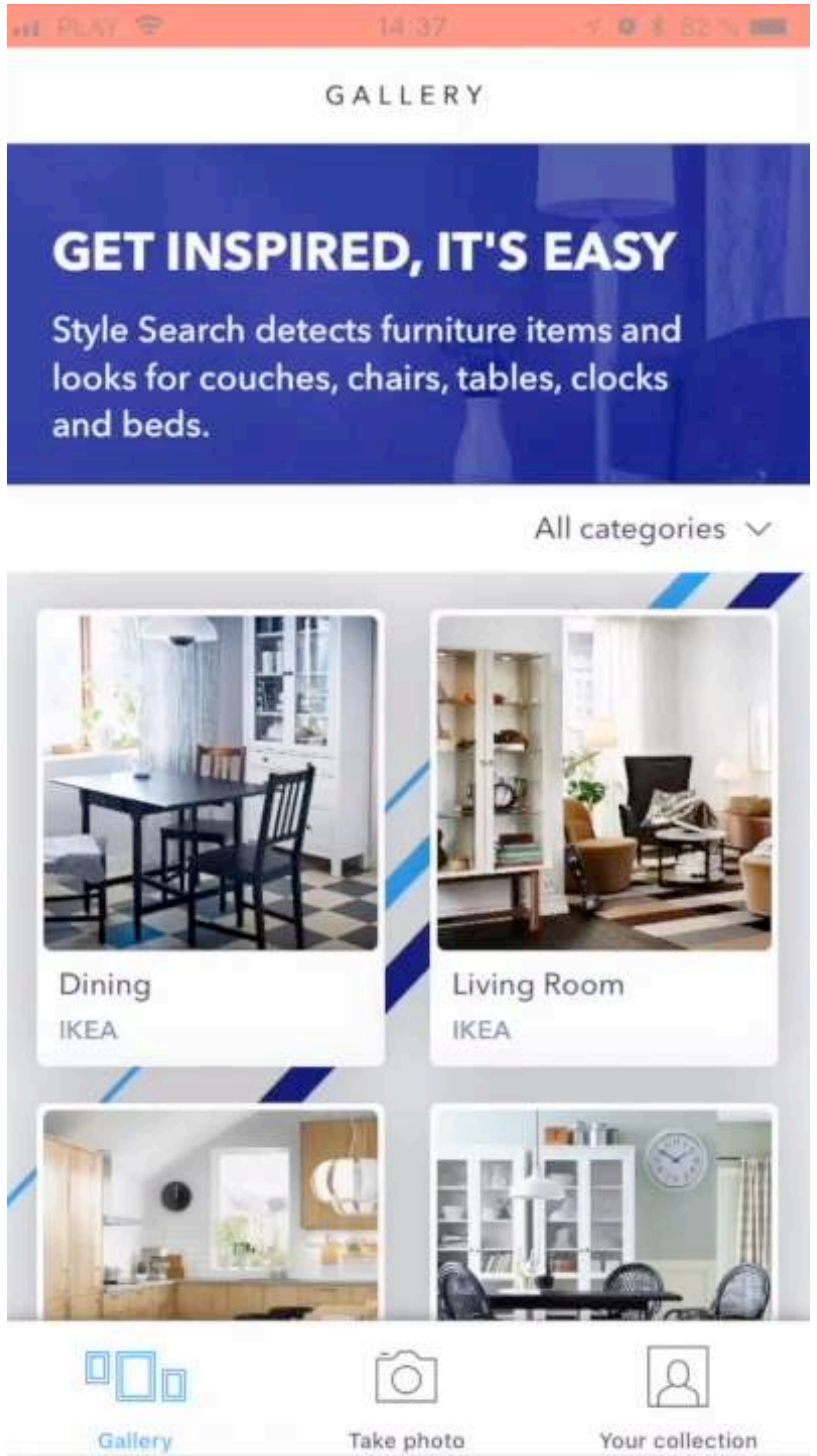
Learning binary image representations and the path towards 3D

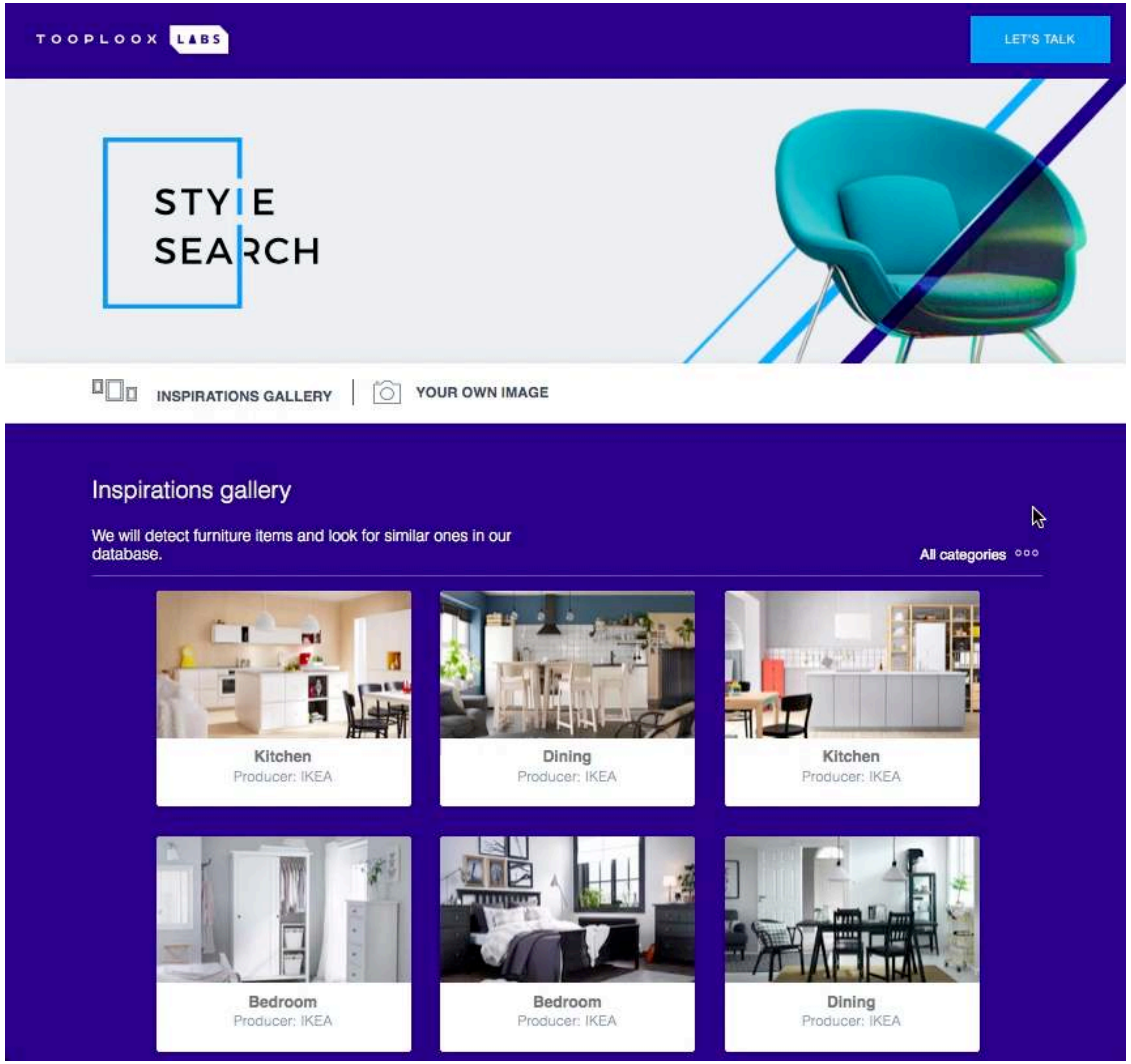
Tomasz Trzciński

Warsaw University of Technology
Tooploox

Applications of image representations

Style search





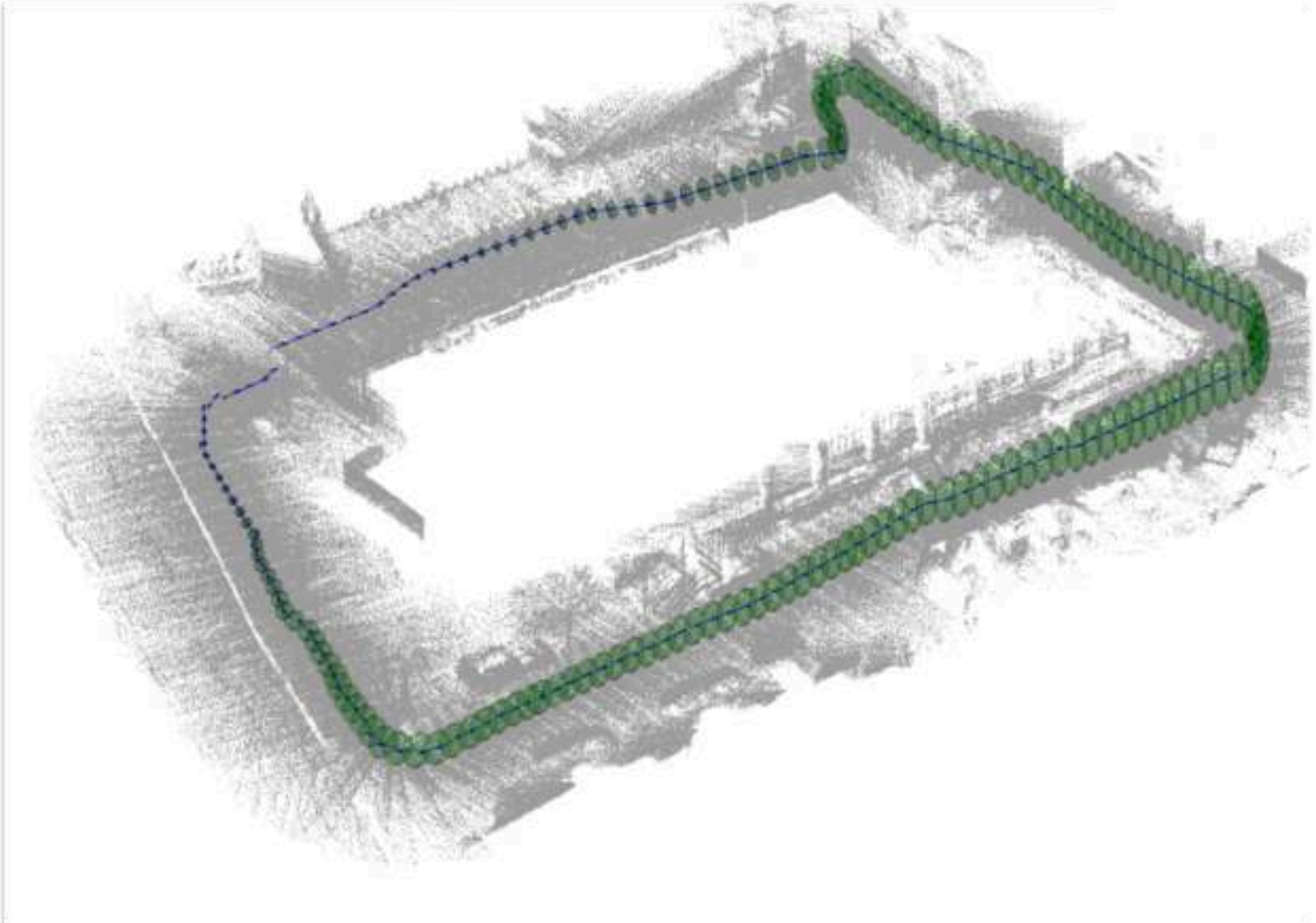
Photometry

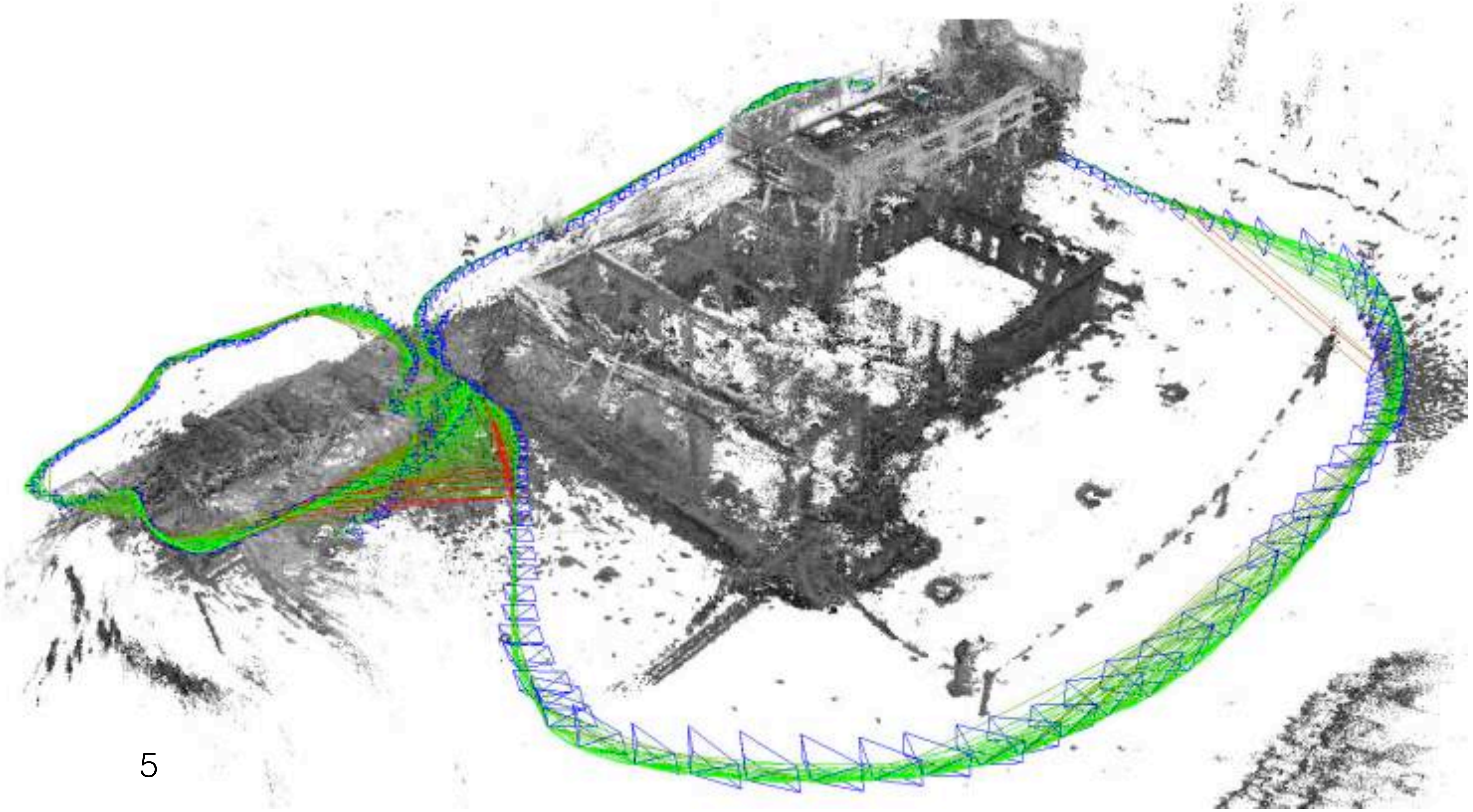
Visual SLAM

Simultaneous **Localization** and **Mapping**

Mapping = creating the map of the environment

Localization = calculating the position of a camera





Re-localisation

When SLAM fails… **re-localization** based on **visual search**

Database

Query

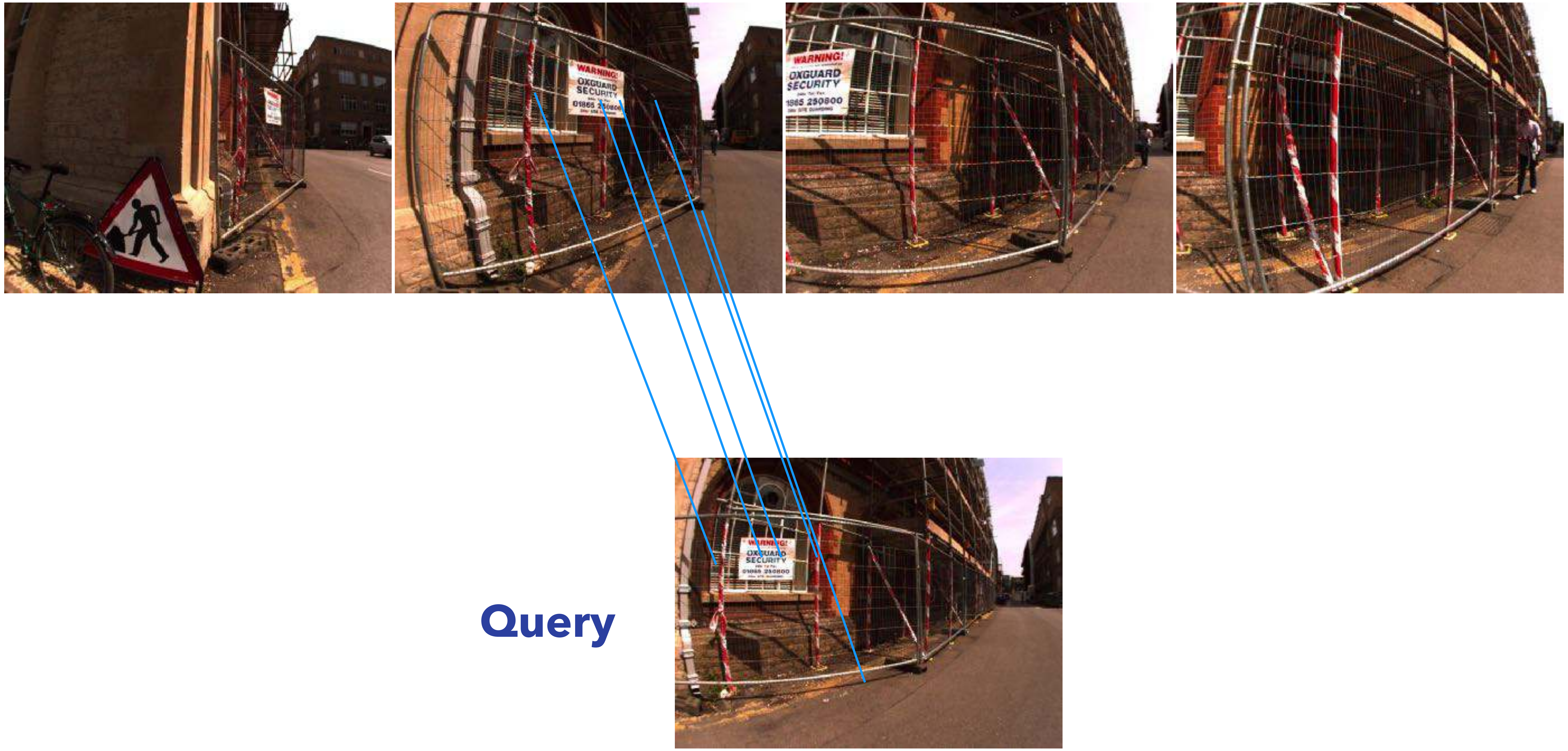


Image representations

Hand-crafted image representations

Image representation = a set of feature descriptors

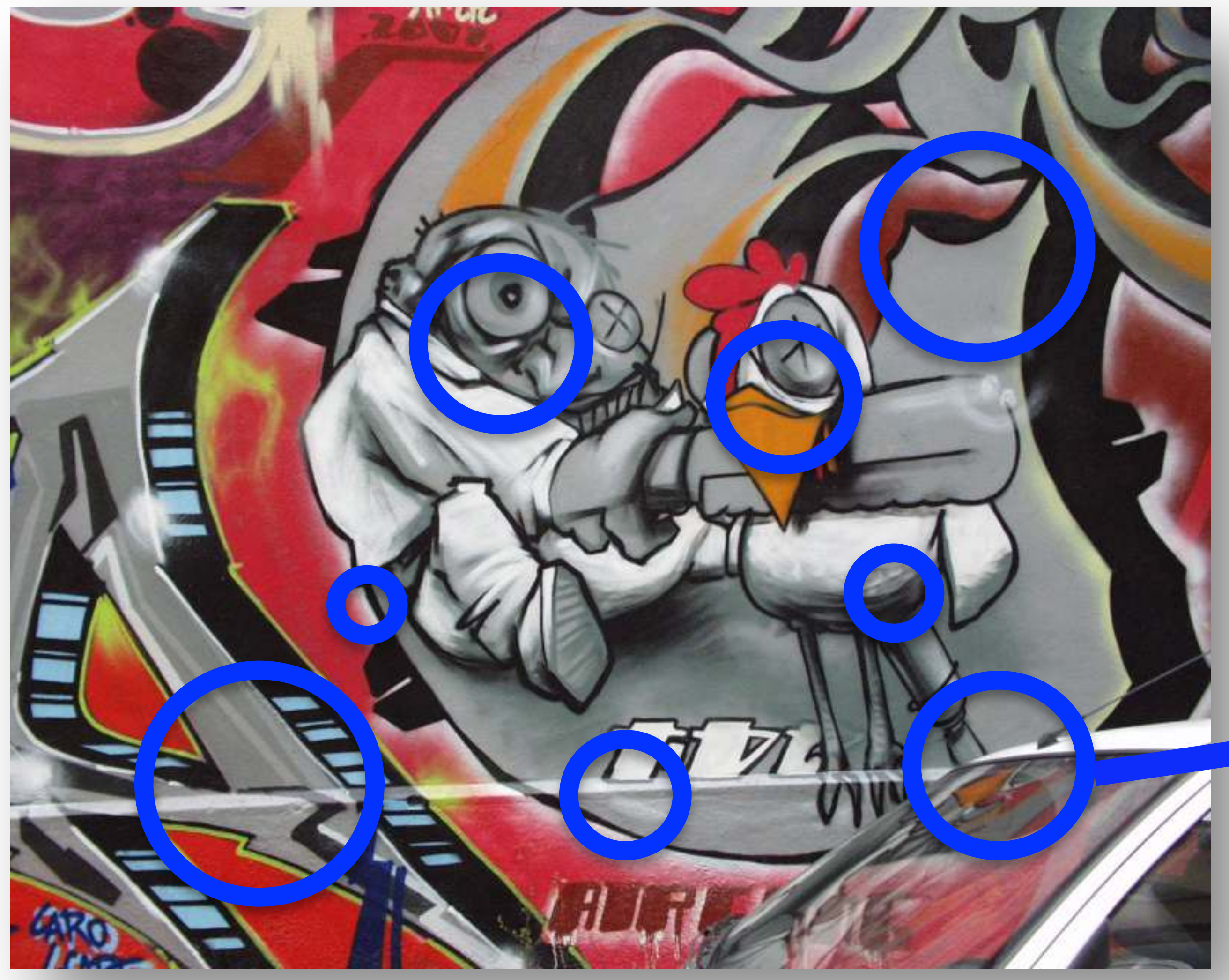
- **Multi-dimensional vector**
- Goal: **invariance to illumination and viewpoint changes**
- Main application**: finding correspondences** between images

Hand-crafted image representations

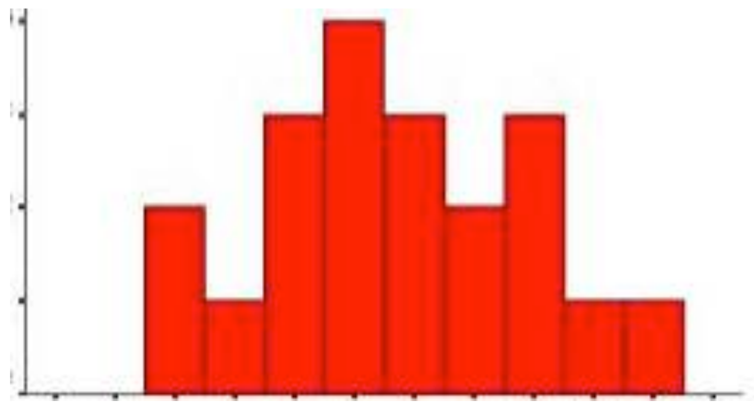
Image representation = a set of feature descriptors

- **Multi-dimensional vector**
- Goal: **invariance to illumination and viewpoint changes**
- Main application**: finding correspondences** between images

How do we compute feature descriptors?



