3D Object Classification with PointNet and Hybrid Point_Variational AutoEncoder

Date: 31.01.2024

Students: Mariam Hergnyan
Khadijatou Trawally

Professor: Lamberto Ballan

UNIVERSITÀ DEGLI STUDI DI PADOVA

3D object classification

N. Sedaghat, M. Zolfaghari, E. Amiri, and T. Brox, “Orientation-boosted voxel nets for 3d object recognition,”

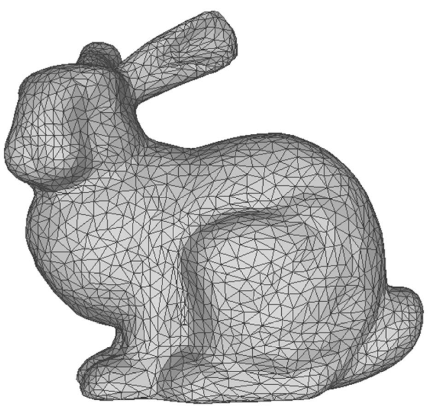
C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,”

A. Daigavane, B. Ravindran, and G. Aggarwal, “Understanding convolutions on graphs,”

Voxel

Point cloud

Polygon mesh

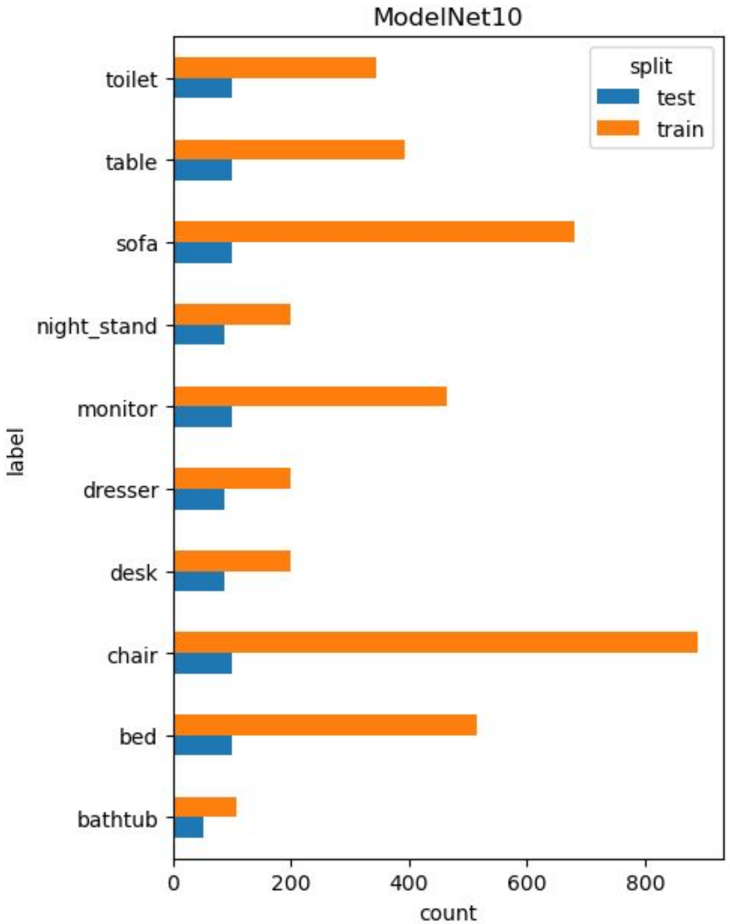


