

Overcoming Mode Collapse and the Curse of Dimensionality

Ke Li

IAS Workshop on Theory of Deep Learning: Where Next?

October 15, 2019

Roadmap

- Two long-standing problems in machine learning and algorithms:

Generative Adversarial Nets (GANs)

Mode Collapse

Nearest Neighbour Search

Curse of Dimensionality

Generative Adversarial Nets

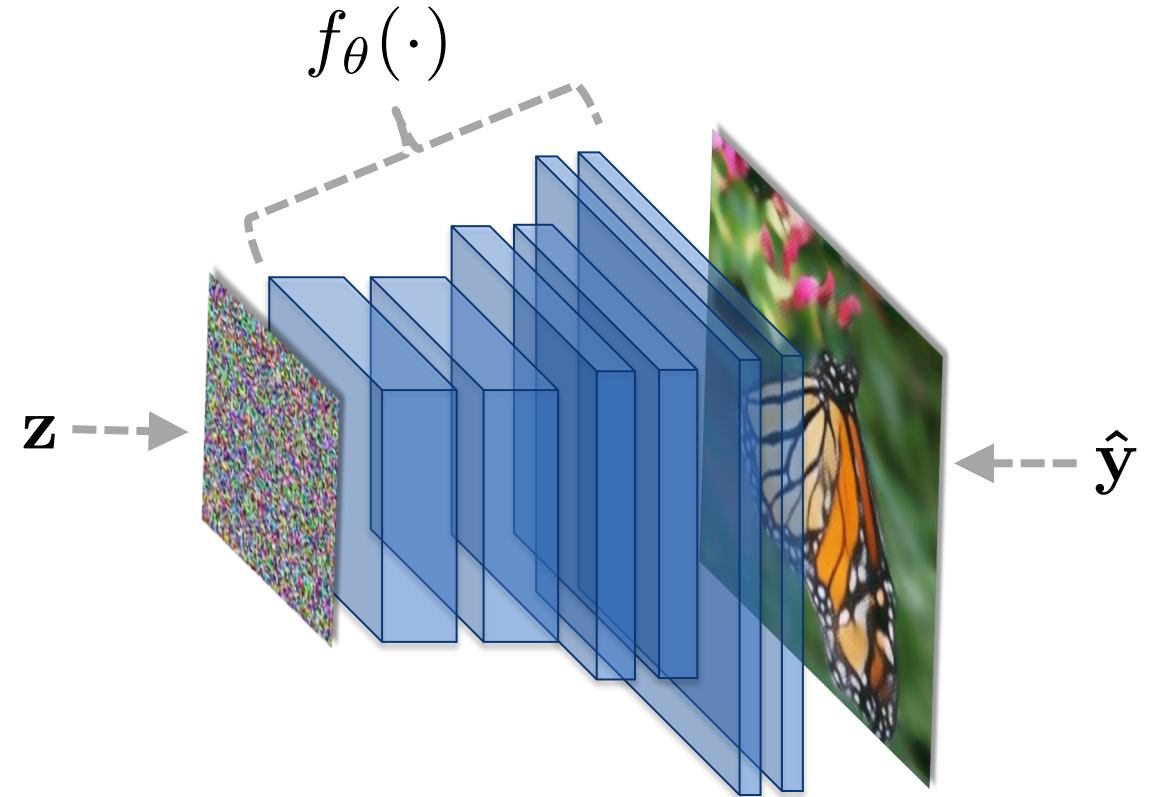
Overcoming Mode Collapse
in Generative Adversarial Nets

Generative Adversarial Nets (GANs)

- The generative model (“the generator”):

$$\hat{\mathbf{y}} = f_{\theta}(\mathbf{z}) \text{ where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$$

- This is an example of an *implicit probabilistic model*, i.e.: a model defined in terms of a sampling procedure.

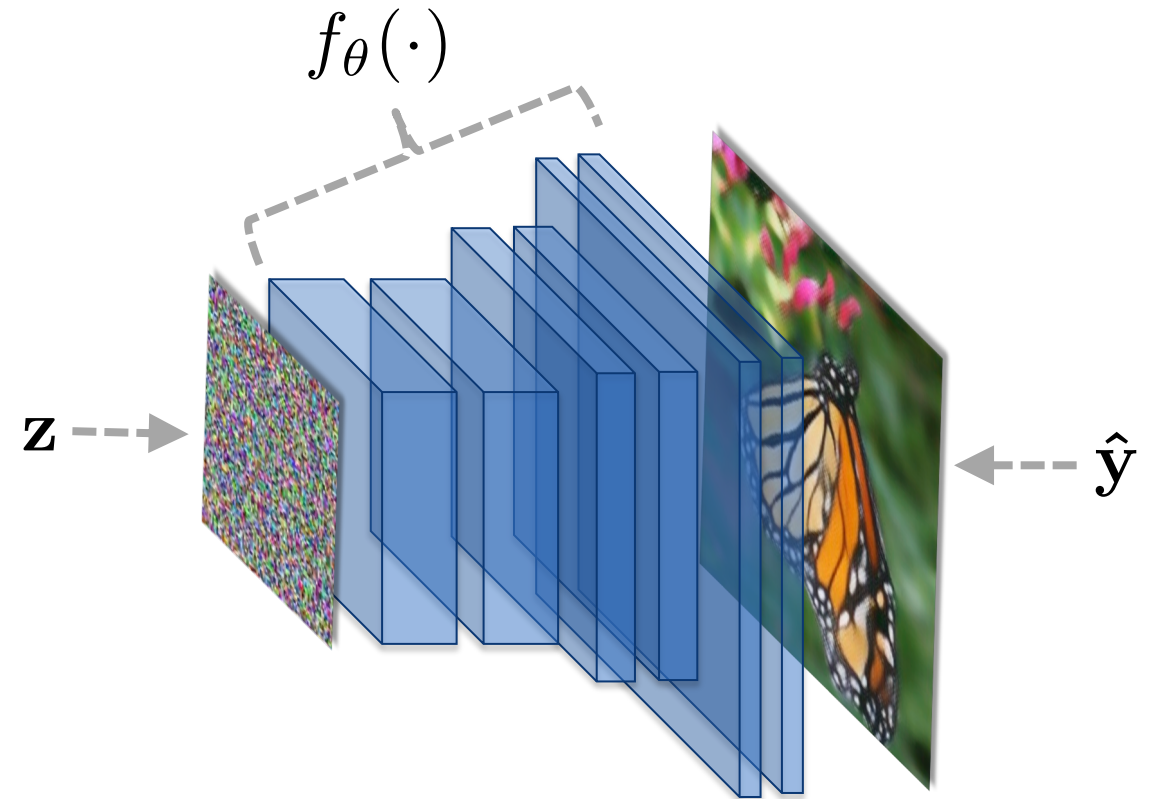


Generative Adversarial Nets (GANs)

- The generative model (“the generator”):

$$\hat{\mathbf{y}} = f_{\theta}(\mathbf{z}) \text{ where } \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$$

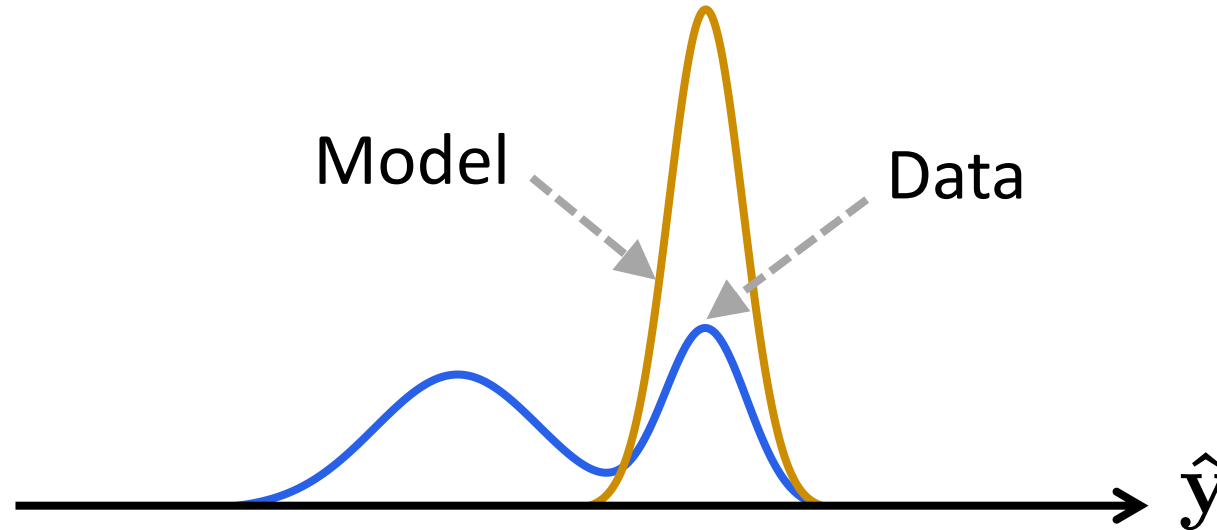
- This is an example of an *implicit probabilistic model*, i.e.: a model defined in terms of a sampling procedure.
- Key Idea: Introduce a classifier (“the discriminator”) and train it to differentiate between model samples and real data.
- Train the generator to fool the discriminator.



Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.



Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.

Model

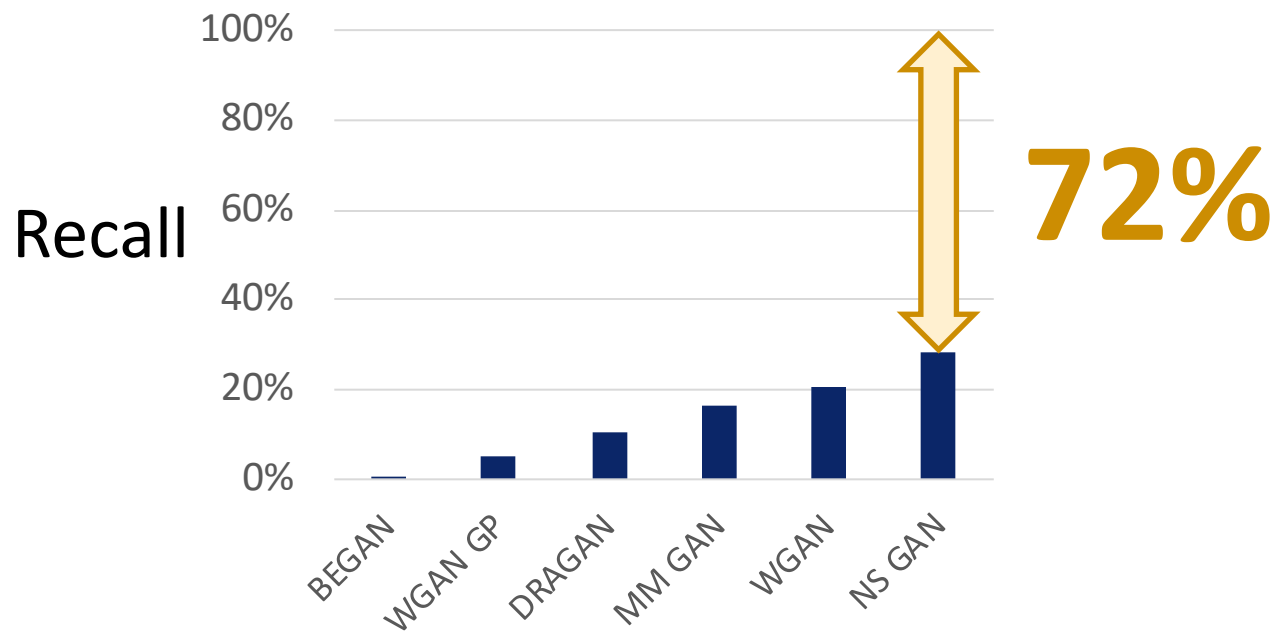
Data

How serious is this problem?

Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.



(Lucic et al., 2018):

- Even the *best* GAN can drop **72%** of modes after extensive hyperparameter tuning

Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.

This turns out to be quite a vexing problem – still not solved.

their learning objective can lead to *mode collapse* (Saatci & Wilson, 2017)

often fail to converge due to mode collapse (Li et al., 2017)

several issues, among these mode-collapse (Guimaraes et al., 2017)

prone to mode collapse (Srivastava et al., 2017)

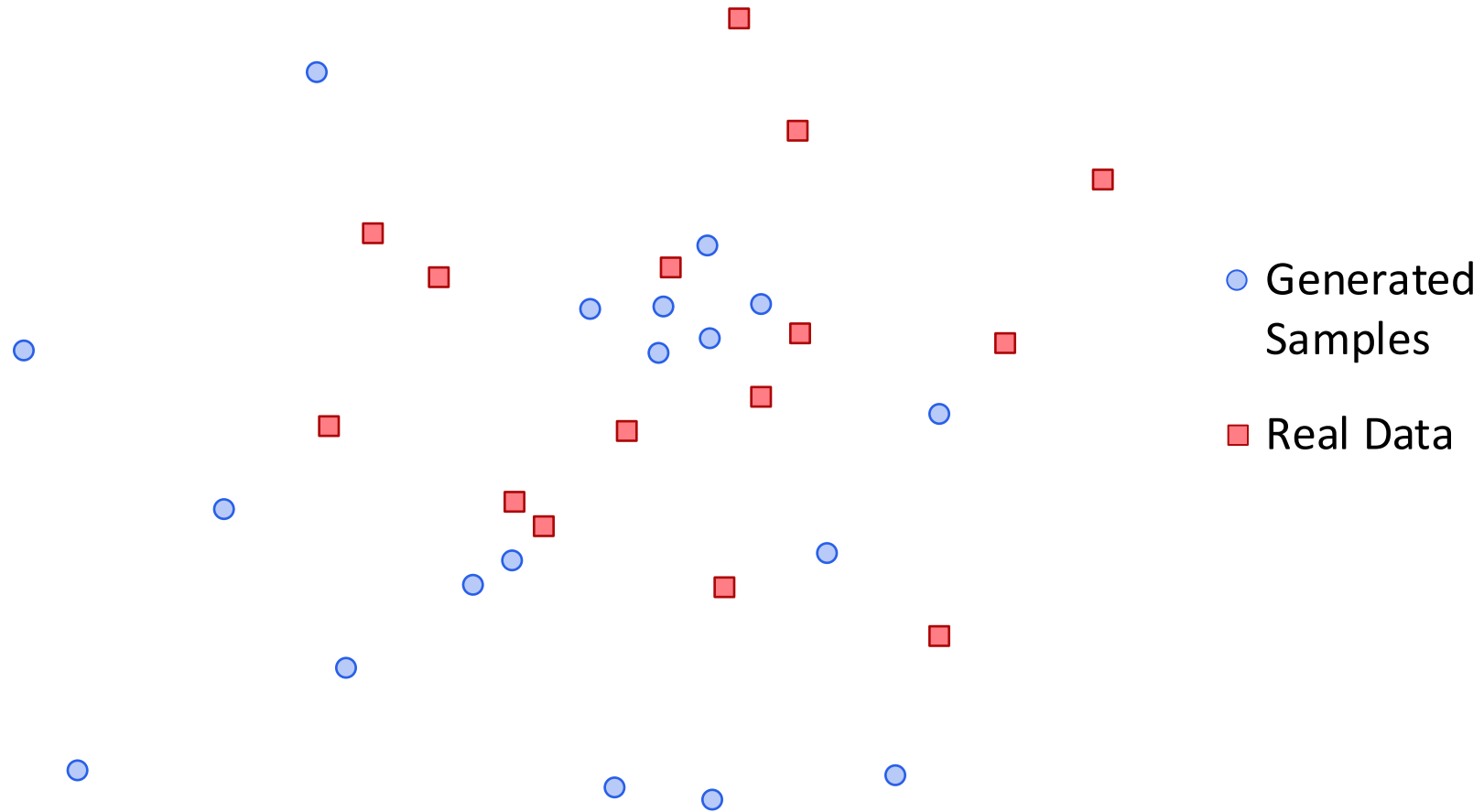
GANs suffer from a severe degree of mode collapse. (Mao et al., 2017)

often suffer from problems such as mode collapse (Thanh-Tung et al., 2018)

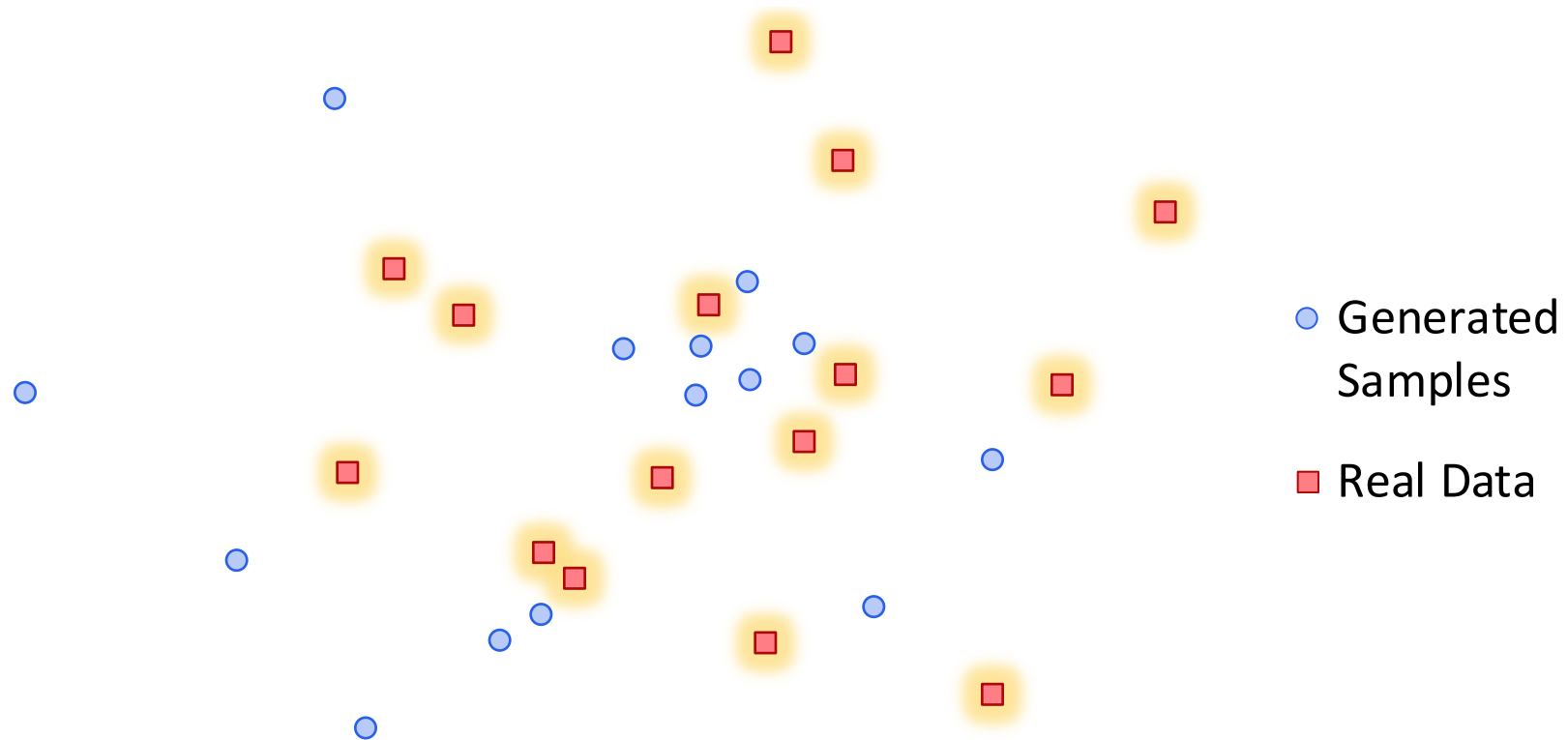
Why Mode Collapse Happens

Overcoming Mode Collapse
in Generative Adversarial Nets

Why Mode Collapse Happens

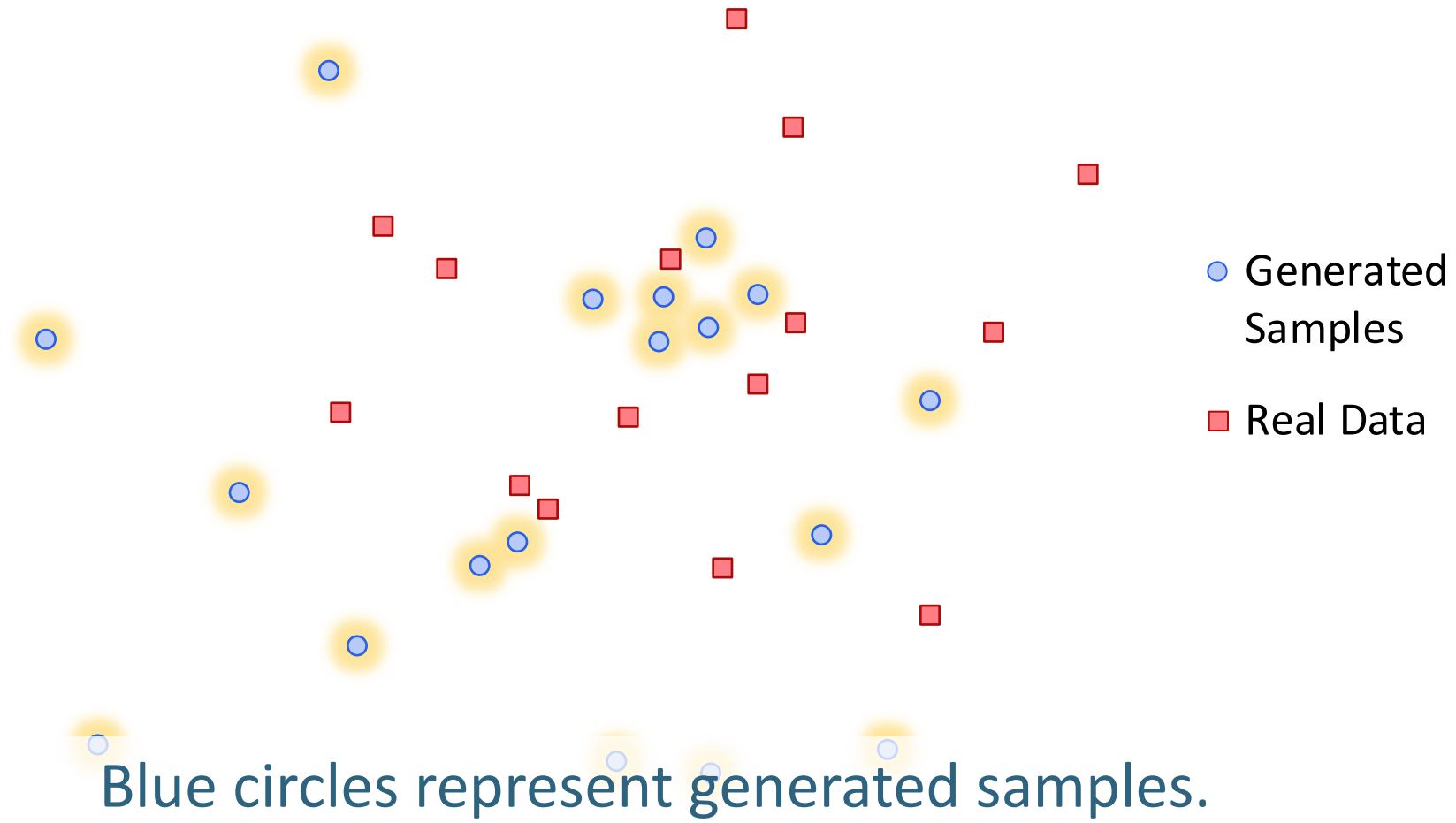


Why Mode Collapse Happens

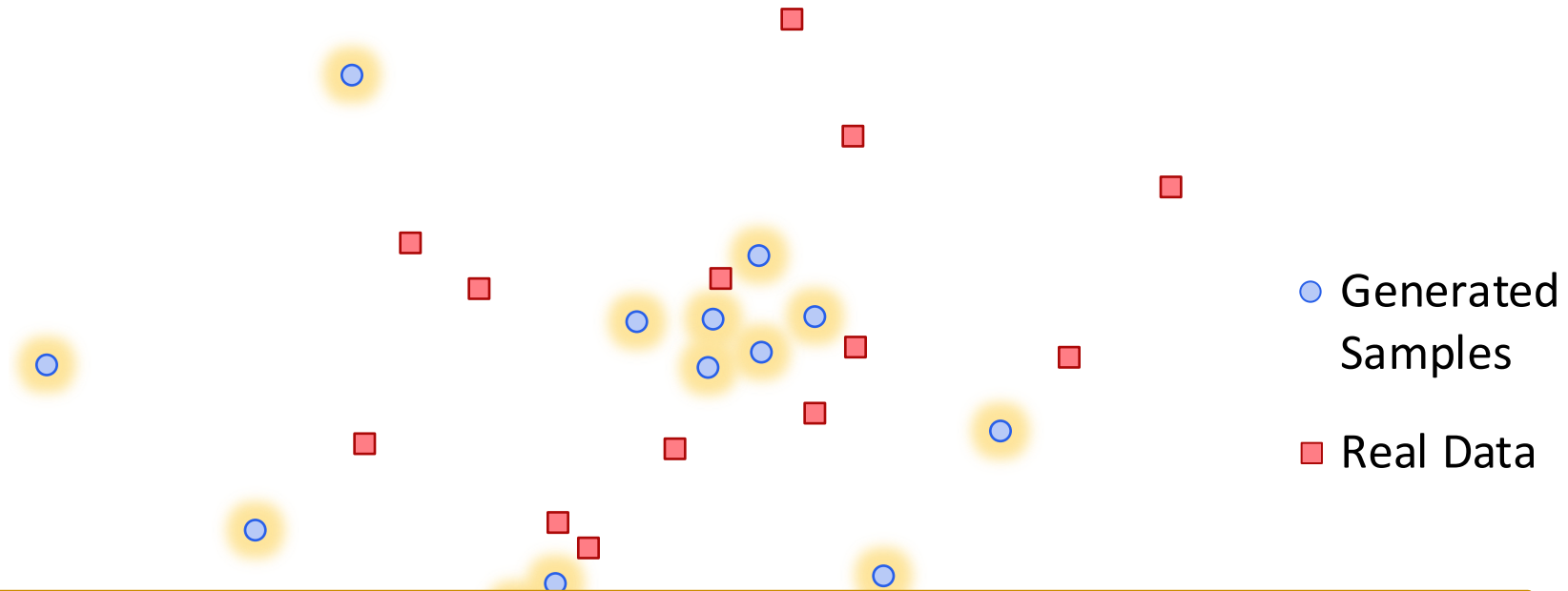


Red squares represent real data examples.

Why Mode Collapse Happens

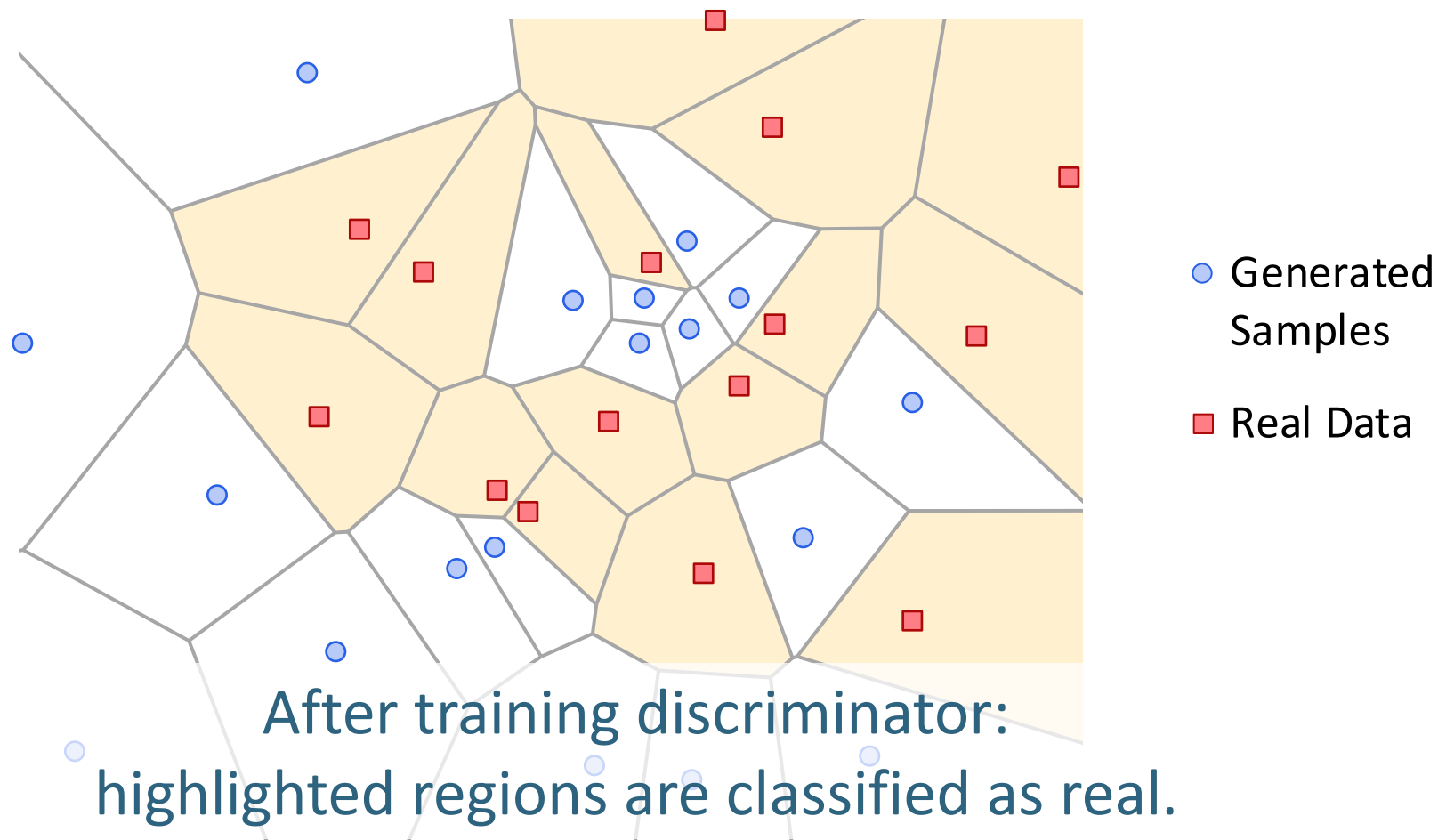


Why Mode Collapse Happens

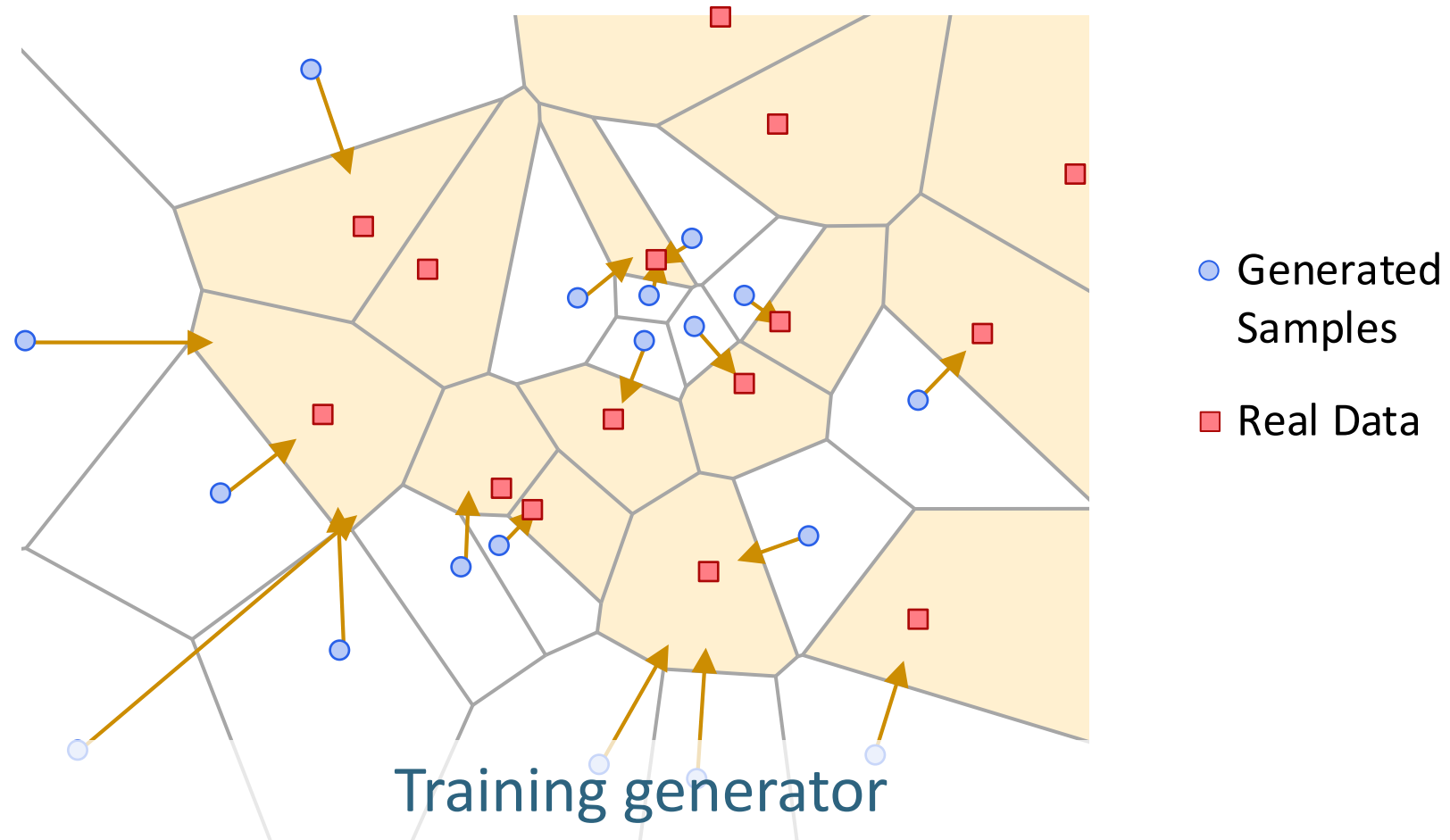


Pay attention to how the blue circles move.

