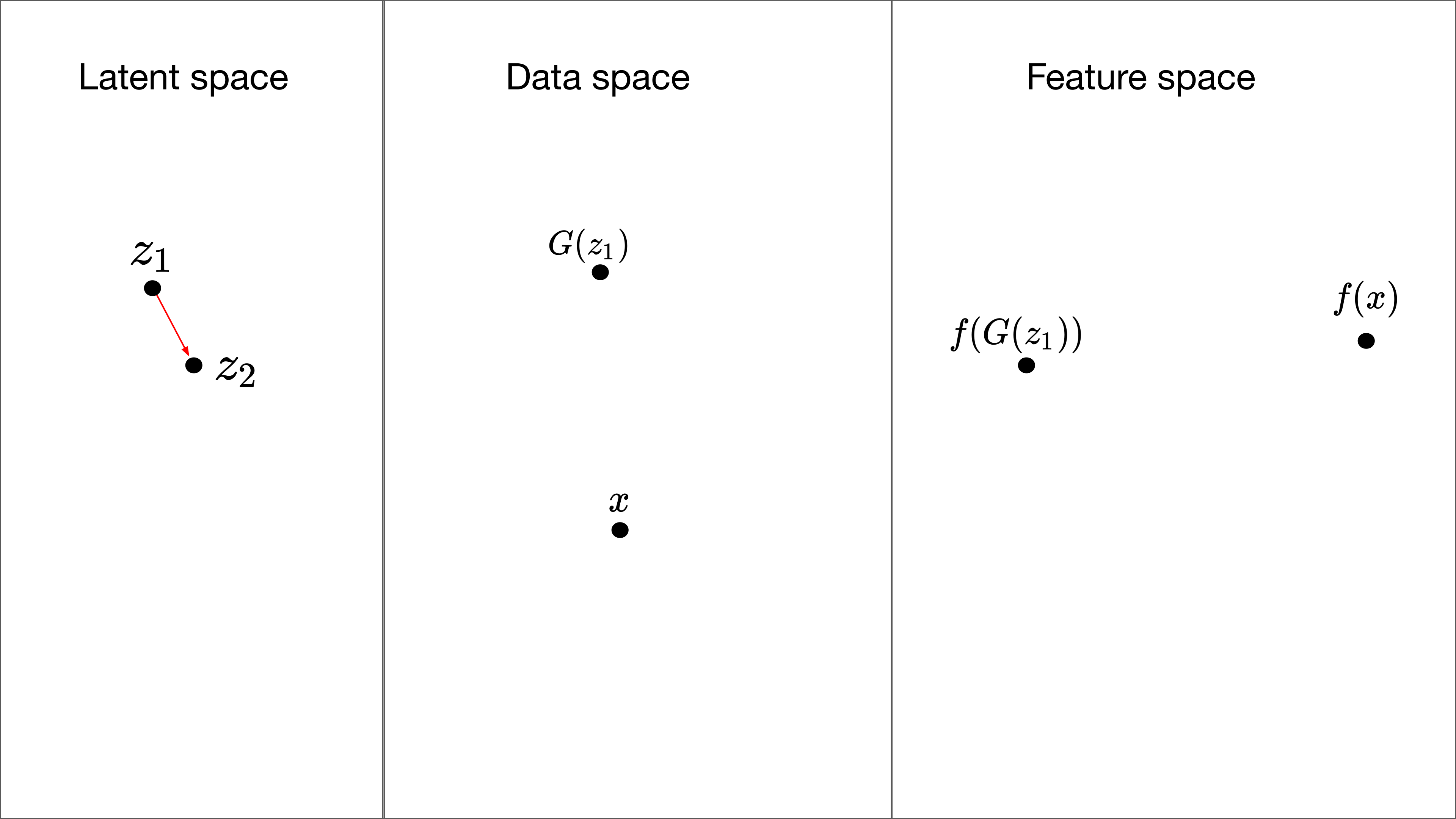


Latent space

Data space

Feature space

Backpropagate gradients



Latent space

z_1

z_2

Data space

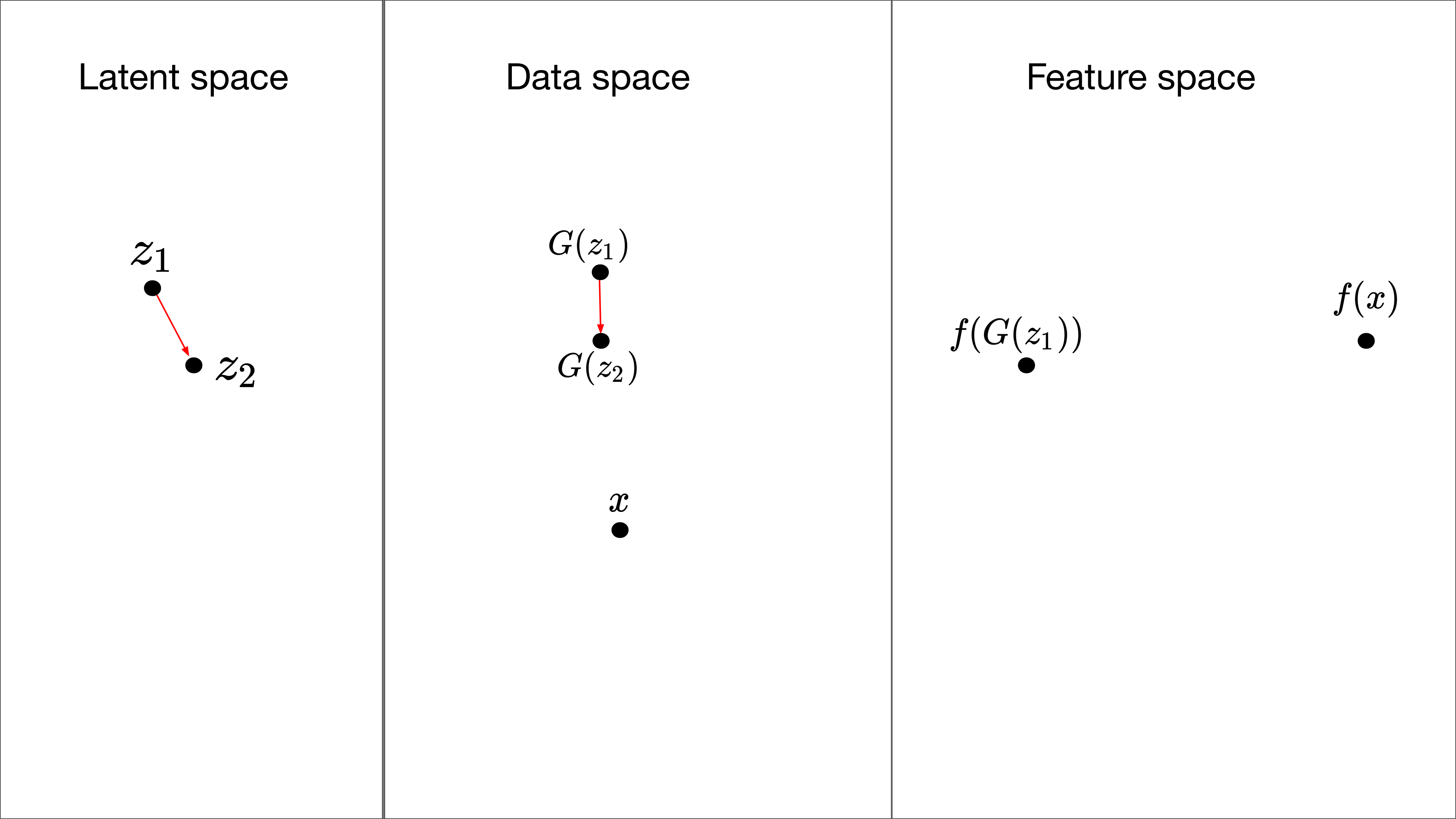
$G(z_1)$

x

Feature space

$f(G(z_1))$

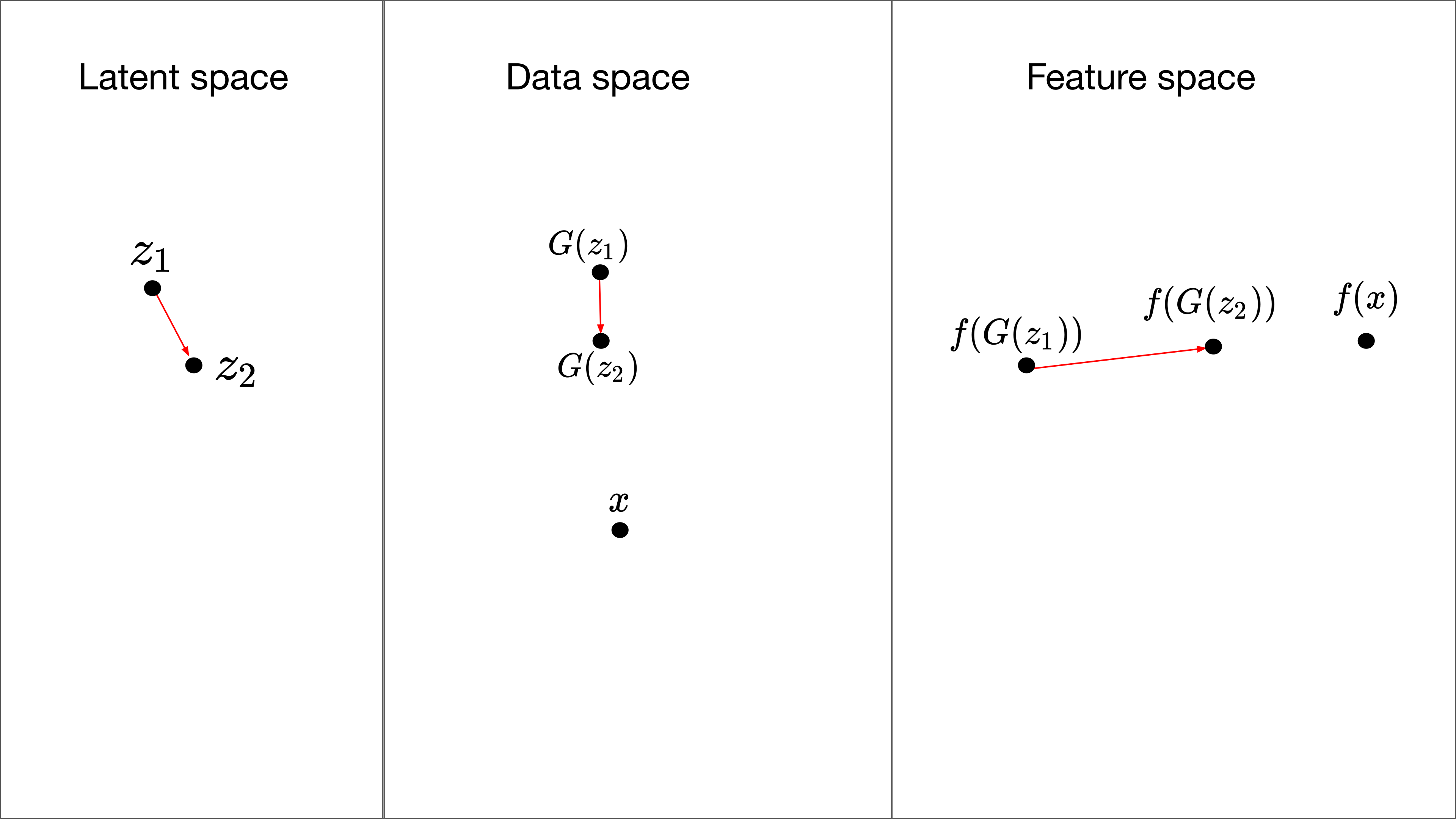
$f(x)$



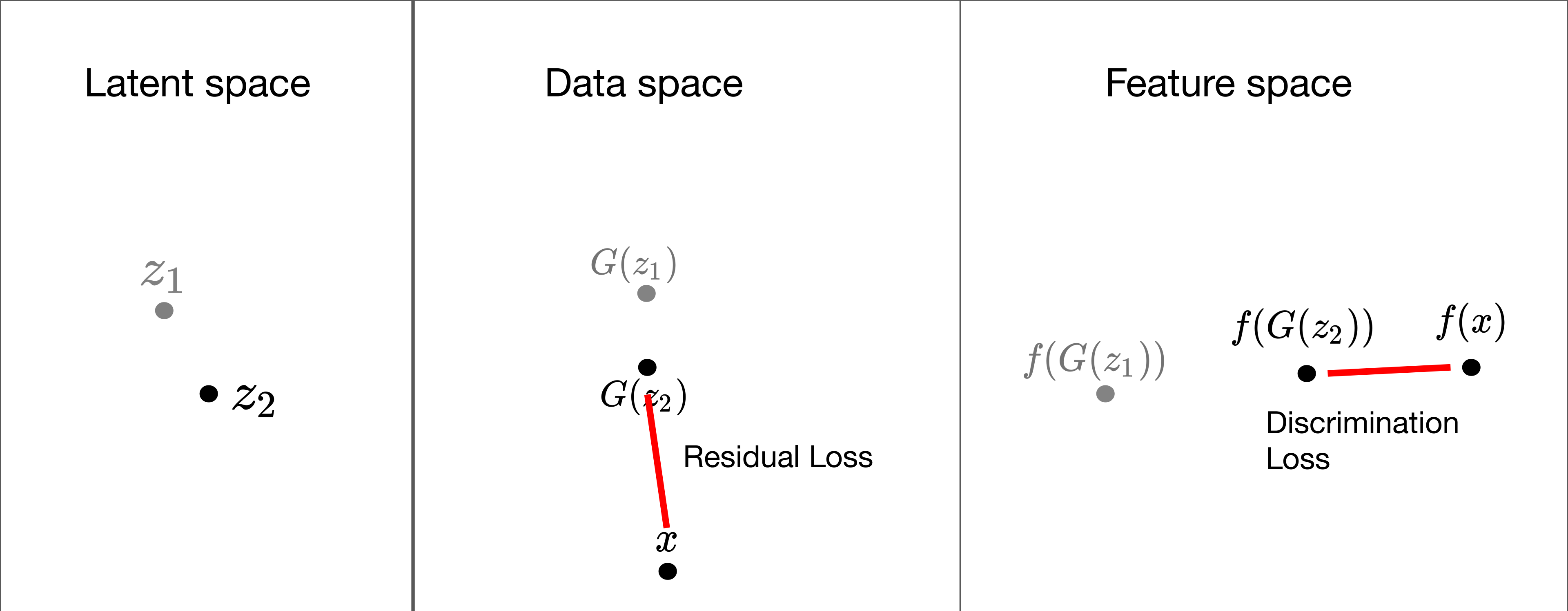
Latent space

Data space

