RepMode

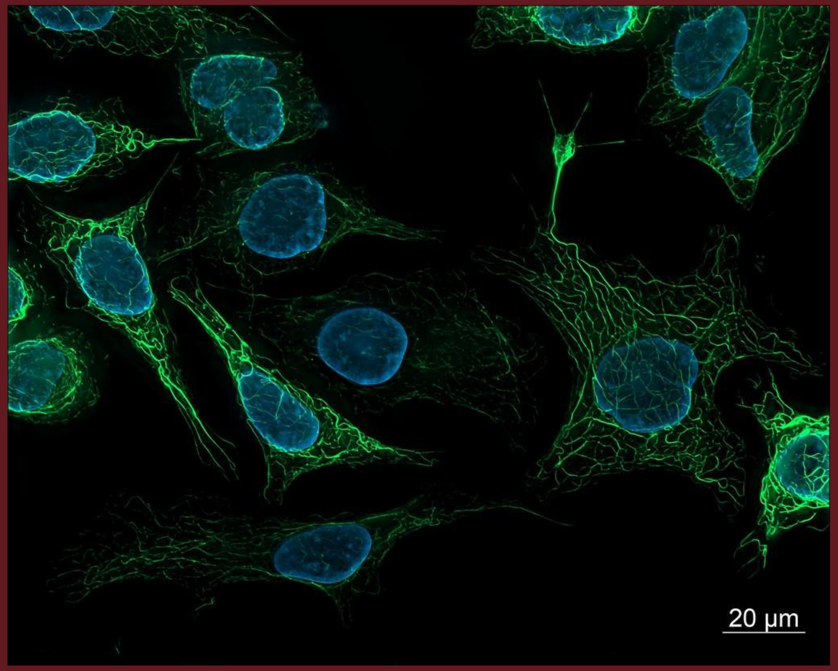
Learning to Re-parameterize Diverse Experts for Subcellular Structure Prediction [8]

Donghao Zhou, Chunbin Gu, Junde Xu, Furui Liu, Qiong Wang, Guangyong Chen, Pheng-Ann Heng

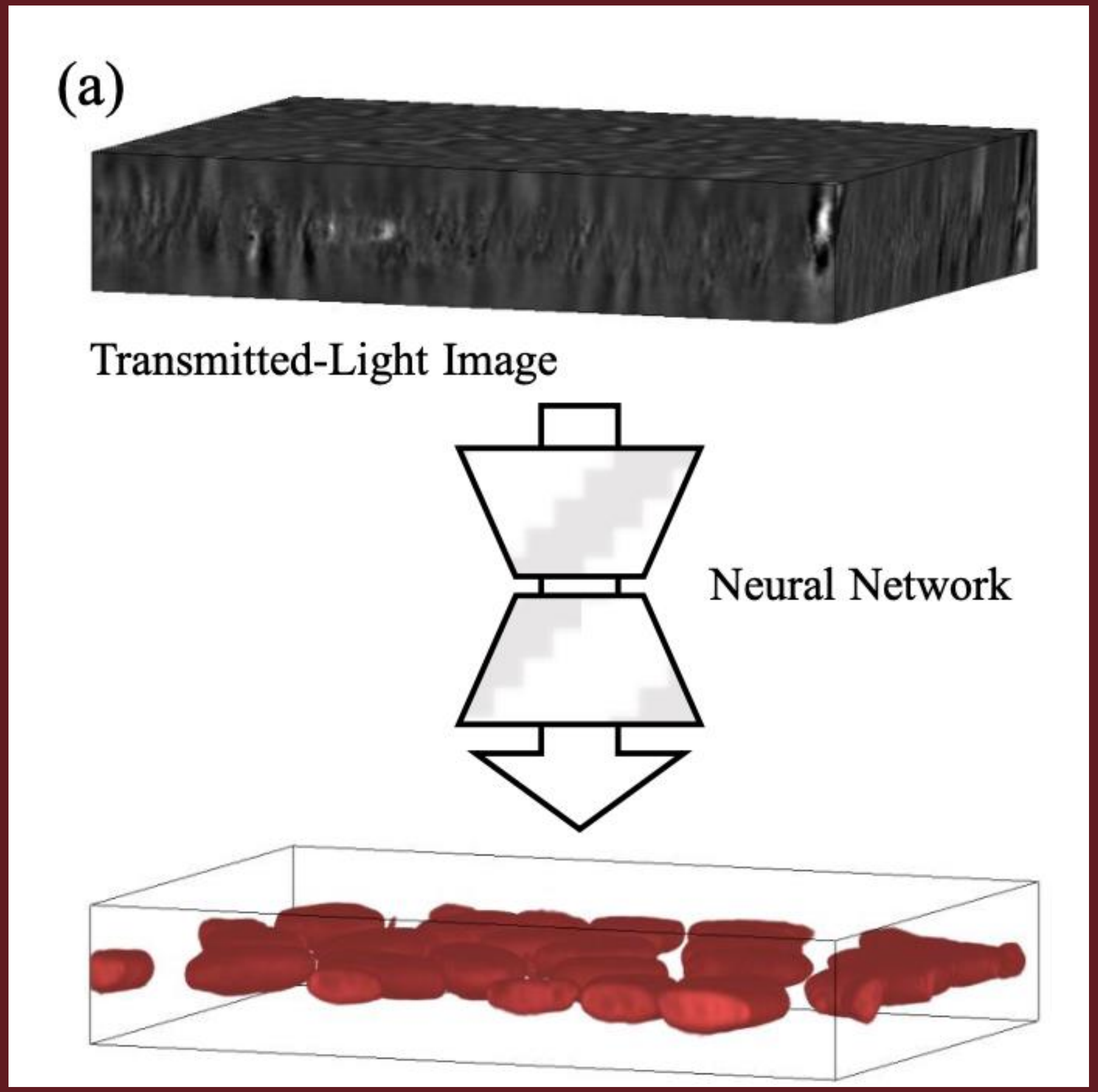
Presented by Mateo Rodriguez, Nicola Debole

Introduction

The model tackles the problem of fluorescence staining, which is a process in which subcellular structures are tainted in order to enhance their visibility.



Fluorescence staining



The model receives as input a 3D transmitted-light image and aims to get the fluorescence stained image.