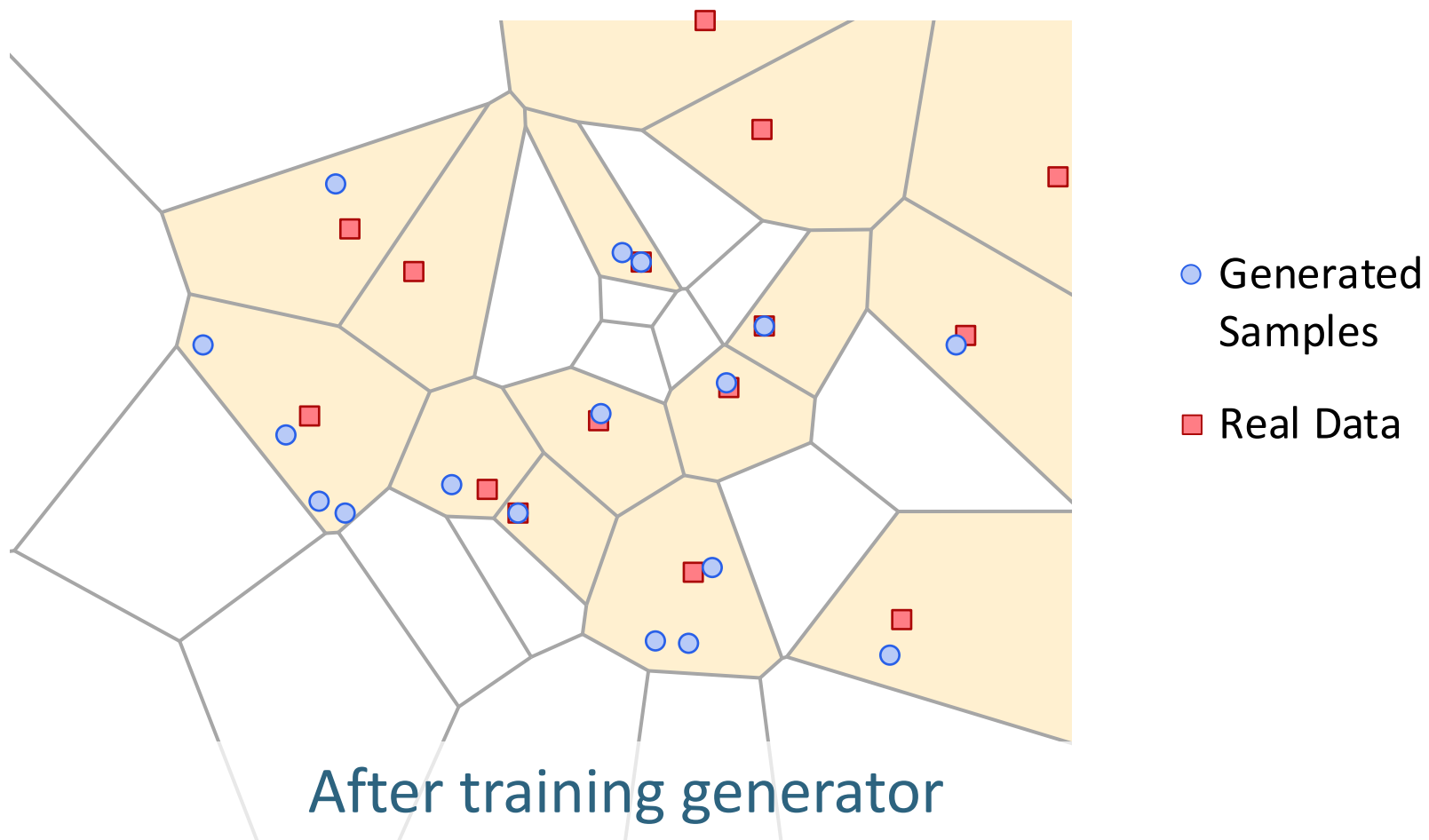
Why Mode Collapse Happens



Why Mode Collapse Happens



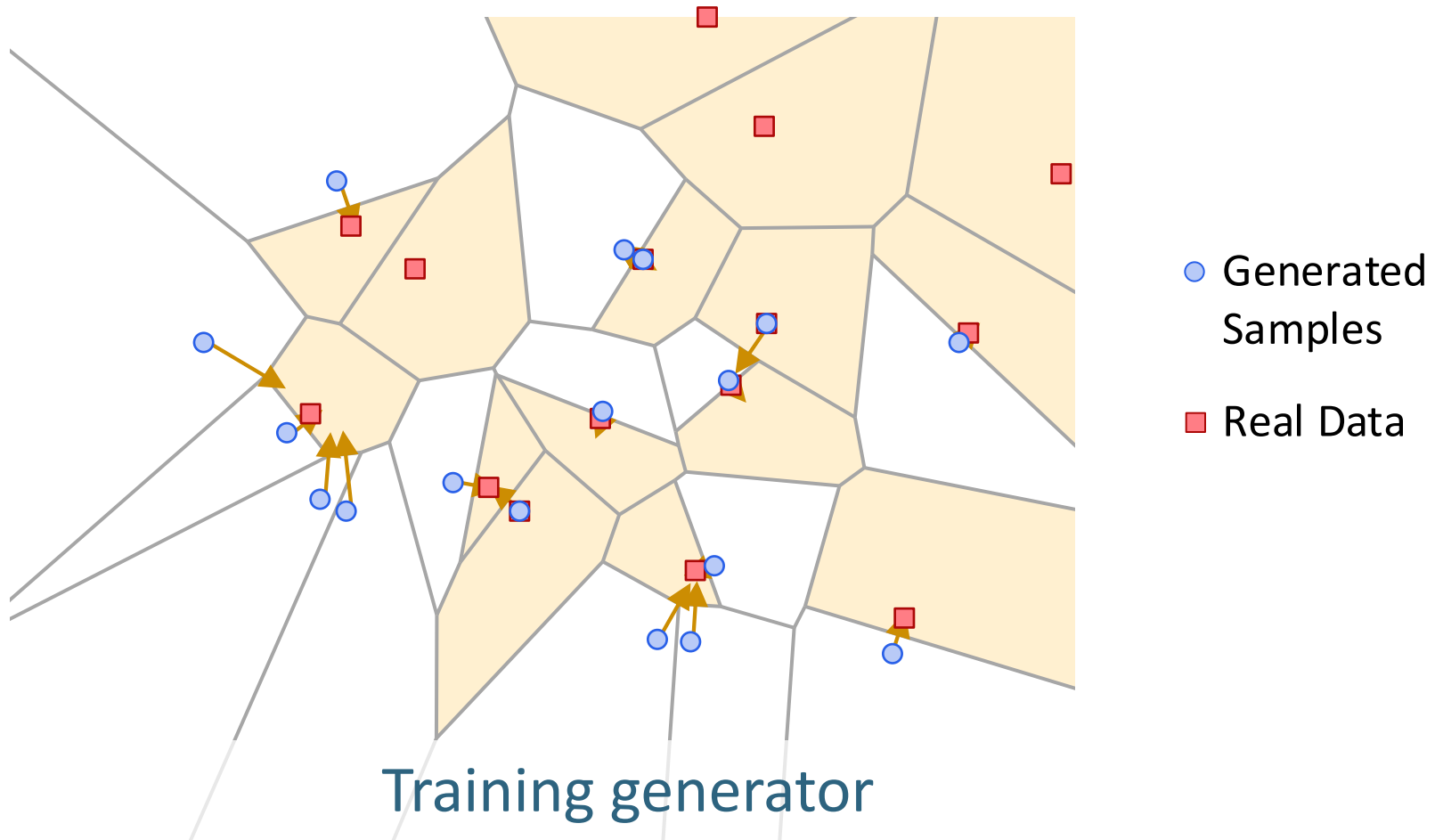
Why Mode Collapse Happens



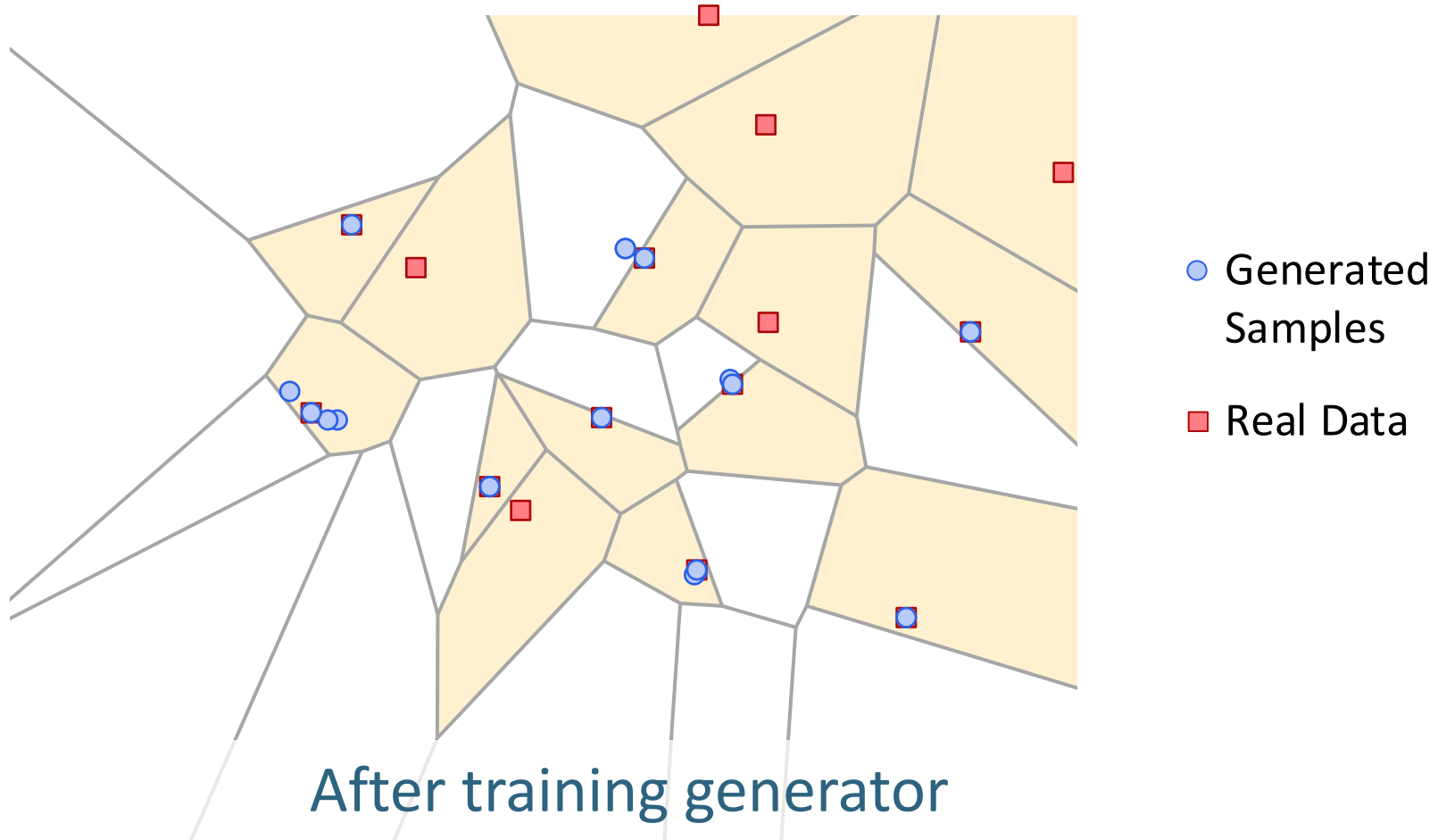
Why Mode Collapse Happens



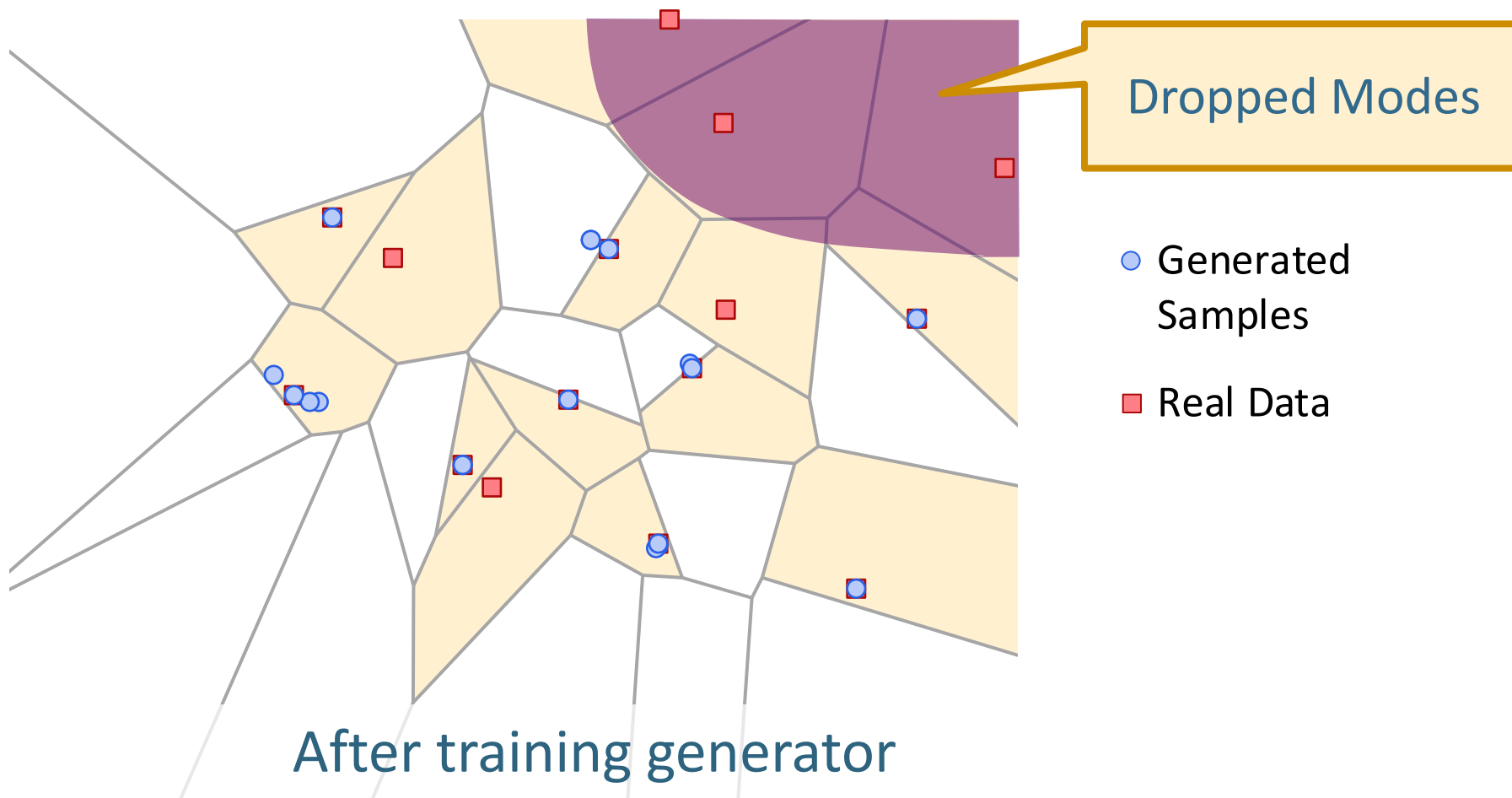
Why Mode Collapse Happens



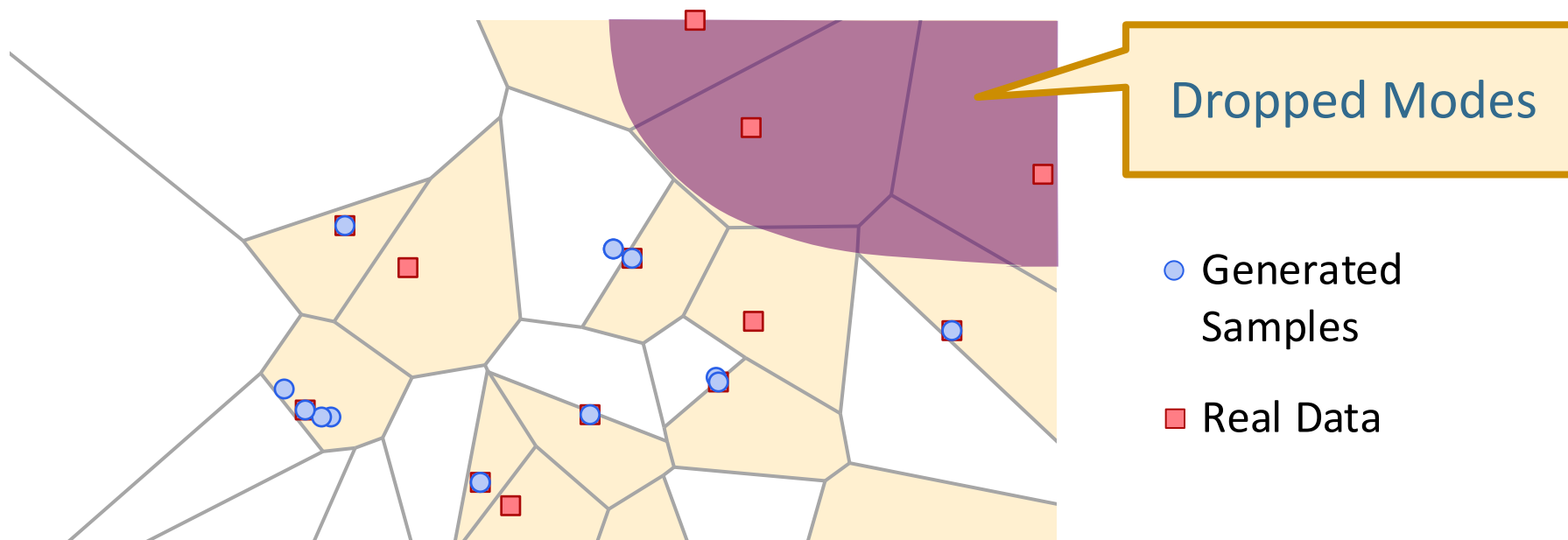
Why Mode Collapse Happens



Why Mode Collapse Happens

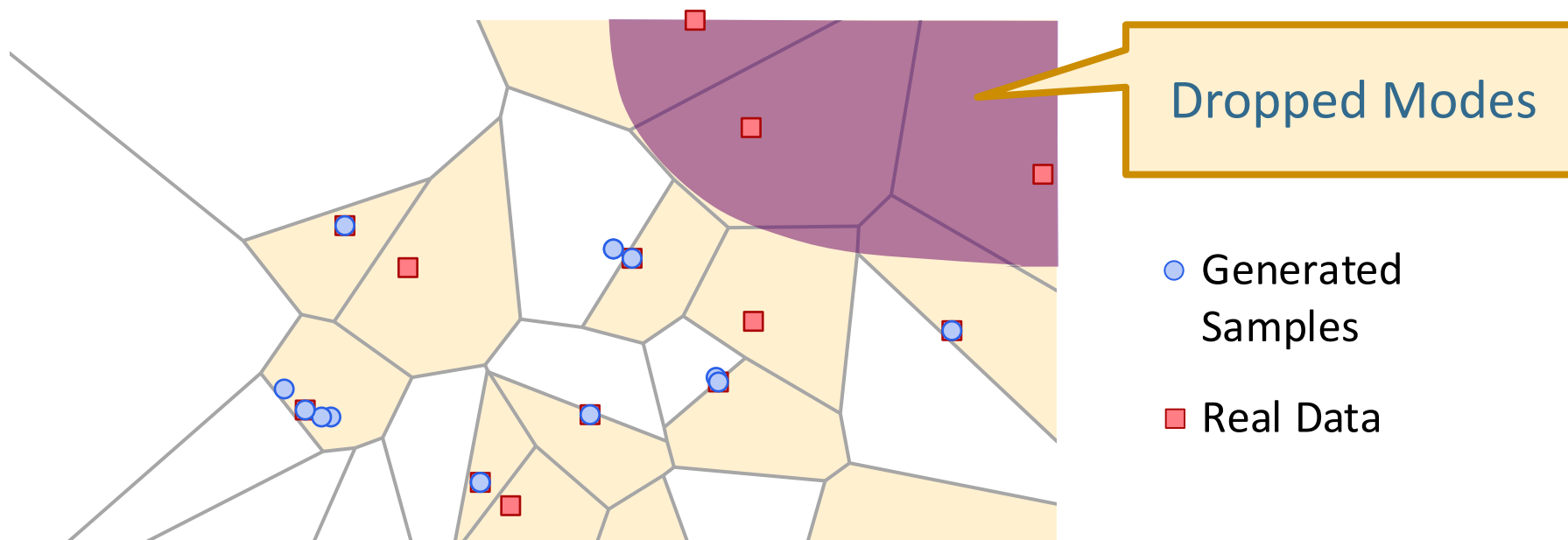


Why Mode Collapse Happens



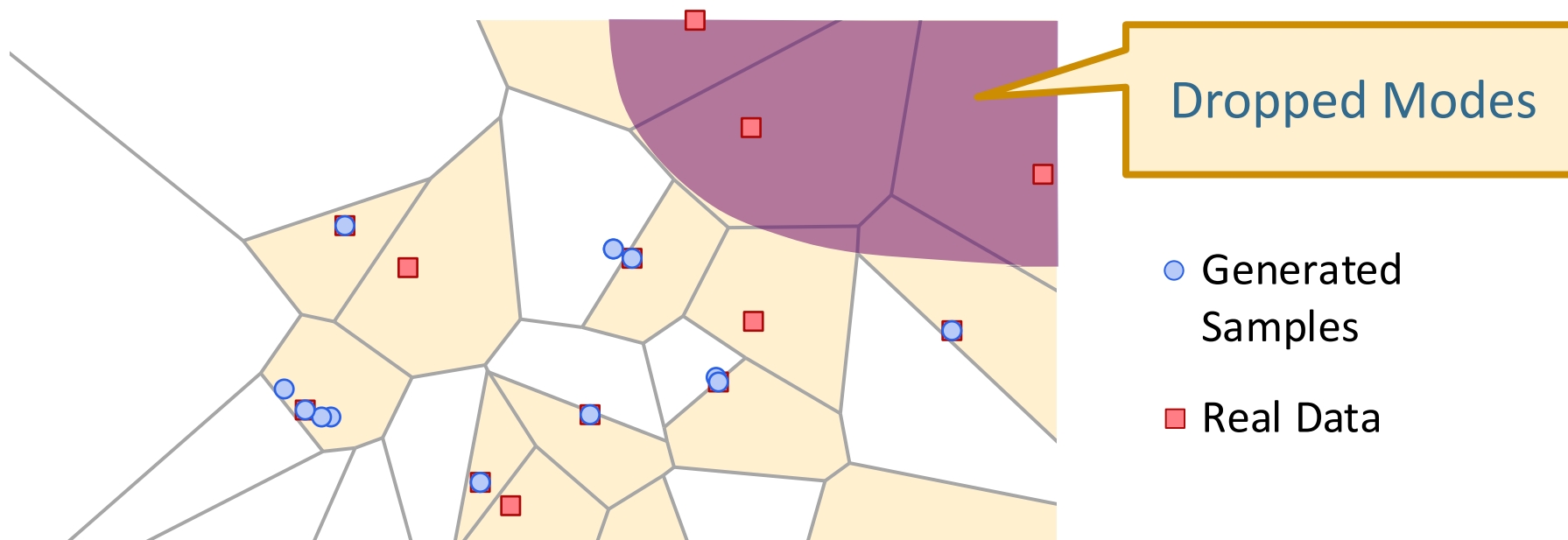
What went wrong?

Why Mode Collapse Happens



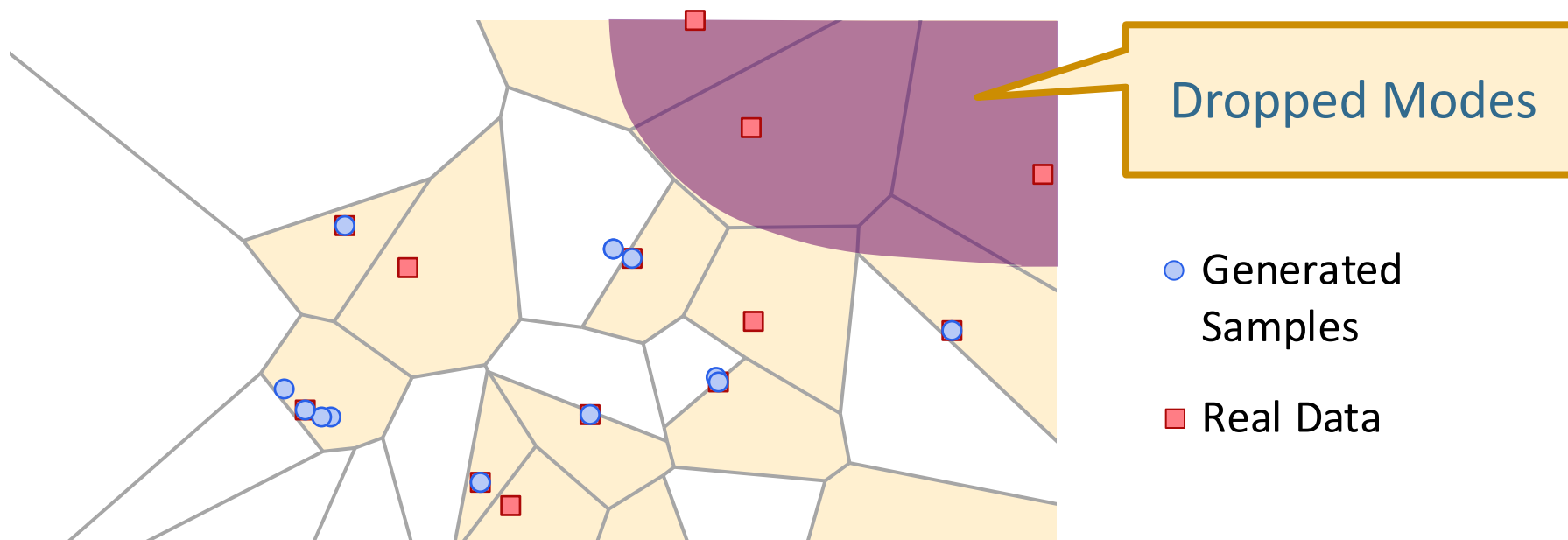
Each generated sample is essentially pushed towards the nearest data example.

Why Mode Collapse Happens



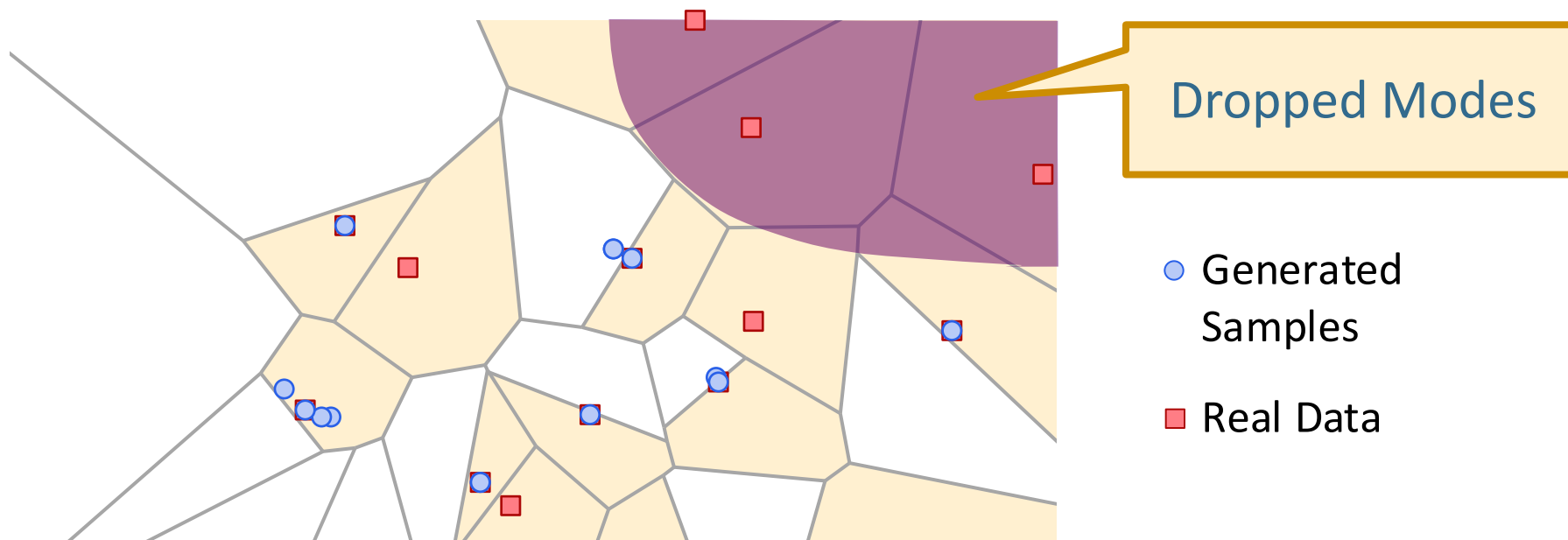
This ensures that each generated sample has a nearby data example.

Why Mode Collapse Happens



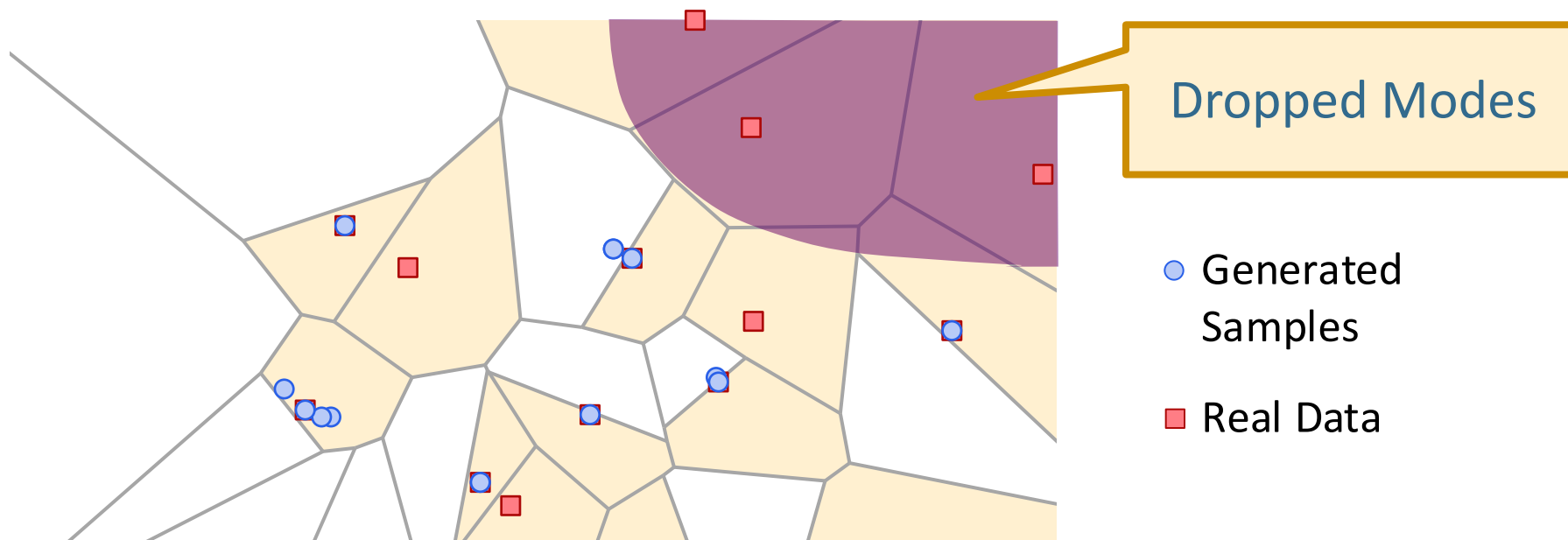
But it does *not* ensure that each data example has a nearby generated sample!

Why Mode Collapse Happens



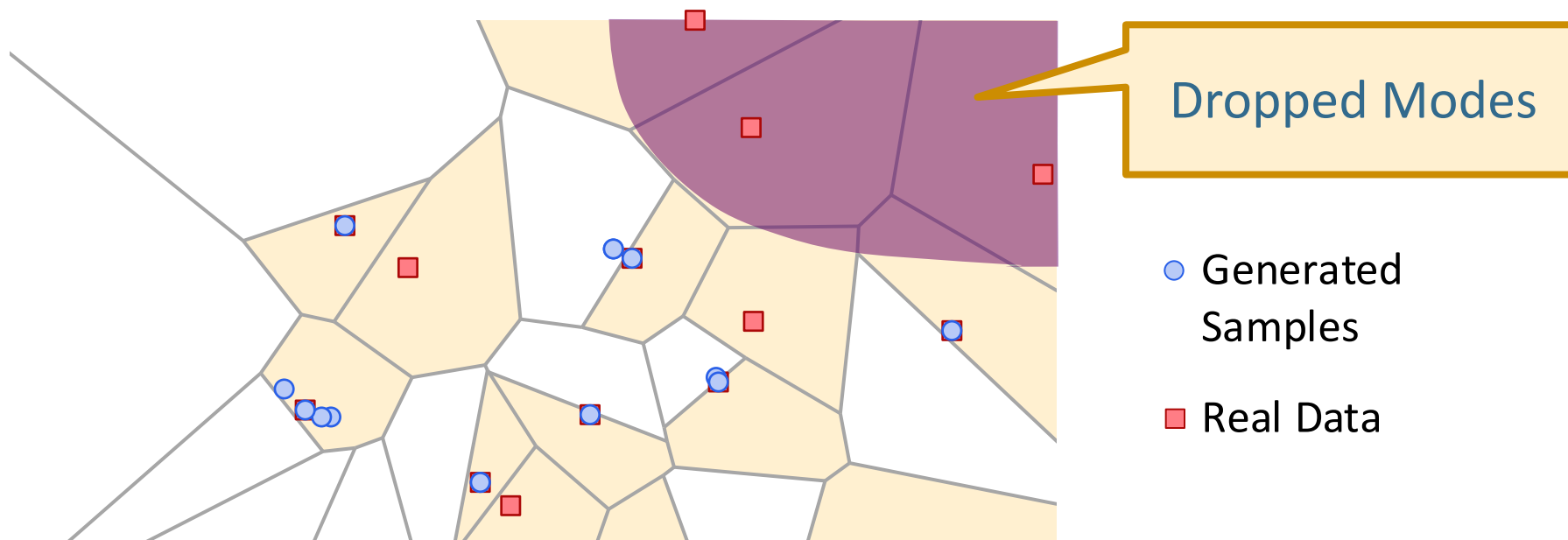
This is why modes are dropped.

Why Mode Collapse Happens



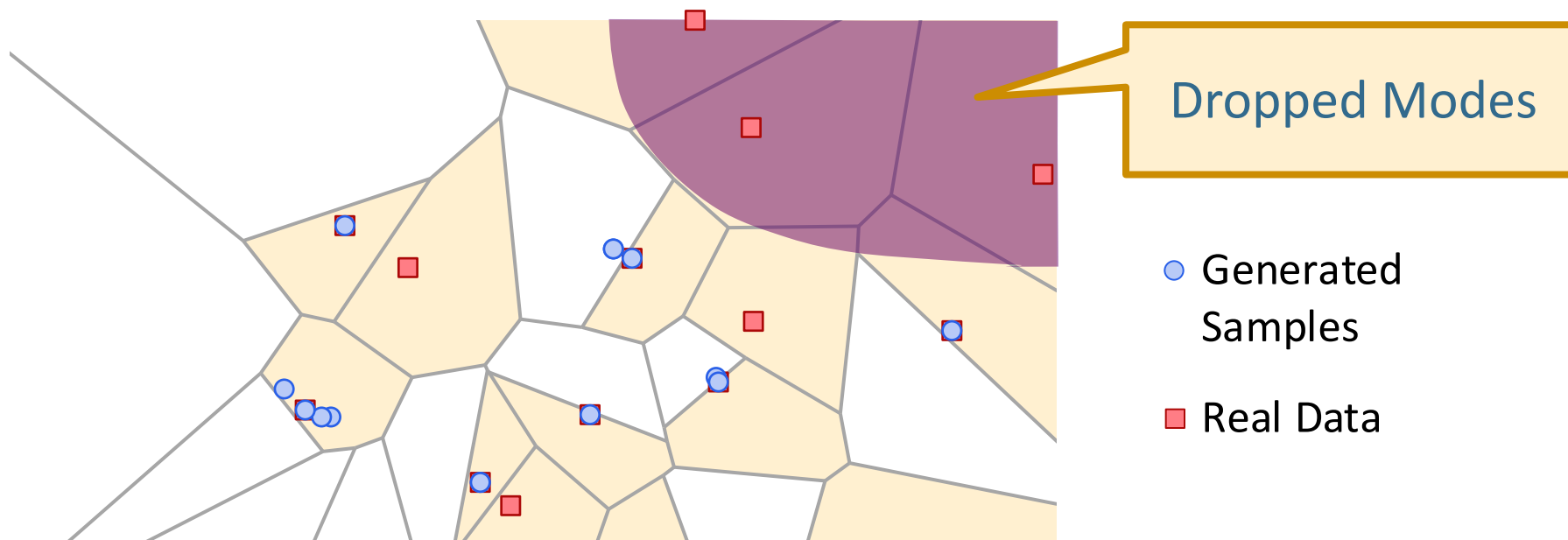
This is why modes are dropped.
How do we fix this?

Why Mode Collapse Happens



Let's do it the other way around. That is, each *data example* will now pull the nearest *generated sample* towards it.

Why Mode Collapse Happens

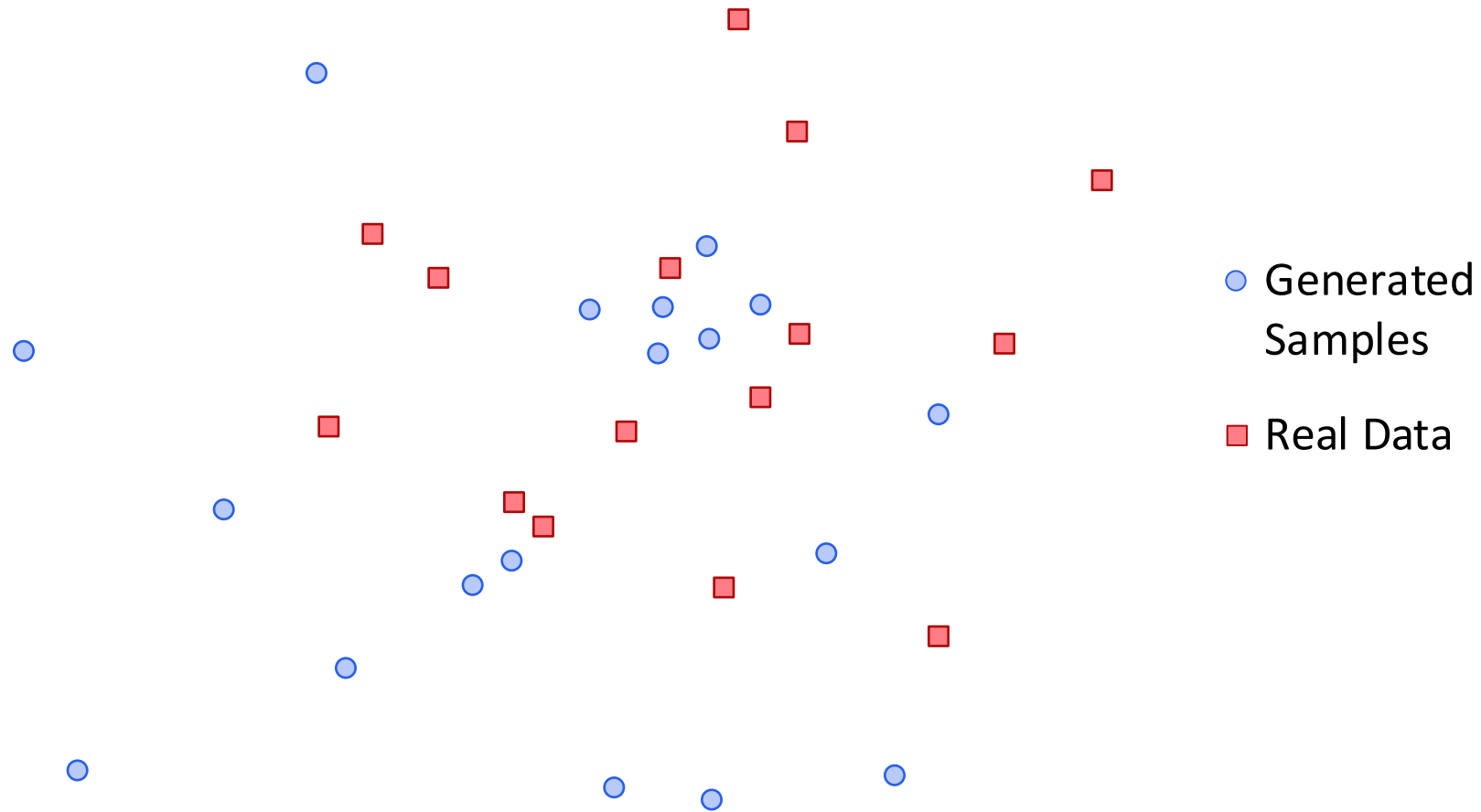


This is the idea behind Implicit Maximum Likelihood Estimation (IMLE).

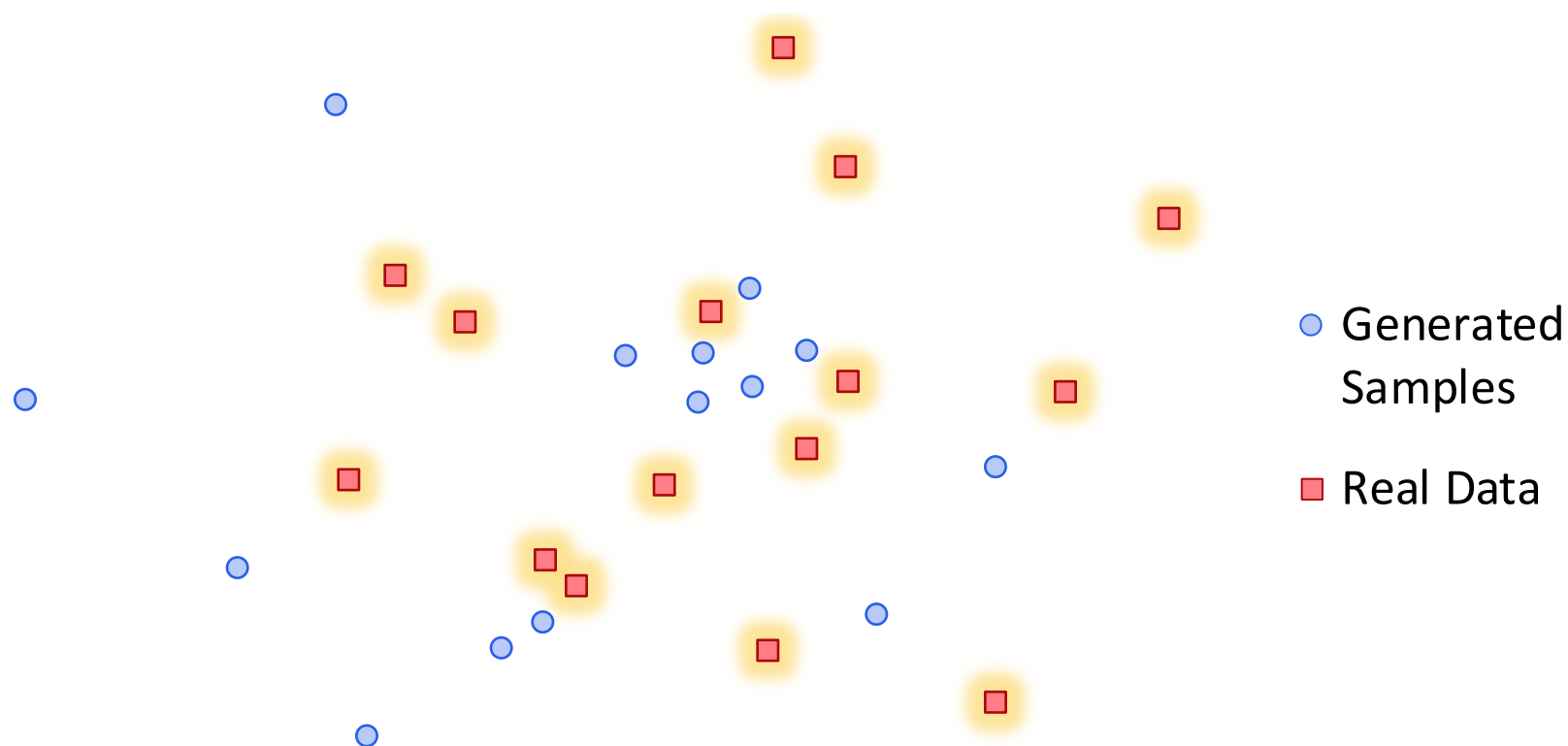
Implicit Maximum Likelihood Estimation (IMLE)

Overcoming Mode Collapse
in Generative Adversarial Nets

Implicit Maximum Likelihood Estimation (IMLE)

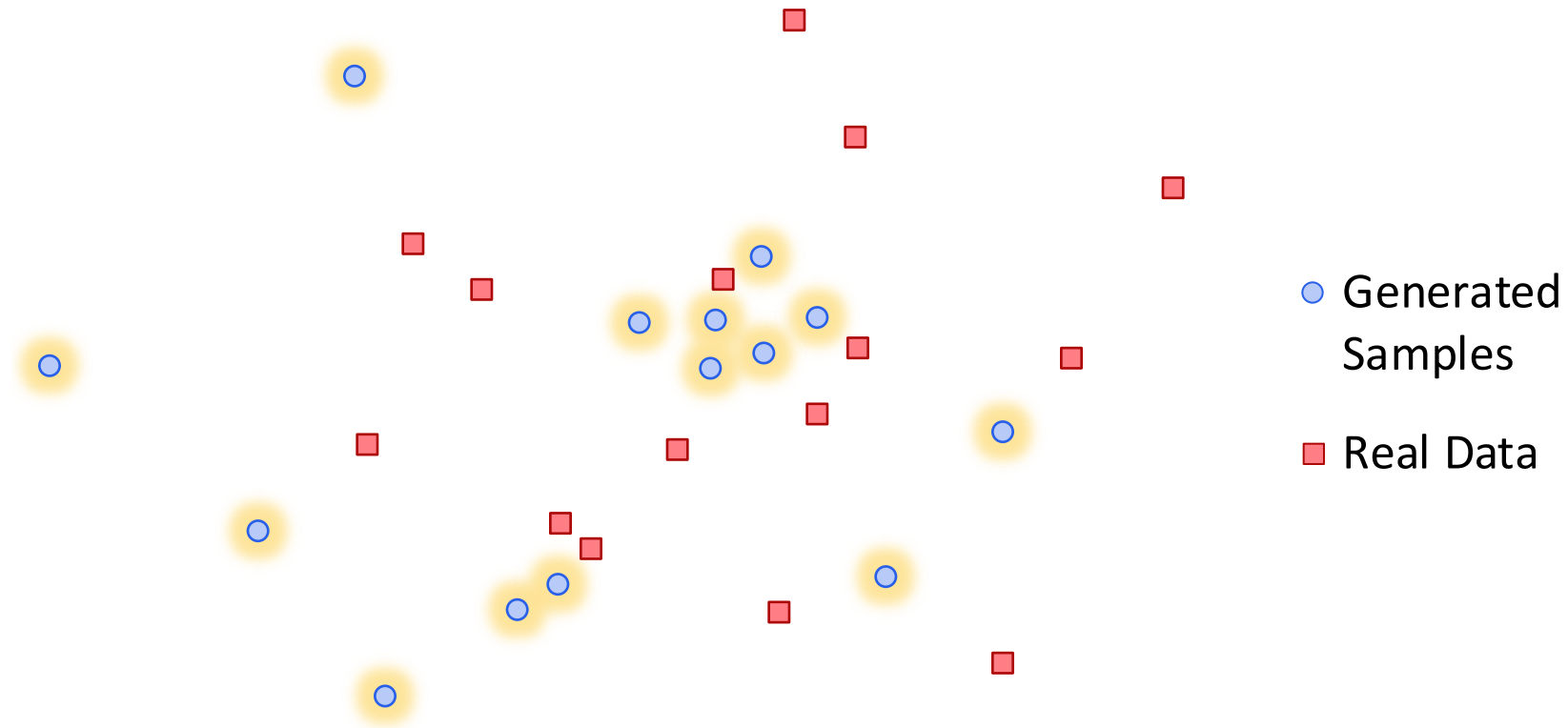


Implicit Maximum Likelihood Estimation (IMLE)



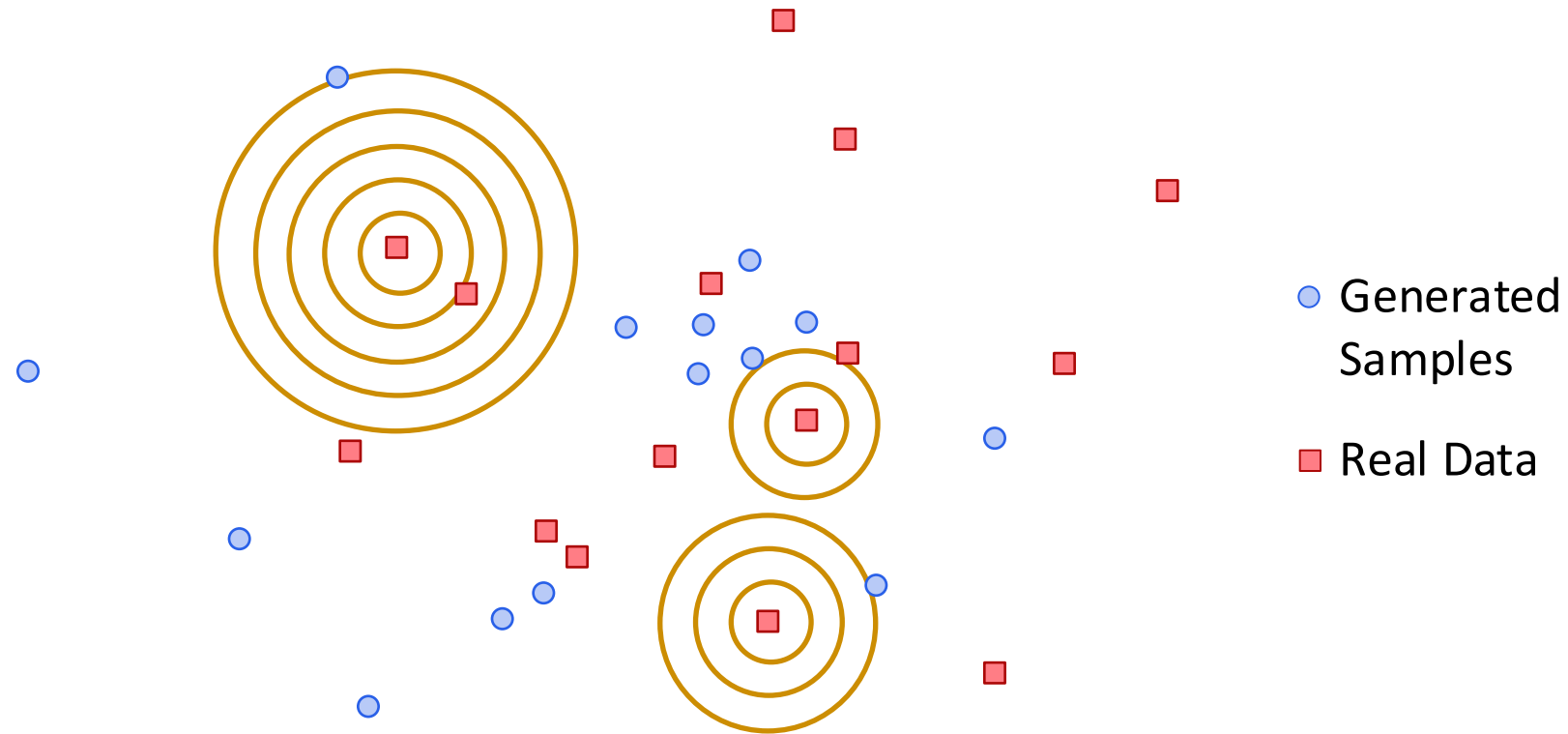
Red squares represent real data examples.

Implicit Maximum Likelihood Estimation (IMLE)



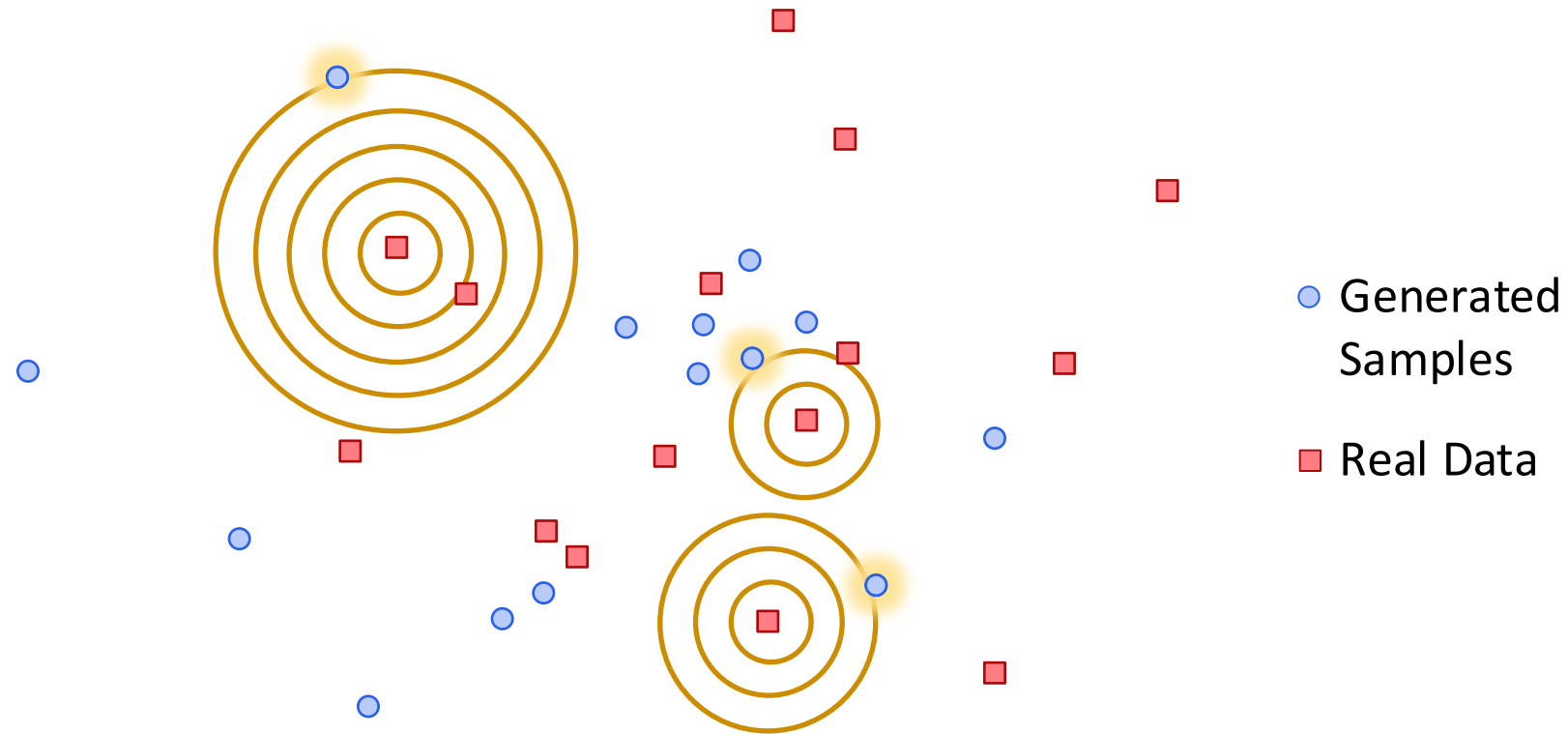
Blue circles represent generated samples.

Implicit Maximum Likelihood Estimation (IMLE)



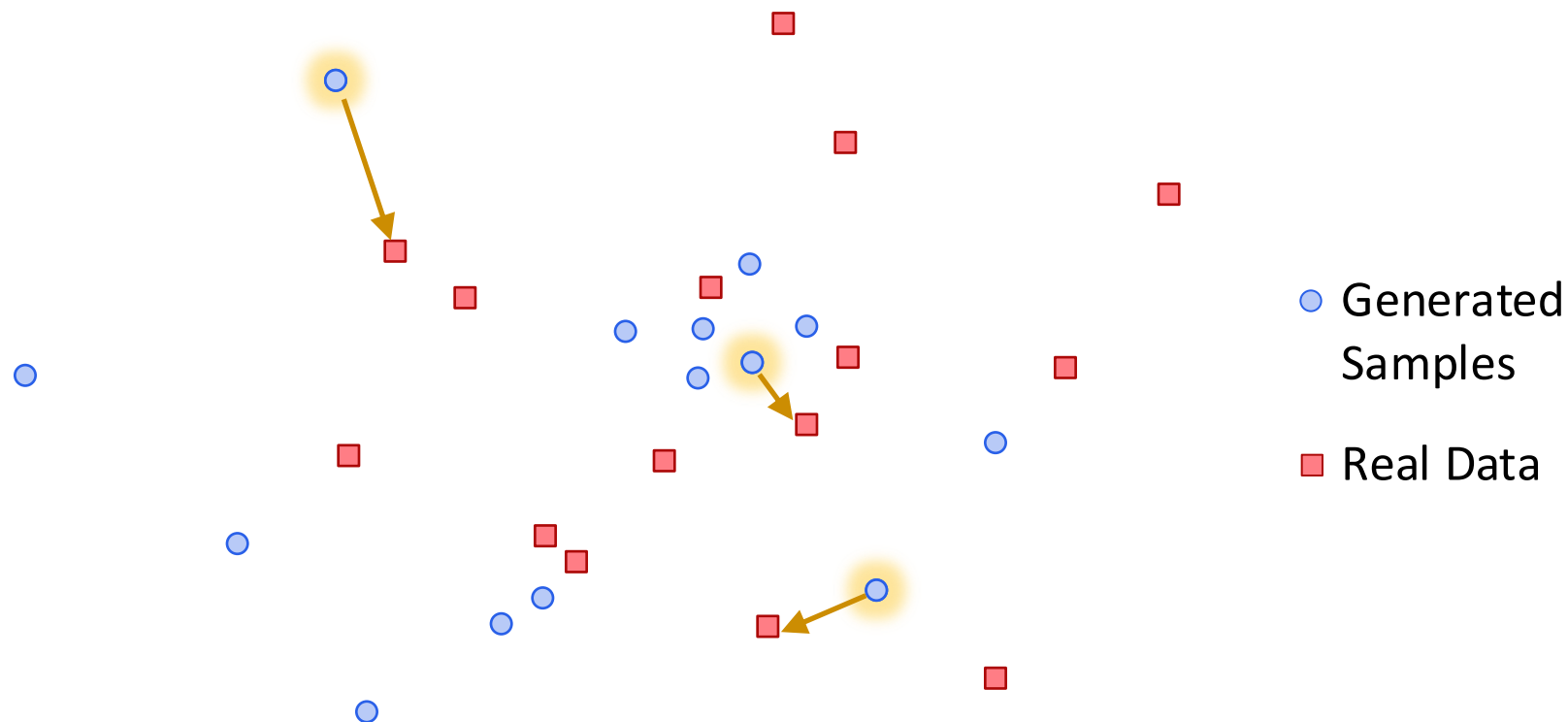
Find the nearest *sample* to each *data example*.

