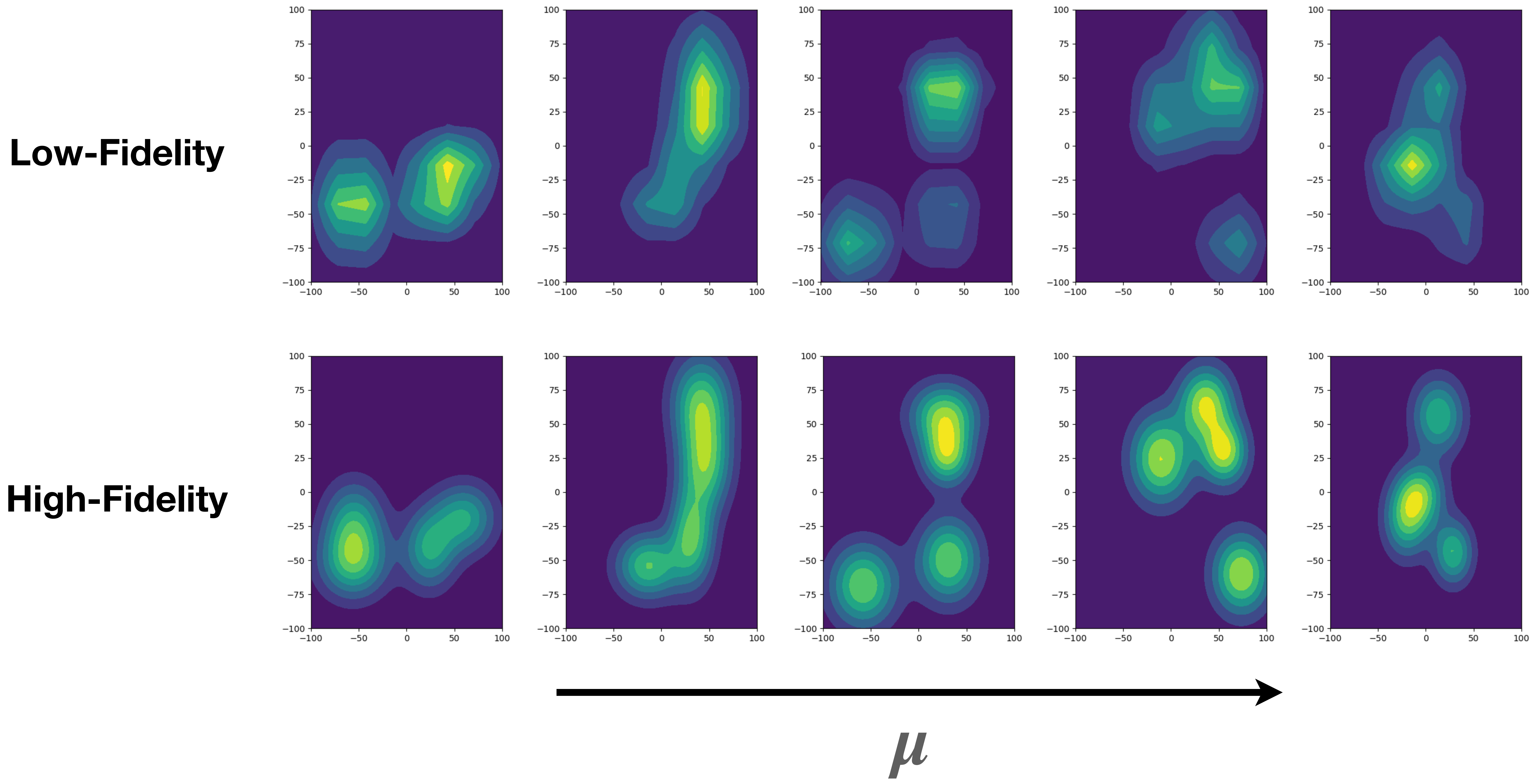
Multi-Fidelity Modeling

Orazio Pinti

- Consider a **parametric physical problem** $x(\boldsymbol{\mu})$, where $x \in \mathbb{R}^D$ indicates the system state or a generic quantity of interest (QoI), and $\boldsymbol{\mu} \in \mathbb{R}^P$ a vector of parameters.

- Assume that **two computational models** are available to solve the problem numerically: a low-fidelity model $x^L = \mathscr{M}^L(\boldsymbol{\mu})$, and a high-fidelity model $x^H = \mathscr{M}^H(\boldsymbol{\mu})$.



We want to construct a **multi-fidelity predictive model** $\mathscr{M}^M(\boldsymbol{\mu})$ to study the behavior of the QoI x in the parameter space $\boldsymbol{\mu}$. This has to be done by employing the low-fidelity model N times, and the high-fidelity model only $n \ll N$ times.

What's a good strategy?

Proposed approach

Step 1:

- Perform LF simulations $\mathscr{D}^L = \{x_i^L\}_{i=1}^N$.
- Compress the LF data with an auto-encoder $h(\,.\,) = d \circ e(\,.\,)$ s.t. $e : x^L \to z^L$ and $d : z^L \to \hat{x}^L$, with $x^L \in R^D$, $z^L \in R^d$ and $d \ll D$.

Step 2:

- Clustering over the latent space: $\mathscr{C}_i = \{z_j^L \,|\, j \in J_i\}$, $i = 1,\ldots,k$ and $\cup_{i=1}^k J_i = \{1,\ldots,N\}$.