The basics - tasks

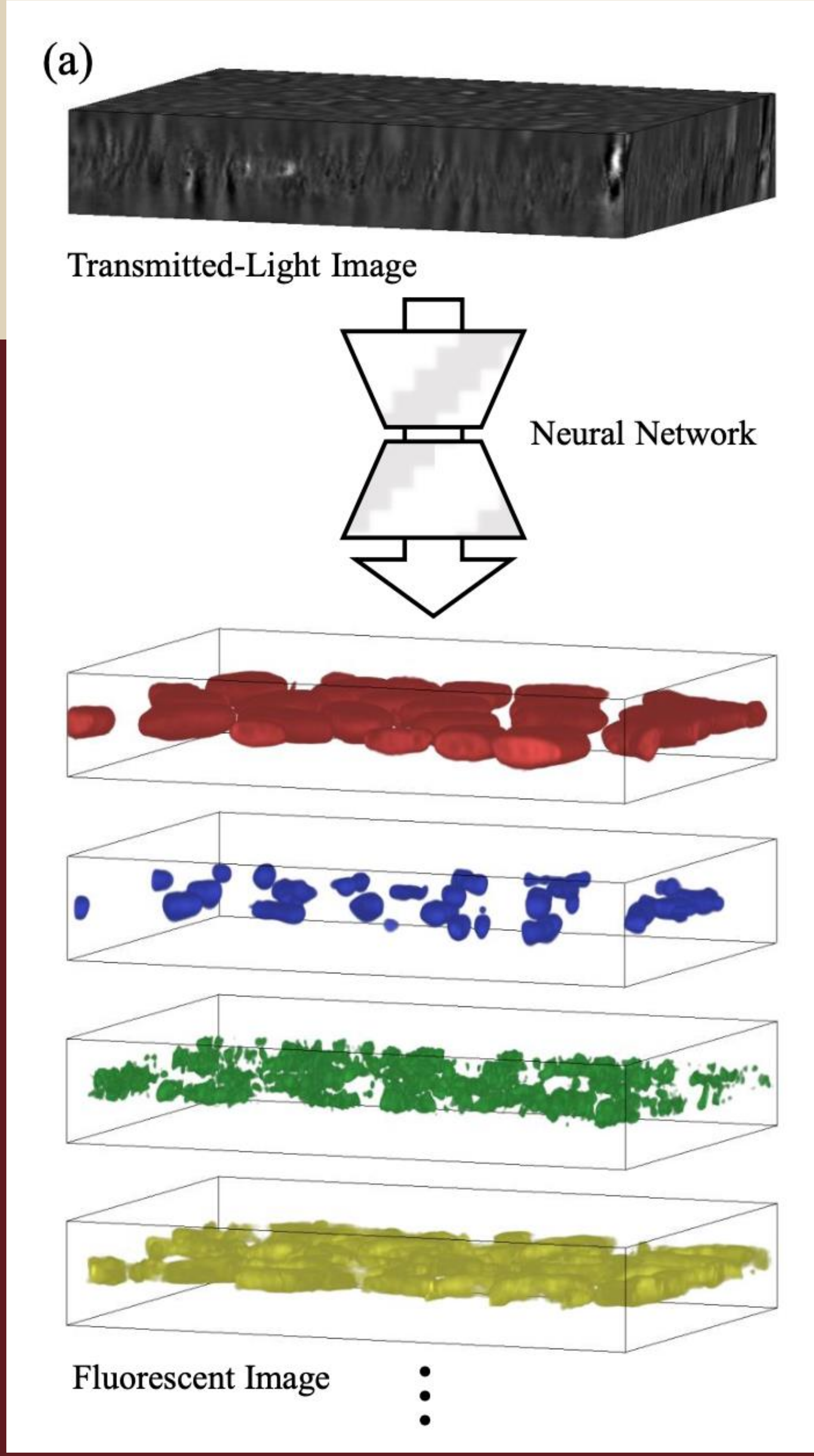
Task Awareness

We want to highlight different areas of the sample

Each subcellular structure is considered a task

The transmitted-light image scan of the sample contains many different structures

The model needs to use different scales



MoDE block

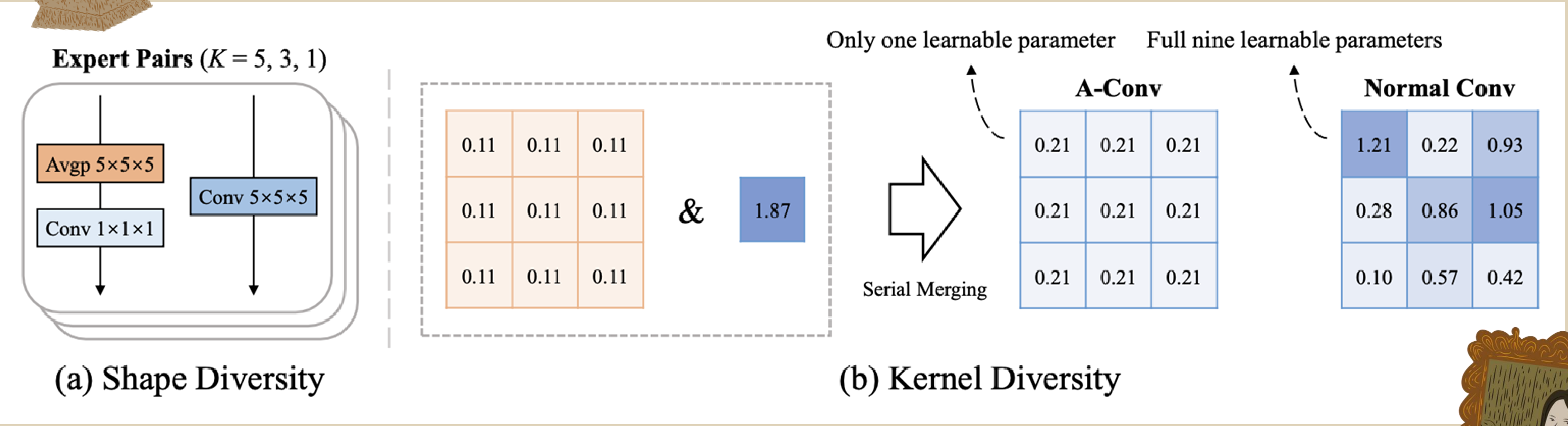
The network employs a block called **Mixture of Diverse Experts (MoDE)**.

IDEA BEHIND: each expert is an **independent learner**, whose output is later combined with the other experts' output in order to obtain a prediction. Different combinations of experts **can solve different tasks**.

MAIN ADVANTAGE: ability to compress information from different tasks into a single block.

Experts

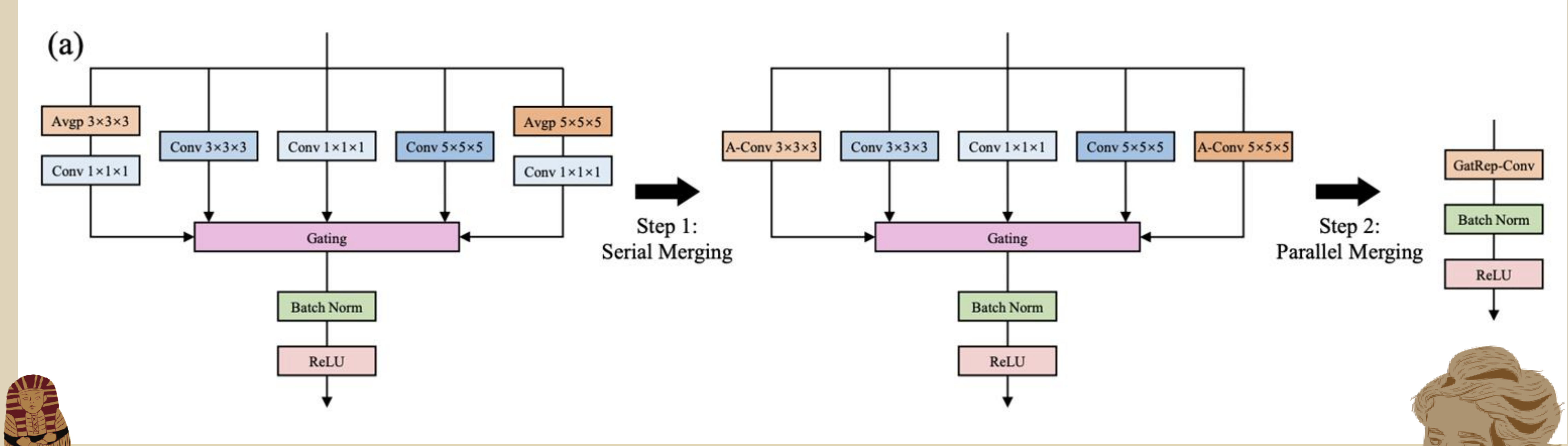
TLDR: The experts are just different convolutions.



A-Conv: 1 parameter
Normal Conv: K x K parameters

Experts

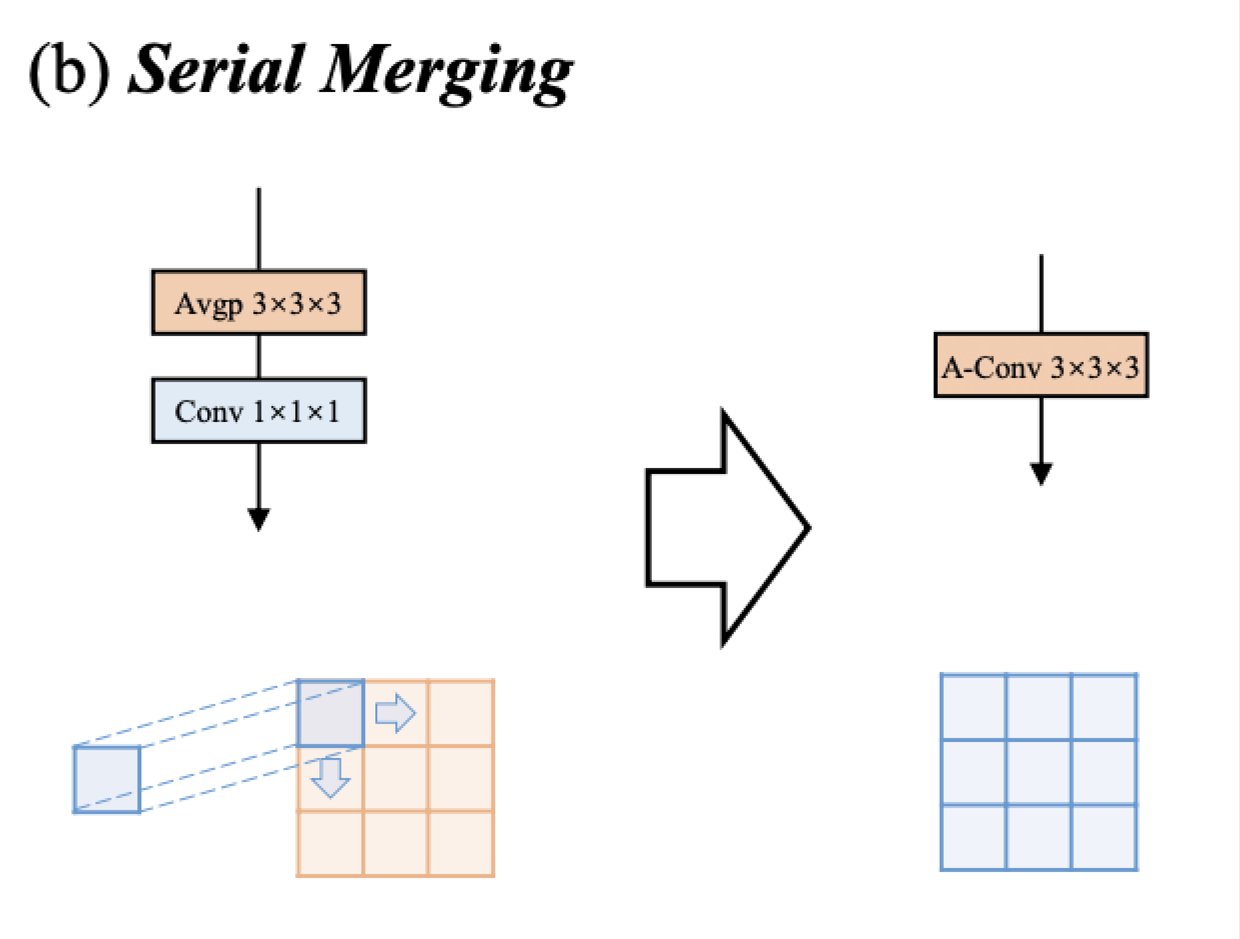
Each MoDE block combines the output of the five experts.

