FORTE

Finding **O**utliers with **R**epresentation **T**ypicality **E**stimation

Debargha Ganguly, Warren Morningstar, Andrew Yu, Vipin Chaudhary

Paper

Code

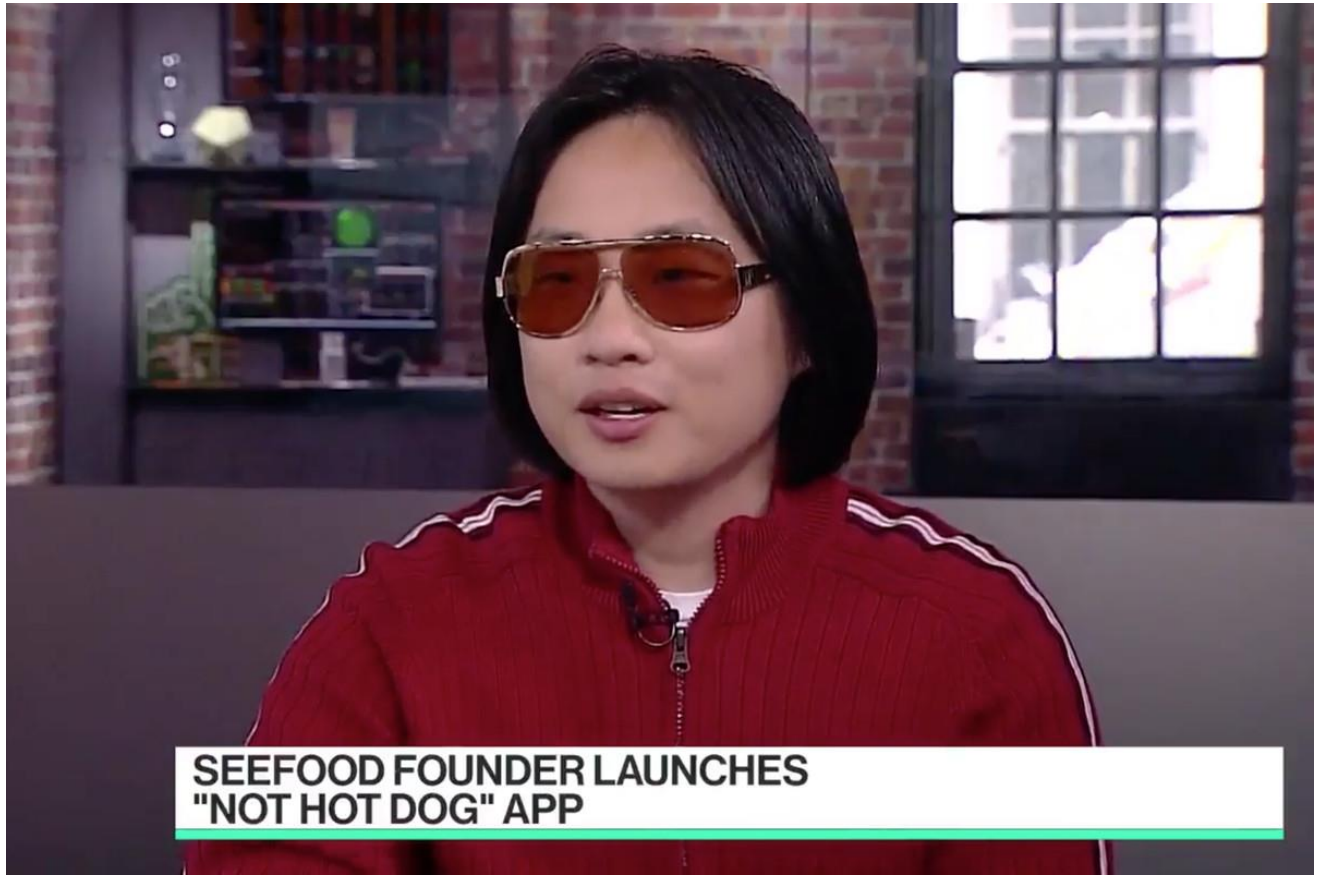
Packaged API

Agenda

- What is OOD? Why is it important?
- Theoretical Results
  - Gaussian Annulus Theorem
  - Typicality for OOD detection
  - Density of States
  - DoSE & Drawbacks
- Forte
- Results
- Domain Generalization
- How can you use Forte

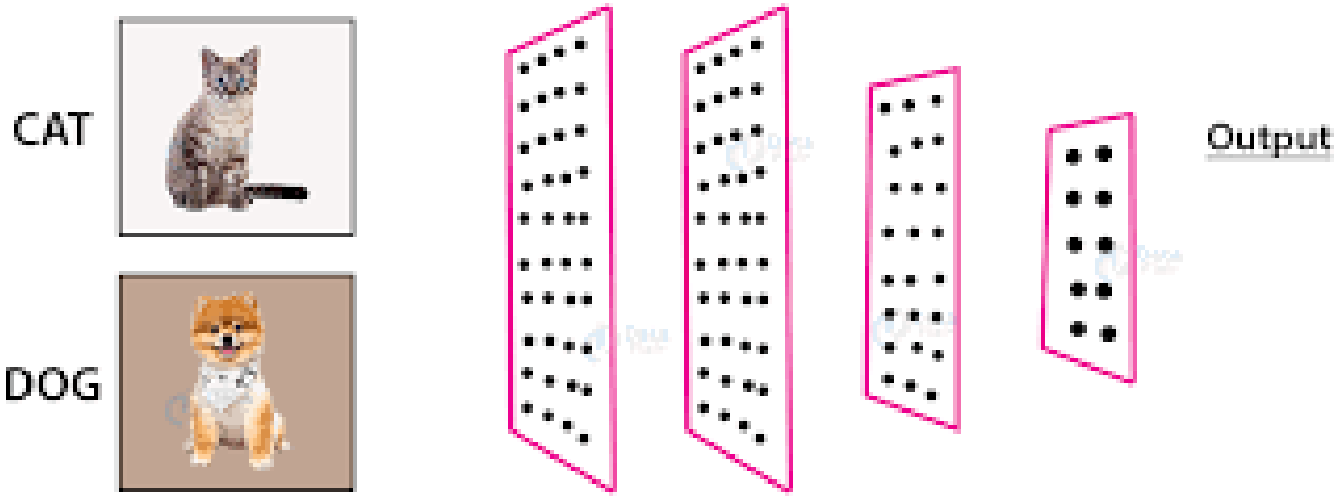
OOD : What & Why?

What is OOD?


SEEFOOD FOUNDER LAUNCHES
"NOT HOT DOG" APP


Hotdog!
Not hotdog!
Share
No Thanks
Share
No Thanks

What is OOD?



Near-OOD

Far-OOD

Formal Definition

A **distribution shift** occurs when the joint distribution changes from $\mathrm{P}_{\mathrm{train}}(\mathrm{X}, \mathrm{Y})$ during training to $\mathrm{P}_{\mathrm{test}}(\mathrm{X}, \mathrm{Y})$ during testing.

This can be decomposed into:

- **Covariate Shift**: Change in $\mathrm{P}(\mathrm{X})$ while $\mathrm{P}(\mathrm{Y}|\mathrm{X})$ remains constant.

$$\mathrm{P}_{\mathrm{train}}(\mathrm{X}) \neq \mathrm{P}_{\mathrm{test}}(\mathrm{X}), \quad \mathrm{P}_{\mathrm{train}}(\mathrm{Y}|\mathrm{X}) = \mathrm{P}_{\mathrm{test}}(\mathrm{Y}|\mathrm{X})$$

- **Label Shift**: Change in $\mathrm{P}(\mathrm{Y})$ while $\mathrm{P}(\mathrm{X}|\mathrm{Y})$ remains constant.

$$P_{\mathrm{train}}(Y) \neq P_{\mathrm{test}}(Y), \quad P_{\mathrm{train}}(X|Y) = P_{\mathrm{test}}(X|Y)$$

Formal Definition

A **distribution shift** occurs when the joint distribution changes from $\mathrm{P}_{\mathrm{train}}(\mathrm{X}, \mathrm{Y})$ during training to $\mathrm{P}_{\mathrm{test}}(\mathrm{X}, \mathrm{Y})$ during testing.

This can be decomposed into:

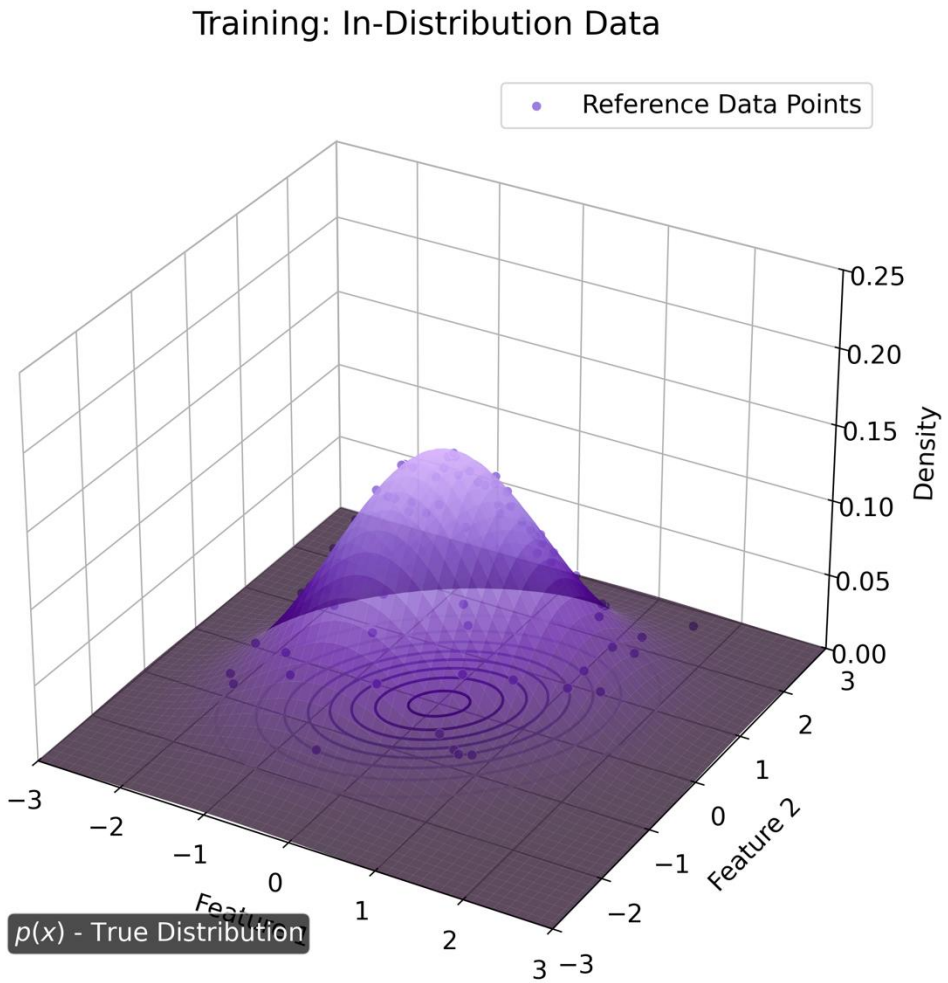
- **Concept Shift**: Change in $\mathrm{P}(\mathrm{Y}|\mathrm{X})$ while $\mathrm{P}(\mathrm{X})$ remains constant

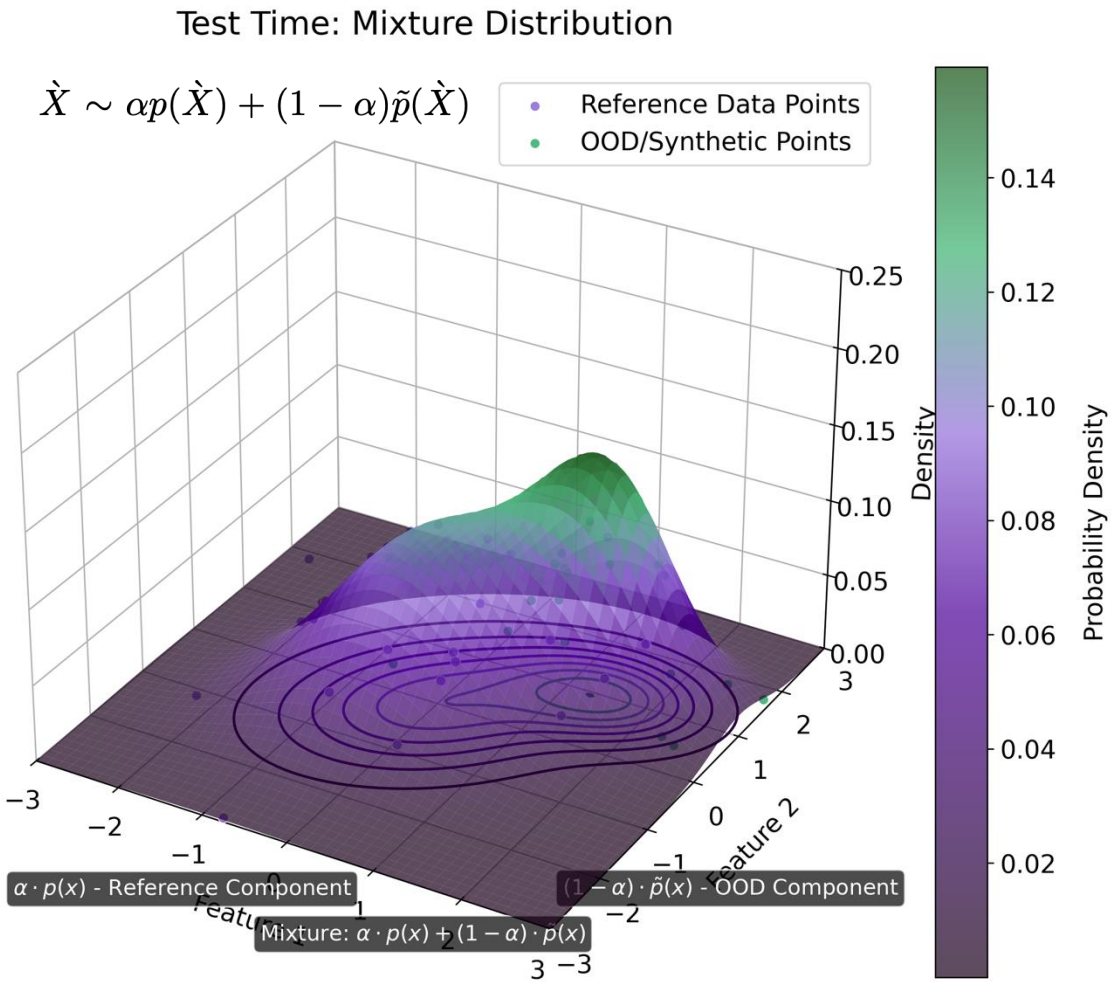
$$P_{\mathrm{train}}(X) = P_{\mathrm{test}}(X), \quad P_{\mathrm{train}}(Y|X) \neq P_{\mathrm{test}}(Y|X)$$

- **General Distribution Shift**: Change in both $\mathrm{P}(\mathrm{X})$ and $\mathrm{P}(\mathrm{Y}|\mathrm{X})$

$$P_{\mathrm{train}}(X) \neq P_{\mathrm{test}}(X), \quad P_{\mathrm{train}}(Y|X) \neq P_{\mathrm{test}}(Y|X)$$

Generalized Test-Time Distribution





Forte Framework analyzes the atypicality of test data $\dot{X}$ drawn from $\dot{X} \sim \alpha p(\dot{X}) + (1-\alpha)\tilde{p}(\dot{X})$ where α is an unknown mixing parameter and $\tilde{p}$ is an unknown OOD distribution.

