Optimization Can Learn Johnson Lindenstrauss Embeddings

Nikos Tsikouras
Constantine Caramanis
Christos Tzamos

Dimensionality Reduction

- The main objective: reduce the dimensionality of the space
- Project a <u>d-dimensional dataset</u> into a <u>k-dimensional space</u> such that:
  - $k << d$
  - Retain property of interest (e.g. preserve pairwise distances)

Advantages in working in the new space:

- Algorithms train faster
- Less complexity
- Less storage
- …

Our Goal

Given a d-dimensional dataset we want to project the dataset into k dimensions while approximately preserving the L_2 norm of each point.

Given dataset $x_1, \ldots, x_\mathrm{n} \in \mathbb{R}^d$, then for any $0 < \varepsilon < 1$, and a sufficiently large k we would like to find a matrix $A \in \mathbb{R}^{k \times d}$ such that for all x:

$$(1-\varepsilon)||x||^2 \leq ||Ax||^2 \leq (1+\varepsilon)||x||^2$$

Our Goal

Given a d-dimensional dataset we want to project the dataset into k dimensions while approximately preserving the L_2 norm of each point.

Given dataset $x_1, \ldots, x_n \in \mathbb{R}^d$, then for any $0 < \varepsilon < 1$, and a sufficiently large k we would like to find a matrix $A \in \mathbb{R}^{k \times d}$ such that for all x:

$$(1-\varepsilon)\cancel{||x||^2} \leq ||Ax||^2 \leq (1+\varepsilon)\cancel{||x||^2}$$ w.l.o.g unit norm

Our Goal

For all x:

$$(1-\varepsilon) \leq ||Ax||^2 \leq (1+\varepsilon)$$

JL GUARANTEE

The Johnson-Lindenstrauss Lemma

The Johnson-Lindenstrauss (JL) lemma states that:

- There **always exists** such a matrix
- You can construct it in a **randomized** way

Gaussian Construction

$$A = \underbrace{\begin{pmatrix} ? & \cdots & ? \\ \vdots & \ddots & \vdots \\ ? & \cdots & ? \end{pmatrix}}_{d} \Bigg\} k$$

Gaussian Construction

$$A = \underbrace{\begin{pmatrix} N(0,1) & \cdots & N(0,1) \\ \vdots & \ddots & \vdots \\ N(0,1) & \cdots & N(0,1) \end{pmatrix}}_{d} \left.\vphantom{\begin{pmatrix} N \\ \vdots \\ N \end{pmatrix}}\right\} k$$

Gaussian Construction

$$A = \underbrace{\begin{pmatrix} N(0,1) & \cdots & N(0,1) \\ \vdots & \ddots & \vdots \\ N(0,1) & \cdots & N(0,1) \end{pmatrix}}_{d} \left.\vphantom{\begin{pmatrix} N \\ \vdots \\ N \end{pmatrix}}\right\} k$$

- Works fast with high probability
- Data agnostic

Can we do better?

Optimization Approach

A naive approach would be to directly optimize matrix A. That is:

$$h(A) = max_{x_{1, \dots, x_n}} \underbrace{\left| \|Ax\|^2 - 1 \right|}_{\text{Distortion}}$$

Optimization Approach

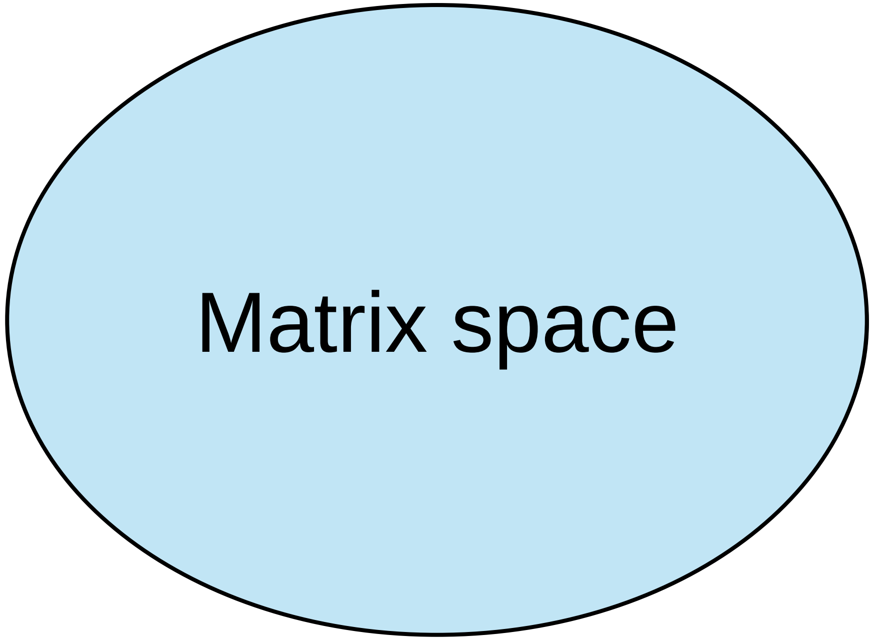
A naive approach would be to directly optimize matrix A. That is:

$$h(A) = max_{x_{1, \ldots, x_n}} \left| ||Ax||^2 - 1 \right|$$

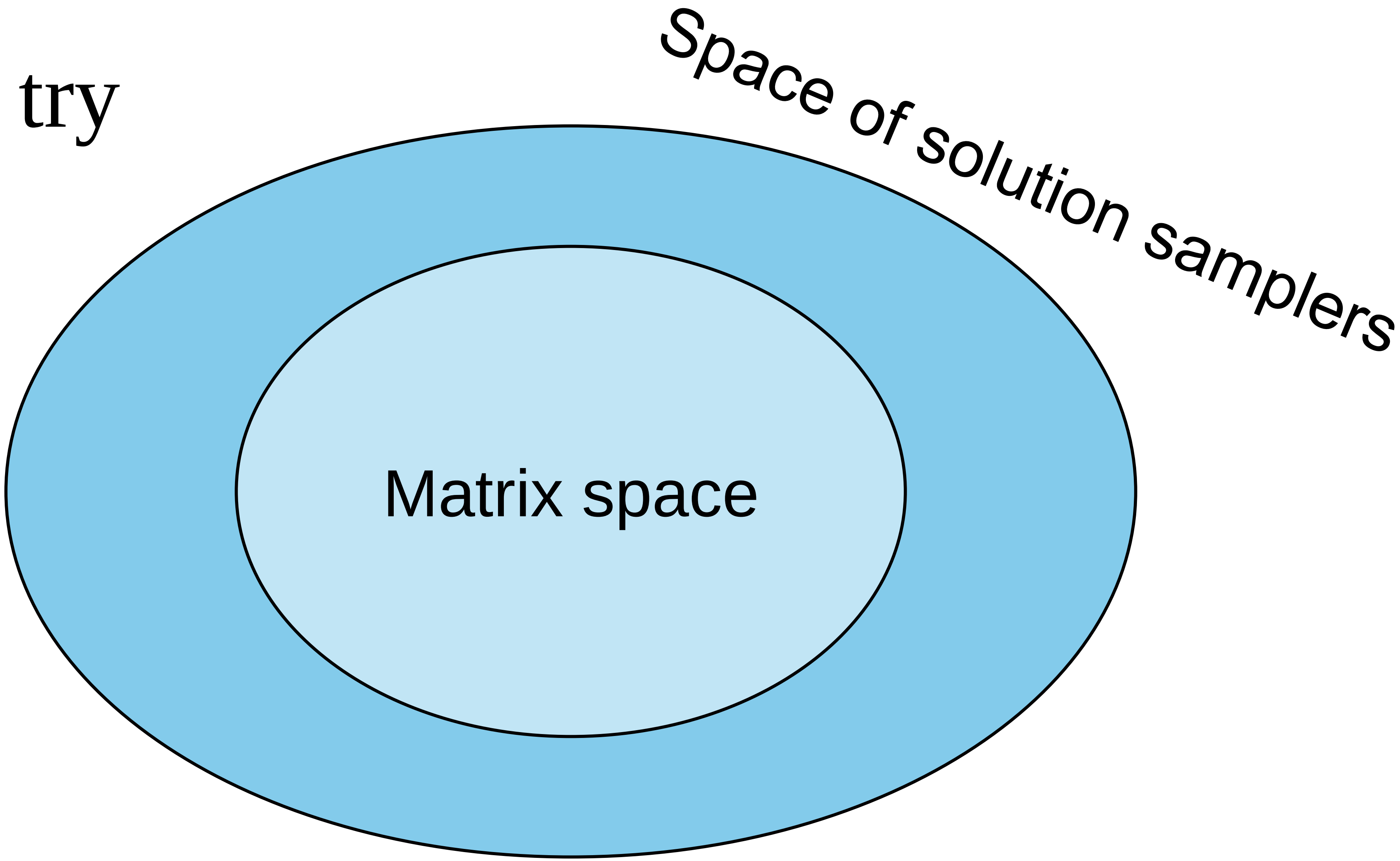
This cannot work as our first result shows:

The maximum distortion objective considered as a function in the space of matrices has many suboptimal local minima.

Second try

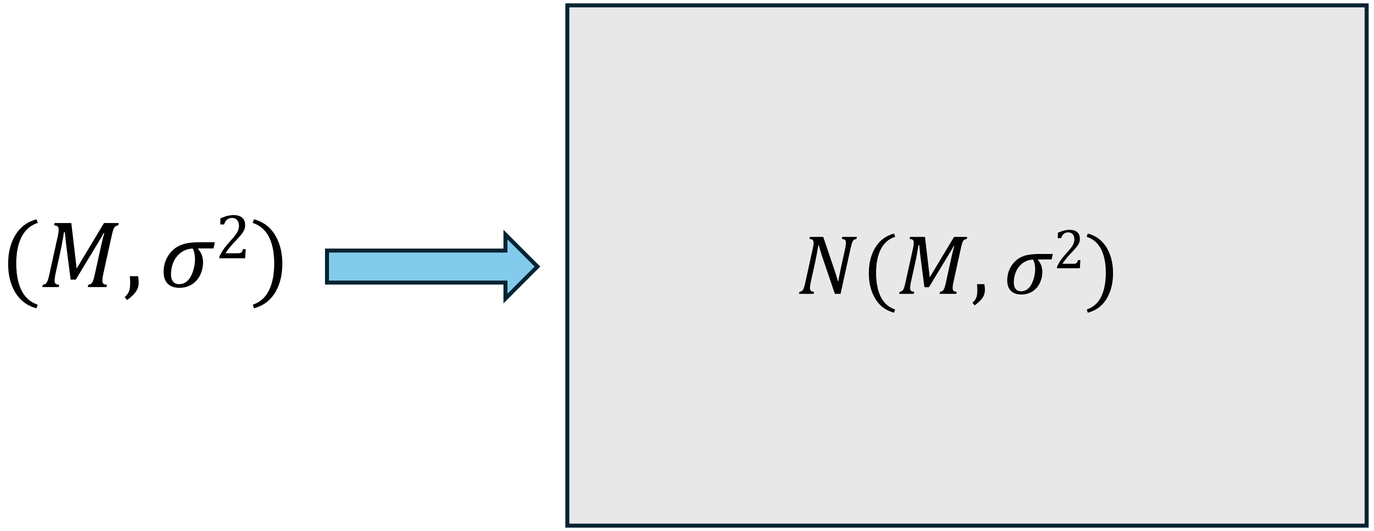


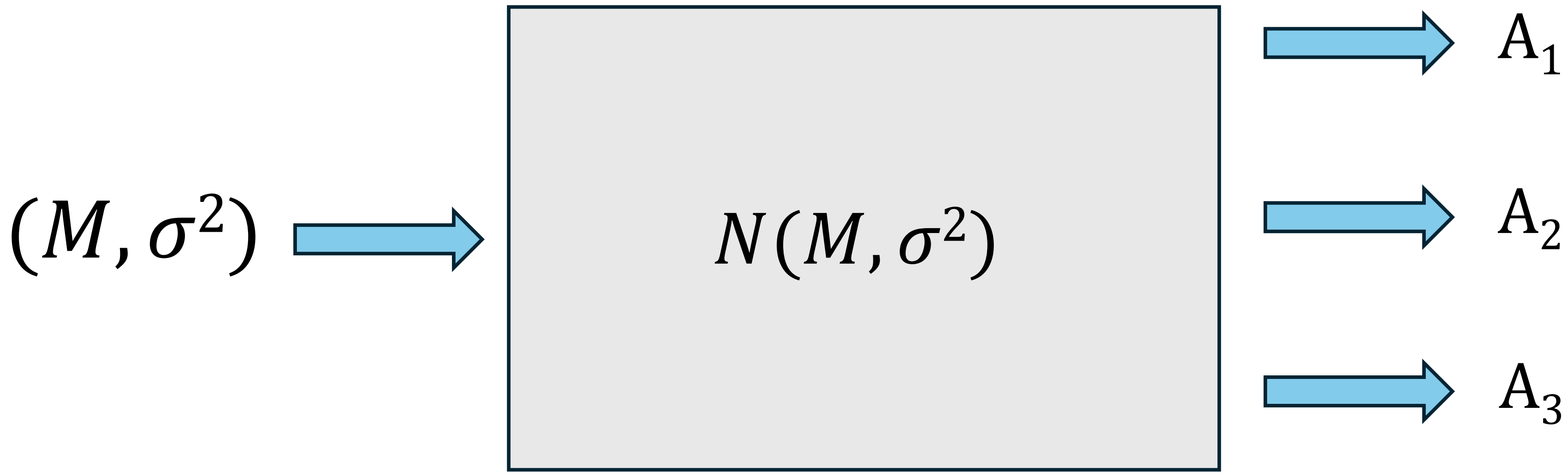
Second try



What are solution samplers?

(M, σ^2)


(M, σ^2)
$N(M, \sigma^2)$



(M, σ^2)
$N(M, \sigma^2)$
A_1
A_2
A_3

Main idea

We want to optimize in the space of solution samplers and find optimal parameters that generate high quality matrices which satisfy the JL guarantee.