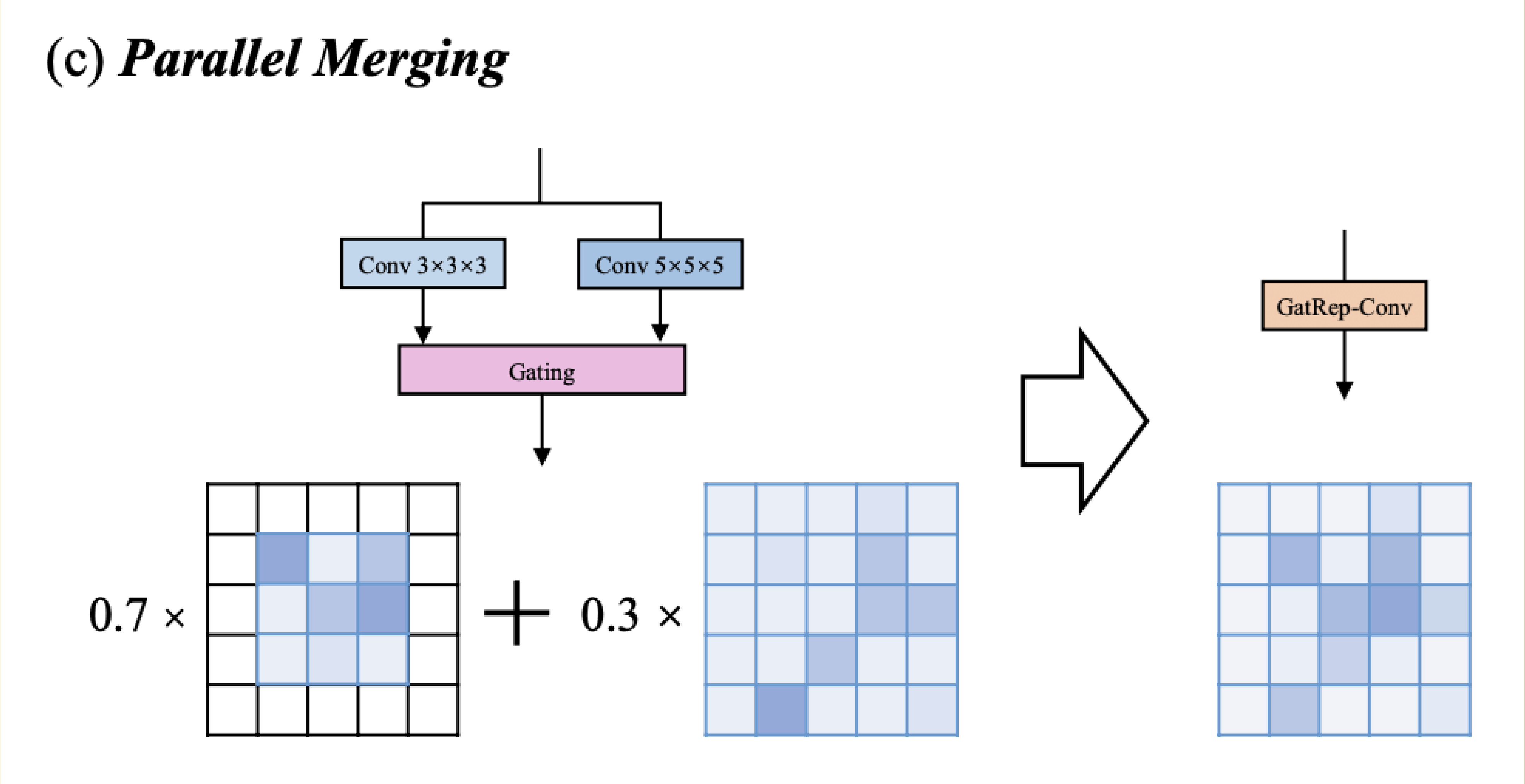


Mixture of Diverse Experts: how to obtain the final convolutional kernel?

Experts

Two merging operations are needed:

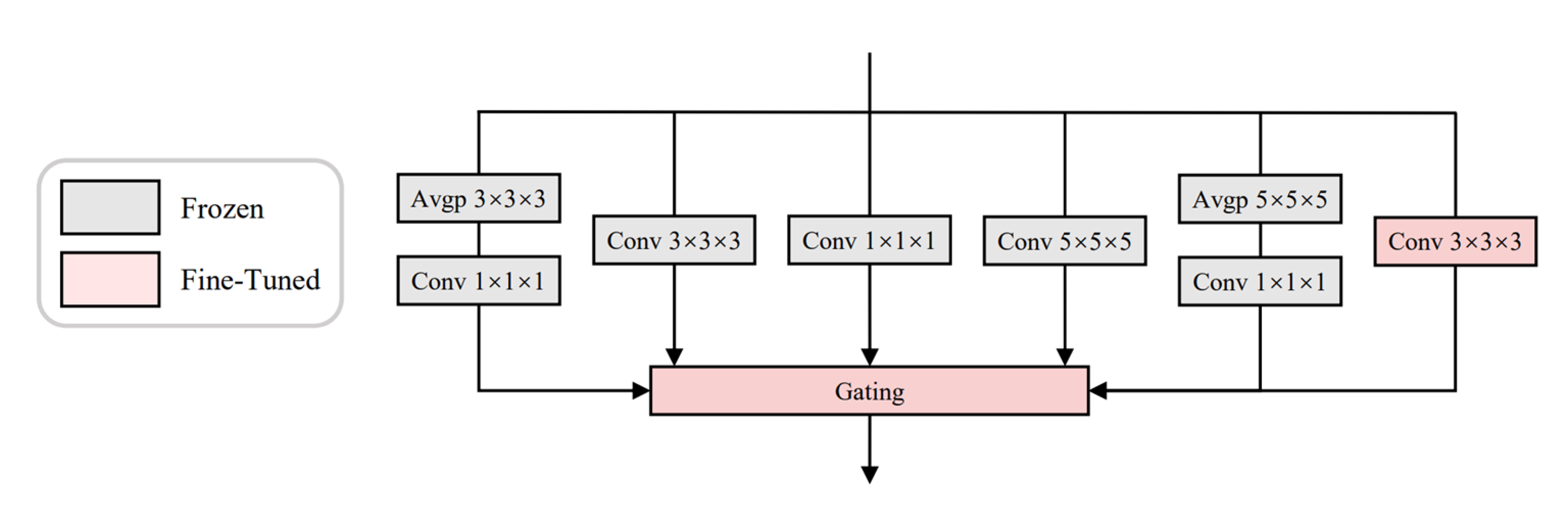




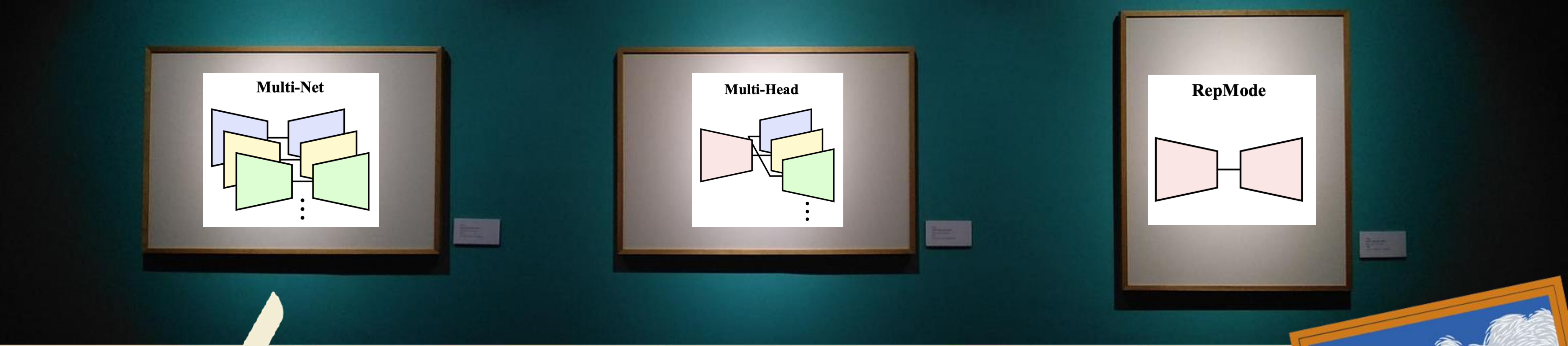
To generate the A-Conv kernel by applying the 2 operations in series

To merge the 5 experts in just one single convolution

Experts



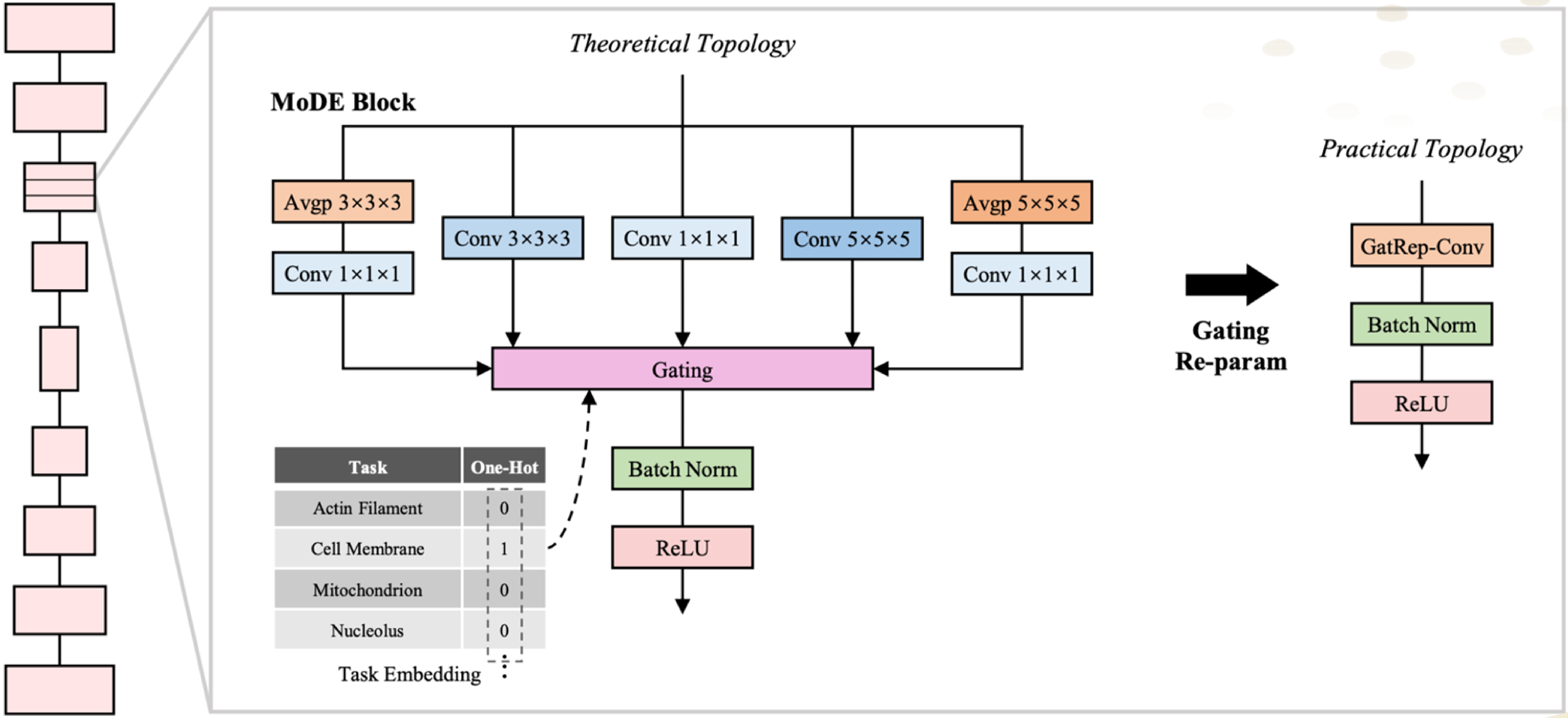
For task incremental learning the model can be retrained with an added expert



Benefits

- Efficient
- Continual Learning (prevents Catastrophic Forgetting)
- End-to-End
- Every sample contributes to the training (= for all the tasks)