Step 3:

- Perform HF simulations $\mathscr{D}^H = \{x_i^H\}_{i=1}^k$ at parameters corresponding to centroids of clusters.
- Train h with all the data $\mathscr{D} = \mathscr{D}^L \cup \mathscr{D}^H$, obtaining $\tilde{h}(\,.\,) = \tilde{d} \circ \tilde{e}(\,.\,)$. Try to preserve the cluster structures $\mathscr{C}_i$ with a soft constraint.

Step 4:

- Compress HF data with encoder $\tilde{e} : x^H \to z^H$

Step 5:

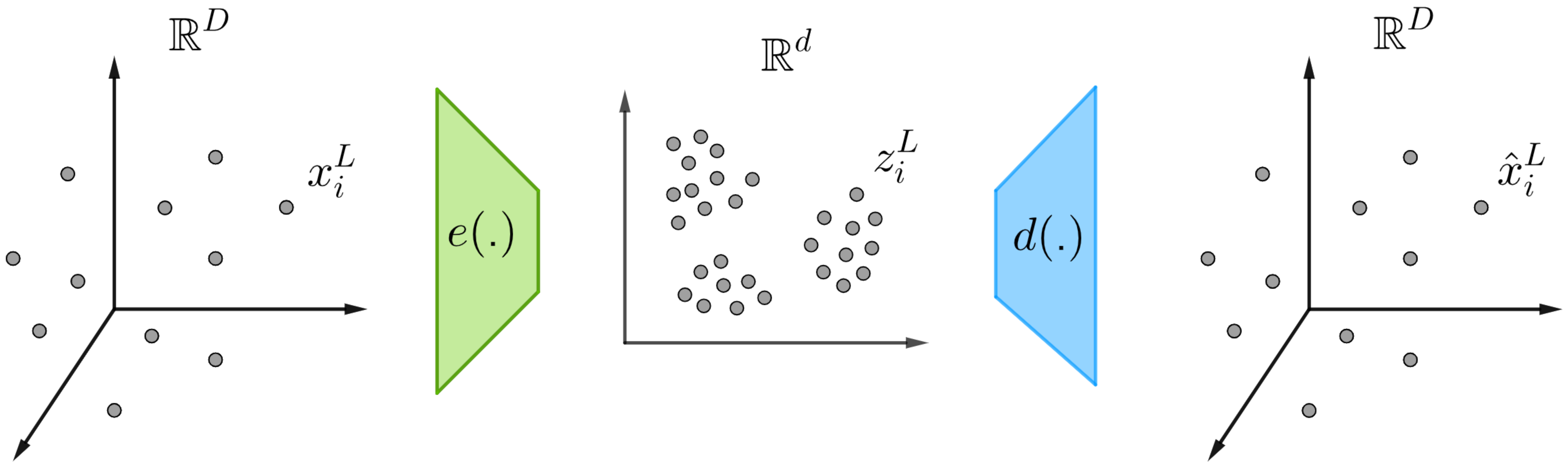
- Transform every embedding z^L to a MF approximation with $t_i : z_i^L \to z_i^M$

Step 6:

- Decode the embedding $\tilde{d} : z^M \to x^M$

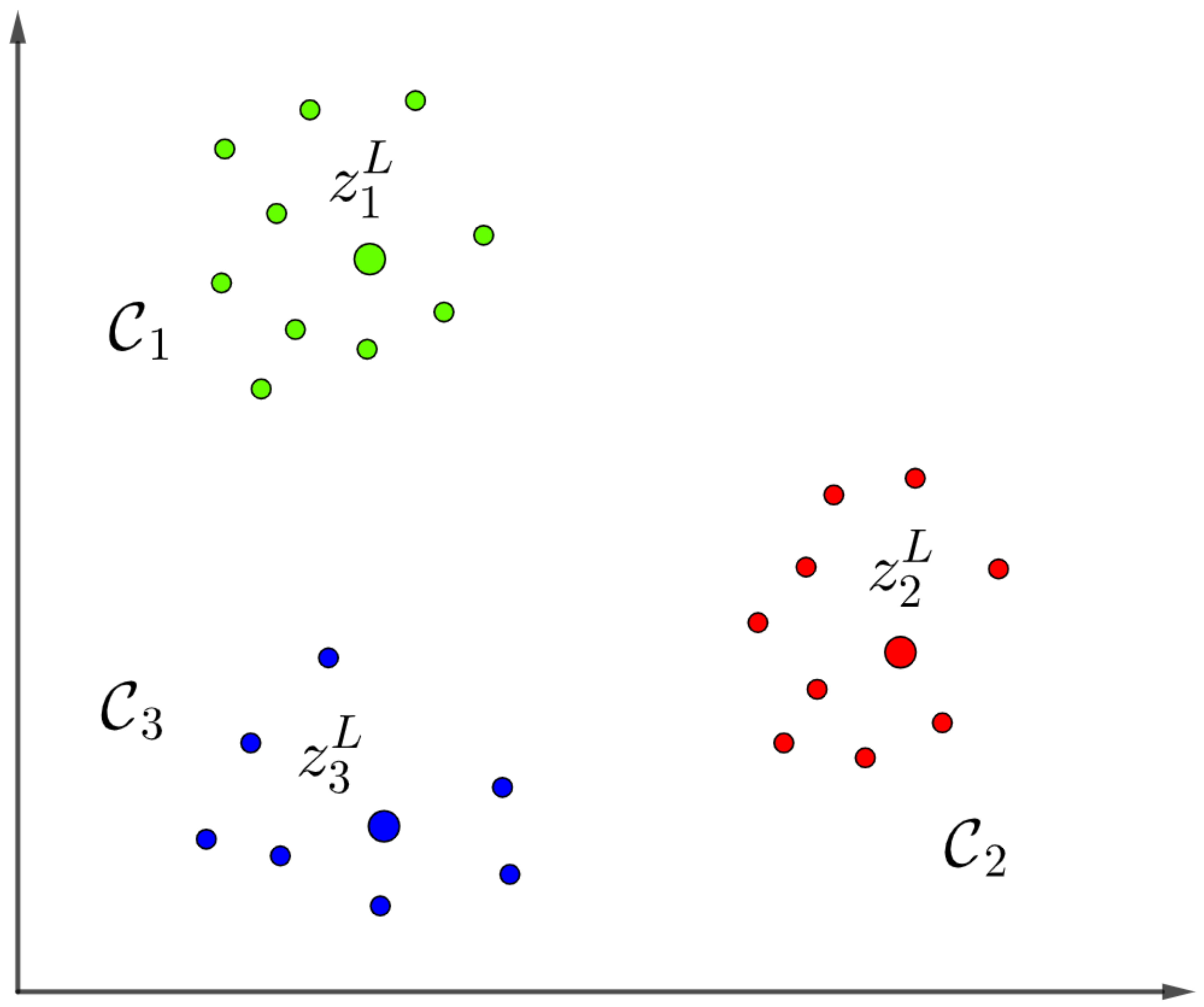
Step 1:

- Perform LF simulations $\mathscr{D}^L = \{x_i^L\}_{i=1}^N$.
- Compress the LF data with an convolutional auto-encoder $h(\,.\,) = d \circ e(\,.\,)$ s.t. $e : x^L \to z^L$ and $d : z^L \to \hat{x}^L$, with $x^L \in R^D,\ z^L \in R^d$ and $d \ll D$.



Step 2:

- Clustering over the latent space: $\mathscr{C}_i = \{z_j^L \,|\, j \in J_i\}$, $\; i = 1,\ldots,k$ and $\cup_{i=1}^{k} J_i = \{1,\ldots,N\}$.

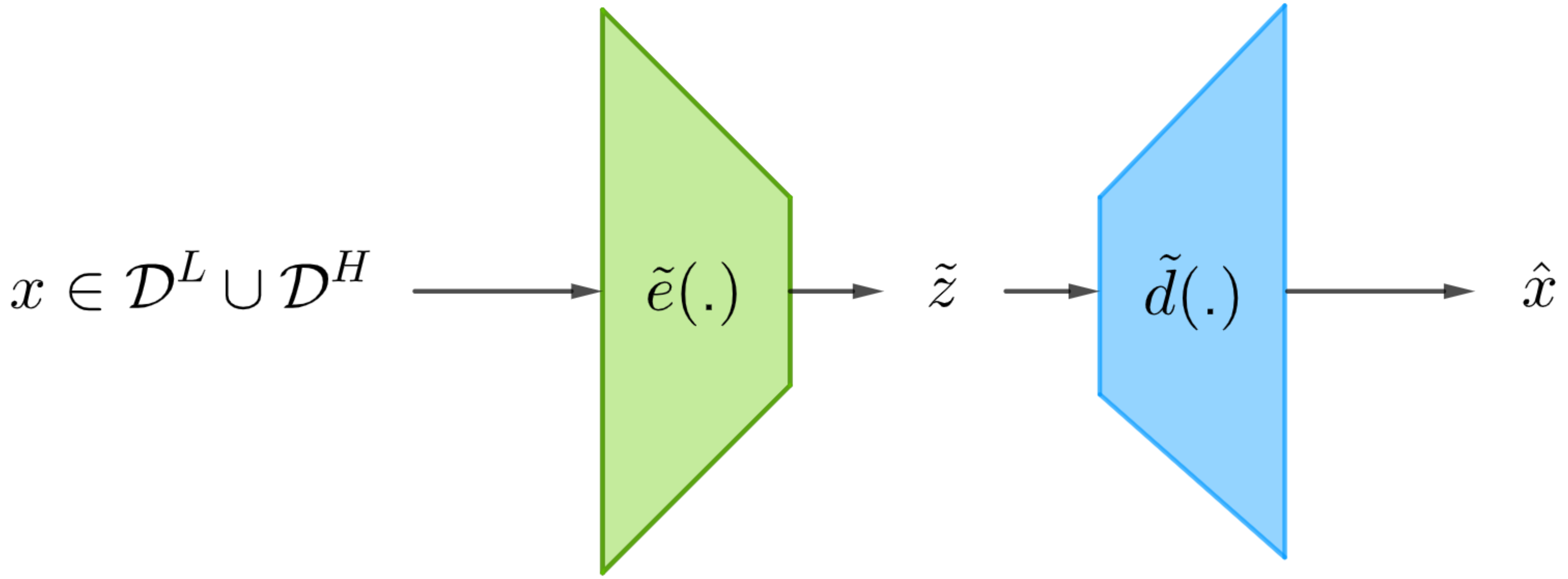


Step 3:

- Perform HF simulations $\mathscr{D}^H = \{x_i^H\}_{i=1}^k$ at parameters corresponding to centroids of clusters.
- Train h with all the data $\mathscr{D} = \mathscr{D}^L \cup \mathscr{D}^H$, obtaining $\tilde{h}(\,.\,) = \tilde{d} \circ \tilde{e}(\,.\,)$. Try to preserve the cluster structures $\mathscr{C}_i$ with a soft constraint.