Implicit Maximum Likelihood Estimation (IMLE)



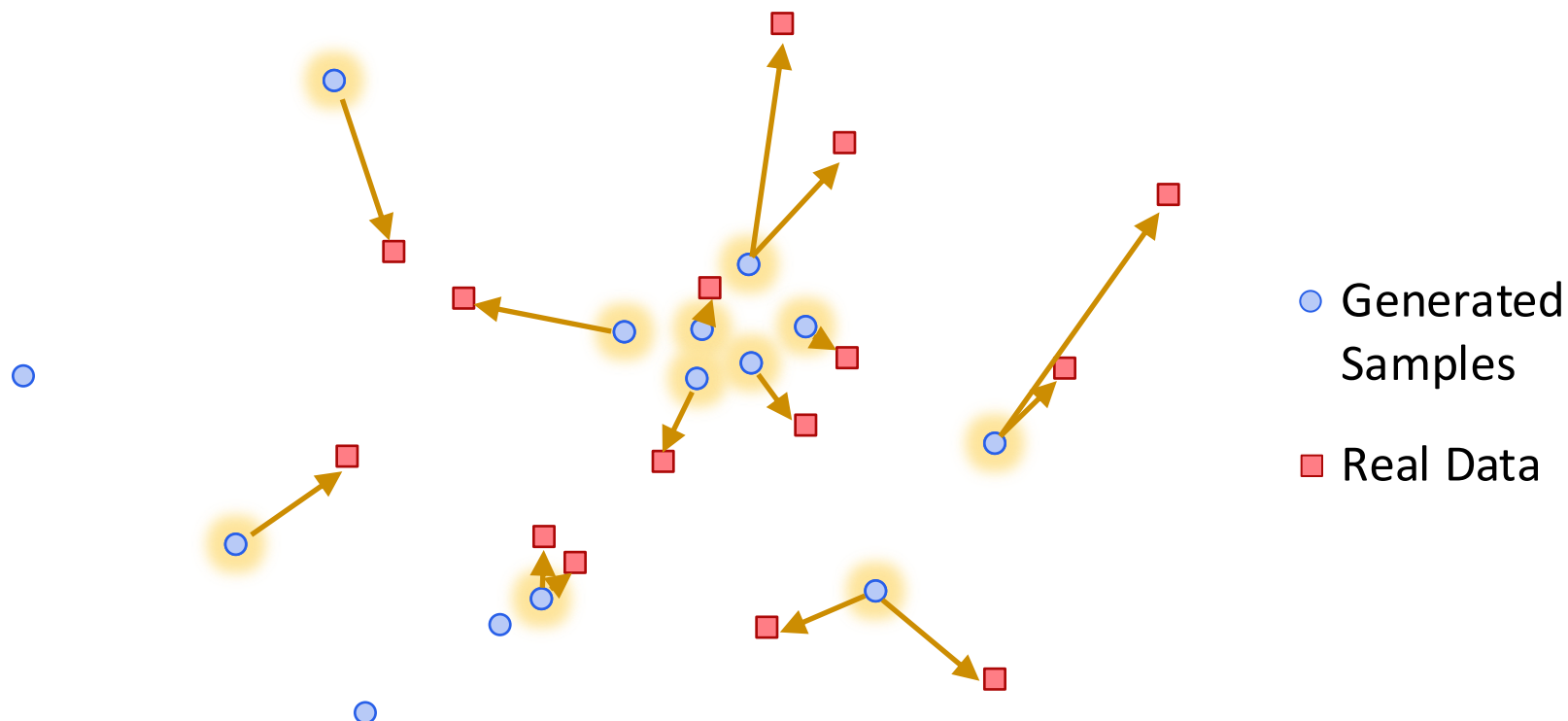
Nearest samples are found.

Implicit Maximum Likelihood Estimation (IMLE)



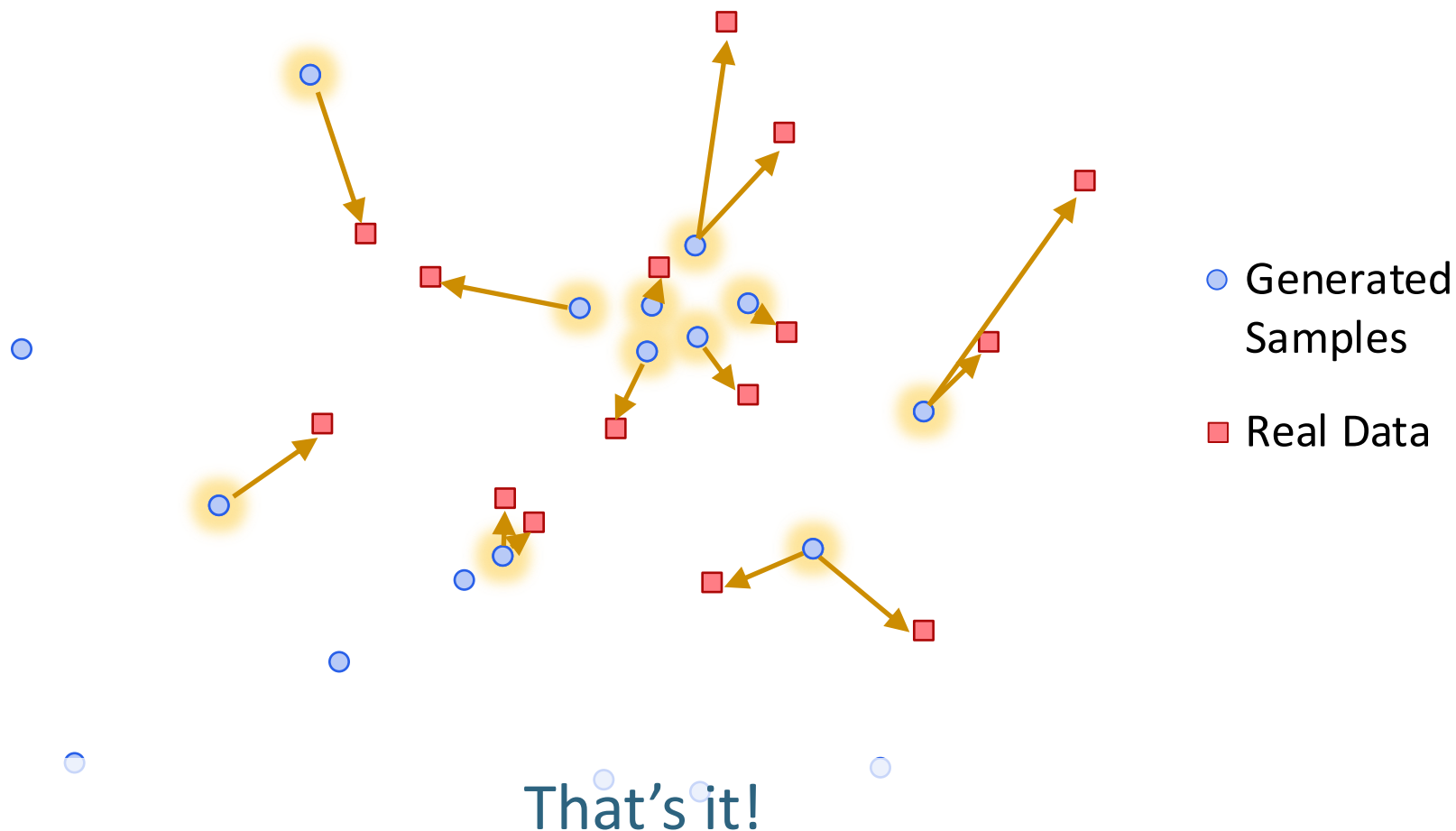
Pull sample towards data example.

Implicit Maximum Likelihood Estimation (IMLE)



Do this for all data examples.

Implicit Maximum Likelihood Estimation (IMLE)



K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv:1809.09087*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are real data examples and $d(\cdot, \cdot)$ is a distance metric

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are real data examples and d is a distance metric

Find the nearest sample to the current data example

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are real data examples and $d(\cdot, \cdot)$ is a distance metric

Involves high-dimensional nearest neighbour search, which used to be hard due to the curse of dimensionality.

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are real data examples and $d(\cdot, \cdot)$ is a distance metric

Average over all
real data examples

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are data points and $d(\cdot, \cdot)$ is a distance metric

Taking expectation over generated samples.

Implicit Maximum Likelihood Estimation (IMLE)

- Objective function:

$$\hat{\theta}_{\text{IMLE}} = \arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right]$$

where $\mathbf{y}_1, \dots, \mathbf{y}_n$ are real data examples and $d(\cdot, \cdot)$ is a distance metric

- Optimize using ordinary stochastic gradient descent (SGD):

- Draw samples $\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$

- Descend in θ : $\theta \leftarrow \theta - \gamma \nabla_{\theta} \left(\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right)$

Connection to Maximum Likelihood

Overcoming Mode Collapse
in Generative Adversarial Nets

Connection to Maximum Likelihood

- Intuition:
 - A model distribution that maximizes likelihood should have high density at each data example.
 - So, samples from the model would be more likely to lie near data examples than elsewhere.
 - Intuitively, to maximize likelihood, we can simply encourage this to happen.

Connection to Maximum Likelihood

- *Theorem (Informal):*

If the model is richly parameterized and the model density $p(\hat{\mathbf{y}}; \theta)$ is smooth in θ and $\hat{\mathbf{y}}$,

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right] = \arg \max_{\theta} \sum_{i=1}^n w_i \log p(\mathbf{y}_i; \theta)$$

where $0 \leq w_i \leq 1 \ \forall i$

Connection to Maximum Likelihood

- Theorem (Informal):*

If the model is richly parameterized and the model density $p(\hat{\mathbf{y}}; \theta)$ is smooth in θ and $\hat{\mathbf{y}}$,

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right] = \arg \max_{\theta} \sum_{i=1}^n w_i \log p(\mathbf{y}_i; \theta)$$

where $0 \leq w_i \leq 1 \ \forall i$

Lower bound on
the log-likelihood

Connection to Maximum Likelihood

- Theorem (Informal):*

If the model is richly parameterized and the model density $p(\hat{\mathbf{y}}; \theta)$ is smooth in θ and $\hat{\mathbf{y}}$,

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right] = \arg \max_{\theta} \sum_{i=1}^n w_i \log p(\mathbf{y}_i; \theta)$$

where $0 \leq w_i \leq 1 \ \forall i$

Tightness depends on the gap between $\min_{i \in [n]} \{p_{\theta^*}(\mathbf{y}_i)\}$ and $\max_{i \in [n]} \{p_{\theta^*}(\mathbf{y}_i)\}$.

Connection to Maximum Likelihood

- Theorem (Informal):*

If the model is richly parameterized and the model density $p(\hat{\mathbf{y}}; \theta)$ is smooth in θ and $\hat{\mathbf{y}}$,

$$\arg \min_{\theta} \mathbb{E}_{\mathbf{z}_1, \dots, \mathbf{z}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{1}{n} \sum_{i=1}^n \min_{j \in \{1, \dots, m\}} d(f_{\theta}(\mathbf{z}_j), \mathbf{y}_i) \right] = \arg \max_{\theta} \sum_{i=1}^n w_i \log p(\mathbf{y}_i; \theta)$$

where $0 \leq w_i \leq 1 \ \forall i$

The more expressive the model, the tighter this bound becomes.

Other Problems of GANs

Overcoming Mode Collapse
in Generative Adversarial Nets

Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.

Problems of GANs

Mode Collapse

- Some/most modes of the true data distribution can be ignored.
- Does *not* learn the true distribution.

Vanishing Gradients

- Gradient becomes near zero when the discriminator is powerful.

Training Instability

- Adversarial formulation – requires minimax optimization, which can only be done heuristically.

IMLE Overcomes These Problems

~~Mode Collapse~~

- Each data example has a nearby sample.

~~Vanishing
Gradients~~

- Gradient only becomes zero when distance to nearest sample is zero.

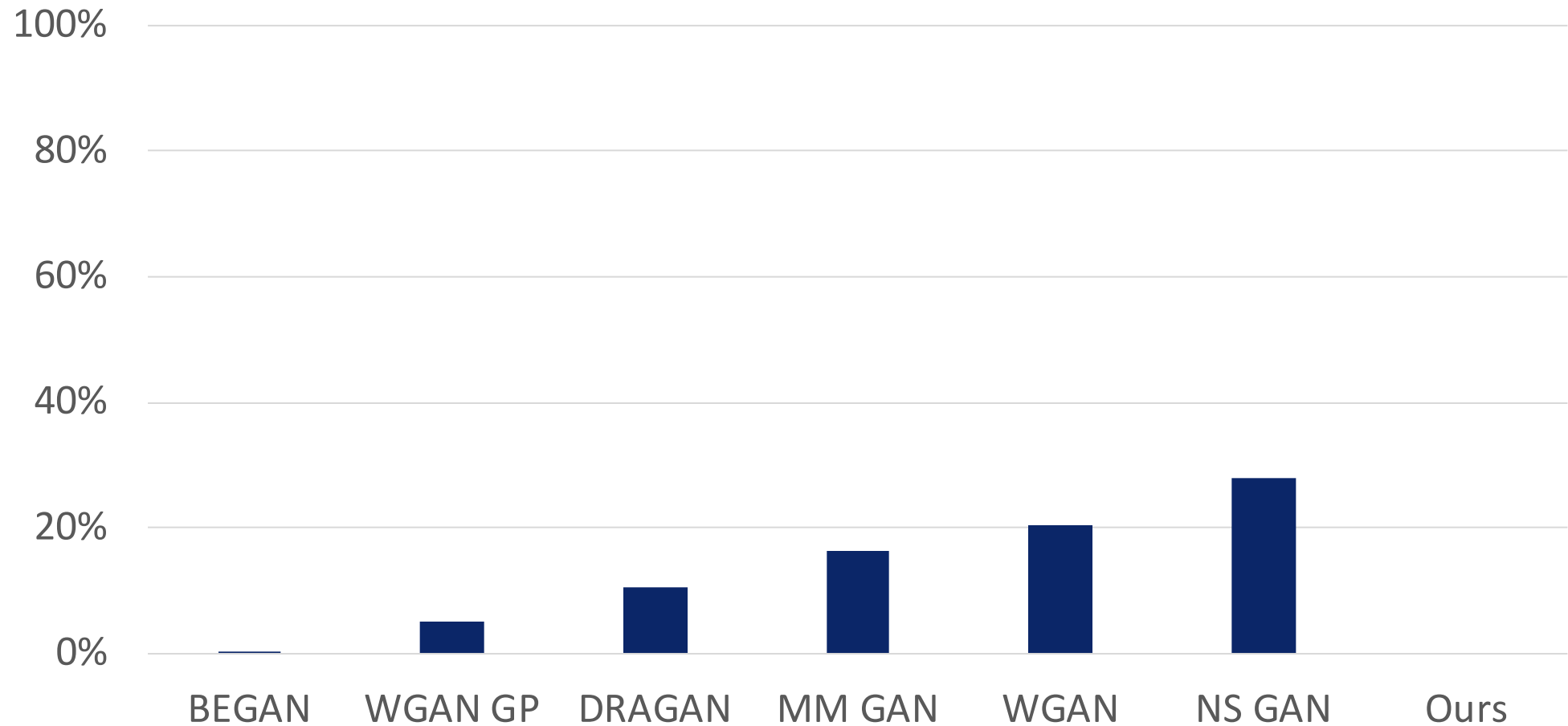
~~Training
Instability~~

- Simple minimization problem – no adversarial formulation.

Empirical Results

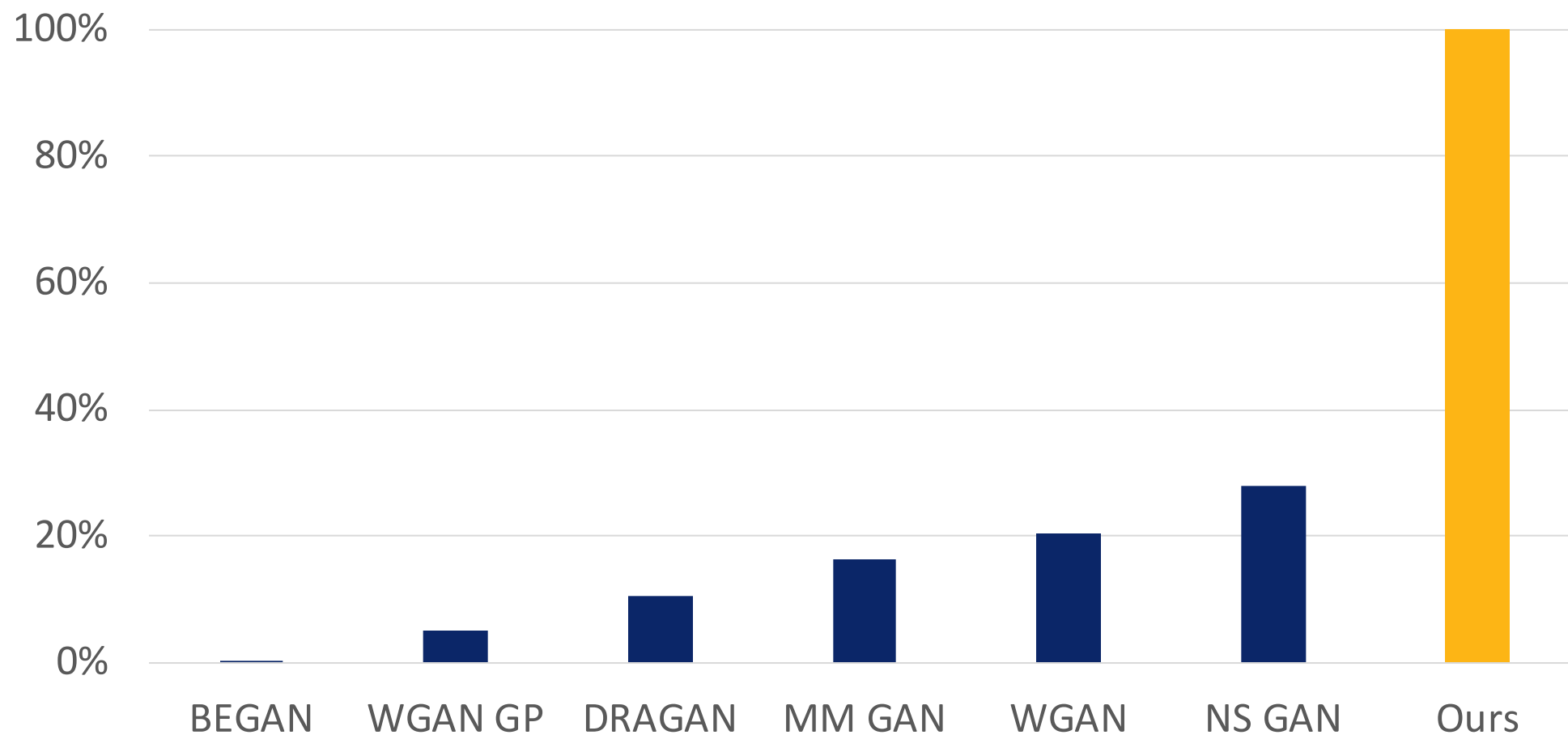
Overcoming Mode Collapse
in Generative Adversarial Nets

Recall



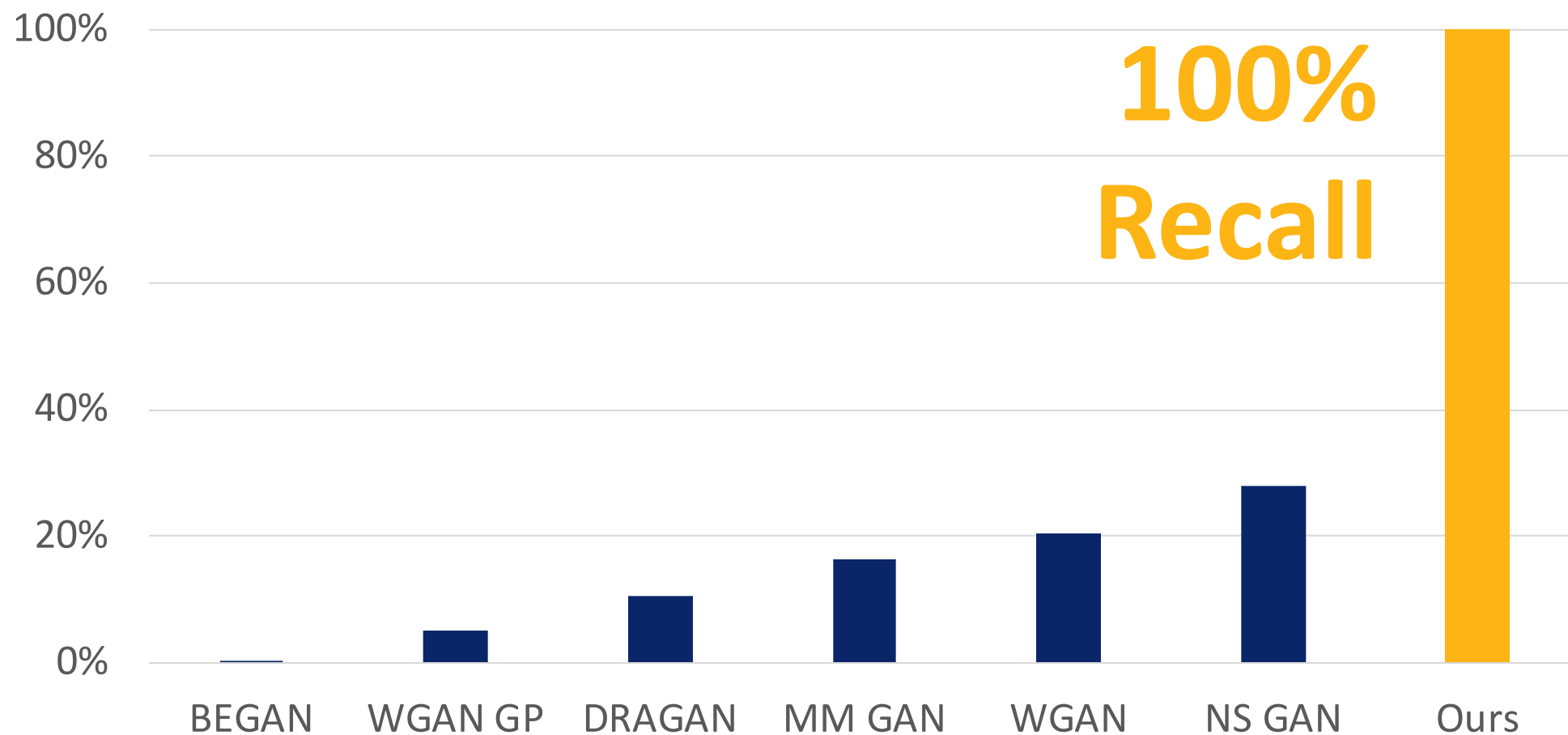
M. Lucic et al. Are GANs Created Equal? A Large-Scale Study. *NeurIPS*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

Recall



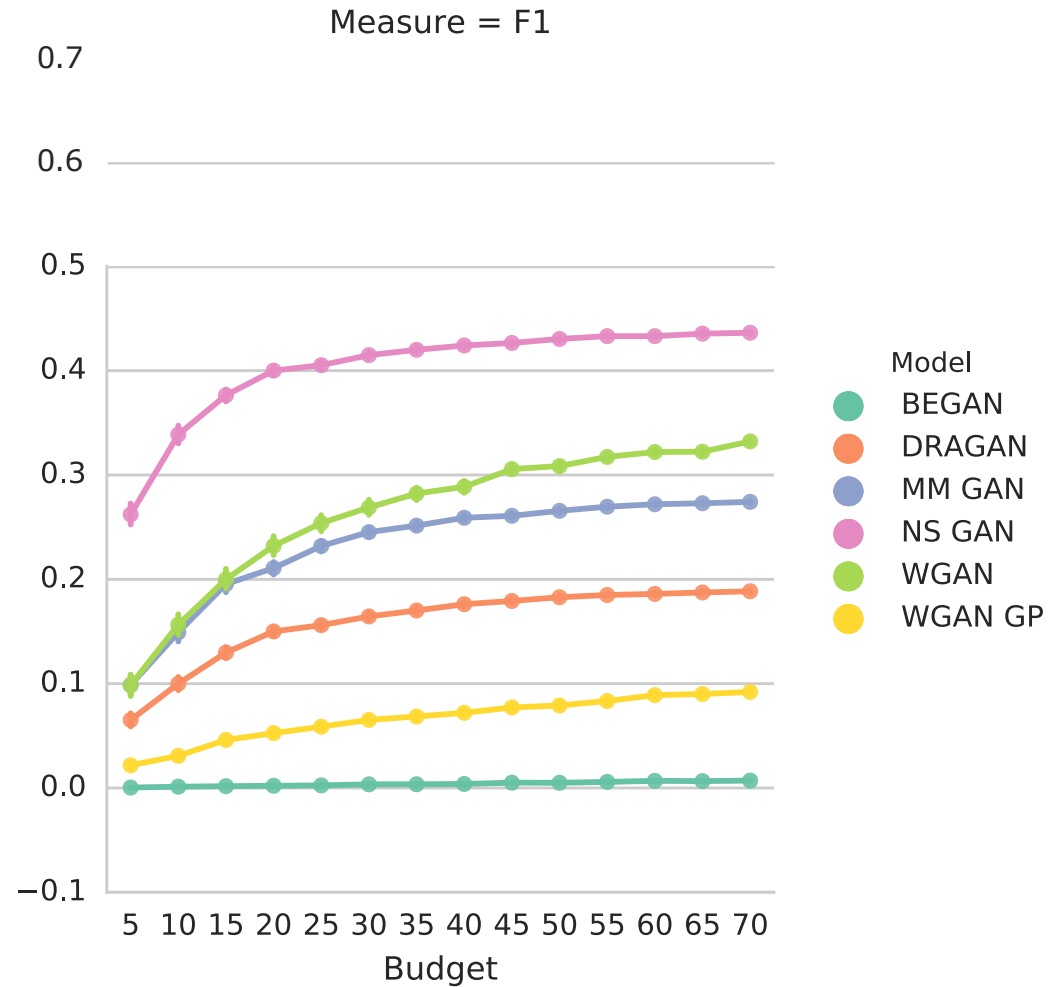
K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv:1809.09087*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

Recall



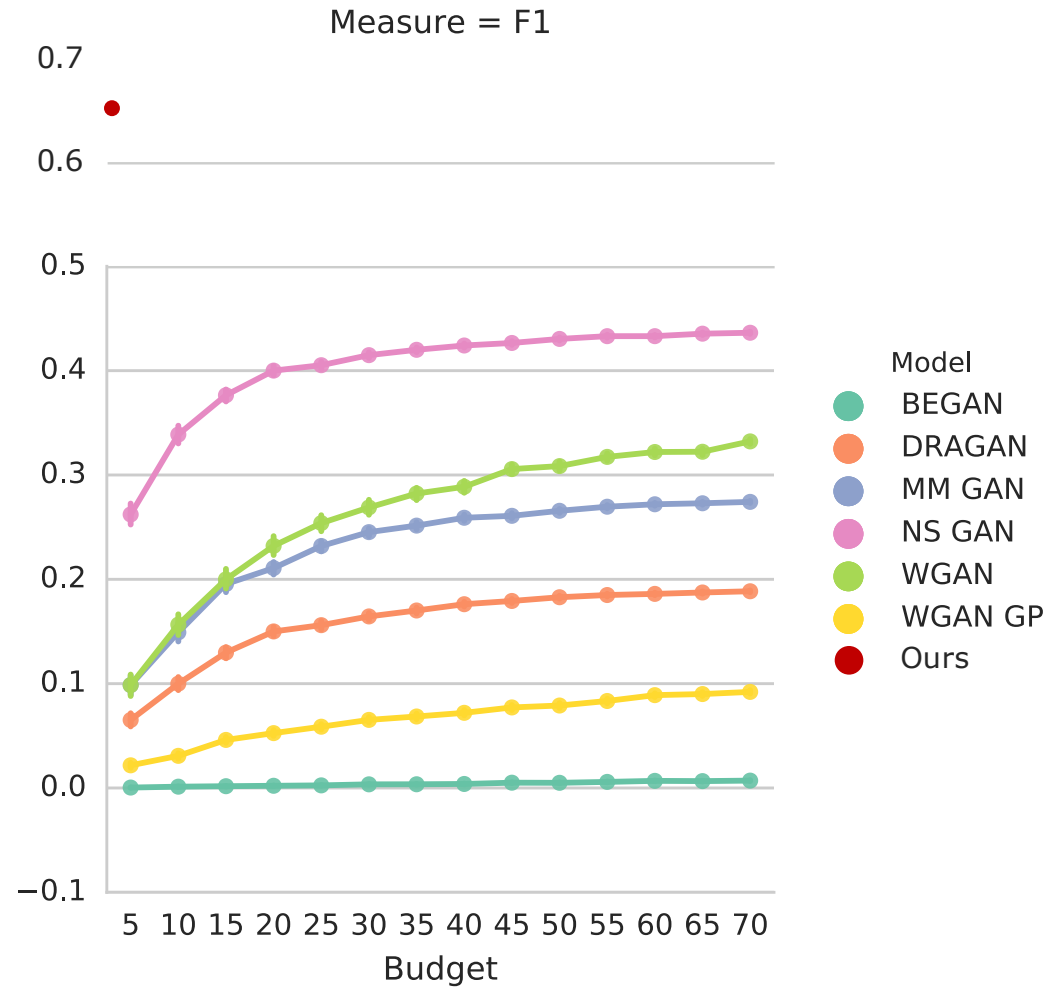
K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv:1809.09087*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

F-Measure



M. Lucic et al. Are GANs Created Equal? A Large-Scale Study. *NeurIPS*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

F-Measure



K. Li and J. Malik. Implicit Maximum Likelihood Estimation. *arXiv:1809.09087*, 2018
Overcoming Mode Collapse
in Generative Adversarial Nets

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

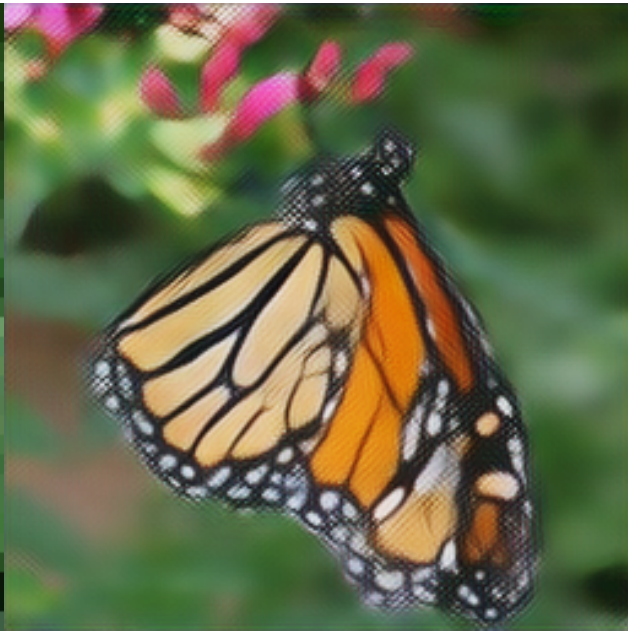
Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

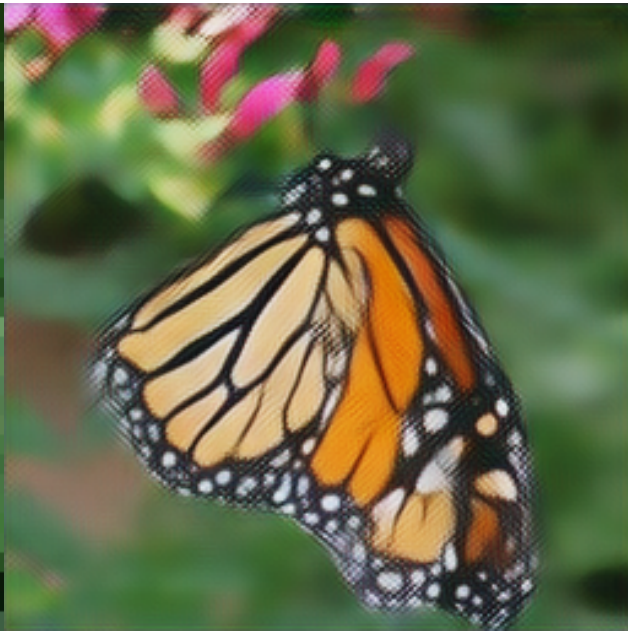
Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

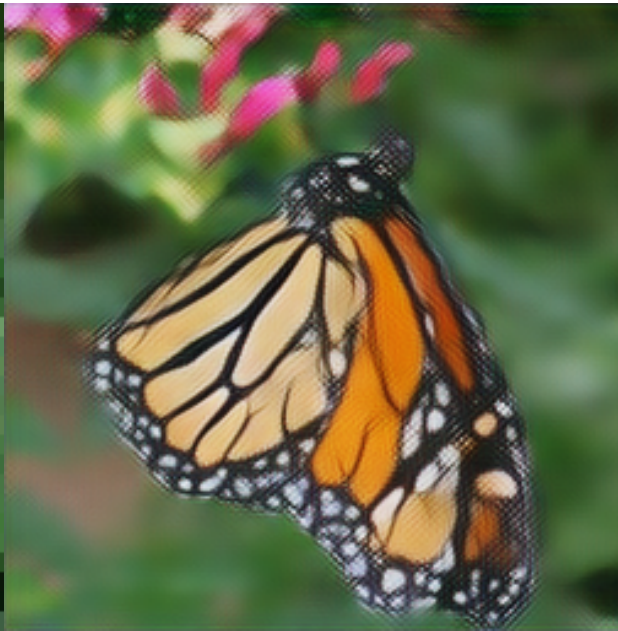
Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

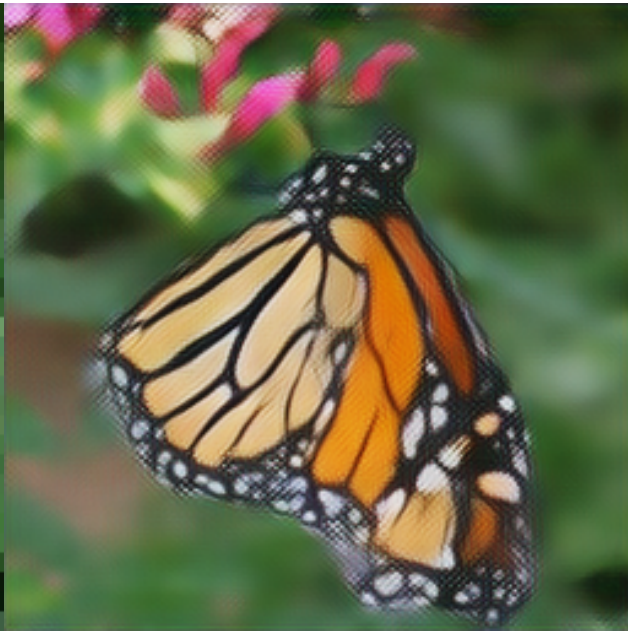
Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

Input



IMLE Samples



SRGAN Samples



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Super-Resolution

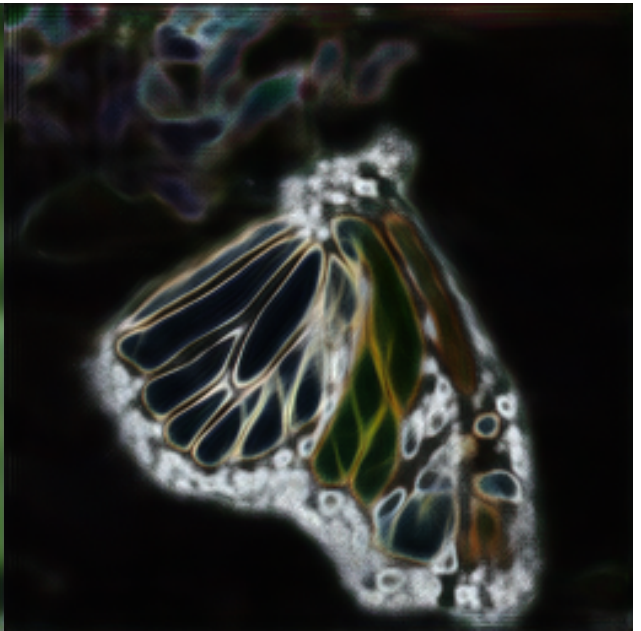
Samples



Mean



Standard Deviation



K. Li*, S. Peng* and J. Malik. Super-Resolution via Conditional Implicit Maximum Likelihood Estimation. *arXiv:1810.01406*, 2018

Overcoming Mode Collapse
in Generative Adversarial Nets



UNIVERSITY OF
TORONTO

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples

K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples

K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Input



IMLE
Samples



Pix2pixHD
Samples



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Interpolation in Latent Space $\mathbf{z}$



(The figure above doesn't play in this PDF version of the slides. See Figure 8 at https://people.eecs.berkeley.edu/~ke.li/projects/imle/scene_layouts/ instead.)

K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets

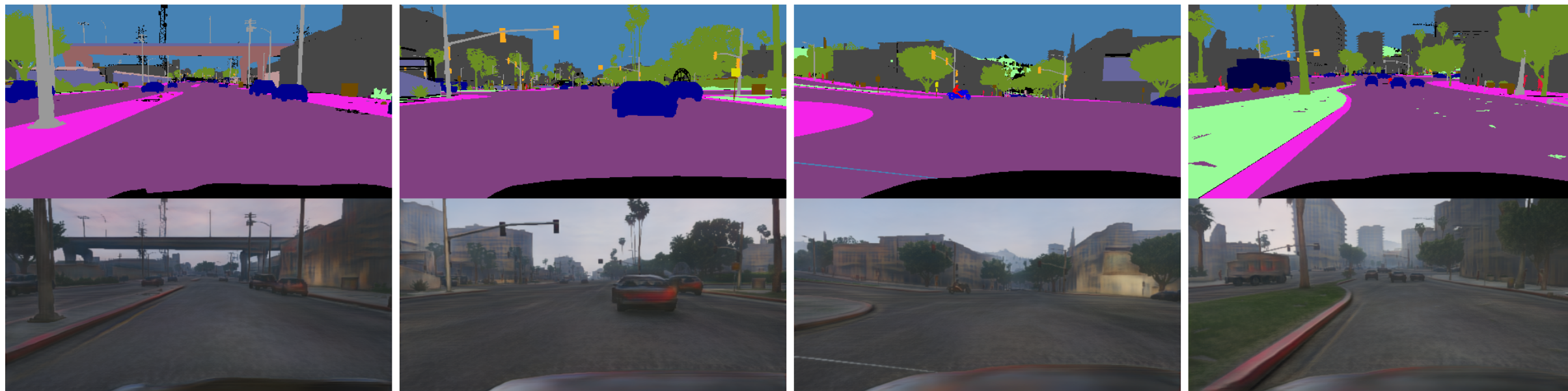


南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets

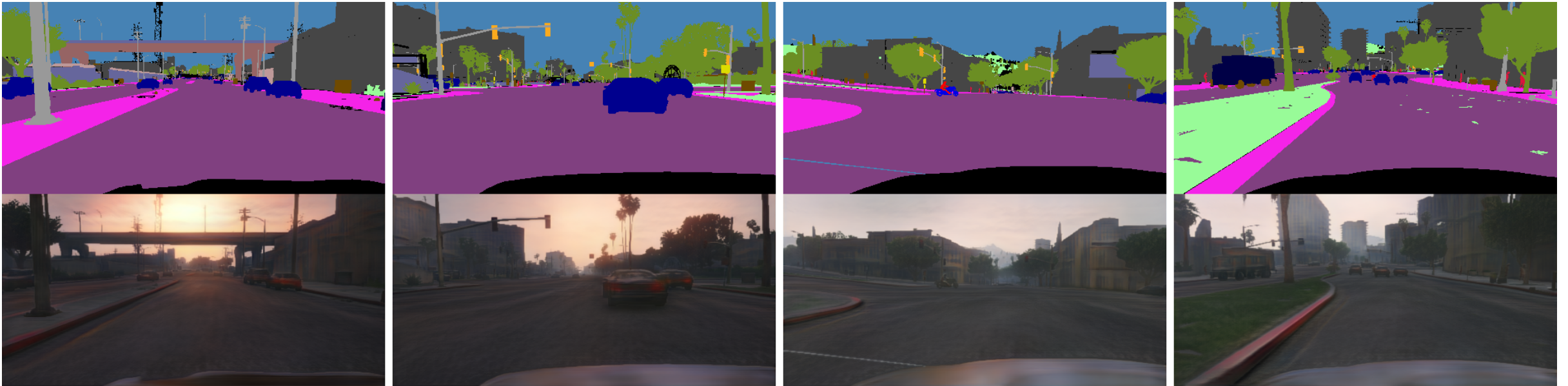


南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019
Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Different inputs with the same $\mathbf{z}$



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



南京大學

Berkeley
UNIVERSITY OF CALIFORNIA

Multimodal Image Synthesis from Scene Layout

Evolving inputs with the same $\mathbf{z}$

IMLE



Pix2pix



K. Li*, T. Zhang* and J. Malik. Diverse Image Synthesis from Semantic Layouts via Conditional IMLE. *ICCV*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets

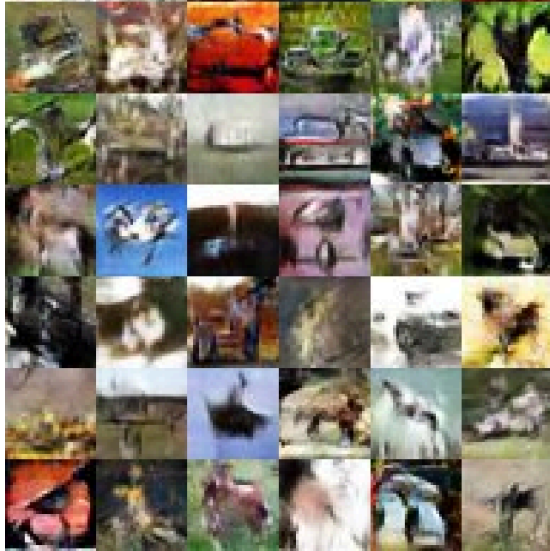


南京大學

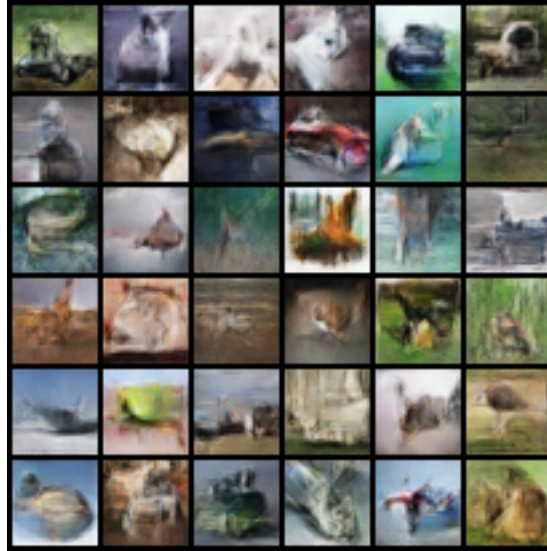
Berkeley
UNIVERSITY OF CALIFORNIA

Unconditional Image Synthesis (Using IMLE on GLO Embeddings)

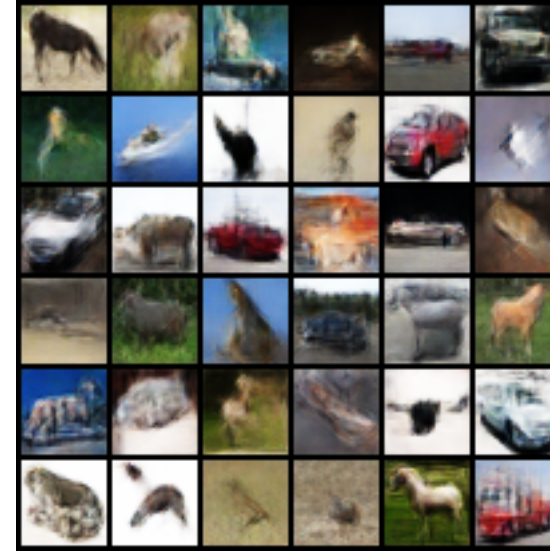
GAN



GLO



GLANN (IMLE+GLO)



Dataset	Adversarial					Non-Adversarial		
	MM GAN	NS GAN	LSGAN	WGAN	BEGAN	VAE	GLO	Ours
MNIST	9.8 ± 0.9	6.8 ± 0.5	7.8 ± 0.6	6.7 ± 0.4	13.1 ± 1.0	23.8 ± 0.6	49.6 ± 0.3	8.6 ± 0.1
Fashion	29.6 ± 1.6	26.5 ± 1.6	30.7 ± 2.2	21.5 ± 1.6	22.9 ± 0.9	58.7 ± 1.2	57.7 ± 0.4	13.0 ± 0.1
Cifar10	72.7 ± 3.6	58.5 ± 1.9	87.1 ± 47.5	55.2 ± 2.3	71.4 ± 1.6	155.7 ± 11.6	65.4 ± 0.2	46.5 ± 0.2
CelebA	65.6 ± 4.2	55.0 ± 3.3	53.9 ± 2.8	41.3 ± 2.0	38.9 ± 0.9	85.7 ± 3.8	52.4 ± 0.5	46.3 ± 0.1

Y. Hoshen, K. Li and J. Malik. Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors. *CVPR*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



Unconditional Image Synthesis

(Using IMLE on GLO Embeddings)

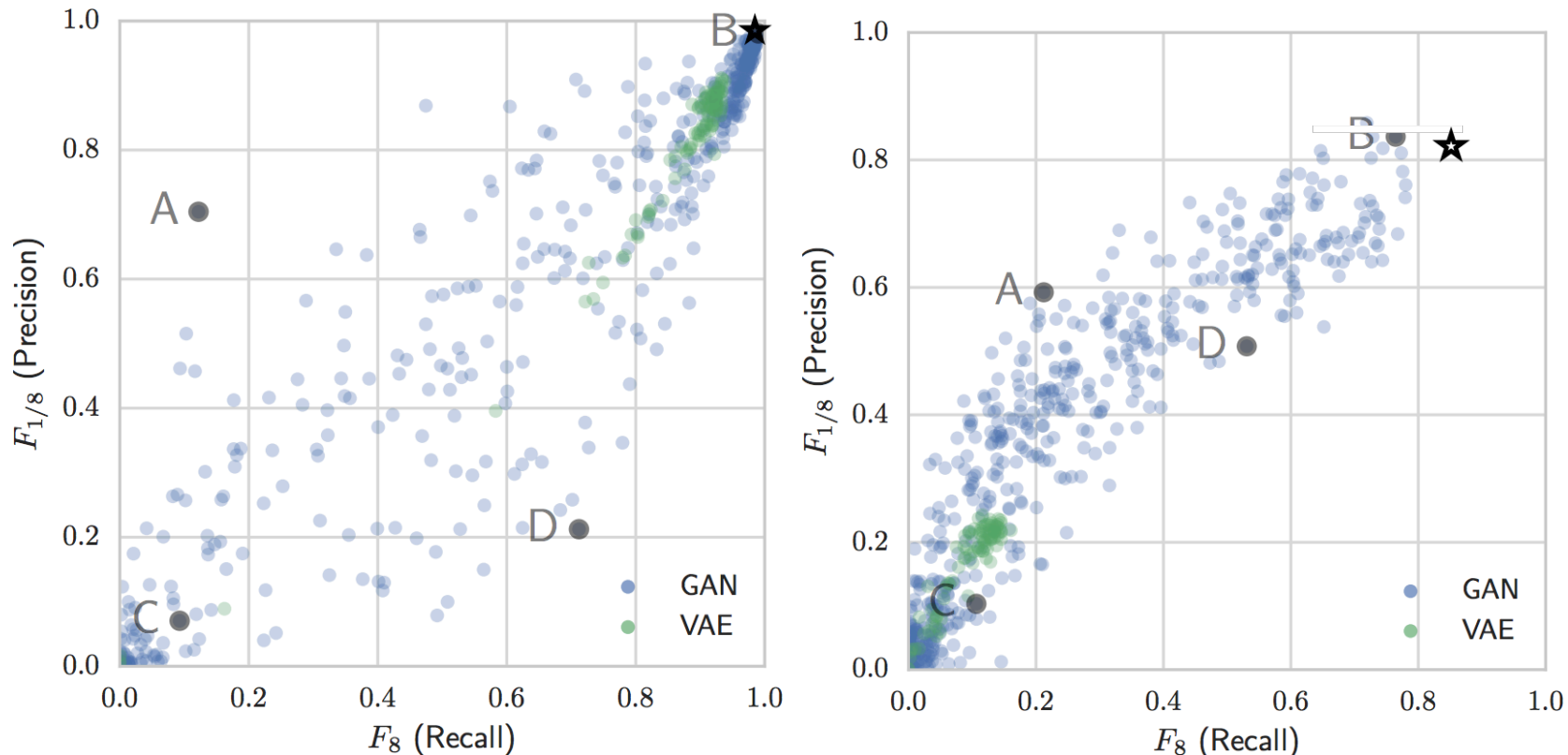


Y. Hoshen, K. Li and J. Malik. Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors. *CVPR*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



Unconditional Image Synthesis (Using IMLE on GLO Embeddings)



Y. Hoshen, K. Li and J. Malik. Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors. *CVPR*, 2019

Overcoming Mode Collapse
in Generative Adversarial Nets



Nearest Neighbour Search

Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

How to Find Nearest Neighbours

- Naïve algorithm:
 1. Compute the distance from the query to each data point.
 2. Sort the distances.
 3. Take the top- k .