Why OOD?

OOD is an important and *non-negotiable* problem for safe and deployable ML:

1. Provides the first line of defense, by preventing silent failures in critical ML Systems
2. Bounds AI capabilities by recognition of model knowledge
3. Allows safe fallback, and enables human oversight when needed

What is wrong with current methods? (OOD)

- Some **require class labels**.
  - MSP (Maximum Softmax Probability), OpenMax, Mahalanobis Distance-based, RMDS (Relative Mahalanobis Distance Scoring)
- Some **require exposure to OOD data** during training.
  - All methods in this category - Outlier Exposure, OE-ENERGY, MOS (Modeling the Open Space), etc.
- Some **present restrictions on architecture of models**
  - Density of States Estimation, DeepSVDD, DAGMM (autoencoder) etc.
- Some have **high deployment overhead.**
  - Posthoc methods like ODIN (temp scaling+preprocessing), VIM, ASH, GradNorm
- Most **do not present strong domain generalization**.
  - MSP, ODIN, Mahalanobis-based methods.

No generalized, data-centric approach yet!

Theoretical Foundations

And Previous SoTA

Gaussian Annulus Theorem

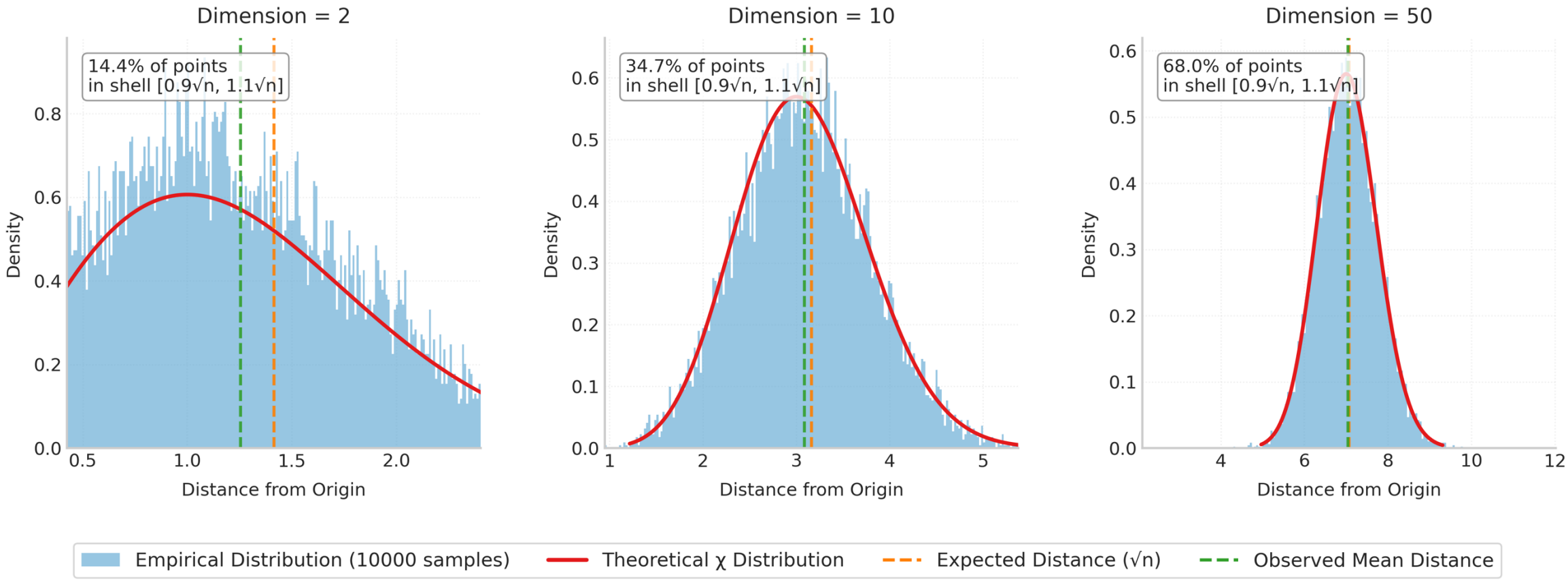
[Theorem] Let $\mathrm{x} \in \mathrm{R}^{\mathrm{n}}$ be a random vector sampled from the standard Gaussian distribution $\mathcal{N}(0, \mathrm{I_n})$ i.e., $\mathrm{x} \sim \mathcal{N}(0, \mathrm{I_n})$.

Then, for any small $\epsilon > 0$, the Euclidean norm of x concentrates around $\sqrt{\mathrm{n}}$, specifically:

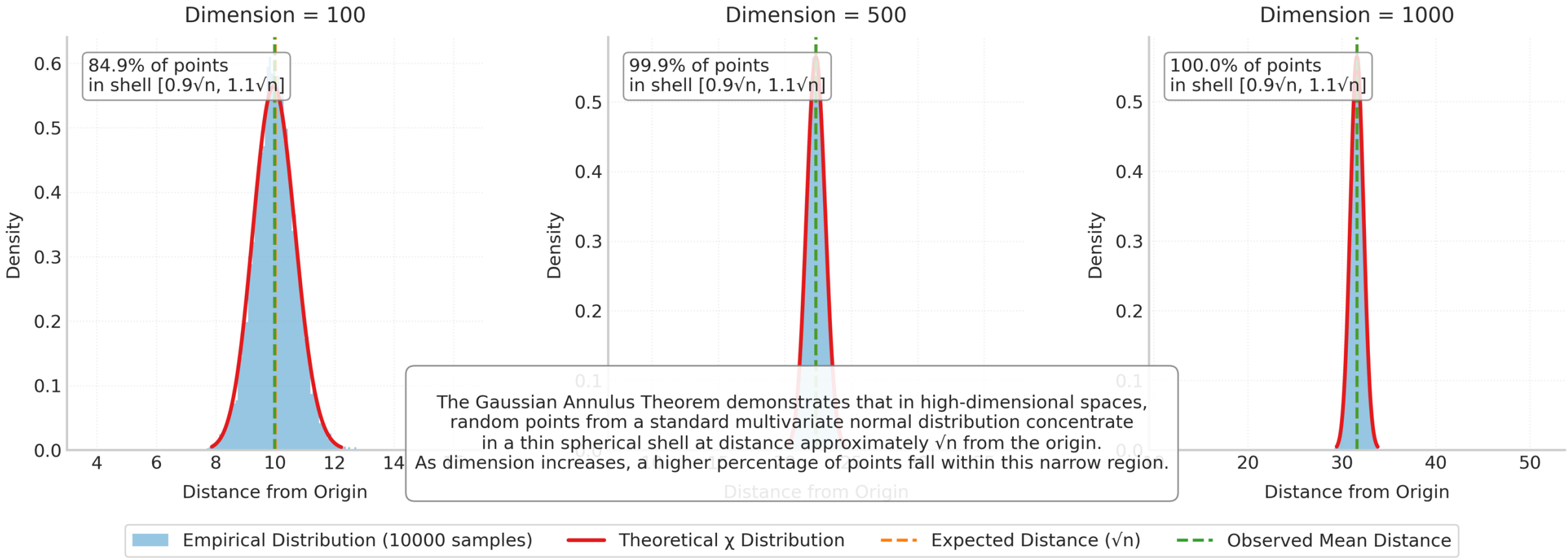
$$\sqrt{\mathrm{n}} - \mathrm{C}\epsilon\sqrt{\mathrm{n}} \leq ||\mathrm{x}|| \leq \sqrt{\mathrm{n}} + \mathrm{C}\epsilon\sqrt{\mathrm{n}}$$

with high probability, where C is a universal constant (typically $\mathrm{C} = \mathrm{O}(1)$).

Gaussian Annulus Theorem



Gaussian Annulus Theorem



Typicality

[Typical Set] Let $\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_\mathrm{n}$ be a sequence of i.i.d. random variables drawn from a discrete distribution with alphabet $\mathcal{X}$ and probability mass function $\mathrm{p}(\mathrm{x})$.

For any small $\epsilon > 0$, the *typical set* (also called the (ϵ-typical set) is defined as:

$$\mathrm{A}_\epsilon^{(\mathrm{n})} = \left\{ \mathrm{x}^\mathrm{n} \in \mathcal{X}^n : \left| -\frac{1}{\mathrm{n}} \log \mathrm{p}\left(\mathrm{x}^\mathrm{n}\right) - \mathrm{H}(\mathrm{X}) \right| \leq \epsilon \right\}$$

where $H(X)$ is the Shannon entropy of the source.

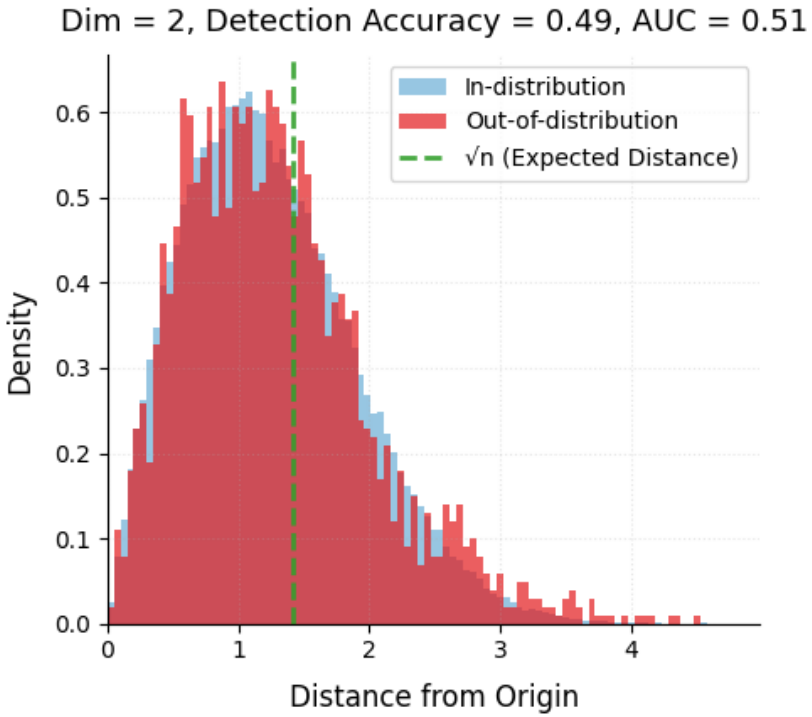
Typicality

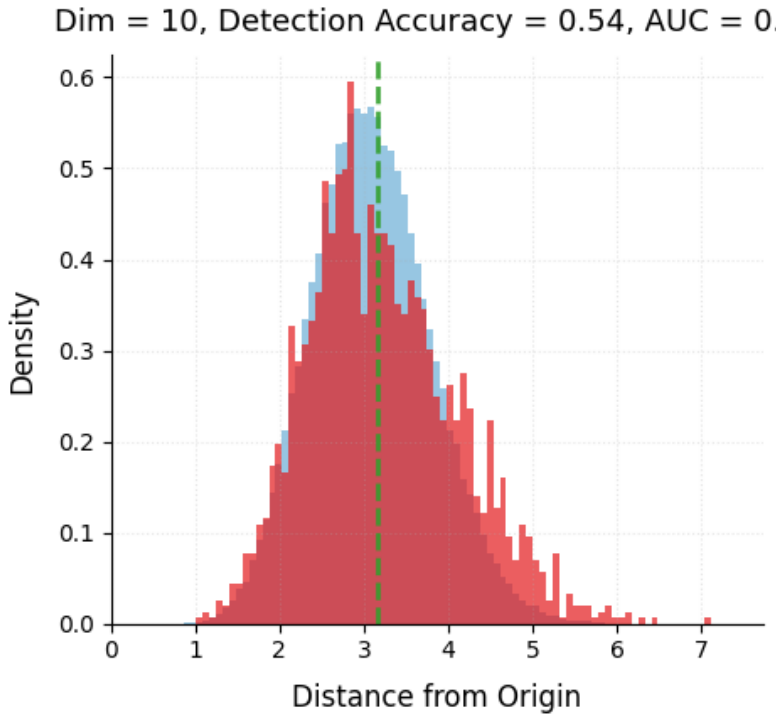
[Typicality Theorem] The typicality theorem (or Asymptotic Equipartition Property, AEP) states that:

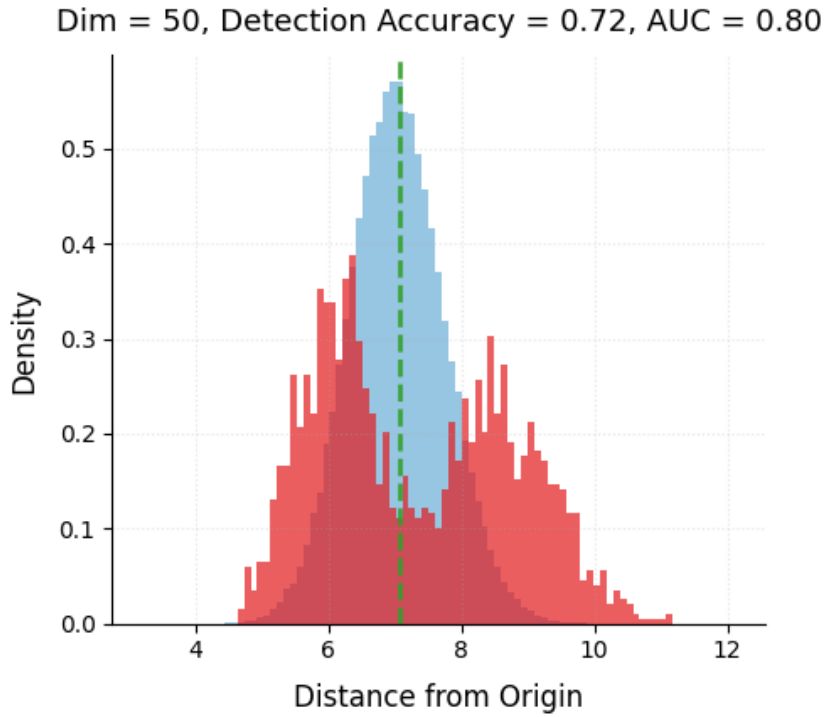
$$\Pr\left(x^n \in A_\epsilon^{(n)}\right) \to 1 \text{ as } n \to \infty$$

That is, for sufficiently large n, almost all sequences fall into the typical set, meaning their empirical entropy closely matches the true entropy of the source.