At the end of the optimization, we would like to have a mean matrix $\boldsymbol{M}^*$ and variance 0 such that when we sample from $N(\boldsymbol{M}^*, 0)$, i.e. deterministically sample $\boldsymbol{M}^*$, we have a matrix that satisfies the JL guarantee.

- We define the matrix of means:

$$\boldsymbol{M} = \begin{pmatrix} \mu_{11} & \cdots & \mu_{\mathrm{k1}} \\ \vdots & \ddots & \vdots \\ \mu_{1\mathrm{k}} & \cdots & \mu_{\mathrm{kd}} \end{pmatrix}$$

- And a common variance σ^2

Then our objective function is:

$$f(\boldsymbol{M}, \sigma^2) = \sum_{j=1}^{n} Pr\left[\left\|Ax_j\right\|^2 \notin (1-\varepsilon, 1+\varepsilon)\right]$$

Then our objective function is:

$$N(\boldsymbol{M}, \sigma^2)$$

$$f(\boldsymbol{M}, \sigma^2) = \sum_{j=1}^{n} Pr\left[\left\|Ax_j\right\|^2 \notin (1-\varepsilon, 1+\varepsilon)\right]$$

Then our objective function is:

$$f(\boldsymbol{M}, \sigma^2) = \sum_{j=1}^{n} Pr\left[\left\|Ax_j\right\|^2 \notin (1-\varepsilon, 1+\varepsilon)\right] \quad + \quad \frac{\sigma^2}{2}$$

Then our objective function is:

$$f(\boldsymbol{M}, \sigma^2) = \sum_{j=1}^{n} Pr\left[\left\|Ax_j\right\|^2 \notin (1-\varepsilon, 1+\varepsilon)\right] \quad + \quad \underbrace{\frac{\sigma^2}{2}}$$

Ensures consistent reduction of variance

Then our objective function is:

$$f(\boldsymbol{M}, \sigma^2) = \underbrace{\sum_{j=1}^{n} Pr\left[\left\|Ax_j\right\|^2 \notin (1-\varepsilon, 1+\varepsilon)\right]}_{\text{Probability that a projected data point does not satisfy the required distortion}} + \underbrace{\frac{\sigma^2}{2}}_{\text{Ensures consistent reduction of variance}}$$

Probability that a projected data point does not satisfy the required distortion

Ensures consistent reduction of variance

Our results

Our main result is two-fold:

First, the qualitative aspect indicates that out optimization landscape exhibits a desirable property.

All second-order stationary points reachable from the origin for our objective function have zero variance and hence correspond to fixed matrices.

Moreover, these matrices satisfy the JL guarantee.

Second, the quantitative aspect demonstrates that we can efficiently minimize our objective function and learn a deterministic JL embedding.

Two step algorithm:

1. If the gradient is sufficiently large, we take a **gradient step.**
2. Otherwise, if the smallest eigenvalue is sufficiently negative, we take a step in that **direction of negative curvature.**

Two step algorithm:

1. If the gradient is sufficiently large, we take a **gradient step.**
2. Otherwise, if the smallest eigenvalue is sufficiently negative, we take a step in that **direction of negative curvature.**

Running the algorithm above for poly(n,k,d) steps returns a matrix that satisfies the JL guarantee deterministically.

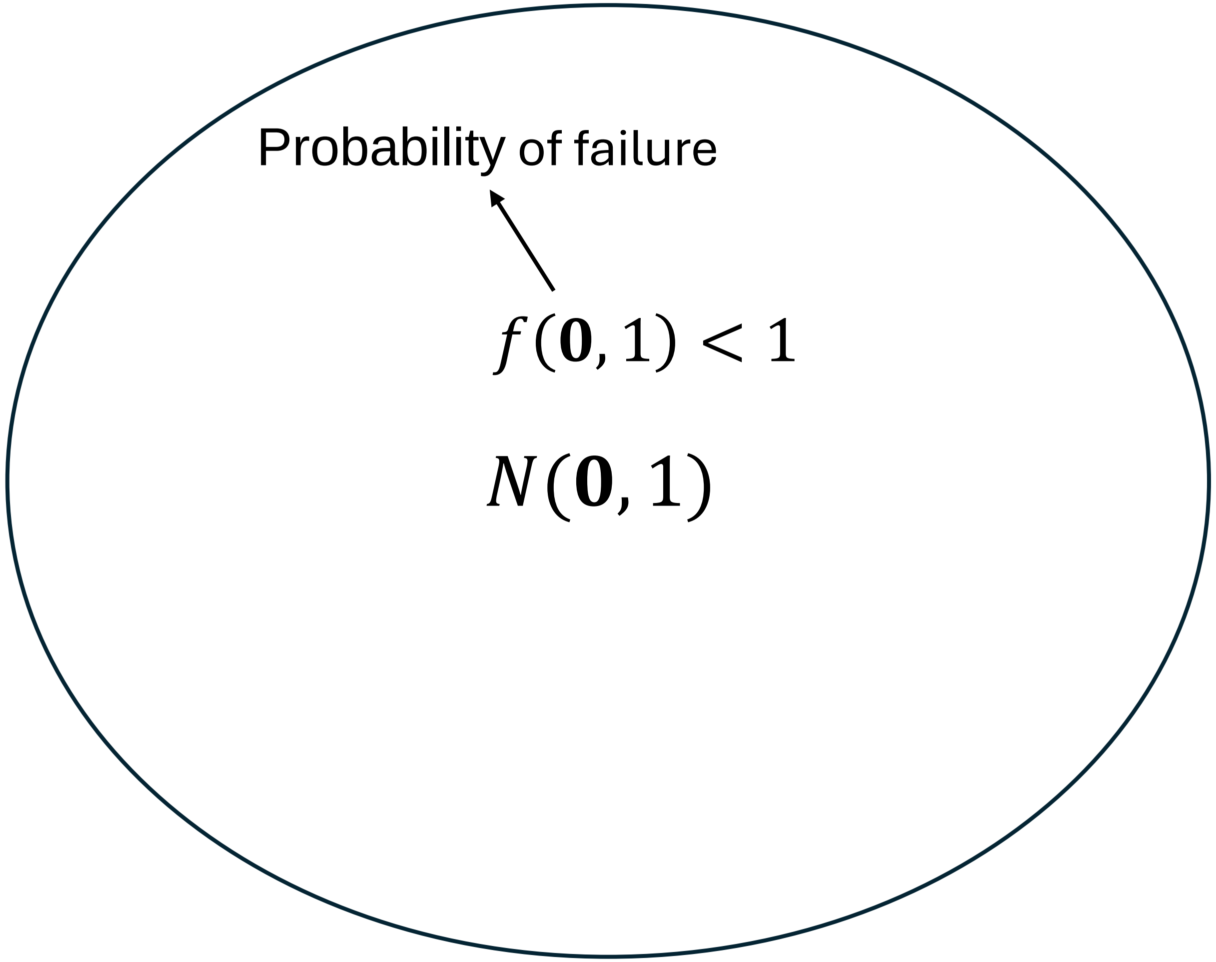
We note that this theorem constitutes a novel approach to *derandomizing* the Gaussian JL transformation.