How to Find Nearest Neighbours

- Naïve algorithm:
 1. Compute the distance from the query to each data point.
 2. Sort the distances.
 3. Take the top- k .
- Problem: Time complexity is linear in the number of data points.
 - Too computationally expensive when the dataset is large.

How to Find Nearest Neighbours

- Naïve algorithm:
 1. Compute the distance from the query to each data point.
 2. Sort the distances.
 3. Take the first k points.
- Problem: Can we come up with a smarter way that avoids going over all data points?
 - Too computationally expensive when the dataset is large.

Many Attempts Over 40 Years



Many Attempts Over 40 Years



(This is a non-exhaustive list.)

The Curse of Dimensionality

- Exponential dependence of query time on some notion of dimensionality.

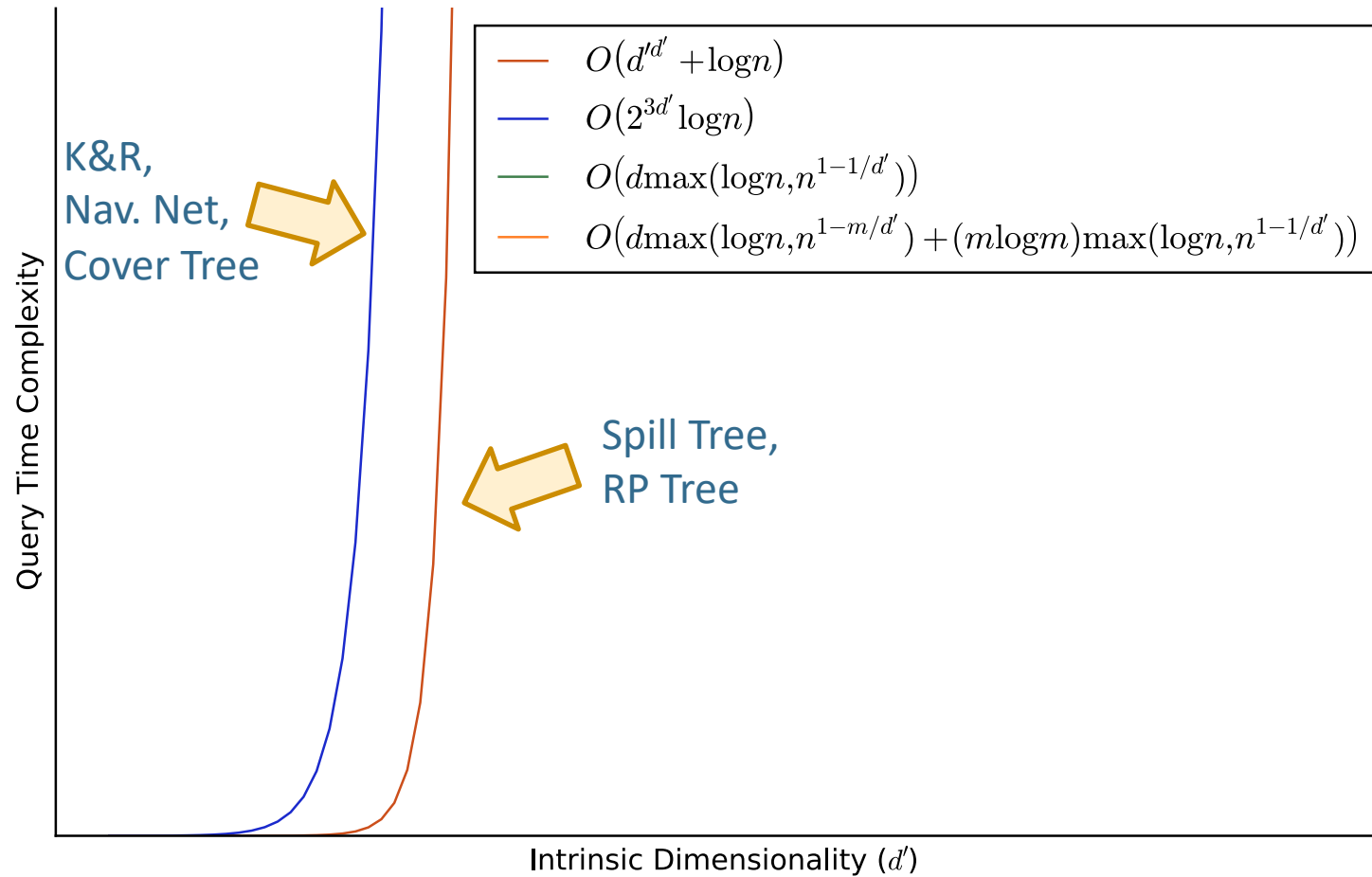
Curse of Ambient Dimensionality

- Plagued deterministic algorithms like k -d trees.
- Overcome by randomization.

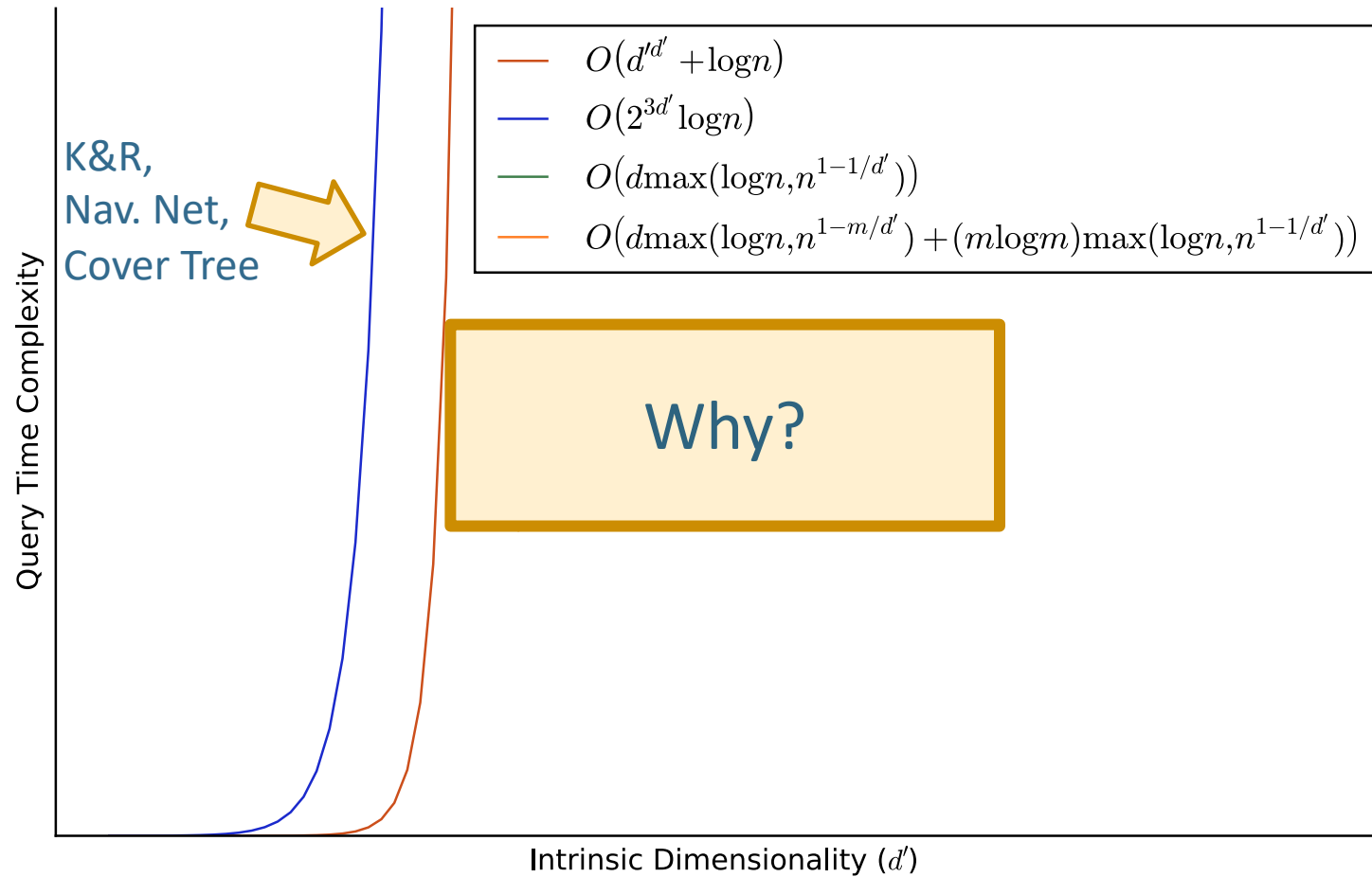
Curse of Intrinsic Dimensionality

- Still a problem, even for algorithms that avoid the curse of ambient dimensionality.

The Curse of Intrinsic Dimensionality

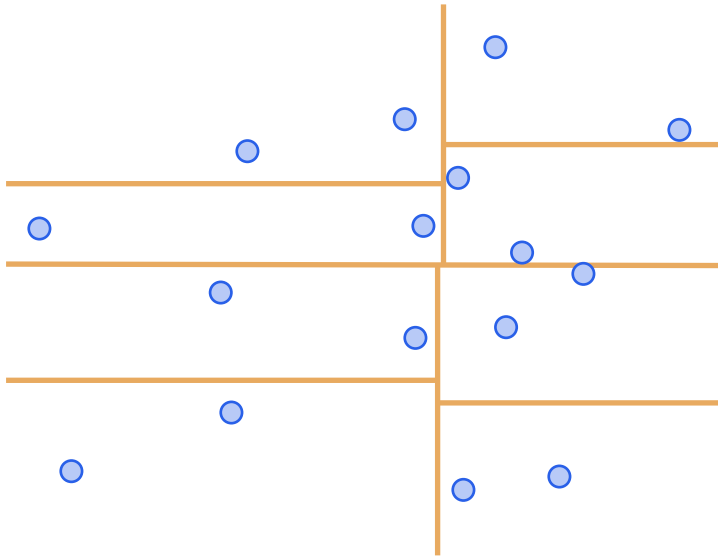


The Curse of Intrinsic Dimensionality

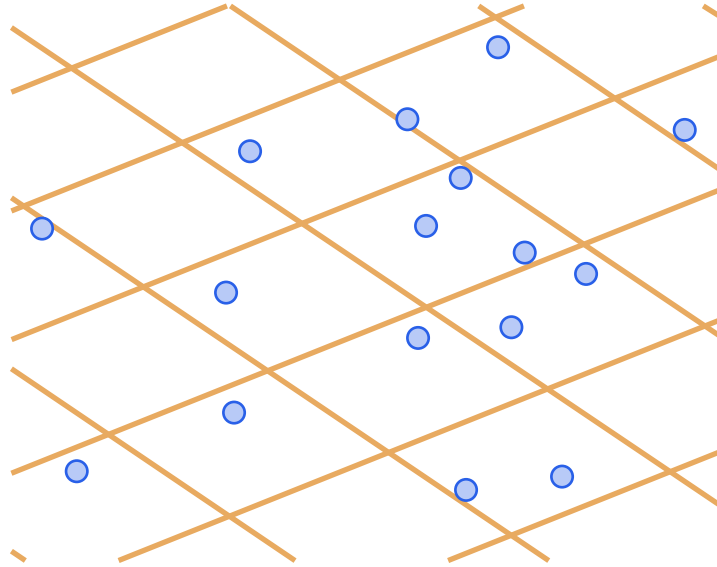


Existing Algorithms Use Space Partitioning

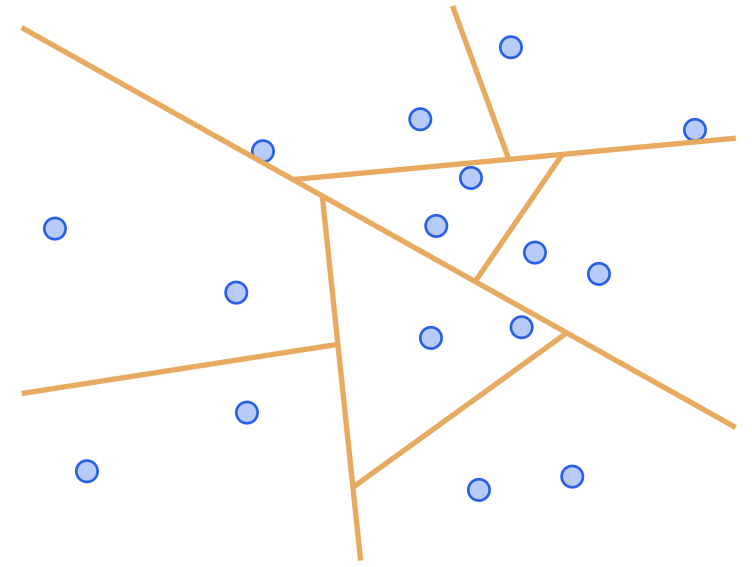
k-d tree



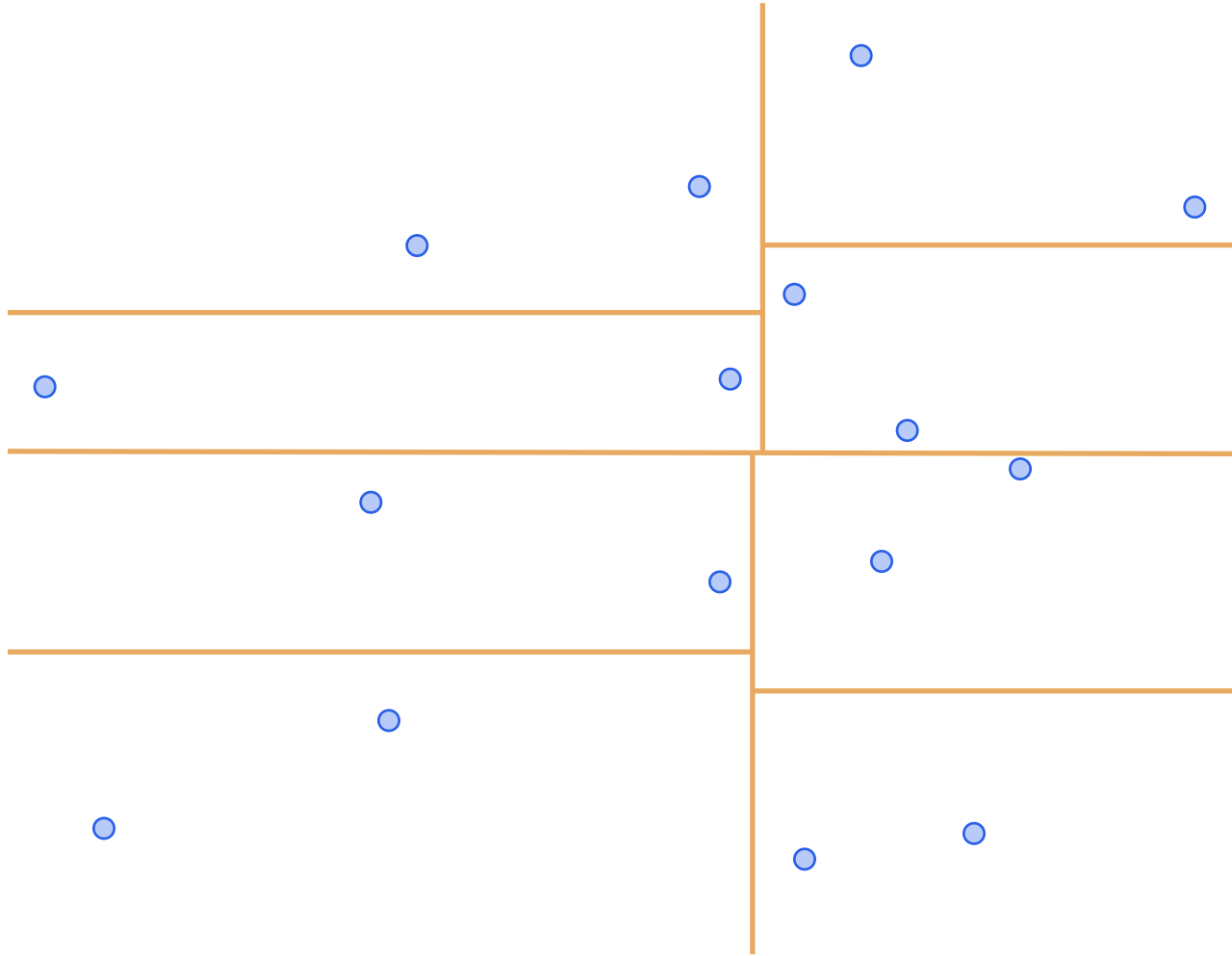
(Euclidean) LSH



RP tree

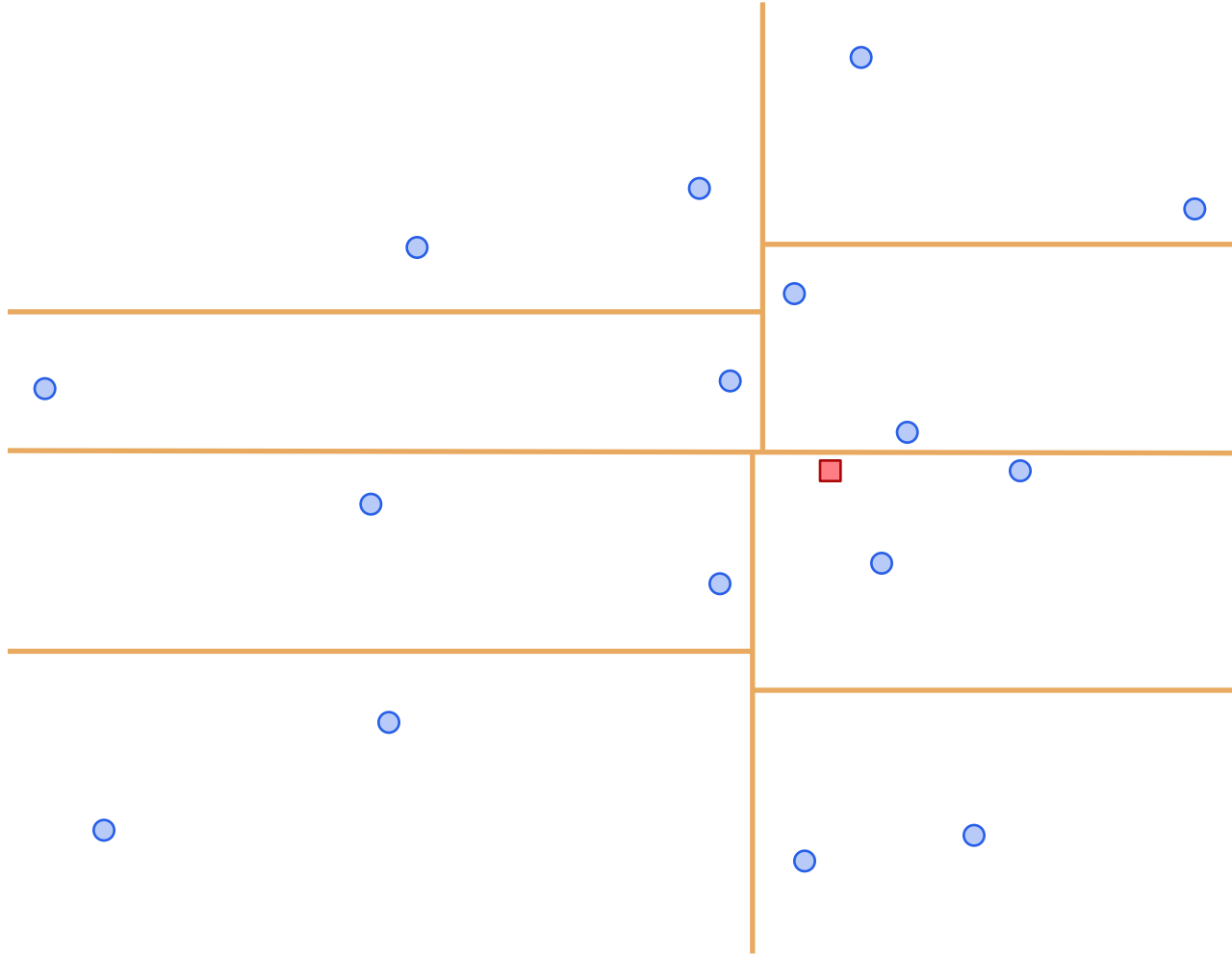


How Space Partitioning Works



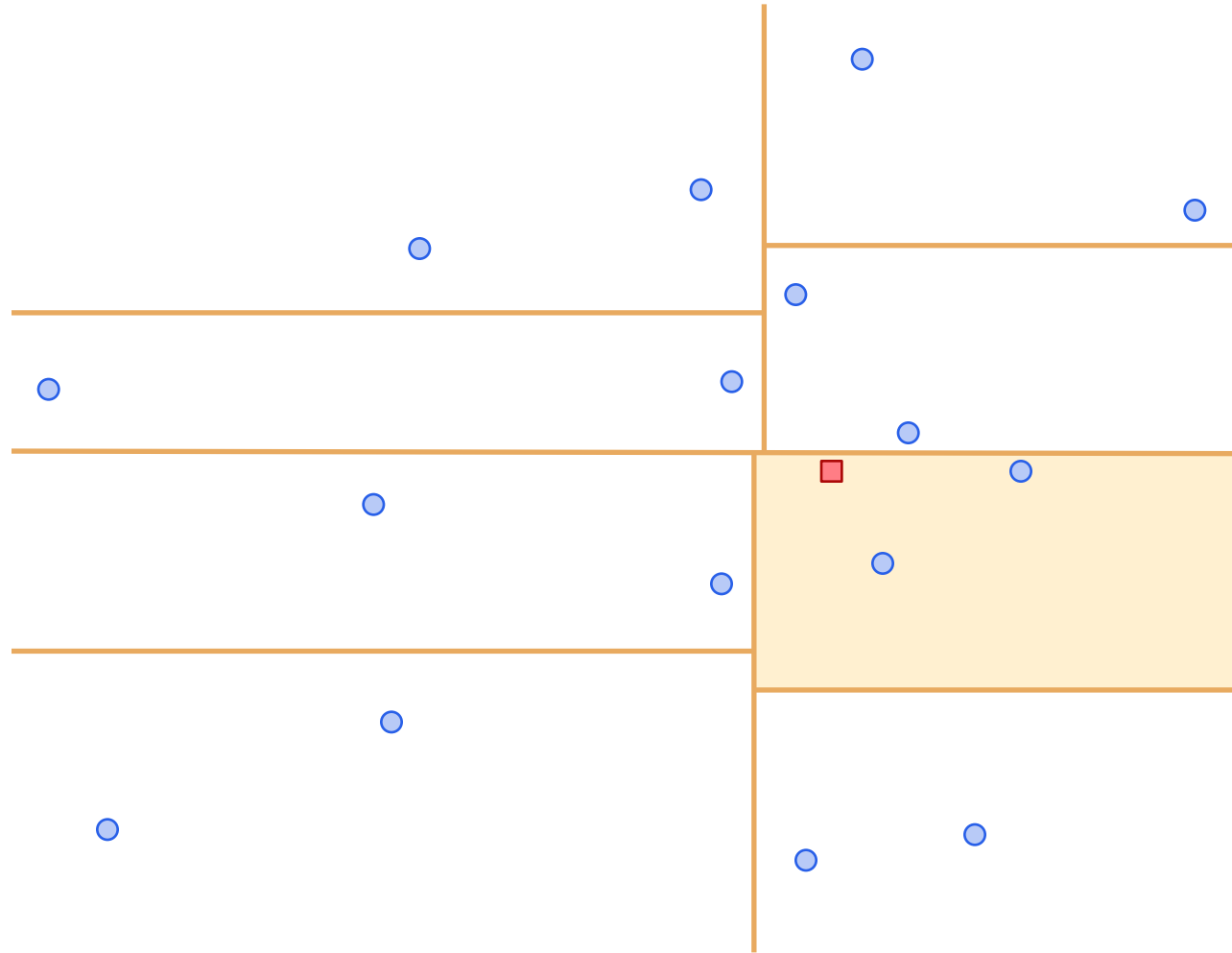
Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

How Space Partitioning Works



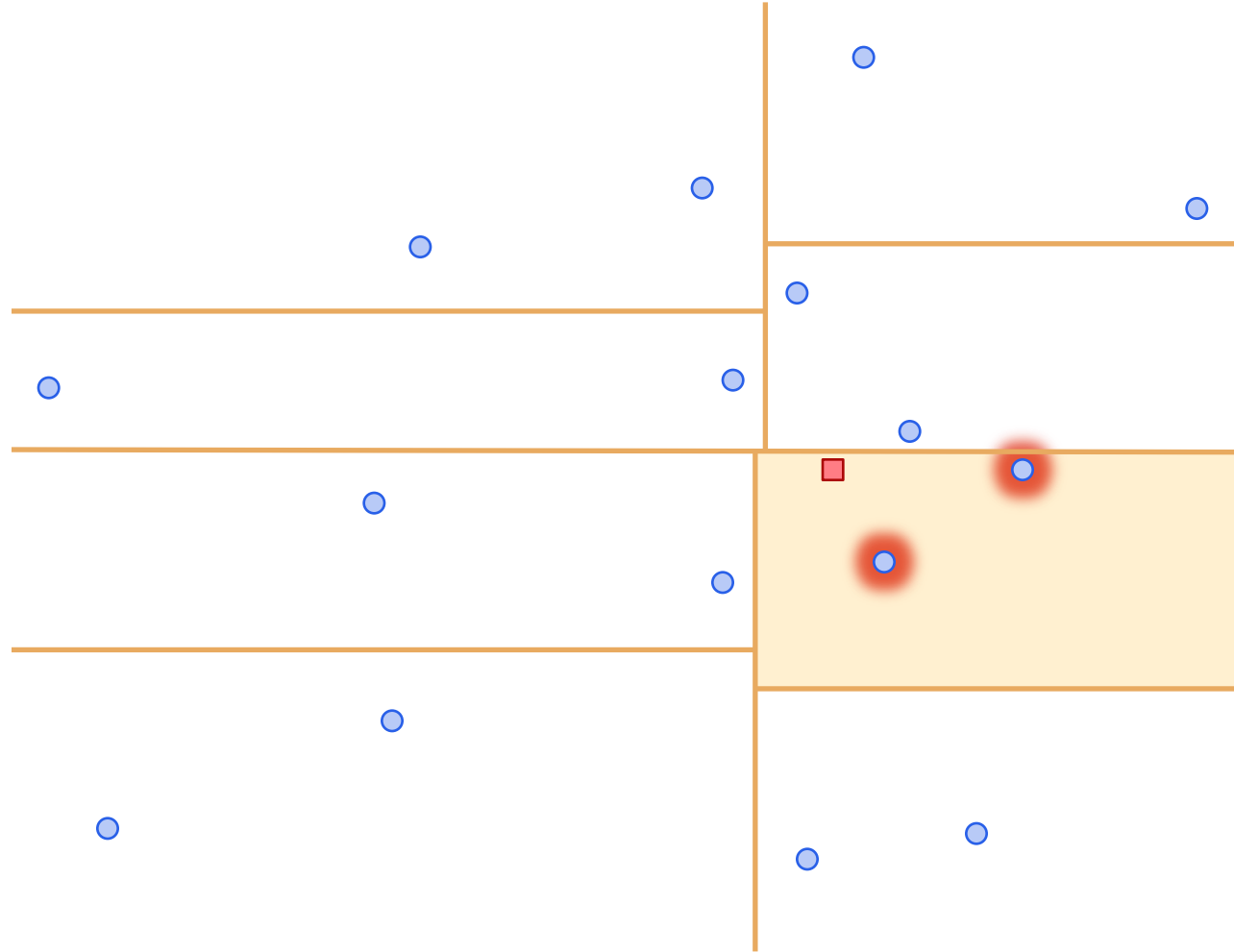
Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

How Space Partitioning Works



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

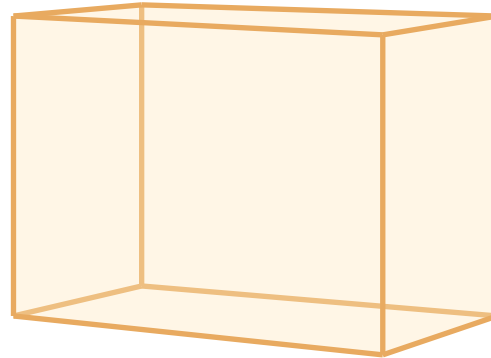
How Space Partitioning Works



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

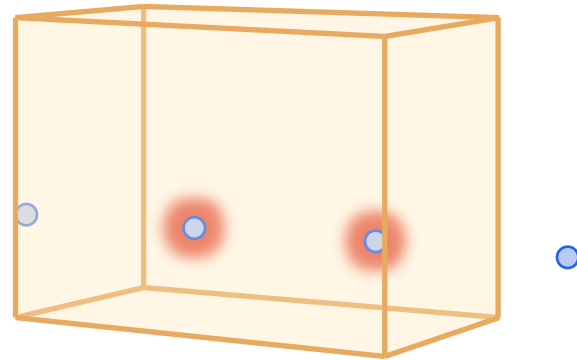


The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

$$d' = 1$$

Intrinsic
Dimensionality

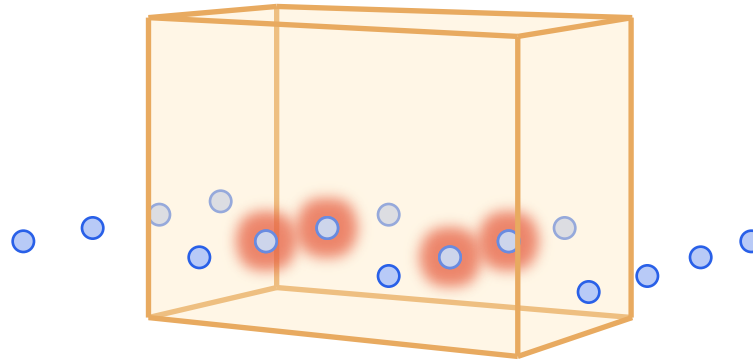


The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

$$d' = 2$$

Intrinsic
Dimensionality

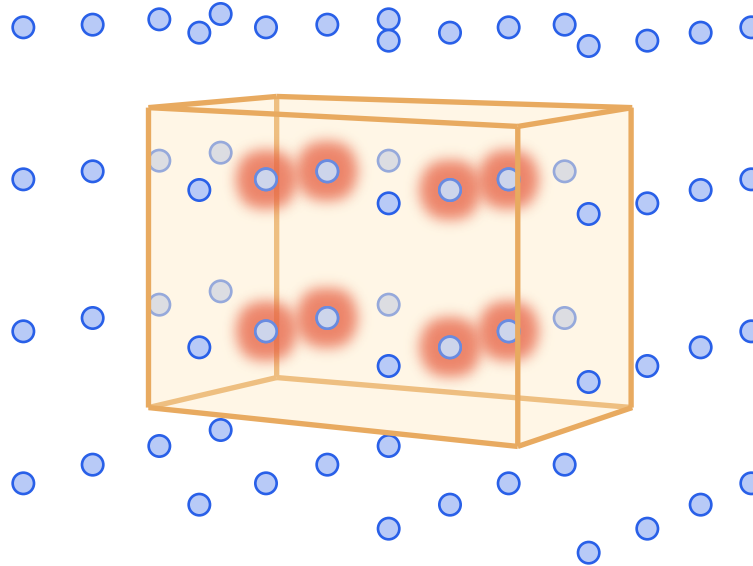


The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

Intrinsic
Dimensionality

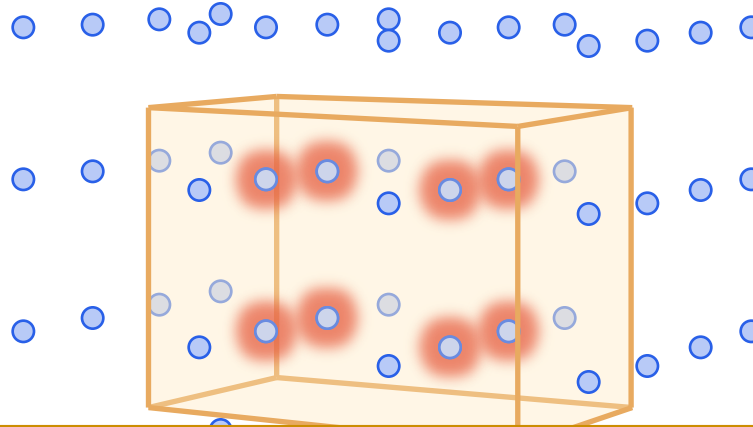


The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

Intrinsic
Dimensionality



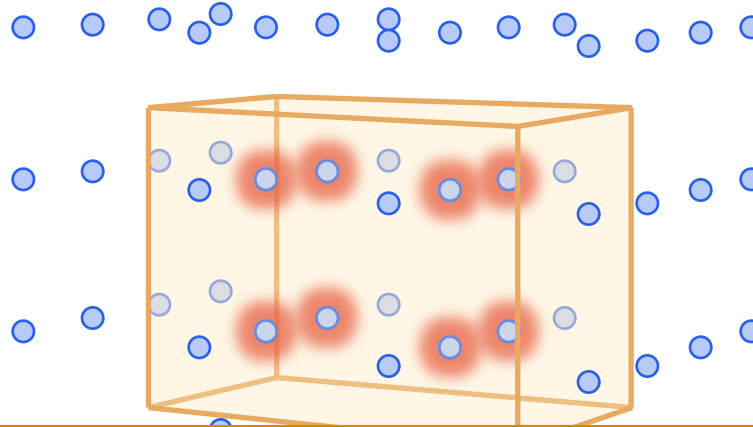
As intrinsic dimensionality increases, the number of points within the cell grows *exponentially*.

The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

Intrinsic
Dimensionality



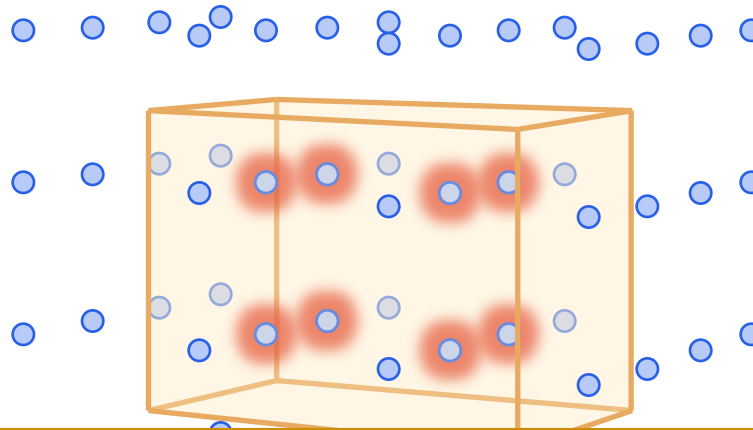
This seems discouraging – regardless of the partitioning scheme, will always get exponential dependence.

The Curse of Intrinsic Dimensionality

- Consider any cell in a data-independent partitioning.

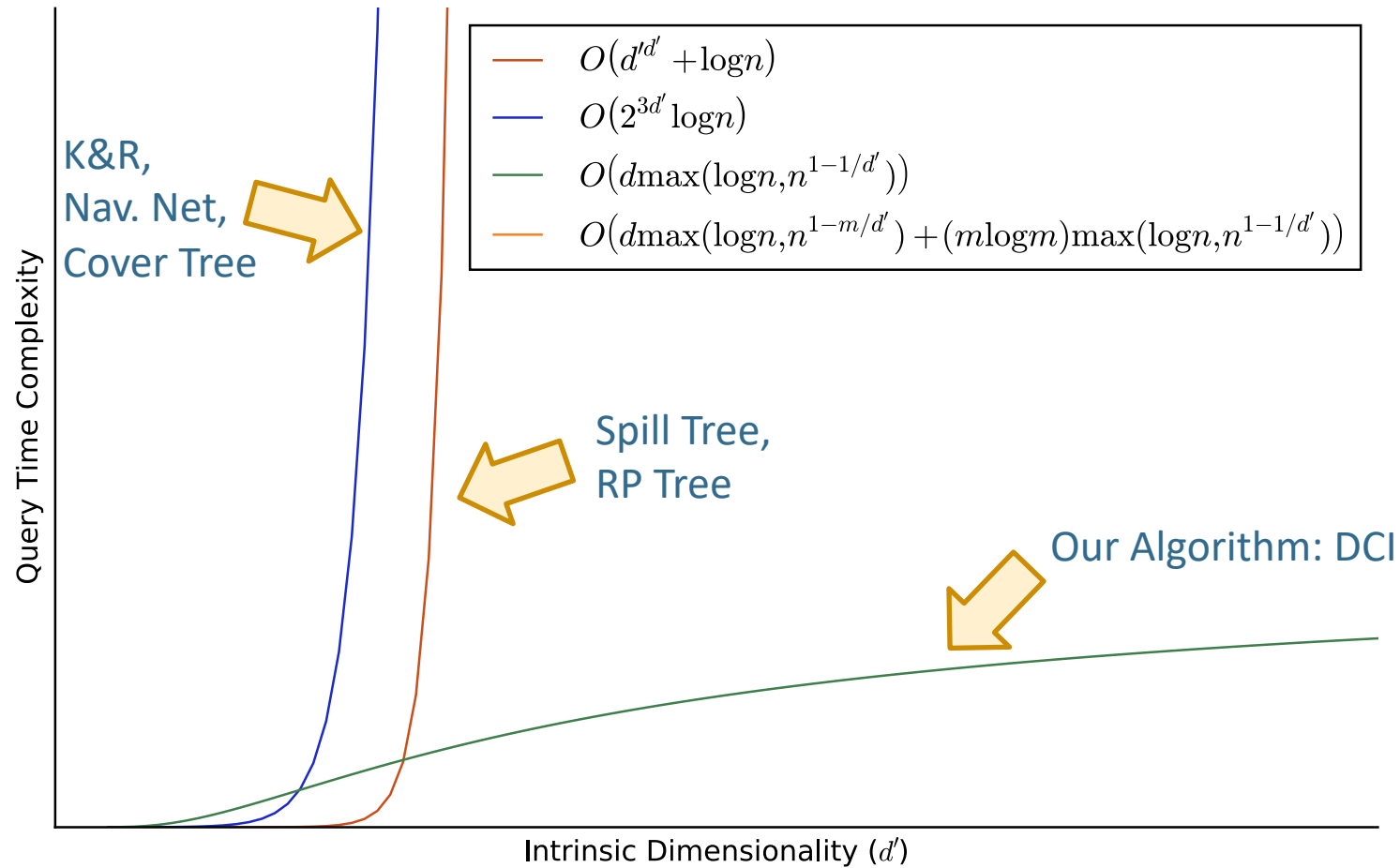
$$d' = 3$$

Intrinsic
Dimensionality

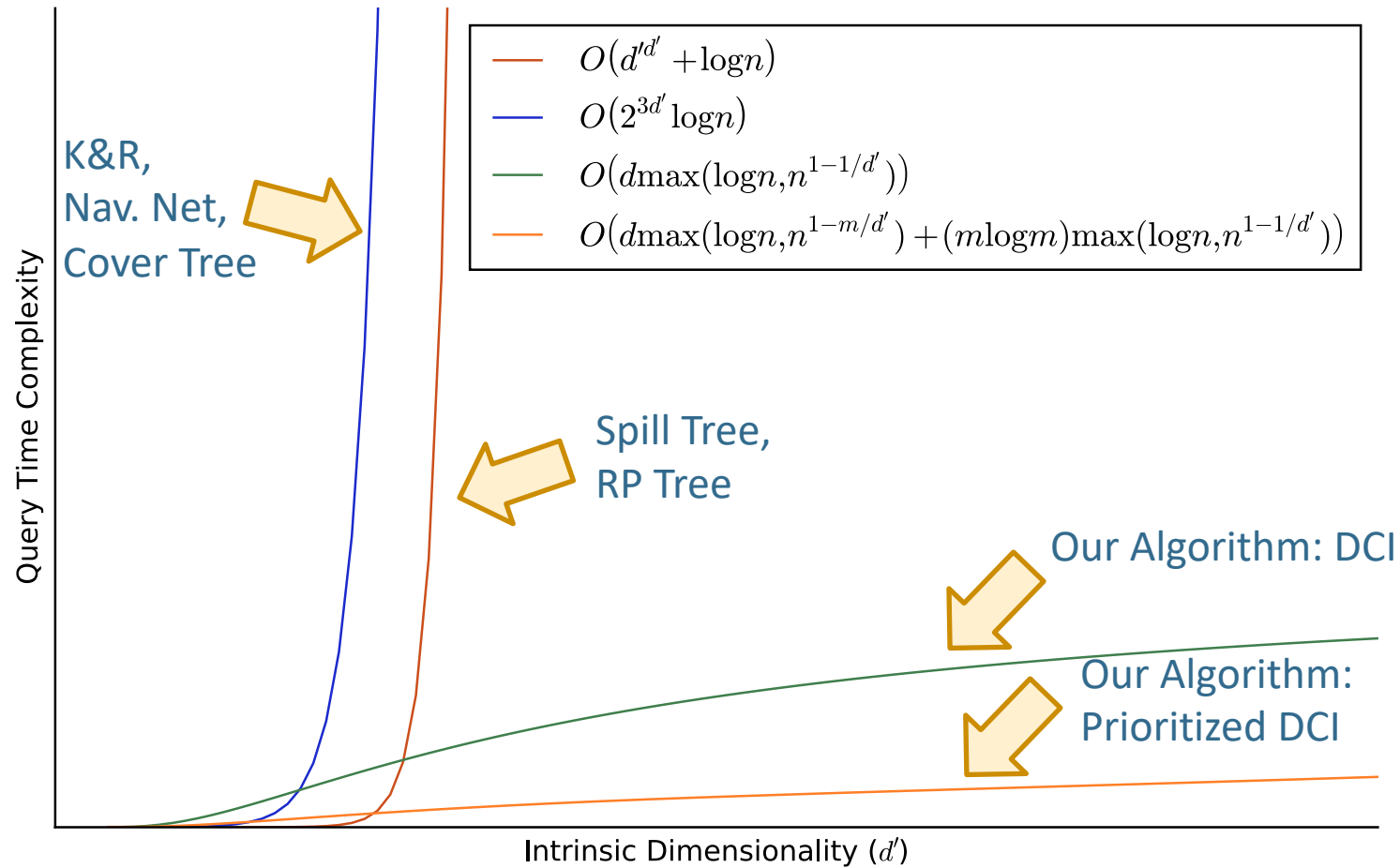


Nevertheless, can we somehow overcome an exponential dependence on intrinsic dimensionality?