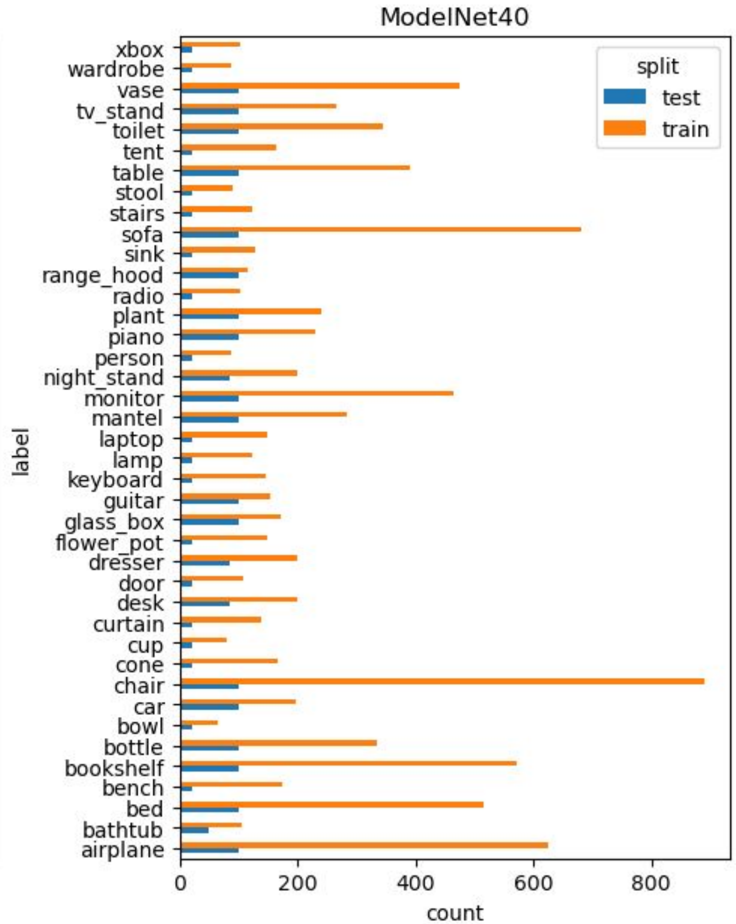
The Dataset

ModelNet40	ModelNet10
40 categories	10 categories
12,311 3D CAD models	4,899 3D CAD models

The Dataset

UNIVERSITÀ DEGLI STUDI DI PADOVA





80% - training
20% - testing

65% - training
15% - validation
20% - testing

$$V = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$$

1) Vertices Extraction:

$$\text{vertices} = \text{mesh.vertices}$$

2. Random Sampling or Poisson Disk Sampling:

   (a) `Random Sampling:`

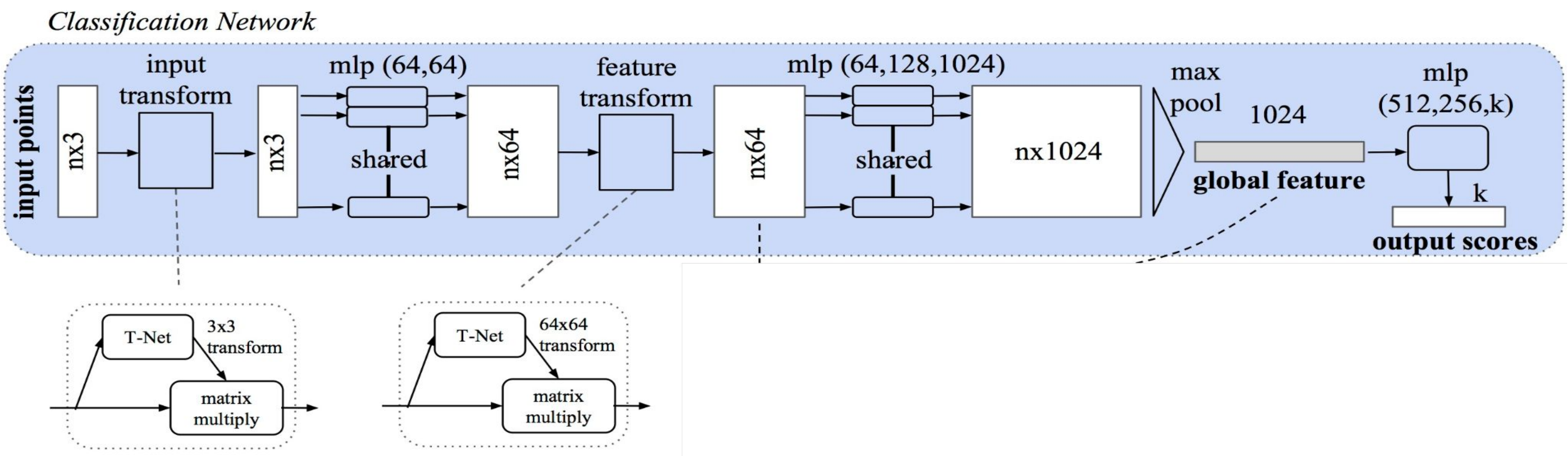
$$P = \{(x_{i_1}, y_{i_1}, z_{i_1}), \ldots, (x_{i_k}, y_{i_k}, z_{i_k})\}$$