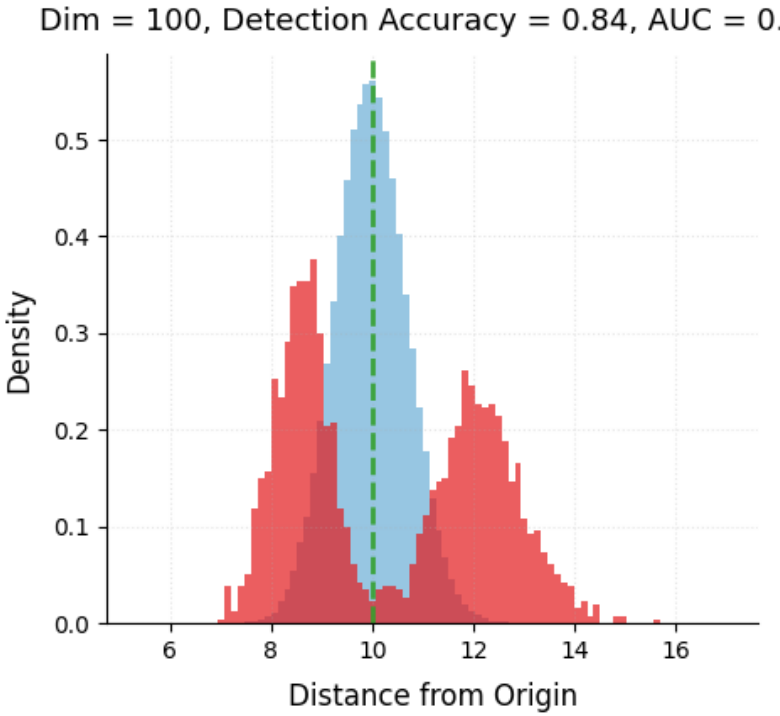
Typicality

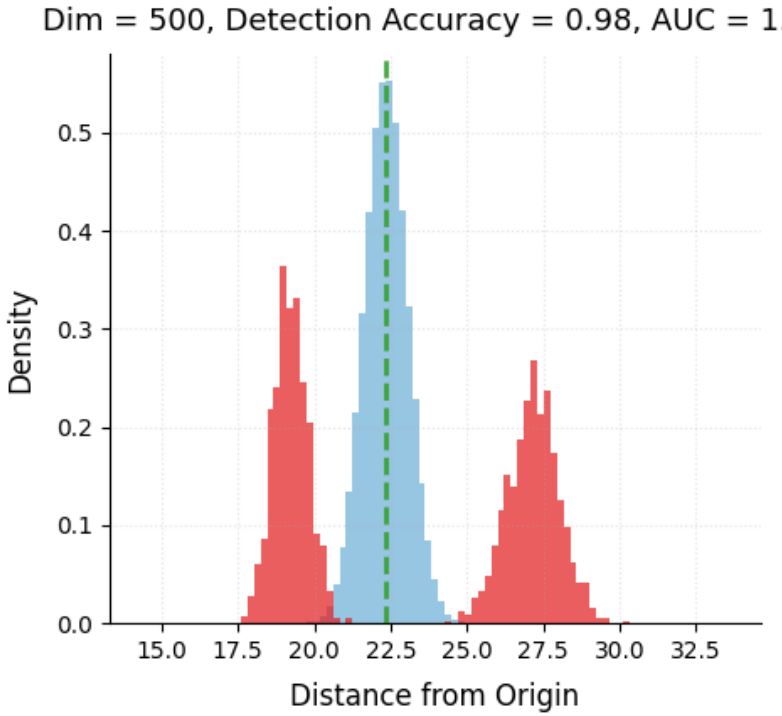

Dim = 2, Detection Accuracy = 0.49, AUC = 0.51
In-distribution
Out-of-distribution
√n (Expected Distance)
Density
Distance from Origin

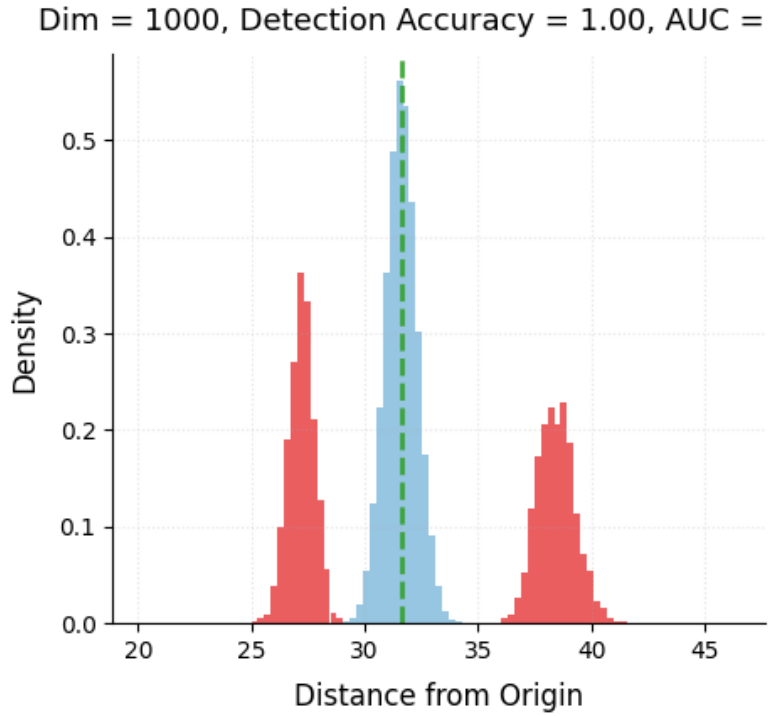

Dim = 10, Detection Accuracy = 0.54, AUC = 0.57
Density
Distance from Origin

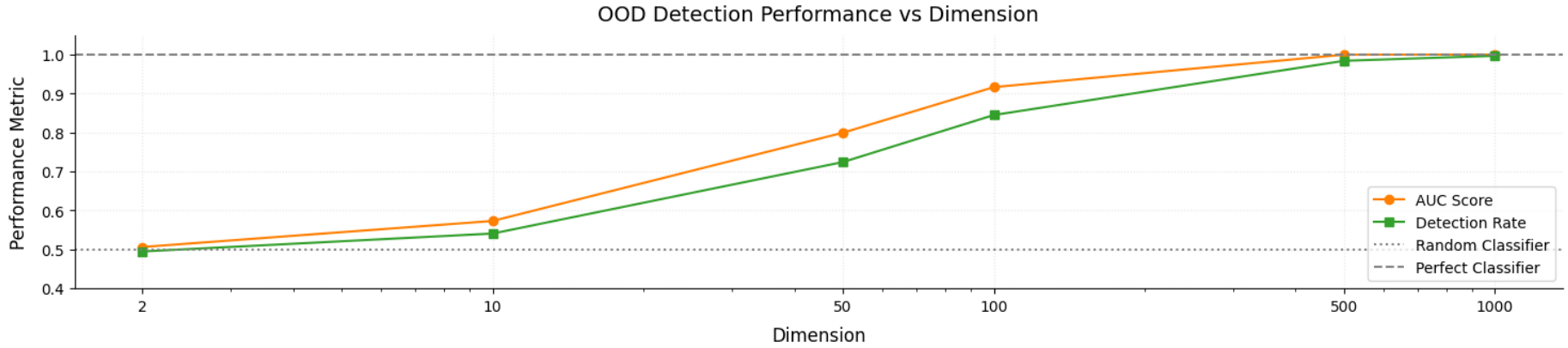

Dim = 50, Detection Accuracy = 0.72, AUC = 0.80
Density
Distance from Origin

Typicality


Dim = 100, Detection Accuracy = 0.84, AUC = 0.92
Density
Distance from Origin


Dim = 500, Detection Accuracy = 0.98, AUC = 1.00
Density
Distance from Origin


Dim = 1000, Detection Accuracy = 1.00, AUC = 1.00
Density
Distance from Origin


OOD Detection Performance vs Dimension
Performance Metric
Dimension
AUC Score
Detection Rate
Random Classifier
Perfect Classifier

Density of States

Setup

- You have a physical system (or dataset) consisting of particles (or data points) $x \in R^n$.
- For each particle x, you compute a scalar property $T_n(x)$ (this could be energy, log probability, max coordinate, etc.).
- The goal is to determine whether a particle is **atypical**, meaning its property $\mathrm{T_n(x)}$ is inconsistent with the equilibrium distribution governing the system.

Density of States

The **density of states** (DoS) for a property T is:

$$g(T) = \int \delta(T_n(x) - T)\, , d\mu(x)$$

Where $d\mu(x)$ is the natural measure on the space of particles (e.g., the product measure for i.i.d. particles, or the Boltzmann measure in physics).

This counts the "number of ways" the system can realize the property T .

In thermodynamics, $g(T)$ essentially gives the entropy contribution associated with a particular macroscopic observable T .

Concentration of Measures

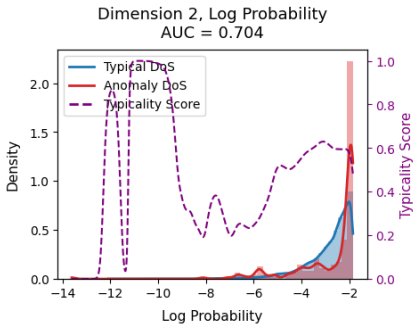
As the system dimension $n \to \infty$, the **distribution of the property T_n** (under the equilibrium measure) concentrates sharply around a typical value $T^\star$, meaning:

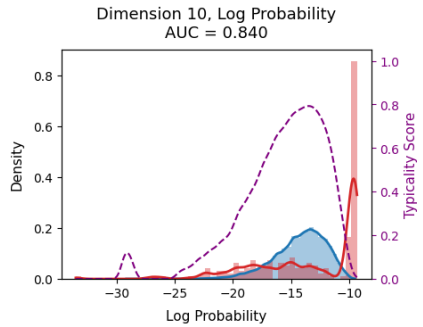
$$\Pr(|T_n - T^\star| \leq \epsilon) \to 1 \text{ as } n \to \infty.$$

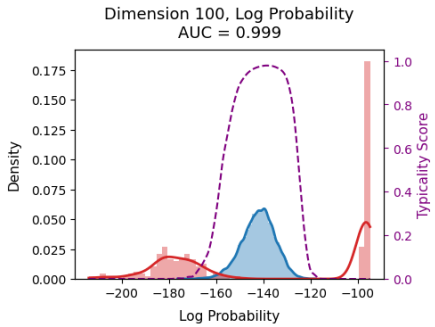
This is a direct consequence of the concentration of measure — almost all particles in the system exhibit the same empirical macroscopic property $T^\star$.

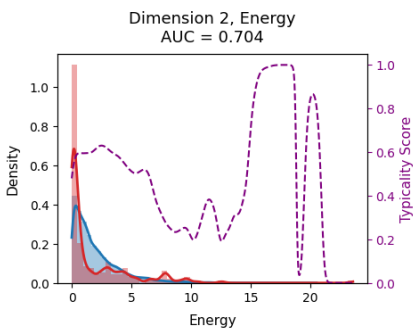
[By Large Deviation Principle] We can say that there is an exponential decay of observing atypical values of T_n that deviate significantly from $T^\star$]