The Curse of Intrinsic Dimensionality



The Curse of Intrinsic Dimensionality



Key Insight

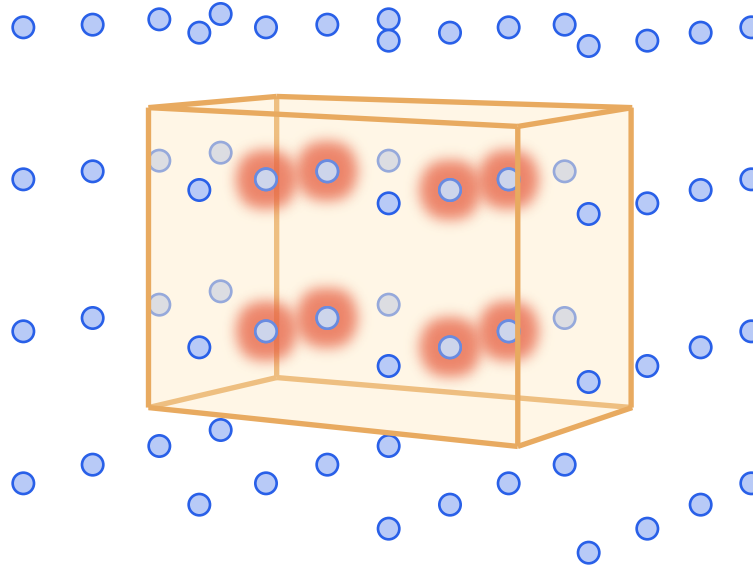
Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Recall...

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

Intrinsic
Dimensionality

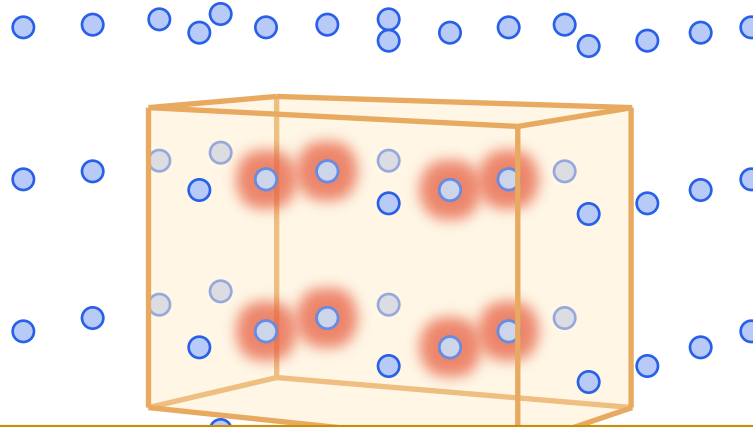


Recall...

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

Intrinsic
Dimensionality



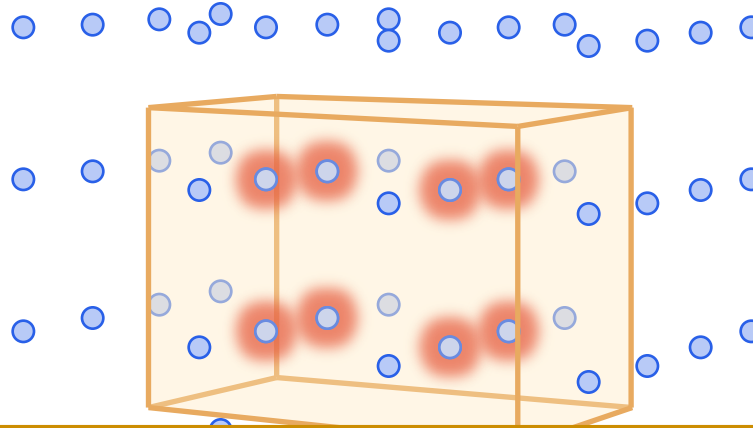
As intrinsic dimensionality increases, the number of points within the cell grows *exponentially*.

Recall...

- Consider any cell in a data-independent partitioning.

$$d' = 3$$

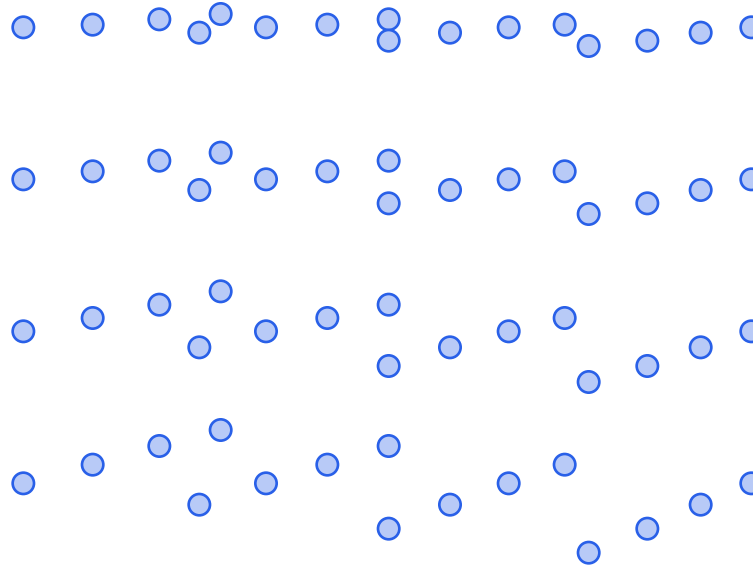
Intrinsic
Dimensionality



Space partitioning is a non-starter.

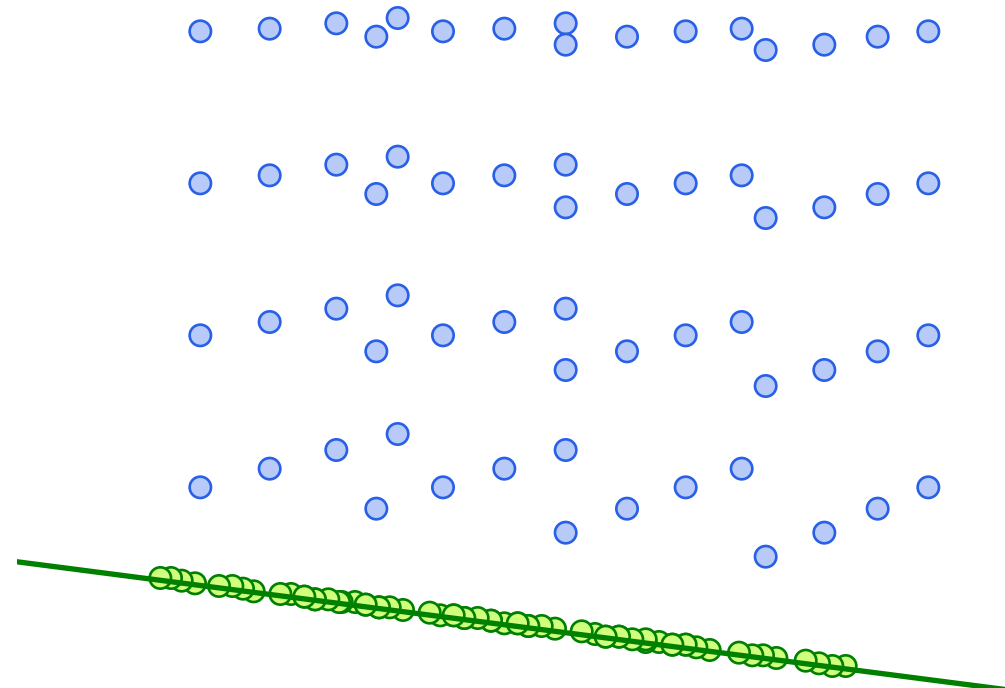
No More Grids

- Let's try eliminating space partitioning.



No More Grids

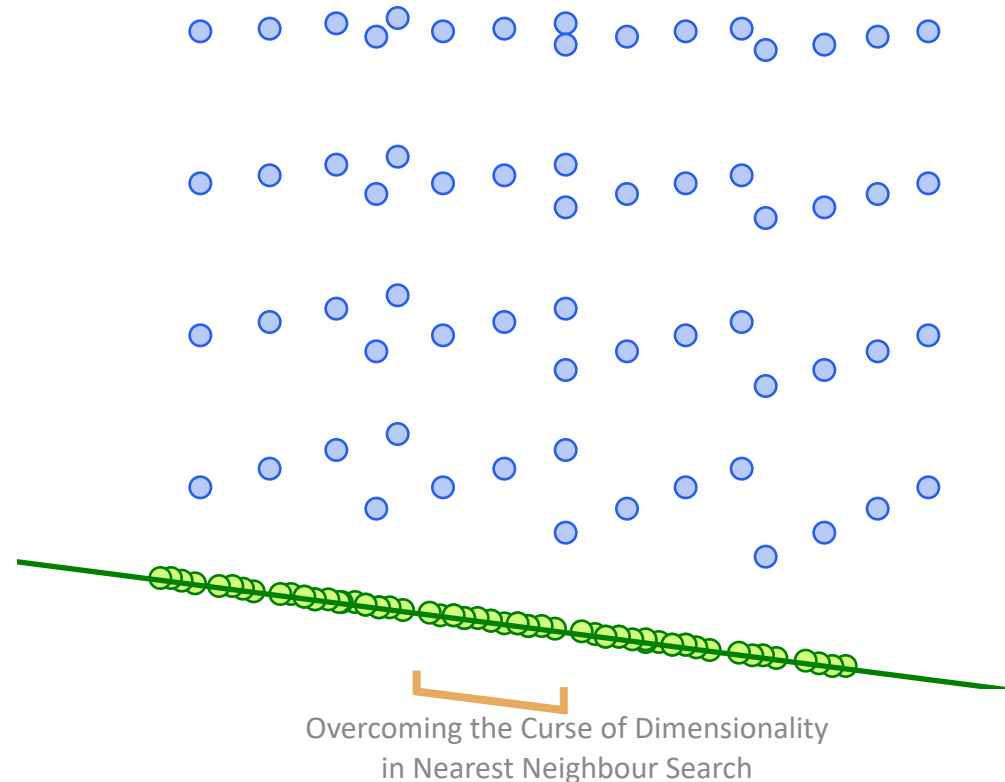
- Instead, we project all data points onto a random line and search along the line.



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

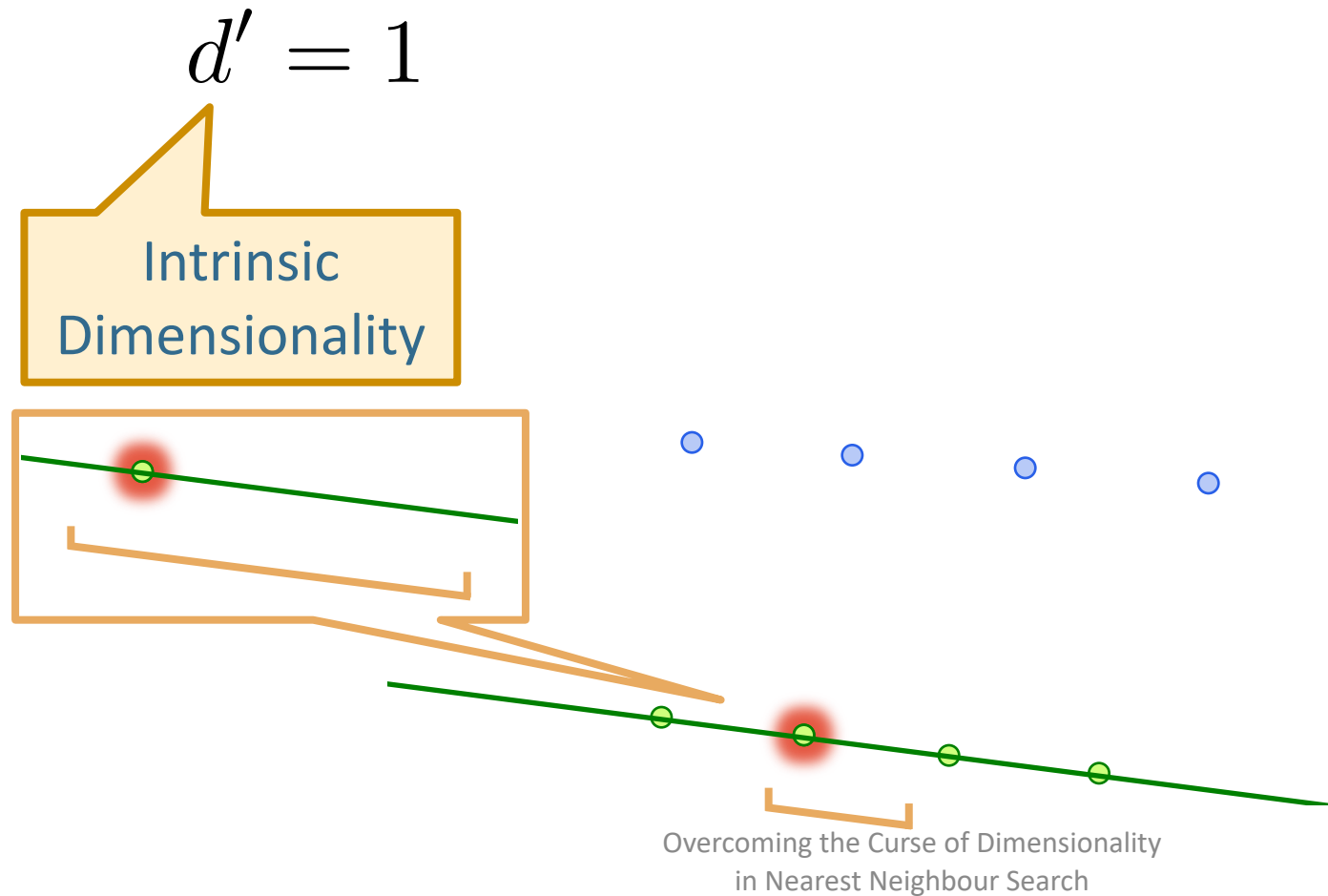
Is This Enough?

- Consider a fixed neighbourhood (i.e.: a data-independent interval) on the line.



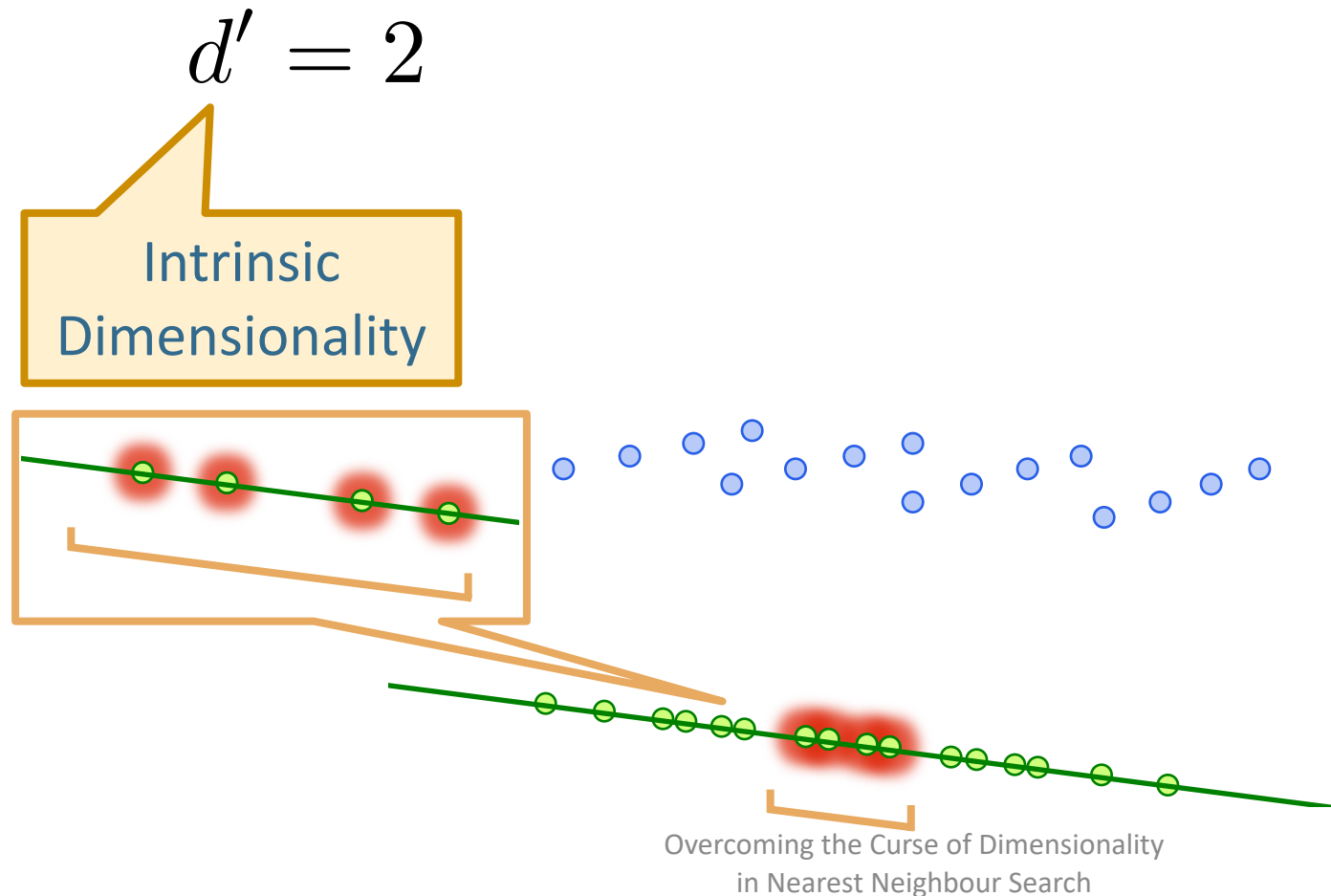
Is This Enough?

- Consider a data-independent interval on the line.



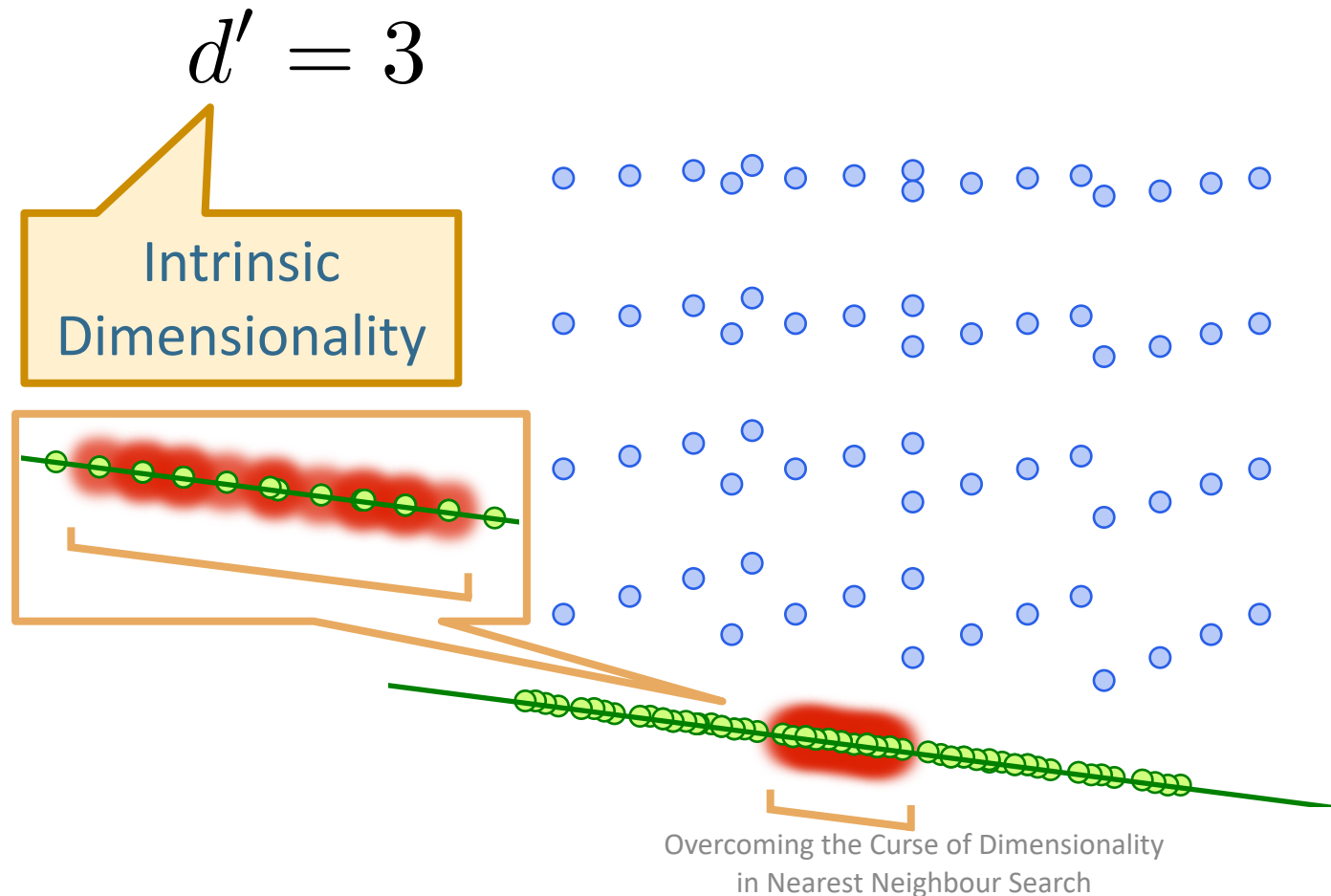
Is This Enough?

- Consider a data-independent interval on the line.



Is This Enough?

- Consider a data-independent interval on the line.

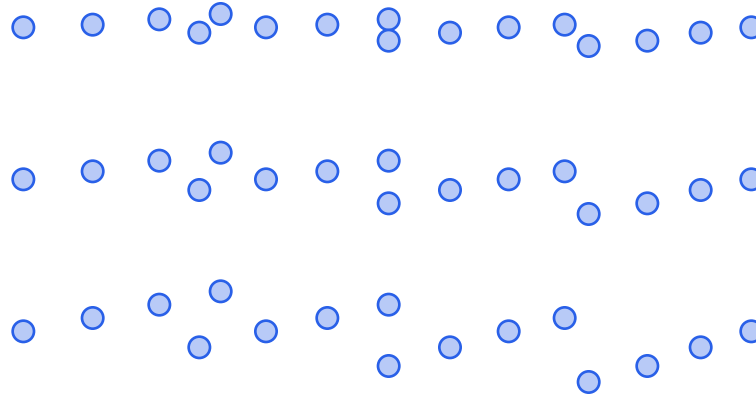
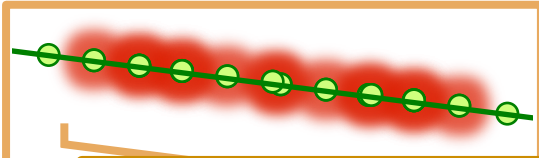


Is This Enough?

- Consider a data-independent interval on the line.

$$d' = 3$$

Intrinsic
Dimensionality



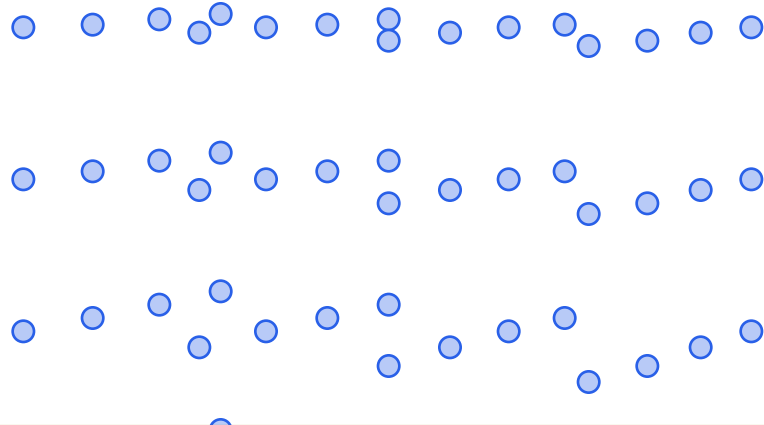
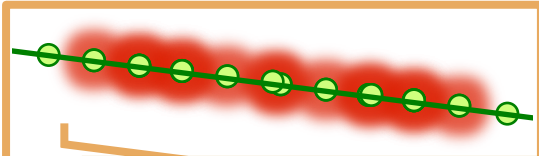
The number of points within the interval still grows
exponentially.

Is This Enough?

- Consider a data-independent interval on the line.

$$d' = 3$$

Intrinsic
Dimensionality



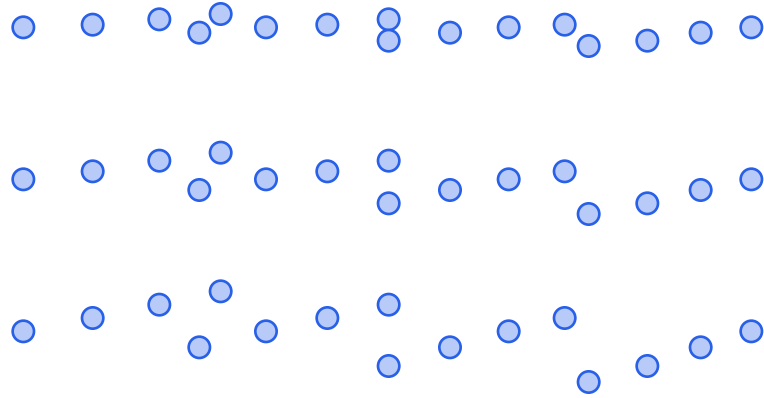
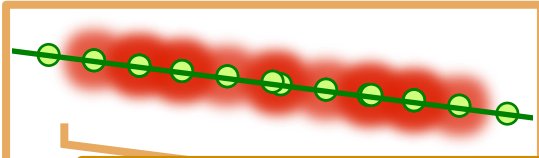
This is not useful!

Is This Enough?

- Consider a data-independent interval on the line.

$$d' = 3$$

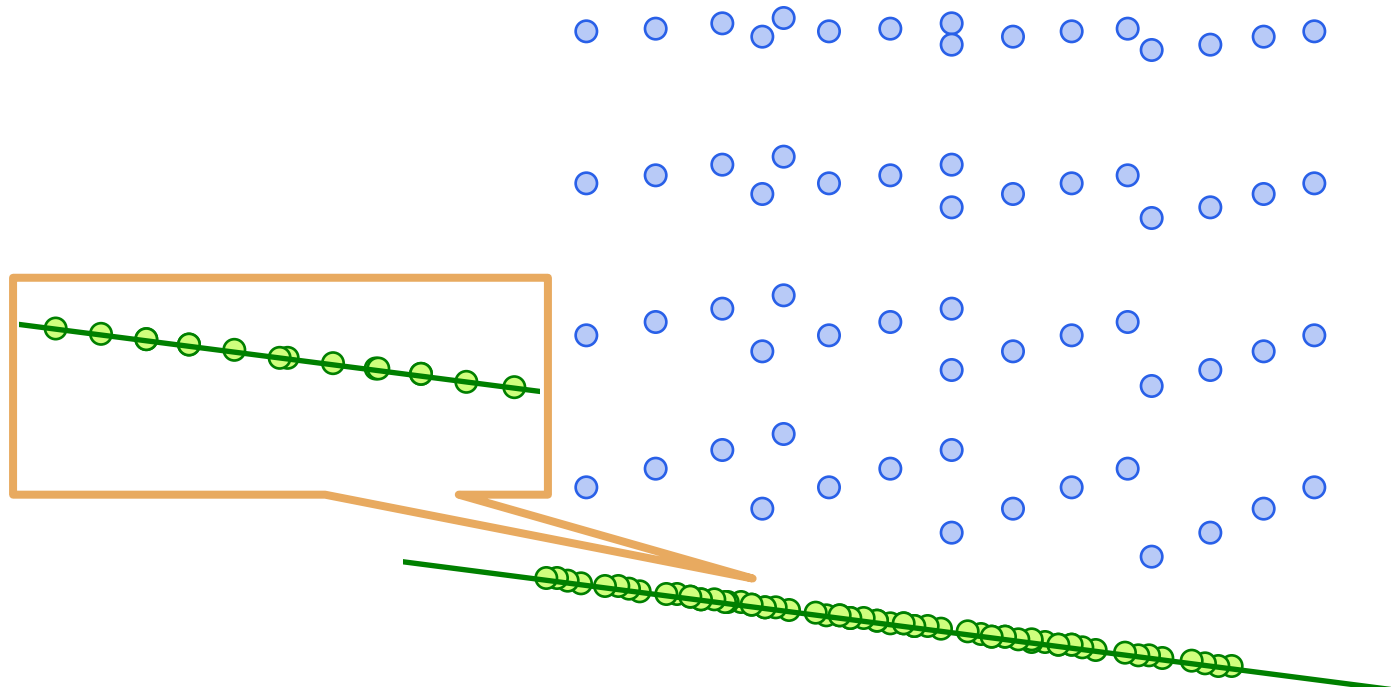
Intrinsic
Dimensionality



Do we really need to pick a (data-independent)
interval?

Closeness in Location vs. Rank

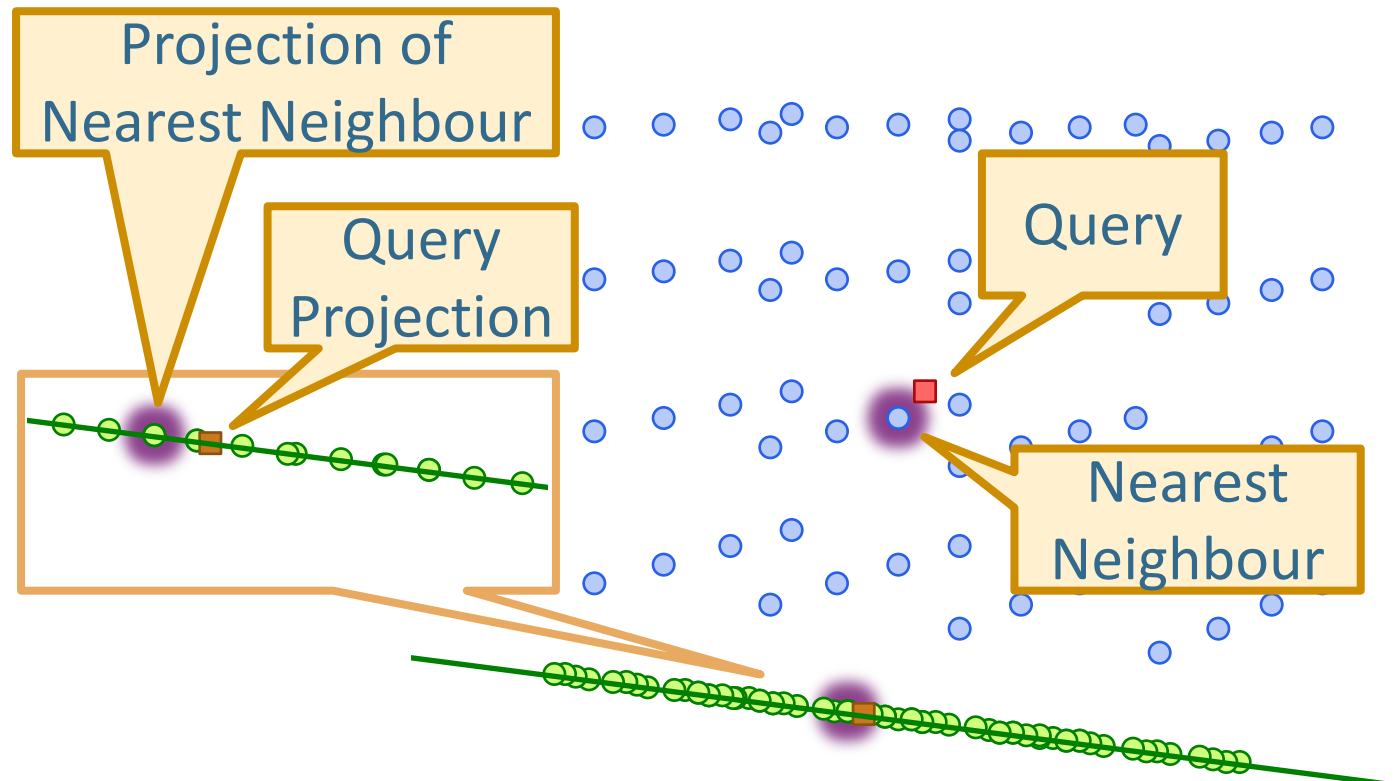
- What do we need to return the correct nearest neighbour?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Closeness in Location vs. Rank

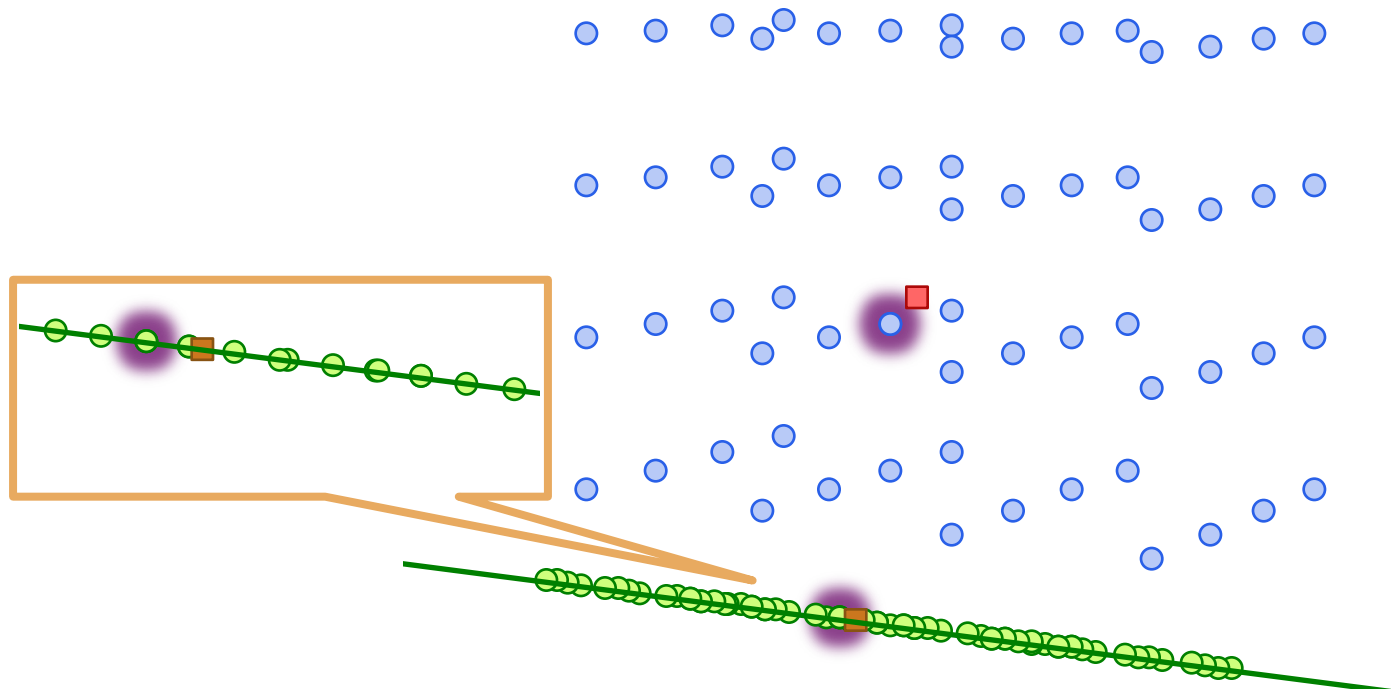
- What do we need to return the correct nearest neighbour?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Closeness in Location vs. Rank

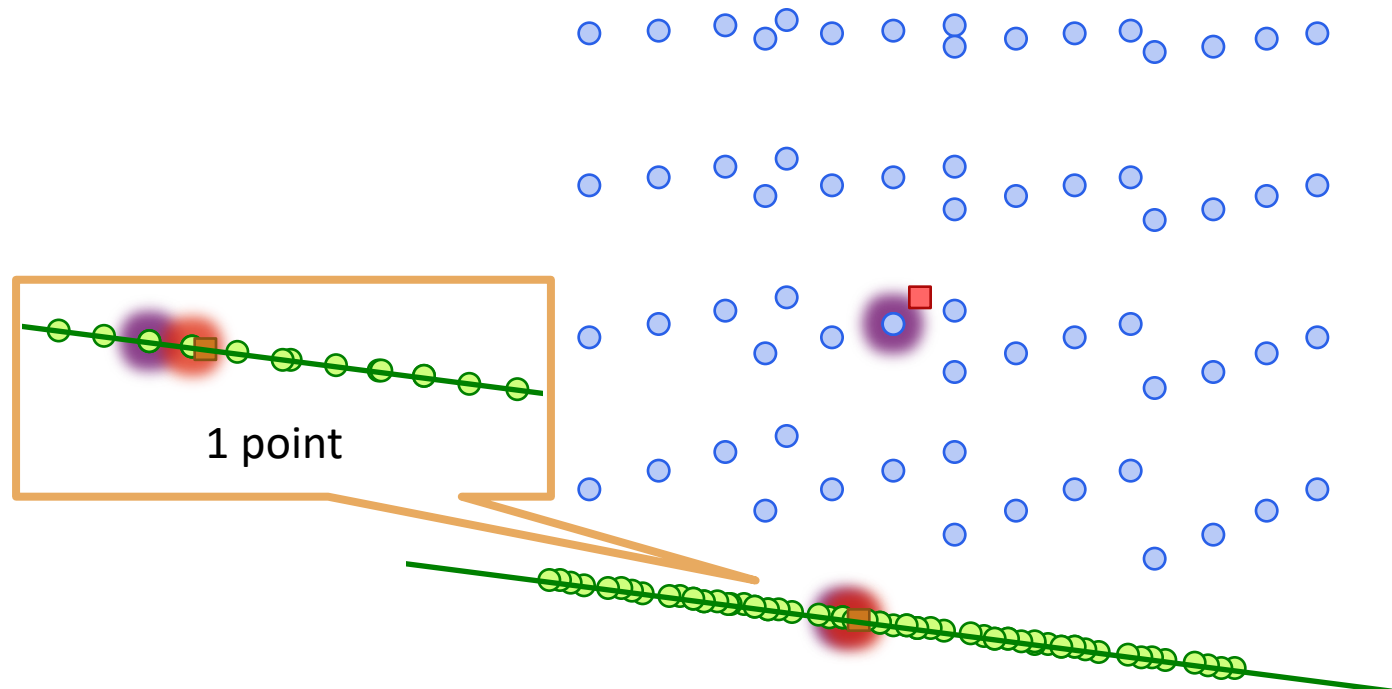
- What if we just count the number of points along the projection line starting from the query projection?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Closeness in Location vs. Rank

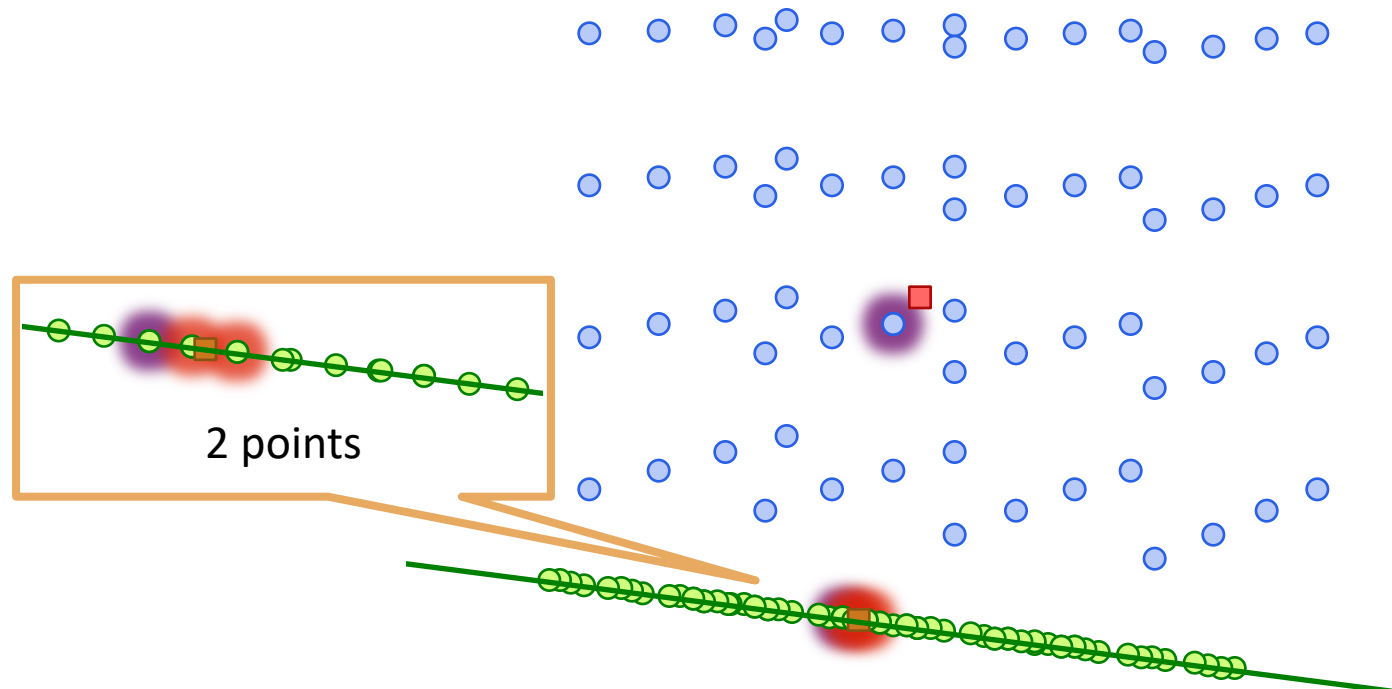
- What if we just count the number of points along the projection line starting from the query projection?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Closeness in Location vs. Rank

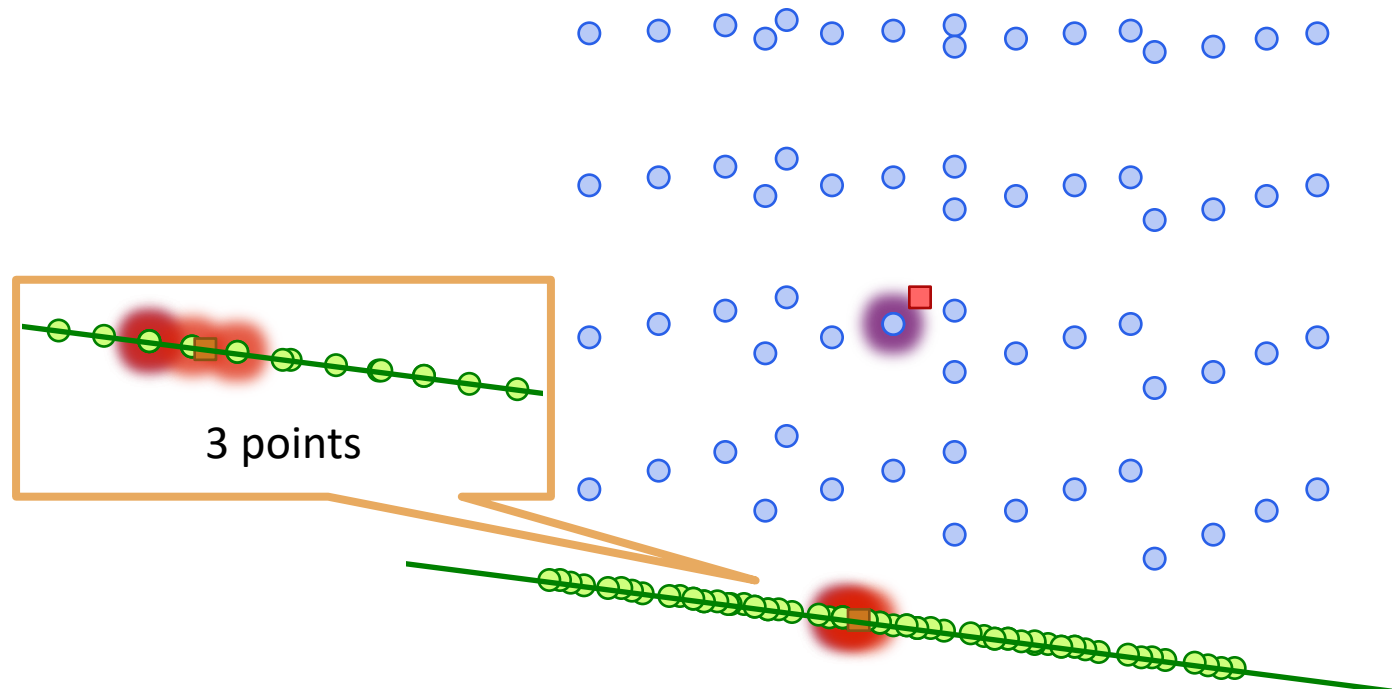
- What if we just count the number of points along the projection line starting from the query projection?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Closeness in Location vs. Rank

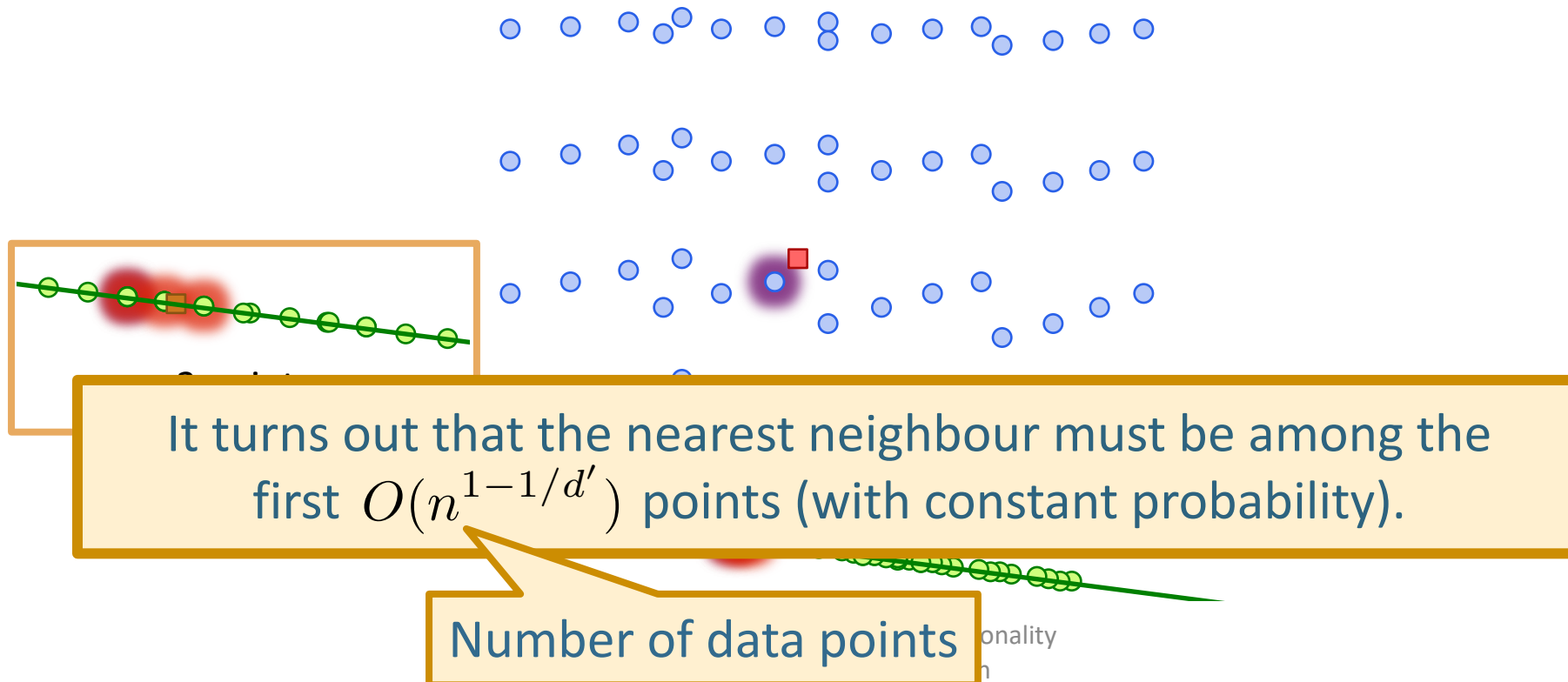
- What if we just count the number of points along the projection line starting from the query projection?



Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

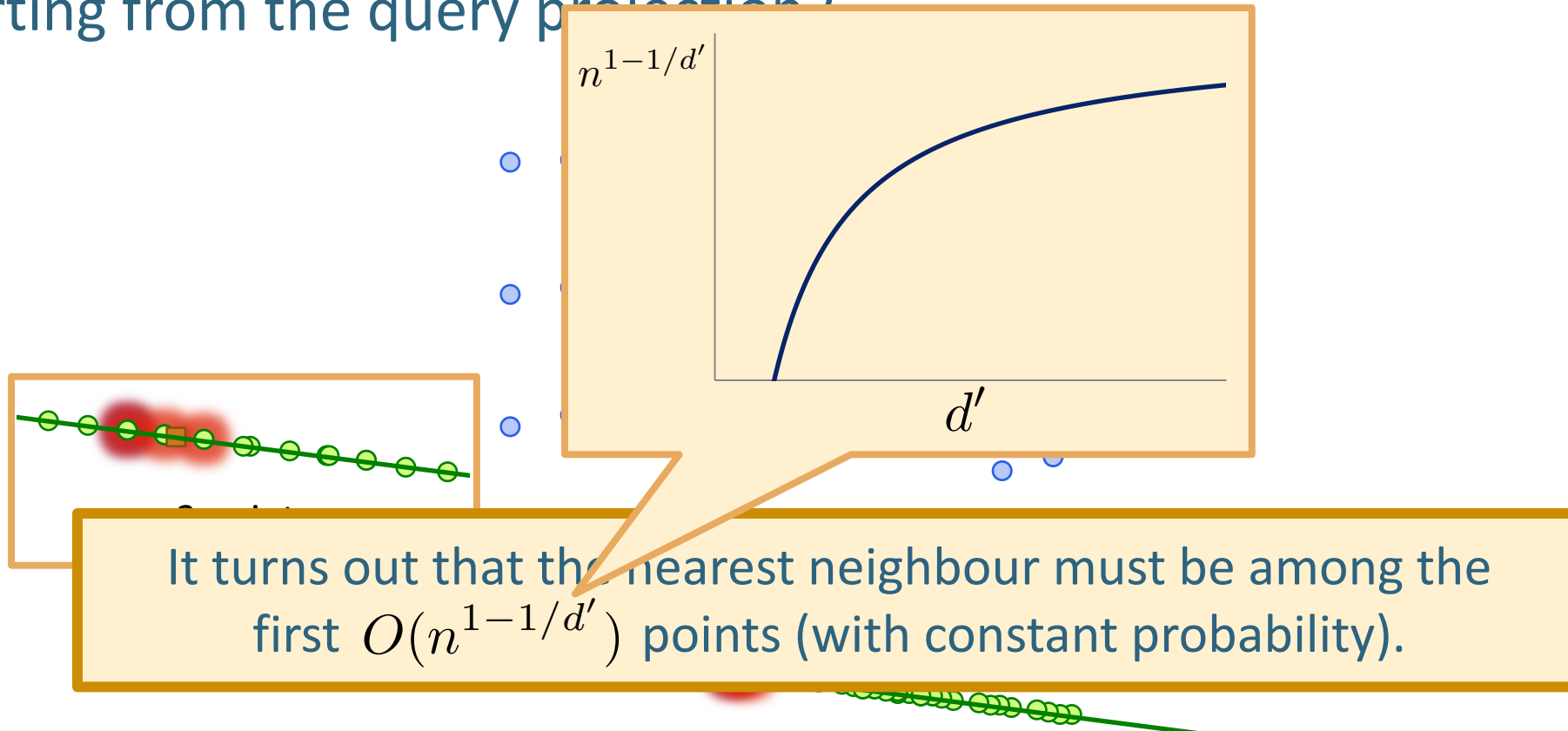
Closeness in Location vs. Rank

- What if we just count the number of points along the projection line starting from the query projection?



Closeness in Location vs. Rank

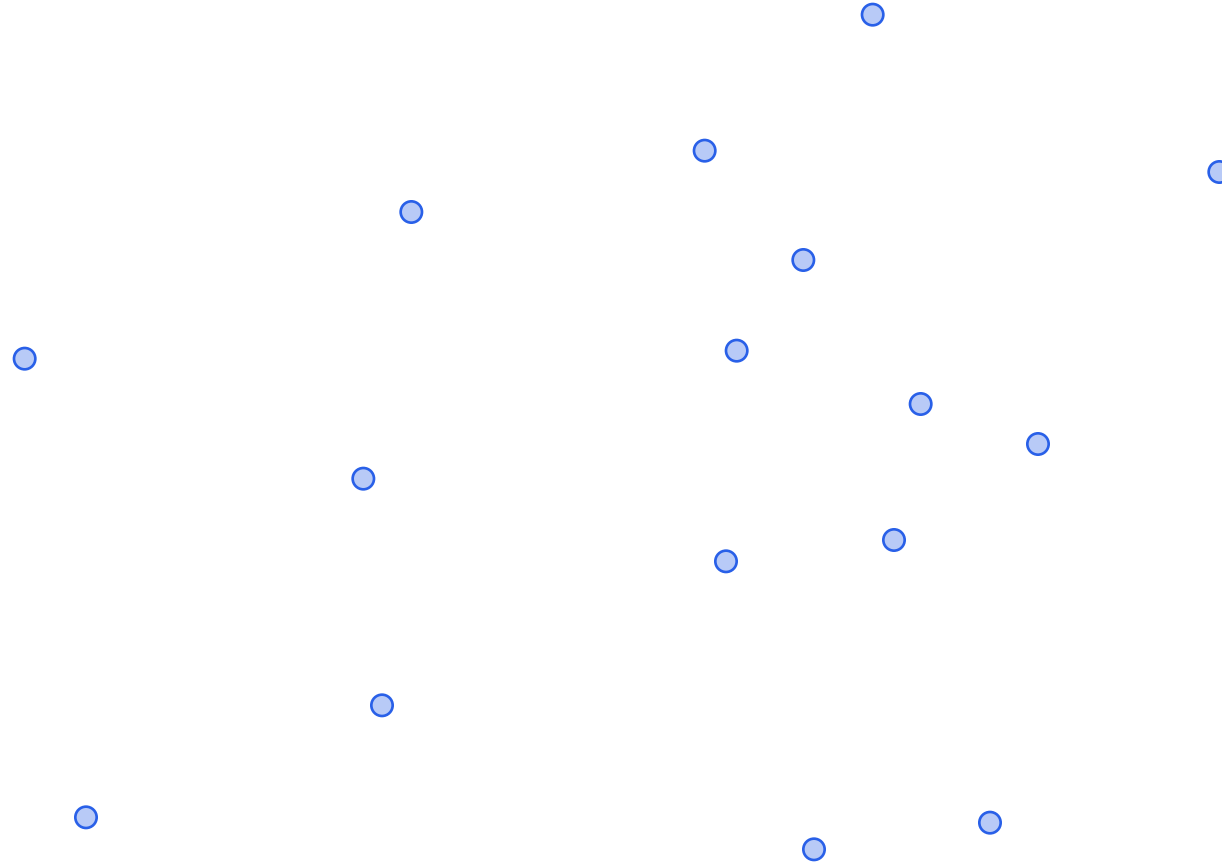
- What if we just count the number of points along the projection line starting from the query projection?



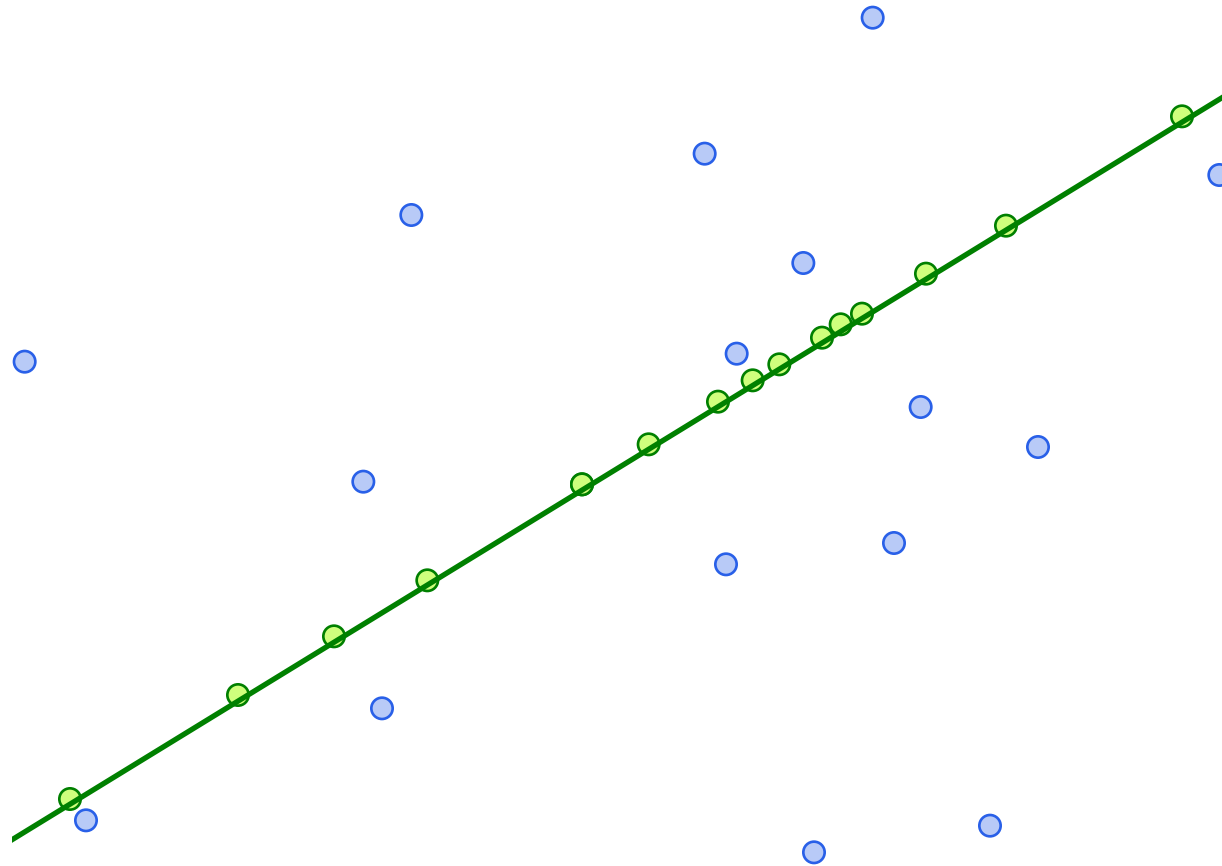
How the Algorithm Works

Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Prioritized DCI

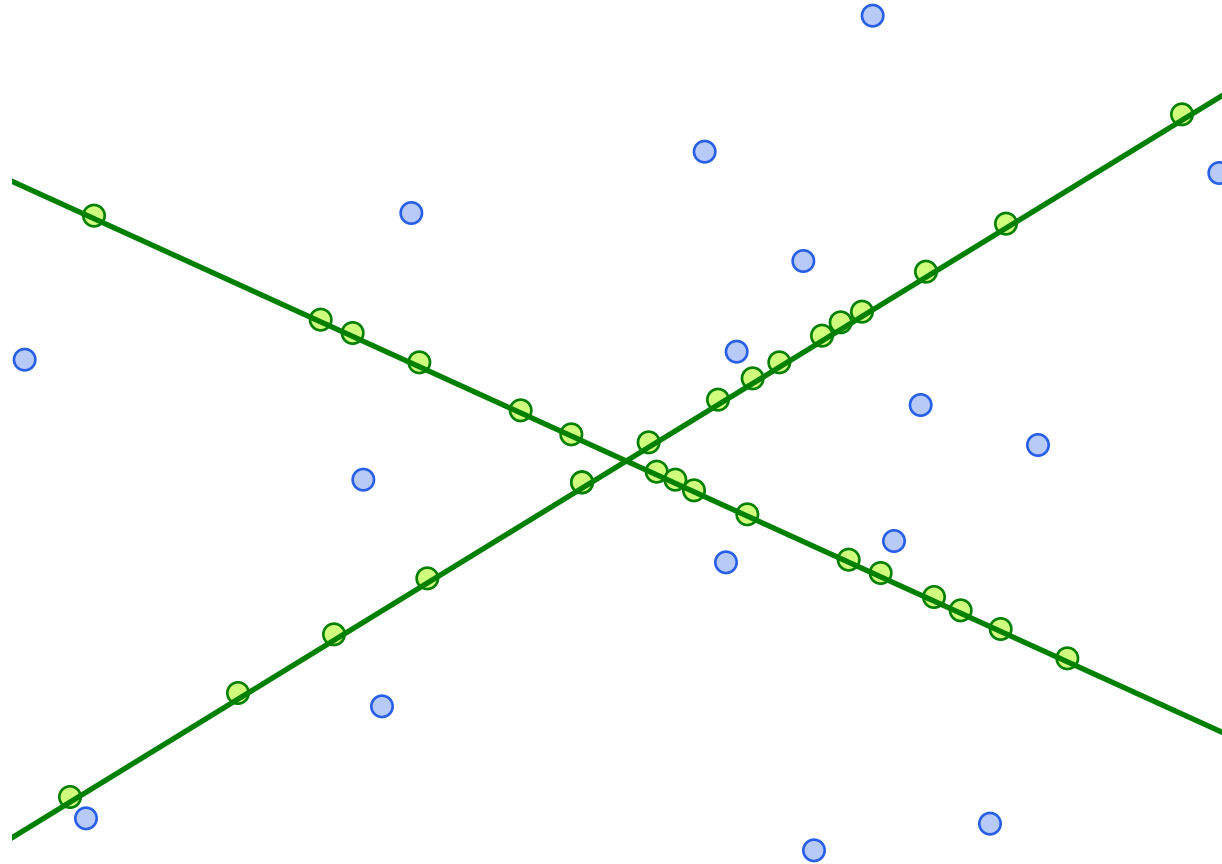


Prioritized DCI



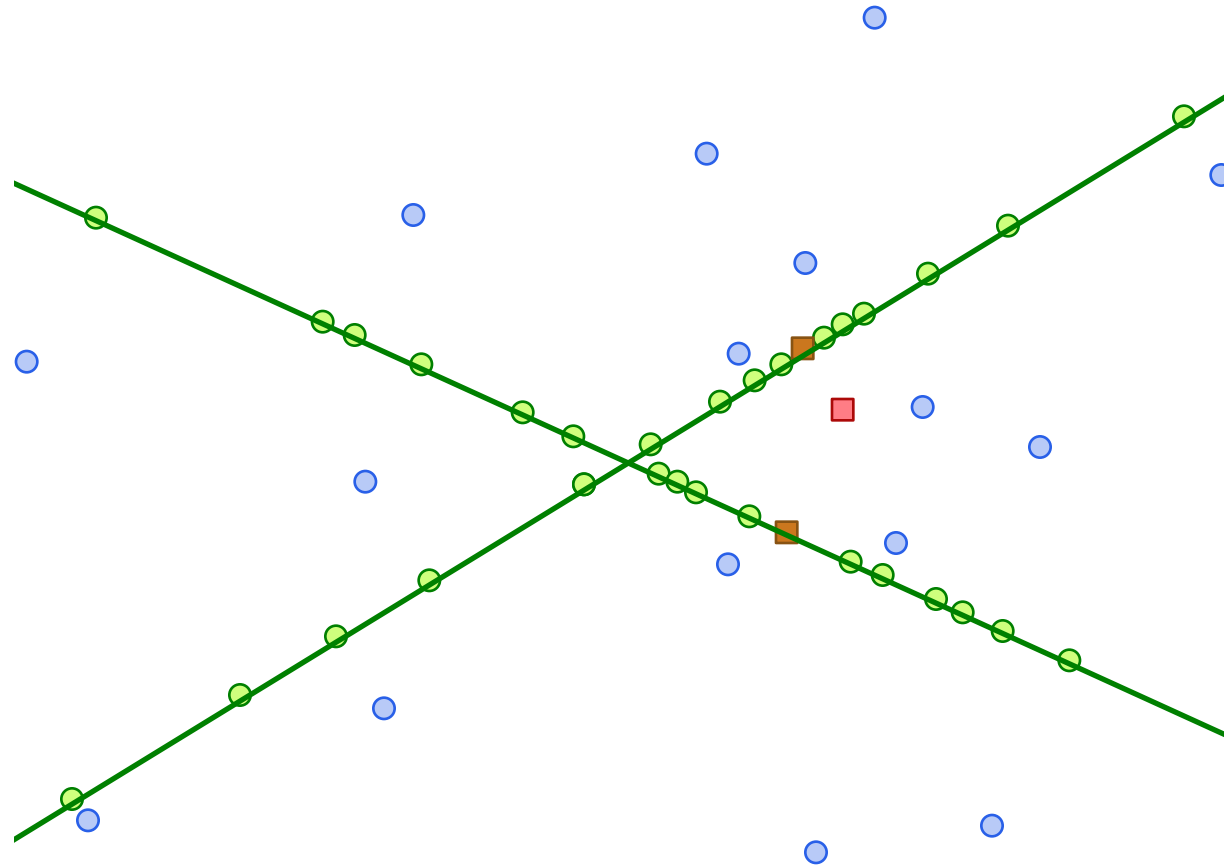
Project all data points along a random direction.

Prioritized DCI



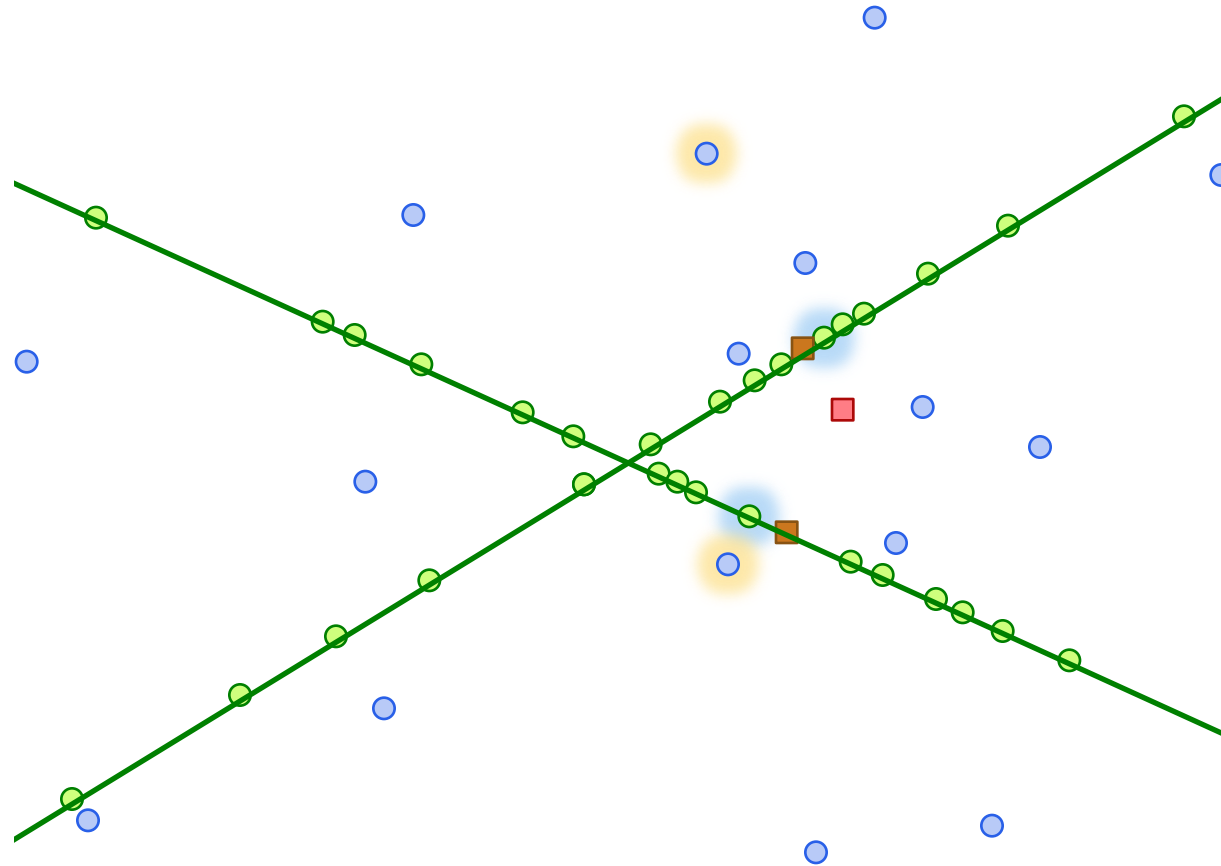
Project all data points along multiple random directions.

Prioritized DCI



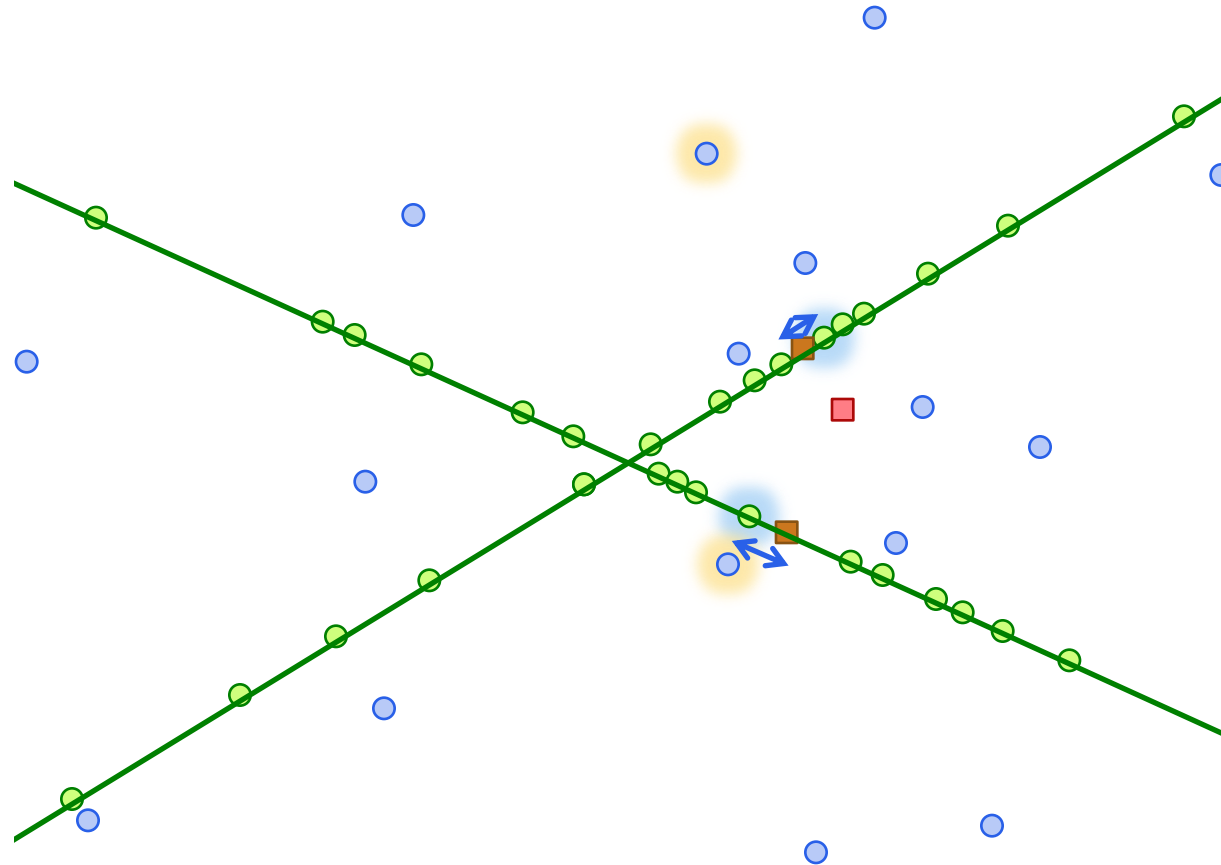
Project the query along each projection direction.

Prioritized DCI



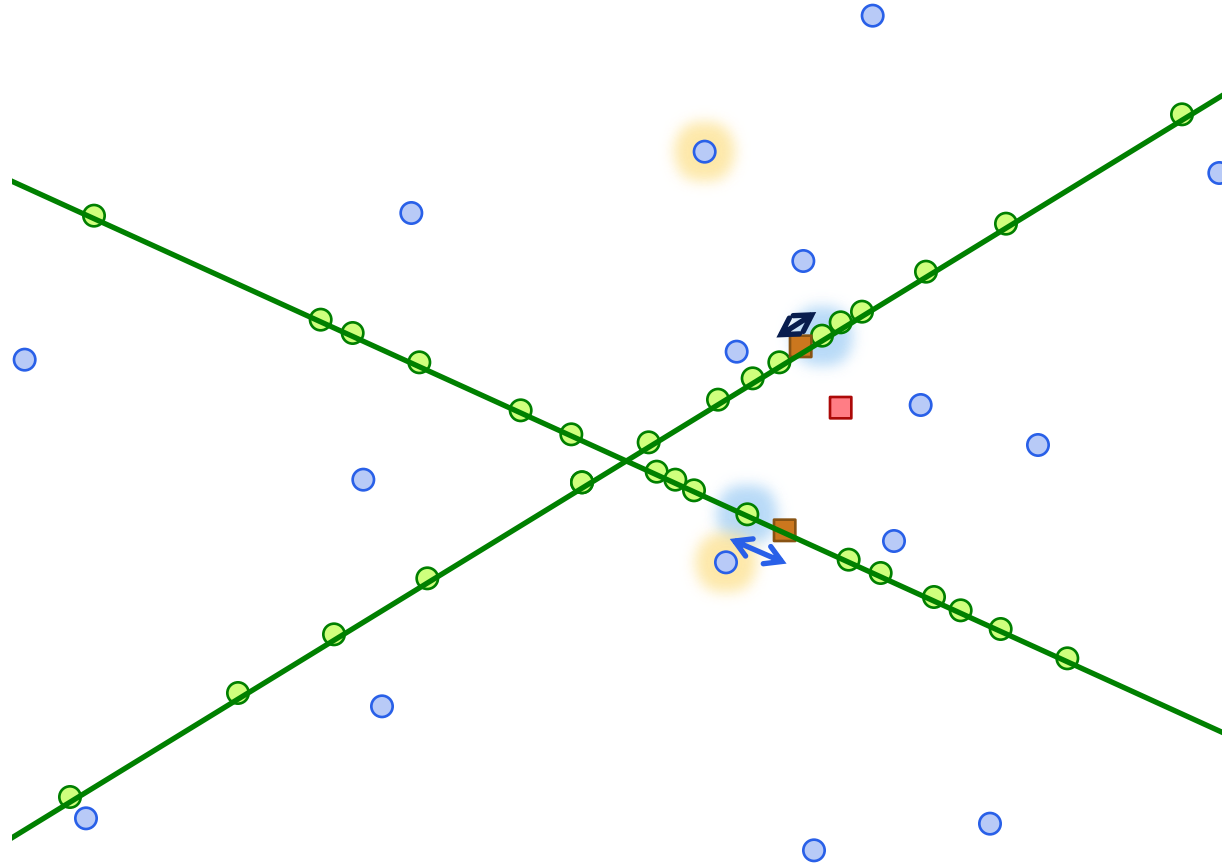
Find the closest point to the query along each projection direction and add them to the frontier.

Prioritized DCI



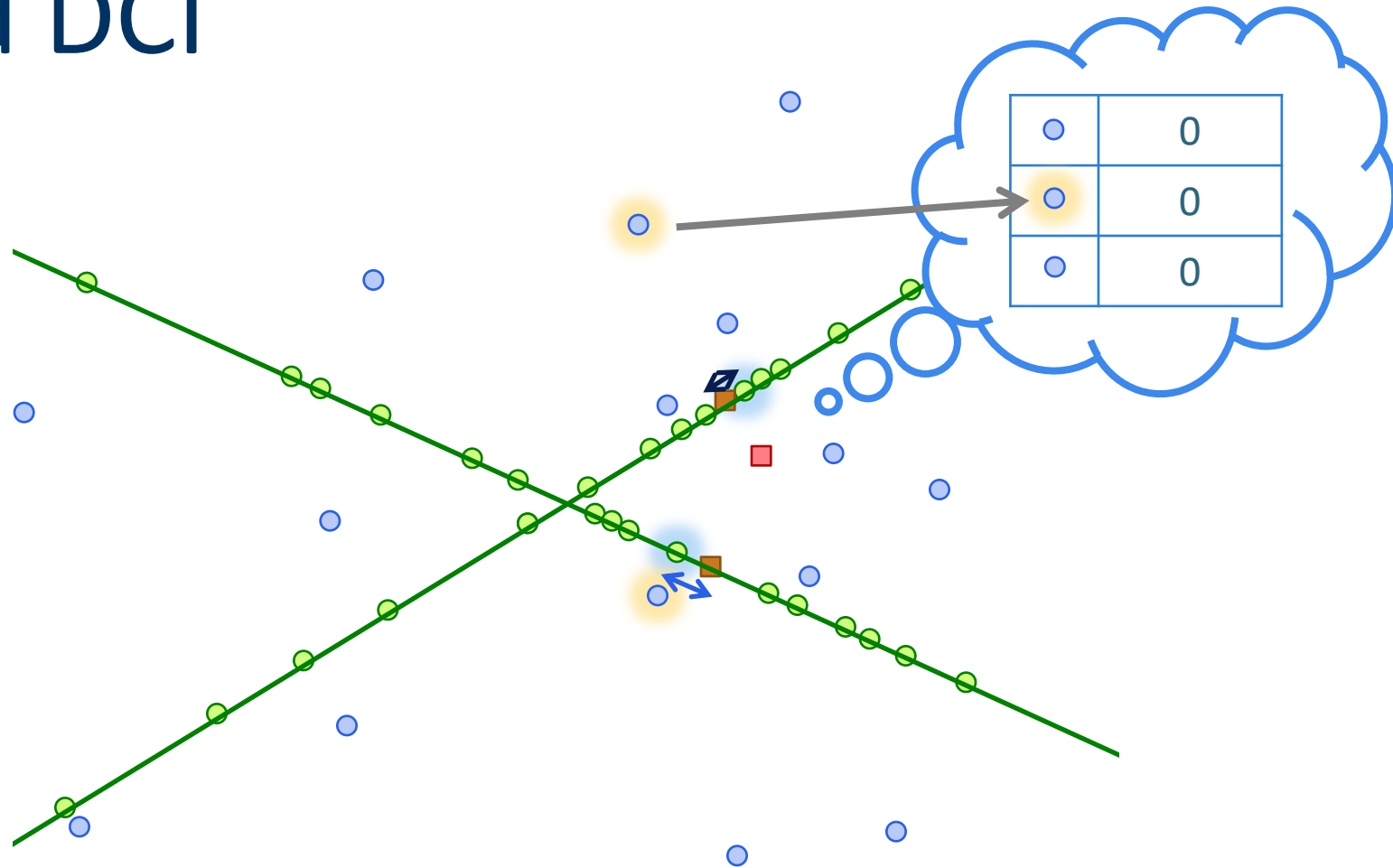
Compare their projected distances to the query.

Prioritized DCI



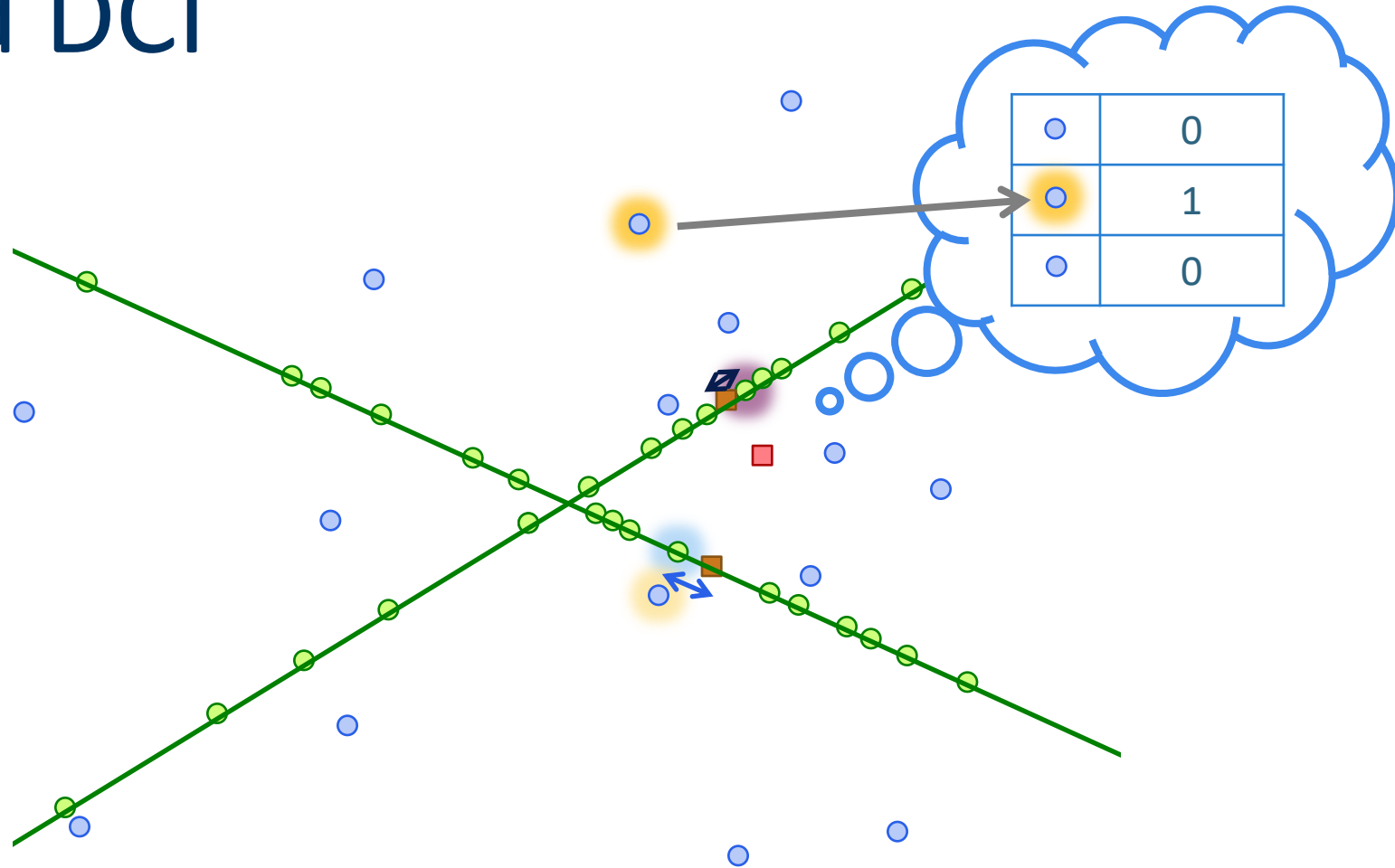
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



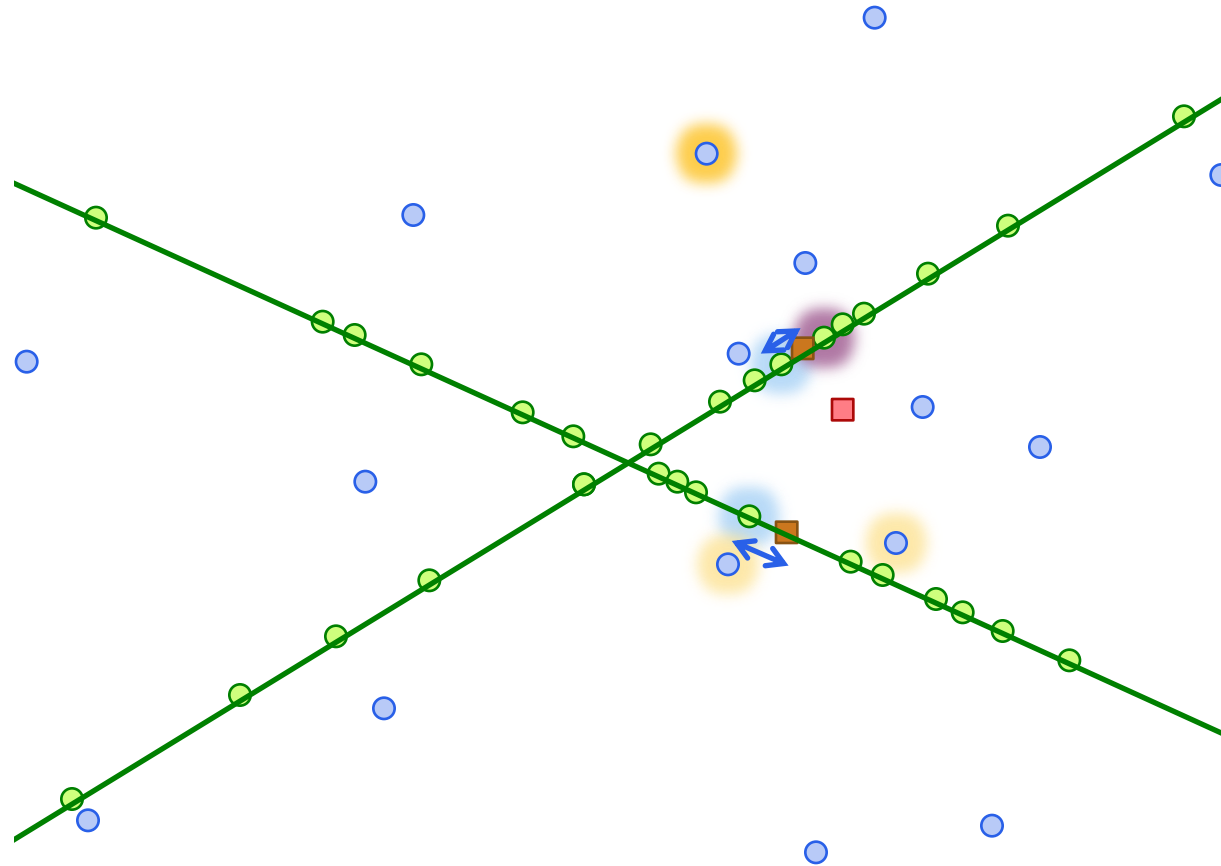
Maintain a hash table for the number of times we have come across each data point.

Prioritized DCI



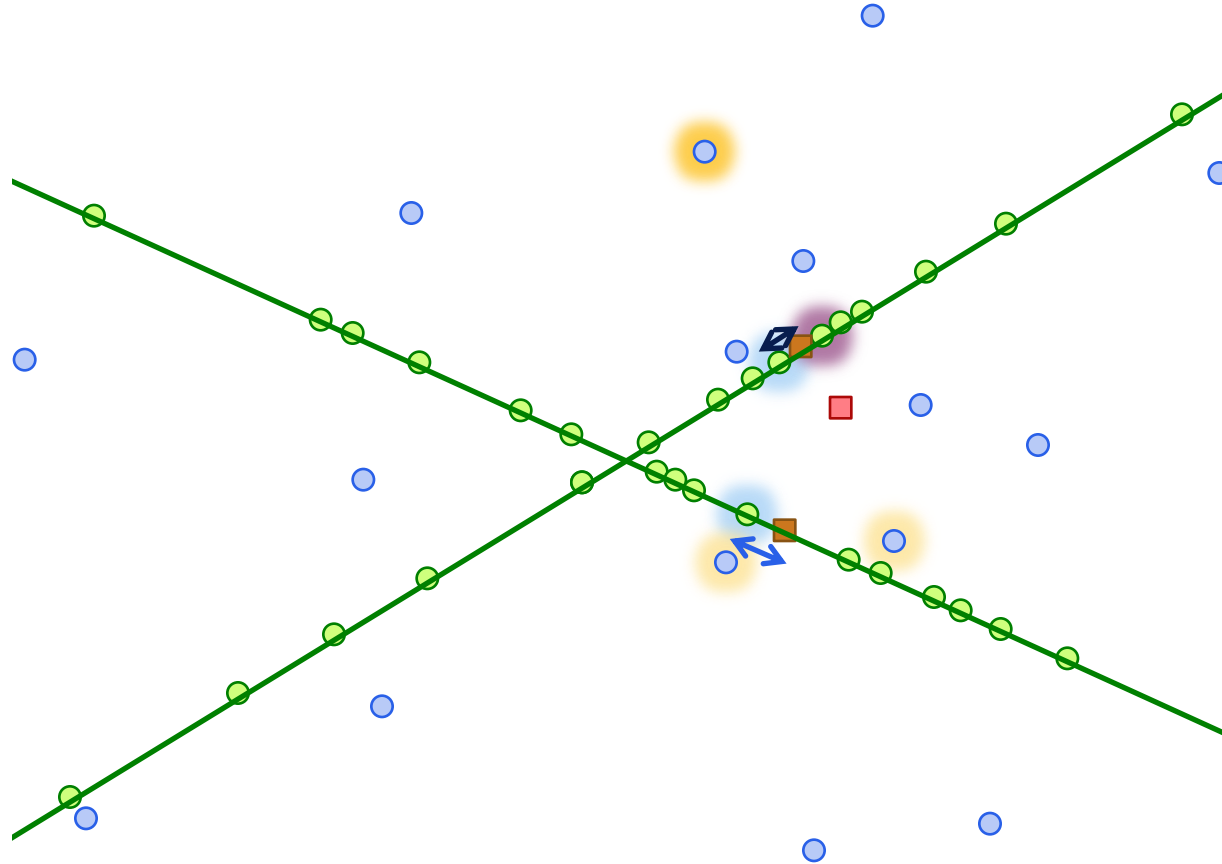
Visit this point and increment the counter for this point in the hash table.