- **Detection** of salient image regions
- **Description** based on image patch properties

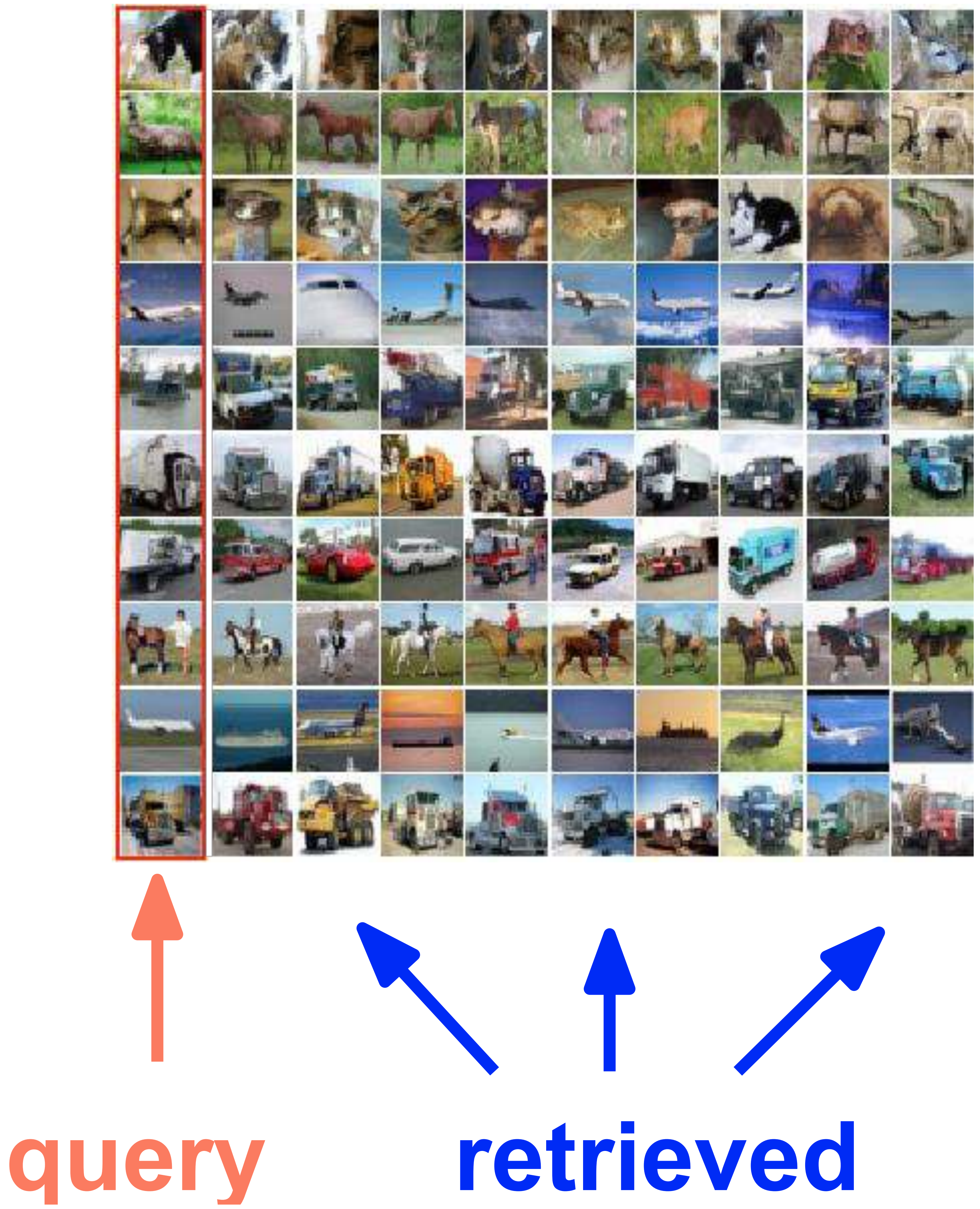


descriptor = [10.14 58.23 … 23.08]

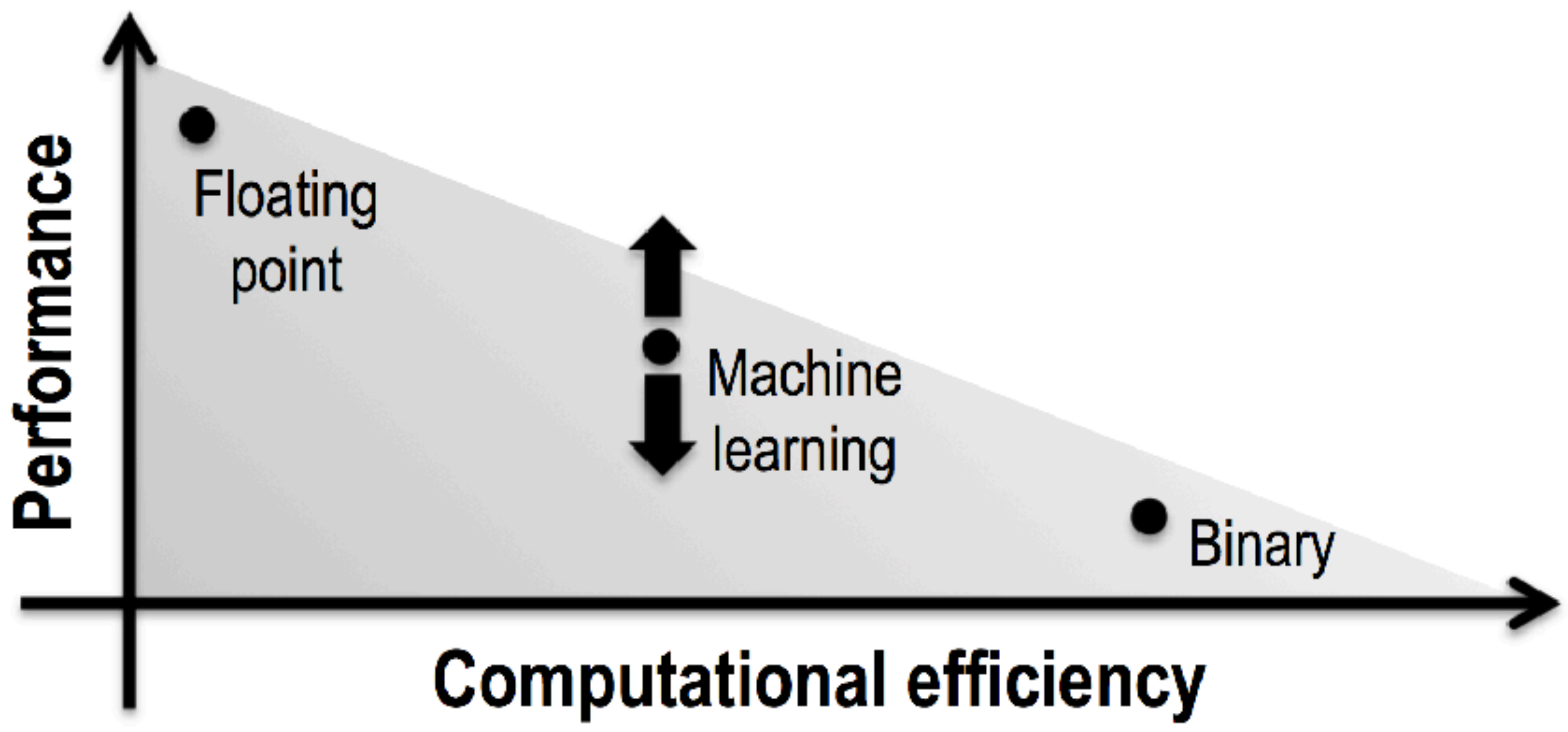
Binary image representations

Why **binary**?

- **data compression**
- **efficient image retrieval**
- **image hashing**



State of the art



Floating-point descriptors: SIFT [*Lowe, IJCV'04*], SURF [*Bay, ECCV'06*]

- **expensive computational and matching costs**

Binary descriptors: BRIEF [*Calonder, TPAMI'12*], ORB [*Rublee, ICCV'11*], FREAK [*Alahi, CVPR'12*]

- **fast, but worse performance than the floating-point competitors**

Machine-learnt descriptors: [*Simonyan, ECCV'12*], DeepDesc [*Simo-Serra, ICCV'15*], DBD-MQ [*Duan, CVPR'17*]

- **state-of-the-art results** with possibility to adjust for diverse data types (medical, natural, IR)

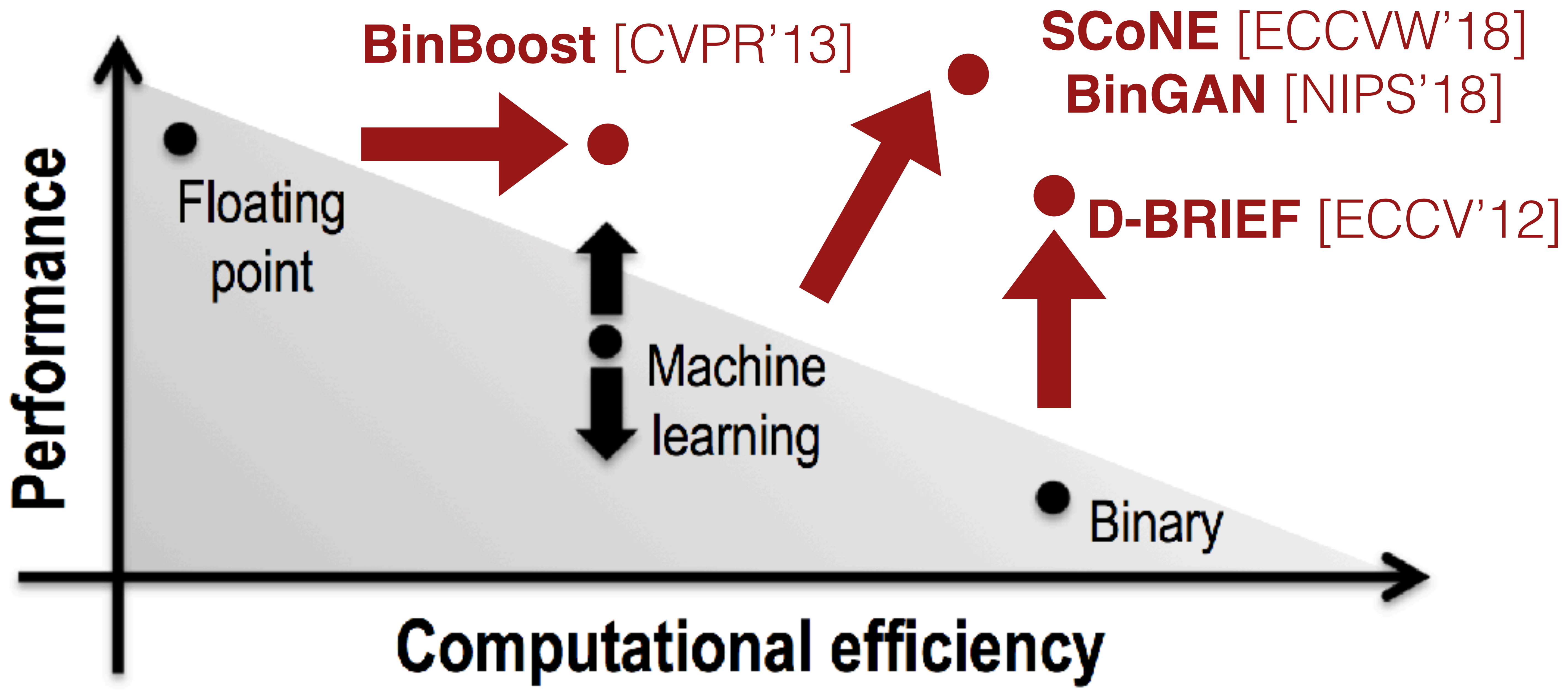
Proposed methods

Bridging the performance gap

Between the state-of-the-art floating-point and binary descriptors by **increasing the computational efficiency**

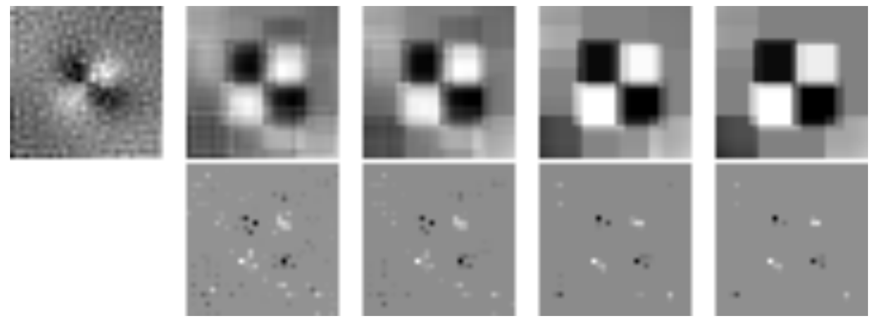
Improving the performance

Using machine learning approaches - boosting, convolutional neural networks, Siamese architecture, Generative Adversarial Networks and others

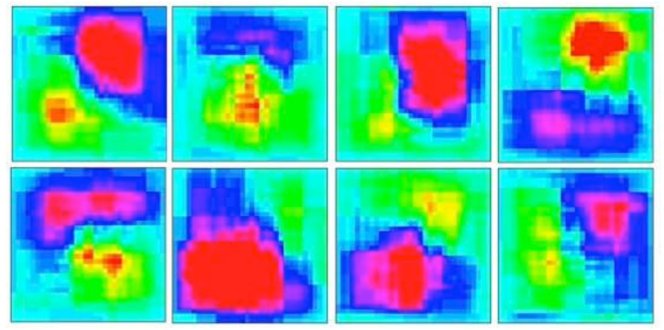


Outline

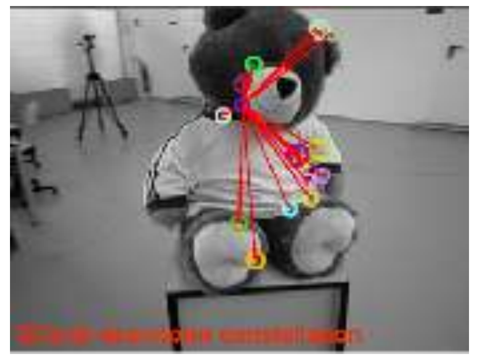
Supervised



D-Brief: Efficient Discriminative Projections for Binary Descriptors [*Trzcinski et al., ECCV'12*]

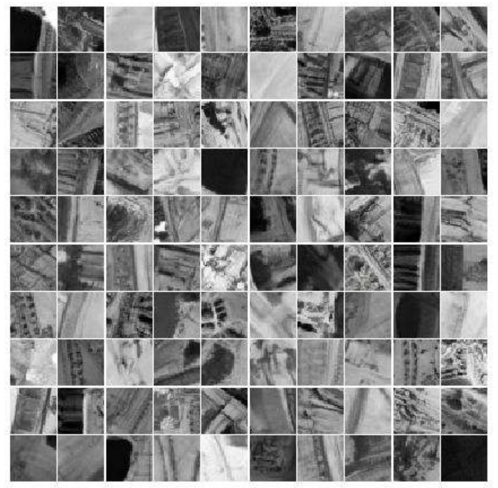


BinBoost: Boosting Binary Keypoint Descriptors [*Trzcinski et al., CVPR'13, also:* NIPS'12, TPAMI'15]



SConE: Siamese Constellation Embedding Descriptor for Image Matching [*Trzcinski et al., ECCV'18 Workshop*]

Unsupervised



BinGAN: Learning Compact Binary Descriptors with a GAN [*Zieba et al. NIPS'18*]

3dAAE: Adversarial Autoencoders for Compact Representations of 3D Point Clouds [Zamorski *et al. CoRR19*]

D-Brief

Objective

Computing binary descriptor

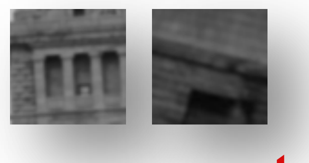
Binary dimension = **projection** applied to image patch intensities and **threshold**

$$\forall_{i \in 1,\dots,N} \quad b_i = \mathrm{sign}(\mathbf{w}_i^T \mathbf{x} + \tau_i)$$

projection image patch threshold

Training objective

$$\min_{\{\mathbf{w}_i,\tau_i\}} \sum_{i \in 1,\dots,N} \sum_{(\mathbf{x},\mathbf{x}') \in \mathcal{N}} \mathrm{sign}(\mathbf{w}_i^T \mathbf{x} + \tau_i)\, \mathrm{sign}(\mathbf{w}_i^T \mathbf{x}' + \tau_i) - \sum_{(\mathbf{x},\mathbf{x}') \in \mathcal{P}} \mathrm{sign}(\mathbf{w}_i^T \mathbf{x} + \tau_i)\, \mathrm{sign}(\mathbf{w}_i^T \mathbf{x}' + \tau_i)$$



Sum over a set of dissimilar patches (x, x')

Sum over a set of similar patches (x, x')

Computational cost

Applying general projections: **complex** and **computationally expensive**

Efficiency & Sparsity

Reducing the computational cost

Projections trained to be a **linear combination** of a **few elements** from a dictionary:

$$\forall_{i\in 1,\dots,N} \quad \mathbf{w}_i = D\mathbf{s}_i$$

Sparsity

Limited number of elements leads to an **increased efficiency.**

$$\min_{\{\mathbf{s}_i,\tau_i\}} \sum_{i\in 1,\dots,N} \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{N}} \mathrm{sign}((\boxed{D\mathbf{s}_i})^T\mathbf{x}+\tau_i)\,\mathrm{sign}((\boxed{D\mathbf{s}_i})^T\mathbf{x}'+\tau_i) - \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{P}} \mathrm{sign}((\boxed{D\mathbf{s}_i})^T\mathbf{x}+\tau_i)\,((\boxed{D\mathbf{s}_i})^T\mathbf{x}'+\tau_i) + \boxed{\lambda|\mathbf{s}_i|_1}$$

linear combination

sparsity constraint

Direct minimization is **difficult** as it involves a non-differentiable **sign function.**

Training

Related objective

Following [*Strecha, TPAMI'12*], we **drop** the sign function and minimize:

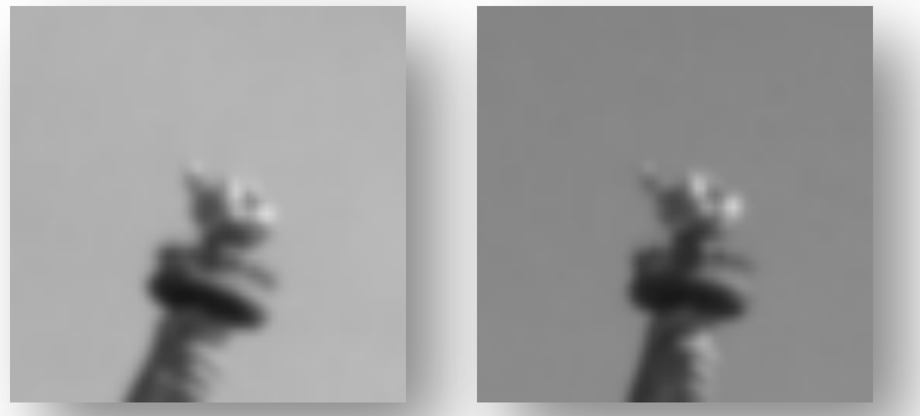
$$\min_{\{\mathbf{s}_i\}} \sum_i \frac{\sum_{(\mathbf{x},\mathbf{x'})\in\mathcal{P}}((D\mathbf{s}_i)^T(\mathbf{x}-\mathbf{x'}))^2}{\sum_{(\mathbf{x},\mathbf{x'})\in\mathcal{N}}((D\mathbf{s}_i)^T(\mathbf{x}-\mathbf{x'}))^2} + \lambda|\mathbf{s}_i|_1$$

The optimal thresholds found through a **one-dimensional** search.

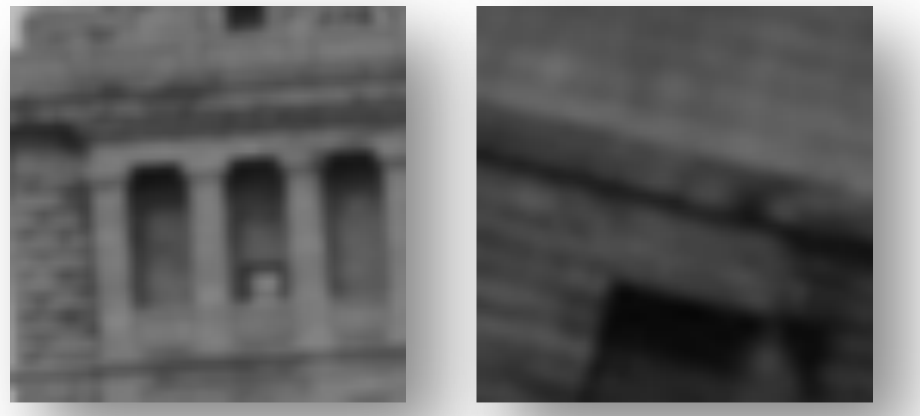
Datasets

Training datasets

Liberty, Notre Dame, Yosemite [*Brown, PAMI'12*] datasets consist of pairs of:

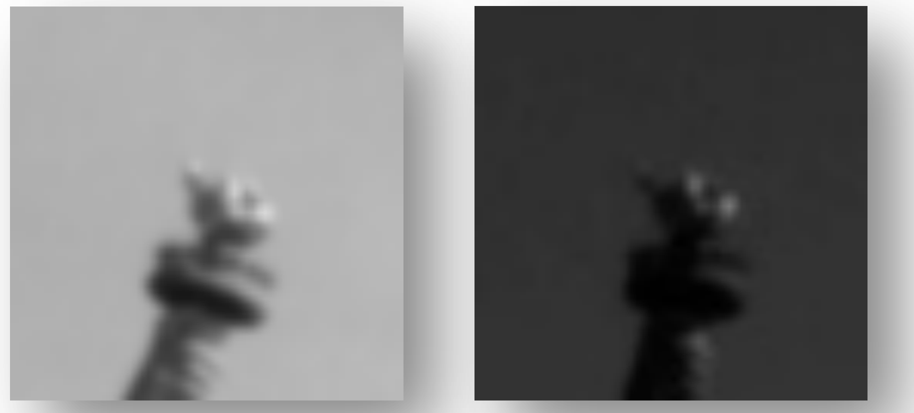


similar (positive) patches

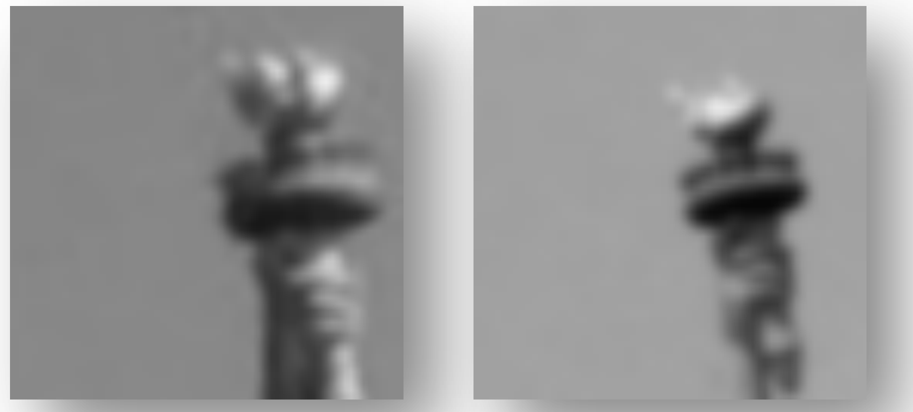


different (negative) patches

Training sets incorporate **various transformations**, e.g.:



intensity change



affine transformation

Dictionaries

Fast response elements

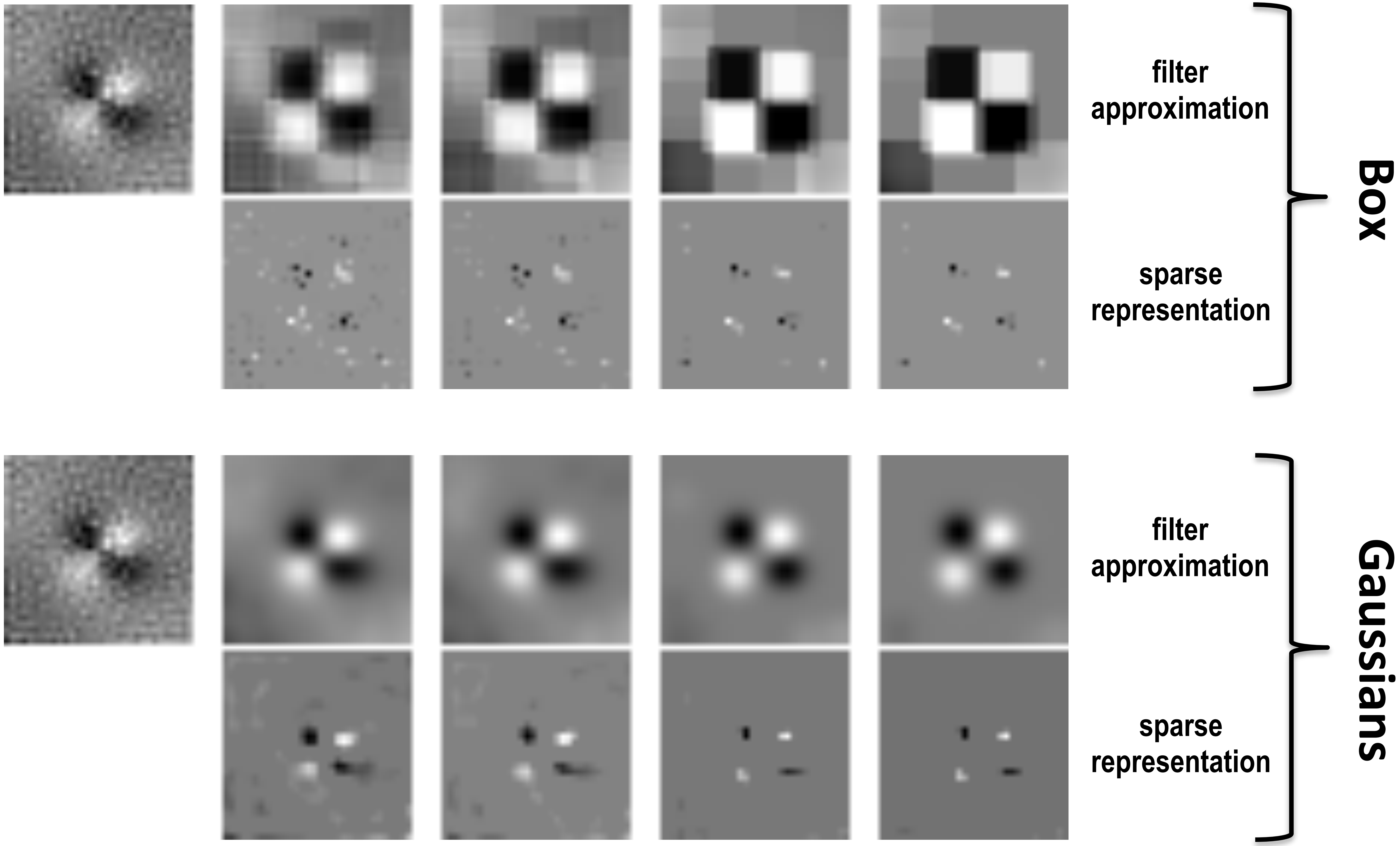
The dictionary elements are designed for the responses to be **computed fast**.