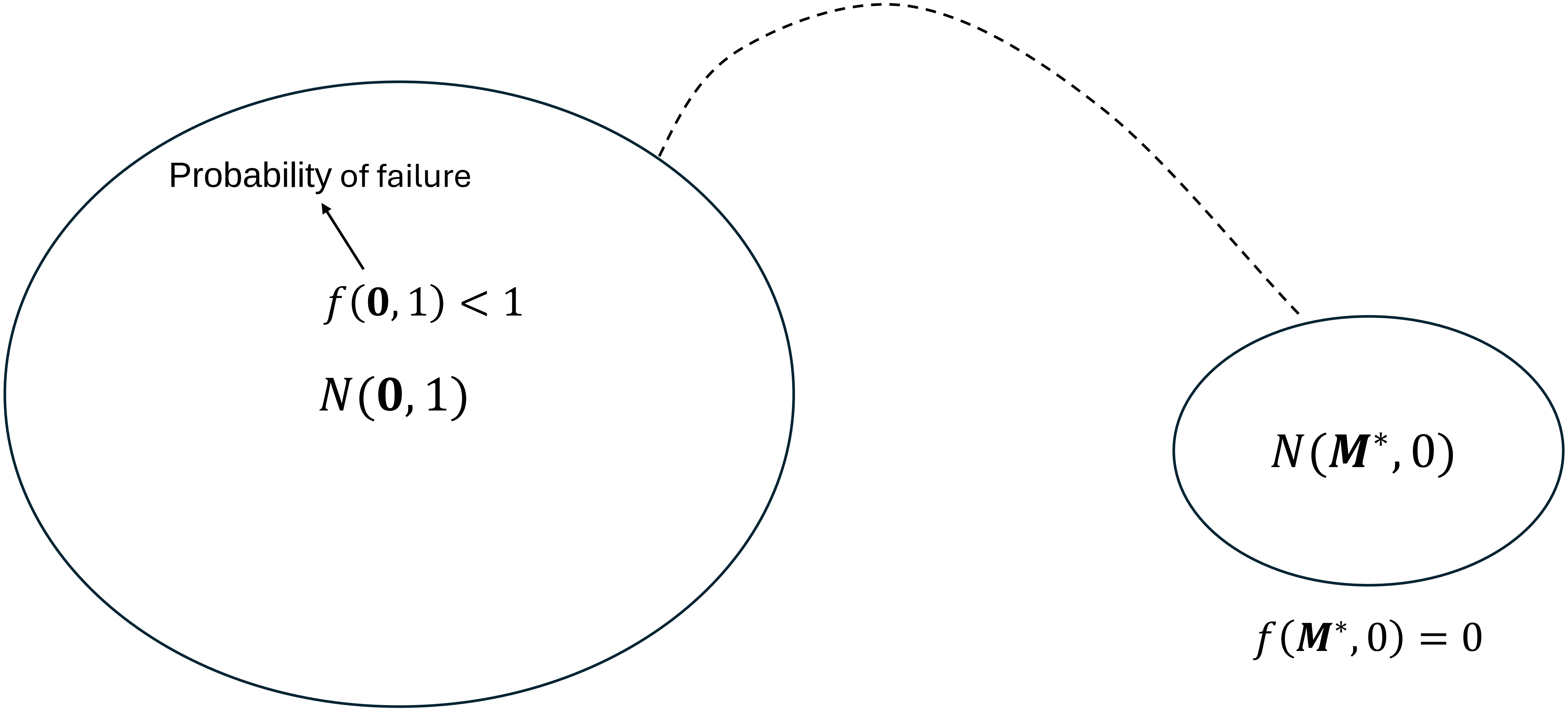
Proof Sketch

- We initialize the matrix of means:

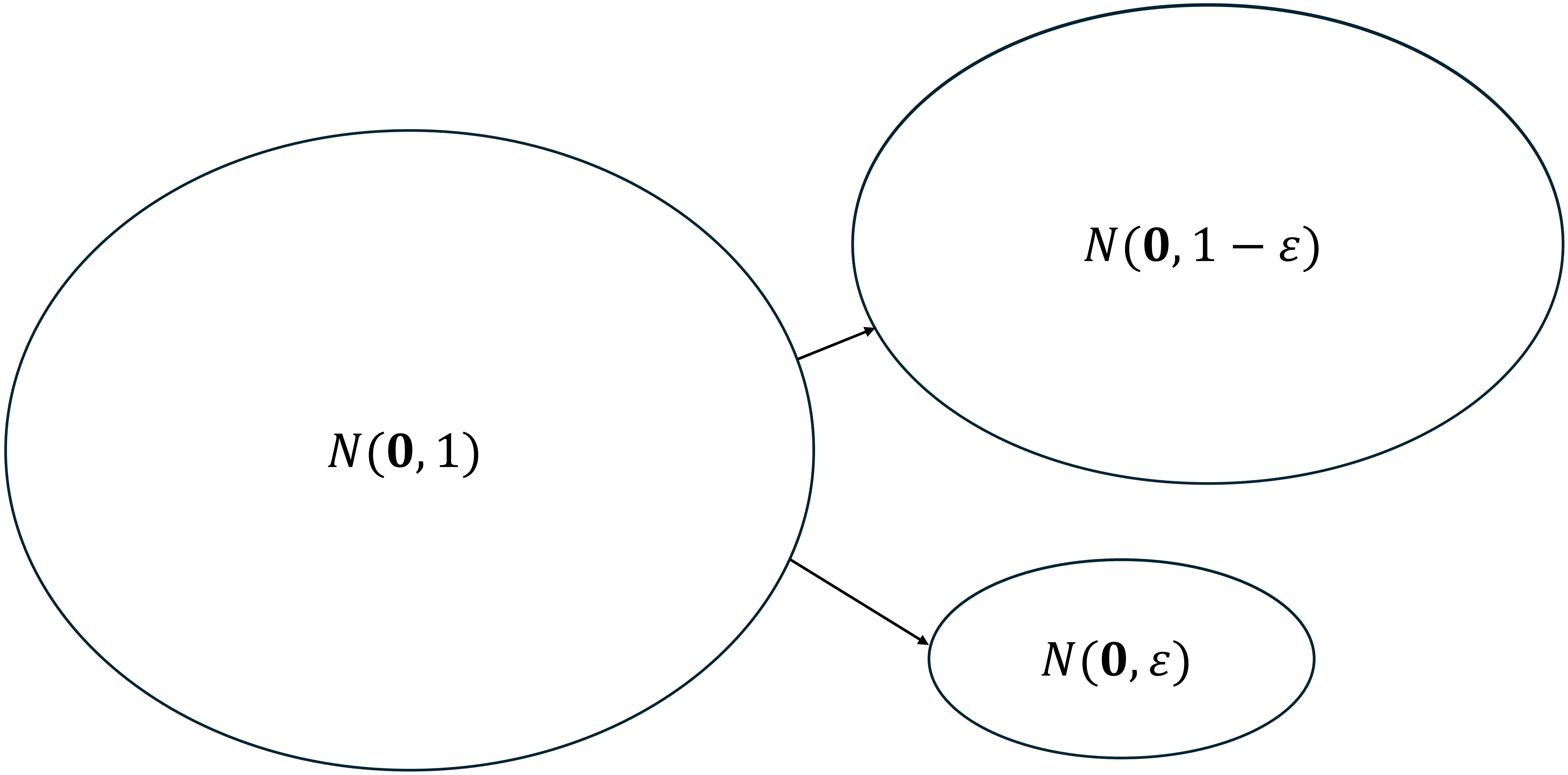
$$\boldsymbol{M} = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} = \mathbf{0}$$