Architecture

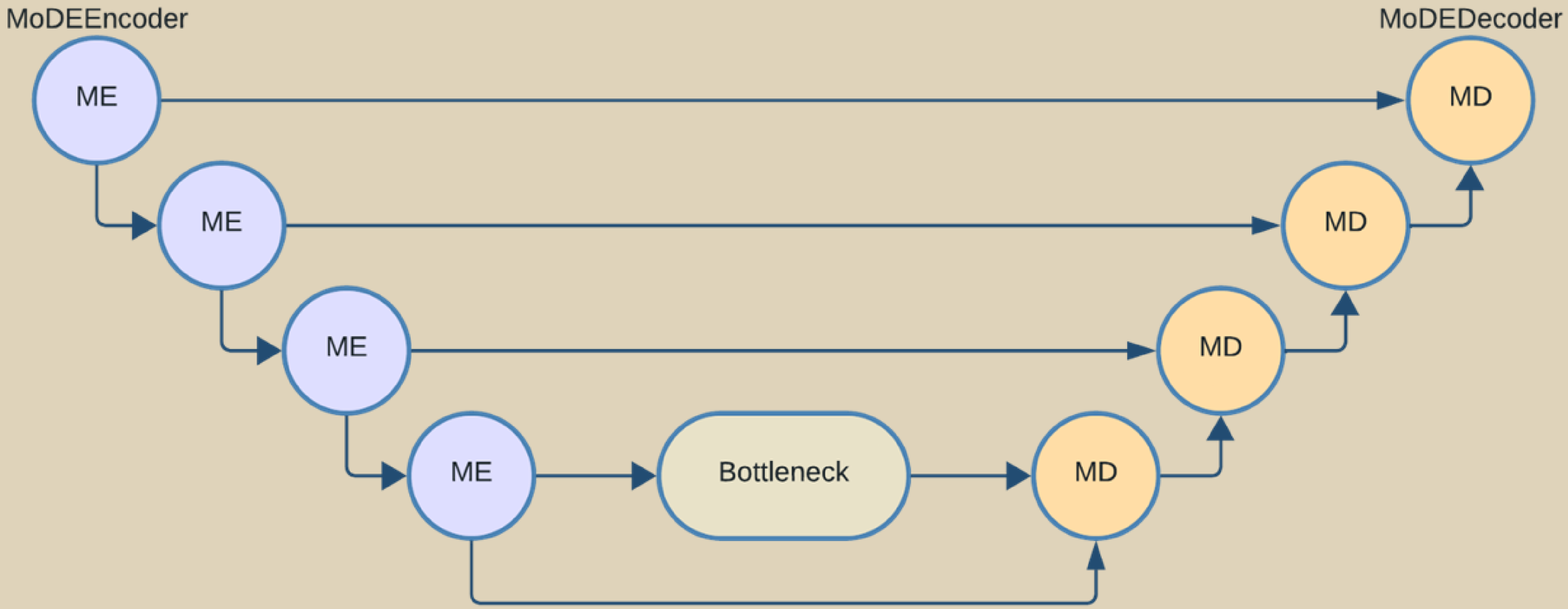


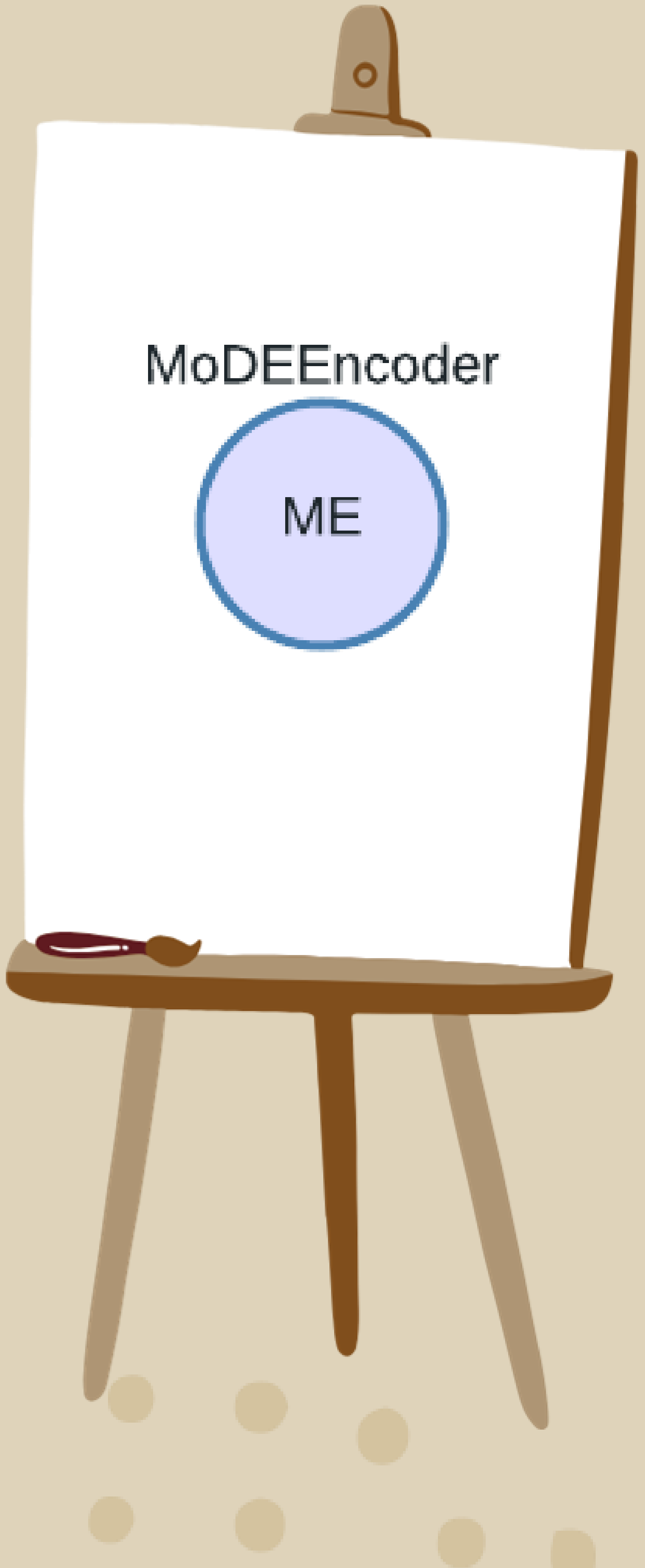
(b) RepMode

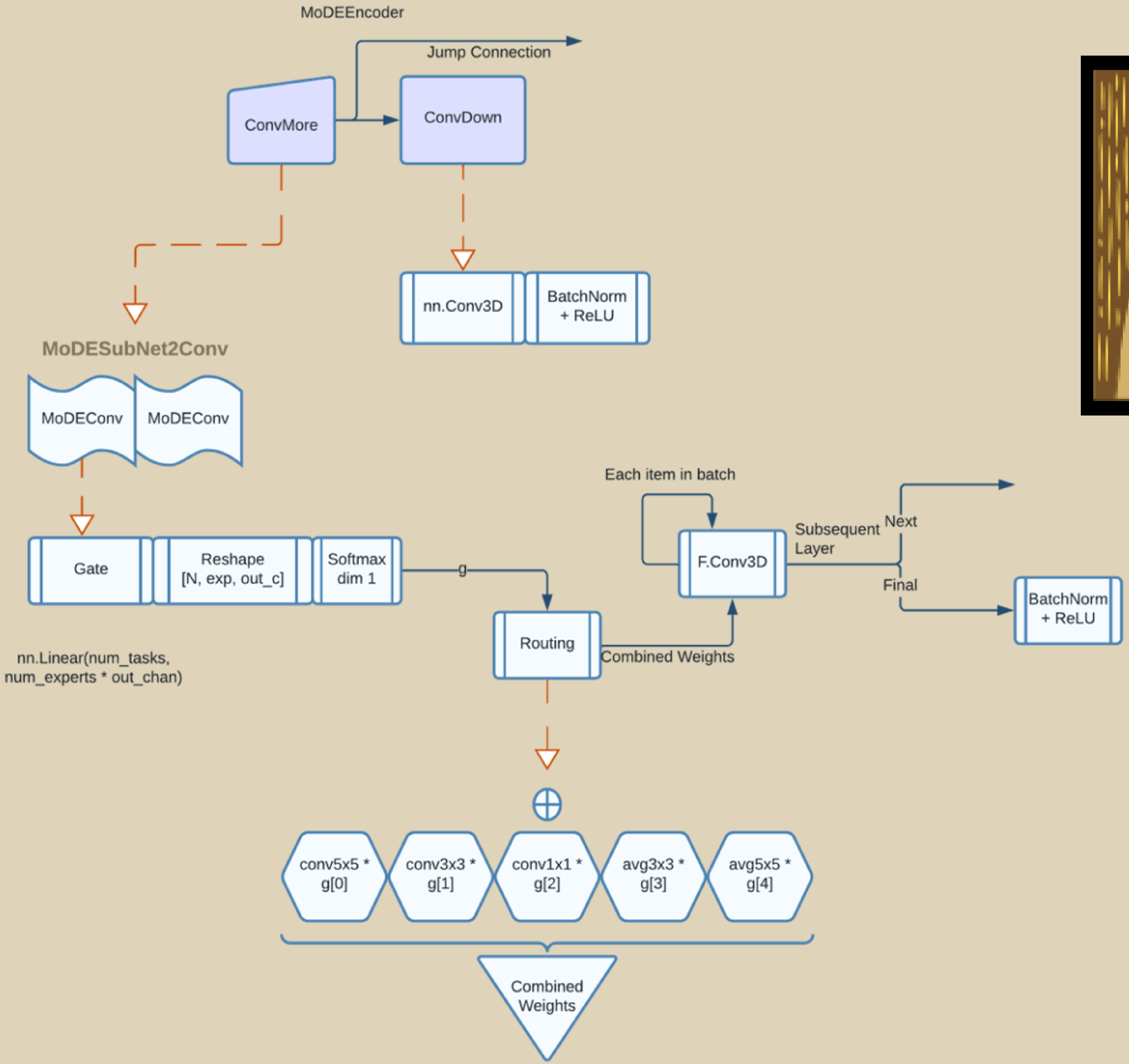
Architecture

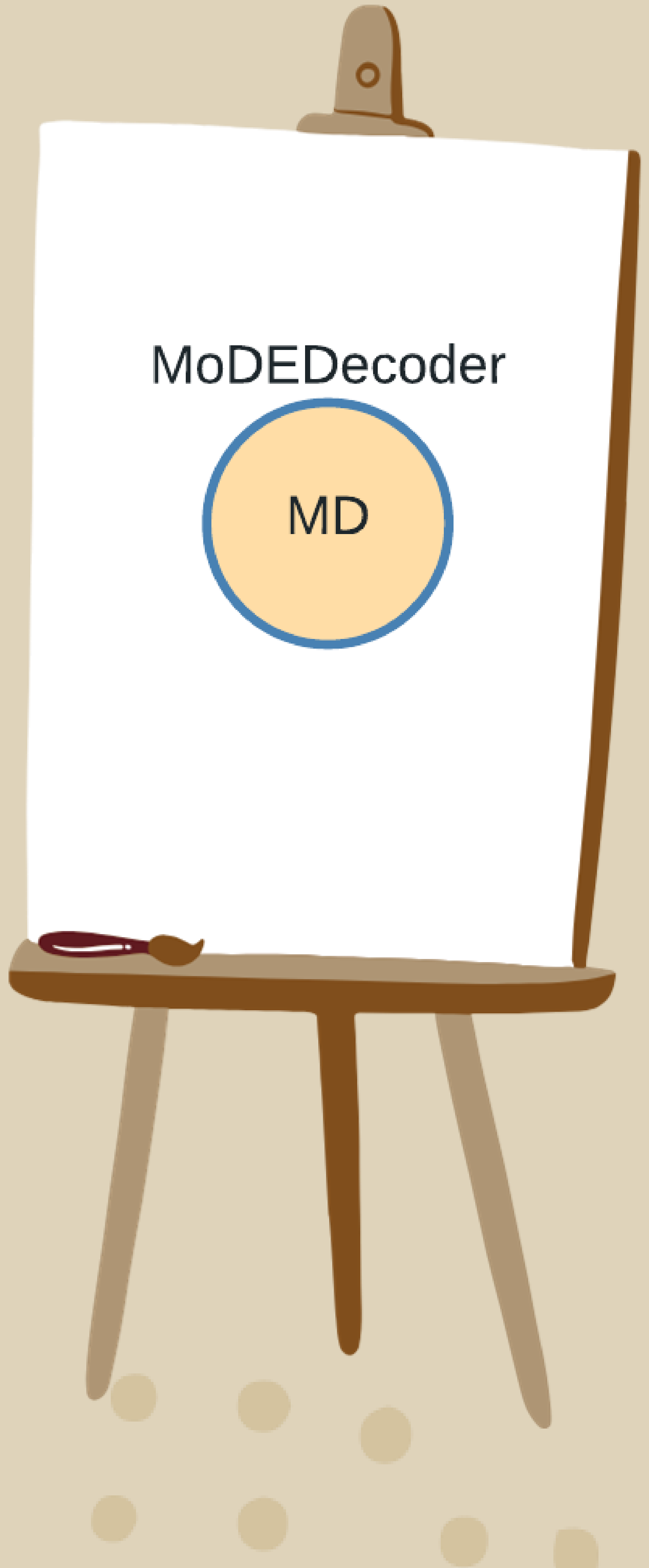
Tensor shapes along the network:

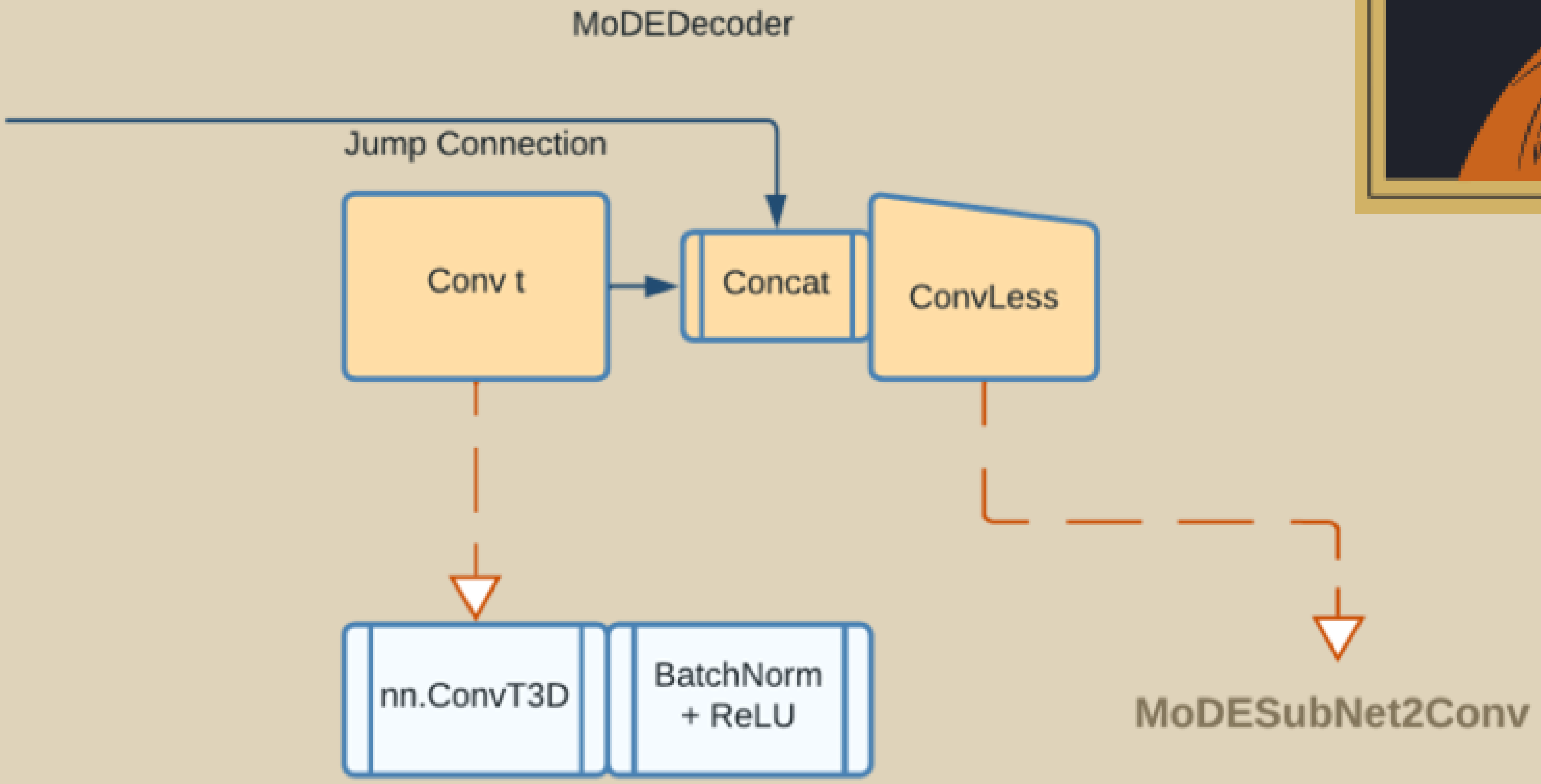
I. [1,64,64,64]
II. [16,32,32,32]
III. [32,16,16,16]
IV. [64,8,8,8]
V. [128,4,4,4]
VI. [256,4,4,4]
VII. [128,8,8,8]
VIII. [64,16,16,16]
IX. [32,32,32,32]
X. [16,64,64,64]
XI. [1,64,64,64]

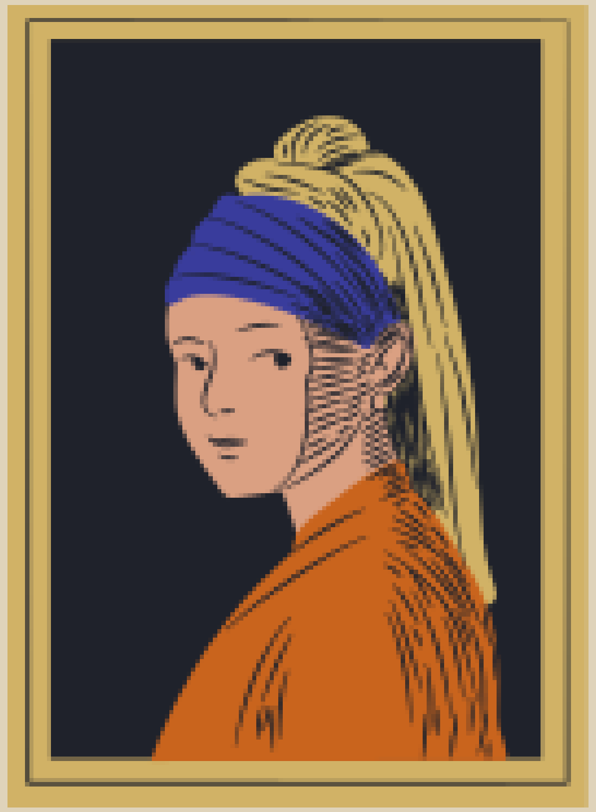



MoDEEncoder
ME


MoDEEncoder
Jump Connection
ConvMore
ConvDown
nn.Conv3D
BatchNorm + ReLU
MoDESubNet2Conv
MoDEConv
MoDEConv
Gate
Reshape [N, exp, out_c]
Softmax dim 1
g
nn.Linear(num_tasks, num_experts * out_chan)
Each item in batch
F.Conv3D
Subsequent Layer
Next
Final
BatchNorm + ReLU
Routing
Combined Weights
conv5x5 * g[0]
conv3x3 * g[1]
conv1x1 * g[2]
avg3x3 * g[3]
avg5x5 * g[4]
Combined Weights


MoDEDecoder
MD


MoDEDecoder
Jump Connection
Conv t
Concat
ConvLess
nn.ConvT3D
BatchNorm
+ ReLU
MoDESubNet2Conv