Feature space

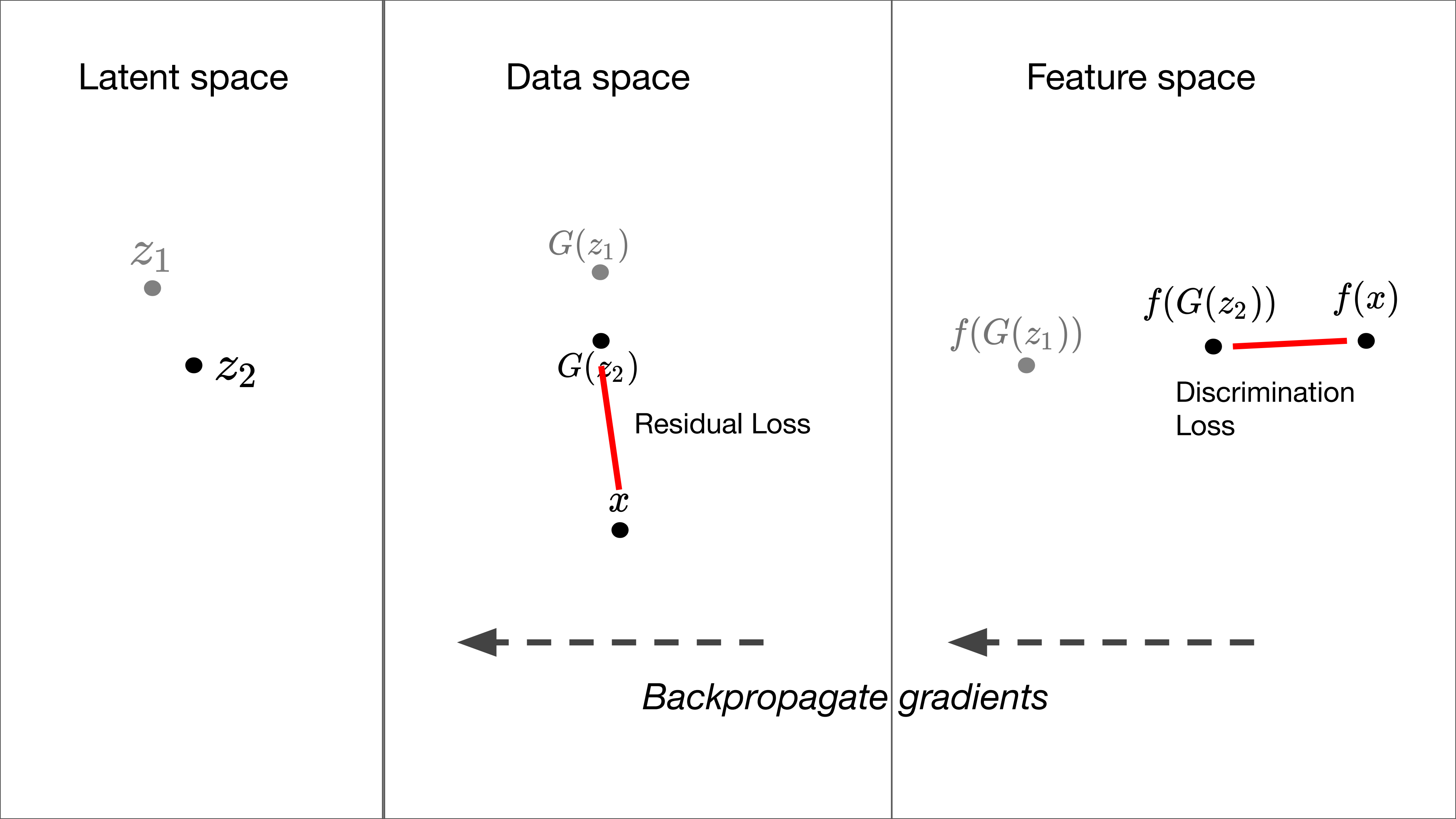


Latent space
z_1
z_2
Data space
$G(z_1)$
$G(z_2)$
x
Feature space
$f(G(z_1))$
$f(G(z_2))$
$f(x)$



$$\mathcal{L}_R(\mathbf{z}_\gamma) = \sum |\mathbf{x} - G(\mathbf{z}_\gamma)|.$$

$$\mathcal{L}_D(\mathbf{z}_\gamma) = \sum |\mathbf{f}(\mathbf{x}) - \mathbf{f}(G(\mathbf{z}_\gamma))|,$$