Dictionaries used

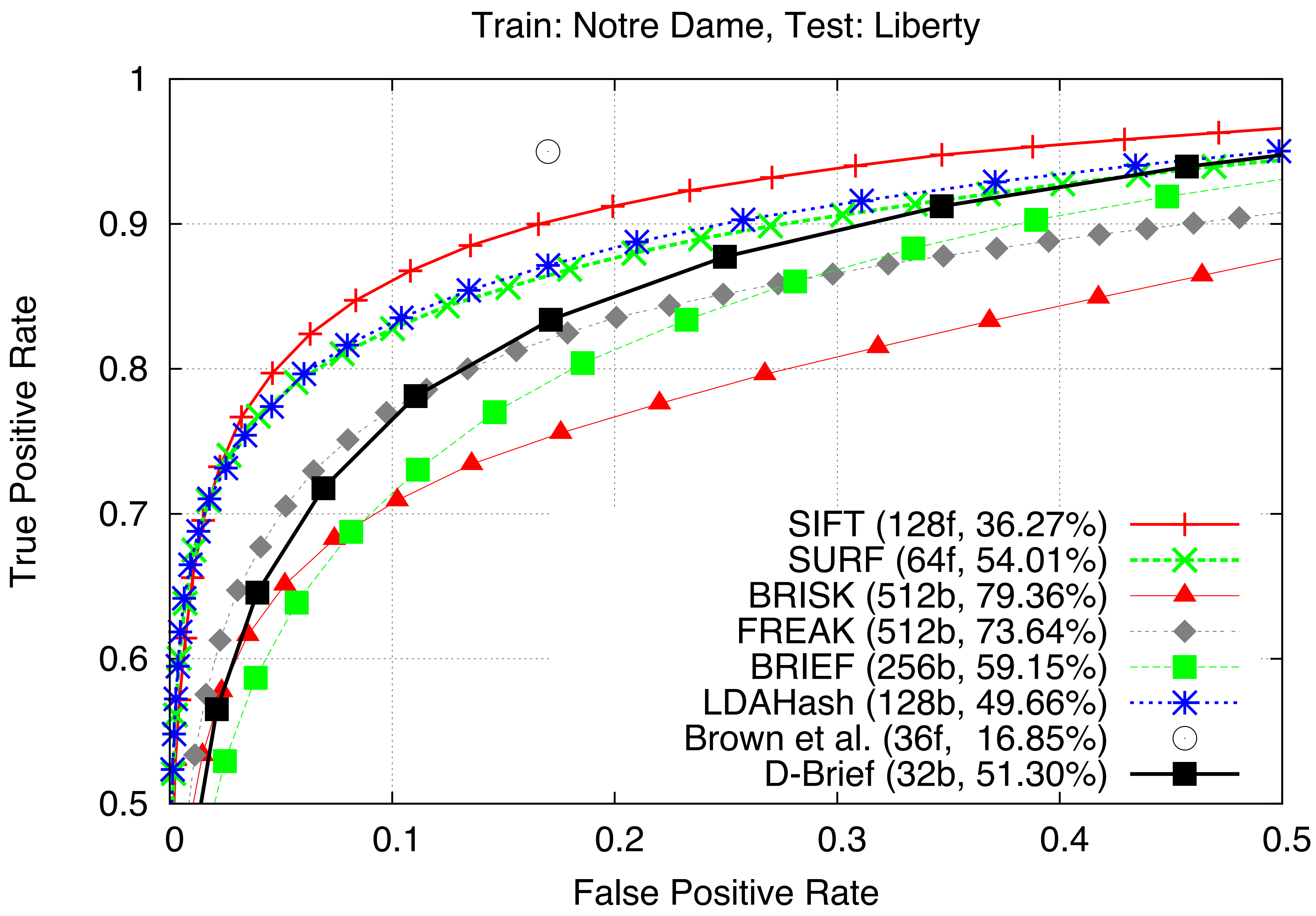
Applying projections efficiently

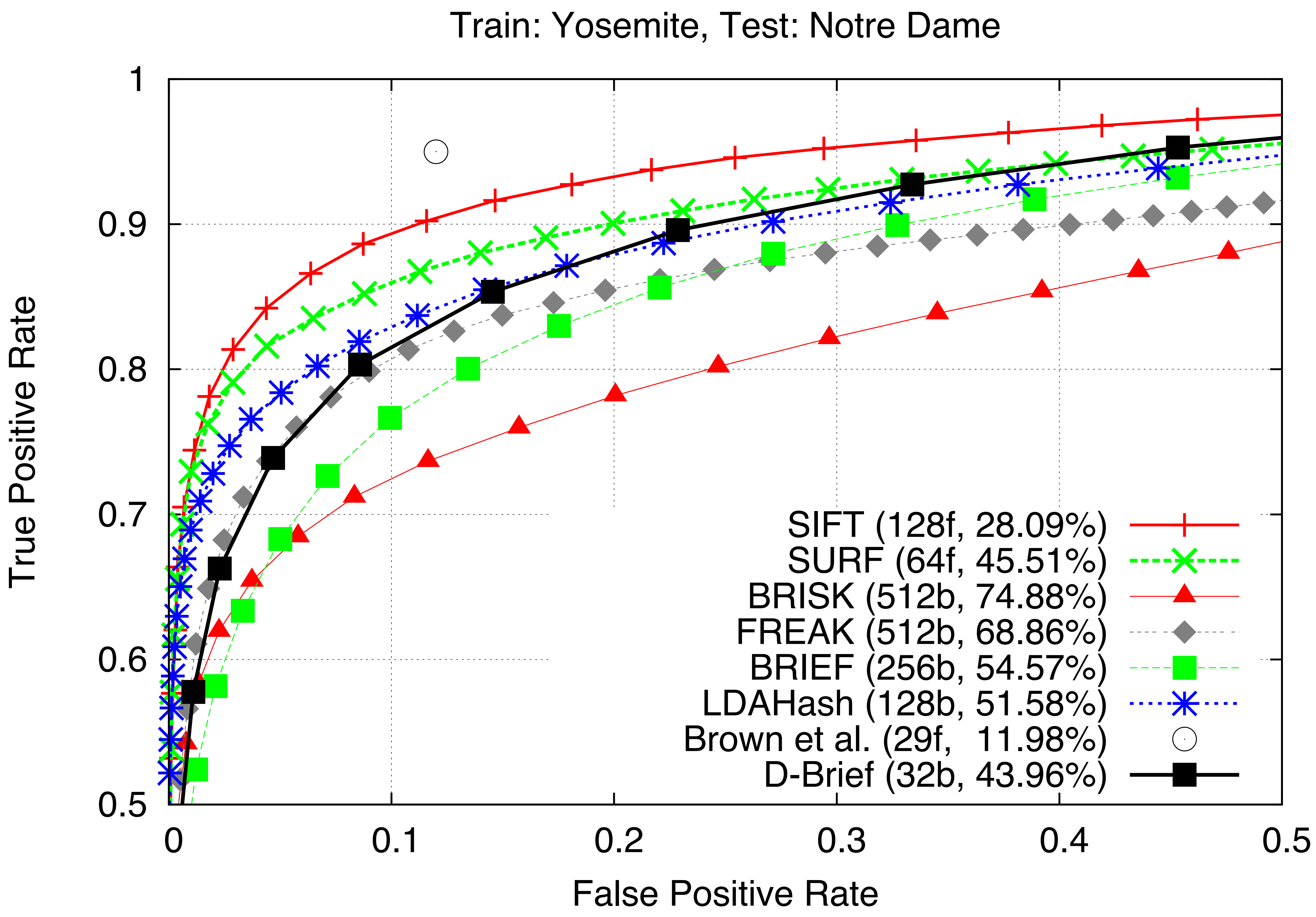
$$\mathbf{w}_i^T \mathbf{x} = (D\mathbf{s}_i)^T \mathbf{x} = \sum_{j \text{ such that } \mathbf{s}_{ij} \neq 0} \mathbf{s}_{ij} D_j^T \mathbf{x}$$

Sample approximations



Comparison with the state of the art





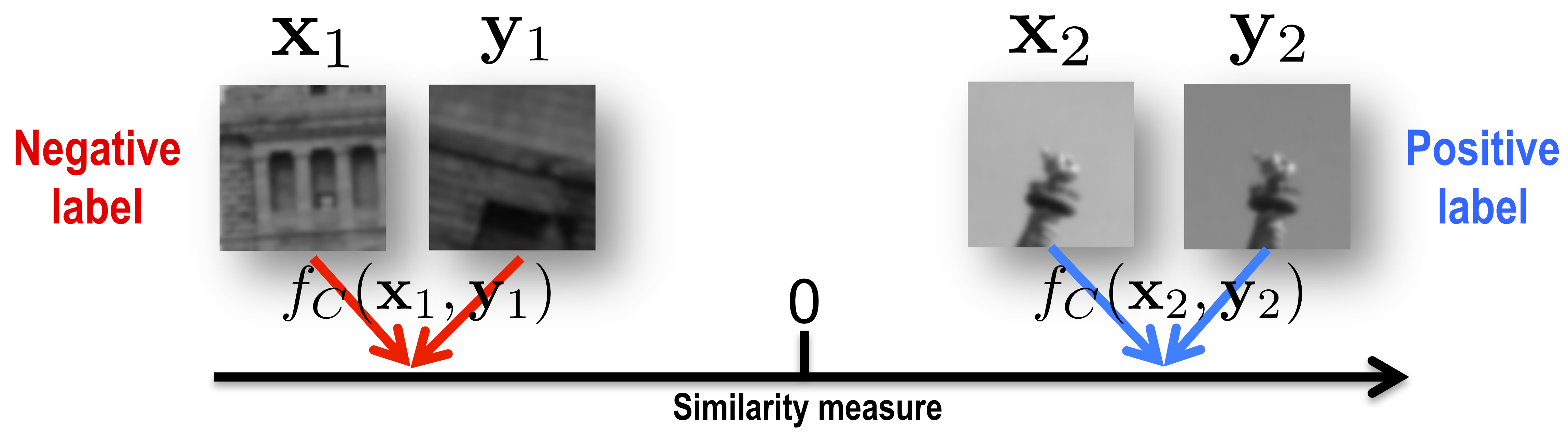
BinBoost

Learning with non-linearities

Room for improvement

D-Brief better than intensity-based descriptors, but…

Feature descriptor learning as a metric learning problem

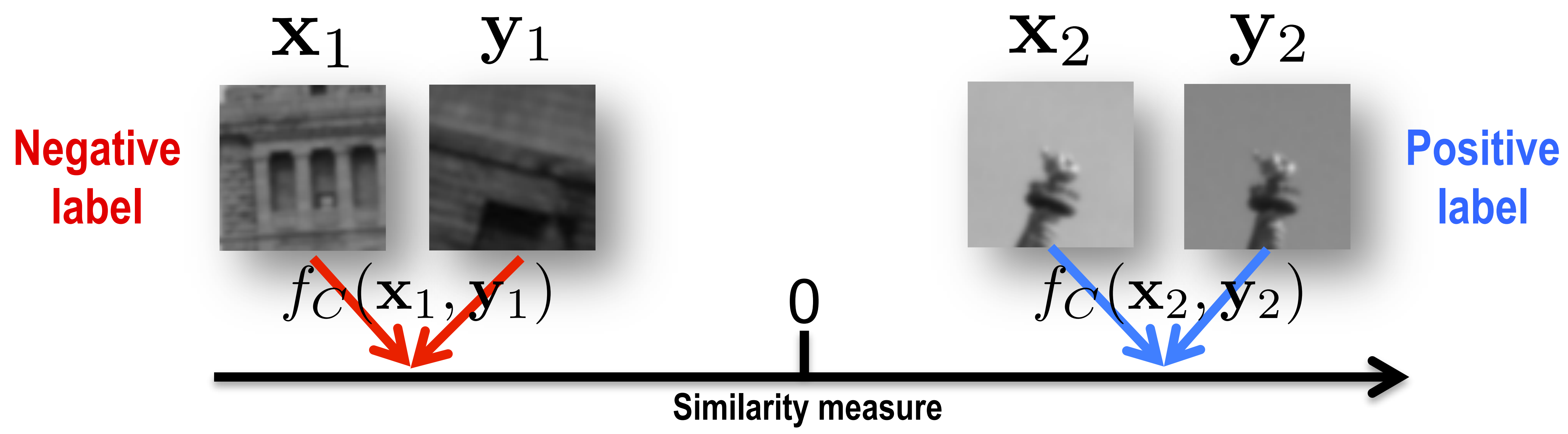


Learning with non-linearities

Room for improvement

D-Brief better than intensity-based descriptors, but…

Feature descriptor learning as a metric learning problem



Boosting

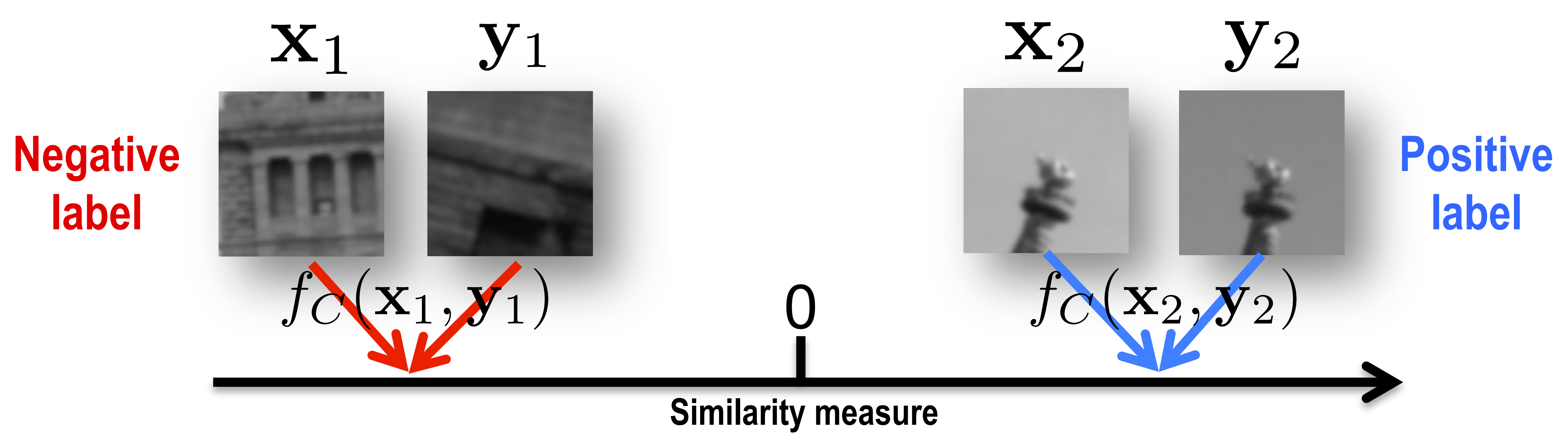
Greedy supervised learning method that trains an **ensemble of weak learners**

Learning with non-linearities

Room for improvement

D-Brief better than intensity-based descriptors, but…

Feature descriptor learning as a metric learning problem



Boosting

Greedy supervised learning method that trains an **ensemble of weak learners**

Novel framework

- Learning binary descriptors with **non-linear filters**
- **Encompasses** many **state-of-the-art descriptor** formulations

Problem formulation

Exponential loss

Patch **x** —> descriptor $C(\mathbf{x}) = [C_1(\mathbf{x}), ..., C_D(\mathbf{x})]$

Similarity function $f(C(\mathbf{x}), C(\mathbf{y})) = f_C(\mathbf{x}, \mathbf{y})$ defined over image patch pairs and labels:

$$\mathcal{L} = \sum_{i=1}^{N} \exp(-l_i f_C(\mathbf{x}_i, \mathbf{y}_i))$$

label

similarity function

Problem formulation

Exponential loss

Patch **x** —> descriptor $C(\mathbf{x}) = [C_1(\mathbf{x}), ..., C_D(\mathbf{x})]$

Similarity function $f(C(\mathbf{x}), C(\mathbf{y})) = f_C(\mathbf{x}, \mathbf{y})$ defined over image patch pairs and labels:

$$\mathcal{L} = \sum_{i=1}^{N} \exp(-\underbrace{l_i}_{\text{label}} \underbrace{f_C(\mathbf{x}_i, \mathbf{y}_i)}_{\text{similarity function}})$$

Similarity function

$$f_C(\mathbf{x}, \mathbf{y}) = C(\mathbf{x})^\top \mathbf{A} C(\mathbf{y})$$

Problem formulation

Exponential loss

Patch **x** —> descriptor $C(\mathbf{x}) = [C_1(\mathbf{x}), ..., C_D(\mathbf{x})]$

Similarity function $f(C(\mathbf{x}), C(\mathbf{y})) = f_C(\mathbf{x}, \mathbf{y})$ defined over image patch pairs and labels:

$$\mathcal{L} = \sum_{i=1}^{N} \exp(-\underbrace{l_i}_{\text{label}} \underbrace{f_C(\mathbf{x}_i, \mathbf{y}_i)}_{\text{similarity function}})$$

Similarity function

$$f_C(\mathbf{x}, \mathbf{y}) = C(\mathbf{x})^\top \mathbf{A} C(\mathbf{y})$$

Boosted Similarity Sensitive Coding (SCC) [*Shakhnarovich, MIT'06*]

Similarity function: weighted sum of thresholded learners' responses:

$$f_{SSC}(\mathbf{x}, \mathbf{y}) = \sum_{d=1}^{D} \alpha_d h_d(\mathbf{x}) h_d(\mathbf{y})$$

FPBoost

Redundancy of Boosted SCC

Mitigated by modelling also the **correlation** between weak learners $\{h_d(.)\}$

$$f_{FP}(\mathbf{x}, \mathbf{y}) = \sum_{k,k'} \alpha_{k,k'} h_k(\mathbf{x}) h_{k'}(\mathbf{y}) = \mathbf{h}(\mathbf{x})^T \boxed{\mathbf{A}} \mathbf{h}(\mathbf{y})$$

symmetric

FPBoost

Redundancy of Boosted SCC

Mitigated by modelling also the **correlation** between weak learners $\{h_d(.)\}$

$$f_{FP}(\mathbf{x},\mathbf{y}) = \sum_{k,k'} \alpha_{k,k'} h_k(\mathbf{x}) h_{k'}(\mathbf{y}) = \mathbf{h}(\mathbf{x})^T \boxed{\mathbf{A}} \mathbf{h}(\mathbf{y})$$

symmetric

$$\mathbf{A}_{\mathbf{diag}} = \begin{bmatrix} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_3 & 0 \\ 0 & 0 & 0 & \alpha_4 \end{bmatrix} \rightarrow \mathbf{A}_{\mathbf{sym}} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\ \alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\ \alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\ \alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4} \end{bmatrix} \qquad \forall_{i,j}\, \alpha_{i,j} = \alpha_{j,i}$$

FPBoost

Redundancy of Boosted SCC

Mitigated by modelling also the **correlation** between weak learners $\{h_d(.)\}$

$$f_{FP}(\mathbf{x},\mathbf{y}) = \sum_{k,k'} \alpha_{k,k'} h_k(\mathbf{x}) h_{k'}(\mathbf{y}) = \mathbf{h}(\mathbf{x})^T \boxed{\mathbf{A}} \mathbf{h}(\mathbf{y})$$

symmetric

Factorization

With **A** constrained to be **symmetric:**

$$\mathbf{A} = \mathbf{B}\mathbf{W}\mathbf{B}^T = \sum_{k=1}^{K} w_k \mathbf{b}_k \mathbf{b}_k^T$$

FPBoost

Redundancy of Boosted SCC

Mitigated by modelling also the **correlation** between weak learners $\{h_d(.)\}$

$$f_{FP}(\mathbf{x},\mathbf{y}) = \sum_{k,k'} \alpha_{k,k'} h_k(\mathbf{x}) h_{k'}(\mathbf{y}) = \mathbf{h}(\mathbf{x})^T \mathbf{A} \mathbf{h}(\mathbf{y})$$

symmetric

Factorization

With **A** constrained to be **symmetric:**

$$\mathbf{A} = \mathbf{B}\mathbf{W}\mathbf{B}^T = \sum_{k=1}^{K} w_k \mathbf{b}_k \mathbf{b}_k^T$$

FPBoost descriptor:

$$C(\mathbf{x}) = \mathbf{B}^T\mathbf{h}(\mathbf{x}) = [\sum_{k=1}^{K} b_{1,k} h_k(\mathbf{x}), \ldots, \sum_{k=1}^{K} b_{D,k} h_k(\mathbf{x})]$$

FPBoost

Redundancy of Boosted SCC

Mitigated by modelling also the **correlation** between weak learners $\{h_d(.)\}$

$$f_{FP}(\mathbf{x},\mathbf{y}) = \sum_{k,k'} \alpha_{k,k'} h_k(\mathbf{x}) h_{k'}(\mathbf{y}) = \mathbf{h}(\mathbf{x})^T \boxed{\mathbf{A}}\mathbf{h}(\mathbf{y})$$

symmetric

Factorization

With **A** constrained to be **symmetric:**

$$\mathbf{A} = \mathbf{B}\mathbf{W}\mathbf{B}^T = \sum_{k=1}^{K} w_k \mathbf{b}_k \mathbf{b}_k^T$$

FPBoost descriptor:

$$C(\mathbf{x}) = \mathbf{B}^T\mathbf{h}(\mathbf{x}) = [\sum_{k=1}^{K} b_{1,k} h_k(\mathbf{x}), \ldots, \sum_{k=1}^{K} b_{D,k} h_k(\mathbf{x})]$$

BinBoost descriptor:

$$f_B(\mathbf{x},\mathbf{y}) = \sum_{d=1}^{D} \boxed{\text{sign}}\left(\mathbf{b}_d^T \mathbf{h}_d(\mathbf{x})\right) \boxed{\text{sign}}\left(\mathbf{b}_d^T \mathbf{h}_d(\mathbf{y})\right) = \sum_{d=1}^{D} C_d(\mathbf{x}) C_d(\mathbf{y})$$