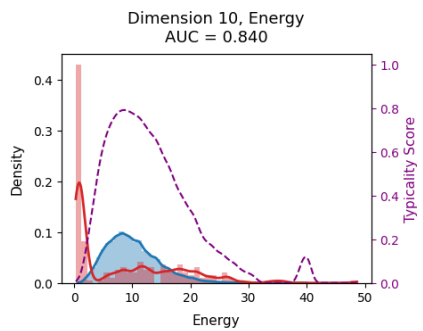
Density of States

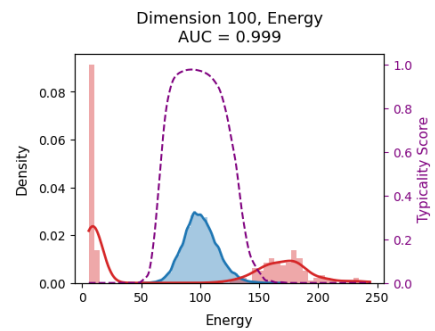


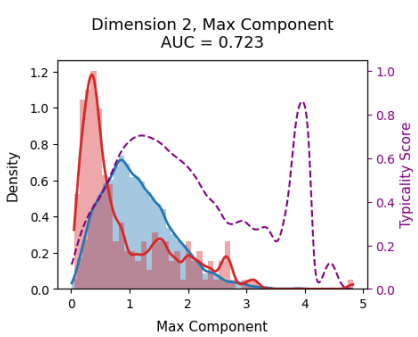


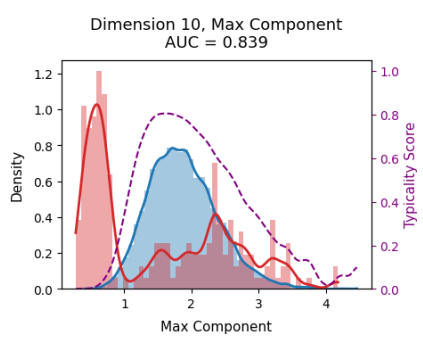


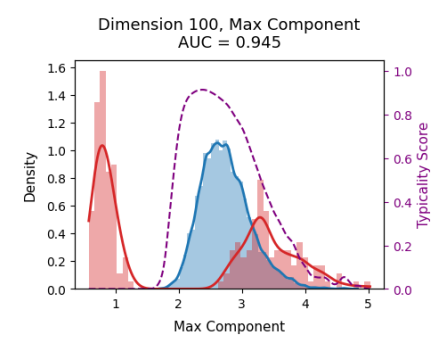




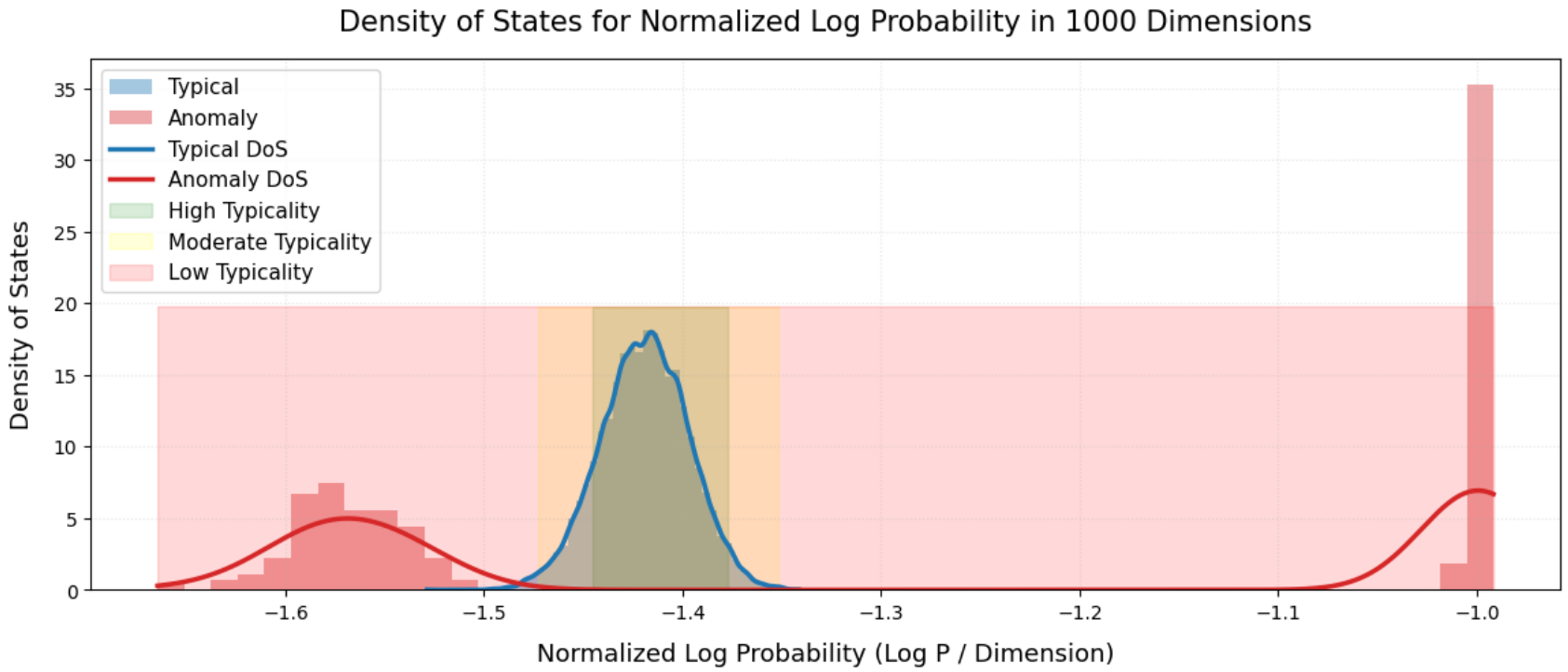


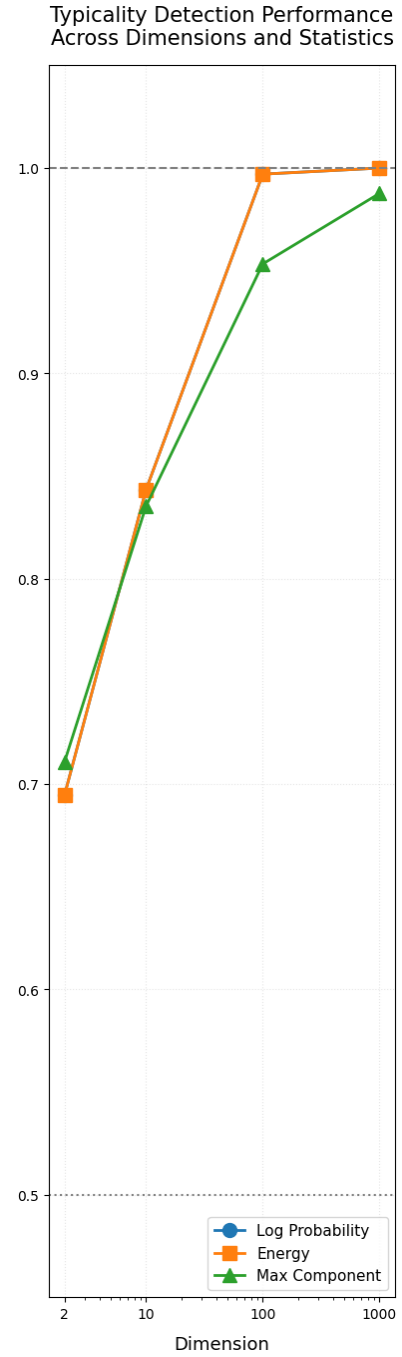






Density of States





Key Takeaway (in plain English):

- **Typicality**: Most sequences look "typical" — their empirical entropy matches the true entropy.
- **Gaussian Annulus**: Most Gaussian vectors live on a thin shell at radius $\sqrt{n}$.
- **Rare Events**: The probability of finding a system in an atypical state decays exponentially with the system size.

Previous work in Typicality Est. in OOD

- WAIC (Watanabe-Akaike Information Criterion)
- Likelihood Ratios
- Typicality Tests
- DoSE

Density of States Estimation

Uses Generative Models & Establishes Prior SoTA in OOD

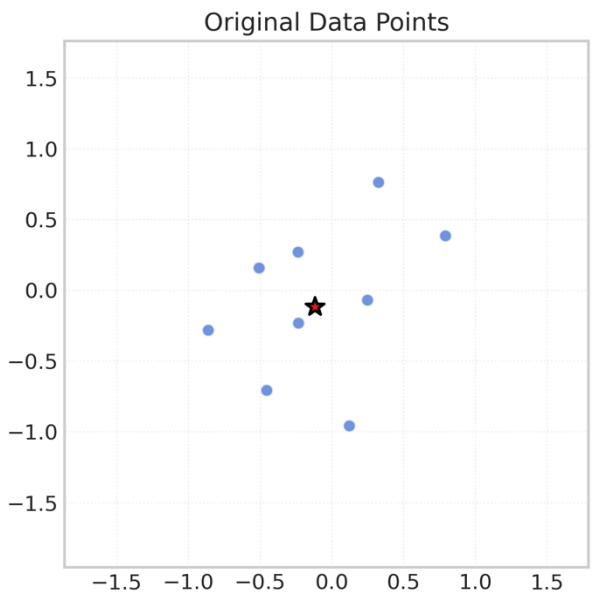
- VAEs
  - Posterior-Prior Cross Entropy — $H[q(Z|X), \theta_n), q(Z)]$
  - Posterior Entropy — $H[q(Z|X, \theta_n)]$
  - Posterior-Prior KL Divergence — $KL[q(Z|X, \theta_n), q(Z)]$
  - Posterior Expected log-likelihood — $E_{q(Z|X,\theta n)}[\log q(X/Z, \theta_n)]$
  - IWAE — $\log E_{q(Z|X, \theta n)}[q(X,Z,\theta_n)/q(Z|X, \theta_n)$
- GLOW
  - Log Likelihood — $q(X|\ \theta_n)$
  - Log Prob of Latent Variable — $q(Z|X, \theta_n)$
  - Log Determinant of Jacobian — $\log |J(X)|$

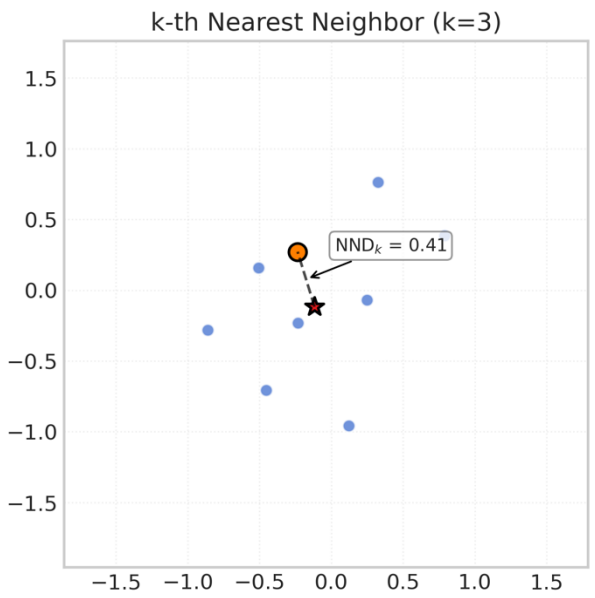
What to improve in DoSE?

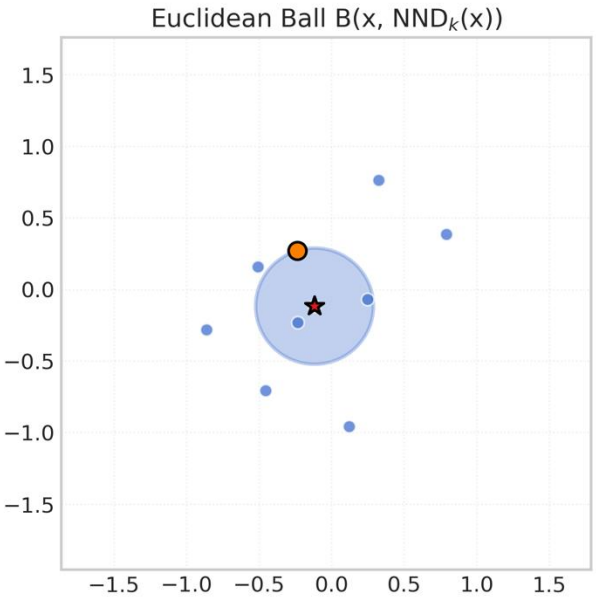
- DoSE requires training generative models
  - What happens when a few samples change? New model training?
  - GLOW has
    - Invertible architecture, exact log-likelihood eval
    - Inefficient, high-memory requirements.
  - VAEs
    - Few Samples? Can’t train VAE. (Sample inefficiency on complex datasets)
    - Poorly structured latent spaces, degraded performance.

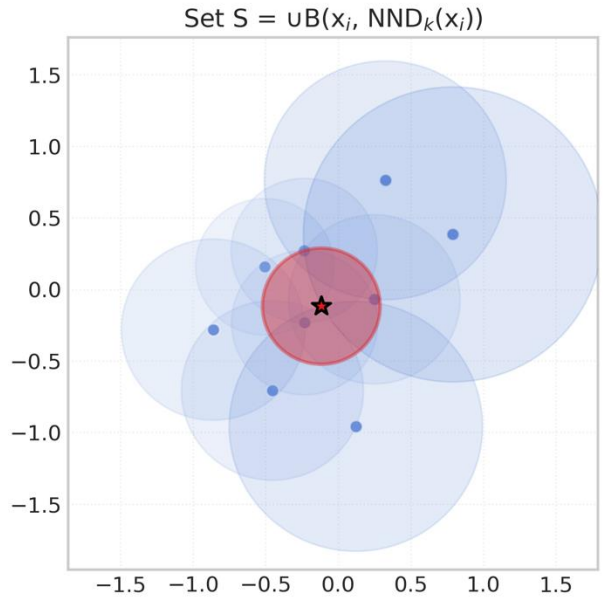
Forte : Methodology

Neighborhood Locality









$X = \{x_i^r\}_{i=1}^m$ (Train Data)

$\{x_j^g\}_{j=1}^n$ (Test Time Data)

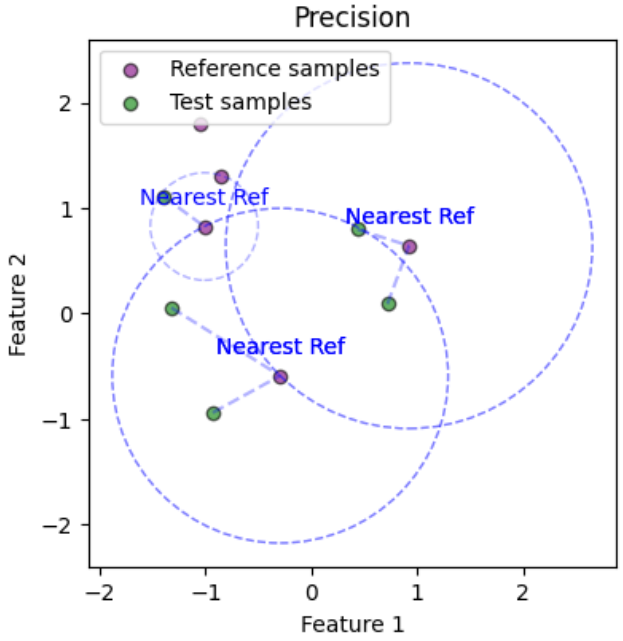
B(x,r) is a Euclidean ball centered at x with radius r, and $NND_k(x^r_i)$ is the distance between x^r_i and its k-th nearest neighbor in $\{x^r_i\}^m_{i=1}$

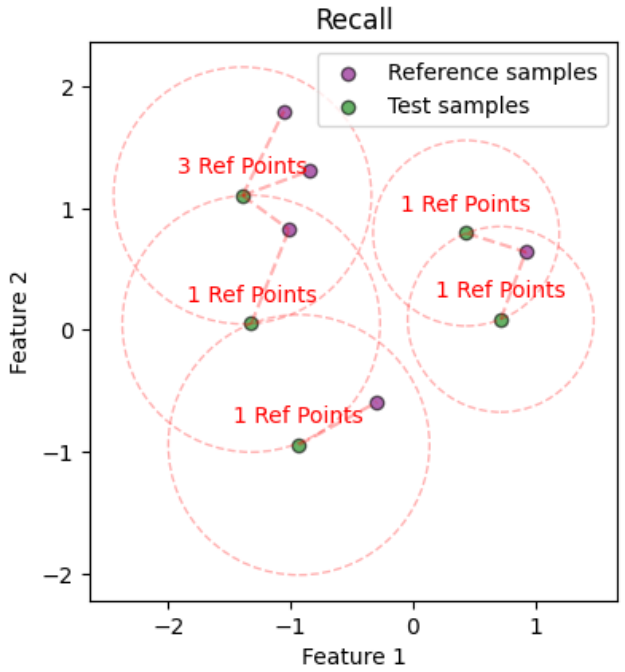
$$S(\{x_i^r\}_{i=1}^m) = \bigcup_{i=1}^m B(x_i^r, \mathrm{NND}_k(x_i^r))$$

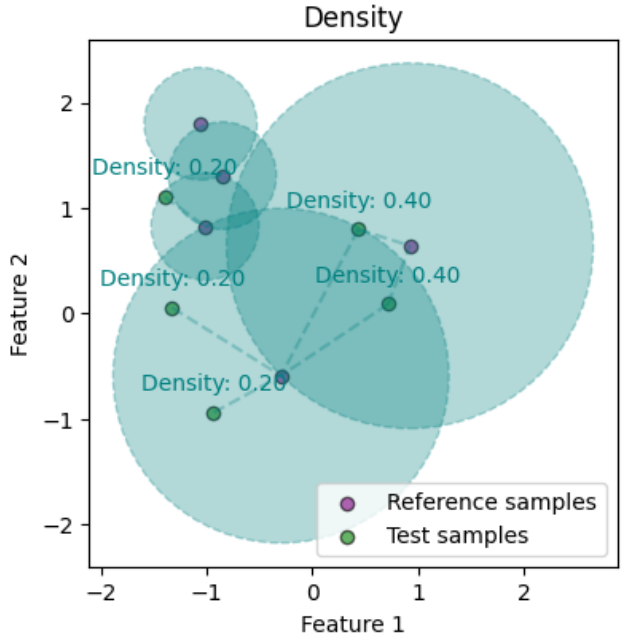
Use Self-Supervised Methods instead of Generative Methods

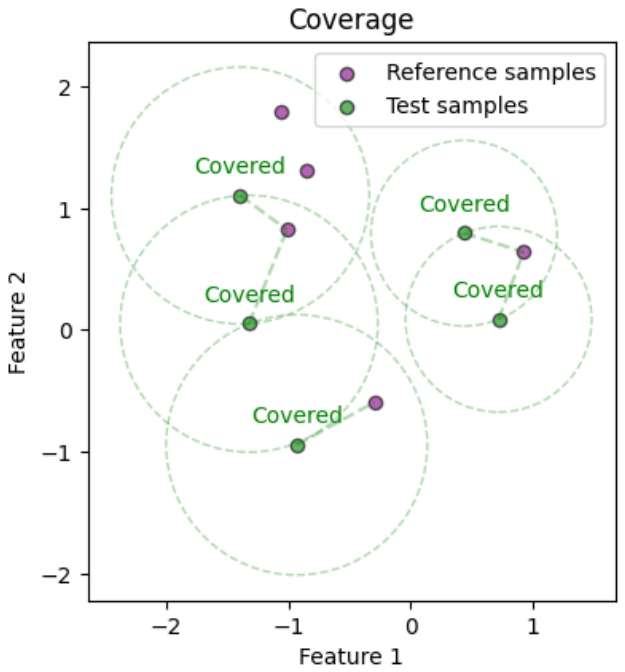
-

Our Measures









$X = \{x_i^r\}_{i=1}^m$ (Train Data)