$$\mathcal{D}^L = \{x_i^L\}_{i=0}^N$$

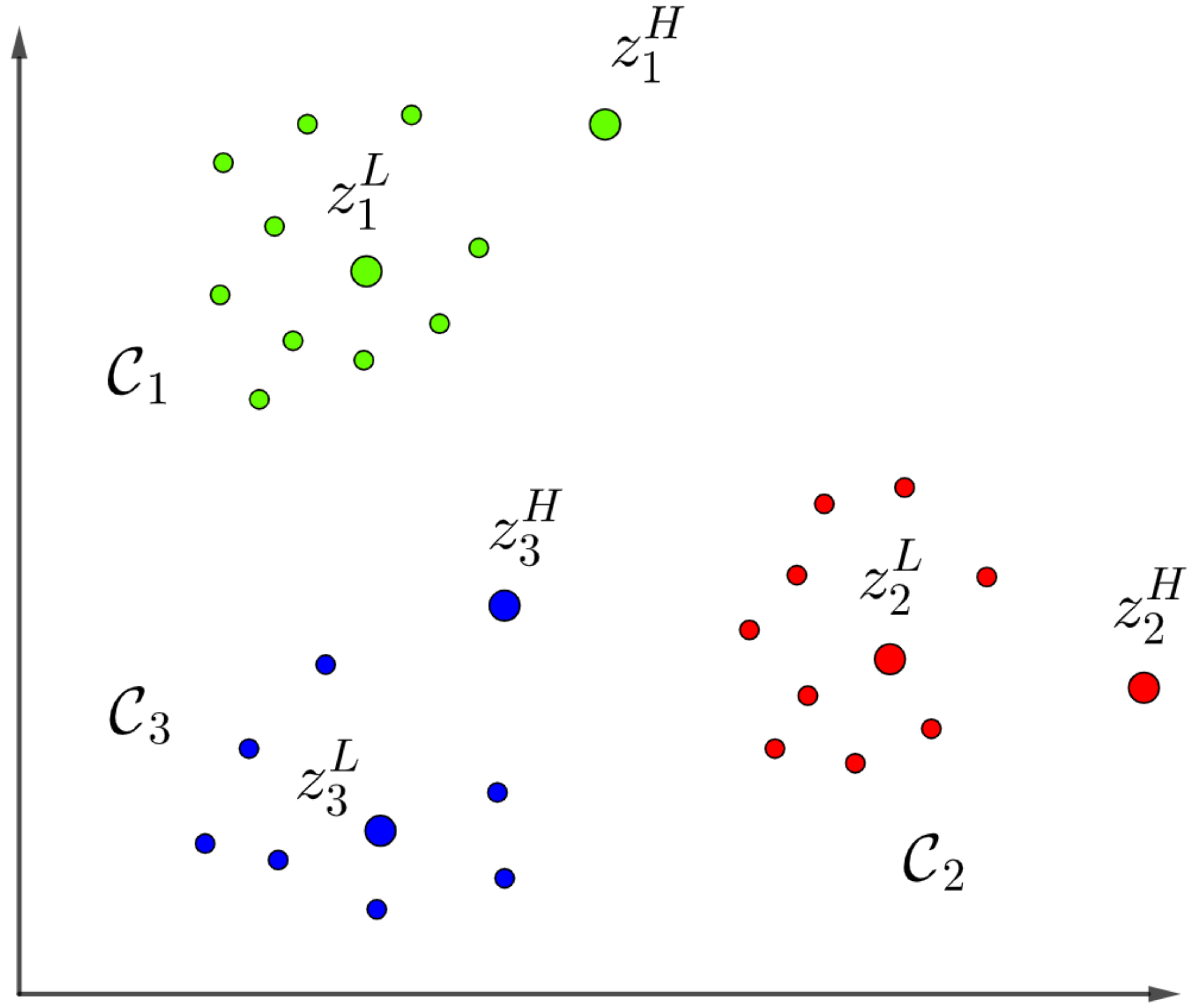
$$\mathcal{D}^H = \{x_i^H\}_{i=0}^k$$



$$\mathcal{L} = \frac{1}{N+k} \sum_{x \in \mathcal{D}^L \cup \mathcal{D}^H} |\hat{x} - x|^2 + \frac{\lambda^L}{N} \sum_{x \in \mathcal{D}^L} |\tilde{z} - z^L|^2$$