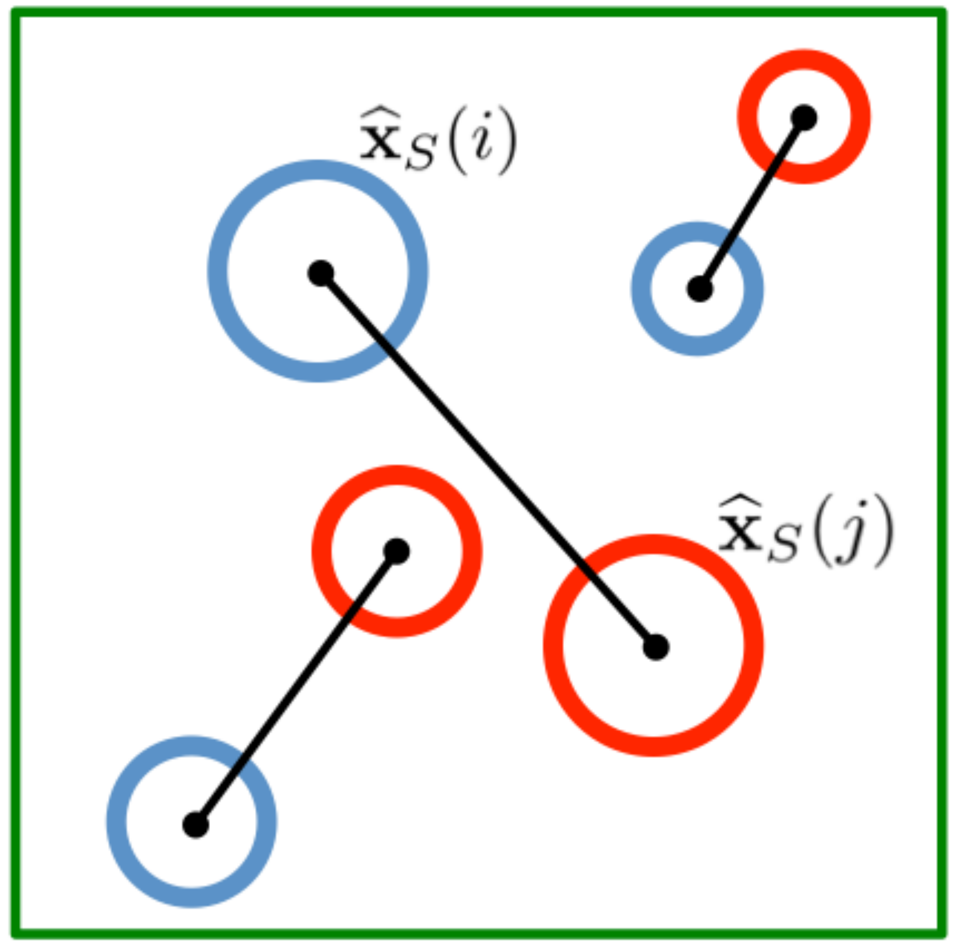
thresholded

Weak learners

Intensity-based learners

Inspired by BRIEF, BRISK or FREAK

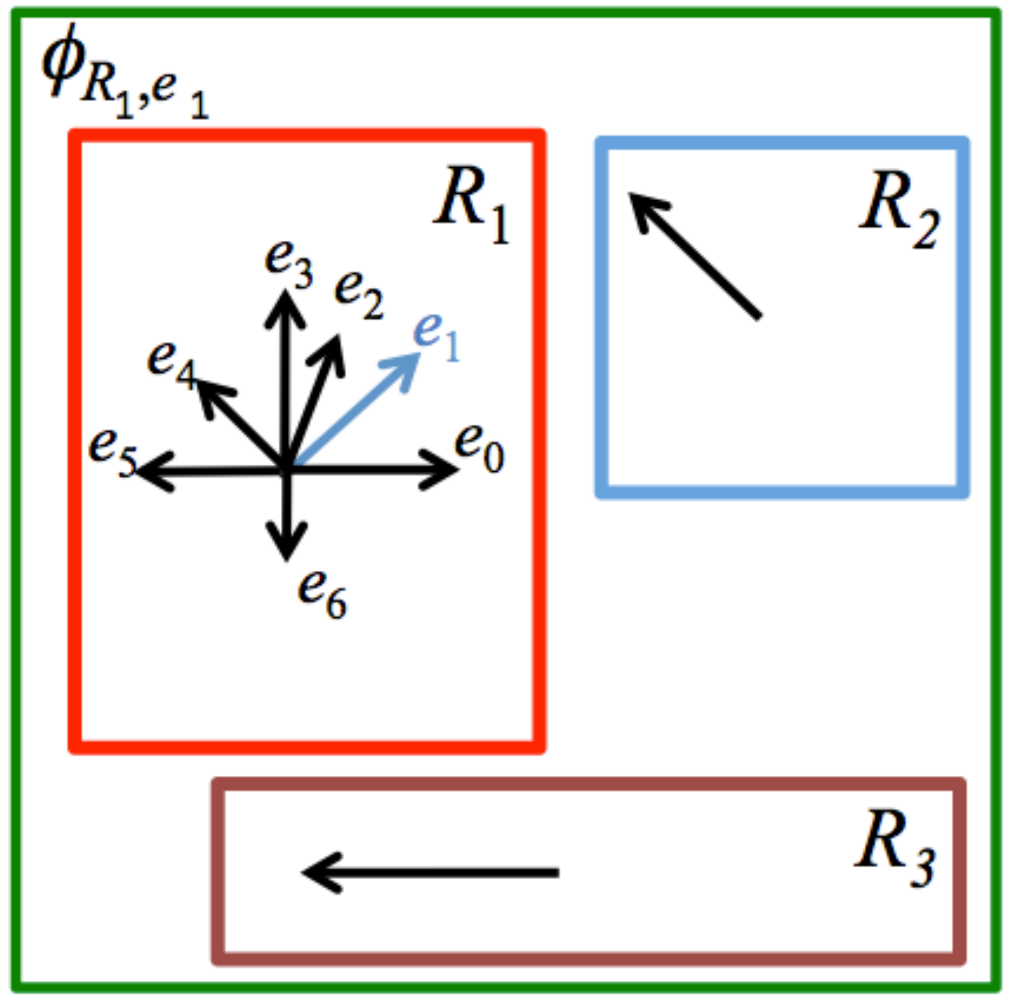
$$\text{response} = \widehat{\mathbf{x}}_S(i) > \widehat{\mathbf{x}}_S(j)$$



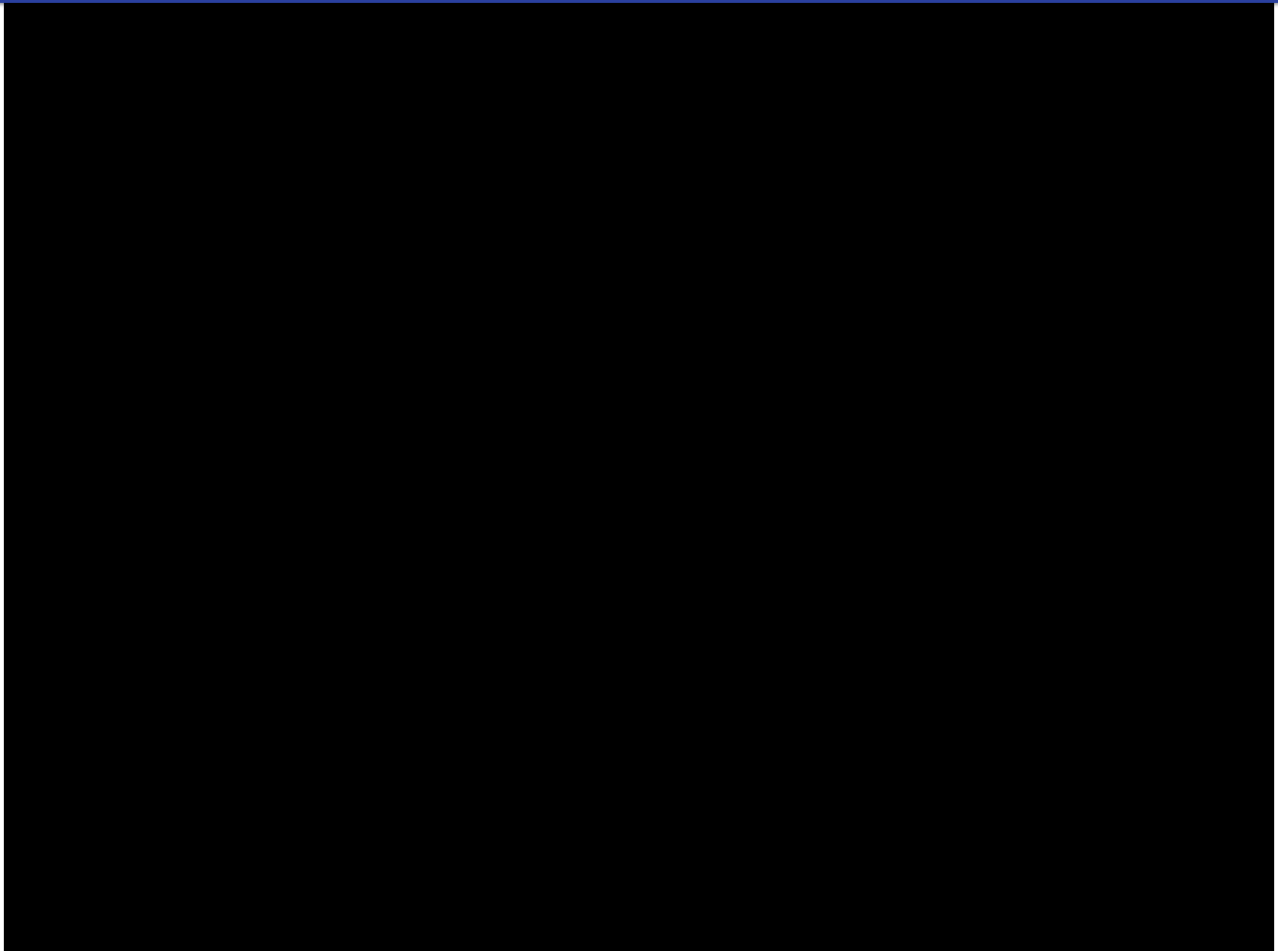
Gradient-based learners

Inspired by SIFT

$$\text{response} = \frac{\sum_{\text{region } R_d} \text{gradients for orientation } e_d}{\sum_{\text{region } R_d} \text{gradients for all orientations}}$$



Gradient-based weak learners



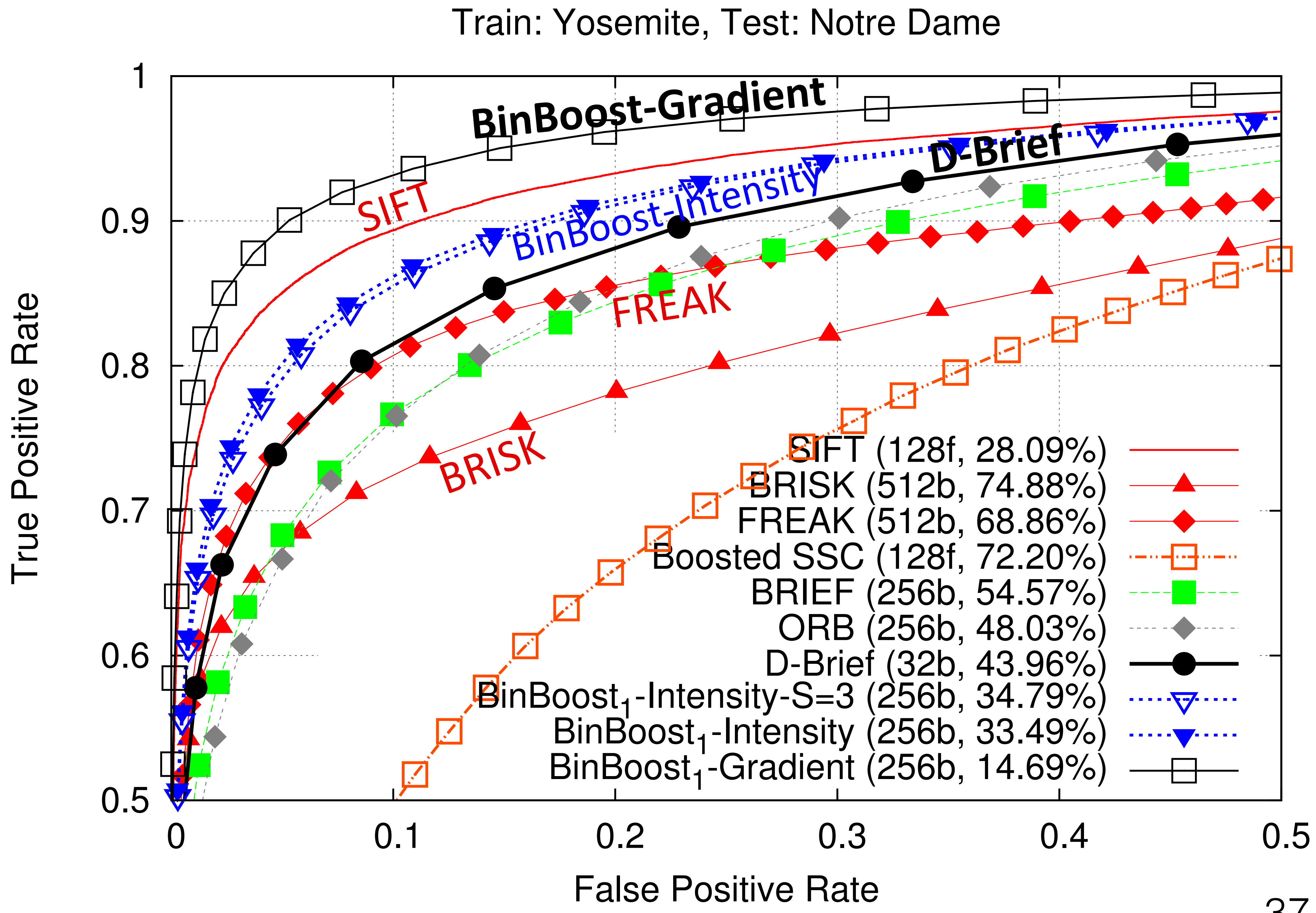
Weak learners comparison

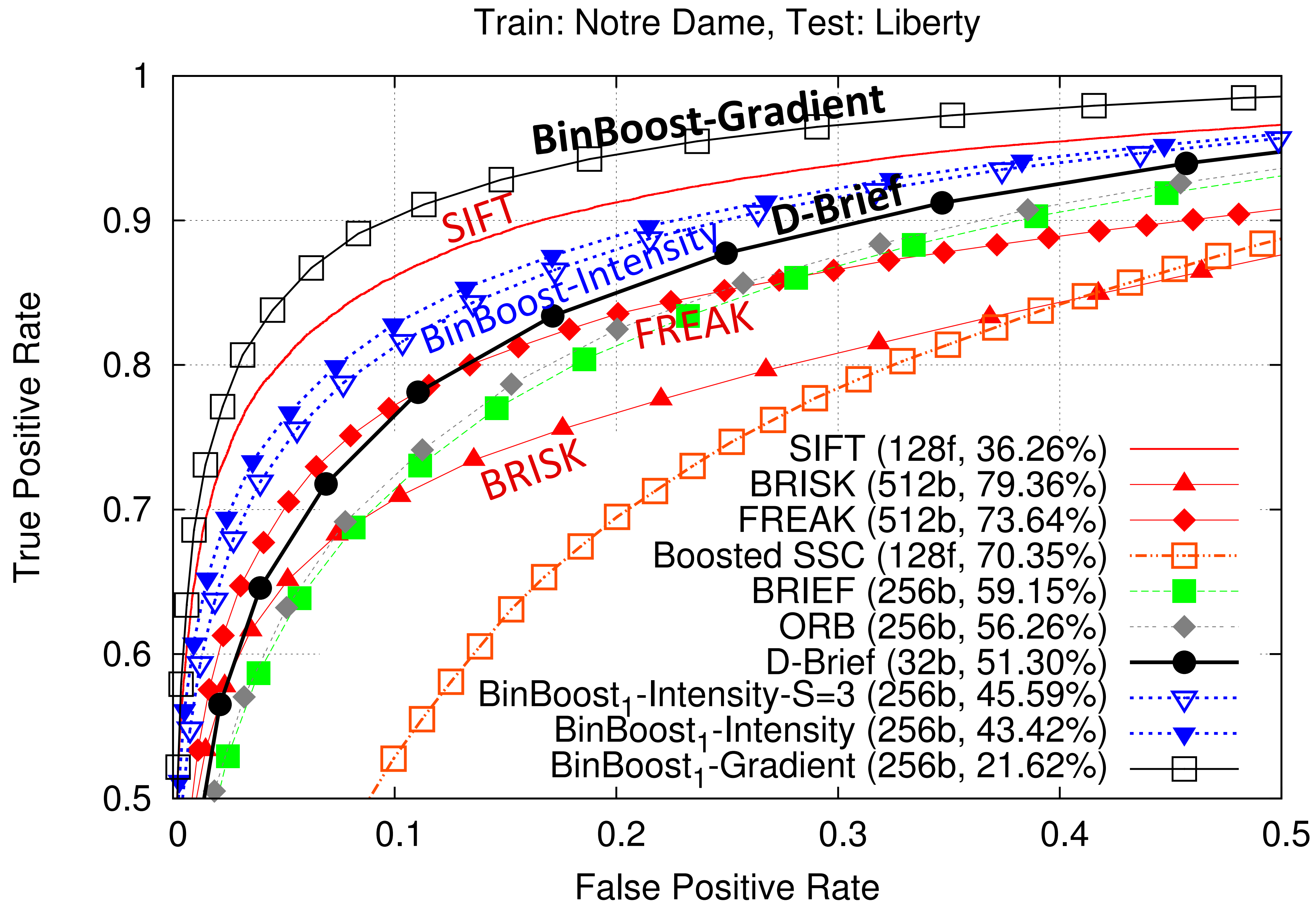
Intensity-based weak learners

Boosting improves the pooling configuration of other binary descriptors

Gradient-based weak learners

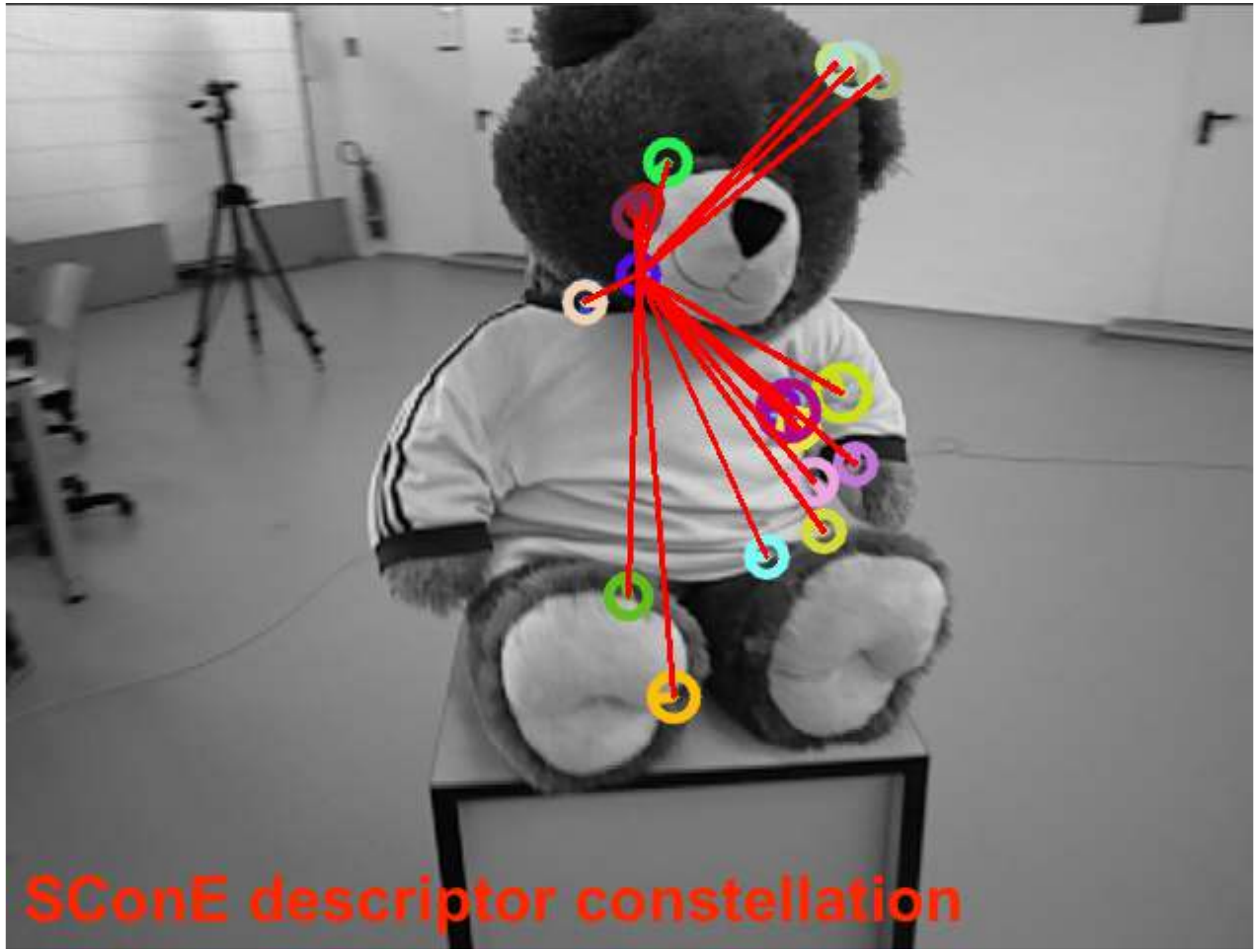
BinBoost-Gradient remains the best boosted descriptor of our framework