$\{x_j^g\}_{j=1}^n$ (Test Time Data)

$\grave{X} \sim \alpha p(\grave{X}) + (1-\alpha)\tilde{p}(\grave{X})$ (Test Time Data Distribution)

$S(\{x_i^r\}_{i=1}^m) = \bigcup_{i=1}^m B(x_i^r, \mathrm{NND}_k(x_i^r))$

B(x,r) is a Euclidean ball centered at x with radius r, and $\mathrm{NND}_k(x^r_i)$ is the distance between x^r_i and its k-th nearest neighbor in $\{x^r_i\}^m_{i=1}$

$$\mathrm{precision}_{\mathrm{pp}}^{(\mathrm{j})} = \mathbb{1}\left(x_j^g \in S(\{x_i^r\}_{i=1}^m)\right)$$

$$\mathrm{recall}_{\mathrm{pp}}^{(\mathrm{j})} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left(x_i^r \in B(x_j^g, \mathrm{NND}_k(x_j^g))\right)$$

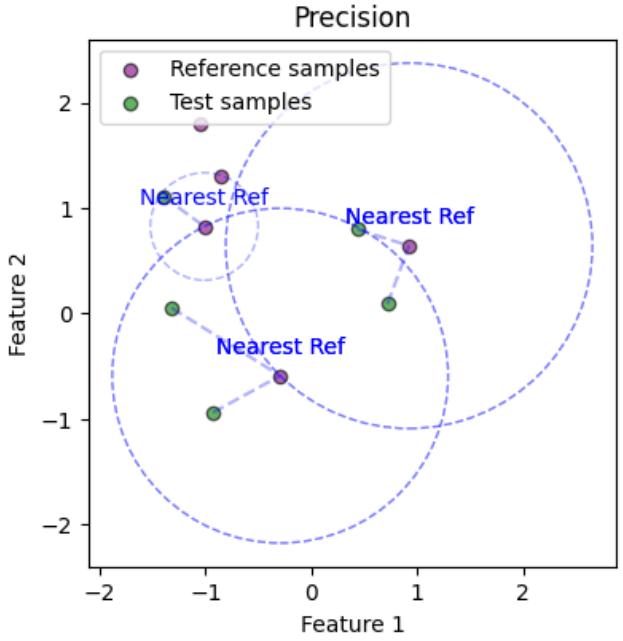
$$\mathrm{density}_{\mathrm{pp}}^{(\mathrm{j})} = \frac{1}{km}\sum_{i=1}^{m} \mathbb{1}\left(x_j^g \in B(x_i^r, \mathrm{NND}_k(x_i^r))\right)$$

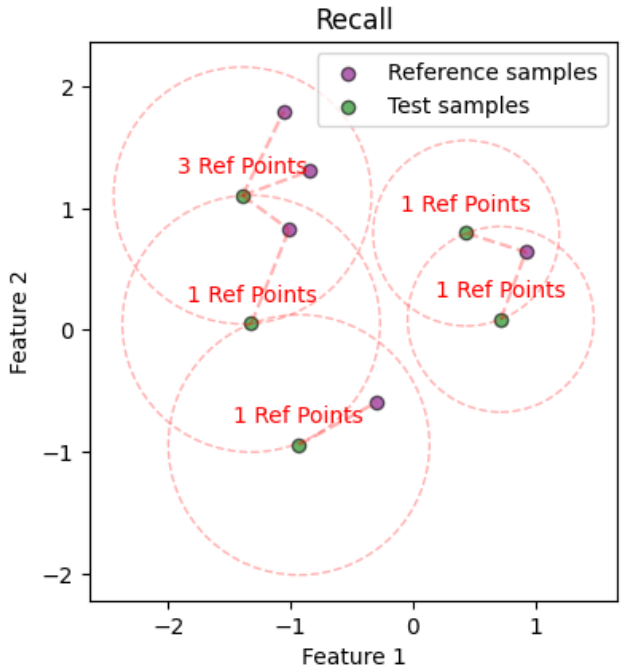
$$\mathrm{coverage}_{\mathrm{pp}}^{(\mathrm{j})} = \mathbb{1}\left(\min_i(d(x_j^g, x_i^r)) < \mathrm{NND}_k(x_j^g)\right)$$

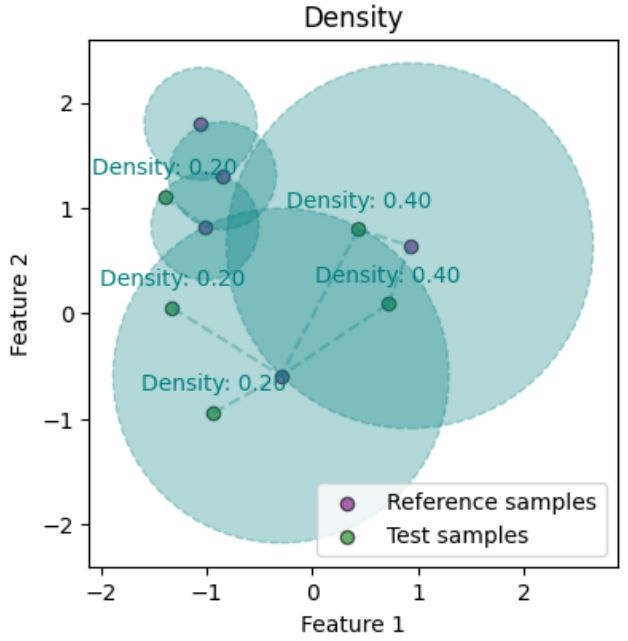
Fit KDE, OCSVM or GMM on distribution of stats on ID data.

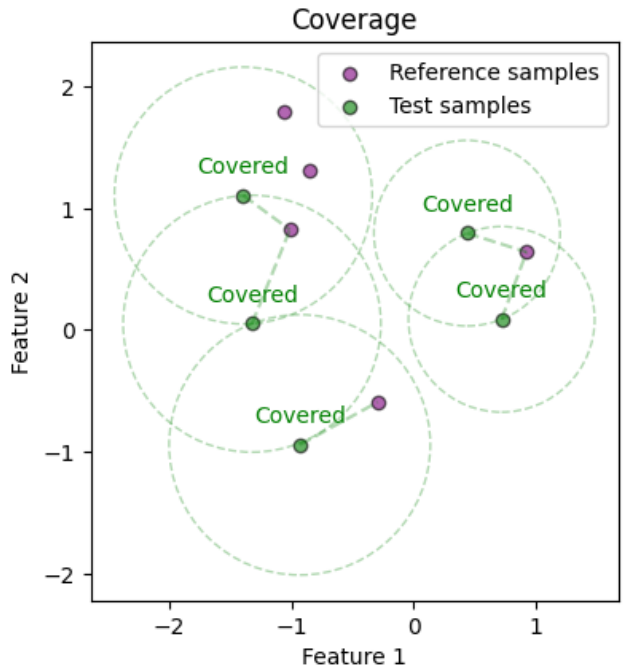
Test stats on mixture distribution to find OOD

Our Measures

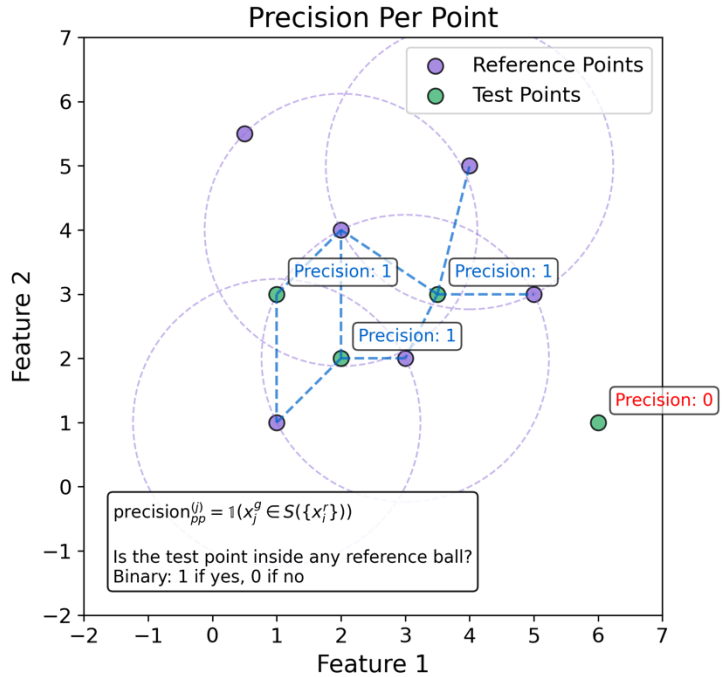


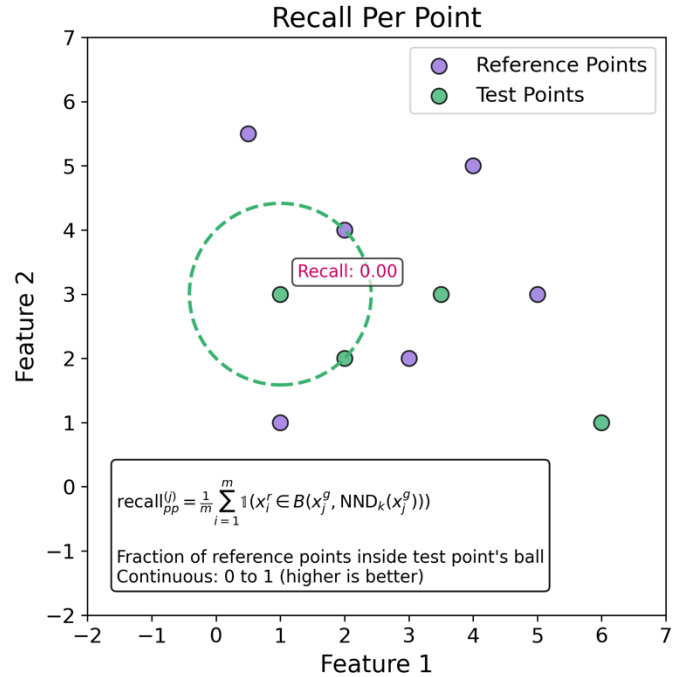




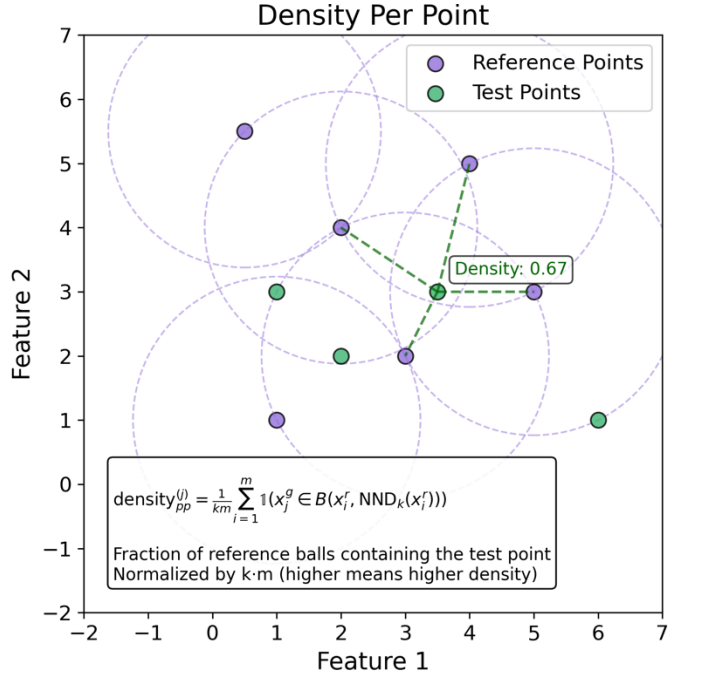


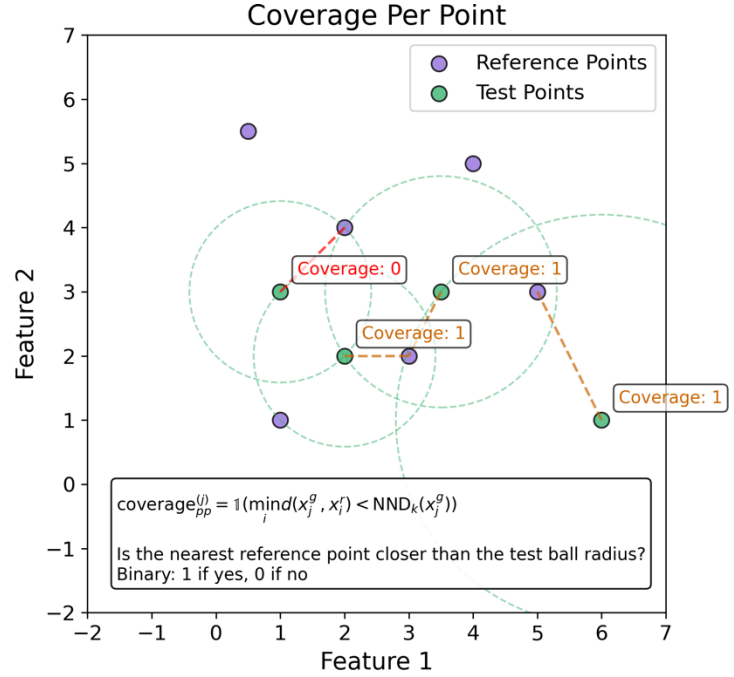
Is the test point inside any reference ball?	Fraction of reference points inside test point's ball	Fraction of reference balls containing the test point Normalized by k.m.	Is the nearest reference point closer than the test-ball radius?
• Binary: 1 if yes, 0 if no. (1 is less OOD)	• Continuous: 0 to 1 (higher is less OOD)	• Higher means higher density	• Binary: 1 if yes, 0 if no.