Latent space

Data space

Feature space

z_1

z_2

$G(z_1)$

$G(z_2)$

Residual Loss

x

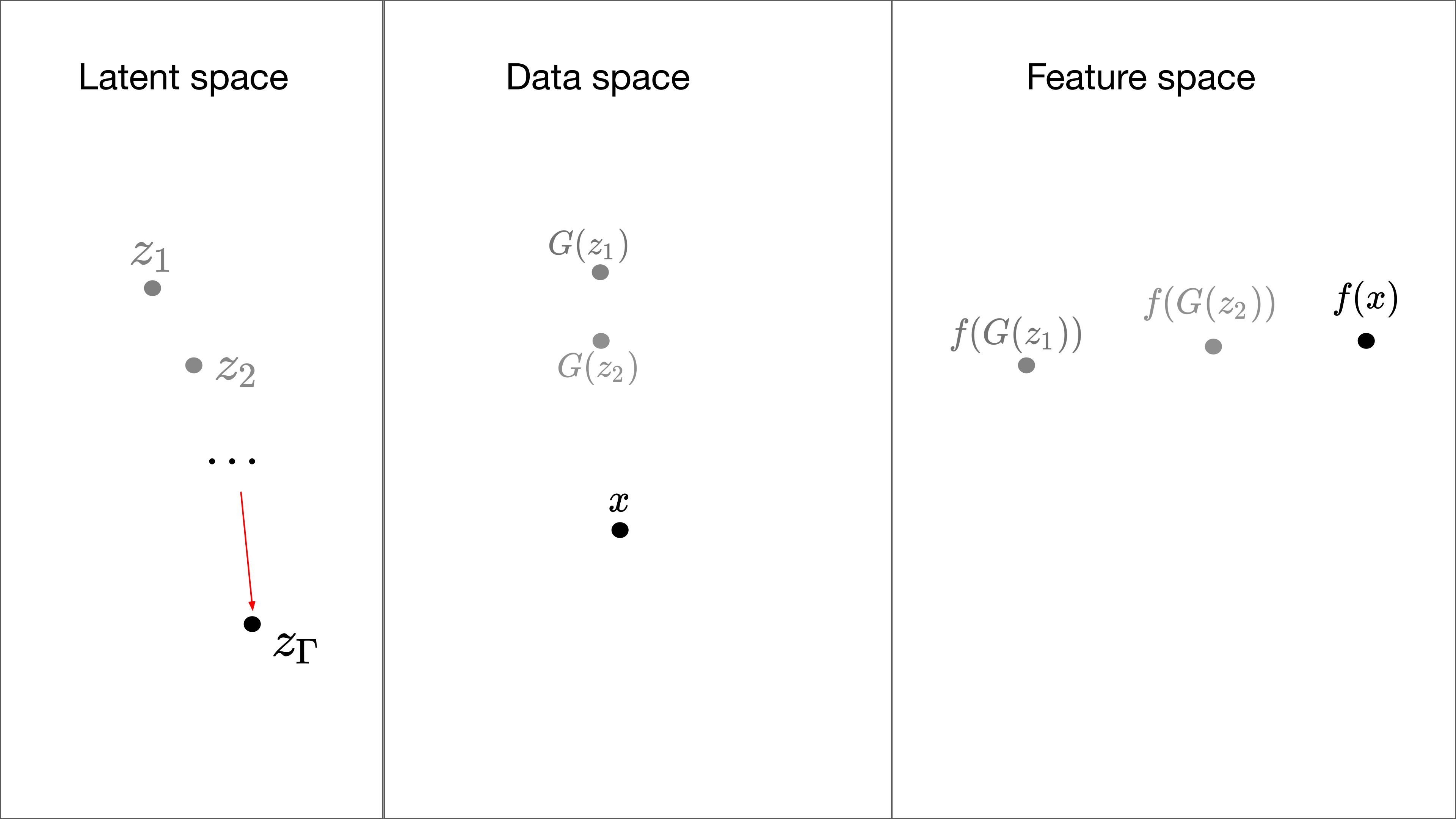
$f(G(z_1))$

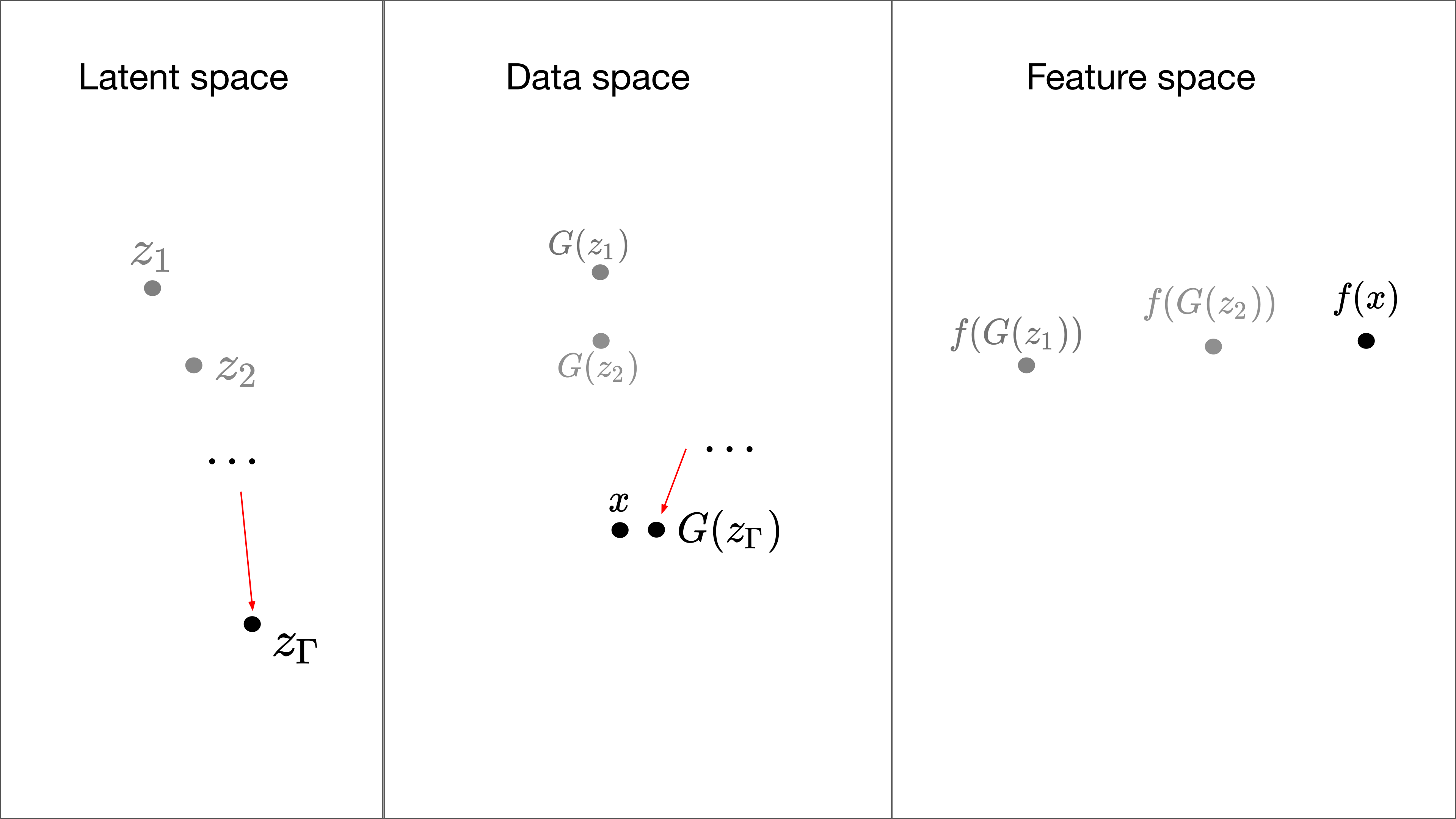
$f(G(z_2))$

$f(x)$

Discrimination Loss

Backpropagate gradients

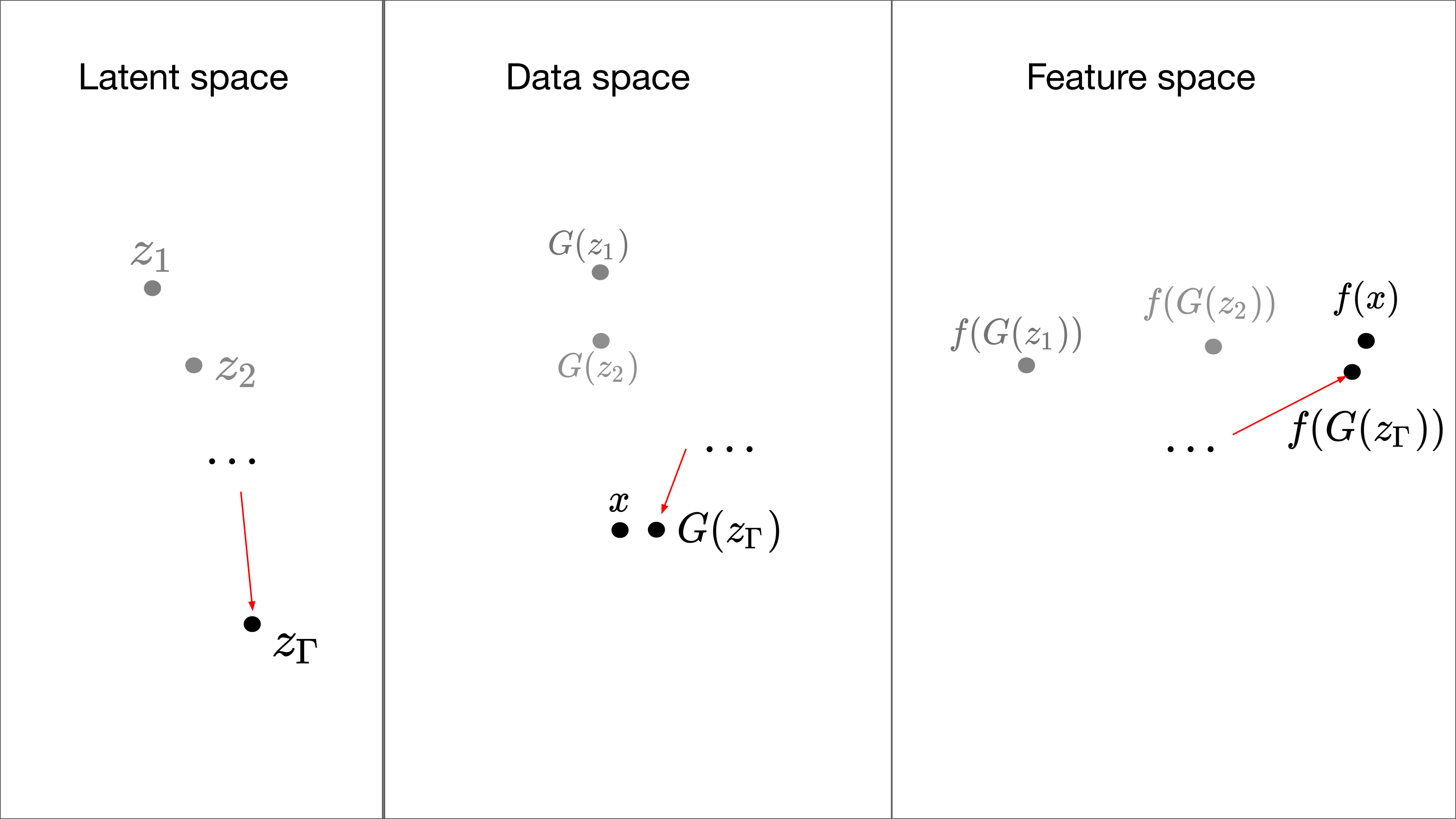

Latent space
Data space
Feature space
z_1
z_2
$\dots$
z_Γ
$G(z_1)$
$G(z_2)$
x
$f(G(z_1))$
$f(G(z_2))$
$f(x)$