   where i_j are randomly chosen indices.

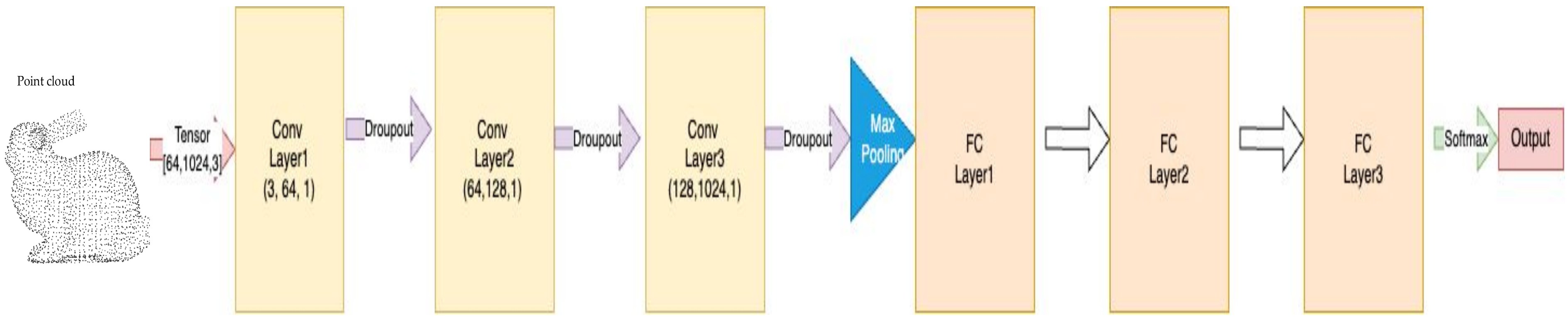
   (b) `Poisson Disk Sampling:`

- Random Jittering: [x′, y′, z′] ← [x, y, z] + noise, noise ~ N (μ, σ)
- Random rotations: [x′, y′, z′] ← R(θ)[x, y, z]
- Normalization: cubic scaling of range between 0 and 1

- Transformation Network - as a matrix of transformation
- BackBone PointNet - prediction depending on the input point clouds
    - The transformation is learned during the NN training