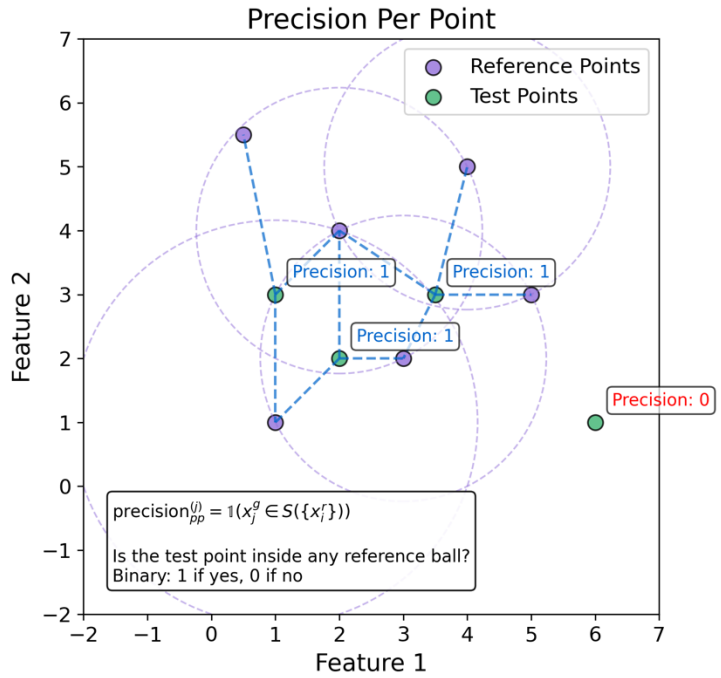

Precision Per Point
Reference Points
Test Points
Precision: 1
Precision: 1
Precision: 1
Precision: 0
$\text{precision}_{pp}^{(j)} = \mathbb{1}(x_j^g \in S(\{x_i^r\}))$
Is the test point inside any reference ball?
Binary: 1 if yes, 0 if no
Feature 1
Feature 2

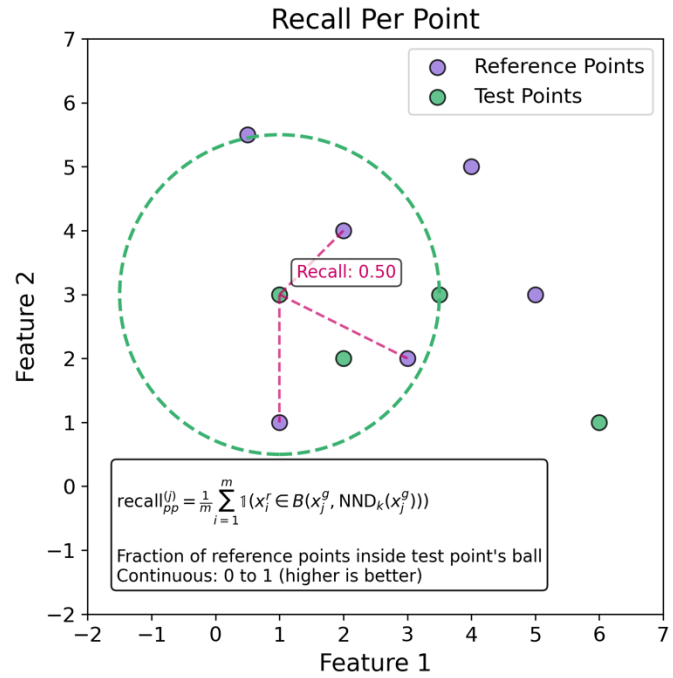

Recall Per Point
Reference Points
Test Points
Recall: 0.00
$\text{recall}_{pp}^{(j)} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(x_i^r \in B(x_j^g, \text{NND}_k(x_j^g)))$
Fraction of reference points inside test point's ball
Continuous: 0 to 1 (higher is better)
Feature 1
Feature 2

K=1

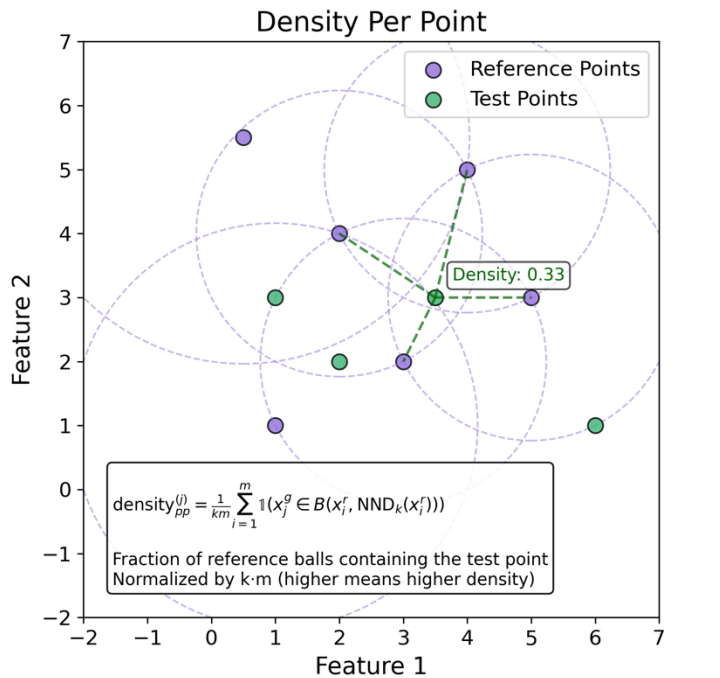

Density Per Point
Reference Points
Test Points
Density: 0.67
$\text{density}_{pp}^{(j)} = \frac{1}{km}\sum_{i=1}^{m} \mathbb{1}(x_j^g \in B(x_i^r, \text{NND}_k(x_i^r)))$
Fraction of reference balls containing the test point
Normalized by k·m (higher means higher density)
Feature 1
Feature 2

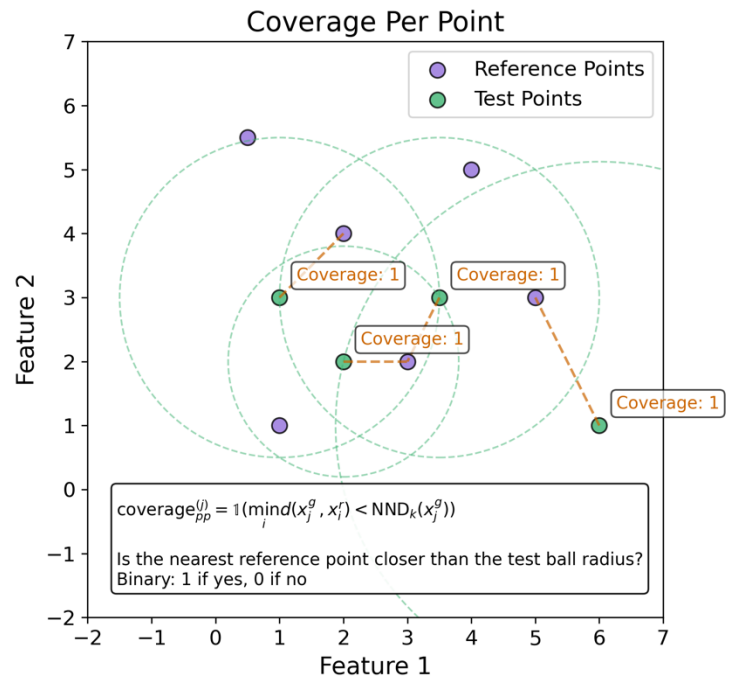

Coverage Per Point
Reference Points
Test Points
Coverage: 0
Coverage: 1
Coverage: 1
Coverage: 1
$\text{coverage}_{pp}^{(j)} = \mathbb{1}(\min_i d(x_j^g, x_i^r) < \text{NND}_k(x_j^g))$
Is the nearest reference point closer than the test ball radius?
Binary: 1 if yes, 0 if no
Feature 1
Feature 2

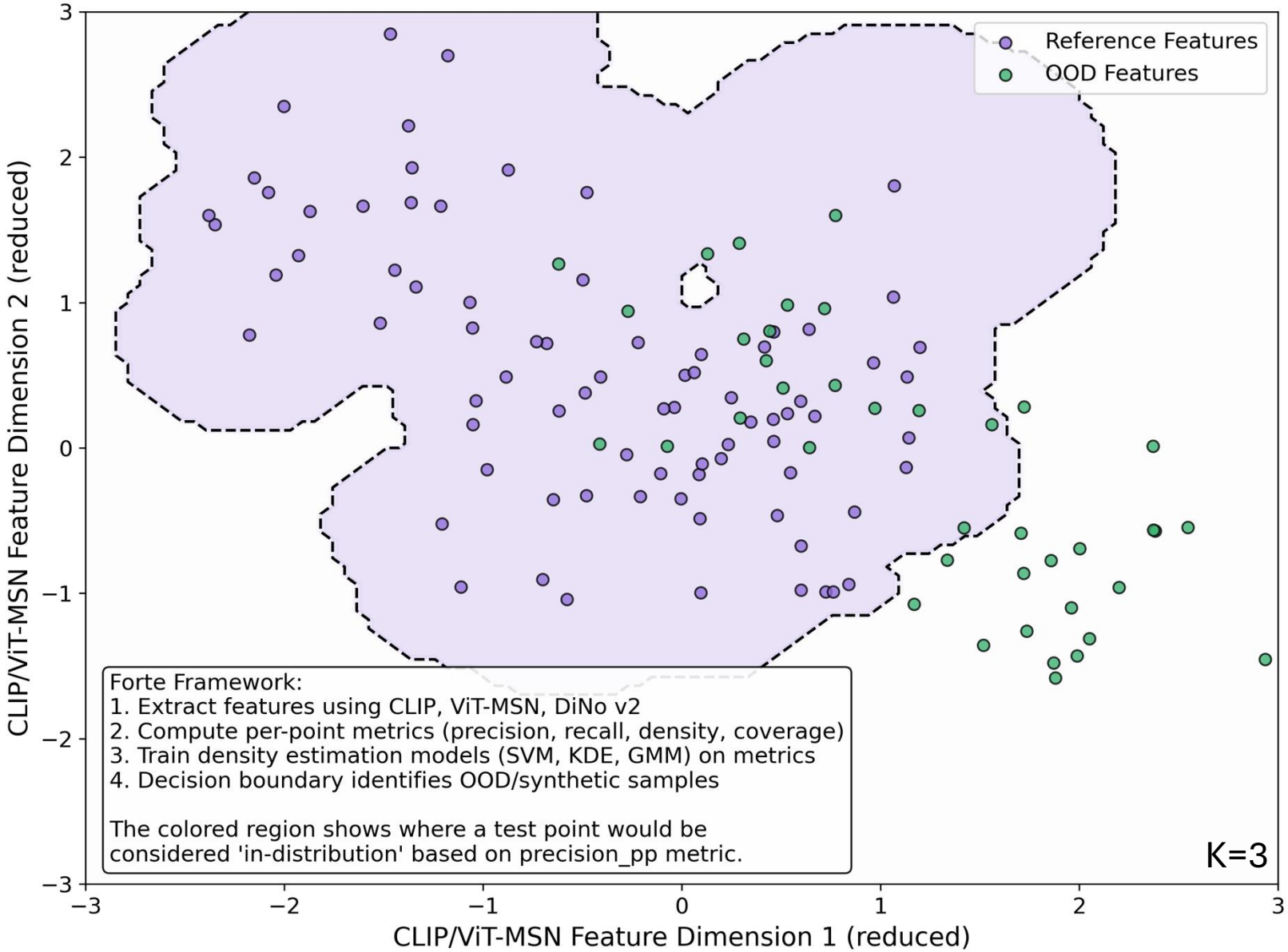

Precision Per Point
Reference Points
Test Points
Precision: 1
Precision: 1
Precision: 1
Precision: 0
$\text{precision}_{pp}^{(j)} = \mathbb{1}(x_j^g \in S(\{x_i^r\}))$
Is the test point inside any reference ball?
Binary: 1 if yes, 0 if no
Feature 1
Feature 2


Recall Per Point
Reference Points
Test Points
Recall: 0.50
$\text{recall}_{pp}^{(j)} = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}(x_i^r \in B(x_j^g, \text{NND}_k(x_j^g)))$
Fraction of reference points inside test point's ball
Continuous: 0 to 1 (higher is better)
Feature 1
Feature 2

K=2

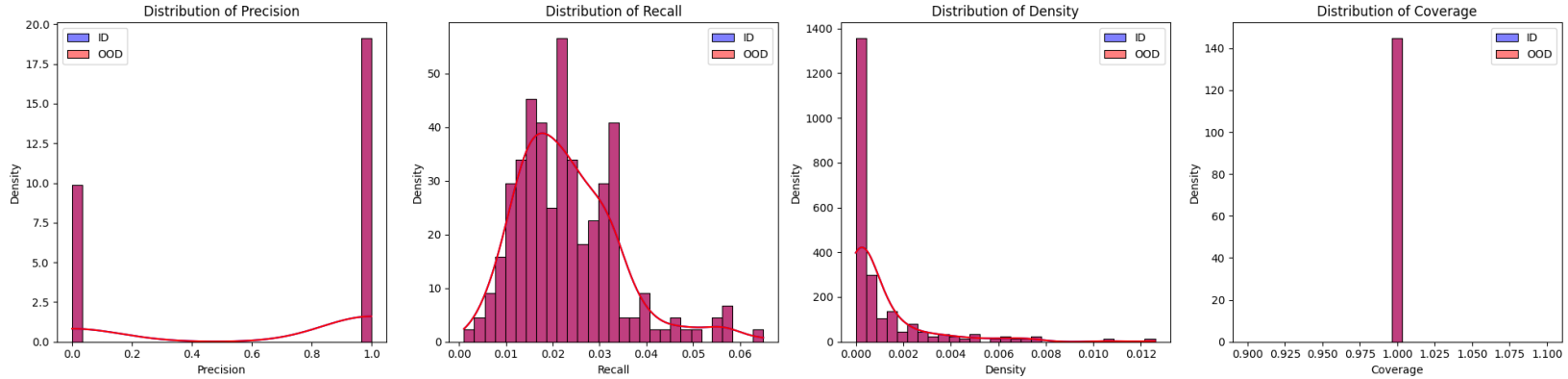

Density Per Point
Reference Points
Test Points
Density: 0.33
$\text{density}_{pp}^{(j)} = \frac{1}{km}\sum_{i=1}^{m} \mathbb{1}(x_j^g \in B(x_i^r, \text{NND}_k(x_i^r)))$
Fraction of reference balls containing the test point
Normalized by k·m (higher means higher density)
Feature 1
Feature 2


Coverage Per Point
Reference Points
Test Points
Coverage: 1
Coverage: 1
Coverage: 1
Coverage: 1
$\text{coverage}_{pp}^{(j)} = \mathbb{1}(\min_i d(x_j^g, x_i^r) < \text{NND}_k(x_j^g))$
Is the nearest reference point closer than the test ball radius?
Binary: 1 if yes, 0 if no
Feature 1
Feature 2

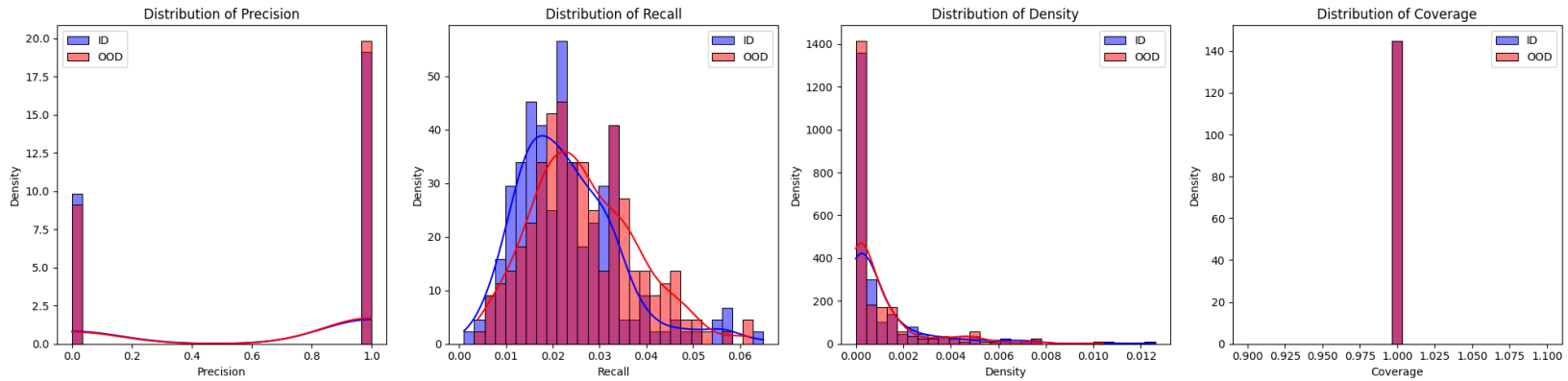

Reference Features
OOD Features
CLIP/ViT-MSN Feature Dimension 2 (reduced)
CLIP/ViT-MSN Feature Dimension 1 (reduced)
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
Forte Framework:
1. Extract features using CLIP, ViT-MSN, DiNo v2
2. Compute per-point metrics (precision, recall, density, coverage)
3. Train density estimation models (SVM, KDE, GMM) on metrics
4. Decision boundary identifies OOD/synthetic samples
The colored region shows where a test point would be considered 'in-distribution' based on precision_pp metric.
K=3