Prioritized DCI



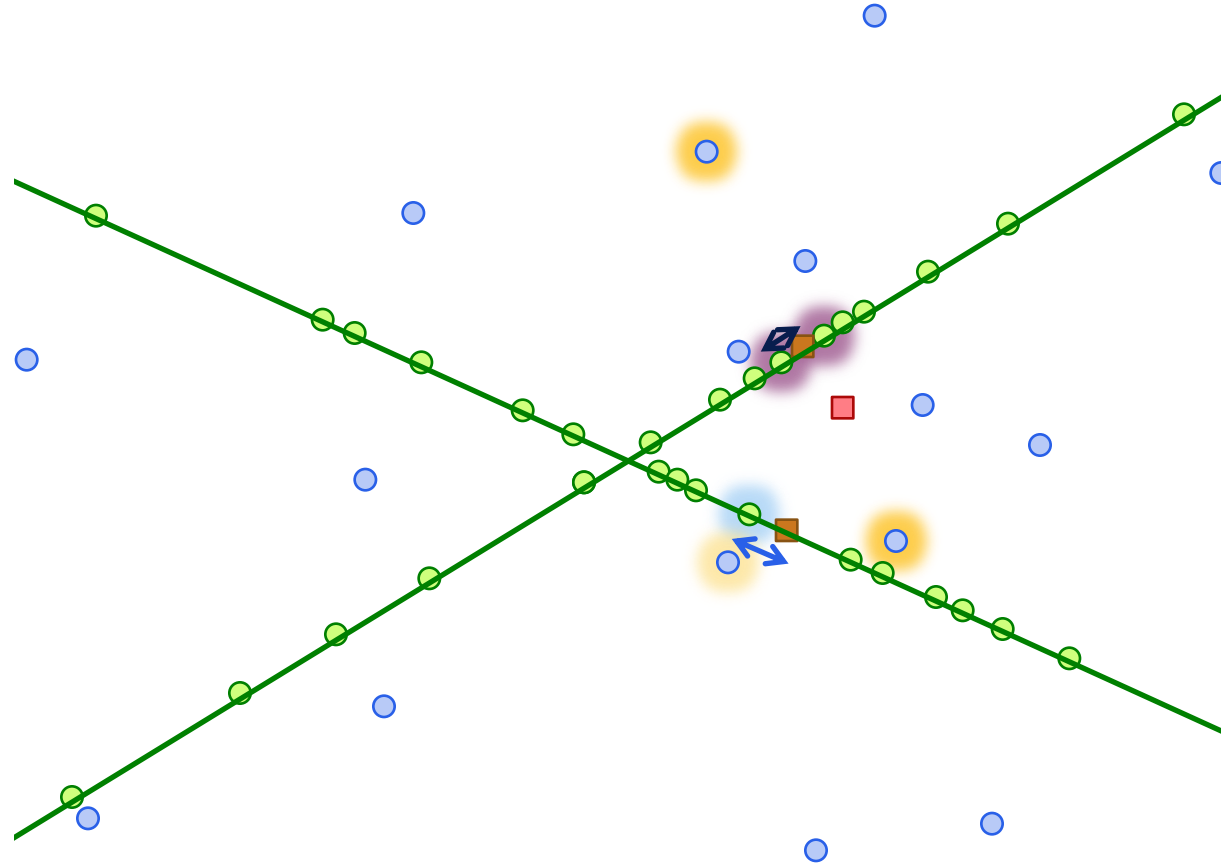
Find the next closest point along the projection direction that has just been processed and add it to the frontier.

Prioritized DCI



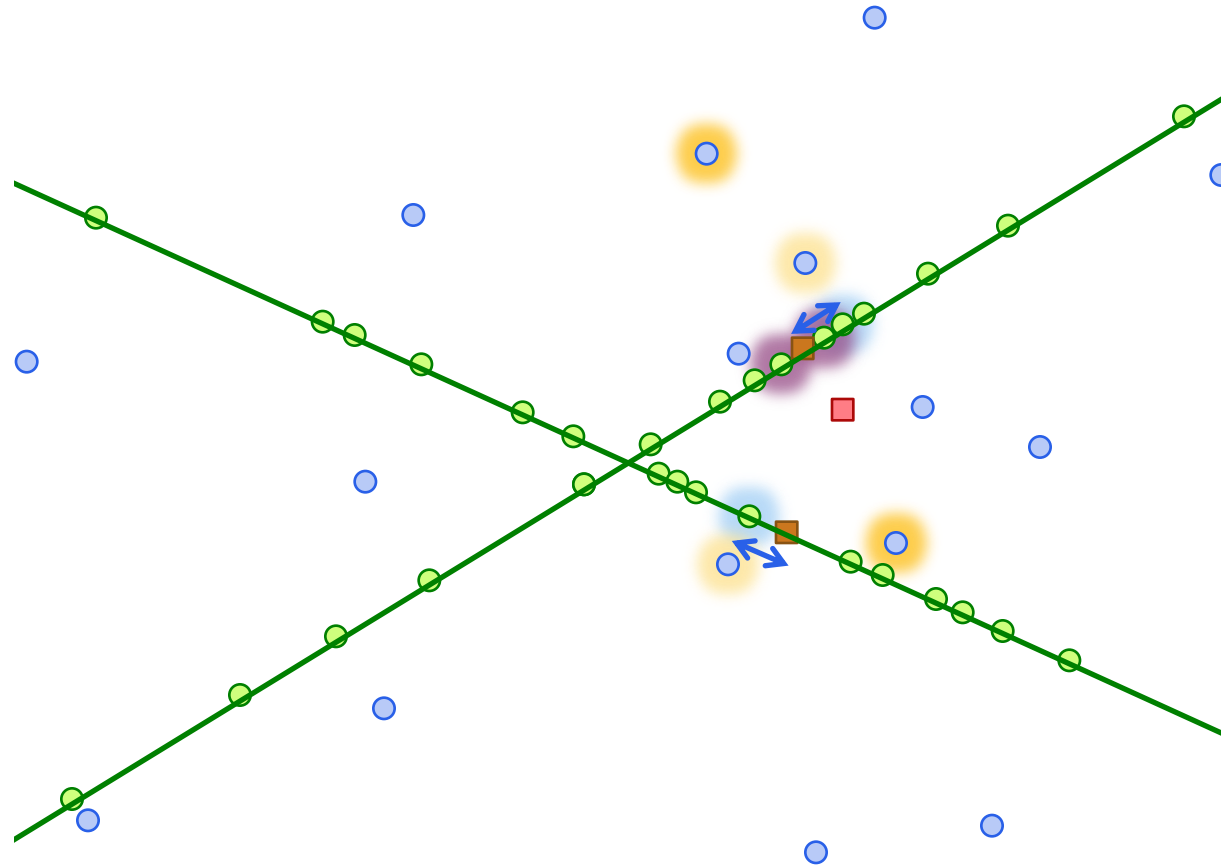
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



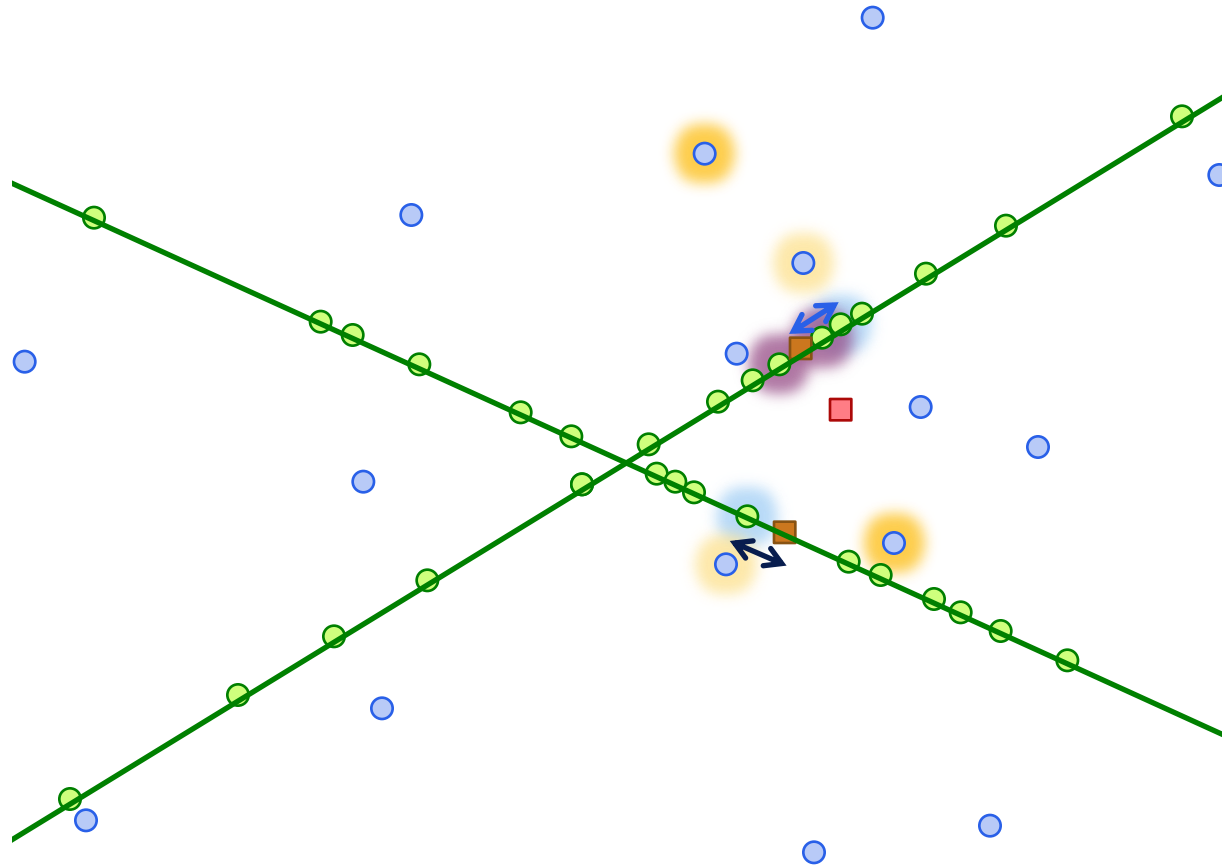
Visit this point.

Prioritized DCI



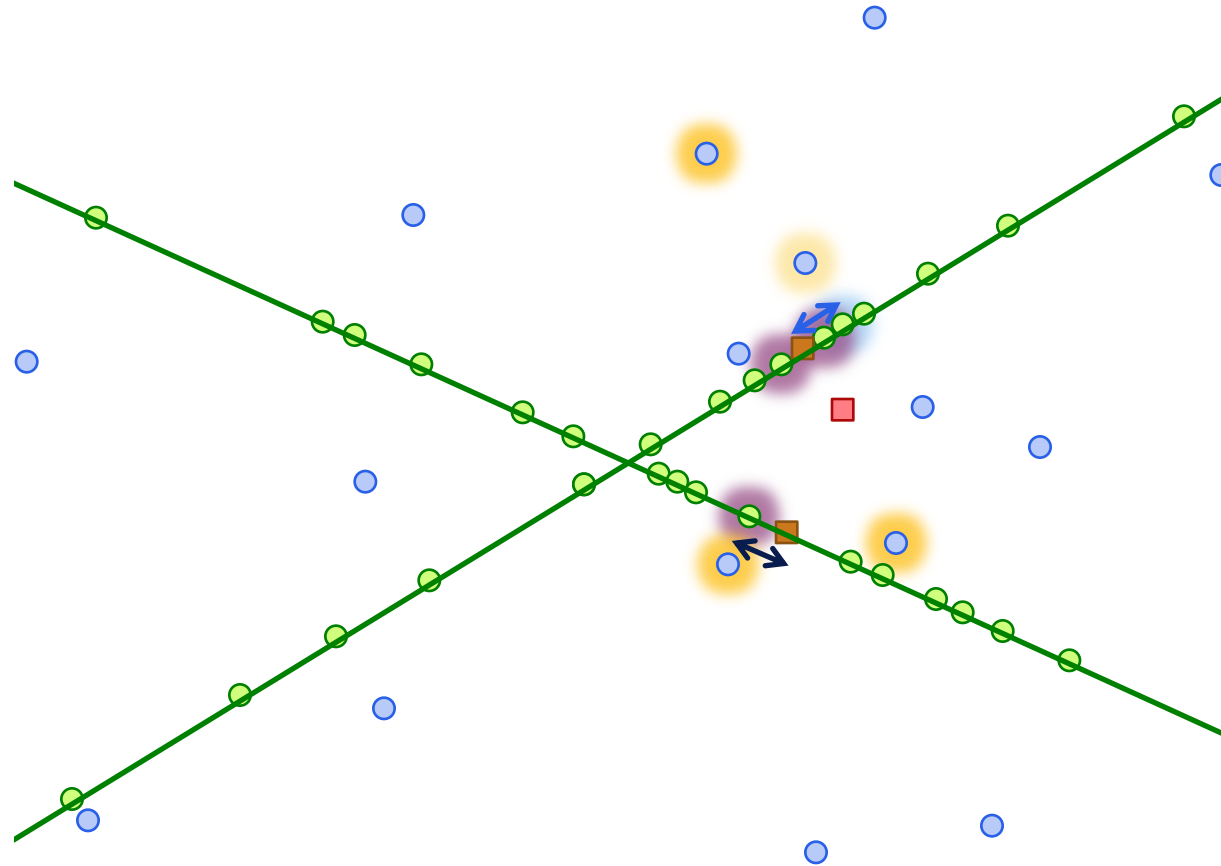
Find the next closest point along the projection direction that has just been processed and add it to the frontier.

Prioritized DCI



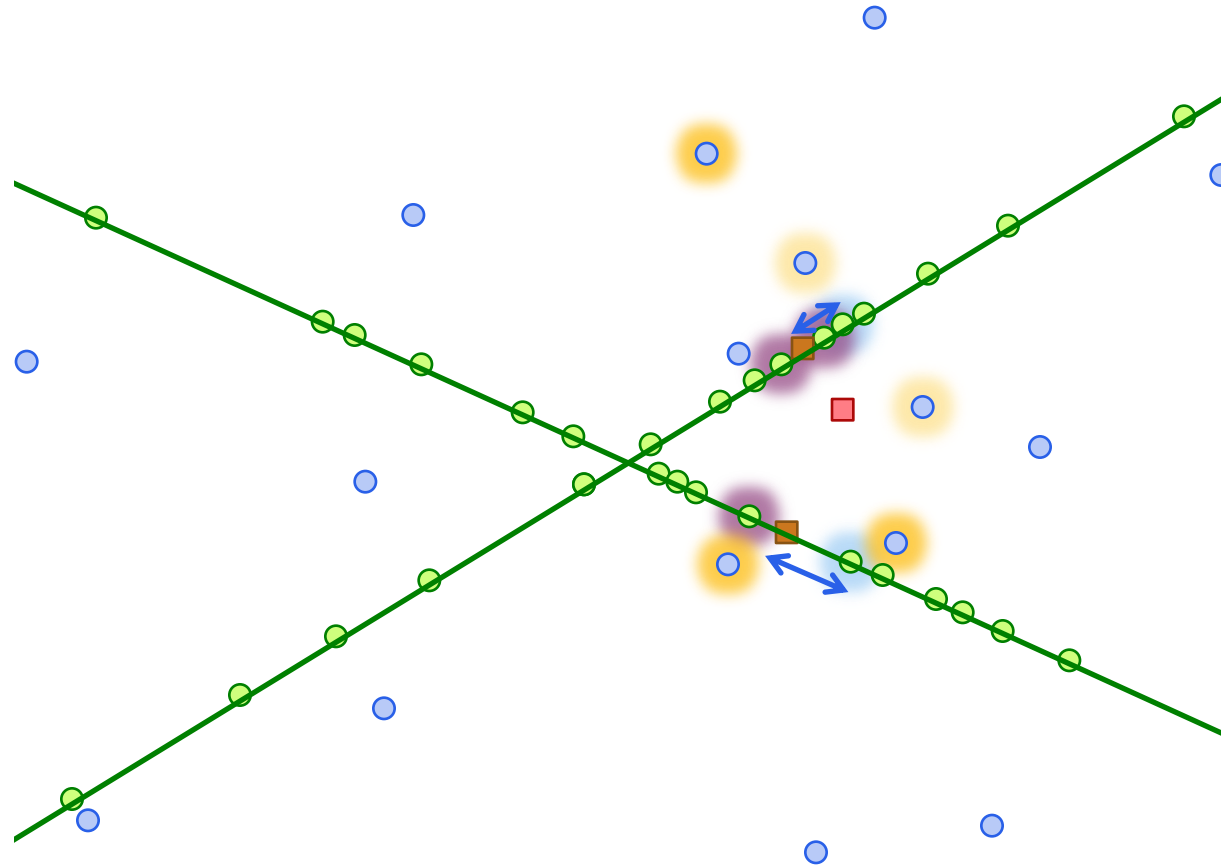
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



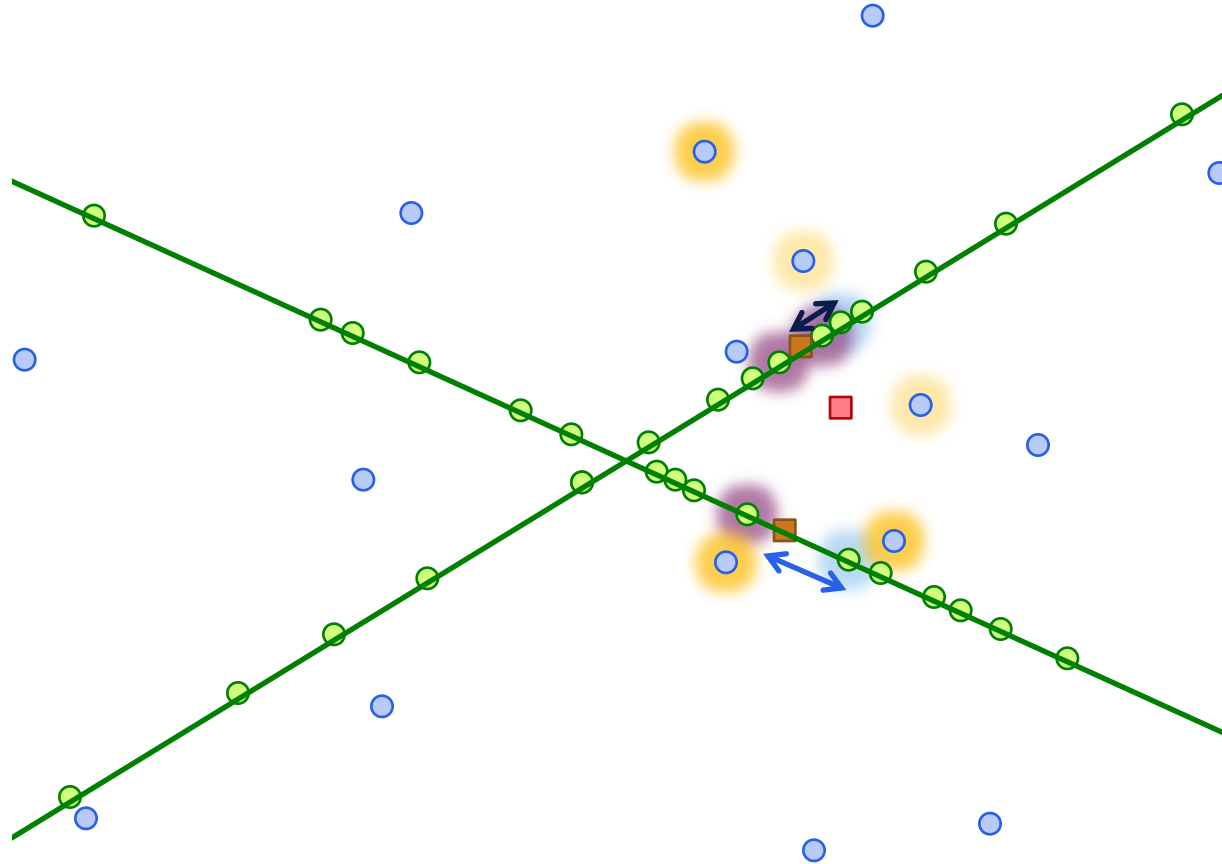
Visit this point.

Prioritized DCI



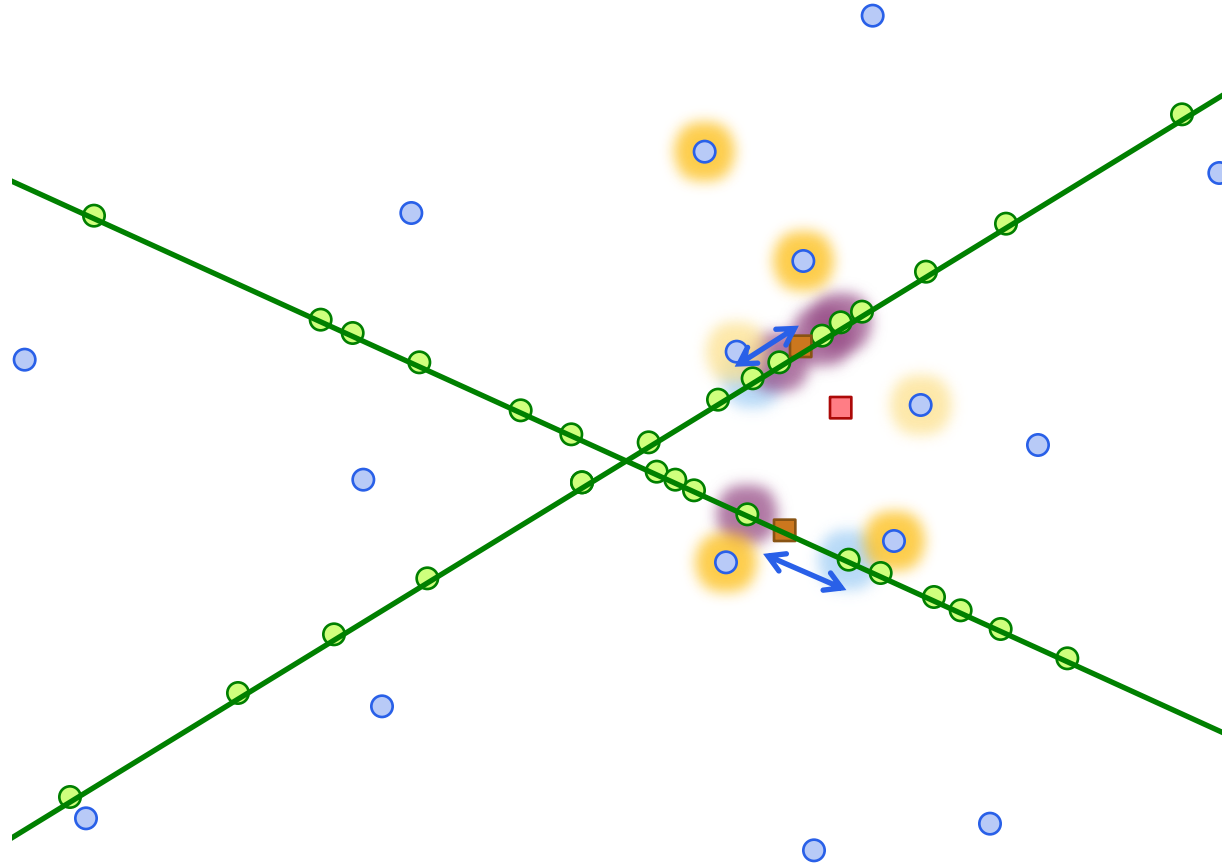
Find the next closest point along the projection direction that has just been processed and add it to the frontier.

Prioritized DCI



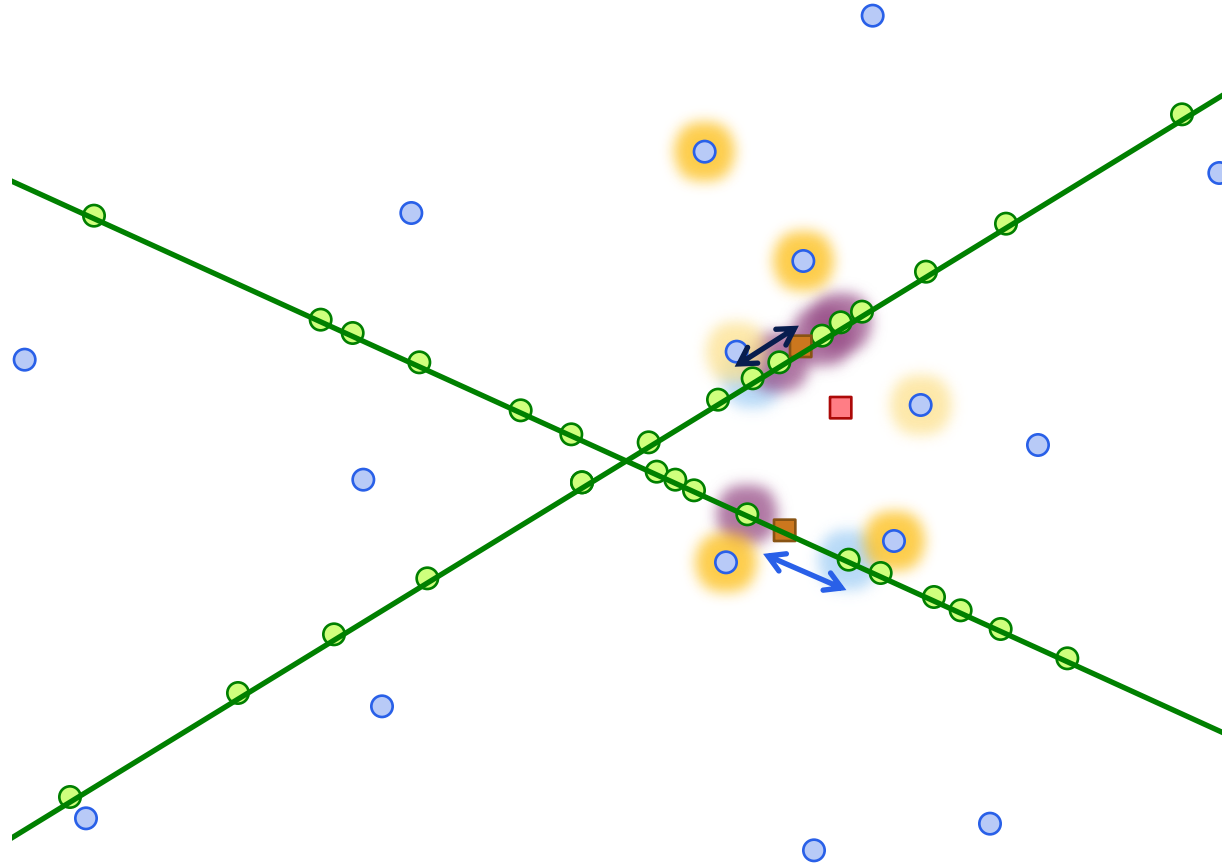
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



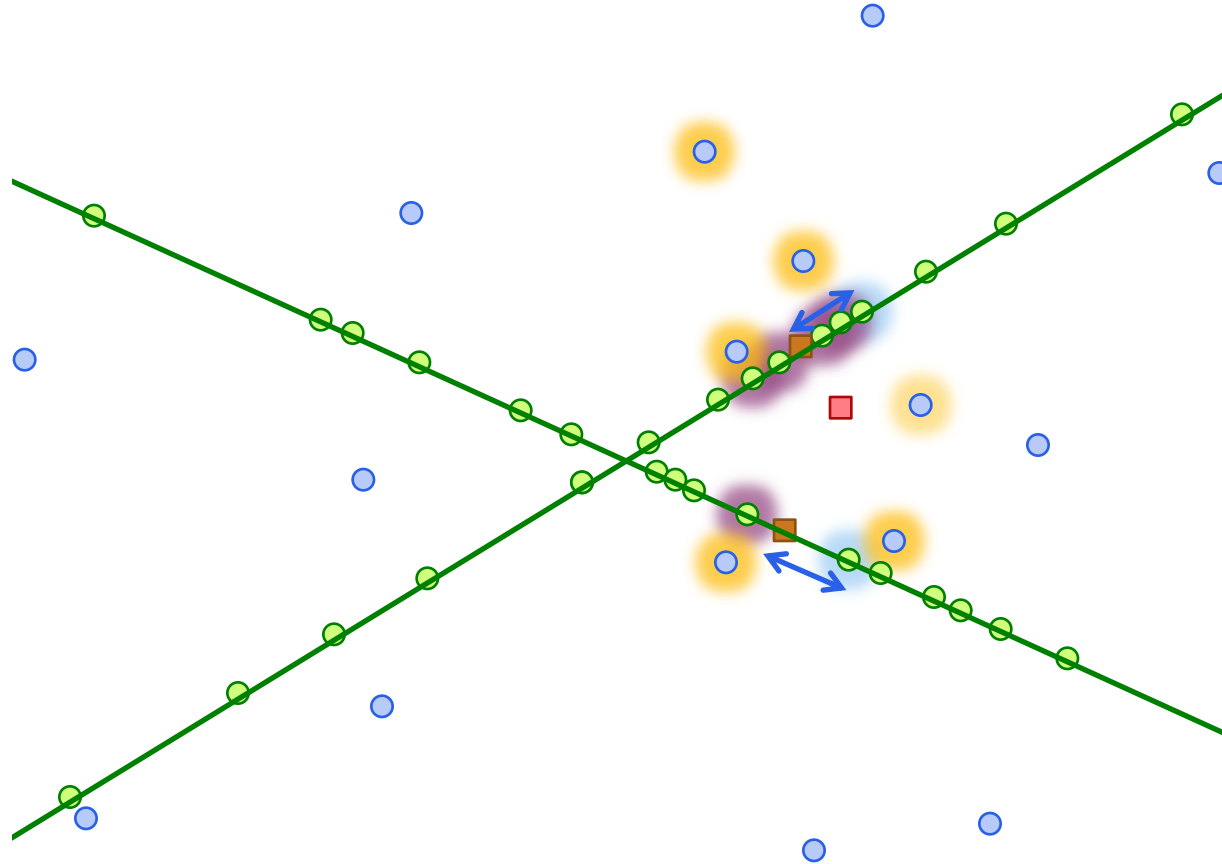
Visit this point and add the next point to the frontier.

Prioritized DCI



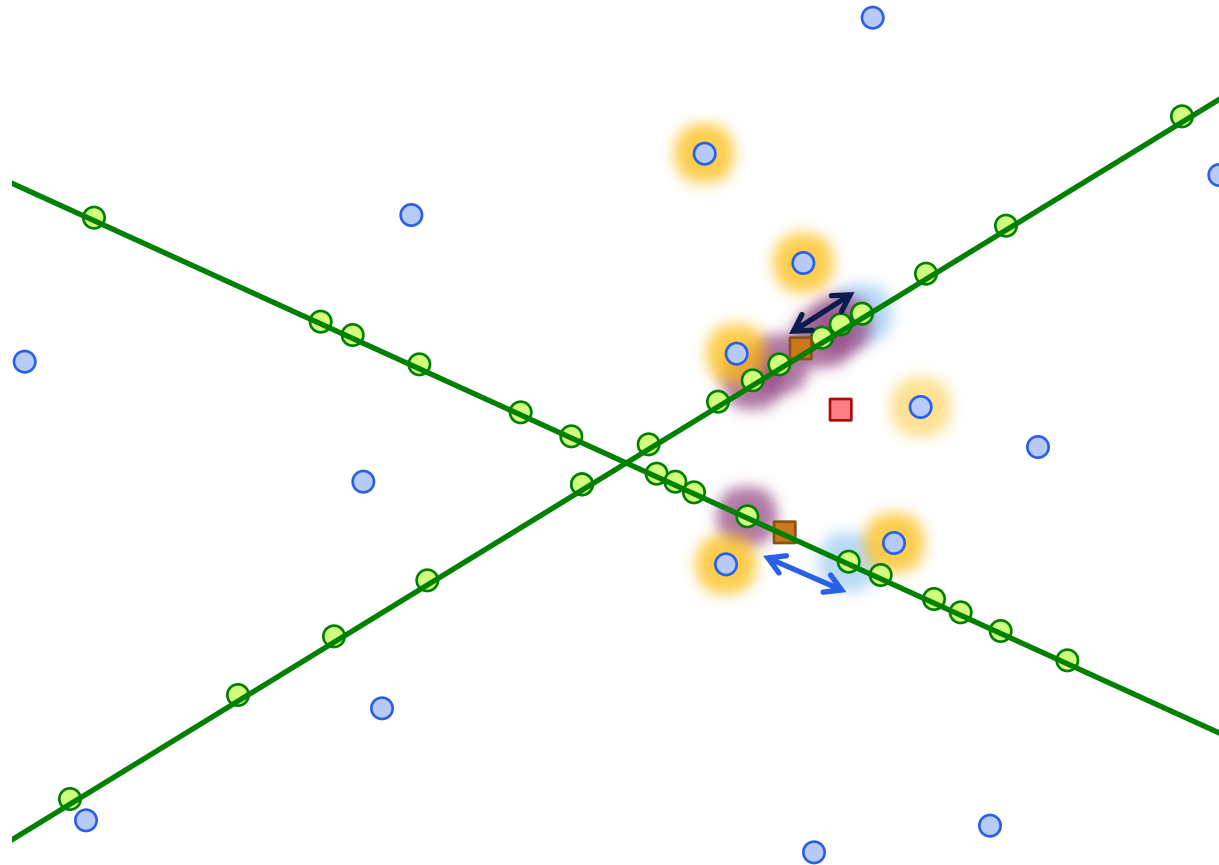
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



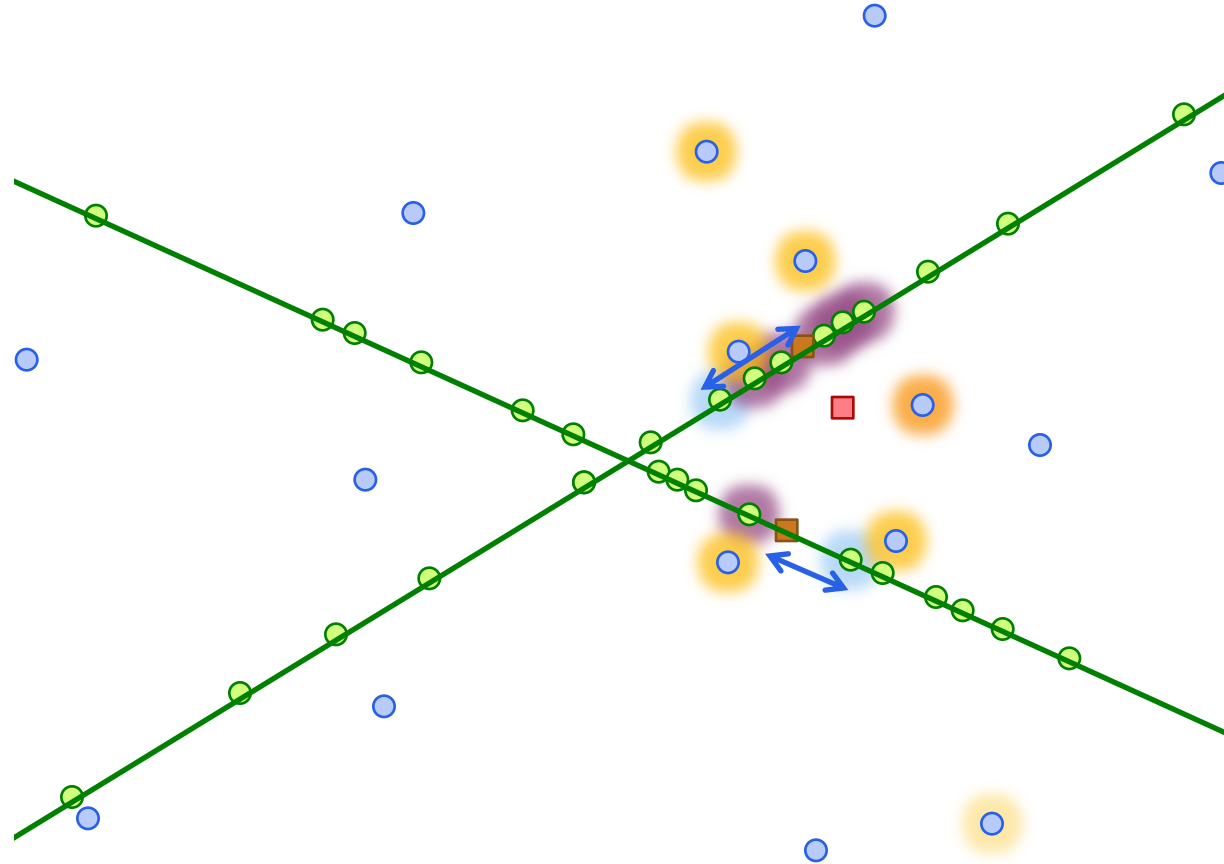
Visit this point and add the next point to the frontier.

Prioritized DCI



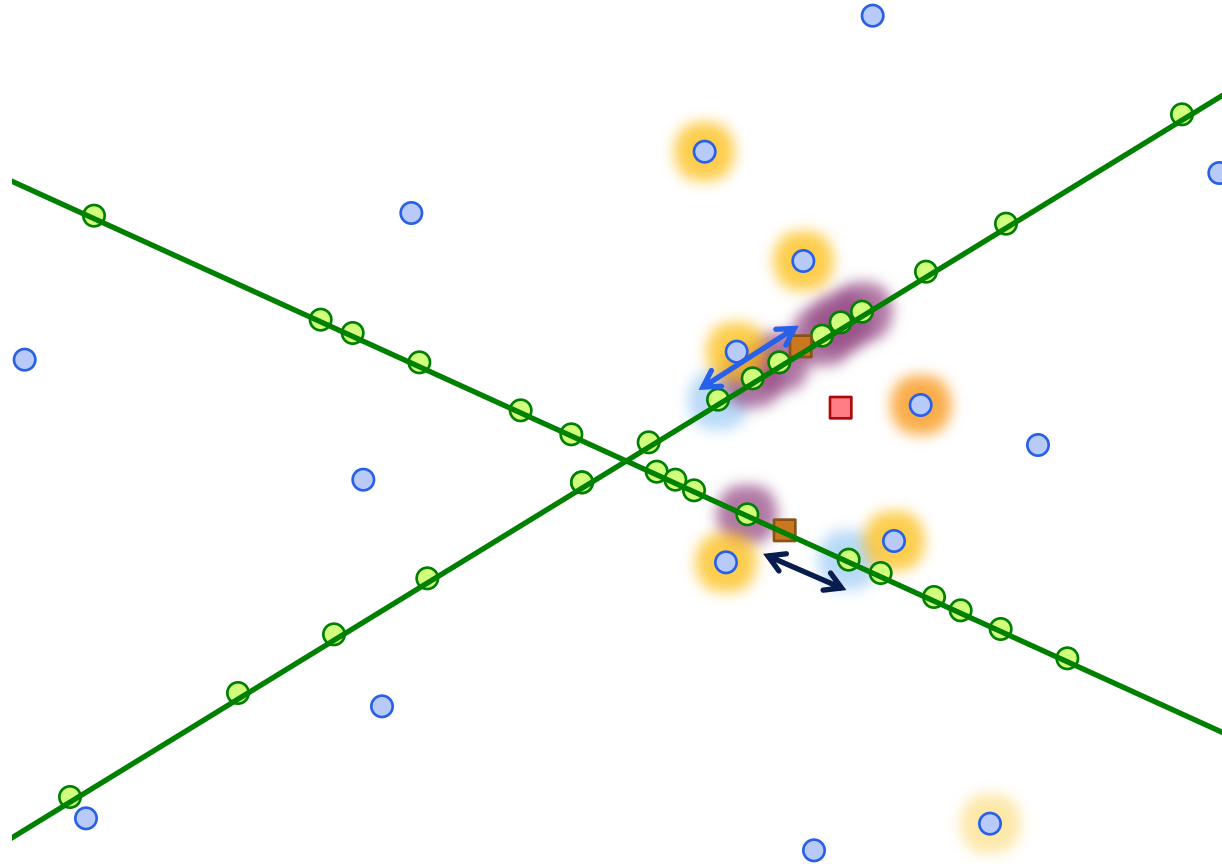
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



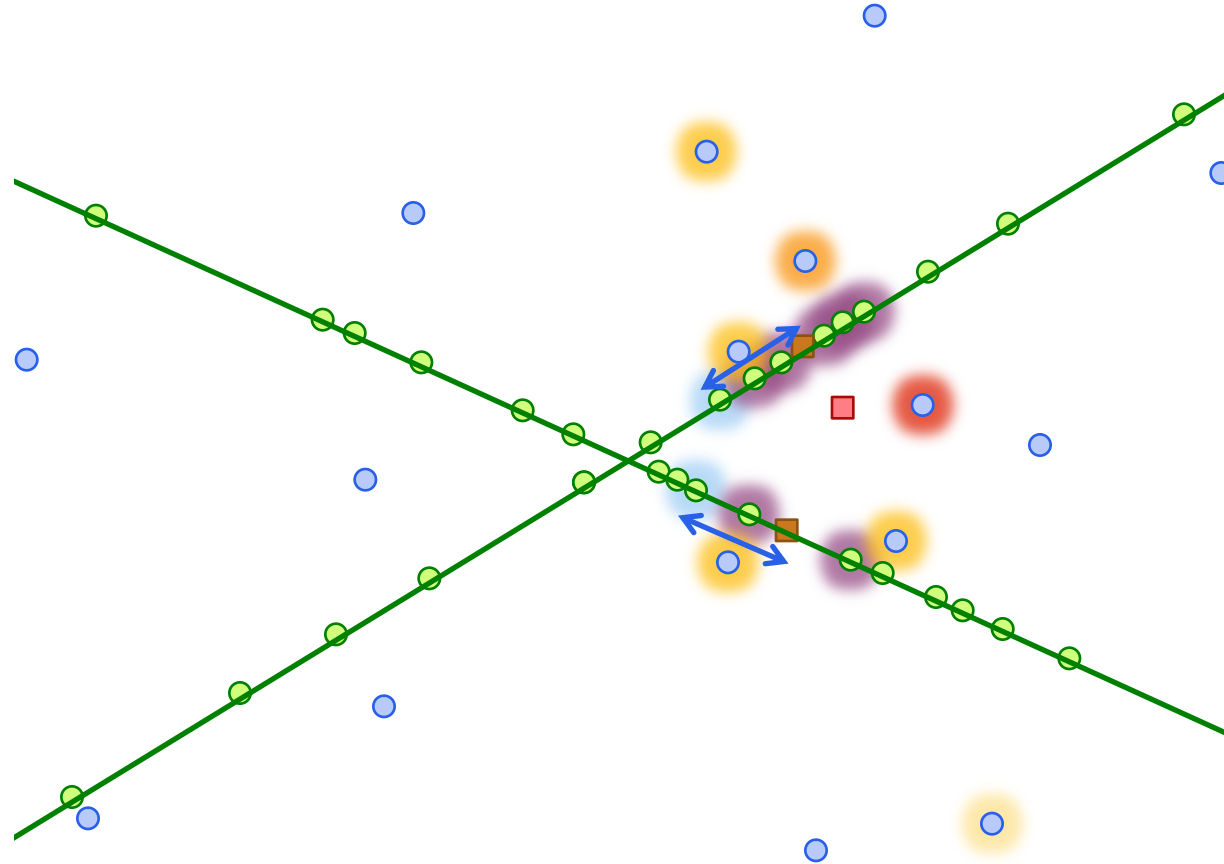
Visit this point and add the next point to the frontier.

Prioritized DCI



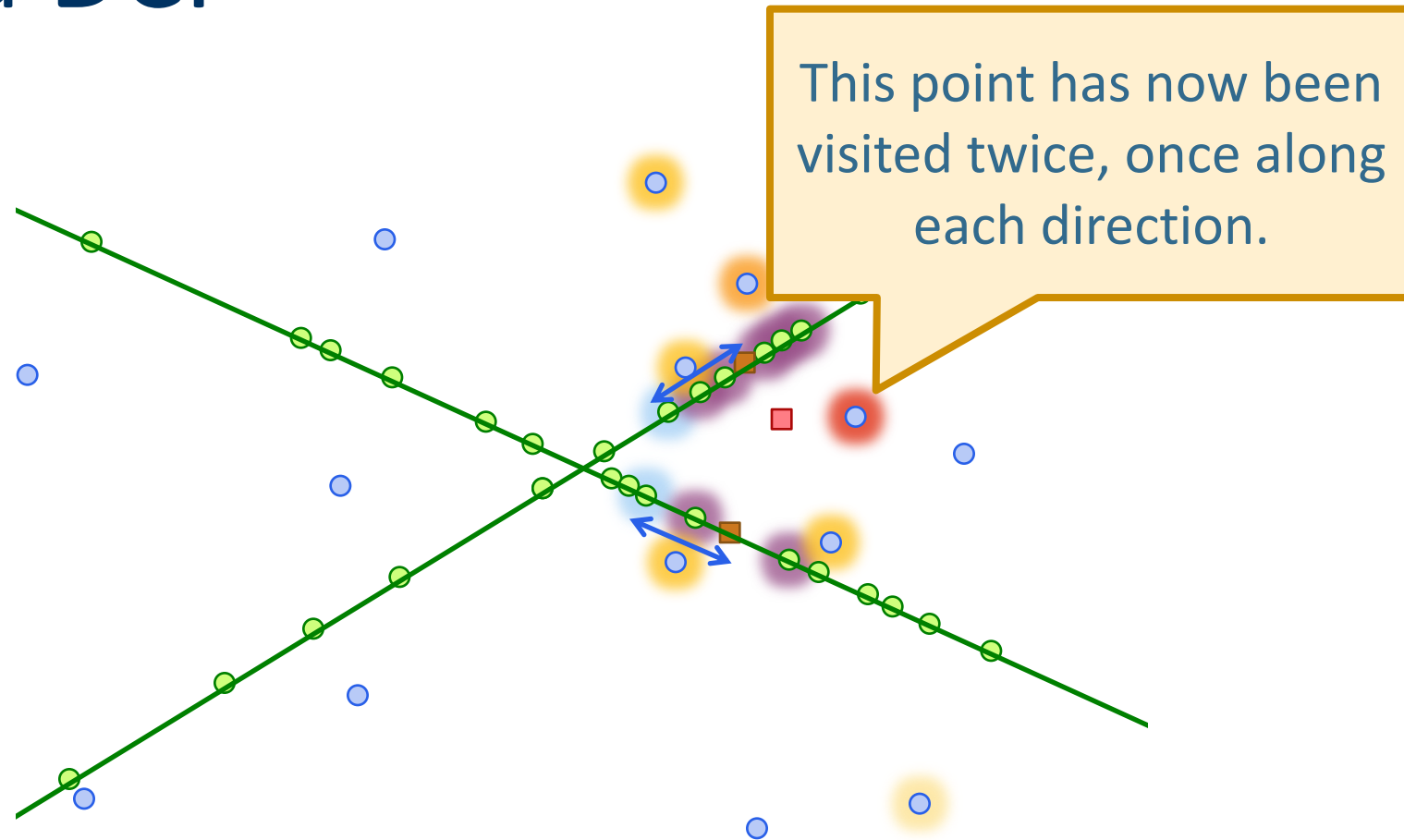
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



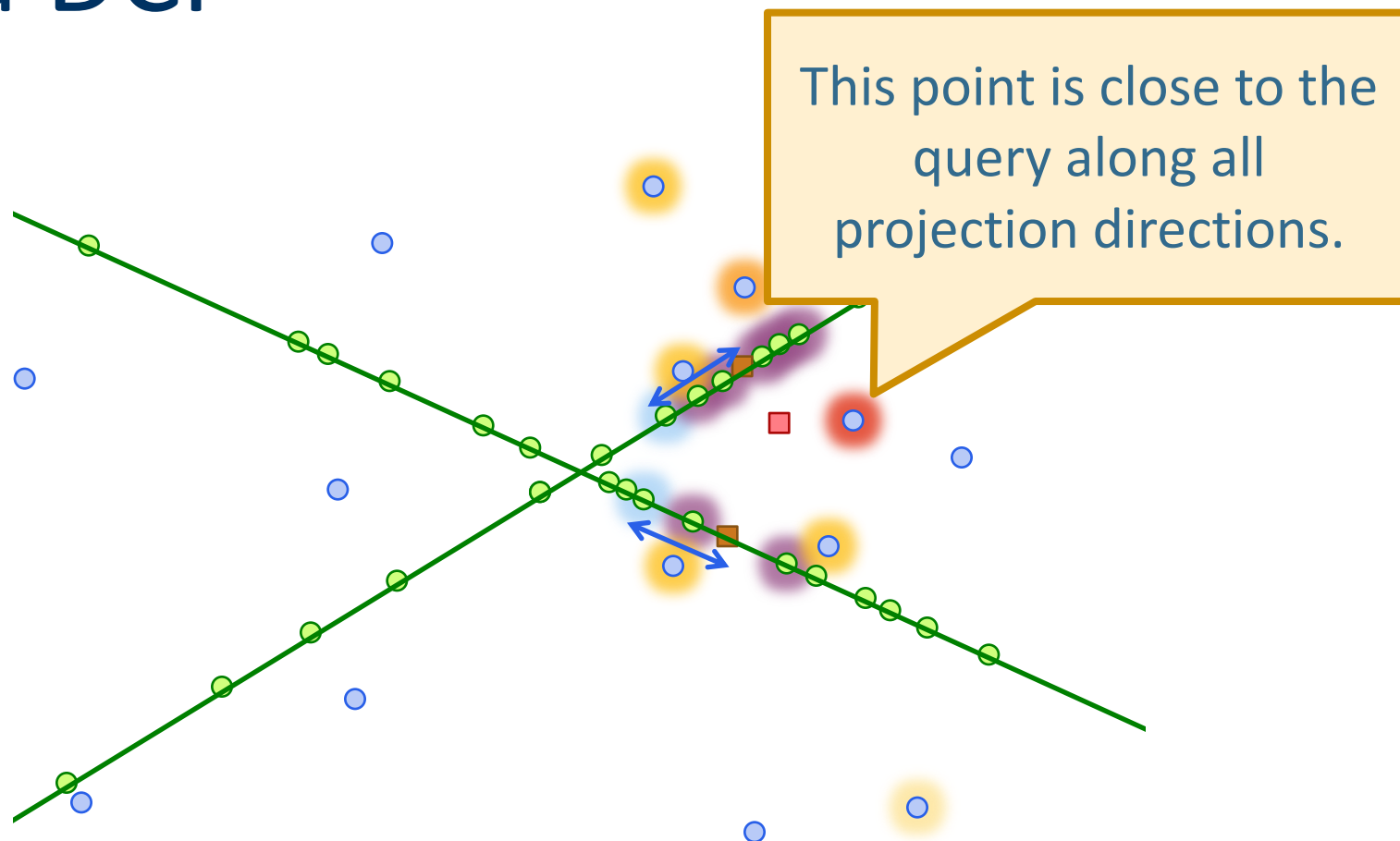
Visit this point and add the next point to the frontier.