Latent space

Data space

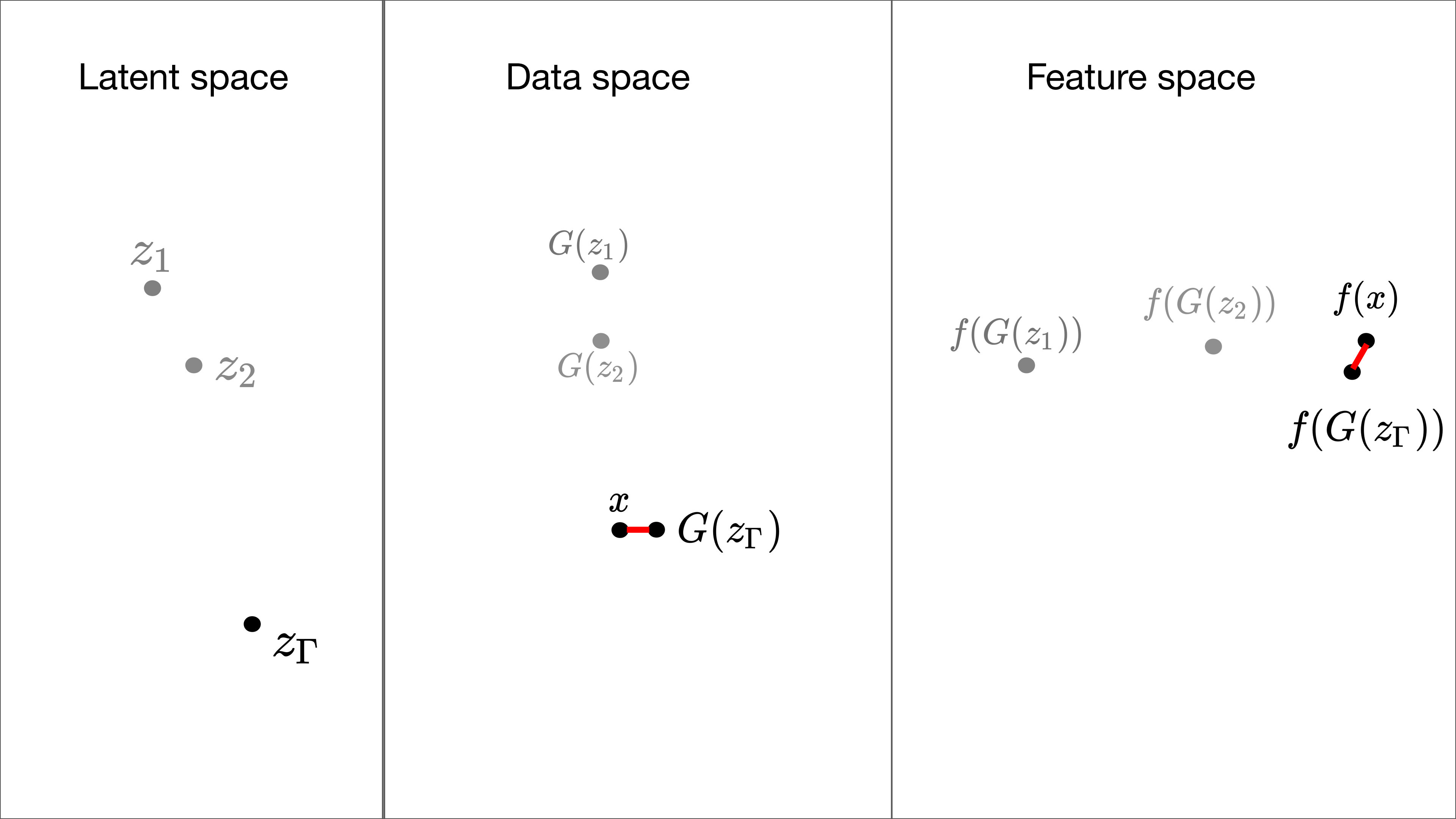
Feature space



Latent space

Data space

Feature space


Latent space
z_1
z_2
z_Γ
Data space
$G(z_1)$
$G(z_2)$
x
$G(z_\Gamma)$
Feature space
$f(G(z_1))$
$f(G(z_2))$
$f(x)$
$f(G(z_\Gamma))$

Overall loss

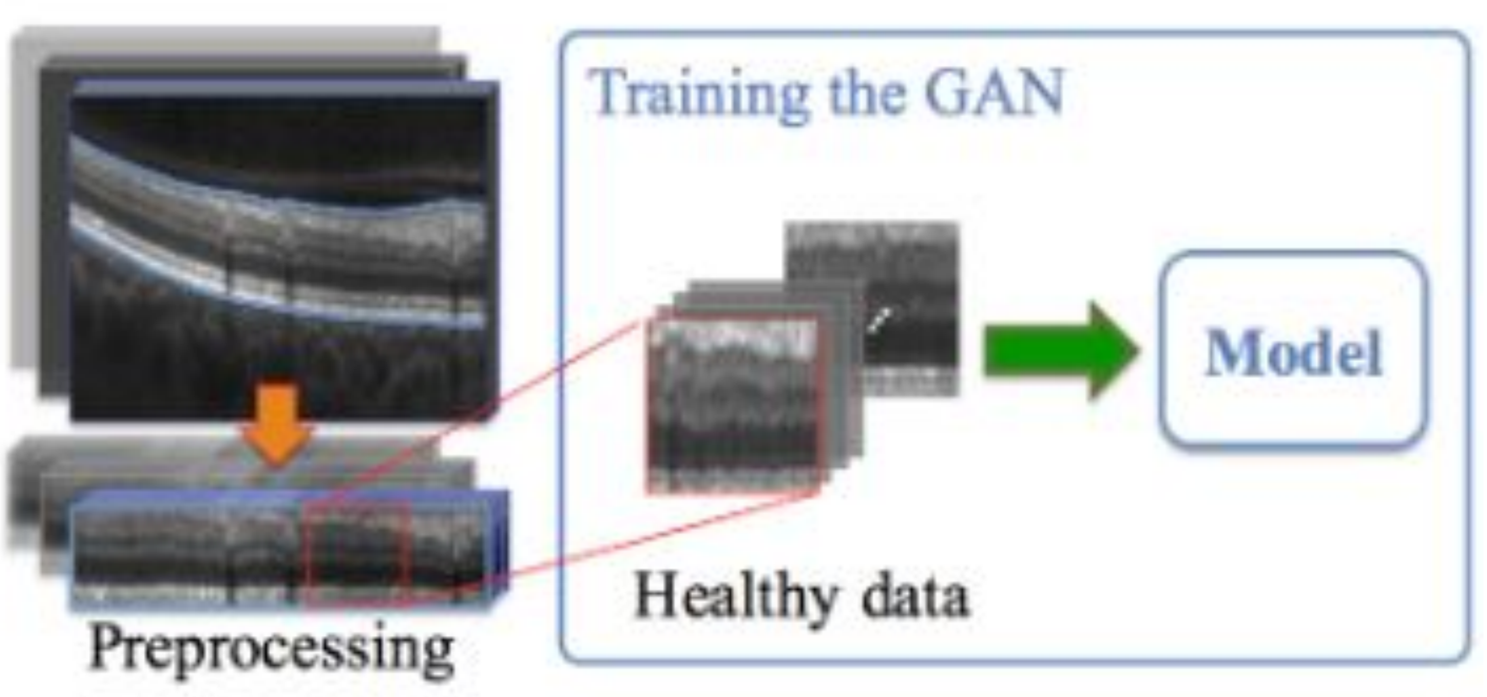
Only coefficients of z are adapted via back-propagation. Trained params of discriminator and generator are kept fixed.

$$\mathcal{L}(\mathbf{z}_\gamma) = (1 - \lambda) \cdot \mathcal{L}_R(\mathbf{z}_\gamma) + \lambda \cdot \mathcal{L}_D(\mathbf{z}_\gamma)$$

Anomaly score

$$A(\mathbf{x}) = (1 - \lambda) \cdot R(\mathbf{x}) + \lambda \cdot D(\mathbf{x})$$

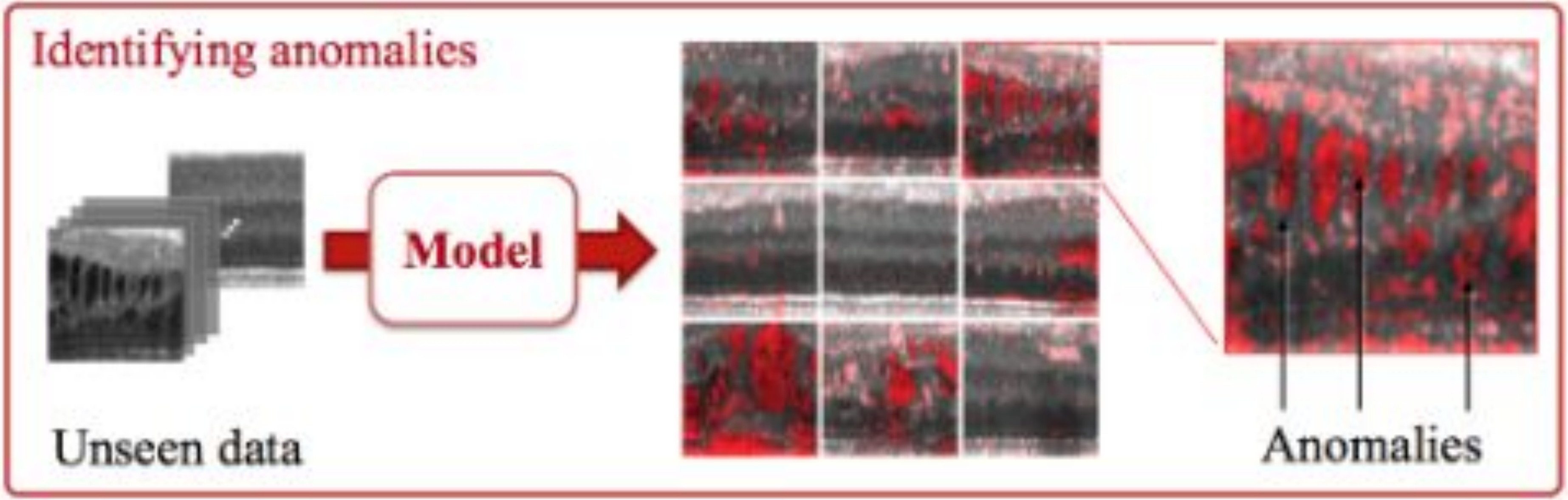
Image Patches



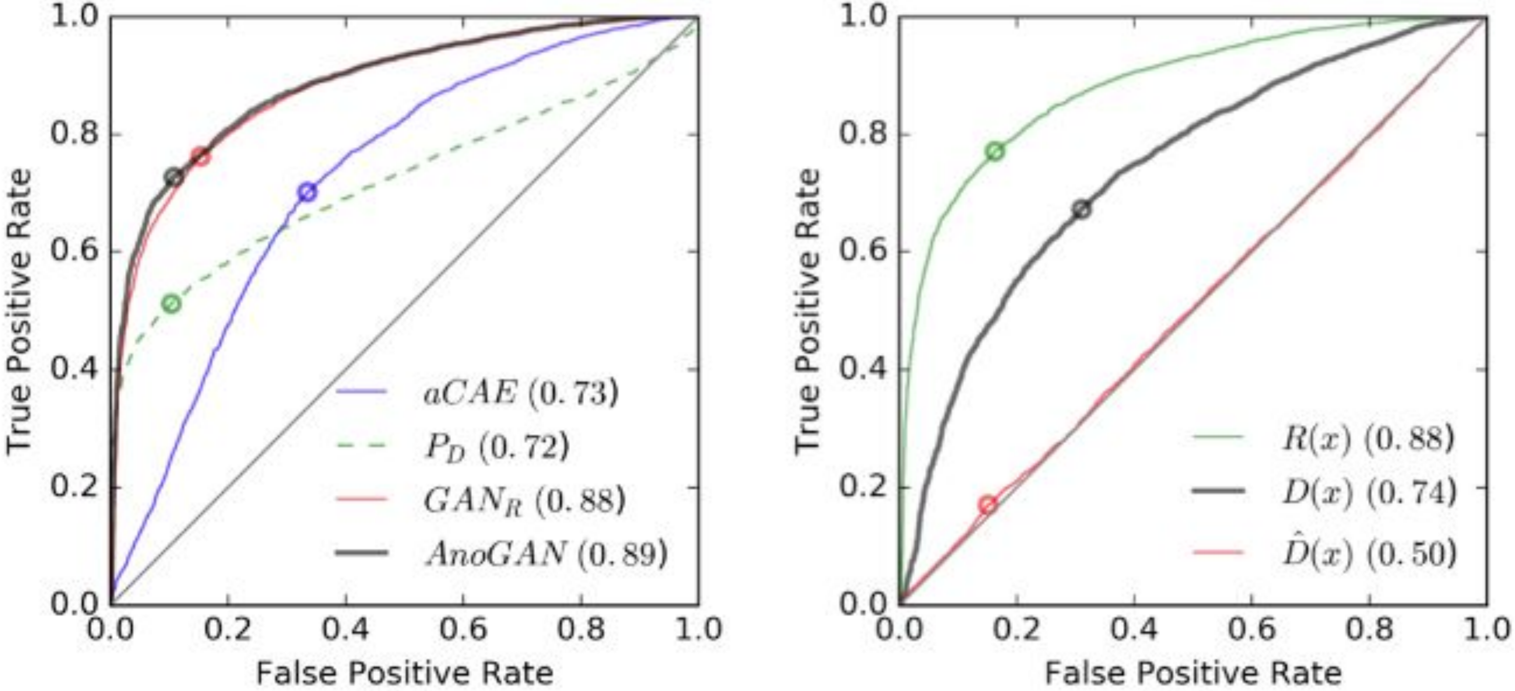
Training: Extract c x c patches for each image.