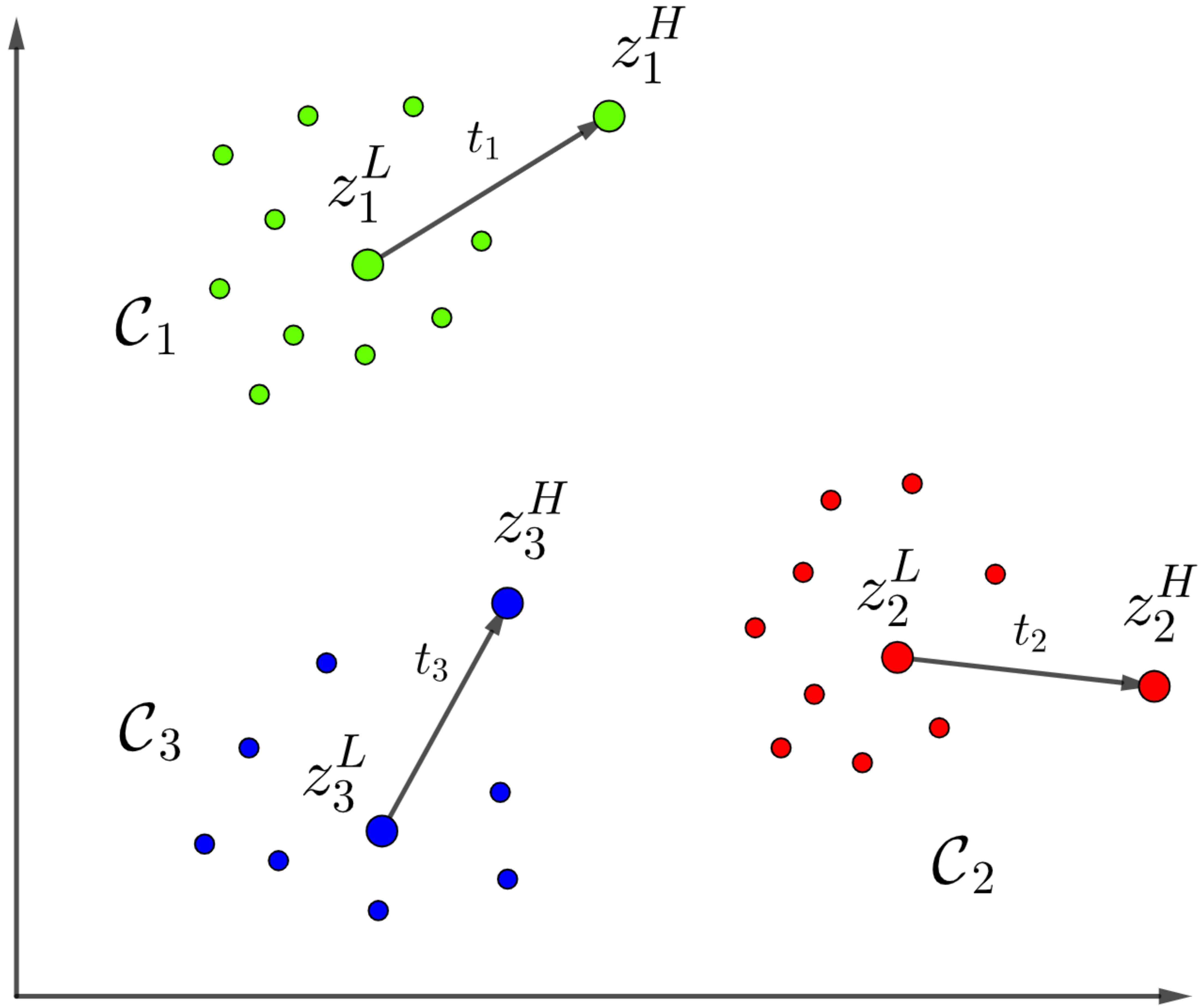
Step 4:

- Compress HF data with encoder $\tilde{e} : x^H \rightarrow z^H$



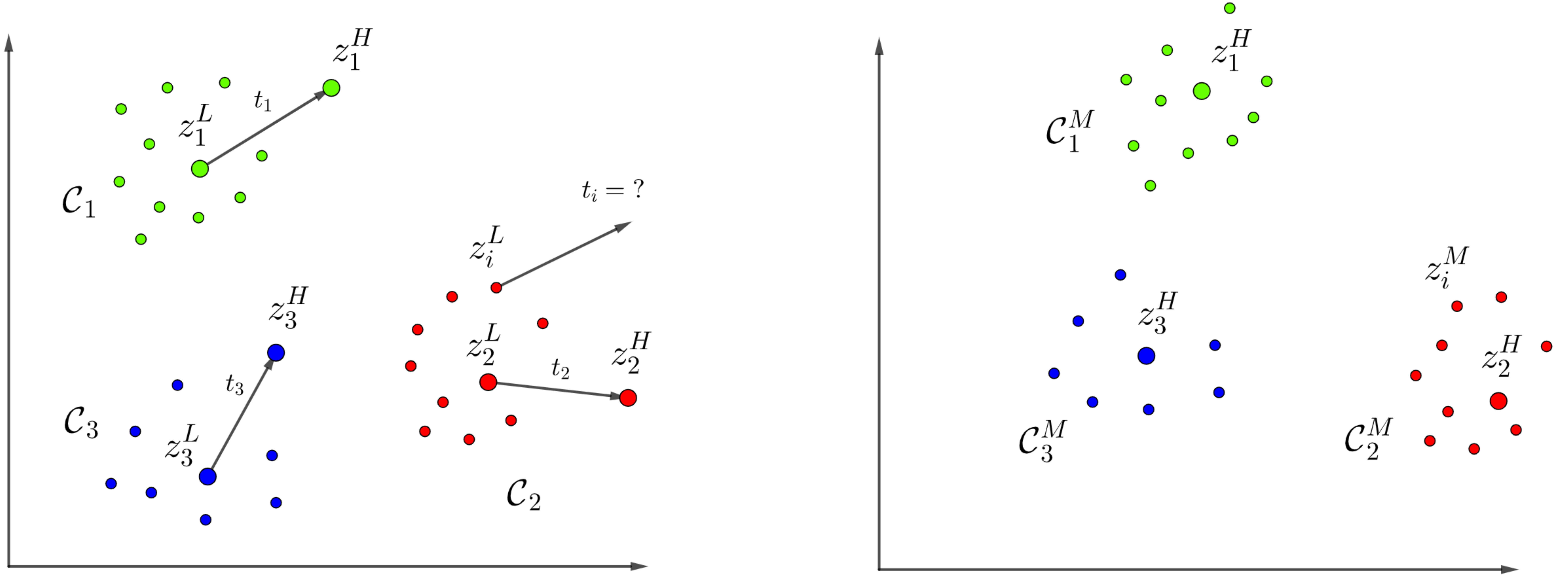
Step 4:

- Compress HF data with encoder $\tilde{e} : x^H \rightarrow z^H$



Step 5:

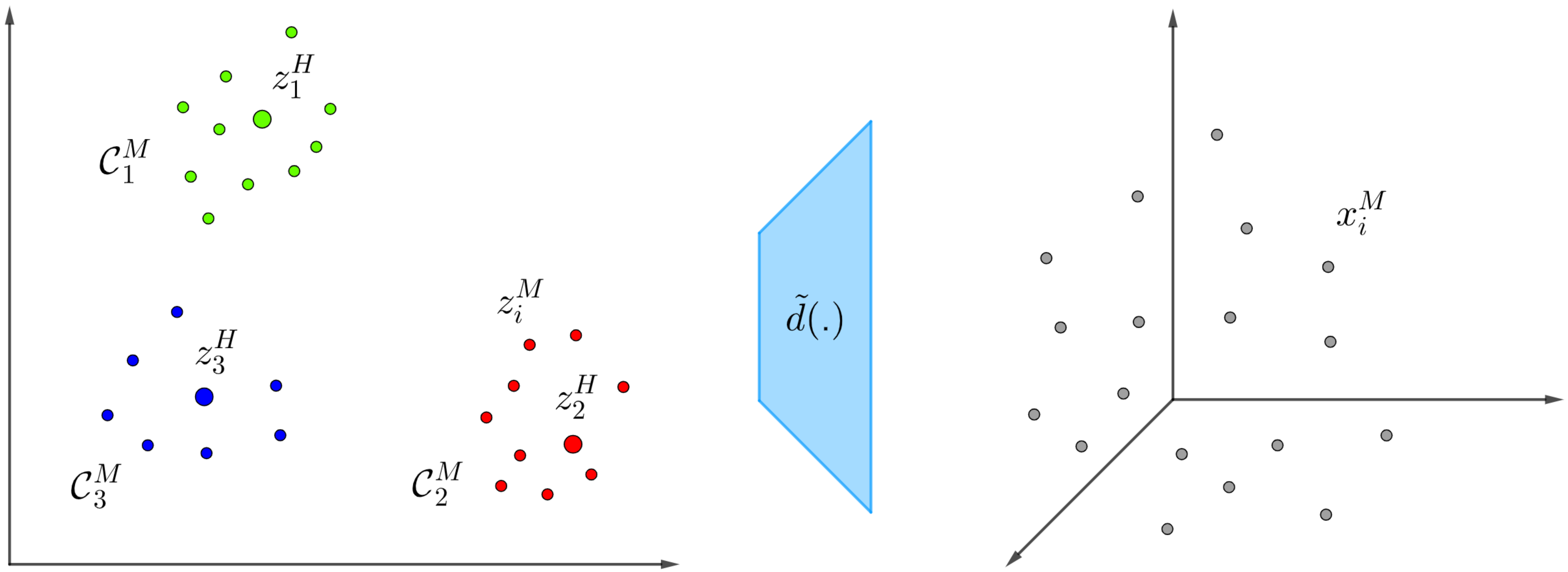
- Transform every embedding z^L to a MF approximation with $t_i : z_i^L \rightarrow z_i^M$



$$t_i = \sum_{j=1}^{k} \alpha_{ij} t_j, \quad \alpha_{ij} = \frac{\gamma_i}{|\, z_i^L - z_j^L \,|^{\beta}}, \quad \sum_{j=1}^{k} \alpha_{ij} = 1$$

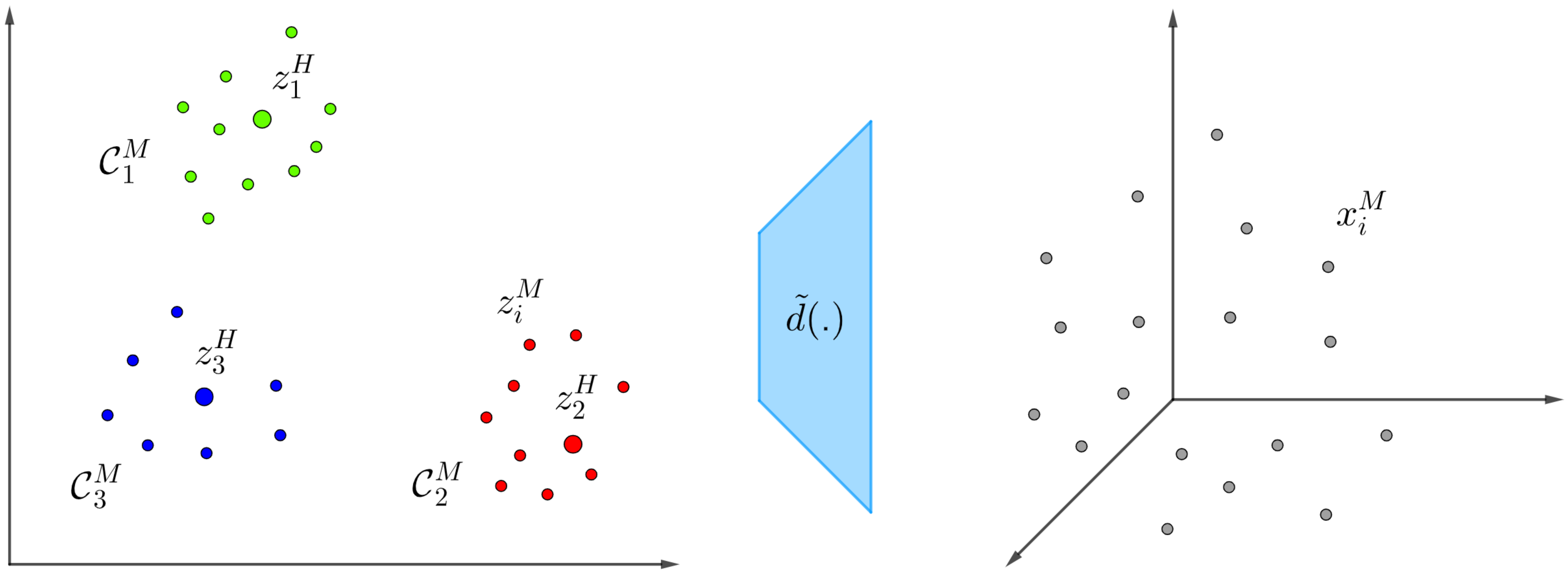
Step 6:

- Decode the embedding $\tilde{d} : z^M \rightarrow x^M$



Problem: The decoder $\tilde{d}(\ .\ )$ needs to process data from regions of the latent space never seen before.