- And a common variance $\sigma^2 = 1$


Probability of failure
$f(\mathbf{0},1) < 1$
$N(\mathbf{0},1)$


Probability of failure
$f(\mathbf{0},1) < 1$
$N(\mathbf{0},1)$
$N(\boldsymbol{M}^*,0)$
$f(\boldsymbol{M}^*,0) = 0$

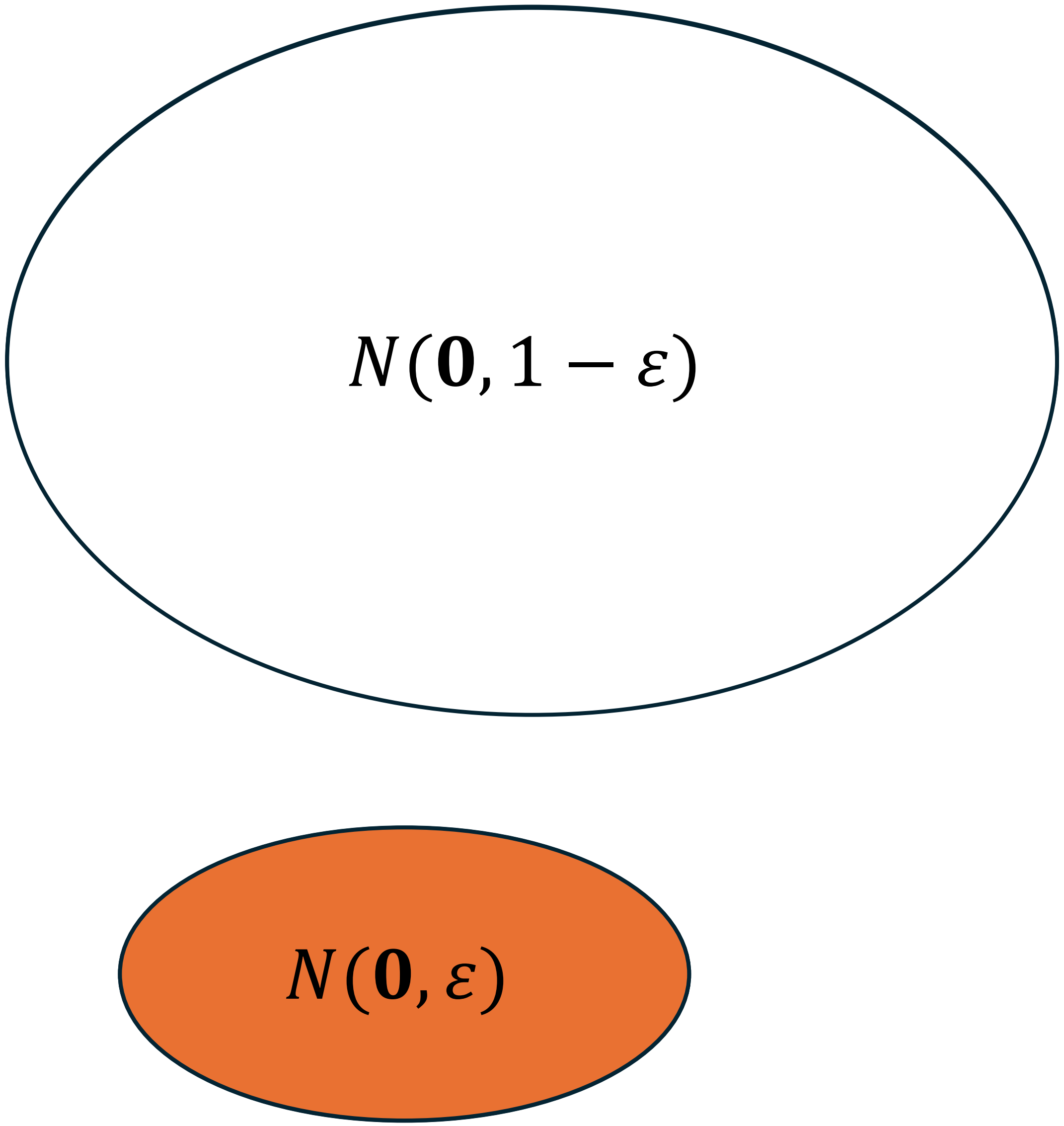
$N(\mathbf{0}, 1)$

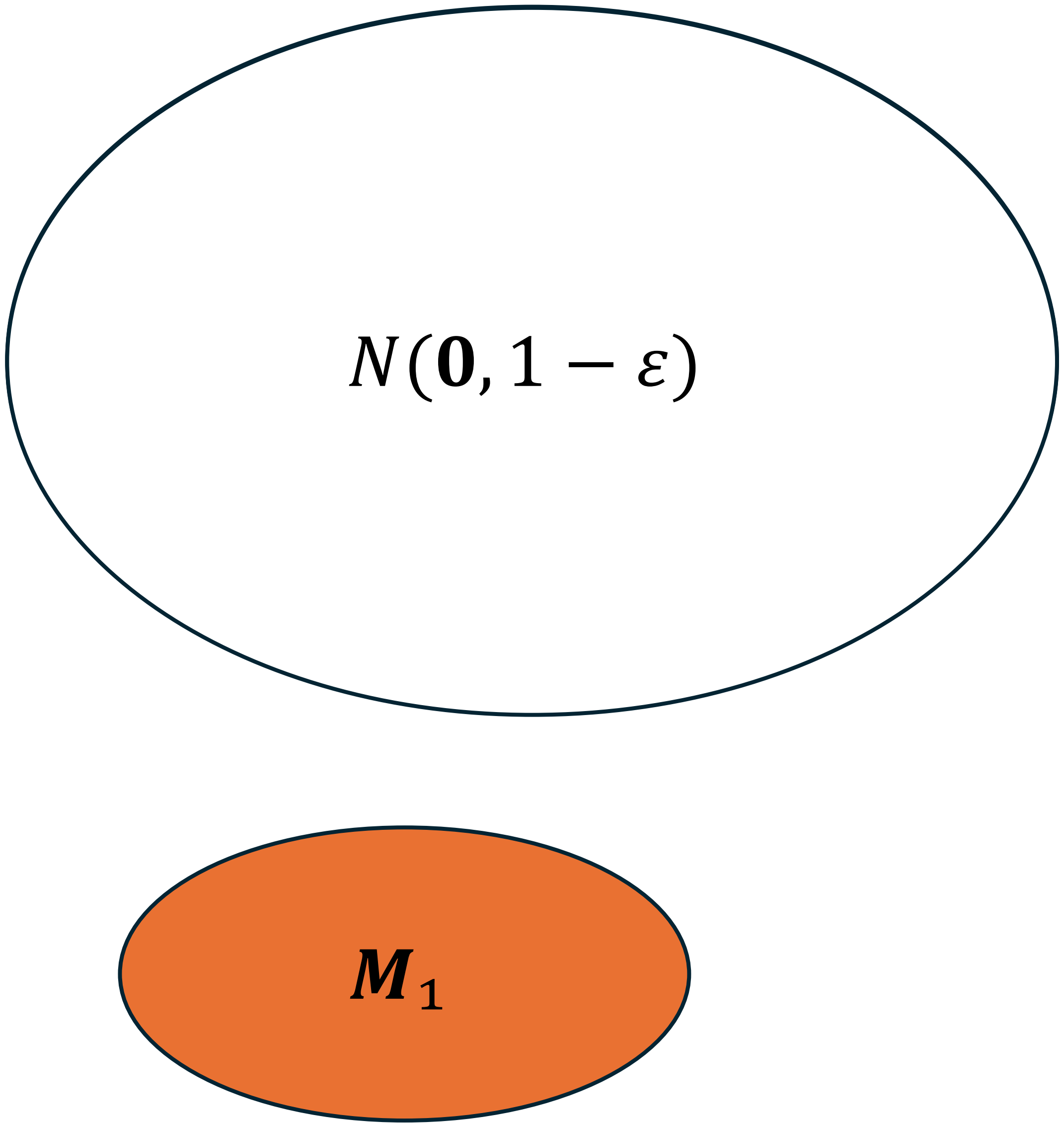


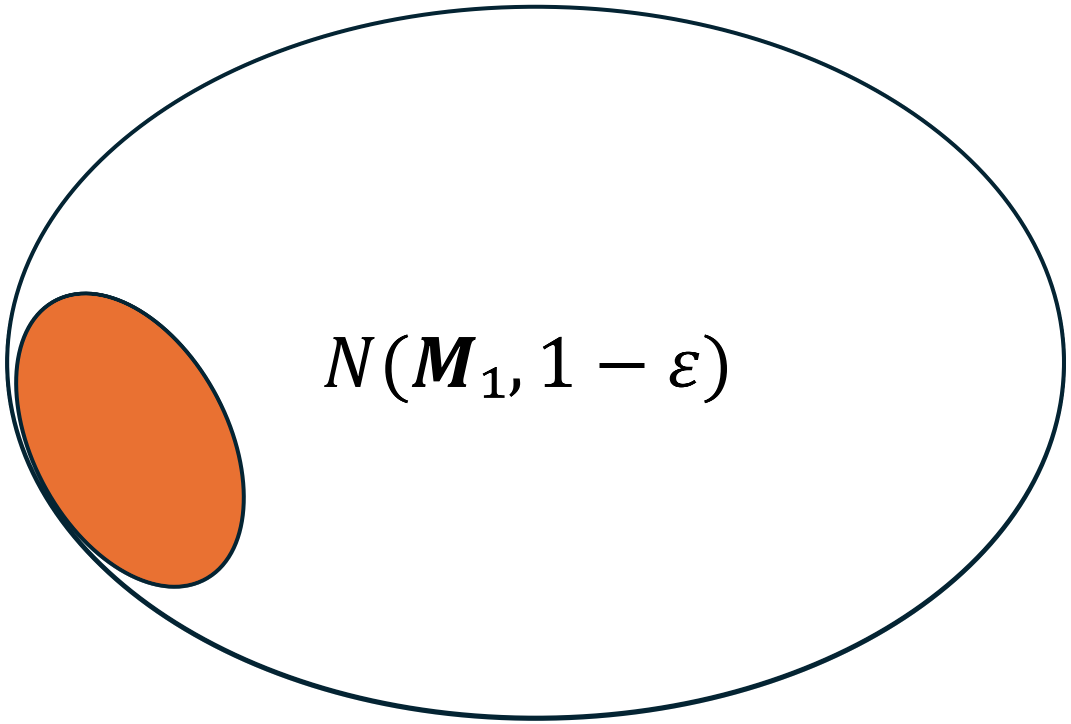
$N(\mathbf{0}, 1)$
$N(\mathbf{0}, 1 - \varepsilon)$
$N(\mathbf{0}, \varepsilon)$