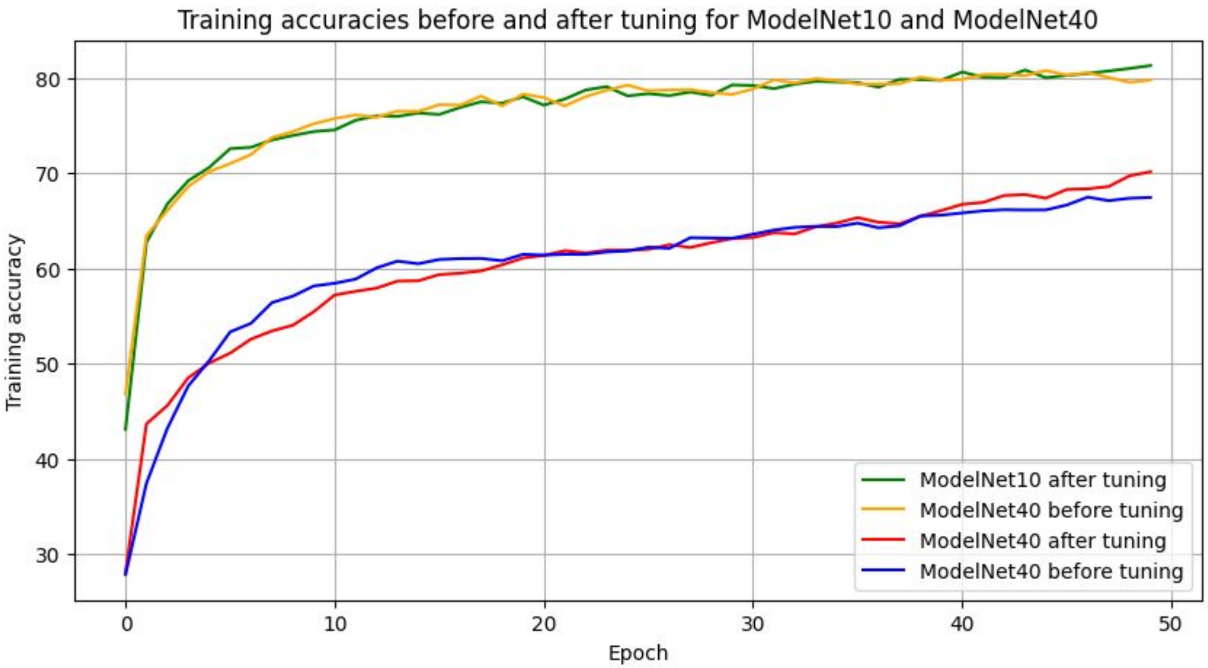


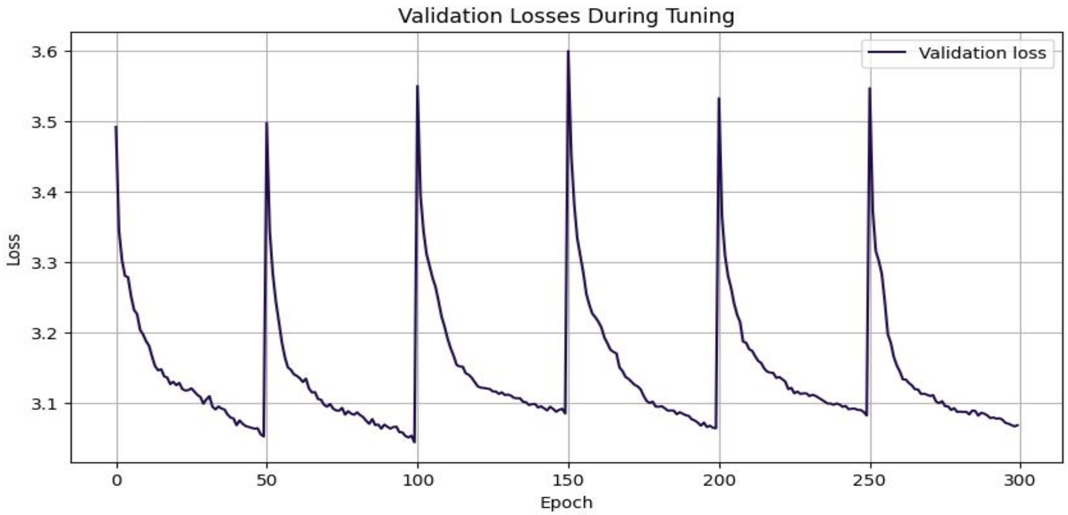
Backbone PointNet



	lr:	batchsize	Accuracy
Initial evaluation (ModelNet10)	1e-4	64	78%
Initial evaluation (ModelNet40)	1e-4	64	69%
Hyperparameter tuning (ModelNet10)	0.0002	128	82%
Hyperparameter tuning (ModelNet40)	0.0001	32	70%

Validation and Hyperparameter Tuning.





$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N} L(\text{outputs}_i, \text{labels}_i)$$

Variational Autoencoders

- the input is encoded as distribution over the latent space
- a point from the latent space is sampled from that distribution
- the sampled point is decoded

Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114

variational autoencoders	input	encoding →	latent distribution	sampling →	sampled latent representation	decoding →	input reconstruction
	x		**p(z|x)**		**z ~ p(z|x)**		**d(z)**

Qi, C. R., Su, H., Niessner, M., Dai, A., Yan, M., & Guibas, L. J. (2016). Generative and Discriminative Voxel Modeling with Convolutional Neural Networks (arXiv:1608.04236v2 [cs.CV]). arXiv preprint arXiv:1608.04236v2.