For testing, we are given image patches and their corresponding labels - 0 or 1.

Image Patches: identifying anomalies



Results



True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
$aCAE$ (0.73)
P_D (0.72)
GAN_R (0.88)
$AnoGAN$ (0.89)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
$R(x)$ (0.88)
$D(x)$ (0.74)
$\hat{D}(x)$ (0.50)

(a)

(b)

Results

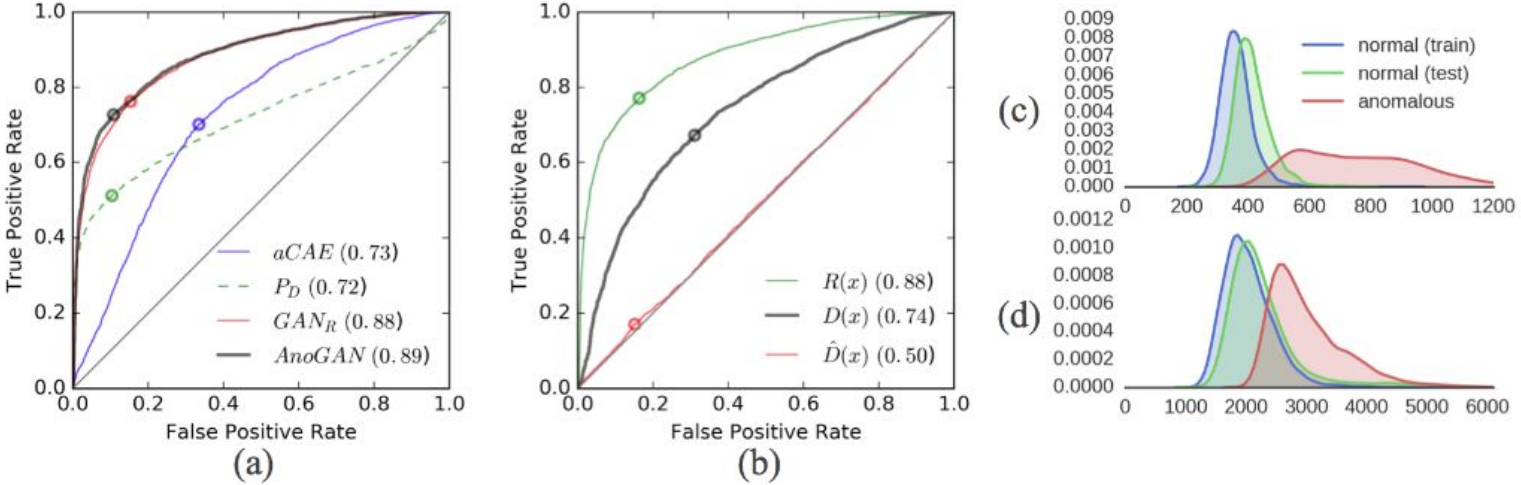
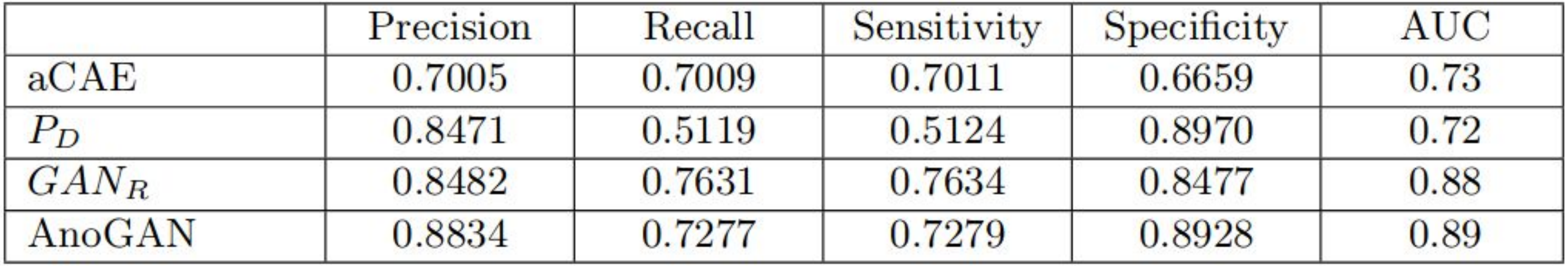


Fig. 4. Image level anomaly detection performance and suitability evaluation. (a) Model comparison: ROC curves based on $aCAE$ (blue), GAN_R (red), the proposed $AnoGAN$ (black), or on the output P_D of the trained discriminator (green). (b) Anomaly score components: ROC curves based on the *residual score* $R(\mathbf{x})$ (green), the *discrimination score* $D(\mathbf{x})$ (black), or the *reference discrimination score* $\hat{D}(\mathbf{x})$ (red). (c) Distribution of the *residual score* and (d) of the *discrimination score*, evaluated on normal images of the training set (blue) or test set (green), and on images extracted from diseased cases (red).

Results



	Precision	Recall	Sensitivity	Specificity	AUC
aCAE	0.7005	0.7009	0.7011	0.6659	0.73
P_D	0.8471	0.5119	0.5124	0.8970	0.72
GAN_R	0.8482	0.7631	0.7634	0.8477	0.88
AnoGAN	0.8834	0.7277	0.7279	0.8928	0.89

FakeGAN: Detecting Deceptive Reviews using Generative Adversarial Networks

Just like GAN, but uses two Discriminator models.

Only one Discriminator is used as a classifier.

Unlike most GAN models, the focus is on improving Discriminator, not Generator.

Based heavily on SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient, 2017.

FakeGAN

$X = X_D \cup X_T$ Deceptive reviews and truthful reviews.

G - generator

D - distinguishes truthful vs. deceptive reviews.

D’- distinguishes between samples generated by G and deceptive samples in the dataset.

D’ guides the generator G to produce samples similar to X_D

D guides the generator to generate samples which seem truthful to D.

FakeGAN

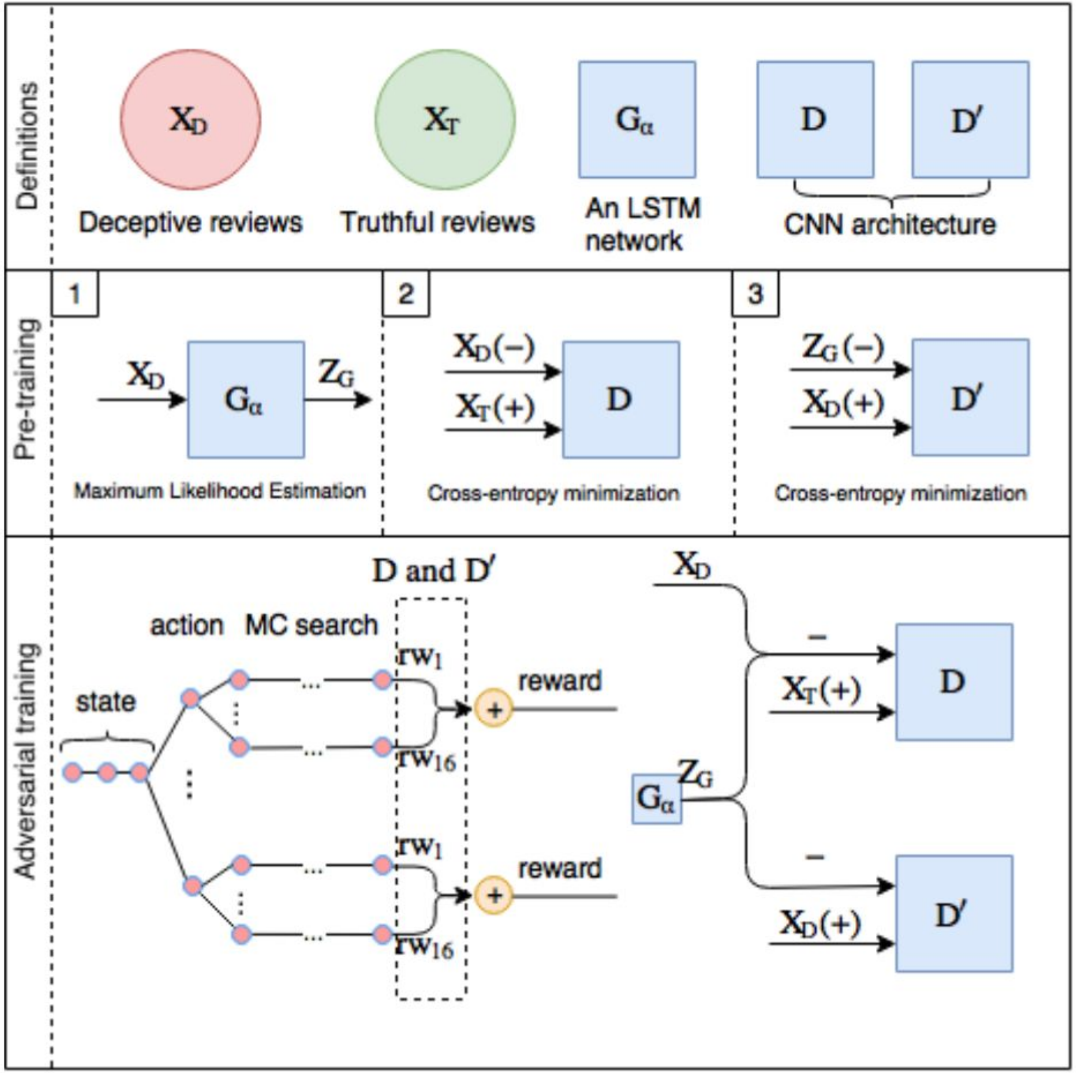
G tries to fool:

D′ by generating deceptive (not generated) reviews

D by generating truthful (not generated or X_D) reviews.