Possible solution: Tune the decoder with self-supervised learning techniques.



Self supervised Learning:

Motivation: Supervised learning requires lots of labeled data, which are expensive to create.

Self-supervised learning aims to get the most out of unlabeled data by training the model with a *pretext task*, and then fine-tune the model with few labeled data for the supervised downstream task.

1. Define a auxiliary pre-task for the model that do not require labels. In other words, derive a supervised-type-of task from unlabeled data.
2. The idea is that while solving the auxiliary task, the model learns useful information about the underlying structure of the data, thus improving its performance for the actual downstream task.

   Examples:

   a. Infer the relative position of patches for classification;

   b. Rotate the image and train the model to predict the angle;

   c. Validate frames order for video-based applications.
3. Among the many self-supervised learning methods that have been explored, *contrastive learning* has proved to be very promising.

Contrastive Learning (aka energy based methods)

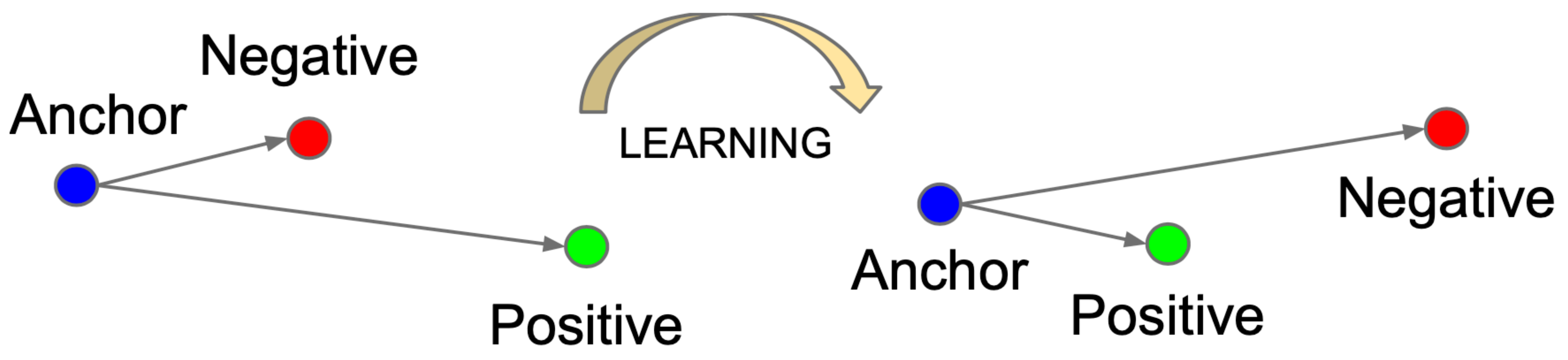
Learn by contrast. The goal is an embedding in which similar input stay close, and dissimilar ones are far apart.

Contrastive Loss: Samples $\{x_i\}$ with label $y_i \in \{1, \ldots, L\}$. This loss takes a pair of inputs and minimizes the embedding distance when they are from the same class, and maximizes the distance otherwise:

$$\mathscr{L}_{cont}(x_i, x_j, \theta) = \mathbf{1}[y_i = y_j]\, | f_\theta(x_i) - f_\theta(x_j) |_2^2 + \mathbf{1}[y_i \neq y_j] \max(0, \epsilon - | f_\theta(x_i) - f_\theta(x_j) |_2^2)$$

Triplet Loss: One anchor input x, one positive sample x^+ and one negative x^- (e.g. x and x^+ belong to the same class and x^- is sampled from a different class). Triplet loss minimizes the distance between x and x^+, and maximize the distance between x and x^-:

$$\mathscr{L}_{trip} = \sum_{x \in \mathscr{X}} \max(0, | f_\theta(x) - f_\theta(x^+) |_2^2 - | f_\theta(x) - f_\theta(x^-) |_2^2 + \epsilon)$$