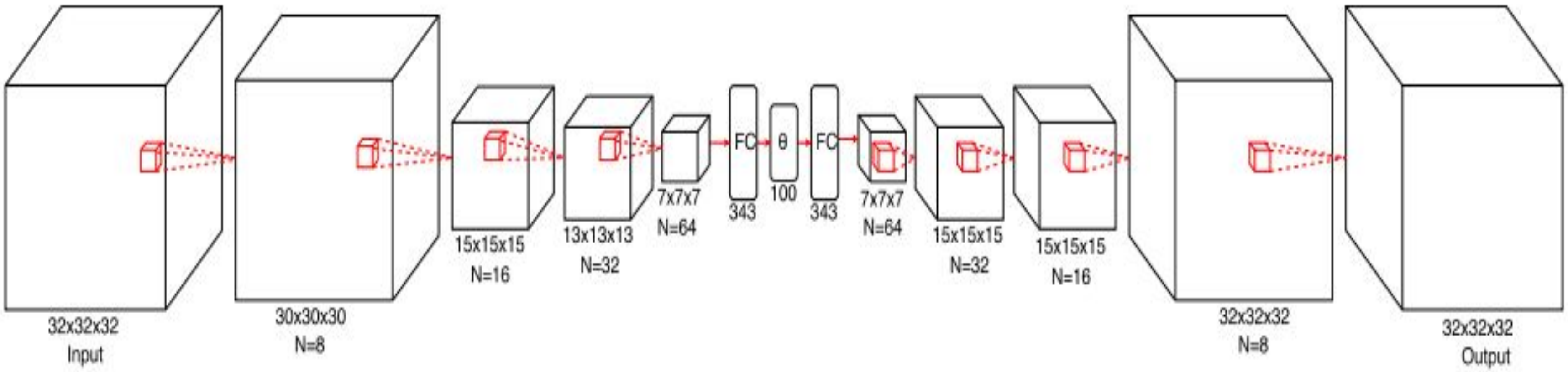


Figure 1: VAE Architecture.

The Hybrid Point-VAE combines discriminative and generative capabilities for enhanced performance