Under Distribution Shift

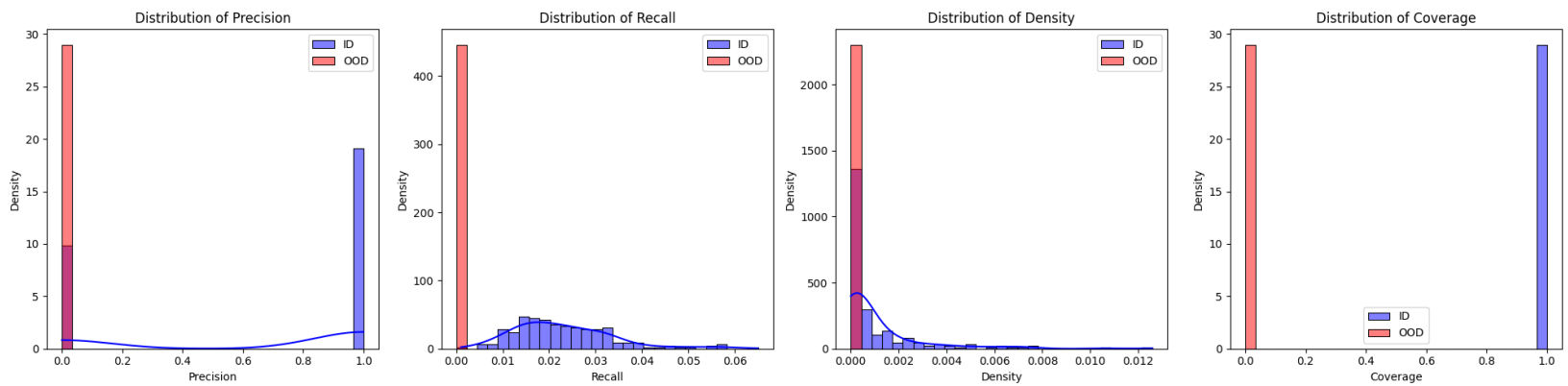
Under no shift



Under tiny shift



Under small shift



Benchmarks

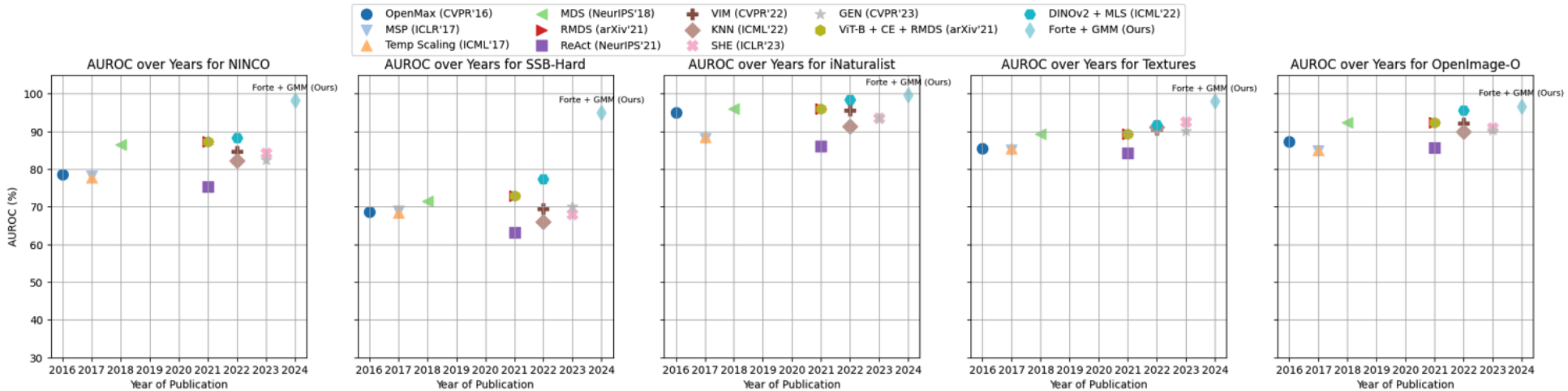


Figure 16: Comparison of AUROC performance across various supervised out-of-distribution detection methods and datasets from the OpenOOD leaderboard. The figure presents results for five datasets (NINCO, SSB-Hard, iNaturalist, Textures, and OpenImage-O) with each subplot showcasing the progression of AUROC scores over the years. Each method is represented with a unique marker and color. Our method "Forte + GMM" is highlighted for its superior performance, demonstrating strong state-of-the-art results across all datasets. The x-axis represents the publication year, while the y-axis denotes the AUROC (%) scores.

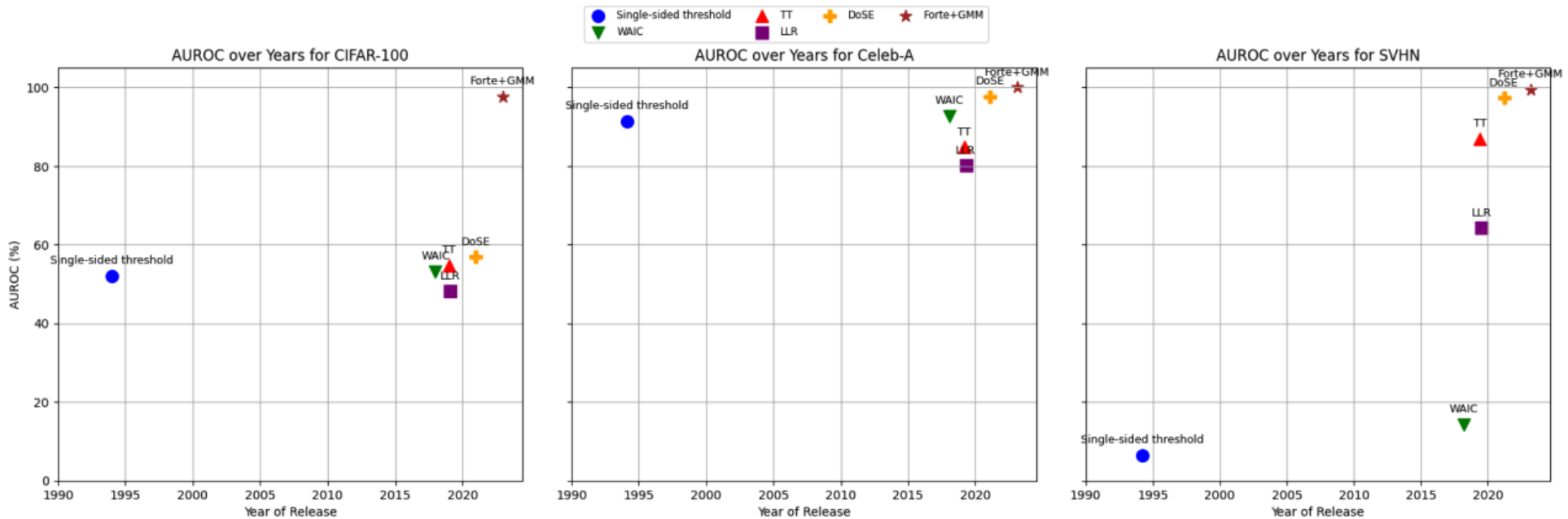


Figure 15: AUROC performance of various unsupervised OOD detection methods over years for the CIFAR-10 (In-distribution) and the CIFAR-100(OOD), Celeb-A(OOD) and SVHN(OOD) dataset. The figure illustrates the progression of anomaly detection techniques, with methods represented using unique markers and colors.

Domain Generalization

Identifying Synthetic Data

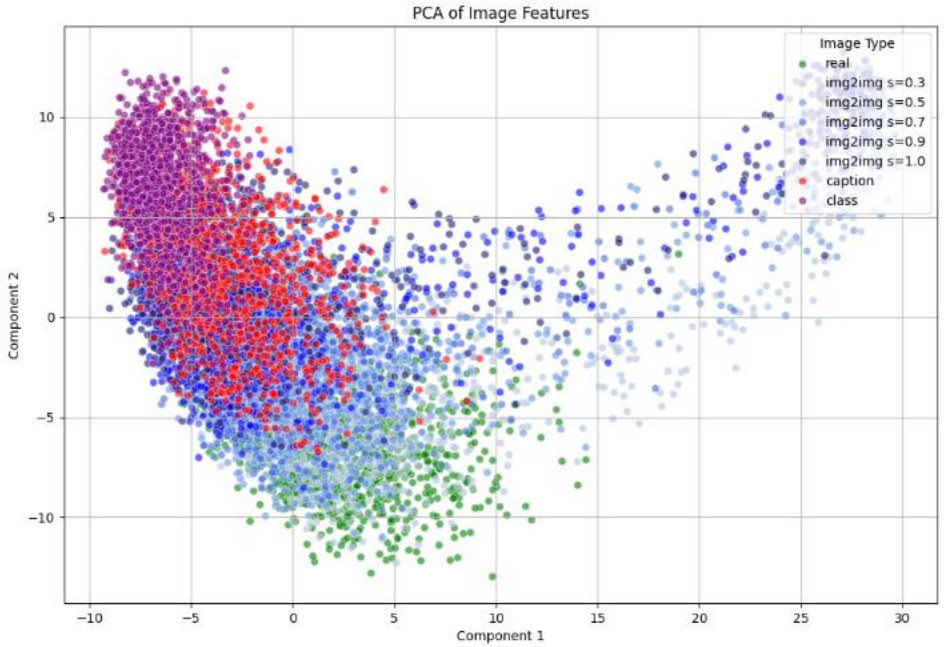
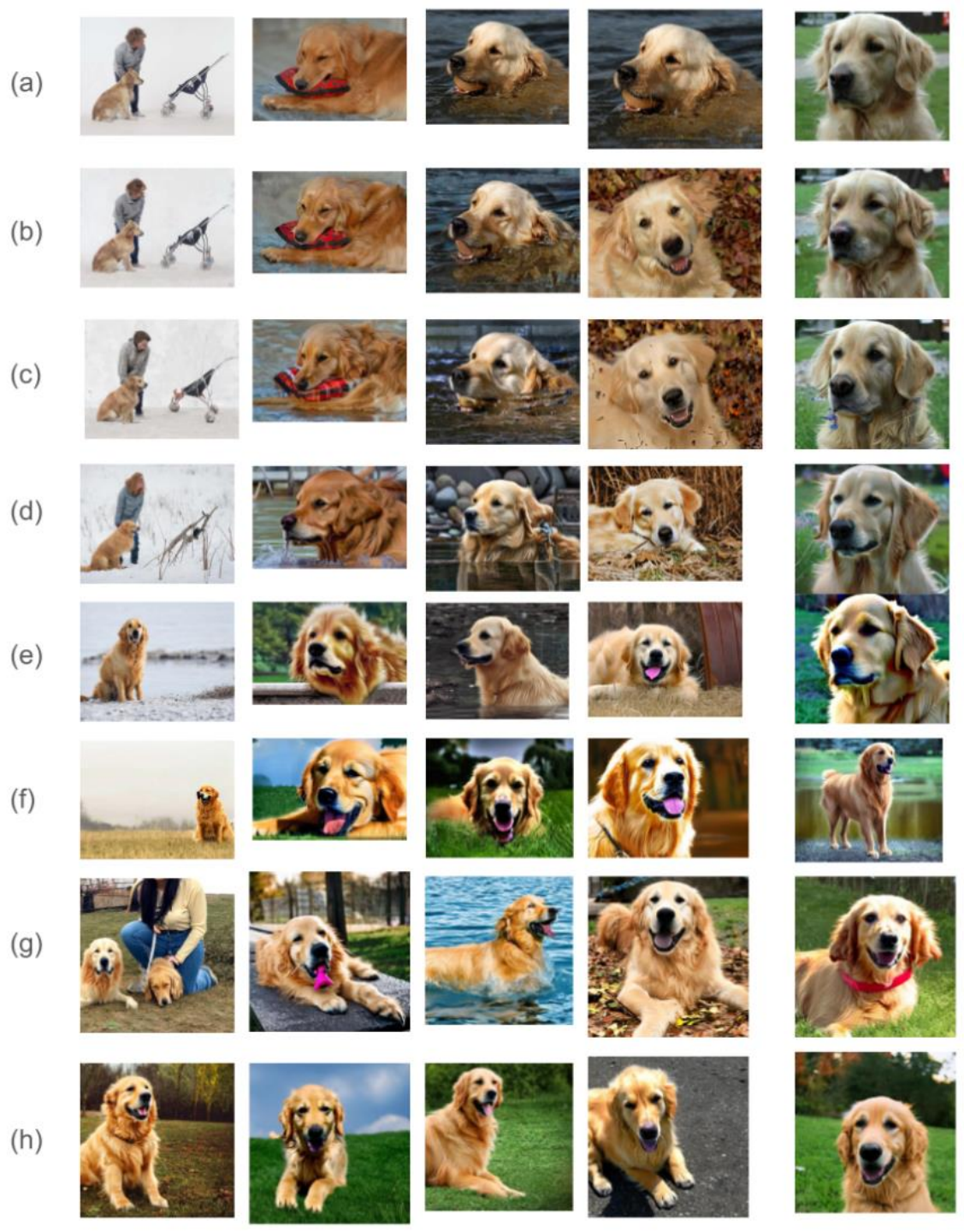


Figure 5: PCA of Golden Retriever Images in CLIP space. We can clearly observe that the distribution of real images is most diverse, whereas the "classname" based generated images are least diverse. We also see a progression of images tending towards the modes of the distribution with increasing strengths allotted to the diffusion process.