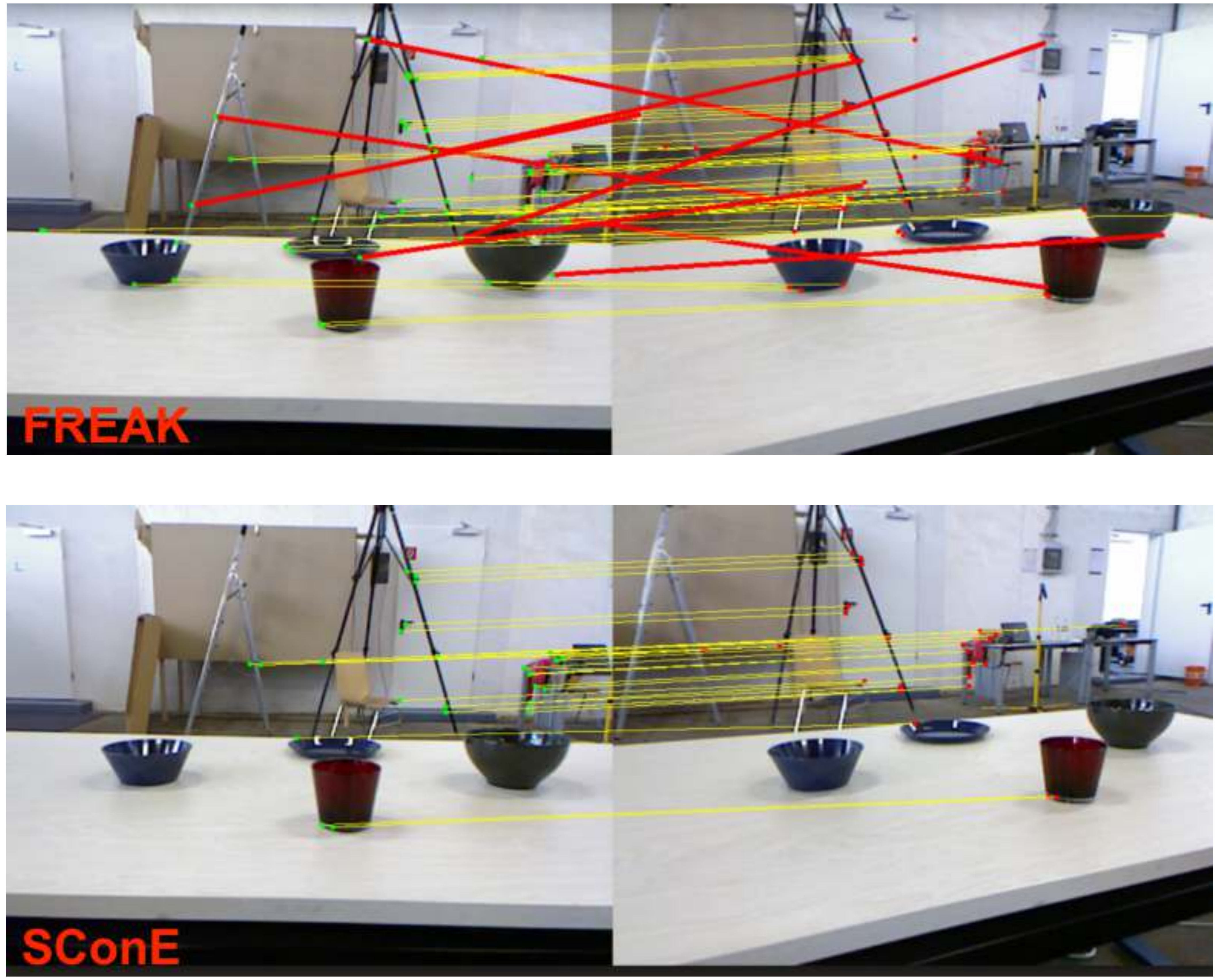




SConE

Siamese Constellation Embedding





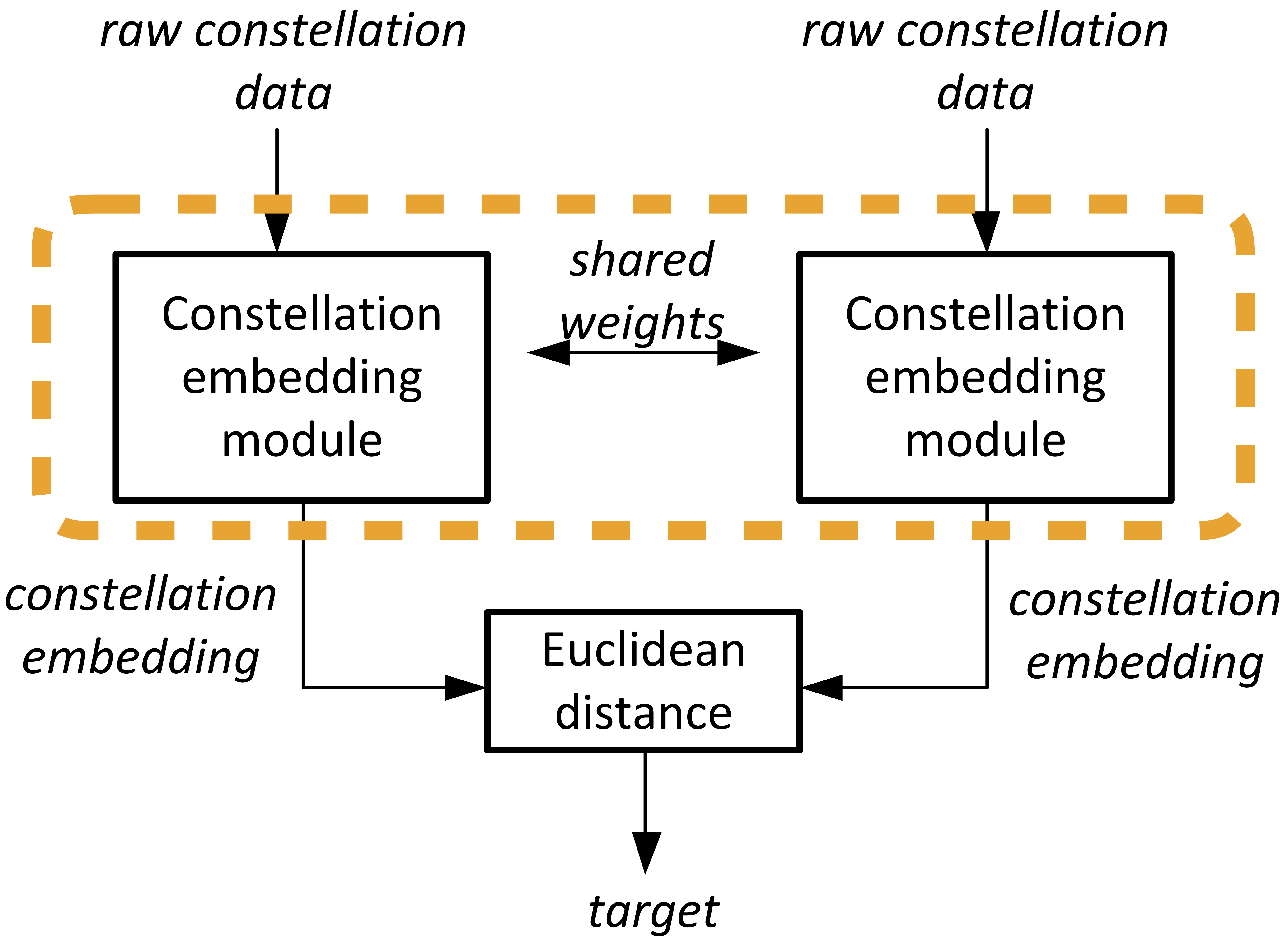
SConE

Siamese Constellation Embedding

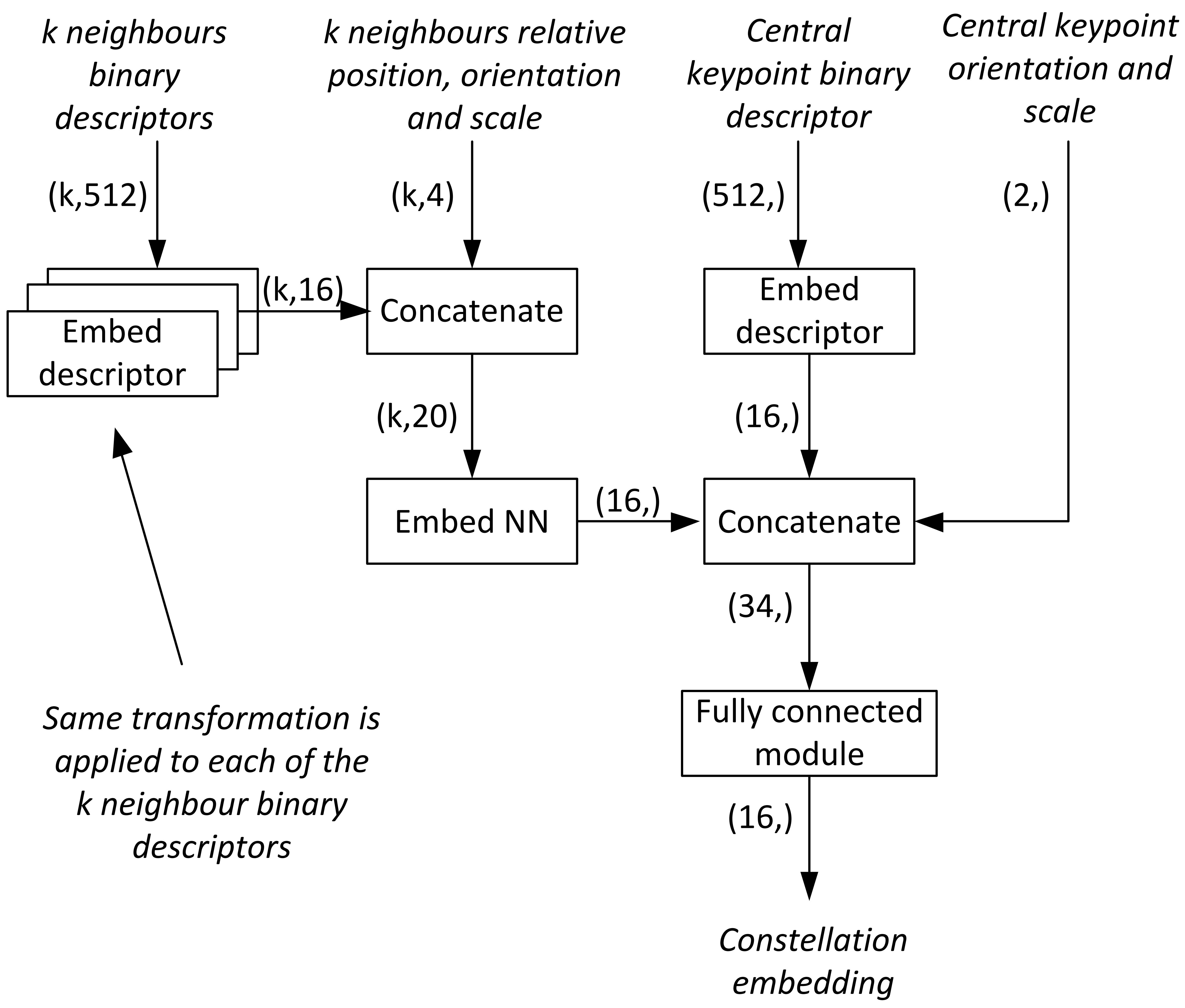
ECCV 2018

Submission ID 1735

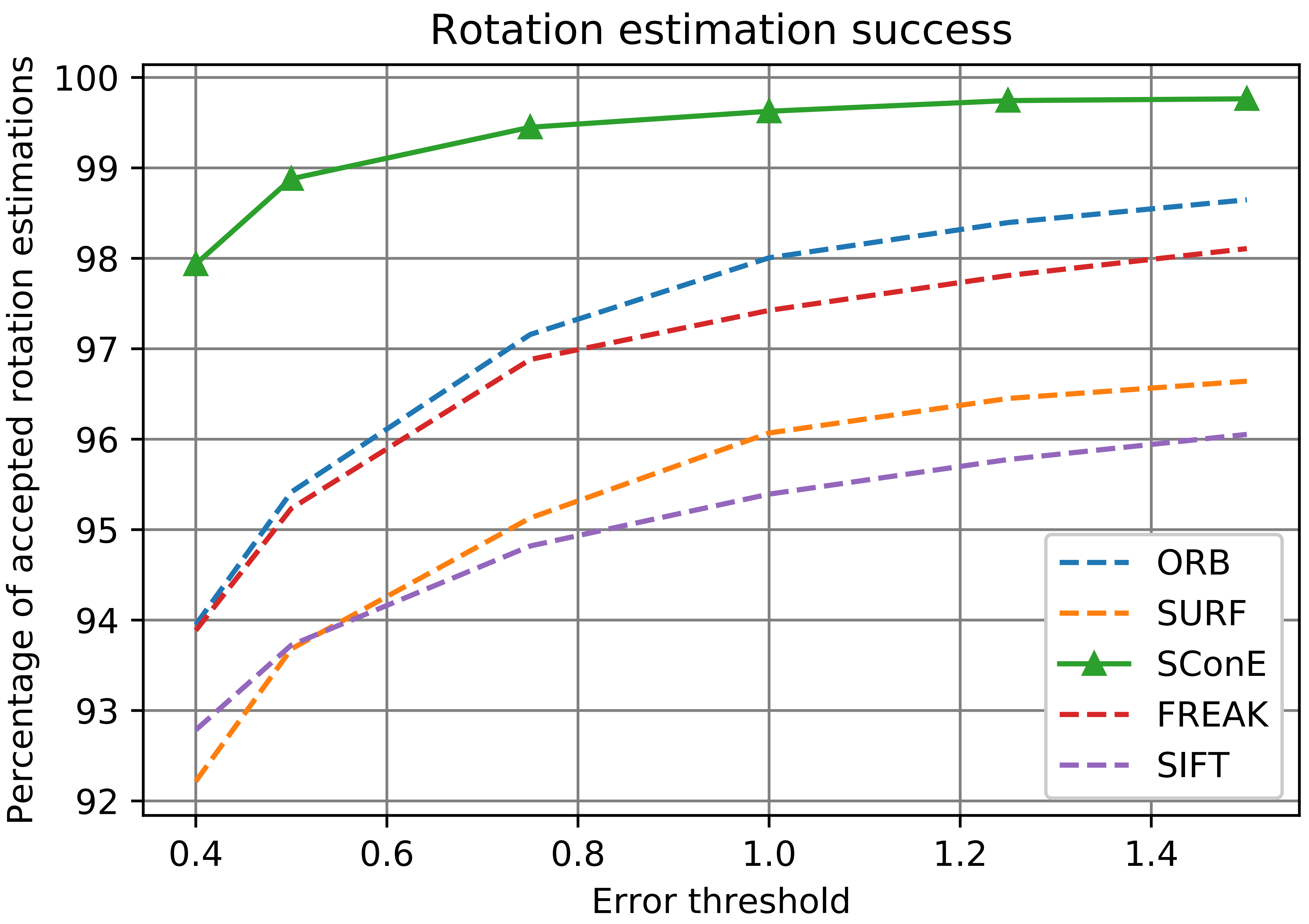
SConE architecture

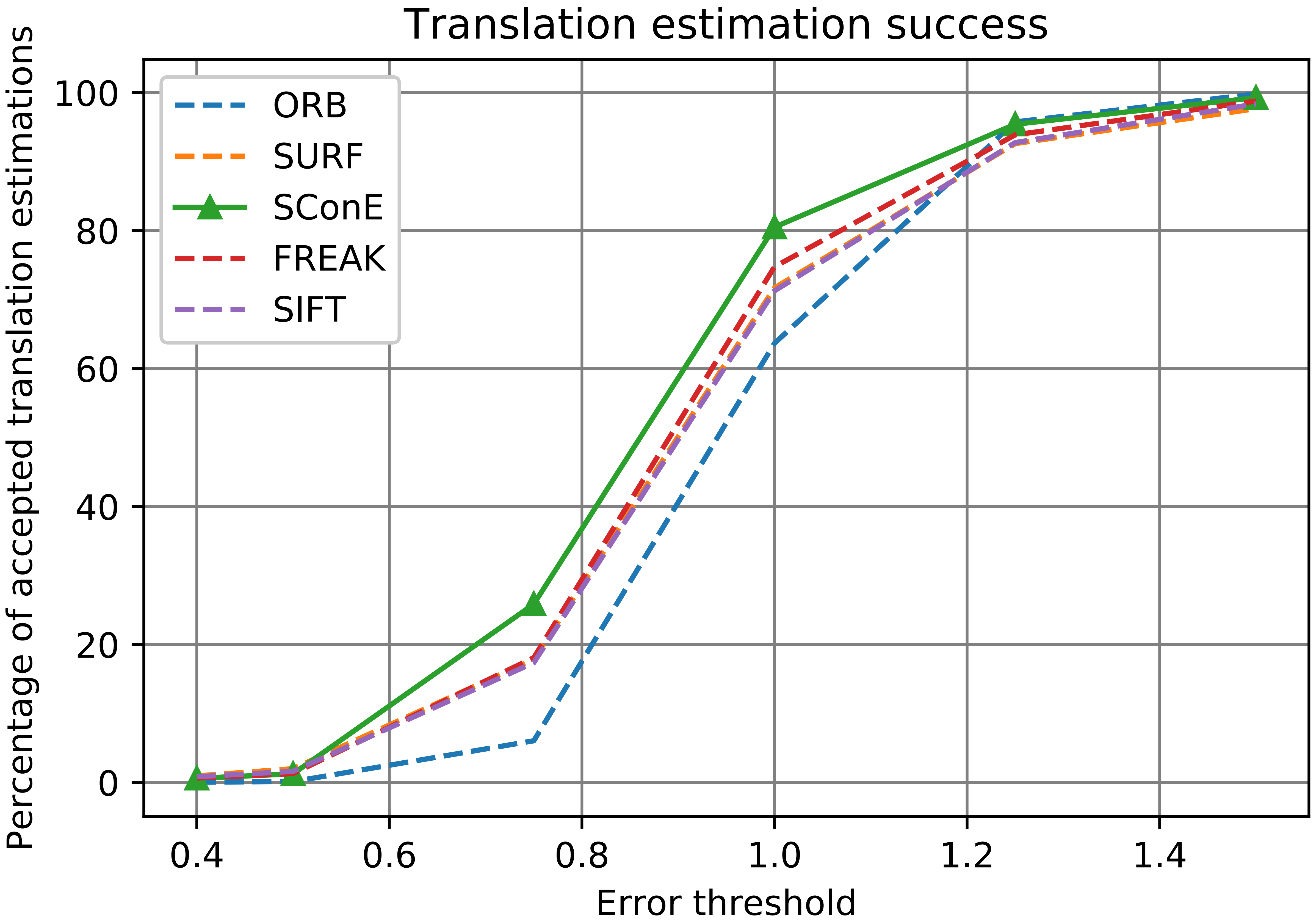


SConE architecture



Results - TUM Dataset

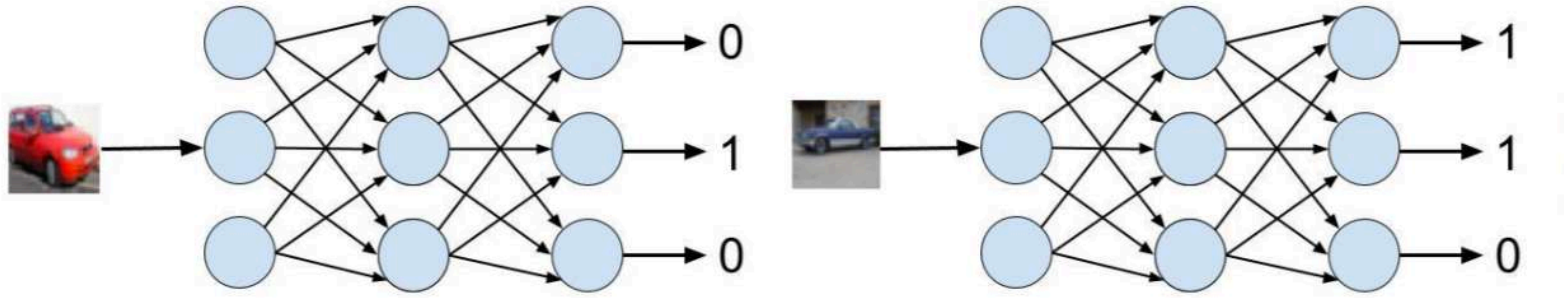




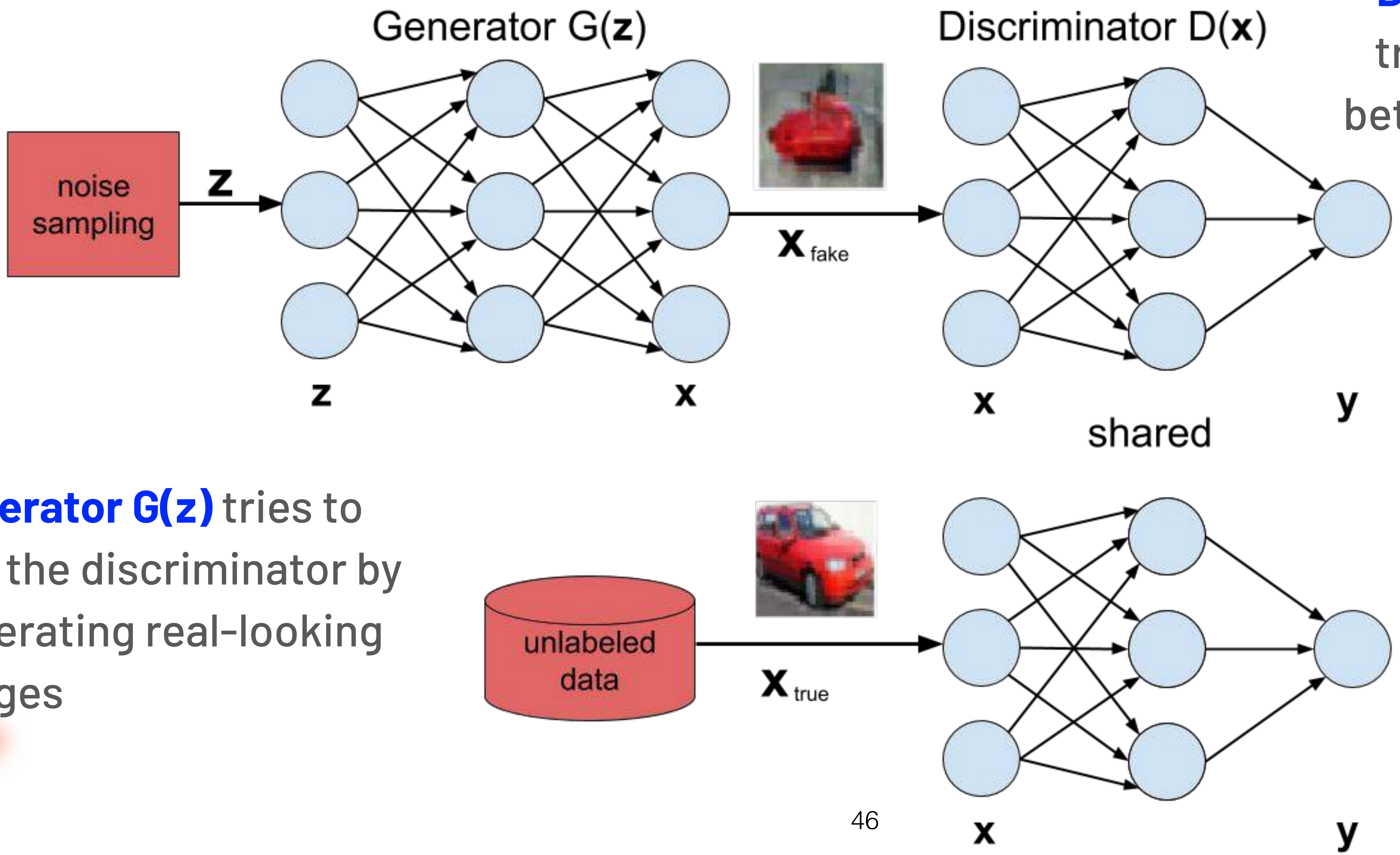
BinGAN

Unsupervised learning

How can we learn descriptors **without costly and imperfect data labeling?**



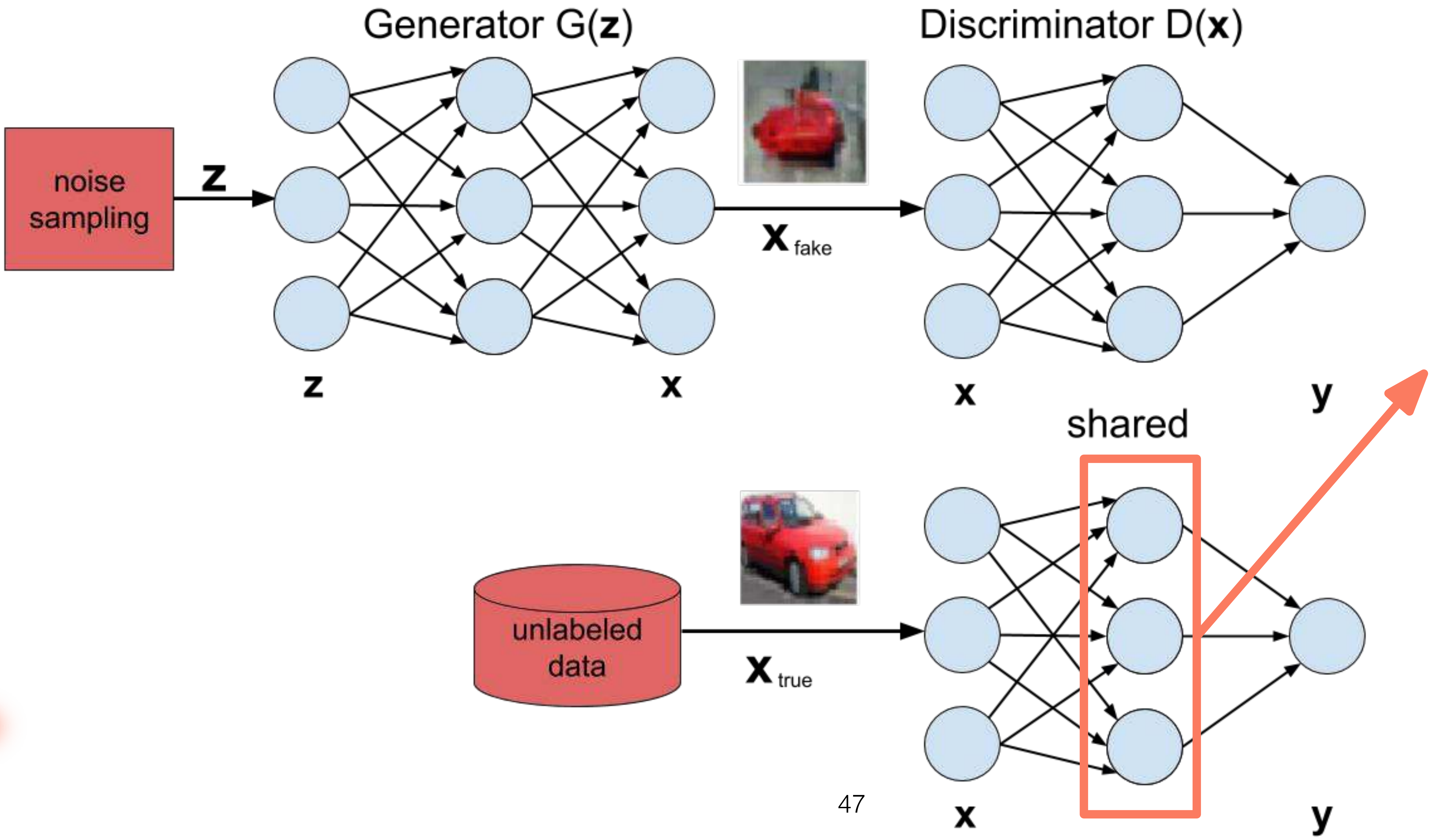
Generative Adversarial Networks



Discriminator D(x) tries to distinguish between real and fake images

Generator G(z) tries to fool the discriminator by generating real-looking images

Discriminator to represent the data



Nice data representation from adversarial training

How to get binary codes?

Discriminator D(x)
loss

$$\boxed{\begin{aligned} L_D = &-\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] \\ &-\mathbb{E}_{z\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \end{aligned}}$$

BinGAN loss

$$L = L_D + \lambda_{BRE} \cdot L_{BRE} + \lambda_{DMR} \cdot L_{DMR}$$

How to get binary codes?

Discriminator D(x)
loss

$$L_D = -\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] - \mathbb{E}_{z\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]$$

BinGAN loss

$$L = L_D + \lambda_{BRE} \cdot L_{BRE} + \lambda_{DMR} \cdot L_{DMR}$$

Binary Representation Entropy (BRE)

How to get binary codes?