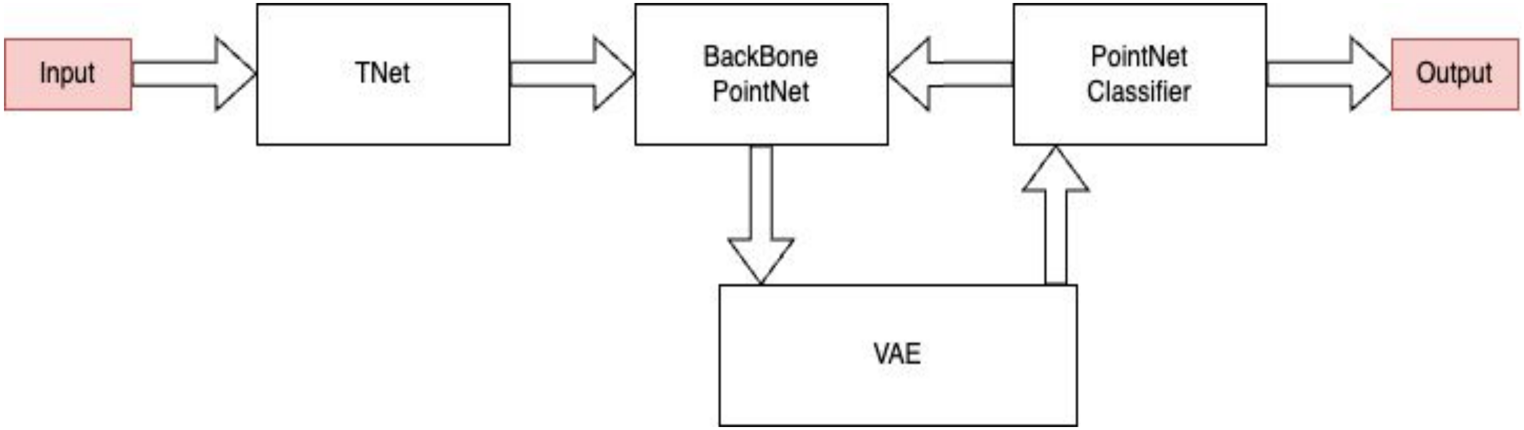
$$Z = backbone\_pointnet(X)$$

$$\mu, \log \sigma = encoder(Z)$$

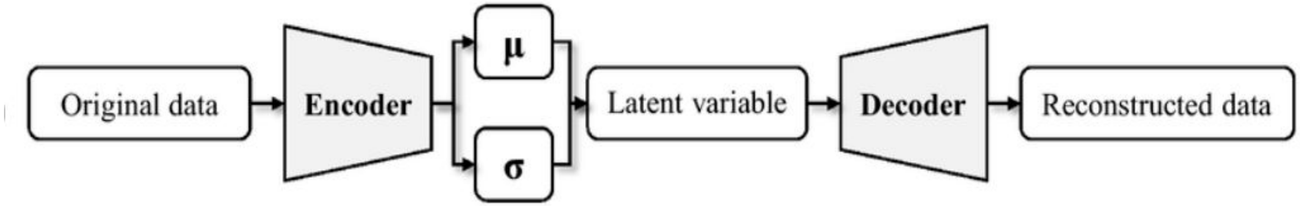
$$Z_{gen} = \mu + log\sigma \odot \varepsilon \quad where \ \varepsilon \sim N(0, 1)$$

$$X_{recon} = Decoder(Z_{gen})$$

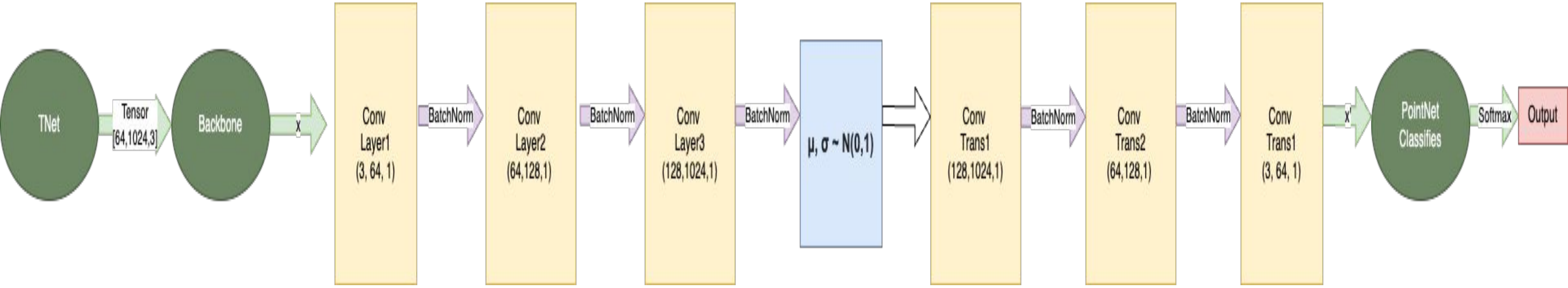
$$output\_cls = pointnet\_classifier(X_{recon})$$



Hybrid Model



Flowchart of the variational autoencoder (VAE)