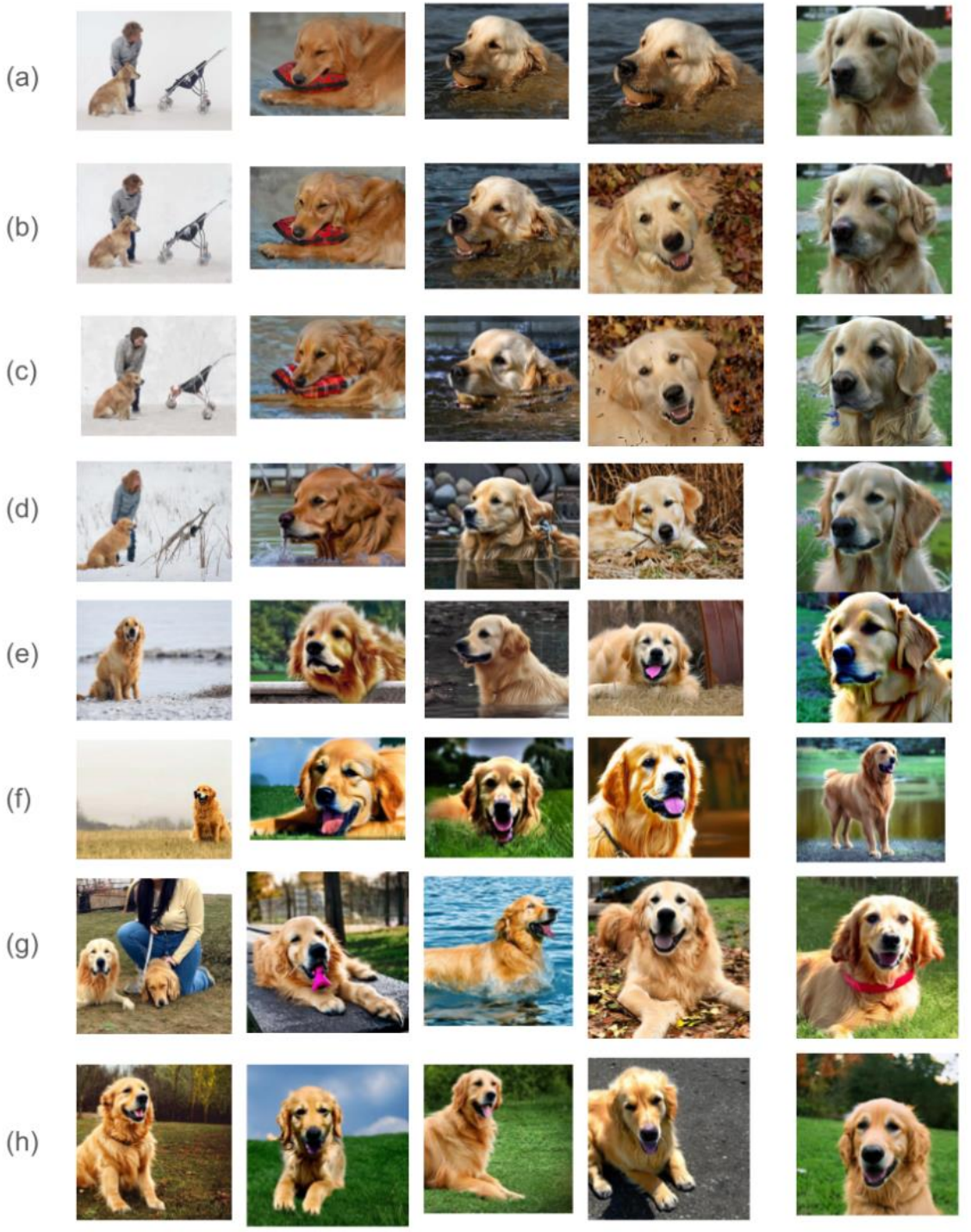
Synthetic images generated for the class "golden retriever" using different techniques. Row (a) shows real images, while rows (b) to (f) display Img2Img generated images with strength parameters 0.3, 0.5, 0.7, 0.9, and 1.0, respectively. Row (g) presents images generated using caption descriptions of the real images in row (a), and row (h) shows images generated solely based on the classname.



Identifying Synthetic Data

Table 4: Performance comparison of Forte+GMM against a CLIP-based baseline, and Gen Eval methods for detecting synthetic images of Golden Retrievers generated using various techniques.

	FD	FD_∞	CMMD Score	Baseline Model CLIP		Forte+GMM	
				AUROC	FPR95	AUROC	FPR95
Img2Img S=0.3	453.63	418.22	0.52	61.19	86.27	**68.28 ± 02.14**	**68.07 ± 05.56**
Img2Img S=0.5	648.97	624.01	0.64	59.49	86.27	**82.93 ± 02.50**	**46.80 ± 10.68**
Img2Img S=0.7	762.17	735.42	0.64	61.23	86.36	**94.19 ± 01.85**	**24.90 ± 12.99**
Img2Img S=0.9	845.96	819.06	0.66	59.05	88.87	**97.59 ± 01.23**	**14.41 ± 13.55**
Img2Img S=1.0	891.39	870.17	0.73	60.15	89.23	**98.11 ± 00.75**	**06.09 ± 05.72**
Caption-based	575.18	546.42	0.90	80.71	71.54	**96.77 ± 01.14**	**18.90 ± 14.38**
Class-based	1,065.96	1,048.18	1.07	75.73	82.23	**98.26 ± 01.12**	**10.22 ± 13.04**



Synthetic images generated for the class "golden retriever" using different techniques. Row (a) shows real images, while rows (b) to (f) display Img2Img generated images with strength parameters 0.3, 0.5, 0.7, 0.9, and 1.0, respectively. Row (g) presents images generated using caption descriptions of the real images in row (a), and row (h) shows images generated solely based on the classname.

Identifying MRI Protocols

Experimental Setup

- **In-distribution data**: FastMRI (2 subsets: FS and NoFS)
- **Out-of-distribution data**: OAI (3 subsets: T1, MPR, TSE)
- Testing which new datasets can align with existing models
- Measuring degree of divergence between subsets

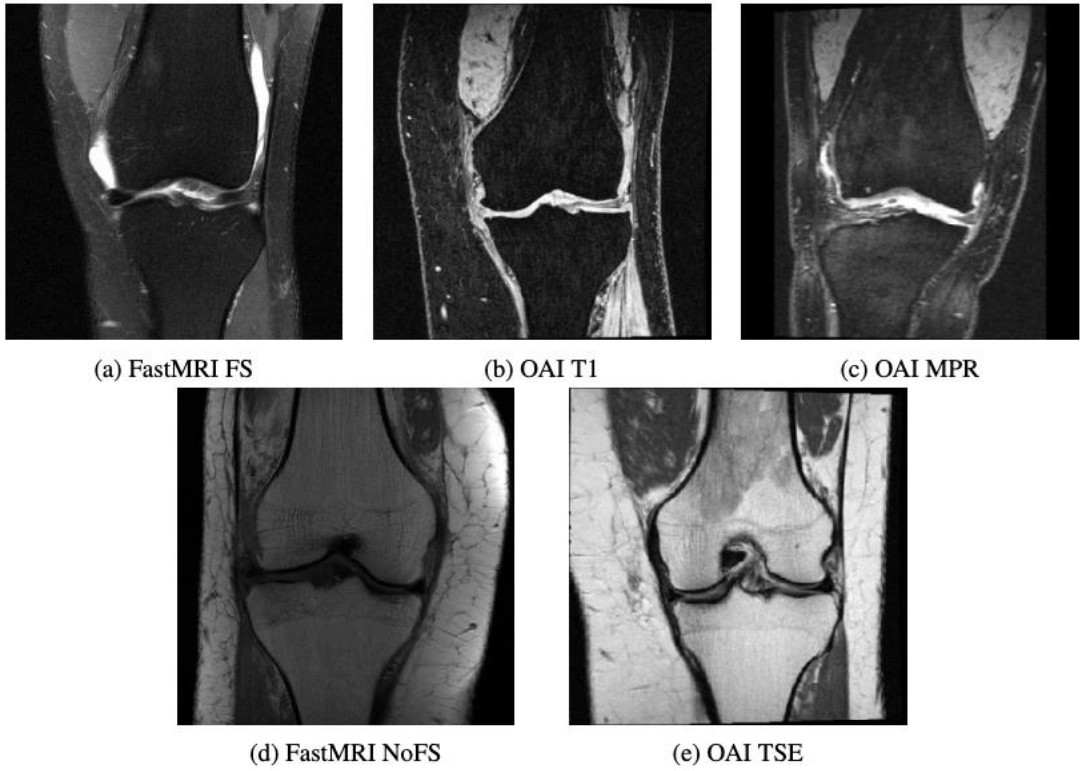
Objective

Using Forte to determine when confronted with new datasets:

1. Which subsets can be aligned with existing models?
2. How much do these subsets diverge from each other?

Table 5: Out-of-distribution (OOD) detection using Forte, applied to medical image datasets. Strong performance by Forte suggests the presence of batch effects and a need for data harmonization. Refer to Appendix E for more details on the FastMRI and OAI datasets. For reliable estimation, performance is measured over 10 random seeds.

Method	Metric	In-Dist: OOD:	FastMRI NoFS OAI TSE	FastMRI FS OAI T1	FastMRI FS OAI MPR
Forte+SVM (Ours)	**AUROC**		**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
	FPR95		**00.00 ± 00.00**	**00.00 ± 00.00**	**00.00 ± 00.00**
Forte+KDE (Ours)	**AUROC**		97.97 ± 5.63	95.98 ± 7.84	95.99 ± 7.86
	FPR95		9.49 ± 26.76	19.75 ± 39.10	19.77 ± 39.31
Forte+GMM (Ours)	**AUROC**		99.95 ± 0.13	99.91 ± 0.16	99.91 ± 0.17
	FPR95		0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00



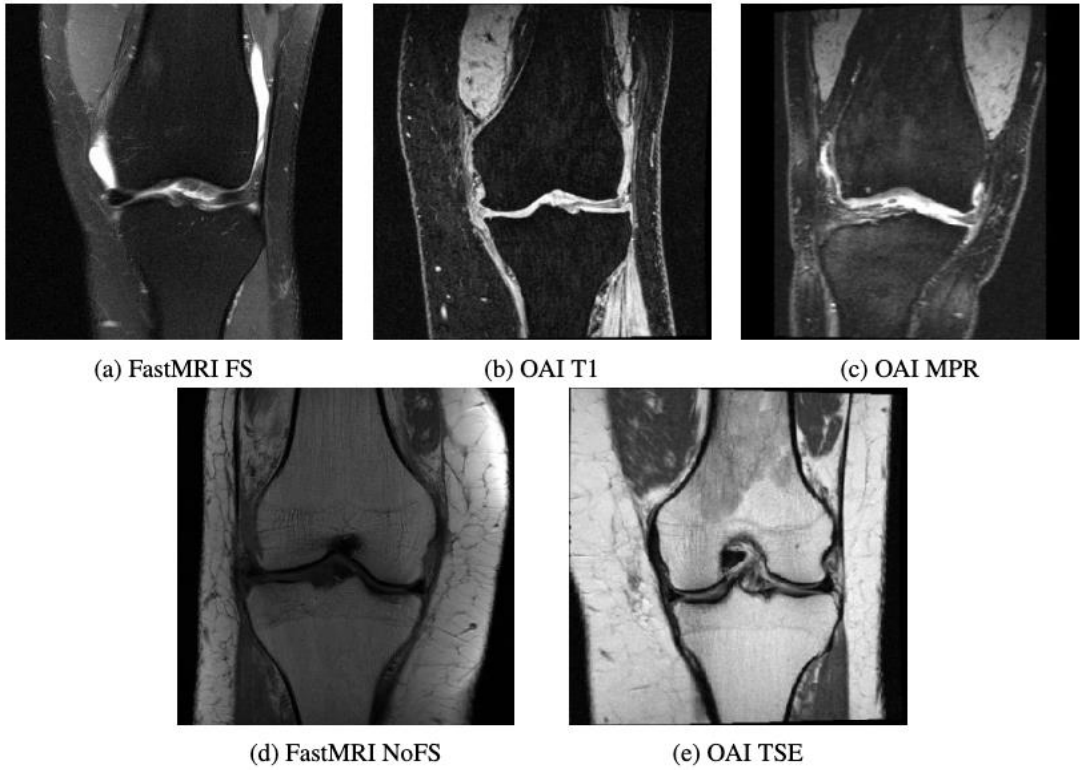
Sample images from five total subsets of the OAI and FastMRI datasets. Fat-suppressed (a-c) and non fat-suppressed (d,e) subsets tested for OOD.

Identifying MRI Protocols

Table 9: Acquisition parameters for MRI, grouped by distributions as used in Section 5.2.

Parameter	FastMRI NoFS	FastMRI FS	OAI TSE	OAI T1	OAI MPR
Sequence	2D TSE PD	2D TSE PD	2D TSE T1w	3D FLASH T1w	3D DESS T2w
FOV (mm^2)	140 × 140	140 × 140	140 × 140	140 × 140	140 × 140
Matrix size	320 × 320	320 × 320	320 × 320	384 × 384	320 × 320
Slice thickness (mm)	3	3	3	0.7	1.5
TR (ms)	2750–3000	2850–3000	800	9.7	14.7
TE (ms)	27–32	33	9	4.0	4.2
Fat suppression	No	Yes	No	Yes	Yes

- MRI datasets exhibit strong batch effects
- Models trained on in-house data perform poorly on new datasets
- Limited clinical dataset sizes prevent batch-specific models
- Even similar acquisition parameters carry enough differences to impact performance



Sample images from five total subsets of the OAI and FastMRI datasets. Fat-suppressed (a-c) and non fat-suppressed (d,e) subsets tested for OOD.

Encoder : Use domain specific embeddings?

Insight into encoders: The results presented in Table. 3 demonstrate that richer representations significantly improve OOD detection performance, as evidenced by the ranking of encoders: CLIP > DINO v2 > MSN when used individually, and CLIP + DINO v2 > CLIP + MSN > DINO v2 + MSN in the two-model combinations. To further investigate this phenomenon, we conducted additional experiments. Specifically, we included DeIT with both ViT-B (Base) and ViT-Ti (Tiny) models and evaluated their OOD detection performance under the settings studied in Tables 2. These results, show that DeIT-B achieves 0.87 AUROC on CIFAR-100, while DeIT-Ti achieves 0.82 AUROC. This aligns with our hypothesis that more informative representations are essential for effective OOD detection. DeIT, trained with an objective approximating supervision, produces less informative embeddings compared to self-supervised encoders like DINO v2. Similarly, the DeIT-Ti model performs worse due to its reduced capacity for generating robust representations. We think these findings provide valuable insights into the utility of different encoders for OOD detection and offer guidance for practitioners seeking optimal performance.

Table 8: Comparison of AUROC and FPR95 performance figures for Base and Tiny DeIT models across the tasks in

Model	In-Dist	OOD Dataset	AUROC	FPR95
Base-DeIT	CIFAR-10	CIFAR-100	0.8712	0.9926
Tiny-DeIT	CIFAR-10	CIFAR-100	0.8261	0.9903
Base-DeIT	CIFAR-10	SVHN	0.9554	0.4604
Tiny-DeIT	CIFAR-10	SVHN	0.9296	0.6195
Base-DeIT	CIFAR-10	Celeb-A	0.9871	0.0015
Tiny-DeIT	CIFAR-10	Celeb-A	0.9929	0.0007

Do you want to deploy on edge? Use with specialized data?
Strong evidence to support that you could swap out DinoV2 etc for your own tiny ViT/CNN (encoder).

Using Forte

With your own data

Notes on the library

- Library fits within ~500 lines of Python, single standalone file and class.
- API provides Scikitlearn type fit() and predict() functions, very easy to use.
- On mac : Embedding is about 0.1s/img, Inference of downstream model is about 0.03 seconds per image.