Discriminator D(x)
loss

$$\boxed{\begin{aligned} L_D = &-\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x}))] \\ &-\mathbb{E}_{z\sim p_z(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \end{aligned}}$$

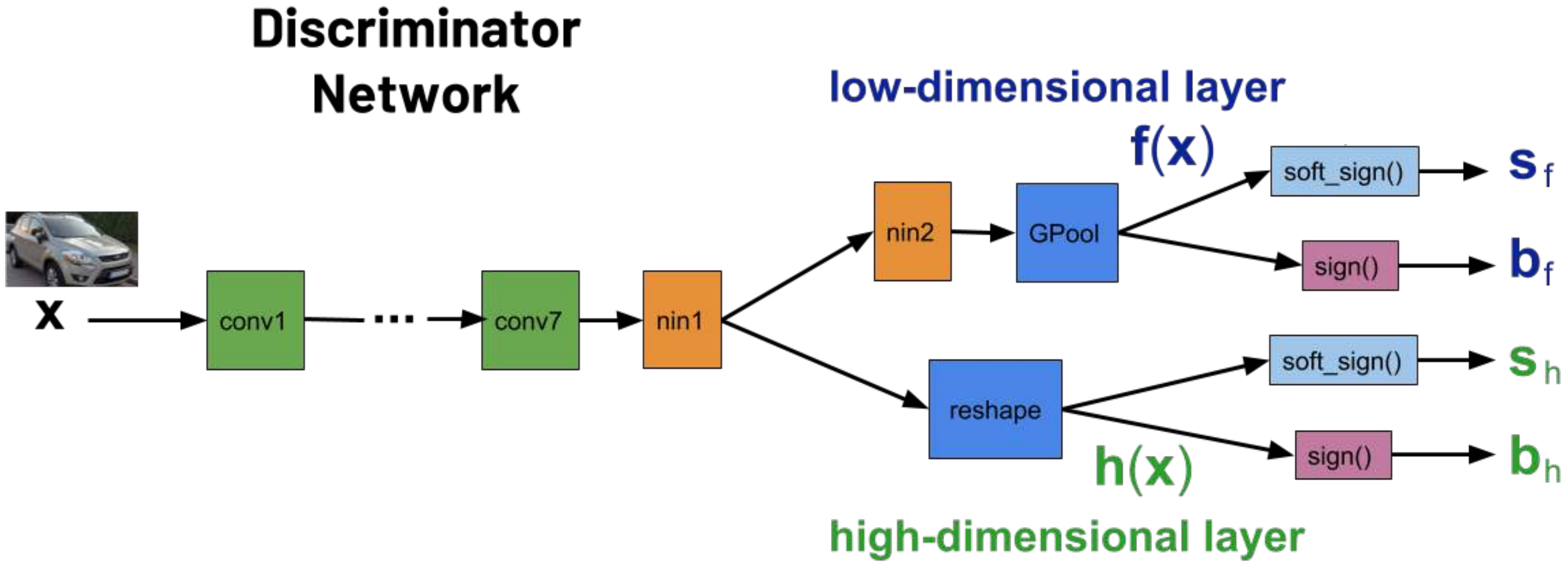
BinGAN loss

$$L = L_D + \lambda_{BRE} \cdot L_{BRE} + \lambda_{DMR} \cdot L_{DMR}$$

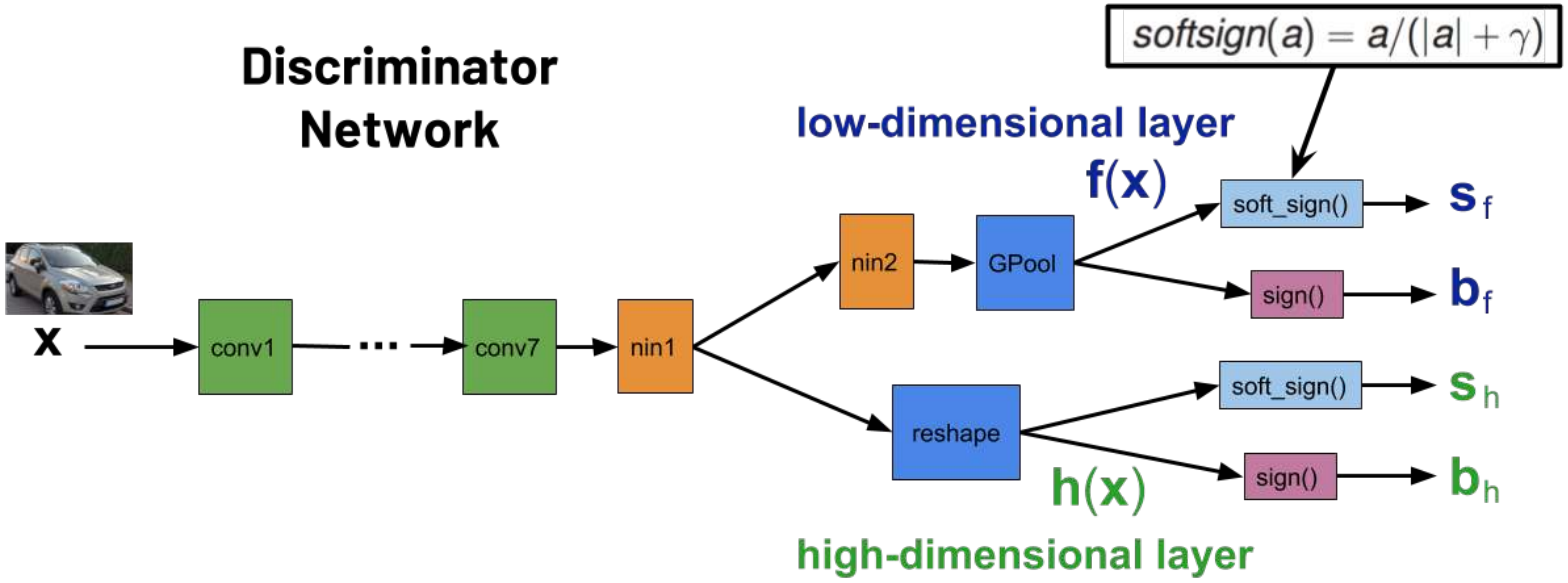
Binary Representation Entropy (BRE)

Distance Matching Regularizer (DMR)

BRE Regularizer



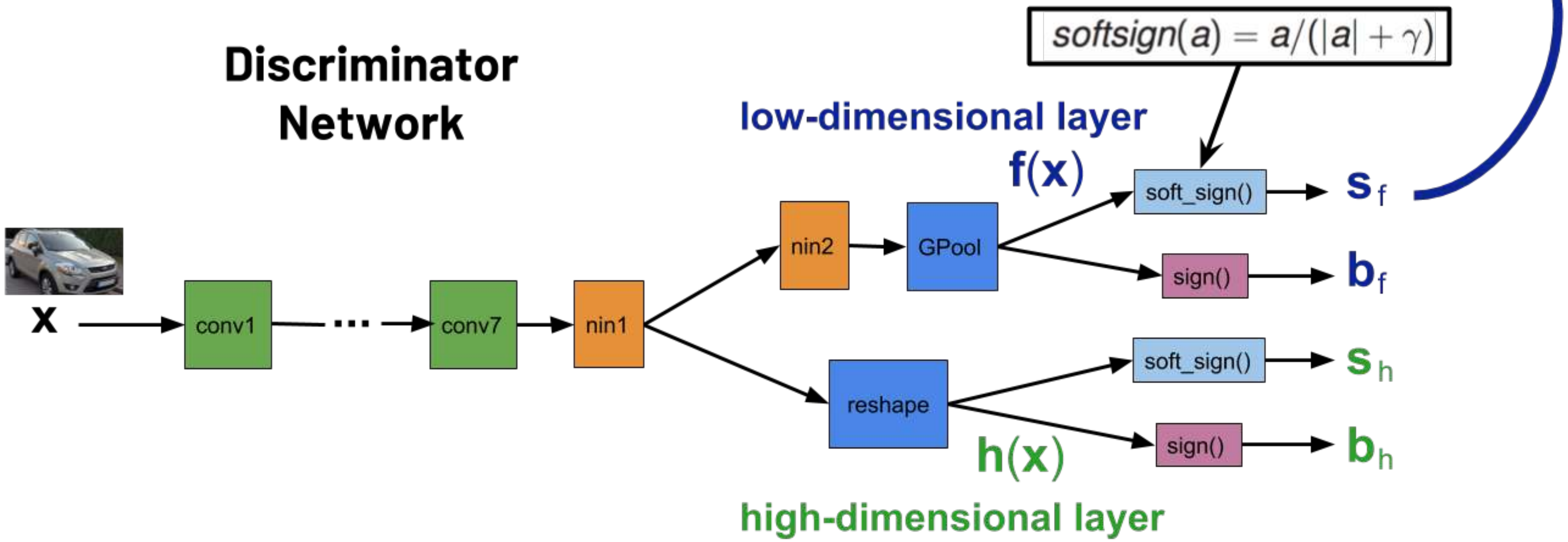
BRE Regularizer



BRE Regularizer

BRE regularizer

$$L_{BRE} = L_{ME} + L_{MAC} = \frac{1}{K}\sum_{k=1}^{K}(\bar{s}_{f,k})^2 + \sum_{k,j=1,k\neq j}^{N} w_{k,j}\frac{|\mathbf{s}_{f,k}^T \cdot \mathbf{s}_{f,j}|}{K}$$



BRE Regularizer

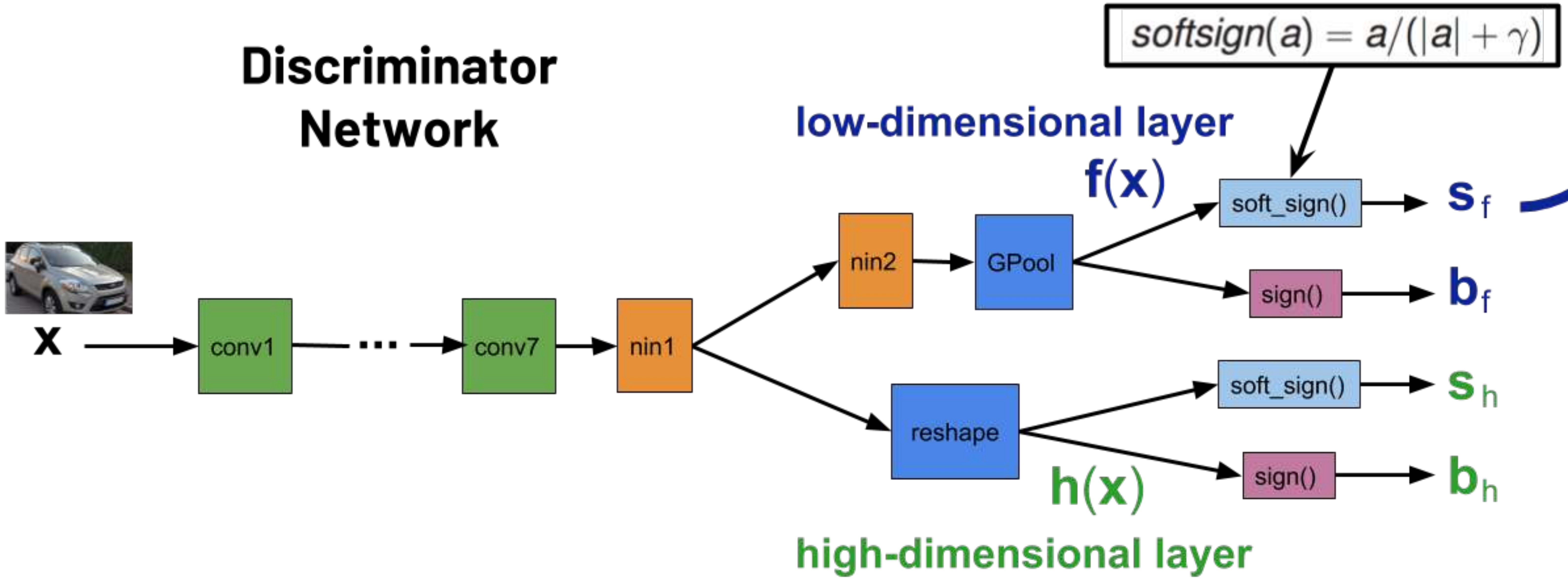
Increase diversity for descriptors with zero dot product

$$\exp\left\{\frac{-|\mathbf{b}_{h,k}^T \mathbf{b}_{h,j}|}{\beta \cdot M}\right\}$$

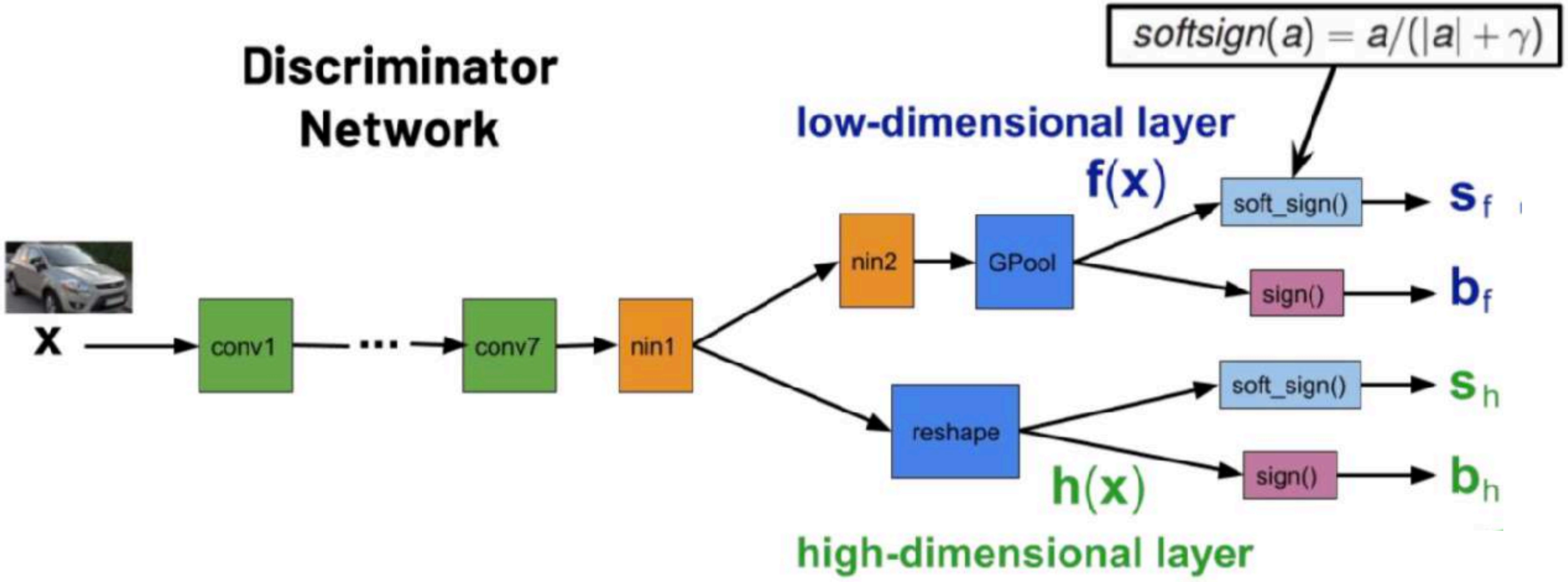
BRE regularizer

$$L_{BRE} = L_{ME} + L_{MAC} = \frac{1}{K}\sum_{k=1}^{K}(\bar{s}_{f,k})^2 + \sum_{k,j=1,k\neq j}^{N} \boxed{w_{k,j}} \frac{|\mathbf{s}_{f,k}^T \cdot \mathbf{s}_{f,j}|}{K}$$

Discriminator Network



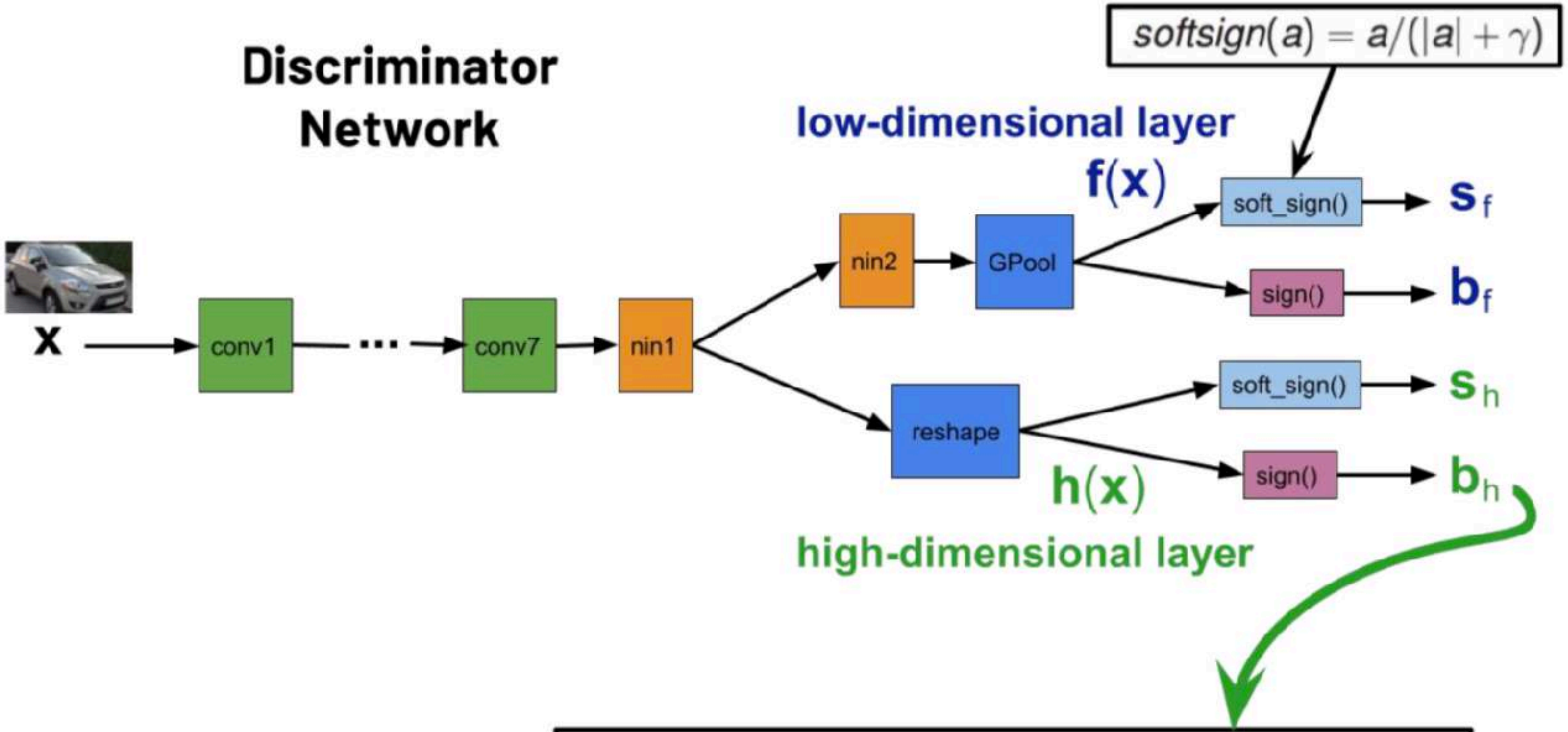
DMR Regularizer



DMR regularizer

$$L_{DMR} = \frac{1}{N(N-1)} \sum_{k,j=1,k\neq j}^{N} \left| \frac{\mathbf{b}_{h,k}^T \mathbf{b}_{h,j}}{M} - \frac{\mathbf{s}_{f,k}^T \mathbf{s}_{f,j}}{K} \right|$$

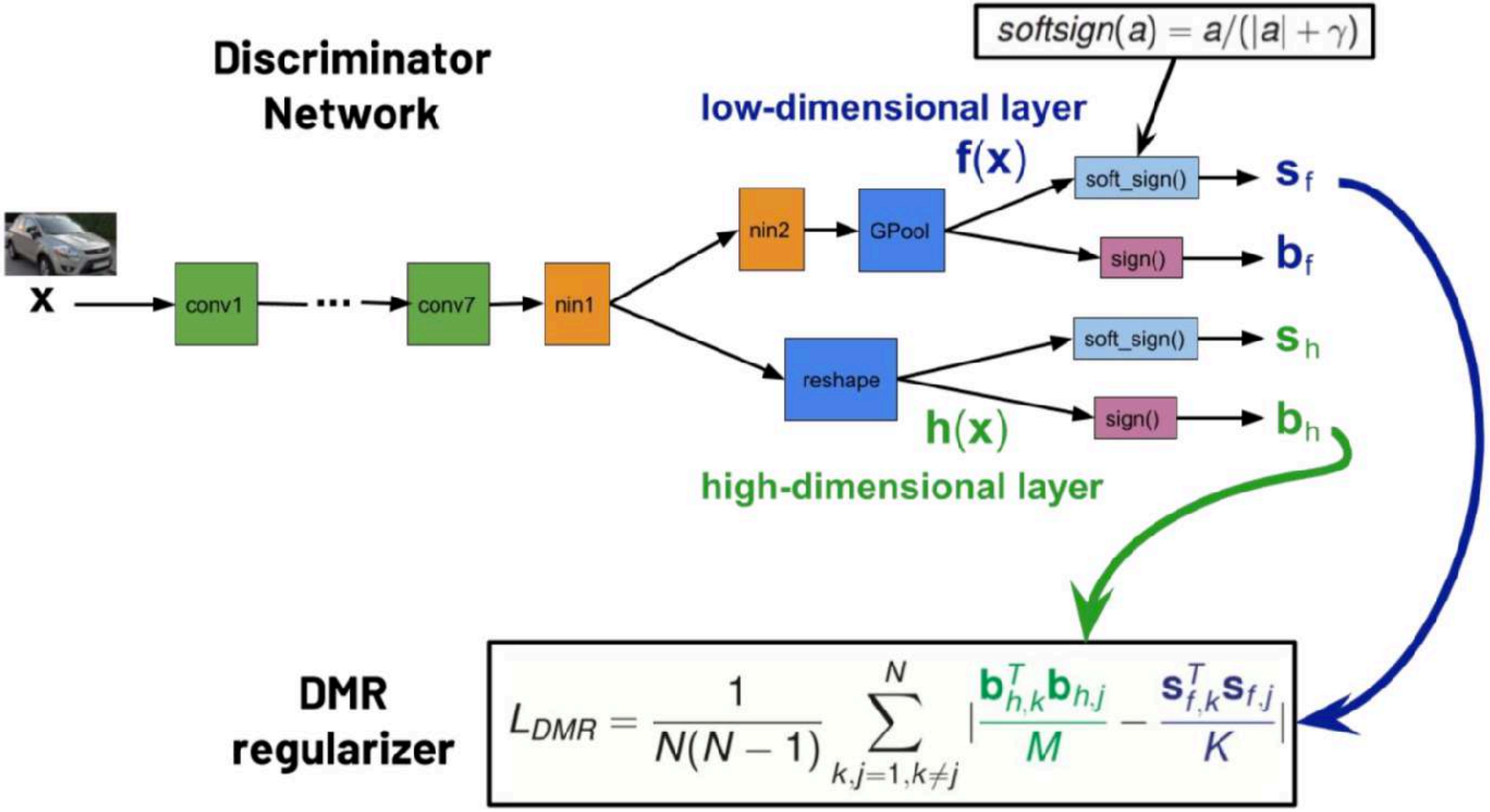
DMR Regularizer



DMR regularizer

$$L_{DMR} = \frac{1}{N(N-1)} \sum_{k,j=1,k\neq j}^{N} \left| \frac{\mathbf{b}_{h,k}^{T}\mathbf{b}_{h,j}}{M} - \frac{\mathbf{s}_{f,k}^{T}\mathbf{s}_{f,j}}{K} \right|$$