G is a policy model from reinforcement learning

G is trained by using a policy gradient and Monte Carlo (MC) search on the expected end reward from the discriminative models D and D′


Definitions
X_D
Deceptive reviews
X_T
Truthful reviews
G_α
An LSTM network
D
D′
CNN architecture
Pre-training
1
X_D
G_α
Z_G
Maximum Likelihood Estimation
2
X_D(−)
X_T(+)
D
Cross-entropy minimization
3
Z_G(−)
X_D(+)
D′
Cross-entropy minimization
Adversarial training
D and D′
action
MC search
state
rw1
rw16
reward
rw1
rw16
reward
+
+
X_D
−
X_T(+)
D
G_α
Z_G
−
X_D(+)
D′

The generator G is defined as a policy model in reinforcement-learning

Action value function:

$$A_{G_\alpha,D,D'}(a = S_L, s = S_{1:L-1}) = D(S_{1:L}) + D'(S_{1:L})$$

To estimate the action value function in every timestep ***t*** a Monte Carlo search is applied ***N*** times with a roll-out policy G'_γ to sample the undetermined last ***L−t*** tokens:

$$\{S^1_{1:L}, S^2_{1:L}, ..., S^N_{1:L}\} = MC_{G'_\gamma}(S_{1:t}, N)$$

Action value estimation as a function of **t**:

$$A_{G_\alpha,D,D'}(a = S_t, s = S_{1:t-1}) = \begin{cases} \frac{1}{N}\sum_{i=1}^{N}(D(S^i_{1:L}) + D'(S^i_{1:L})) \\ D(S_{1:L}) + D'(S_{1:L}) \end{cases}$$

Overall objective function:

$$J(\alpha) = \sum_{S_1 \in \chi} G_\alpha(S_1|S_0) \ . \ A_{G_\alpha,D,D'}(a = S_1, s = S_0)$$

Gradient:

$$\nabla_\alpha J(\alpha) = \sum_{t=1}^{T} \mathbb{E}_{S_{1:t-1} \sim G_\alpha}[\sum_{S_t \in \chi} \nabla_\alpha G_\alpha(S_t|S_{1:t-1}) \ . \ A_{G_\alpha,D,D'}(a = S_t, s = S_{1:t-1})]$$

Update generator’s parameters:

$$\alpha \leftarrow \alpha + \lambda \nabla_{\alpha} J(\alpha)$$

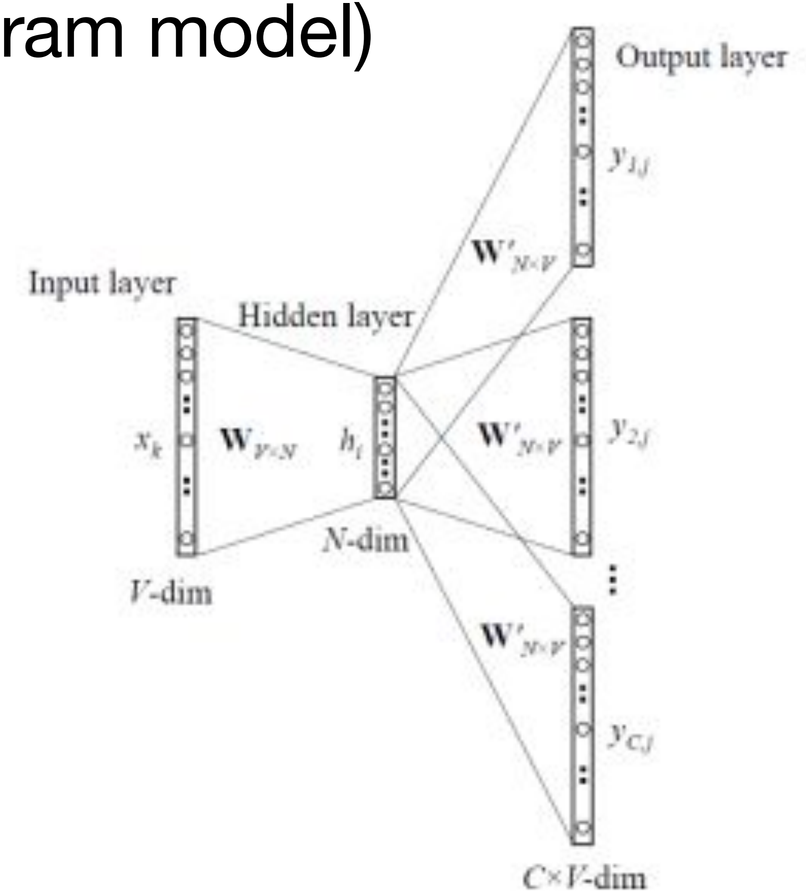
and re-train discriminative models D and D′ using following objective functions:

$$min(-\mathbb{E}_{S\sim X_T}[\log D(S)] - \mathbb{E}_{S\sim X_D \vee G_\alpha}[1 - \log D(S)])$$

$$min(-\mathbb{E}_{S\sim X_D}[\log D'(S)] - \mathbb{E}_{S\sim G_\alpha}[1 - \log D'(S)])$$

Word embeddings (skip-gram model)

- We use word embeddings to "translate" words into vectors, so it can be fed into neural networks
- Popular and standard way to represent words in NLP tasks
- Word embeddings capture hidden information about a language, like word analogies or semantics
- We are using a pre-trained word embedding model, FastText [5], [6]



3. Proposed Work

Origin of our proposed work

In AnoGAN, Schlegl et al. [2] uses healthy anatomy image patches (normal data) to train a generative adversarial model, then uses anomaly scores to detect anomalous image patches.

Our proposed work is an adaptation of this approach towards text based anomaly detection

We came up with 2 possible adaptations from image to text data:

1. Anomaly detection as text classification using GANs
2. AnoGAN based approach using text patches

Anomaly detection as text classification using GANs

1. Formulate the anomaly detection task as a two-class classification problem of discriminating between normal and anomalous data

2. Train the GAN using only normal data
   - Generator learns the distribution of normal data
   - Discriminator learns what normal data looks like

3. Perform anomaly detection using discriminator, and classify normal vs. anomalous
   - When classifying feed both normal and anomalous data into discriminator

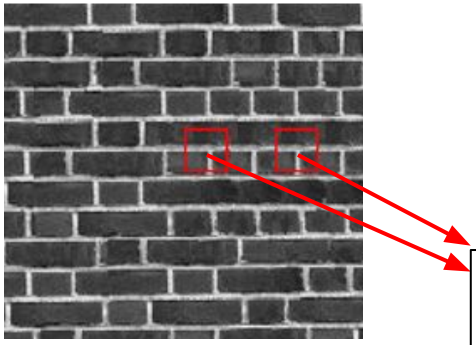
Anomaly detection as text classification using GANs

- Hypothesis:
  - Discriminator will learn what normal data looks like, and will be able to classify it
  - When presented with anomalous data, it will recognize it as not normal, and classify it anomalous
- BUT, discriminator learns to distinguish between real vs. generated, and we are trying to classify normal vs. anomalous
  - One big assumption with this approach is that classifying real vs. generated behaves that same way as normal vs. anomalous
  - If this assumption fails, this can be corrected in future work
- Why would this work better ?
  - The use of GAN in text based anomaly detection is very much unexplored, while it was proven to be successful in image based anomaly detection

Text patches

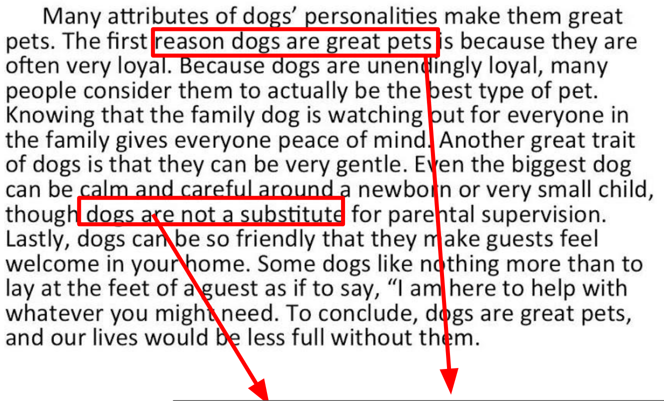
- Second approach requires finding an equivalent of image patches for text

Here are image patches:



2D image patches of size c×c from randomly sampled positions

We propose “text patches”:



Many attributes of dogs’ personalities make them great pets. The first reason dogs are great pets is because they are often very loyal. Because dogs are unendingly loyal, many people consider them to actually be the best type of pet. Knowing that the family dog is watching out for everyone in the family gives everyone peace of mind. Another great trait of dogs is that they can be very gentle. Even the biggest dog can be calm and careful around a newborn or very small child, though dogs are not a substitute for parental supervision. Lastly, dogs can be so friendly that they make guests feel welcome in your home. Some dogs like nothing more than to lay at the feet of a guest as if to say, “I am here to help with whatever you might need. To conclude, dogs are great pets, and our lives would be less full without them.

1D “text patch” of size c consecutive words from randomly sampled positions

AnoGAN based approach using Text patches

We propose to:

1. Replace image patches with text patches
2. Adapt AnoGan to work with text patches and detect text based anomalies

Hypothesis:

- Anomaly score calculation with AnoGAN is possible for text data as well

Why would this work better ?

- Some text patches could contain anomalous contextual details summarized by the word embeddings, which the AnoGAN could be trained to detect

4. Experimental design

Experiments - Text classification approach

- Anomaly detection in the form of depression detection

1. Train GAN on anomalous data
    - Use Generator to generate more anomalous data → fix class imbalance issue

2. Train GAN model on normal data
    - Try out different model architectures for both generator and discriminator
    - Use discriminator to classify between normal and anomalous

Experiments - AnoGAN based approach

Experiments:

1. Train on “non-anomalous” text (non-depression tweets).
2. Given a query tweet, use generator to generate the closest possible match.
3. Use word-vector distance to compute anomaly score. If any text-patch is anomalous - whole text is anomalous.
4. Utilize LSTM for remembering sequential text data.

Datasets:

1. Depressive tweets (Example query: “Is this tweet depressive?”)
2. Shakespeare plays. (Example query: “Is this text written in style of Shakespeare?”)

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[2] Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., & Langs, G. (2017, June). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In International Conference on Information Processing in Medical Imaging (pp. 146-157). Springer, Cham.

[3] Aroyehun, S. T., & Gelbukh, A. (2018). Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (pp. 90-97).

[4] Risch, J., & Krestel, R. (2018). Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)* (pp. 150-158).