$$N(\mathbf{0}, 1 - \varepsilon)$$

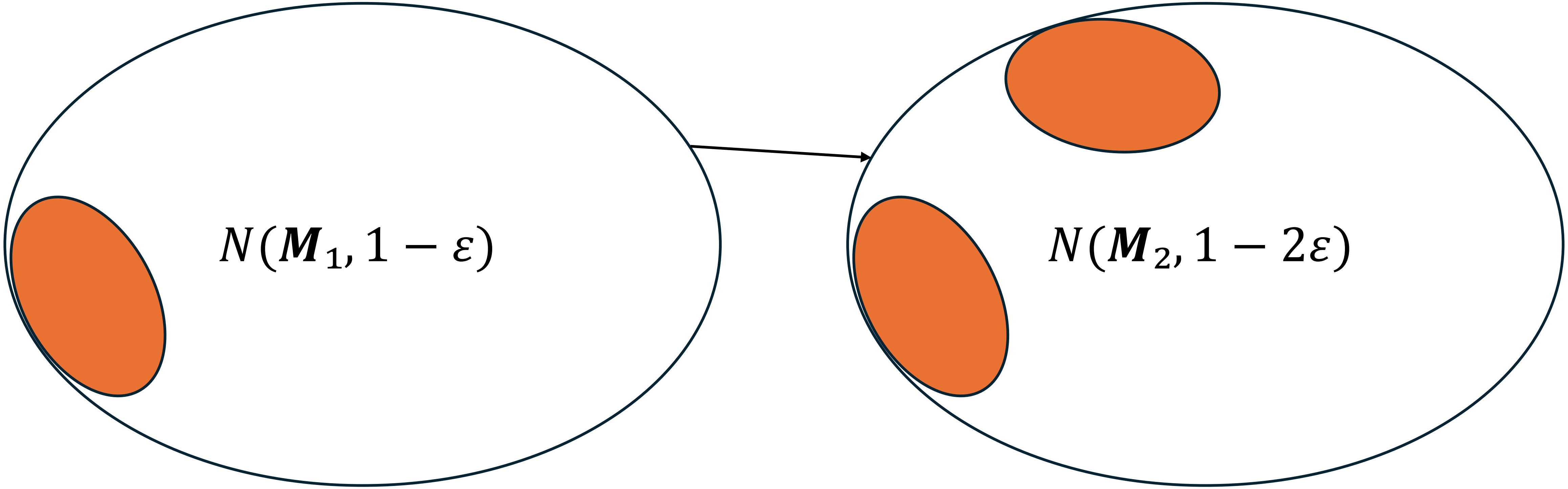
$$N(\mathbf{0}, \varepsilon)$$

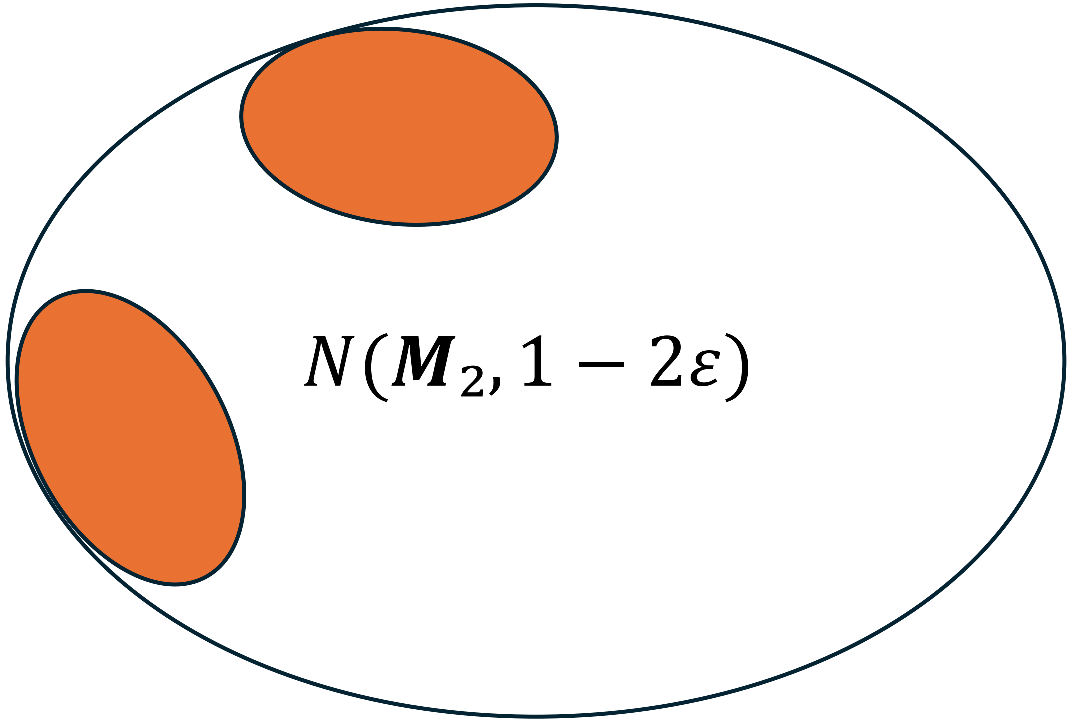

$N(\mathbf{0}, 1-\varepsilon)$
$N(\mathbf{0}, \varepsilon)$


$N(\mathbf{0}, 1-\varepsilon)$
$\boldsymbol{M}_1$

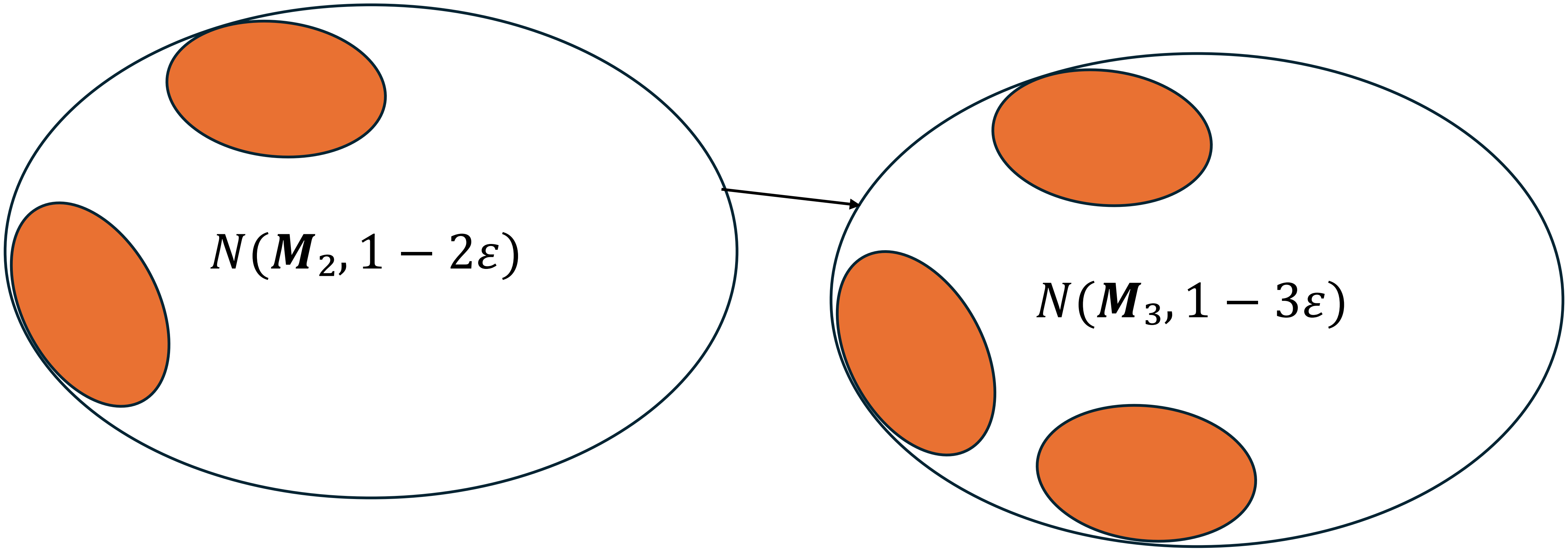


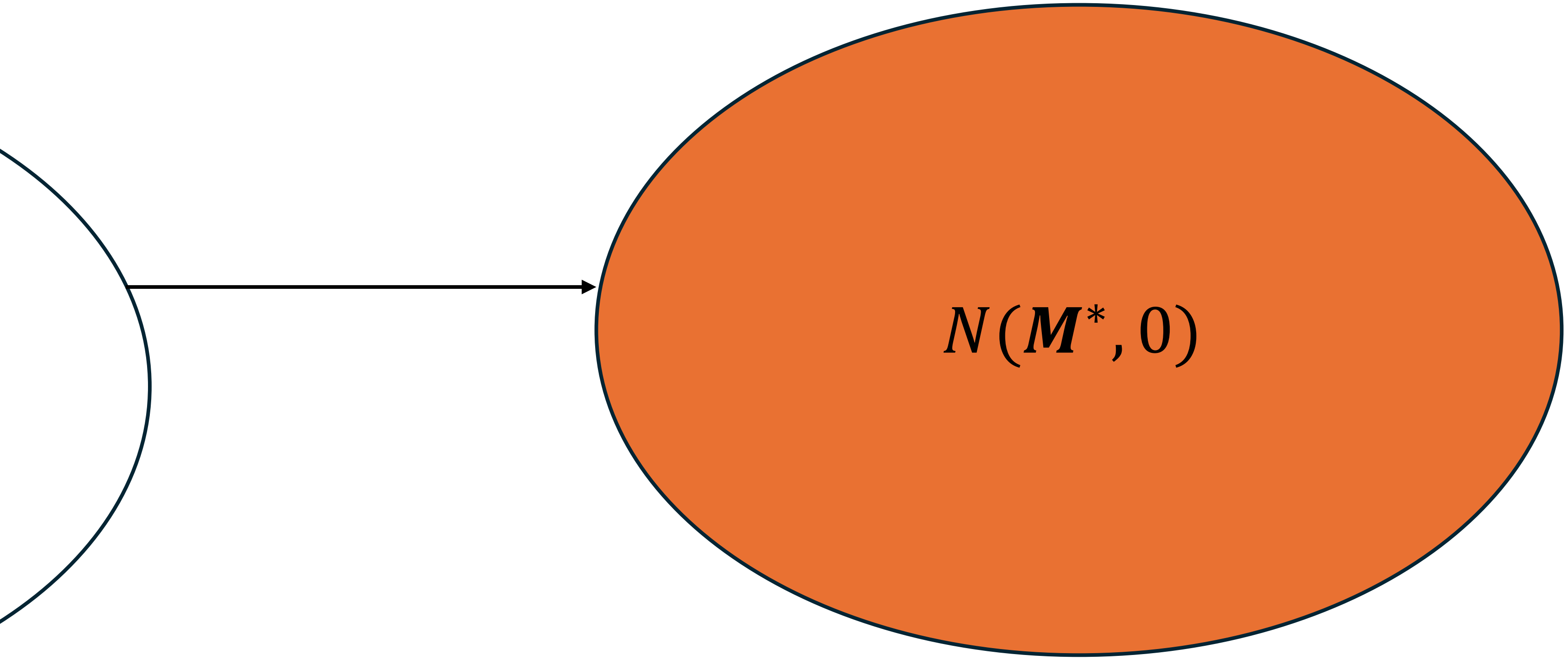
$N(\boldsymbol{M}_1, 1 - \varepsilon)$

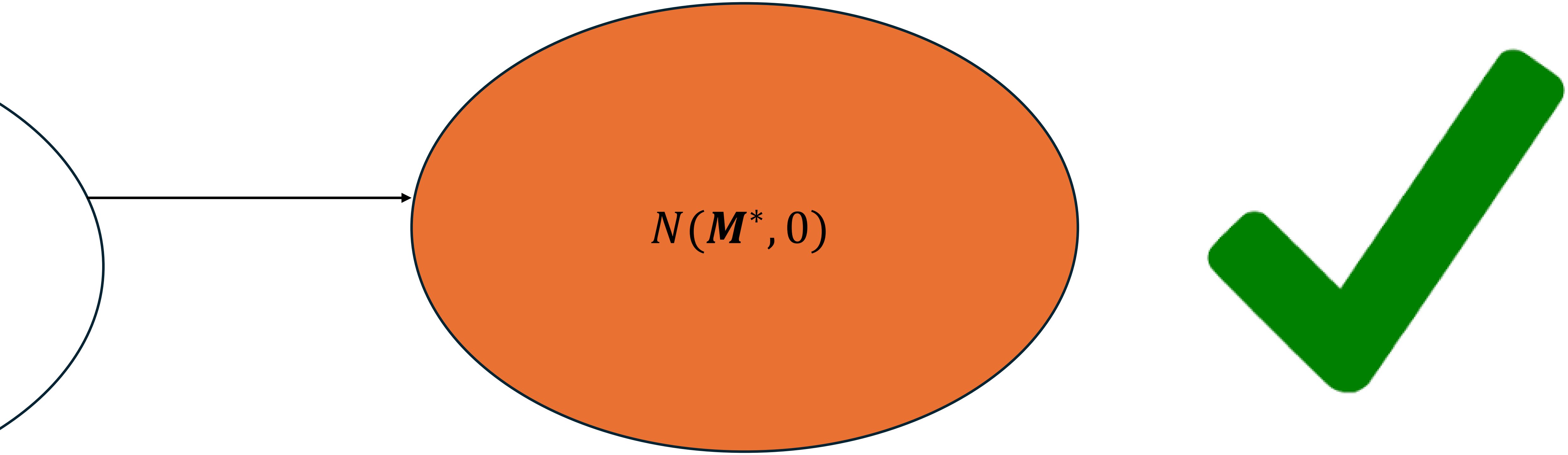

$N(\boldsymbol{M}_1, 1-\varepsilon)$
$N(\boldsymbol{M}_2, 1-2\varepsilon)$