Prioritized DCI



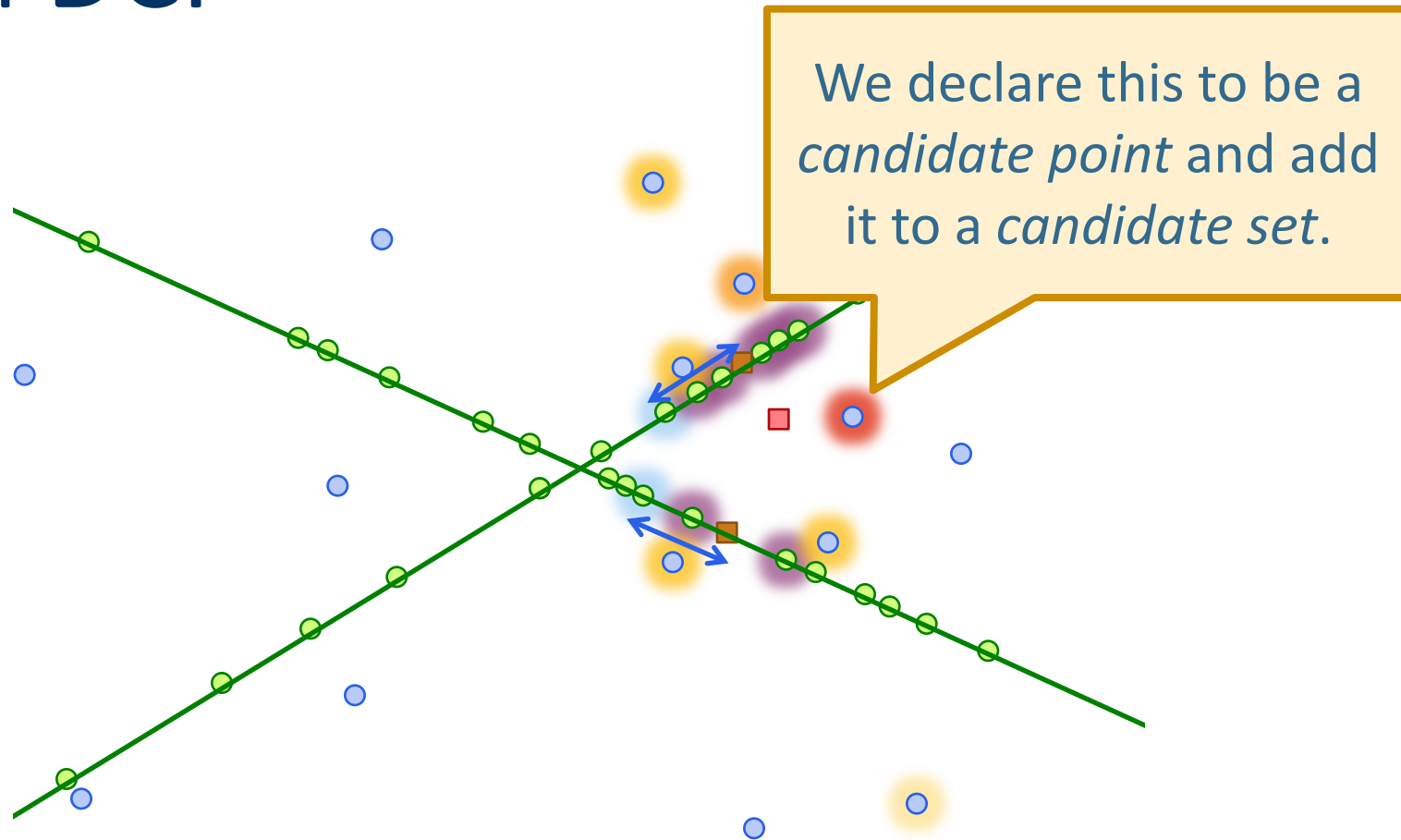
Visit this point and add the next point to the frontier.

Prioritized DCI



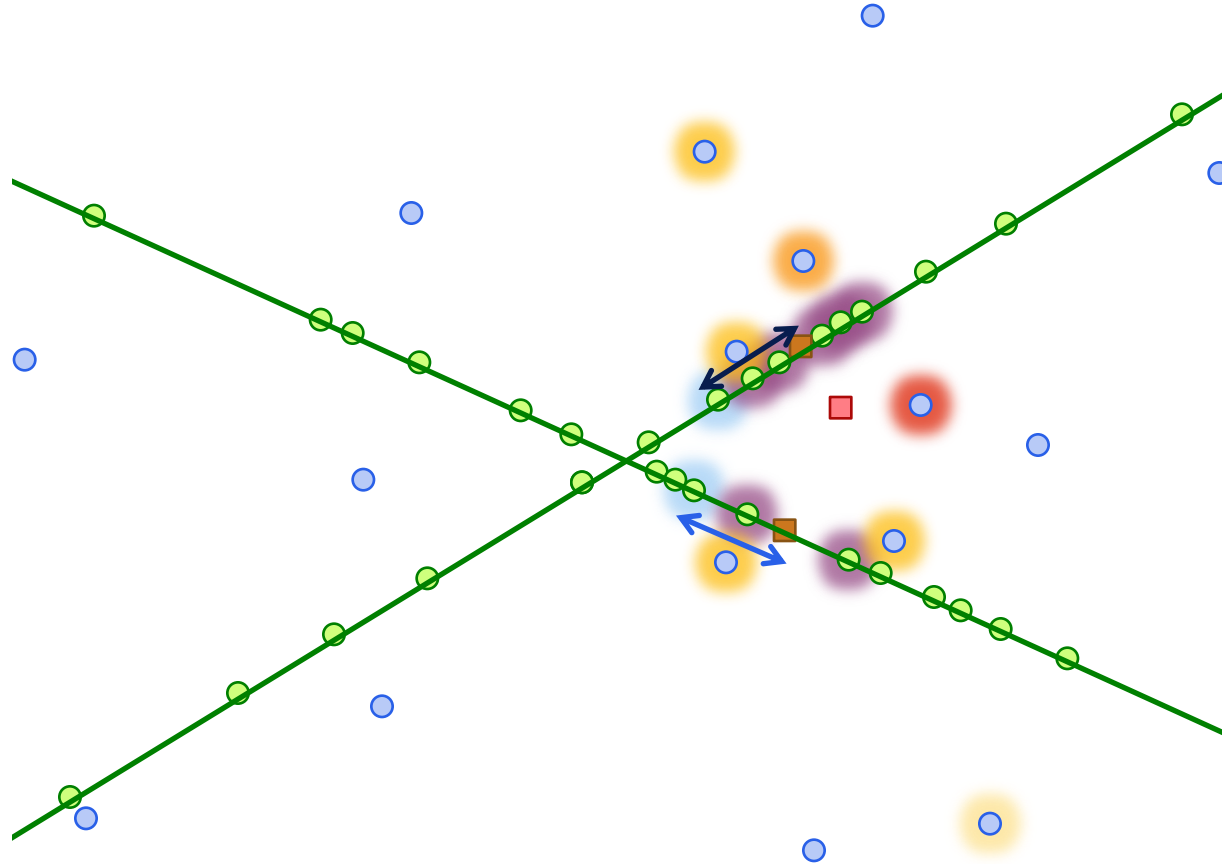
Visit this point and add the next point to the frontier.

Prioritized DCI



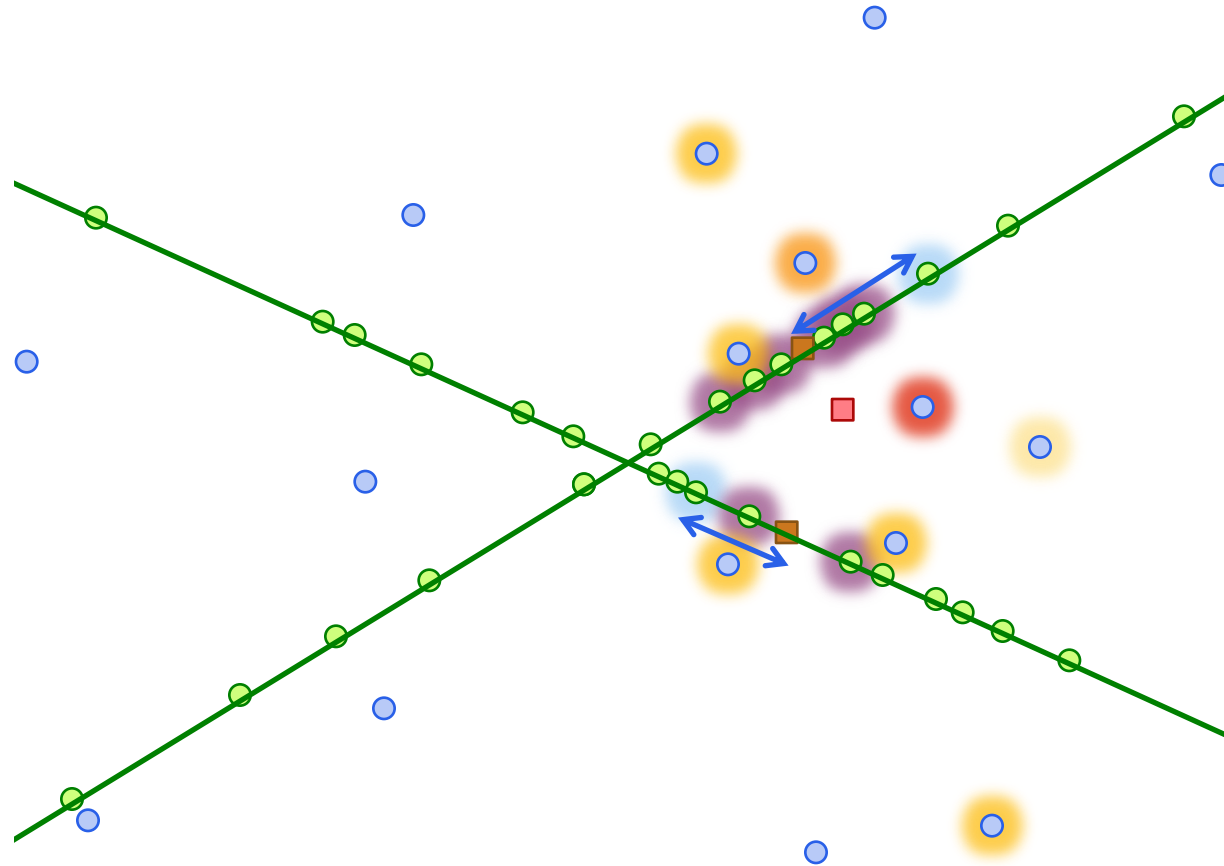
Visit this point and add the next point to the frontier.

Prioritized DCI



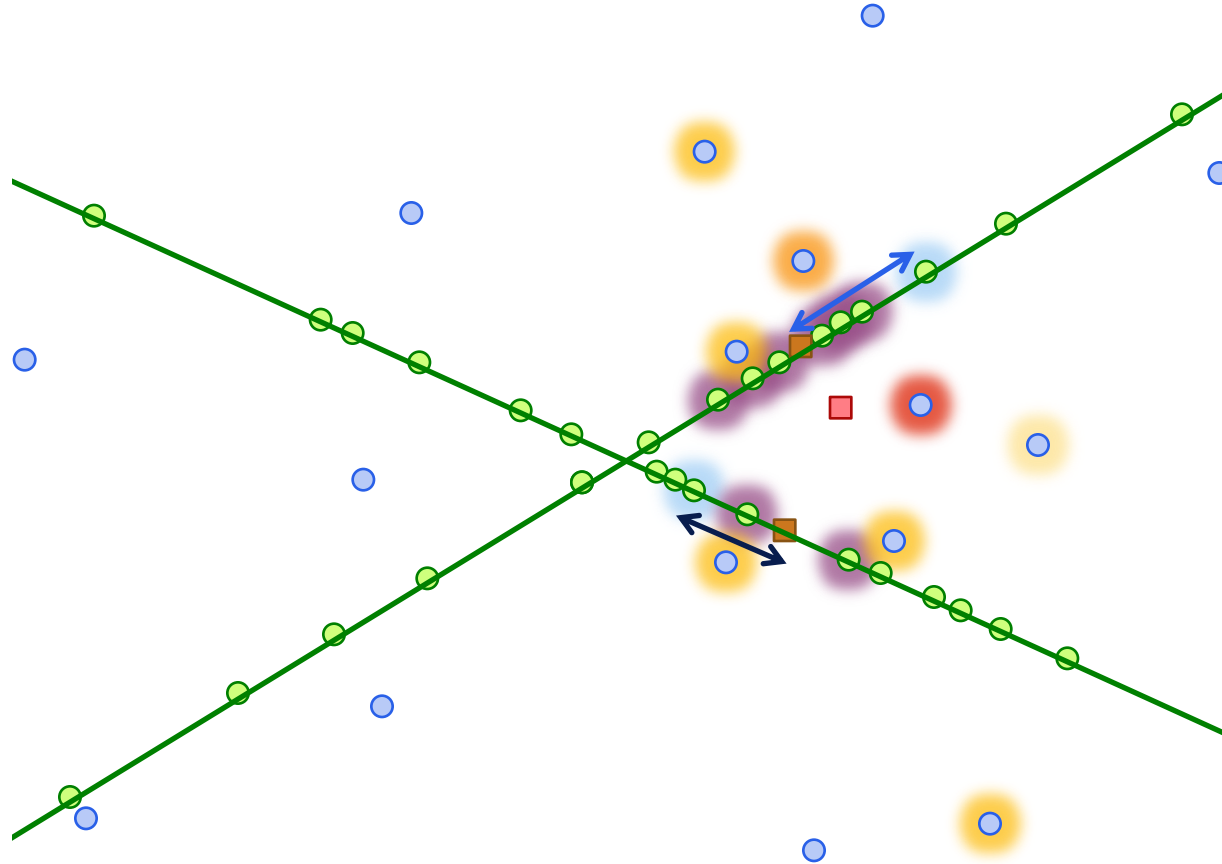
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



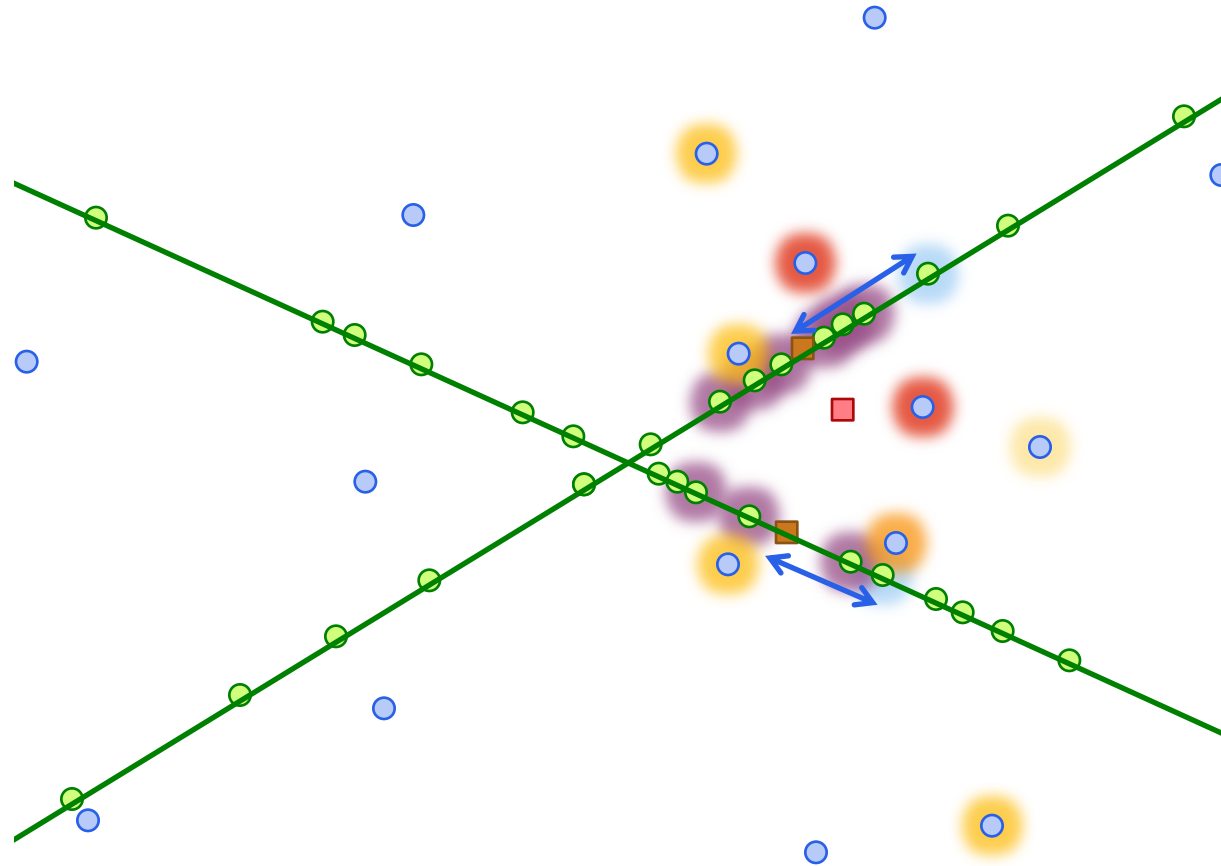
Visit this point and add the next point to the frontier.

Prioritized DCI



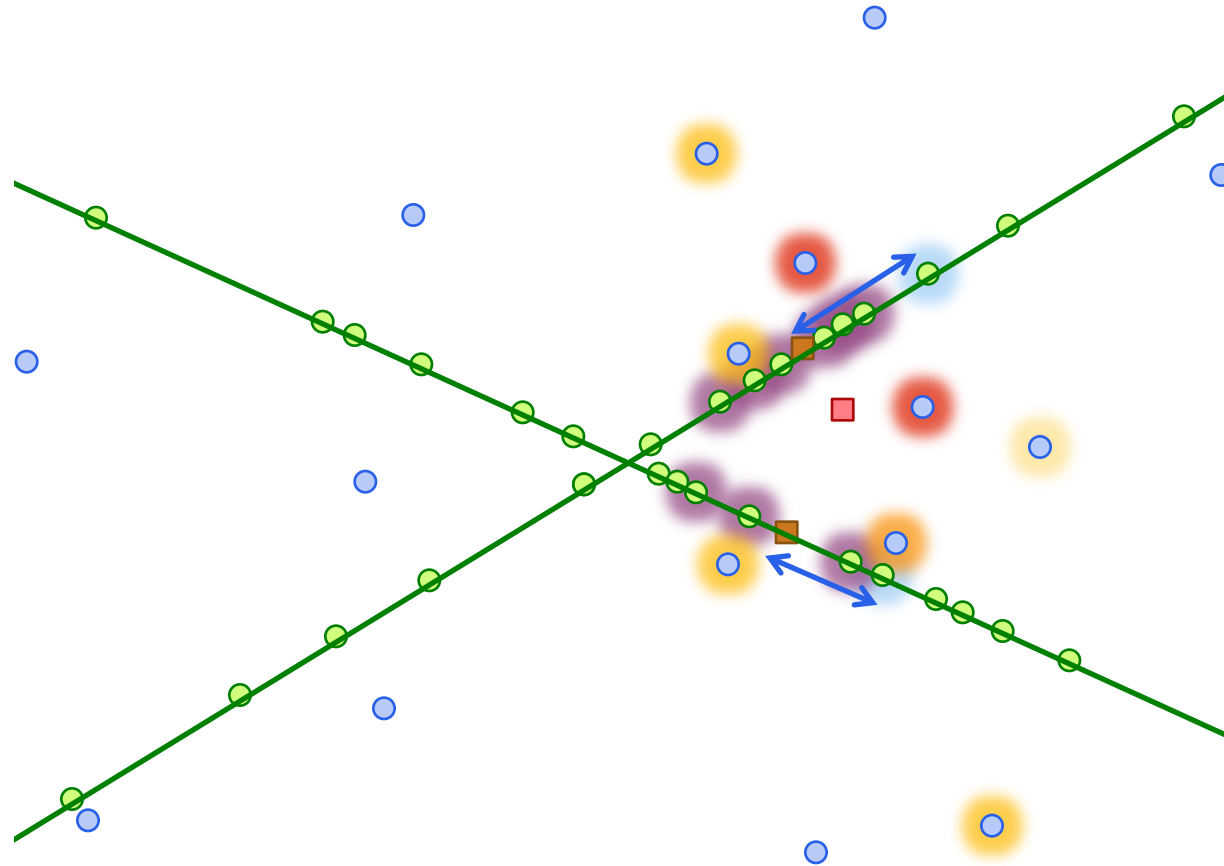
Pick the point on the frontier with the shortest projected distance.

Prioritized DCI



Visit this point (which now becomes a candidate point), add it to the candidate set and add the next point to the frontier.

Prioritized DCI



After we have sufficient candidate points, we return the k candidate points that are closest to the query.

Intuition

- Points are added to the candidate set in the order of their maximum projected distance to the query.
- Maximum projected distance is a lower bound on the true distance.
- As the number of projection directions increases, this lower bound approaches the true distance.

$$\max_j \{ |\langle p^i, u_j \rangle - \langle q, u_j \rangle| \} = \max_j \{ |\langle p^i - q, u_j \rangle| \} \leq \|p^i - q\|_2$$

where $\|u_j\|_2 = 1 \quad \forall j$

Complexity

- Query Time:

$$O \left((d(m + k \max((n/k)^{1-m/d'}, \log(n/k))) + mk \log m \max((n/k)^{1-1/d'}, \log(n/k))) \log(1/\epsilon) \right)$$

where:

- $m \geq 1$ is the number of projection directions chosen by the user.
- k is the number of neighbours.
- n is the number of data points.

Complexity

- Query Time:

$$O \left((d(m + k \max((n/k)^{1-m/d'}, \log(n/k))) + mk \log m \max((n/k)^{1-1/d'}, \log(n/k))) \log(1/\epsilon) \right)$$

where:

- $m \geq 1$ is the number of projection directions chosen by the user.
- k is the number of neighbours.
- n is the number of data points.

Complexity

- Query Time:

$$O \left(d \left(m + k \max \left(\left(\frac{n}{k} \right)^{1-m/d'}, \log(n/k) \right) \right) + \right. \\ \left. nk \log m \max \left(\left(\frac{n}{k} \right)^{1-1/d'}, \log(n/k) \right) \log(1/\epsilon) \right)$$

where

Linear dependence on
ambient dimensionality

- $m \geq 1$ is the number of projection directions chosen by the user.
- k is the number of neighbours.
- n is the number of data points.

Complexity

- Query Time:

$$O \left((d(m + k \max((n/k)^{1-m/d'}, \log(n/k))) + mk \log m \max((n/k)^{1-1/d'}, \log(n/k))) \log(1/\epsilon) \right)$$

where:

- $m \geq 1$ is the number of projection directions chosen by the user.
- k is the number of neighbours.
- n is the number of data points.

*Sublinear dependence on
intrinsic dimensionality*

Complexity

- Space: $O(mn)$
- Insertion Time: $O(m(d + \log n))$
- Deletion Time: $O(m \log n)$

Complexity

- Space: $O(mn)$
- Insertion Time: $O(m(d + \log n))$
- Deletion Time: $O(m \log n)$

Linear dependence on dataset size (cf. LSH, which has a near-quadratic dependence for high approximation quality)

Complexity

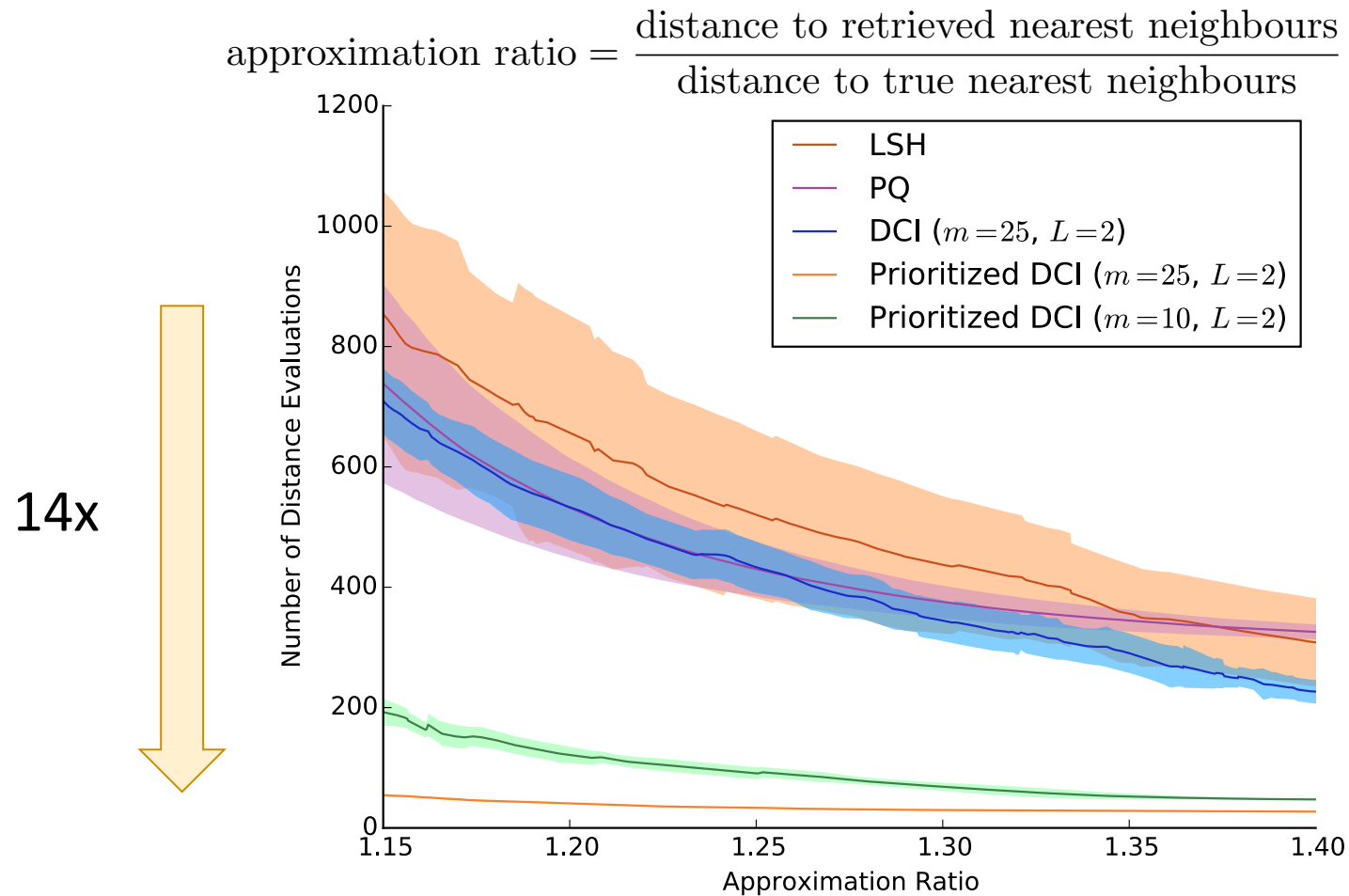
- Space: $O(mn)$
- Insertion Time: $O(m(d + \log n))$
- Deletion Time: $O(m \log n)$

Can insert and delete points without reconstructing the data structure, unlike most prior methods.

Experiments

Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

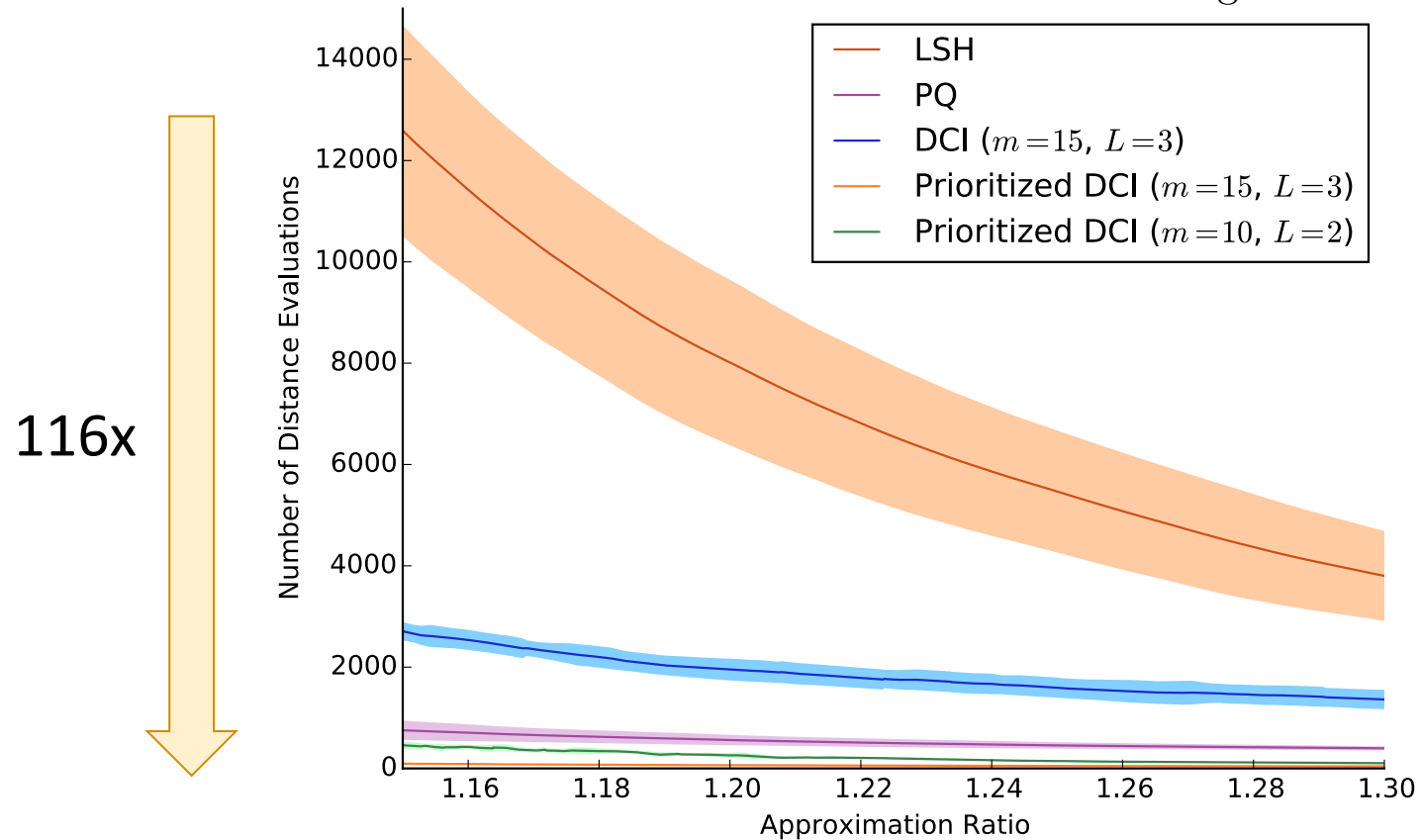
Query Time on CIFAR-100



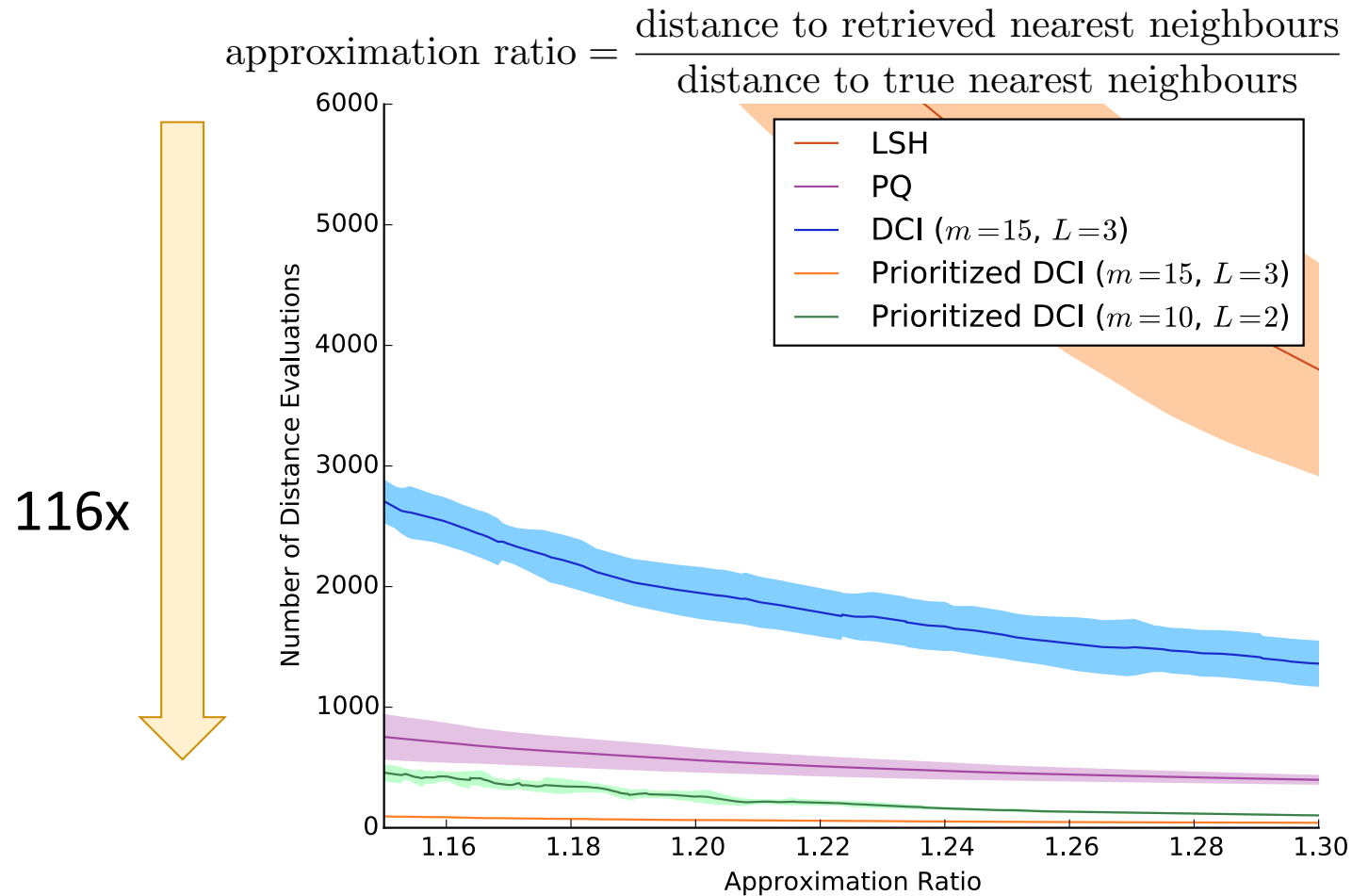
K. Li and J. Malik. Fast k -nearest neighbour search via Prioritized DCI. *ICML*, 2017
Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Query Time on MNIST

$$\text{approximation ratio} = \frac{\text{distance to retrieved nearest neighbours}}{\text{distance to true nearest neighbours}}$$

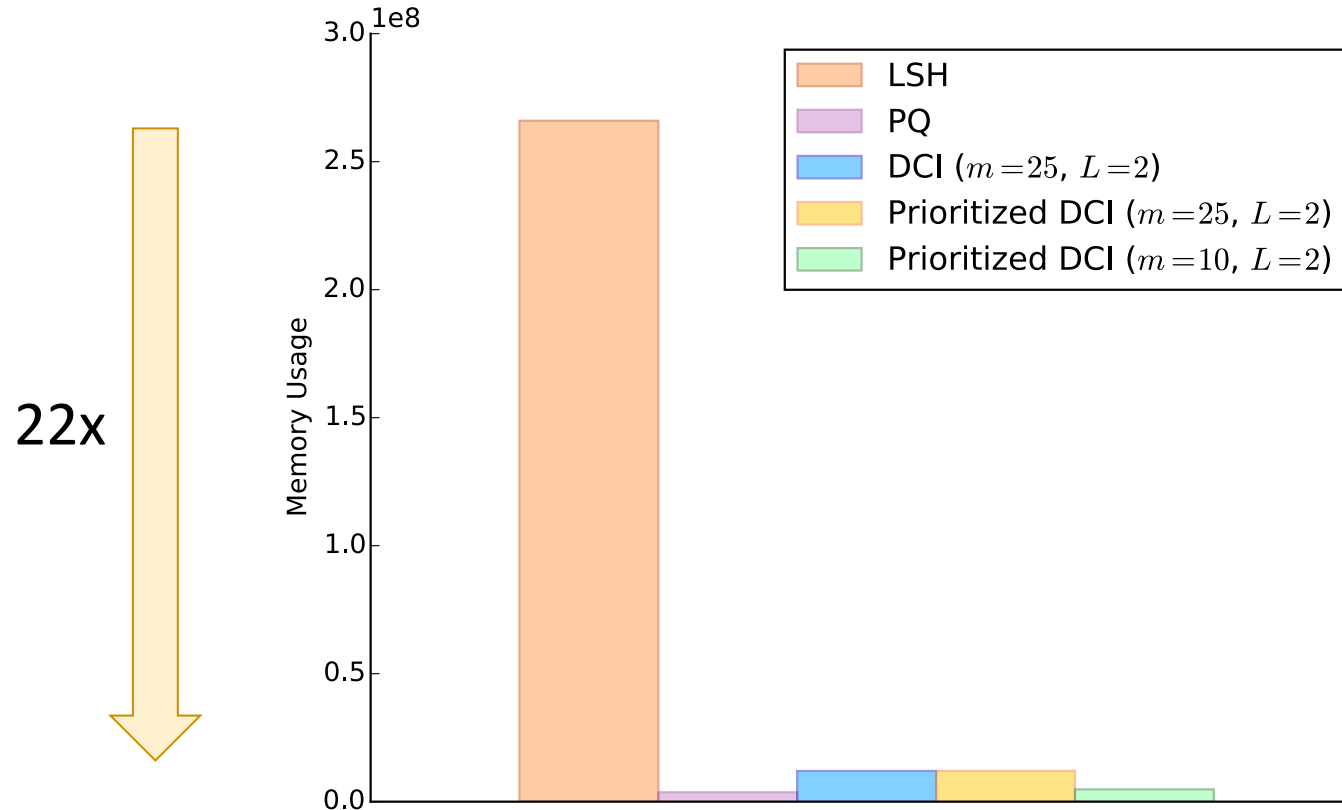


Query Time on MNIST

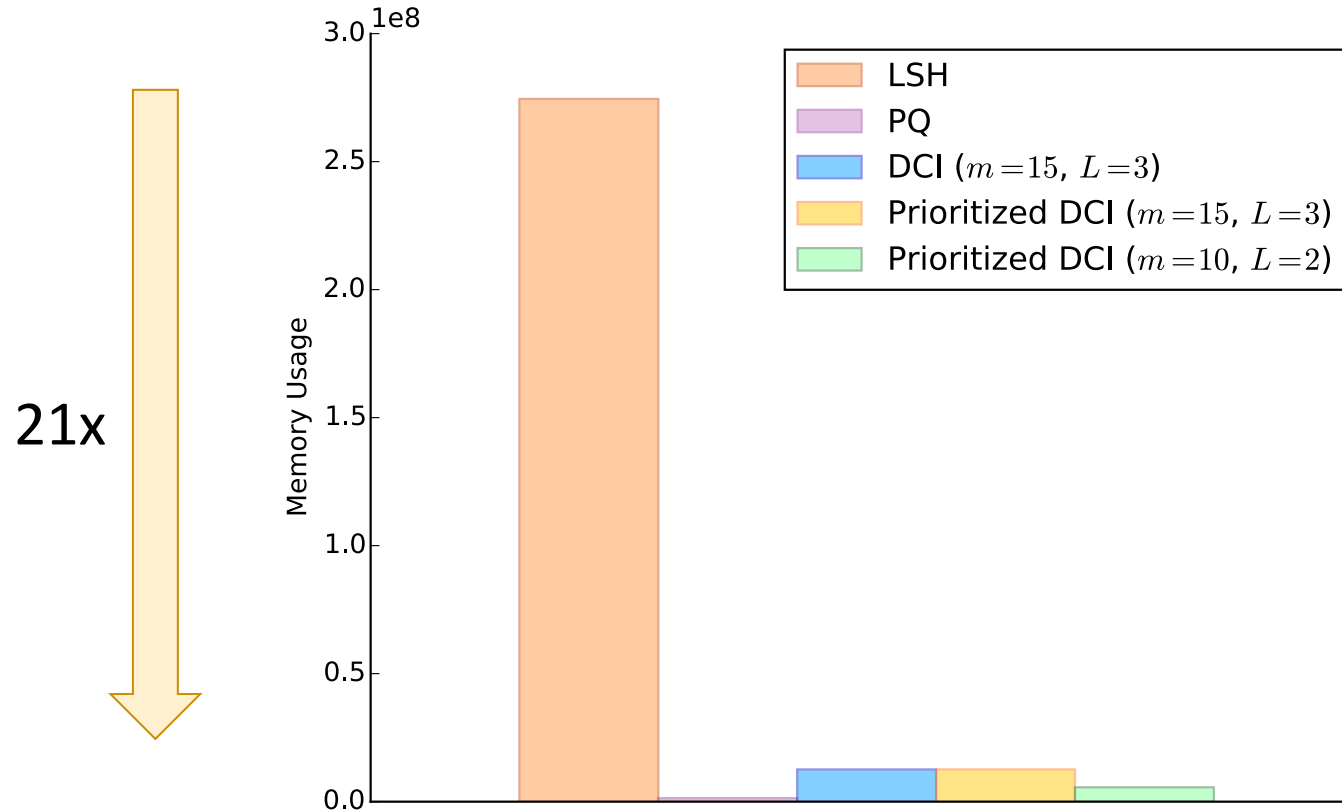


K. Li and J. Malik. Fast k -nearest neighbour search via Prioritized DCI. *ICML*, 2017
Overcoming the Curse of Dimensionality
in Nearest Neighbour Search

Space Efficiency on CIFAR-100



Space Efficiency on MNIST



For More Details...

Implicit Maximum Likelihood Estimation

Ke Li, Jitendra Malik

arXiv:1809.09087, 2018

On the Implicit Assumptions of GANs

Ke Li, Jitendra Malik

NeurIPS Workshop on Critiquing and Correcting Trends in Machine Learning, 2018

Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors

Yedid Hoshen, Ke Li, Jitendra Malik

CVPR, 2019

Super-Resolution via Conditional Implicit Maximum Likelihood Estimation

Ke Li*, Shichong Peng*, Jitendra Malik

arXiv:1810.01406, 2018

Diverse Image Synthesis from Semantic Layouts via Conditional IMLE

Ke Li*, Tianhao Zhang*, Jitendra Malik

ICCV, 2019

Fast k -Nearest Neighbour Search via Dynamic Continuous Indexing

Ke Li, Jitendra Malik

ICML, 2016

Fast k -Nearest Neighbour Search via Prioritized DCI

Ke Li, Jitendra Malik

ICML, 2017

Questions?

E-mail: ke.li@eecs.berkeley.edu

Website: <http://www.eecs.berkeley.edu/~ke.li>

Twitter: @KL_Div