[5] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

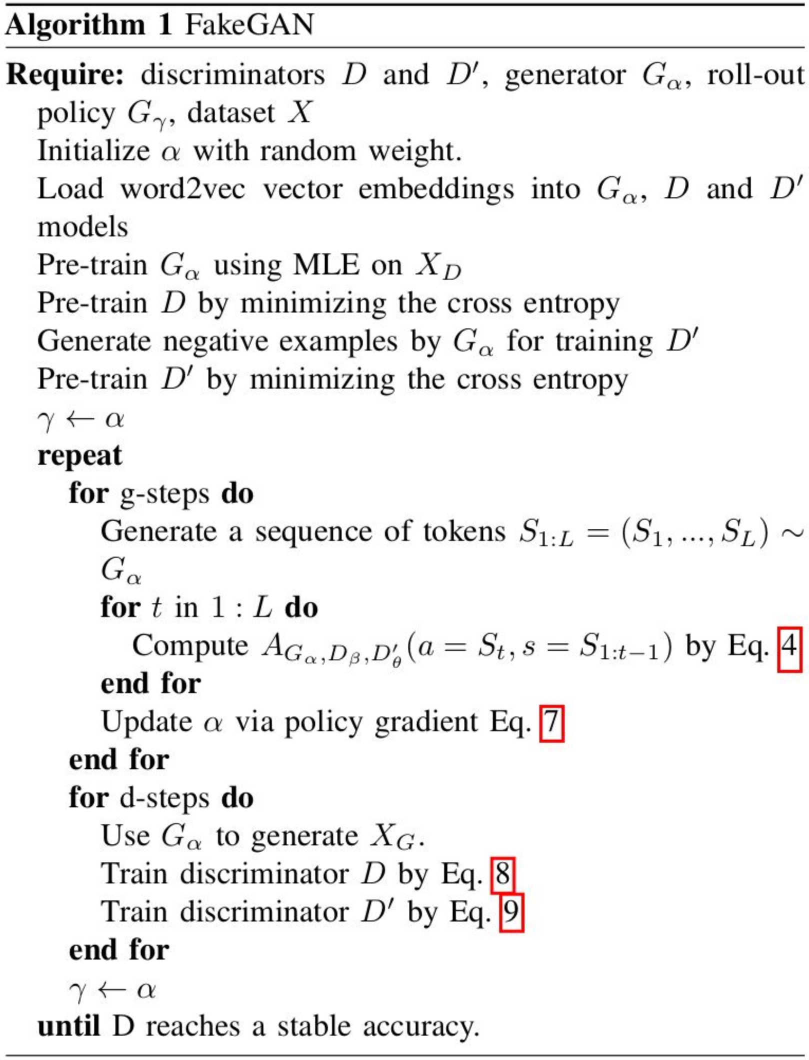
[6] Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

[7] Aghakhani, H., Machiry, A., Nilizadeh, S., Kruegel, C., & Vigna, G. (2018). Detecting Deceptive Reviews using Generative Adversarial Networks. *arXiv preprint arXiv:1805.10364*.

[8] Zunaira Jamil, Diana Inkpen, Prasadith Buddhitha, and Kenton White. Monitoring Tweets for Depression to Detect At-risk Users. In Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology - From Linguistic Signal to Clinical Reality (CLPsych 2017), at ACL 2017, Vancouver, BC, Aug 2017

The End

Extra slides



Algorithm 1 FakeGAN

```
Require: discriminators $D$ and $D'$, generator $G_\alpha$, roll-out policy $G_\gamma$, dataset $X$
  Initialize $\alpha$ with random weight.
  Load word2vec vector embeddings into $G_\alpha$, $D$ and $D'$ models
  Pre-train $G_\alpha$ using MLE on $X_D$
  Pre-train $D$ by minimizing the cross entropy
  Generate negative examples by $G_\alpha$ for training $D'$
  Pre-train $D'$ by minimizing the cross entropy
  $\gamma \leftarrow \alpha$
  repeat
    for g-steps do
      Generate a sequence of tokens $S_{1:L} = (S_1, ..., S_L) \sim G_\alpha$
      for $t$ in $1 : L$ do
        Compute $A_{G_\alpha, D_\beta, D'_\theta}(a = S_t, s = S_{1:t-1})$ by Eq. 4
      end for
      Update $\alpha$ via policy gradient Eq. 7
    end for
    for d-steps do
      Use $G_\alpha$ to generate $X_G$.
      Train discriminator $D$ by Eq. 8
      Train discriminator $D'$ by Eq. 9
    end for
    $\gamma \leftarrow \alpha$
  until D reaches a stable accuracy.
```

Generative Adversarial Networks (GAN)

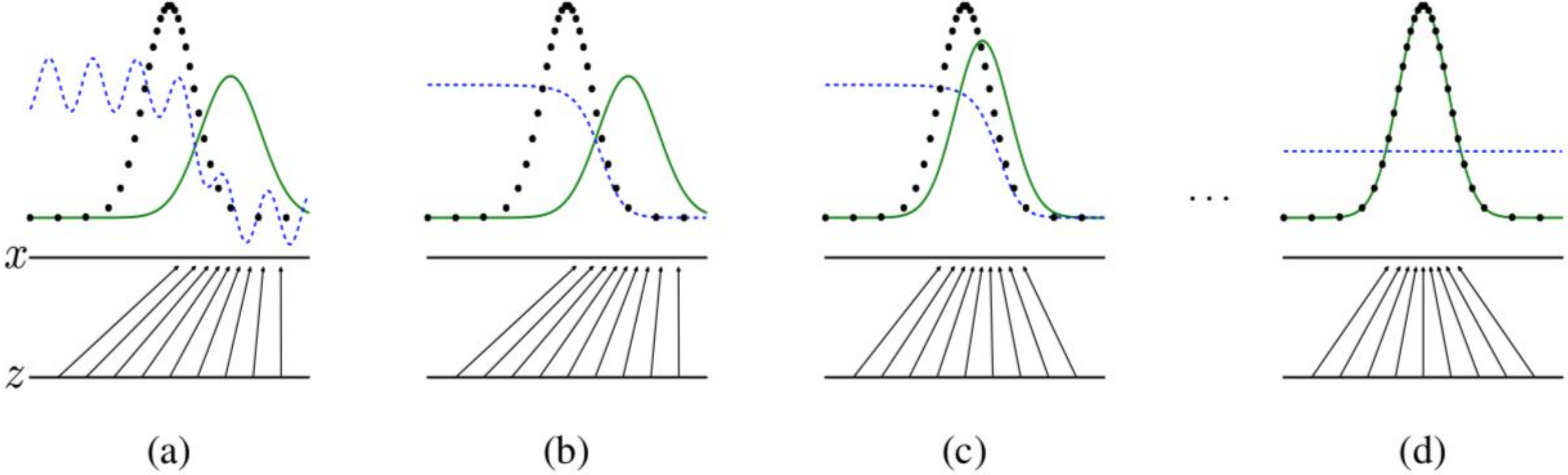
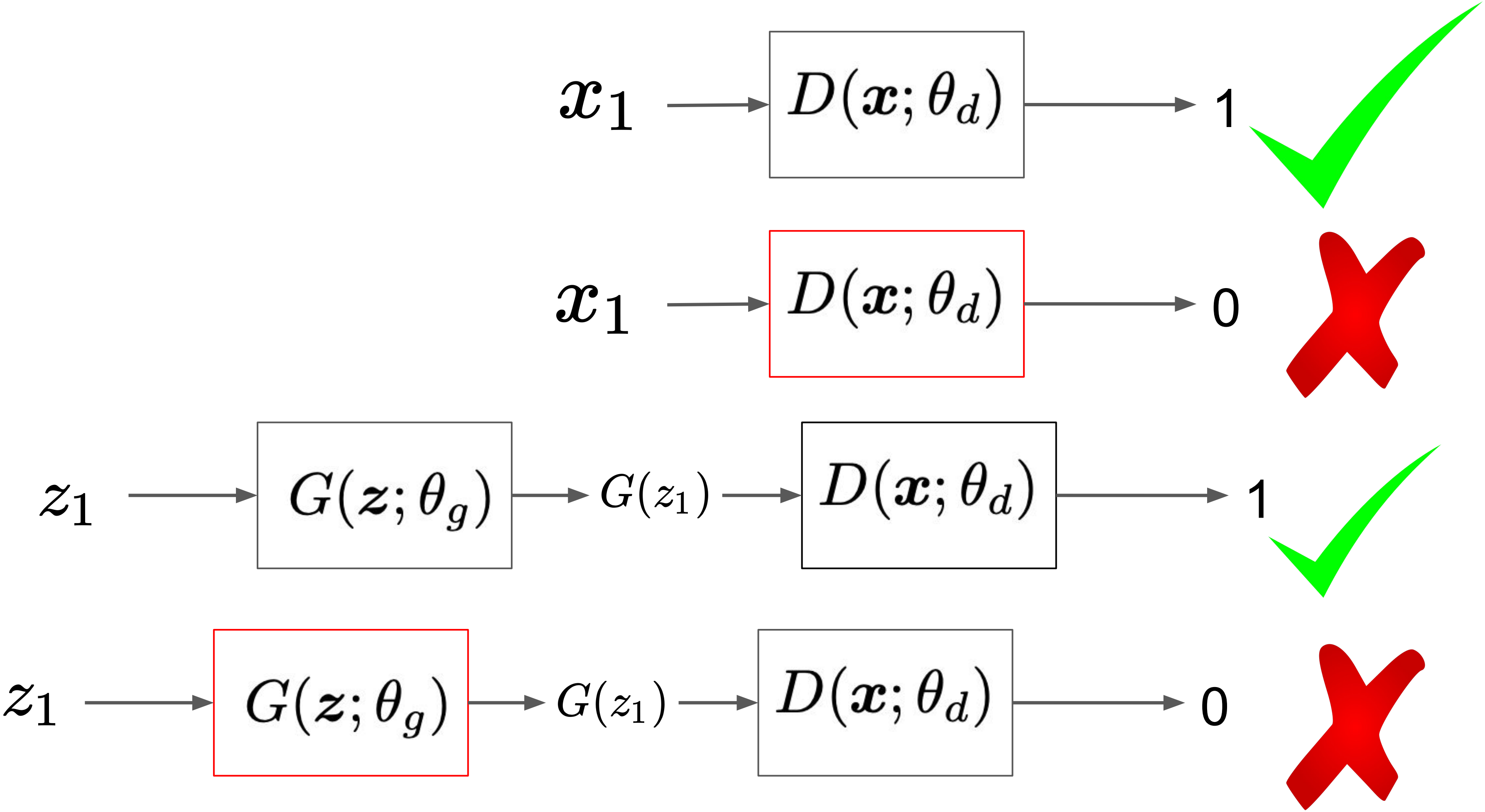


Figure 1: Generative adversarial nets are trained by simultaneously updating the **d**iscriminative distribution (D, blue, dashed line) so that it discriminates between samples from the data generating distribution (black, dotted line) $p_{\boldsymbol{x}}$ from those of the **g**enerative distribution p_g (G) (green, solid line). The lower horizontal line is