Training Accuracies and Losses for the Hybrid Model

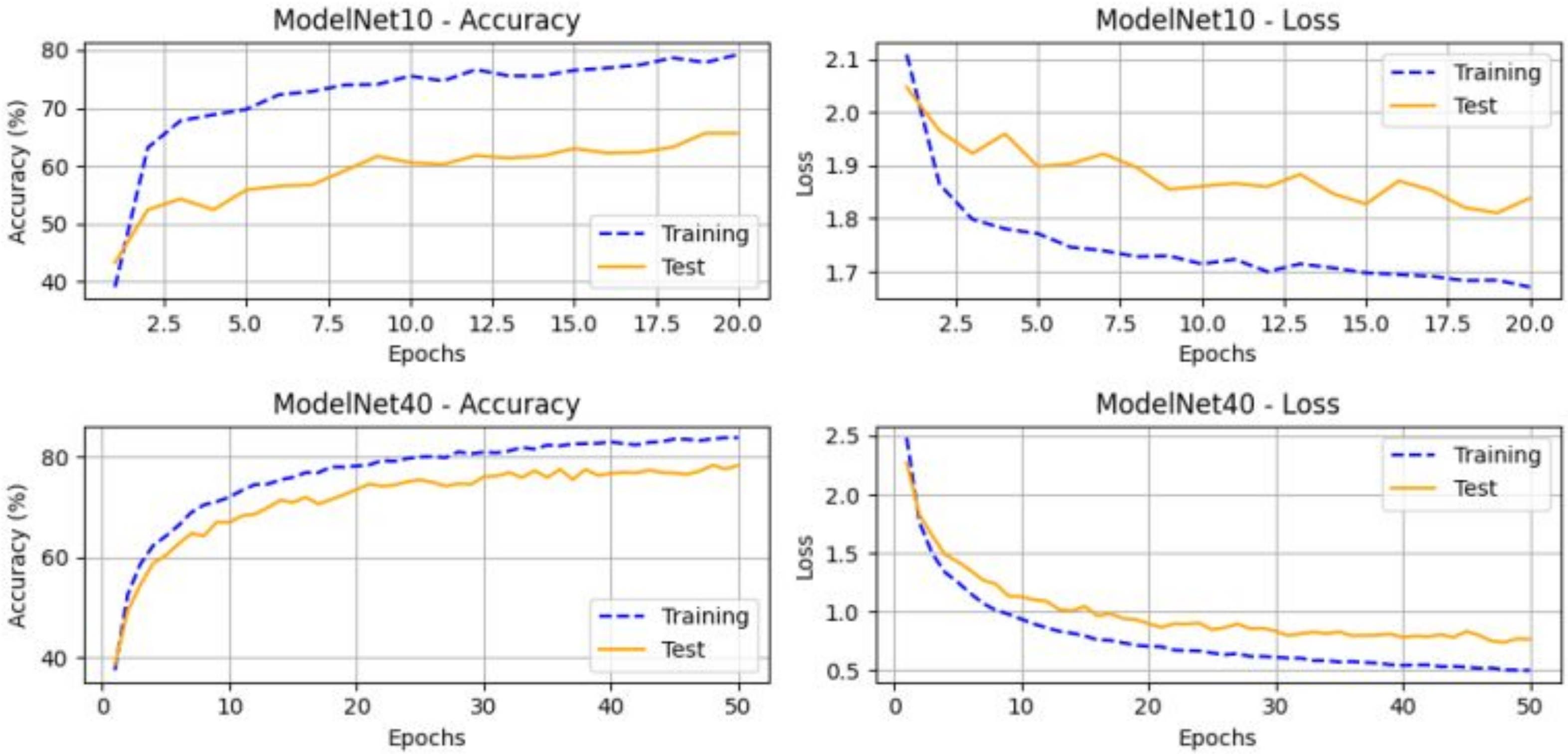


Figure 4. Training accuracies for hybrid model with ModelNet10 and ModelNet40

Architecture	Dataset	Accuracy in percentage
PointNet(initial)	ModelNet10	78
PointNet(initial)	ModelNet40	69
PointNet(tuned)	ModelNet10	82
PointNet(tuned)	ModelNet40	70
Hybrid	ModelNet10	79
Hybrid	ModelNet40	84

Table 1. Comparison of training accuracies

- Sometimes a lighter model works better
- Sometimes not…
- A smaller dataset may be easier to classify
- But again it’s not always true

Future work:

- more hyperparameter tuning(egs. k-fold cross validation, dynamic lr.)
- use a different dataset with more complex object
- potentially architectural modifications like adding the VAE loss in the total loss function

Tech Stack

PyTorch

OPEN3D

Thank you

Università degli Studi di Padova