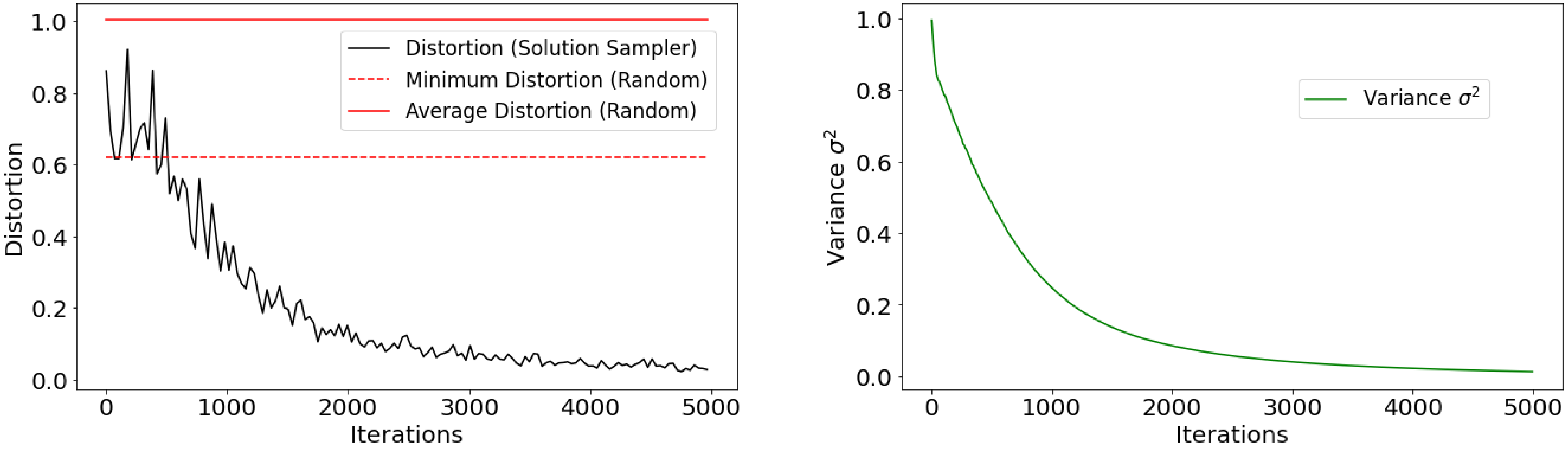


$N(\boldsymbol{M}_2, 1-2\varepsilon)$


$N(\boldsymbol{M}_2, 1-2\varepsilon)$
$N(\boldsymbol{M}_3, 1-3\varepsilon)$


$N(\boldsymbol{M}^{*}, 0)$


$N(\boldsymbol{M}^*, 0)$



Distortion (Solution Sampler)
Minimum Distortion (Random)
Average Distortion (Random)
Distortion
Iterations
Variance σ^2
Variance σ^2
Iterations

THANK YOU

If you have any questions, feel free to email me ☺

Website: https://nikostsikouras.github.io/