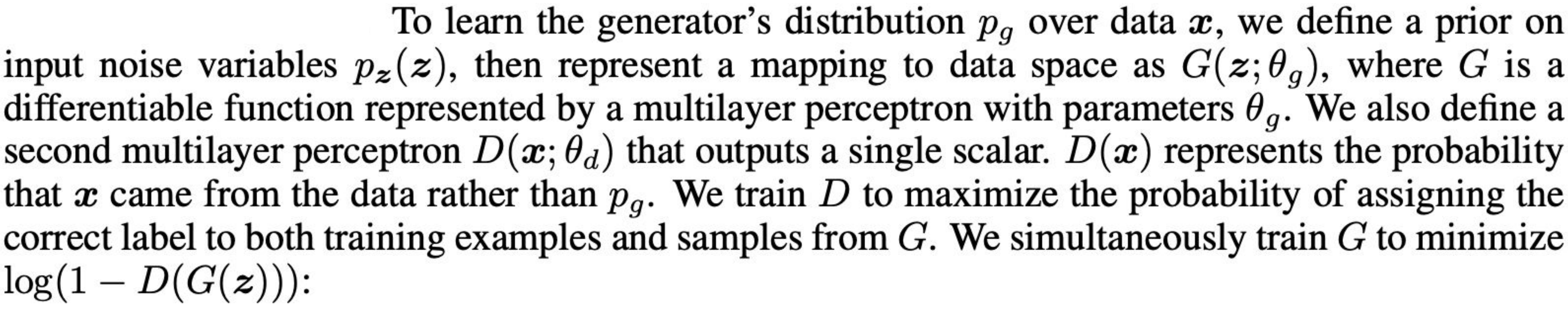
$x_1 \rightarrow \boxed{D(\boldsymbol{x}; \theta_d)} \rightarrow 1$ ✓

$x_1 \rightarrow \boxed{D(\boldsymbol{x}; \theta_d)} \rightarrow 0$ ✗

$z_1 \rightarrow \boxed{G(\boldsymbol{z}; \theta_g)} \rightarrow G(z_1) \rightarrow \boxed{D(\boldsymbol{x}; \theta_d)} \rightarrow 1$ ✓

$z_1 \rightarrow \boxed{G(\boldsymbol{z}; \theta_g)} \rightarrow G(z_1) \rightarrow \boxed{D(\boldsymbol{x}; \theta_d)} \rightarrow 0$ ✗

Generative Adversarial Networks (GAN)



To learn the generator's distribution p_g over data $\boldsymbol{x}$, we define a prior on input noise variables $p_{\boldsymbol{z}}(\boldsymbol{z})$, then represent a mapping to data space as $G(\boldsymbol{z}; \theta_g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θ_g. We also define a second multilayer perceptron $D(\boldsymbol{x}; \theta_d)$ that outputs a single scalar. $D(\boldsymbol{x})$ represents the probability that $\boldsymbol{x}$ came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1 - D(G(\boldsymbol{z})))$:

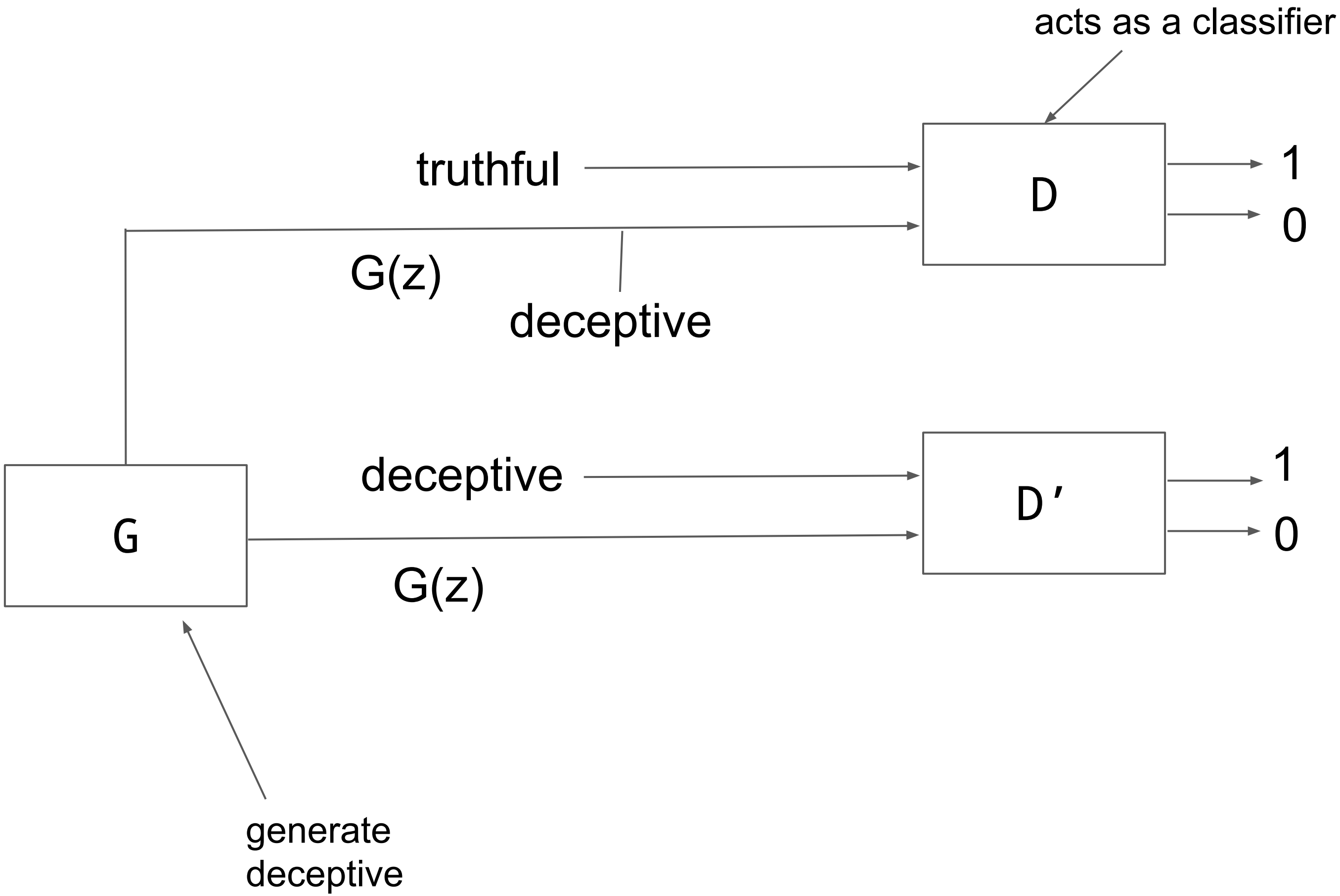
Anomaly Detection Task

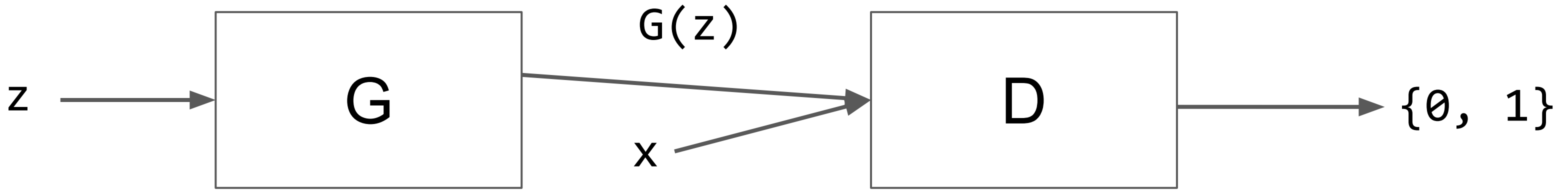
What specific problem is being tackled ?

- Anomaly detection in the form of depression detection in twitter data
- Data comes from:
  - Zunaira Jamil, Diana Inkpen, Prasadith Buddhitha, and Kenton White. Monitoring Tweets for Depression to Detect At-risk Users. In Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology - From Linguistic Signal to Clinical Reality (CLPsych 2017), at ACL 2017, Vancouver, BC, Aug 2017

Why is it important?

- Detect of people at risk of depression, so help could be provided
  - Example: reaching out, offering to listen, point to resources

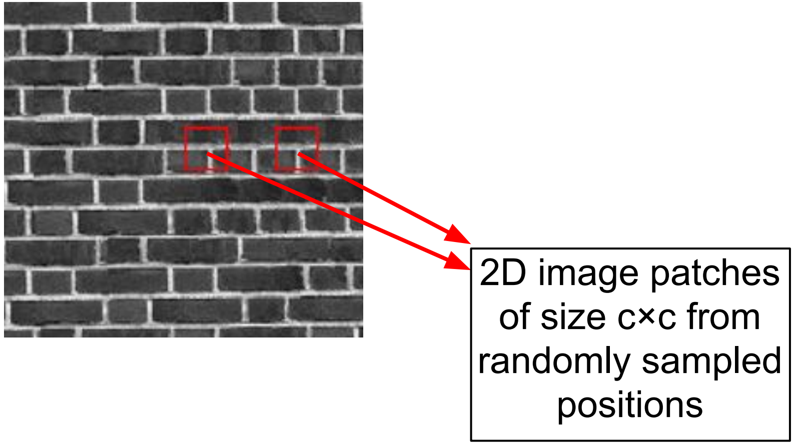

acts as a classifier
truthful
D
1
0
G(z)
deceptive
deceptive
D'
1
0
G
G(z)
generate
deceptive


z
G
G(z)
x
D
{0, 1}

Text patches

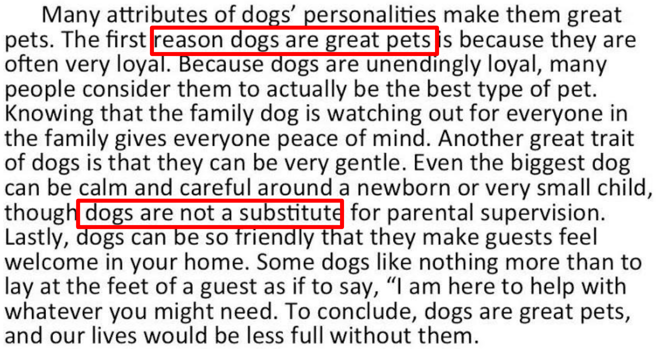
- Second approach requires finding an equivalent of image patches for text

Here are image patches:



2D image patches of size c×c from randomly sampled positions

We propose “text patches”:



Many attributes of dogs’ personalities make them great pets. The first reason dogs are great pets is because they are often very loyal. Because dogs are unendingly loyal, many people consider them to actually be the best type of pet. Knowing that the family dog is watching out for everyone in the family gives everyone peace of mind. Another great trait of dogs is that they can be very gentle. Even the biggest dog can be calm and careful around a newborn or very small child, though dogs are not a substitute for parental supervision. Lastly, dogs can be so friendly that they make guests feel welcome in your home. Some dogs like nothing more than to lay at the feet of a guest as if to say, “I am here to help with whatever you might need. To conclude, dogs are great pets, and our lives would be less full without them.