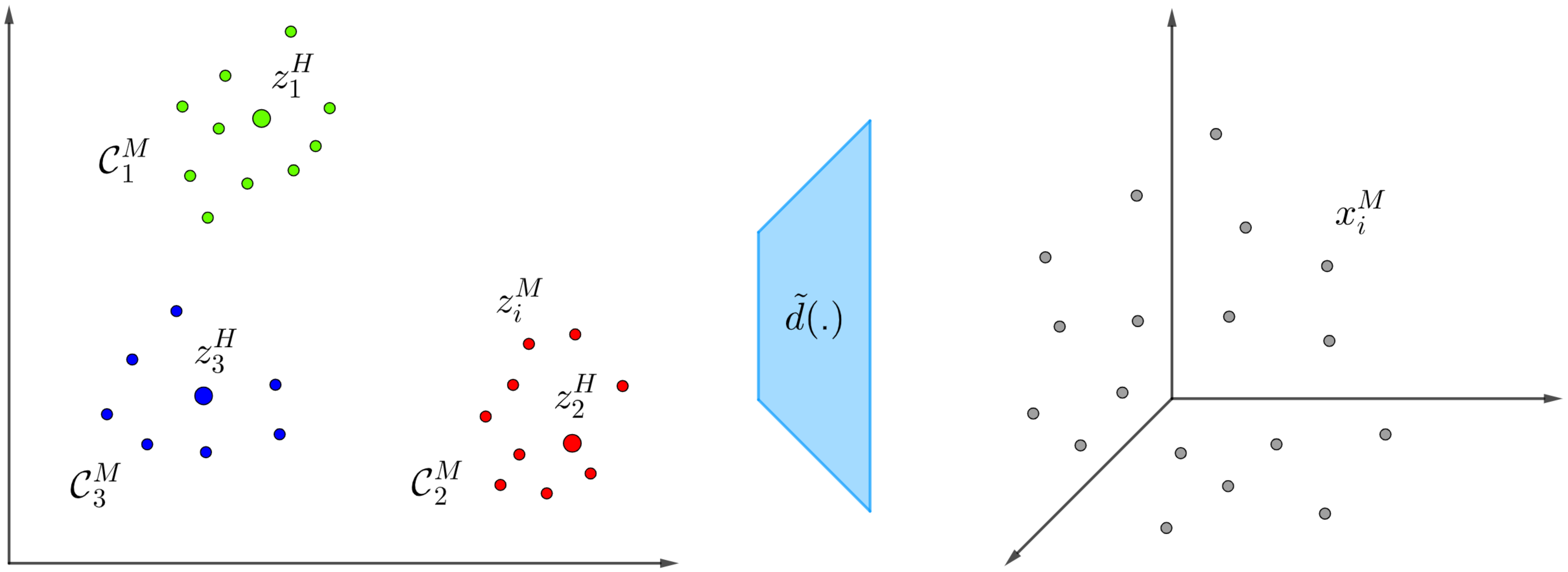


Problem: The decoder $\tilde{d}(\,.\,)$ needs to process data from regions of the latent space never seen before.

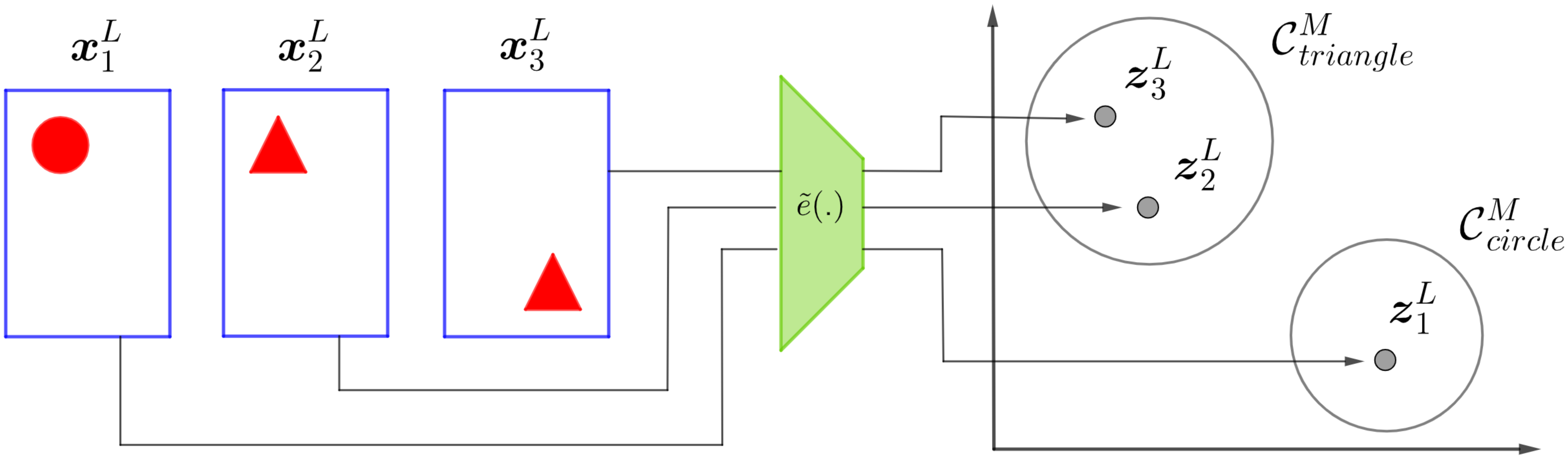
Possible solution: Tune the decoder with self-supervised learning by minimizing $\mathscr{L}_{SL}$ and $\mathscr{L}_{SSL}$:

$$\mathscr{L}_{SL} = \frac{1}{k}\sum_{i=1}^{k} |\, x_i^H - \tilde{d}(z_i^M)\,|^2$$

$$\mathscr{L}_{SSL} = \sum_{i} \max(0, |\,\tilde{d}(z_i^M) - \tilde{d}(z_j^M)\,|_2^2 - |\,\tilde{d}(z_i^M) - \tilde{d}(z_k^M)\,|^2 + \epsilon) \text{ where } z_i^M, z_j^M \in \mathscr{C}_a^M \text{ and } z_k^M \in \mathscr{C}_b^M.$$



New problem: the l_2 proximity in the latent space does not imply the l_2 proximity in the physical space



$||\boldsymbol{x}_1^L - \boldsymbol{x}_2^L||_2^2 \ll ||\boldsymbol{x}_2^L - \boldsymbol{x}_3^L||_2^2$

$||\boldsymbol{z}_1^L - \boldsymbol{z}_2^L||_2^2 \gg ||\boldsymbol{z}_2^L - \boldsymbol{z}_3^L||_2^2$