DMR Regularizer



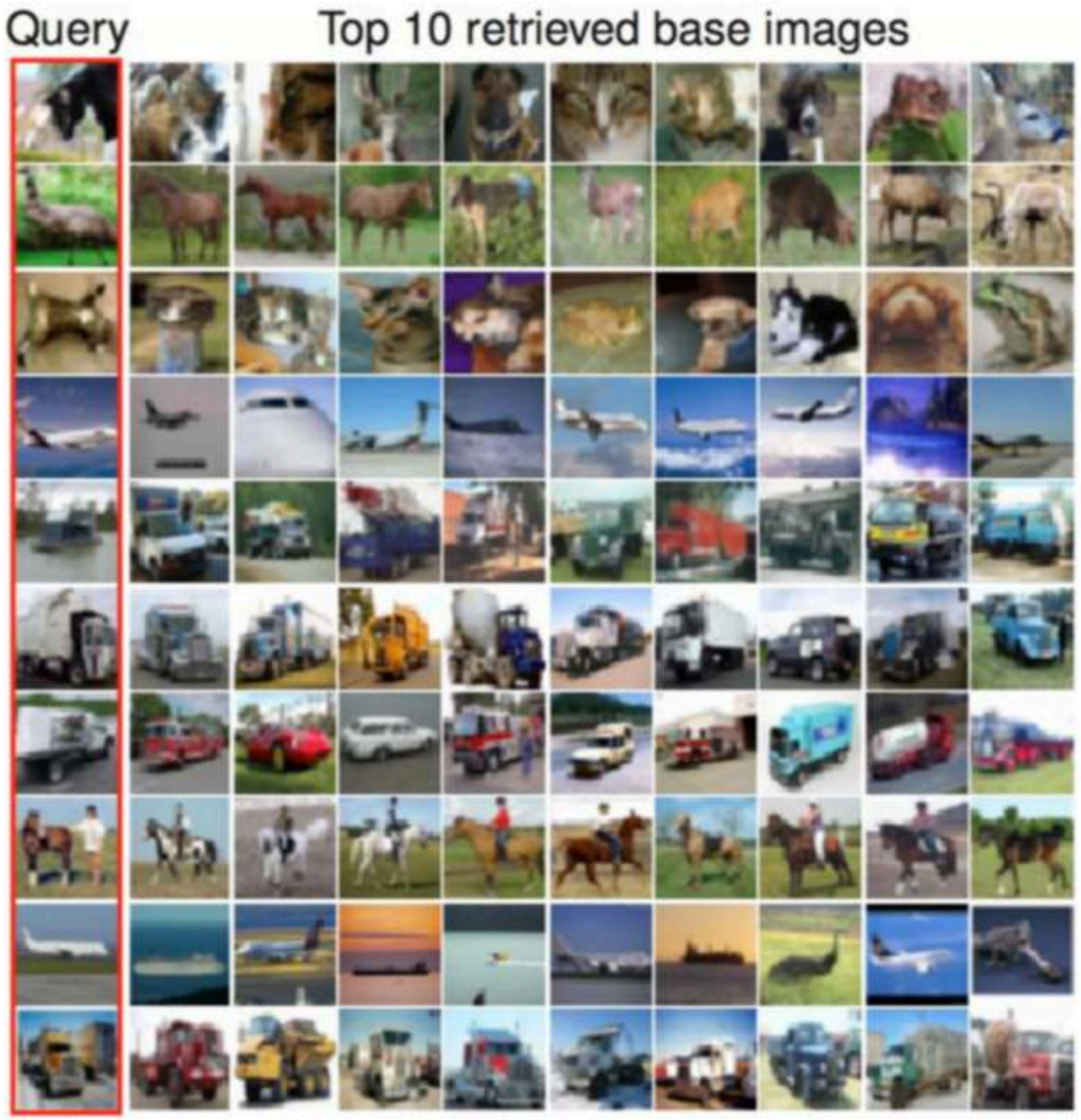
Results - image matching

Train	Yosemite		Notre Dame		Liberty		Average
Test	Notre Dame	Liberty	Yosemite	Liberty	Notre Dame	Yosemite	FPR@95%
Supervised							
LDAHash (16 bytes)	51.58	49.66	52.95	49.66	51.58	52.95	51.40
D-BRIEF (4 bytes)	43.96	53.39	46.22	51.30	43.10	47.29	47.54
BinBoost (8 bytes)	14.54	21.67	18.96	20.49	16.90	22.88	19.24
RFD (50-70 bytes)	11.68	19.40	14.50	19.35	13.23	16.99	15.86
Binary L2-Net (32 bytes)	2.51	6.65	4.04	4.01	1.9	5.61	4.12
Unsupervised							
SIFT (128 bytes)	28.09	36.27	29.15	36.27	28.09	29.15	31.17
BRISK (64 bytes)	74.88	79.36	73.21	79.36	74.88	73.21	75.81
BRIEF (32 bytes)	54.57	59.15	54.96	59.15	54.57	54.96	56.23
DeepBit (32 bytes)	29.60	34.41	63.68	32.06	26.66	57.61	40.67
DBD-MQ (32 bytes)	27.20	33.11	57.24	31.10	**25.78**	57.15	38.59
BinGAN (32 bytes)	**16.88**	**26.08**	**40.80**	**25.76**	27.84	**47.64**	**30.76**

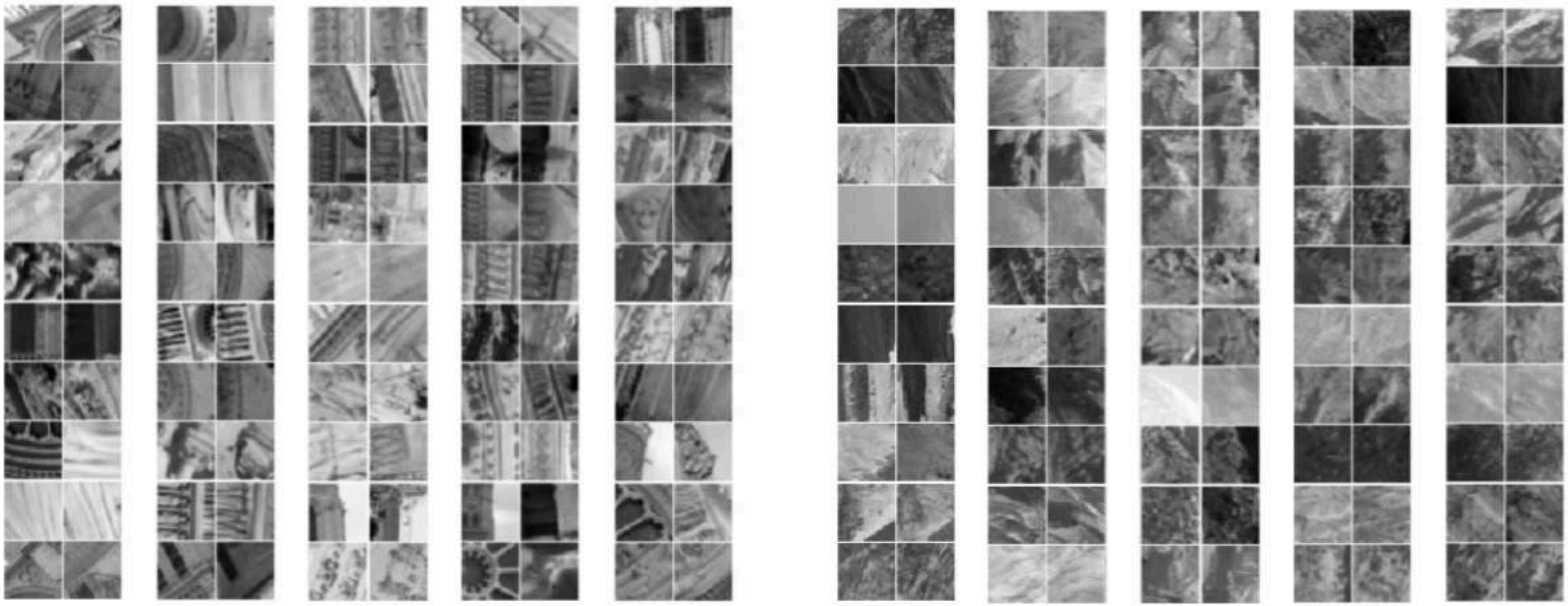
Results - image retrieval

Mean Average Precision (**mAP**) - **top 1000**.

Method	16 bit	32 bit	64 bit
KHM	13.59	13.93	14.46
SphH	13.98	14.58	15.38
SpeH	12.55	12.42	12.56
SH	12.95	14.09	13.89
PCAH	12.91	12.60	12.10
LSH	12.55	13.76	15.07
PCA-ITQ	15.67	16.20	16.64
DH	16.17	16.62	16.96
DeepBit	19.43	24.86	27.73
DBD-MQ	21.53	26.50	31.85
BinGAN	**30.05**	**34.65**	**36.77**



Semi-supervised learning?



3dAAE

Representing 3D point clouds

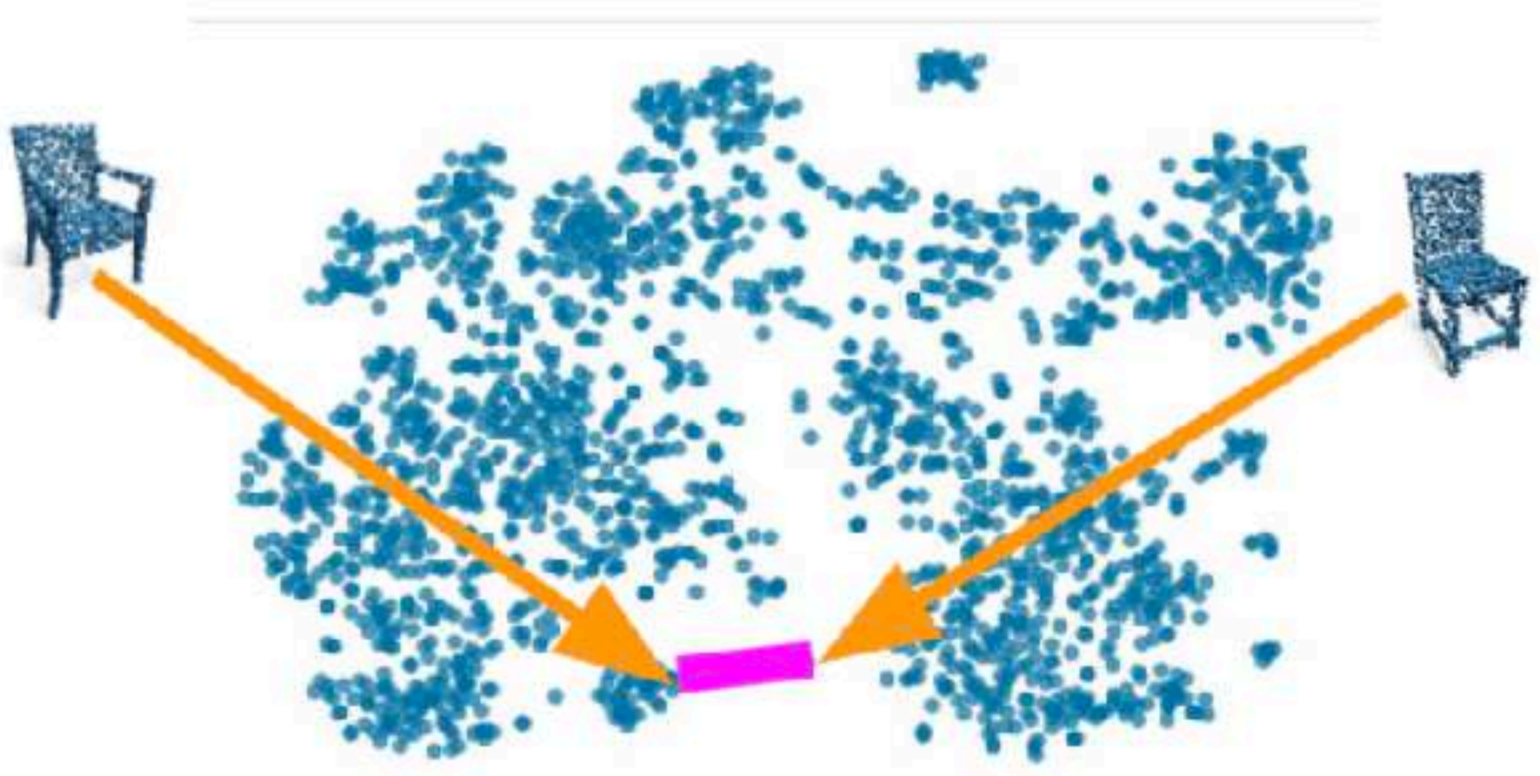
We seek **representations** of **3D point clouds** that can be useful for:

- sampling unseen examples (data augmentation)
- interpolating between 3D point clouds
- retrieval
- reconstruction
- clustering

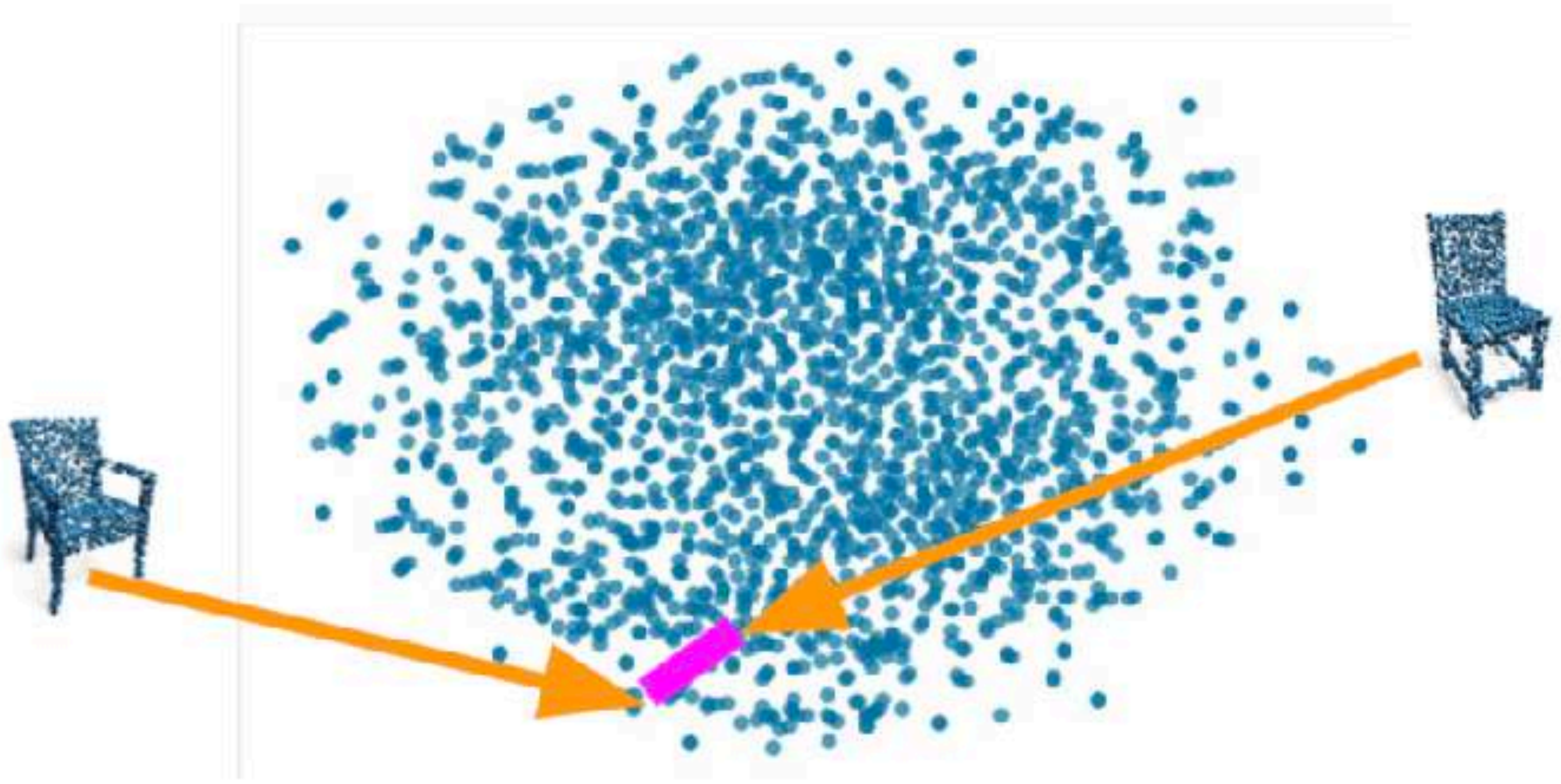
… and they need to be **efficient.**

Current methods

- **representation learning** decoupled from **generation**
- **assume normal distribution** in the latent space
- interpolation space **not continuous**

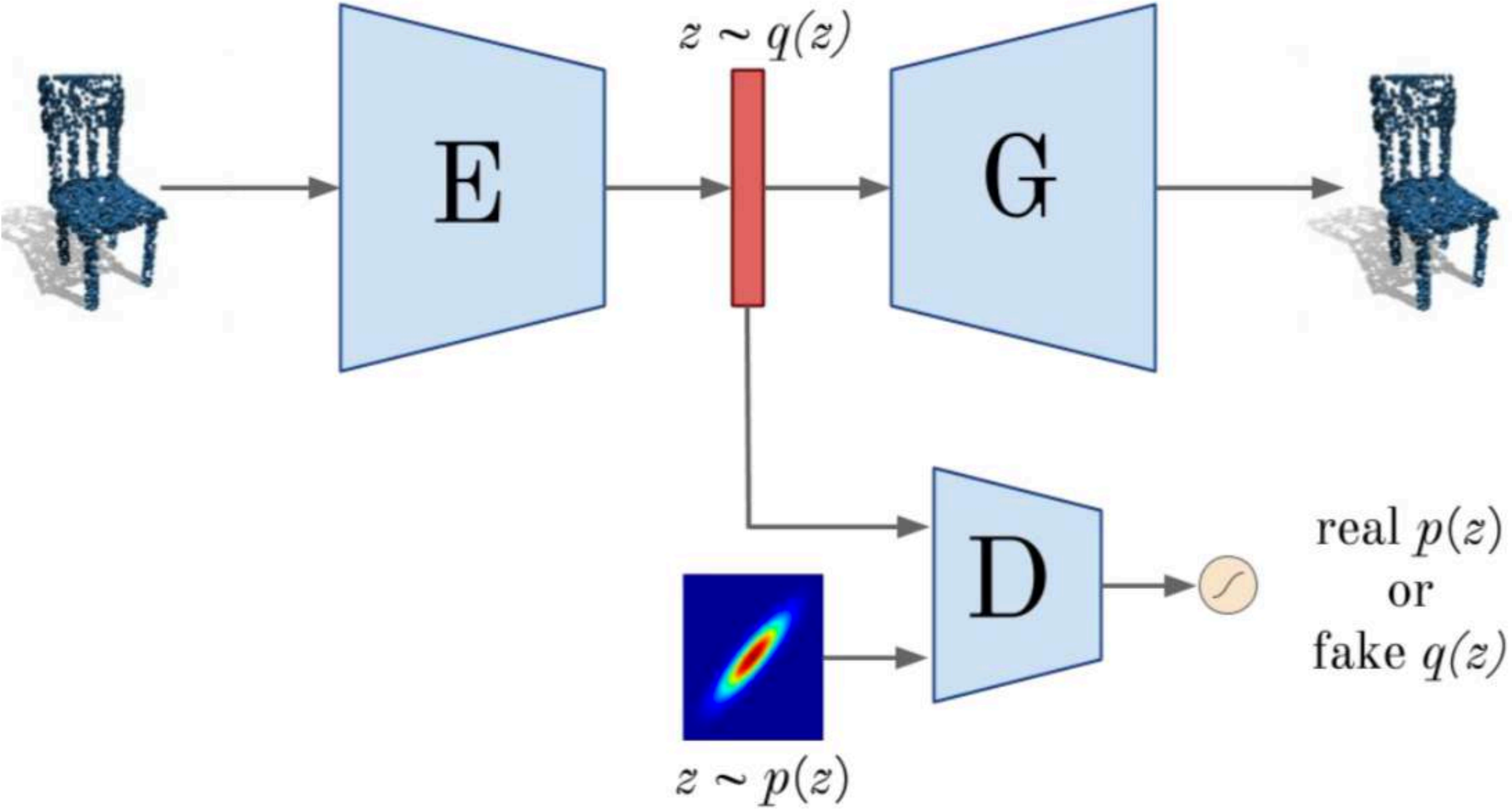


current

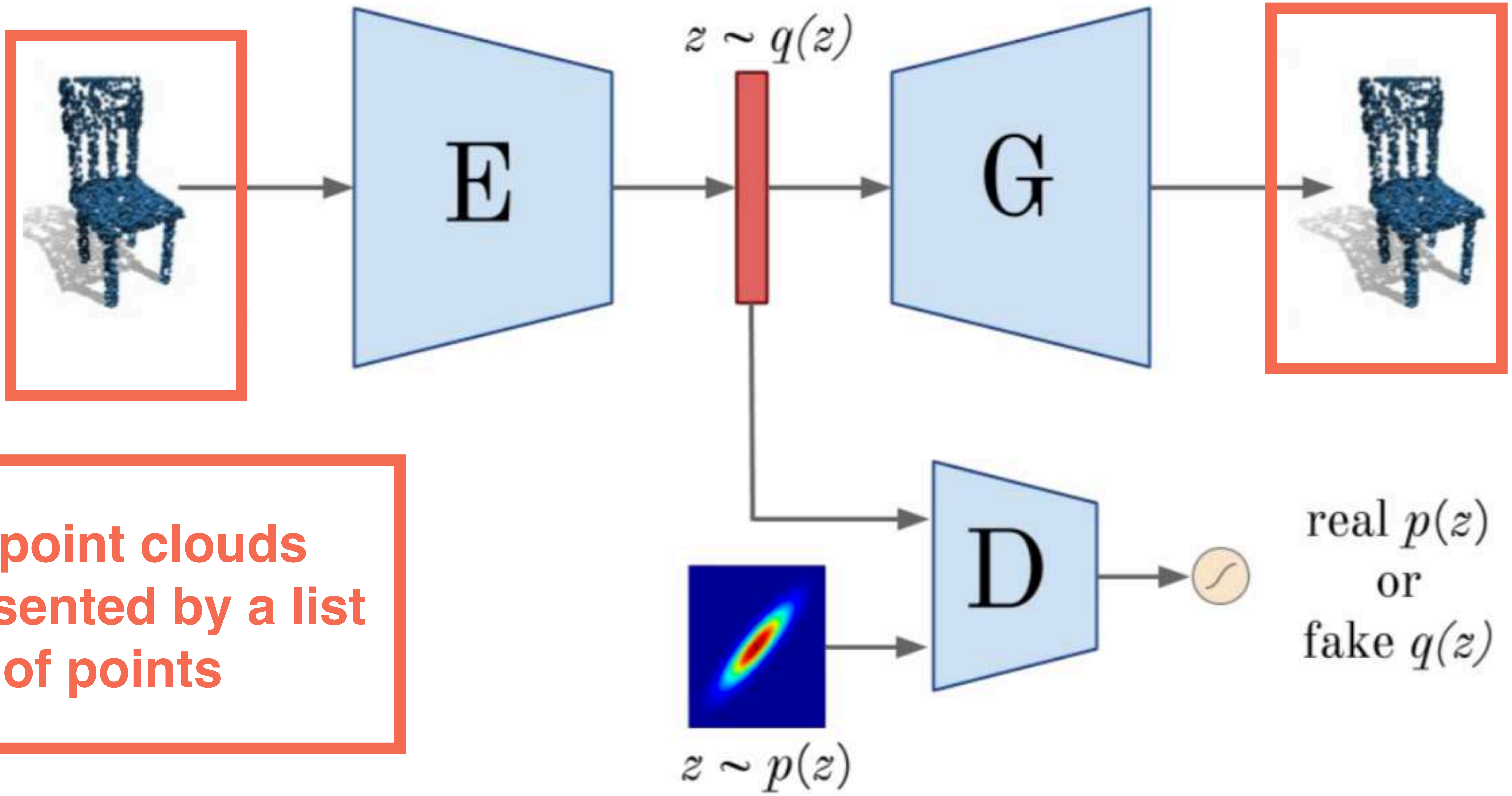


expected

Our method: 3dAAE

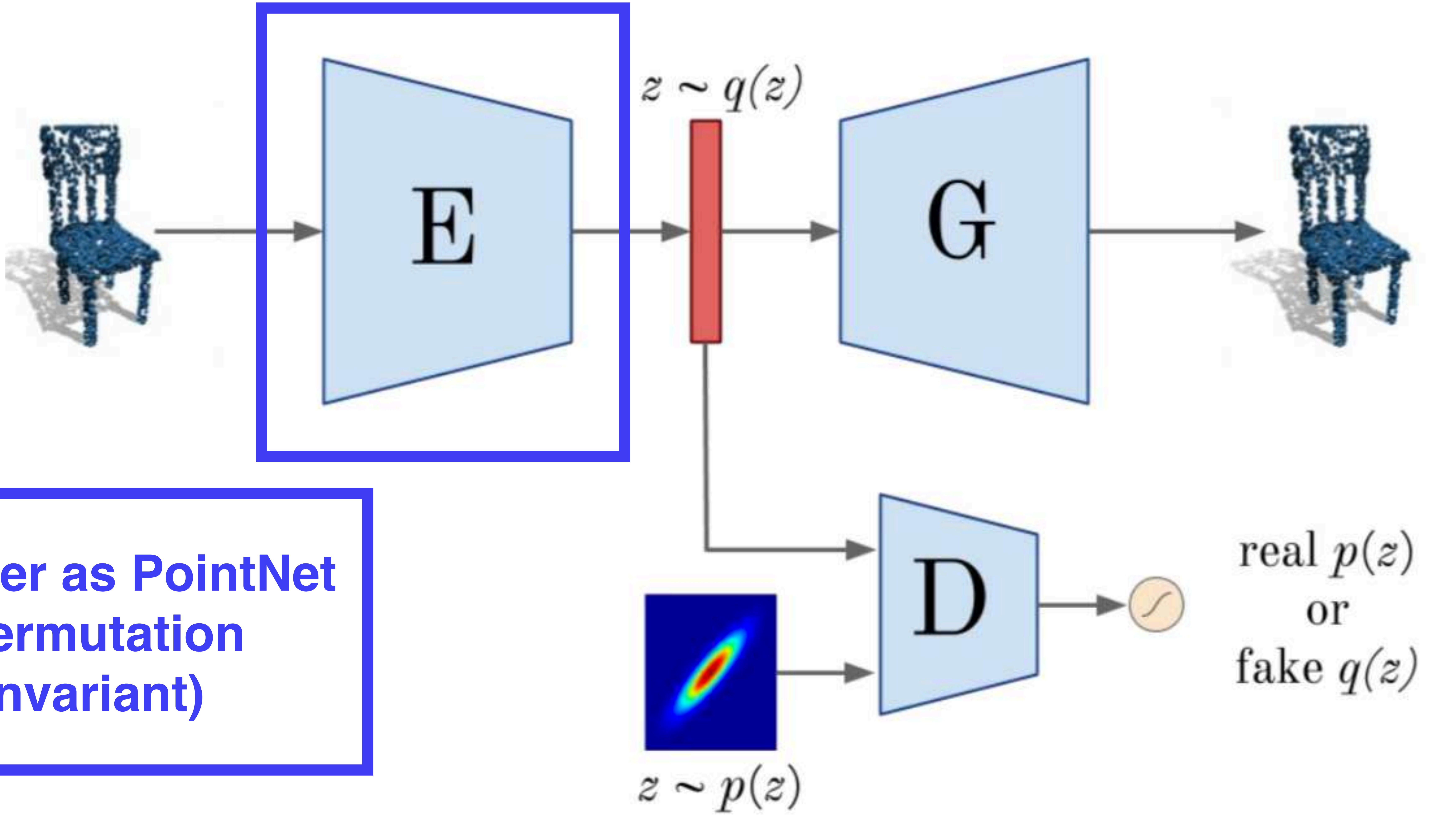


Our method: 3dAAE



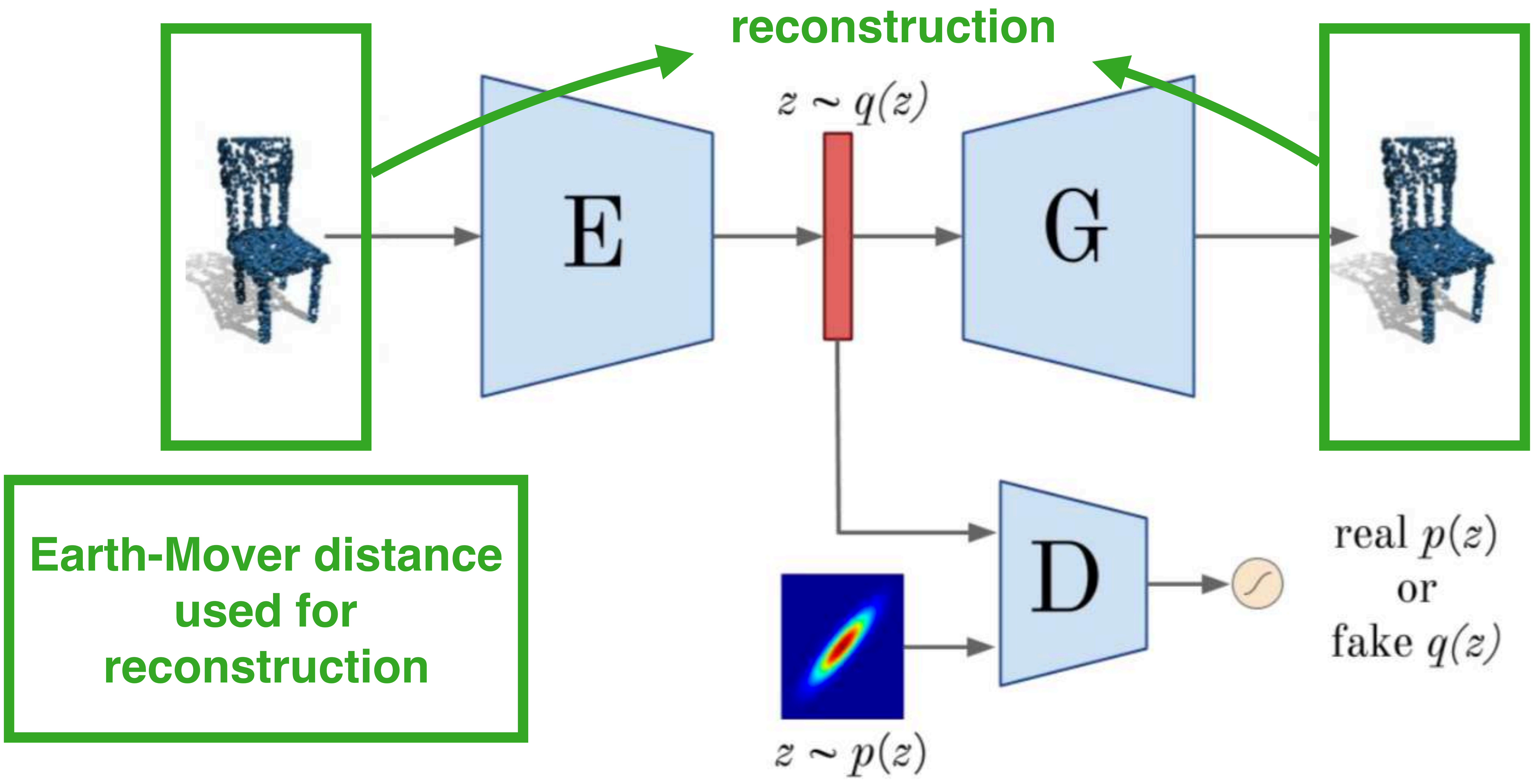
3D point clouds represented by a list of points

Our method: 3dAAE

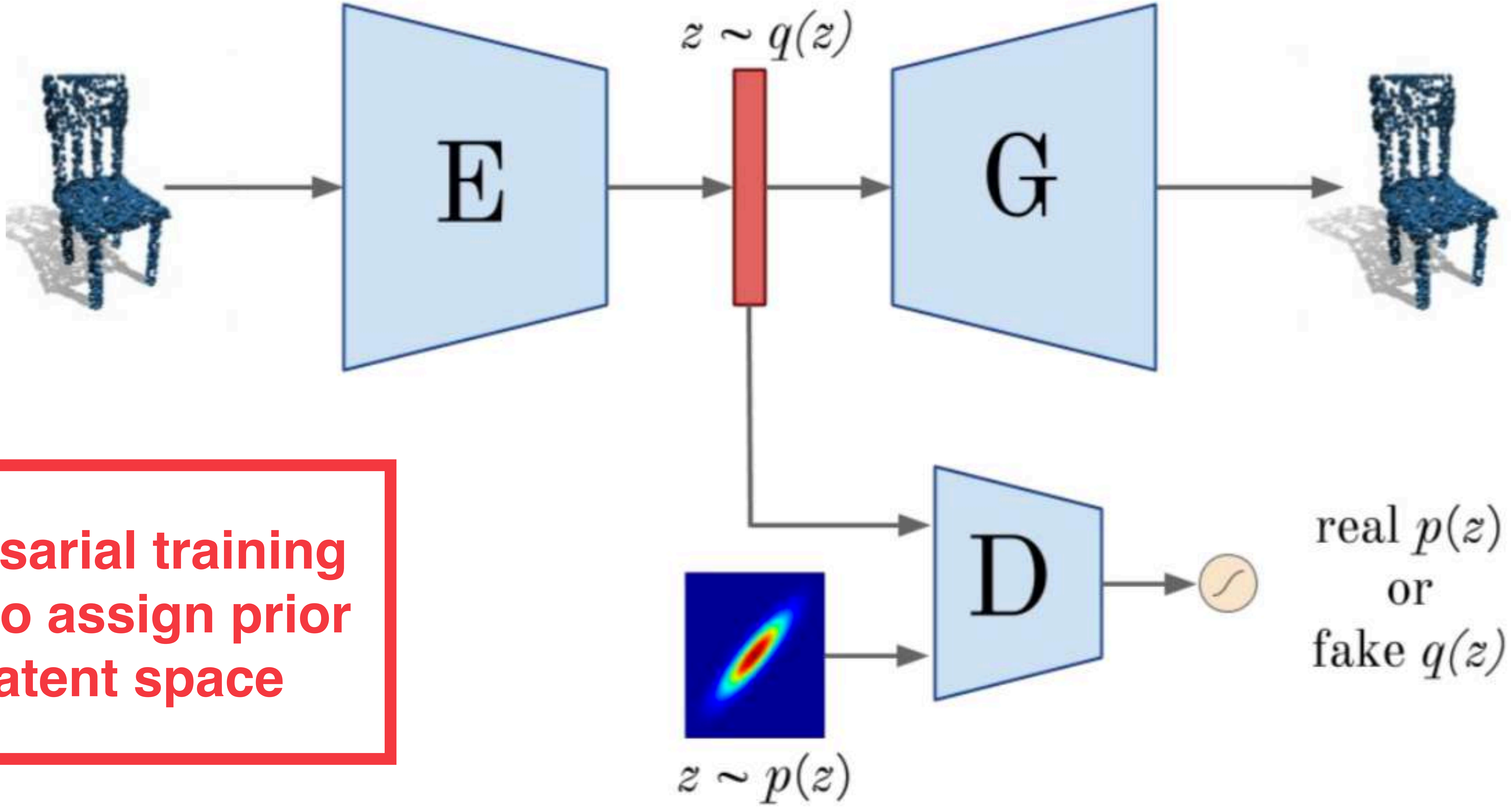


Encoder as PointNet (permutation invariant)

Our method: 3dAAE

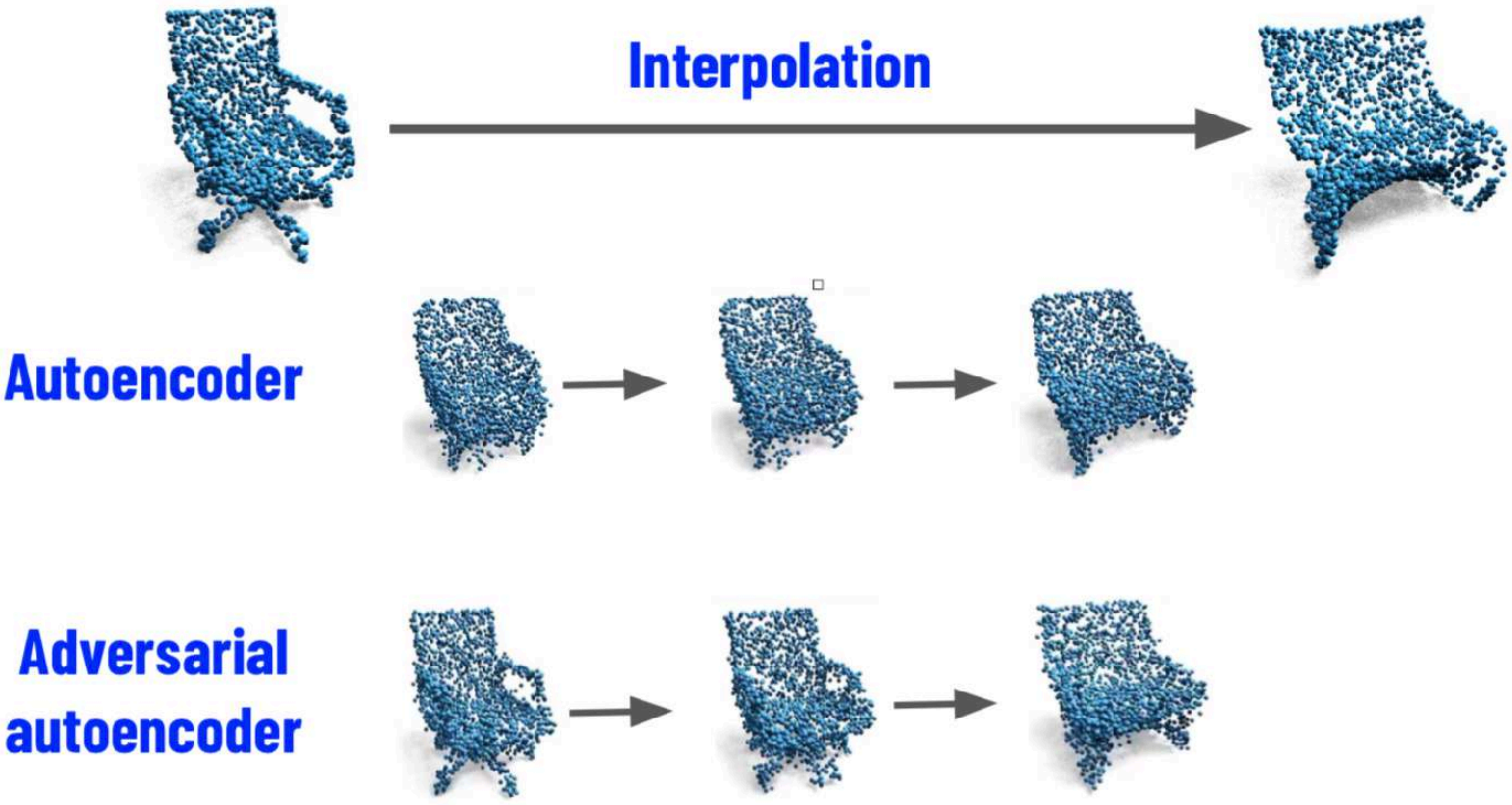


Our method: 3dAAE

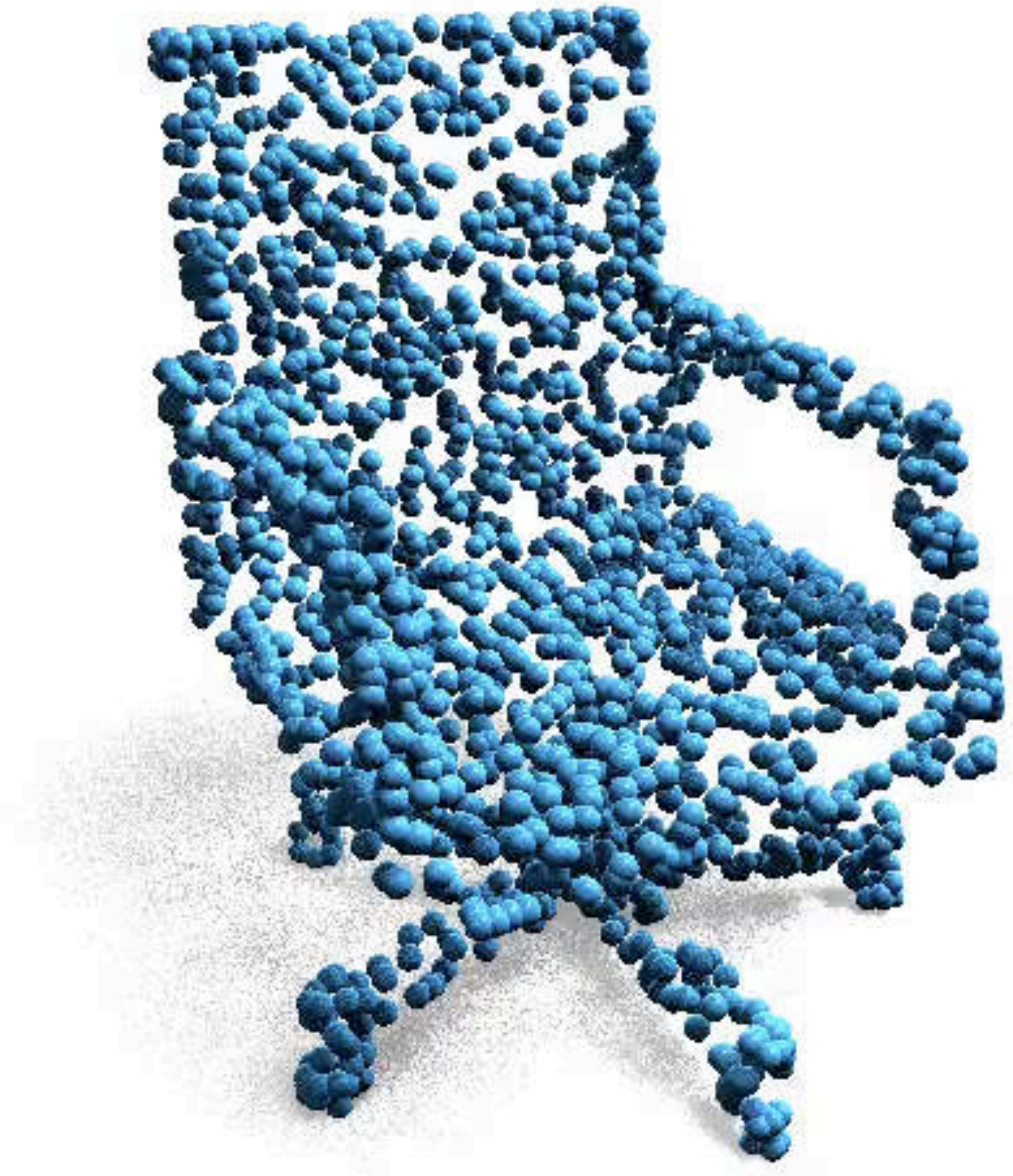


Adversarial training used to assign prior to latent space

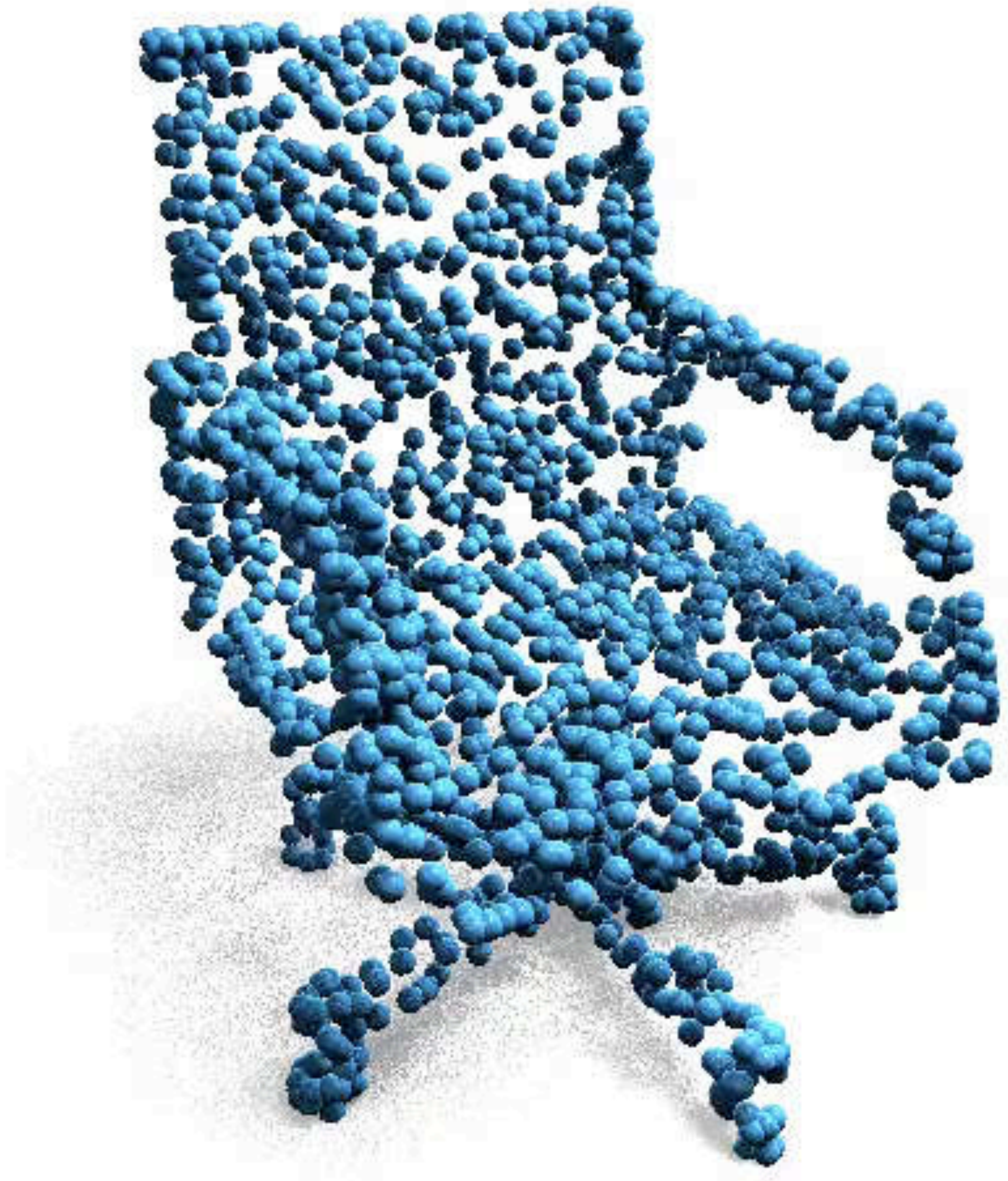
Interpolation



Interpolation - video



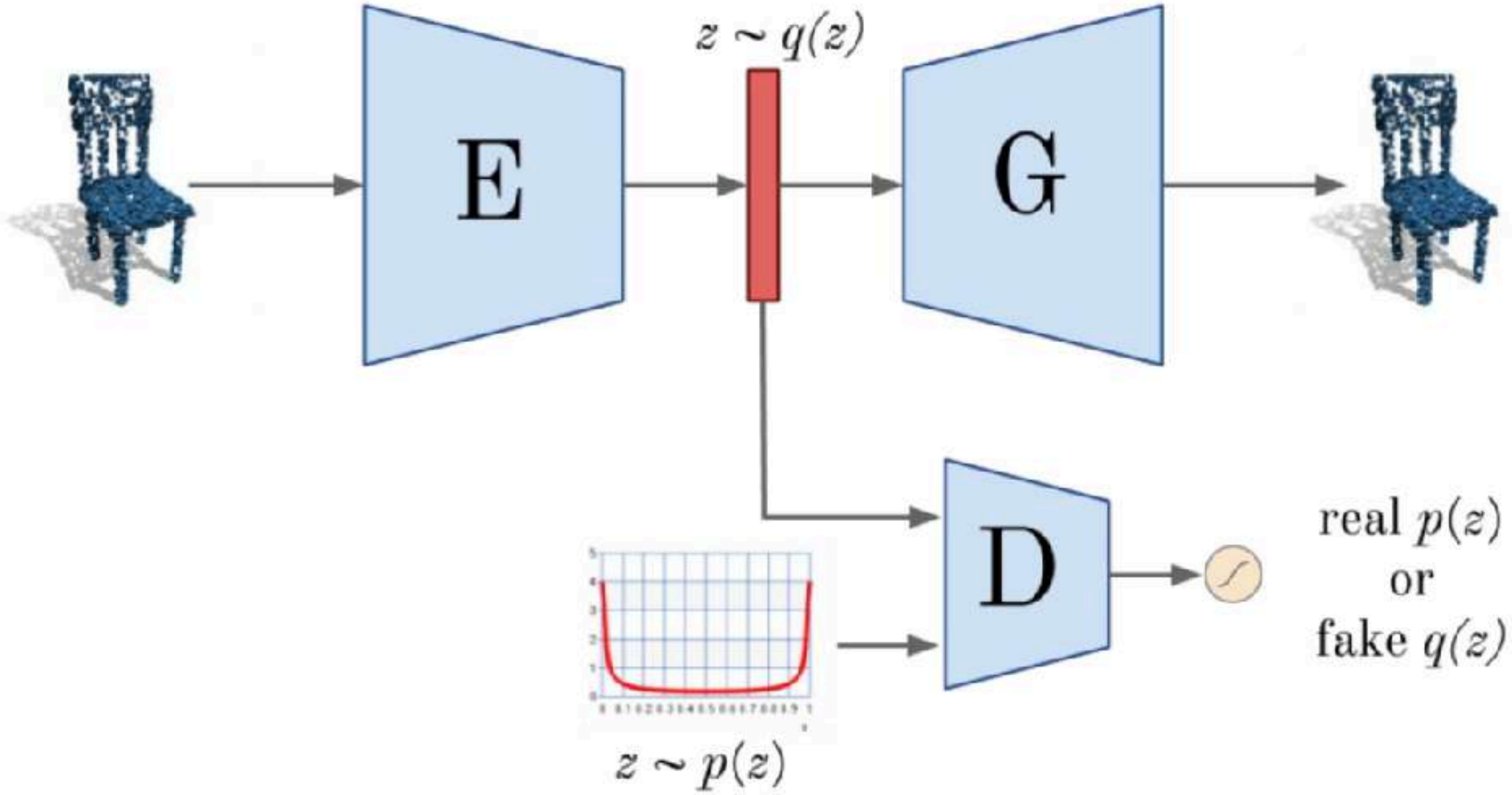
baseline



ours (3dAAE)

Interpolation

3dAAA Binary



Method	Numeric	Binary
AE	0.829	0.787
3dAAE	0.807	0.768
3dAAE-Bernoulli	0.913	0.892
3dAAE-Beta	**0.939**	**0.921**

Table 4. Retrieval results on *ShapeNet* dataset with 5 categories: car, rifle, sofa and table. We report results for numerical and binary features.

Conclusions

Conclusions

- Learning representation useful for **retrieval** and **matching**
- Binary descriptors are **efficient** and **cheap** to store
- They can be learnt with **linear projections, boosting** and **Siamese Networks** (ECCV'12, NIPS'12, CVPR'12, ECCVW18)
- Compact and robust representations (also for 3D points) can be learnt in an **unsupervised manner** using **generative models** (NeurIPS'18)

Acknowledgments





TU Graz

université de BORDEAUX

University of Victoria

NYU

TOOPLOOX scientific board











TOOPLOOX AI

Warsaw University of Technology

voyage

Google

Wrocław University of Technology

THANK YOU!

tomasz.trzcinski@pw.edu.pl
tomasz.trzcinski@tooploox.com