Extension

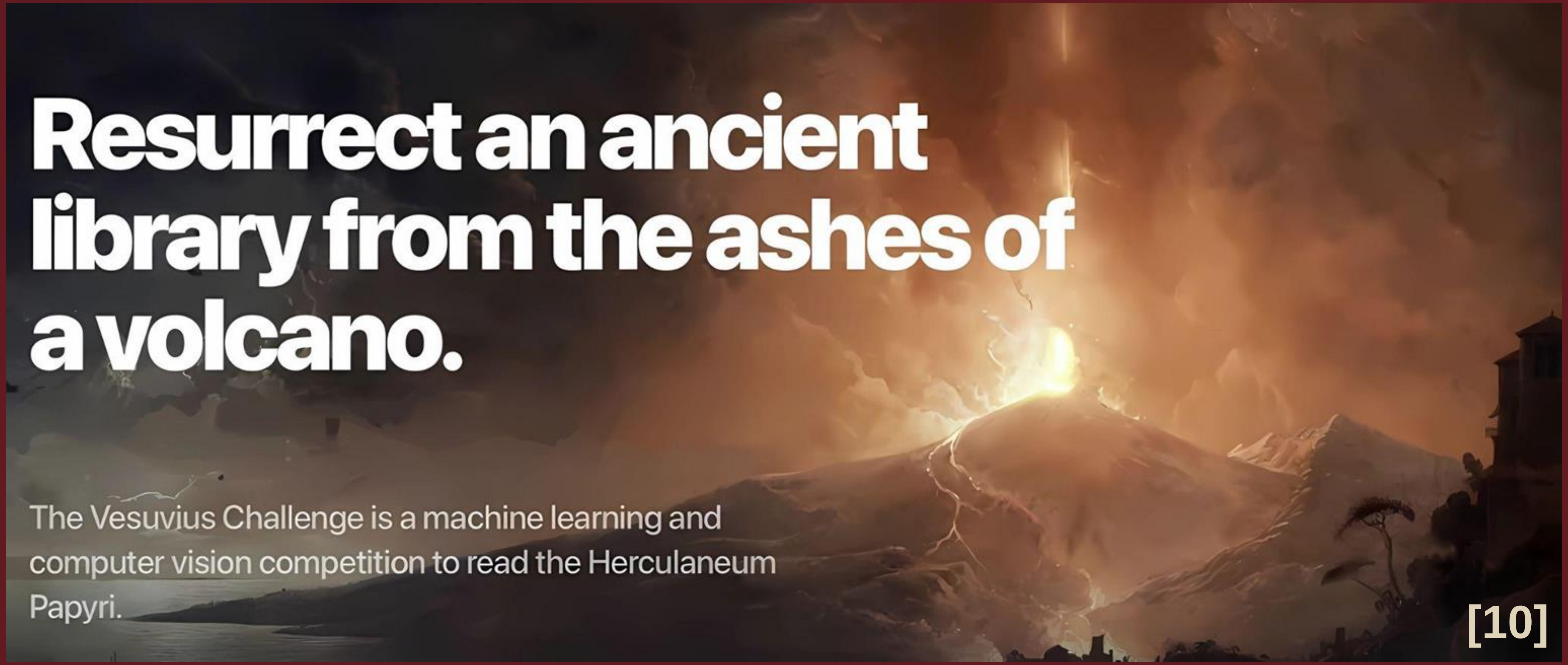
Can this model be used for anything else?
How can it be improved?

Well...

...we need something that

Has few labeled examples	Has various tasks	Is computationally expensive	3D data

Vesuvius Challenge

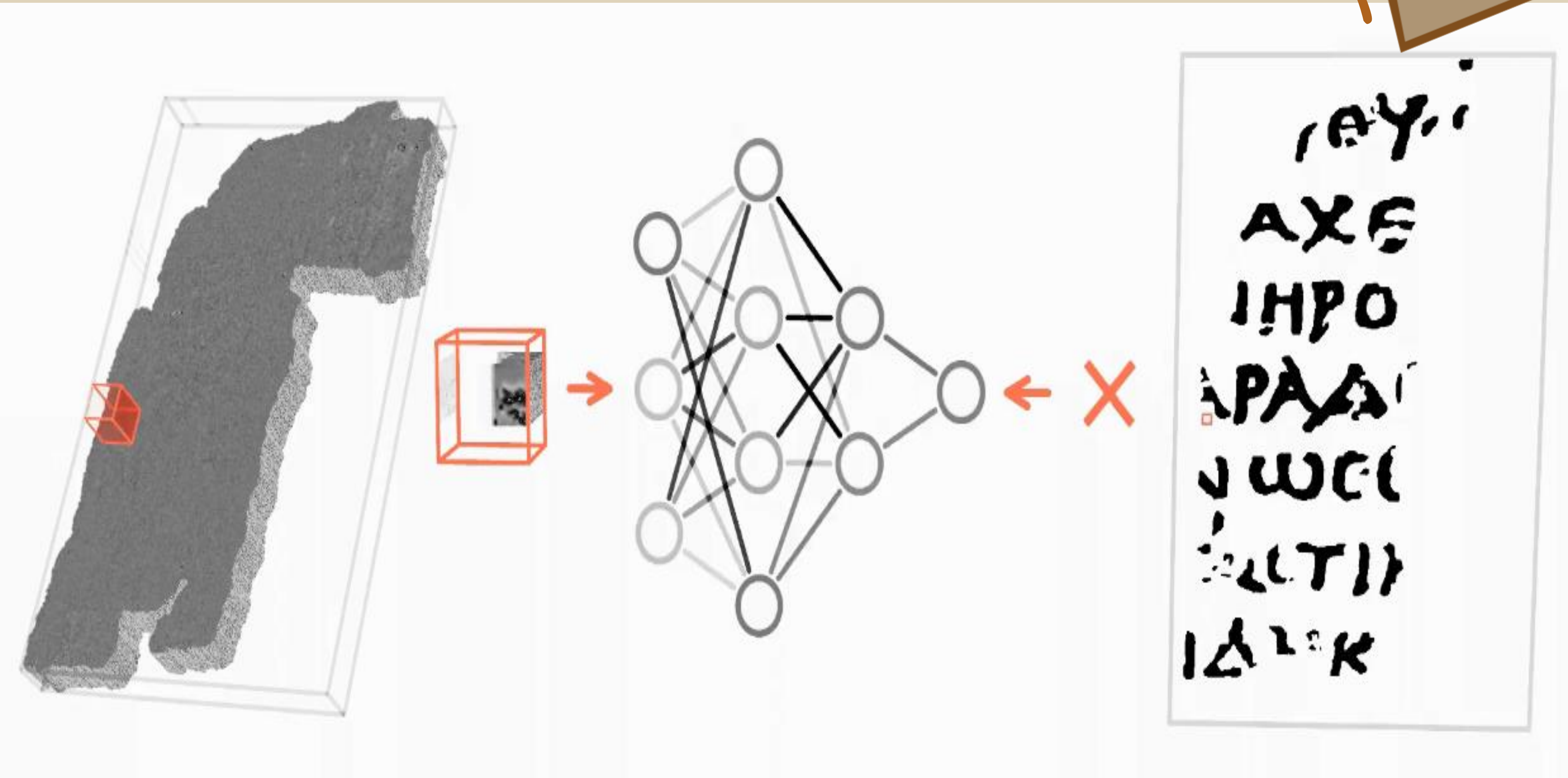


Has few labeled examples	Has various tasks	Is computationally expensive	3D data
✓	✗	✓	✓

The task

Unwrapping

Ink Detection



Modifications

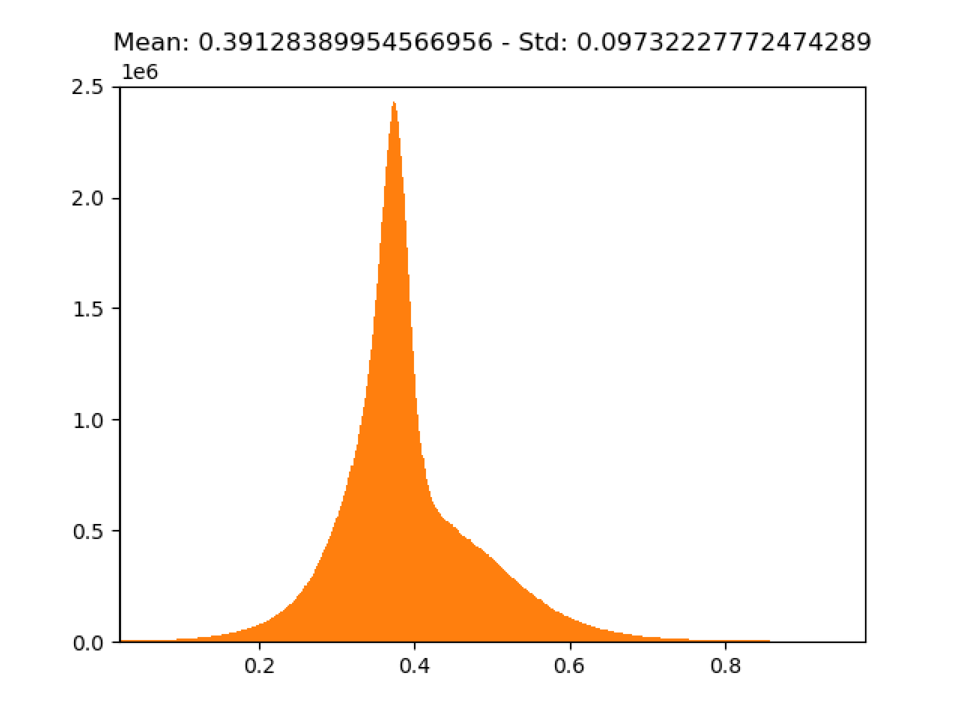
Updating the model from [7]:

- FReLU [2] and Mish [6]
- AdamW with LR scheduler [3]
- DiceLoss, softBCE [9]
- 2D image extraction

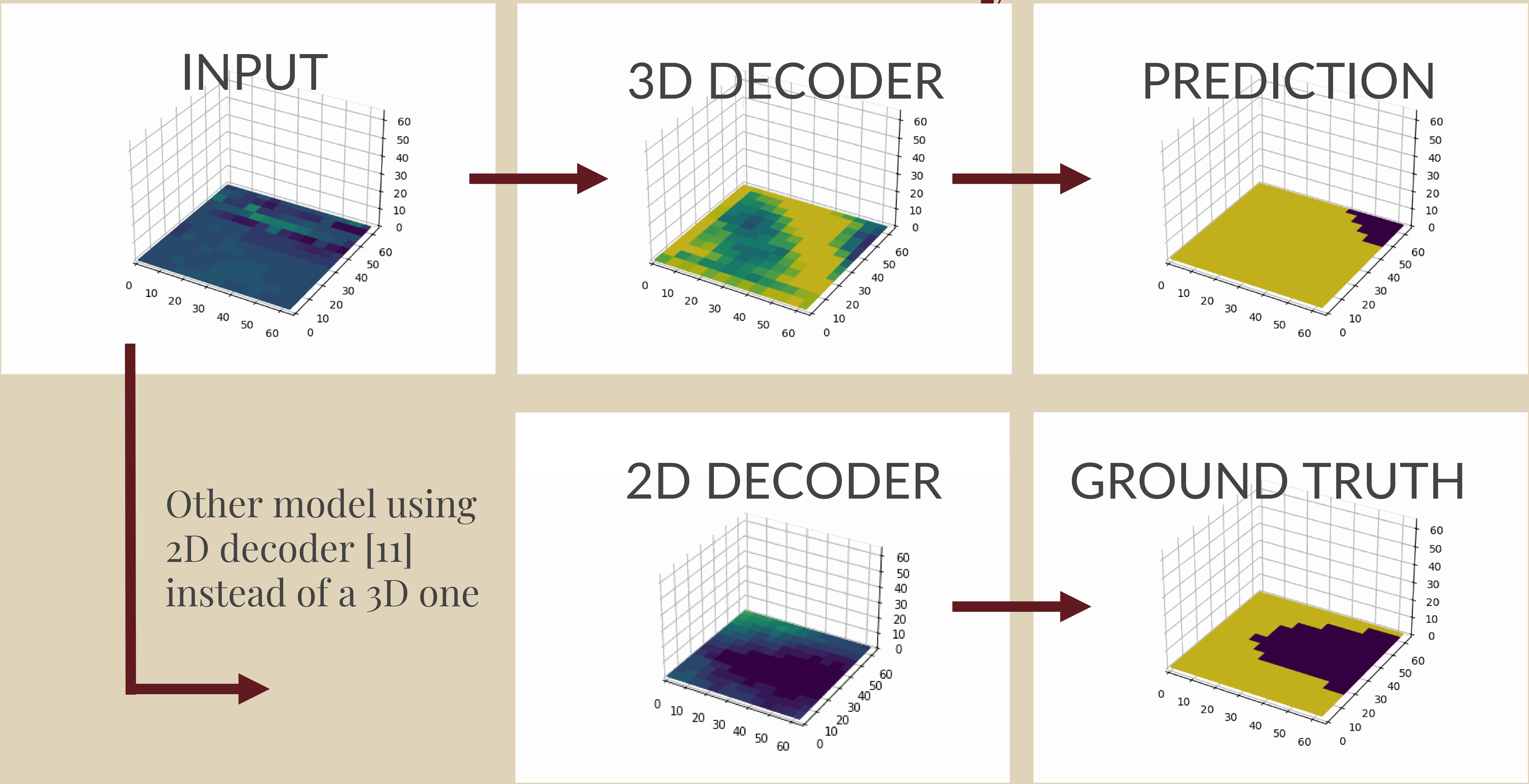
Data Augmentation (Analysis)

Data Analysis

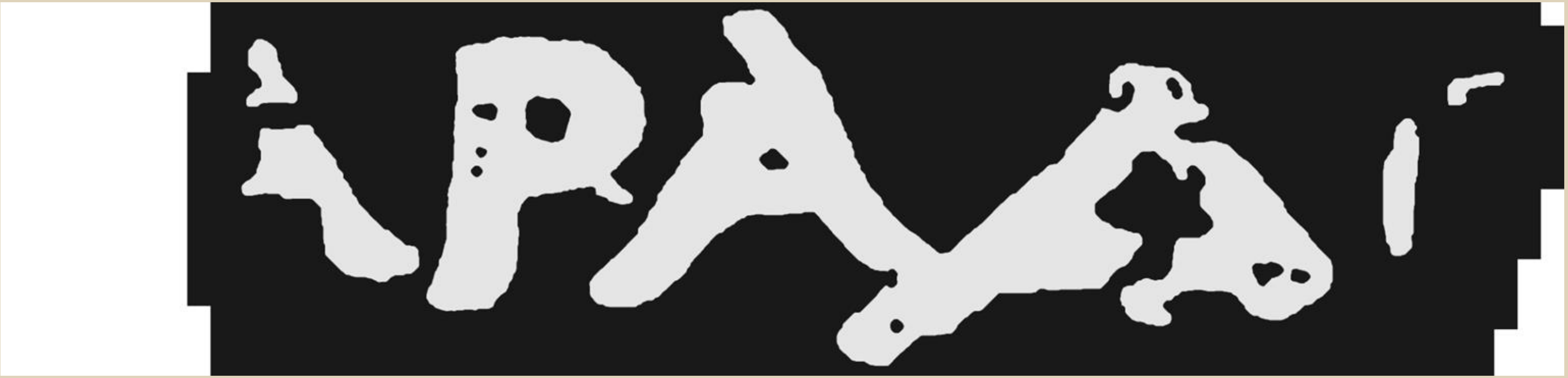
The data seems to have the same distribution despite the presence or absence of ink.

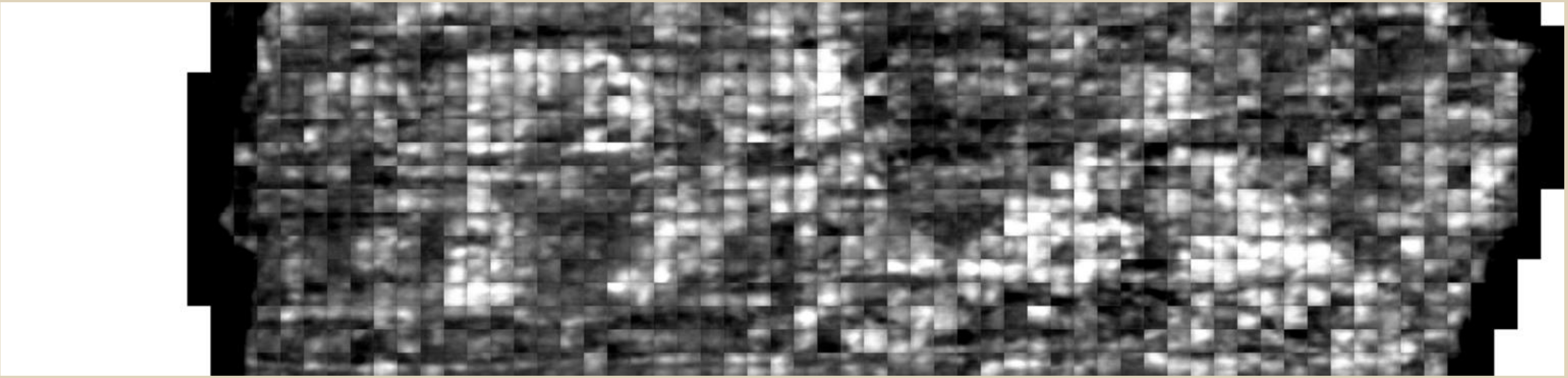


Network Analysis



Results Analysis



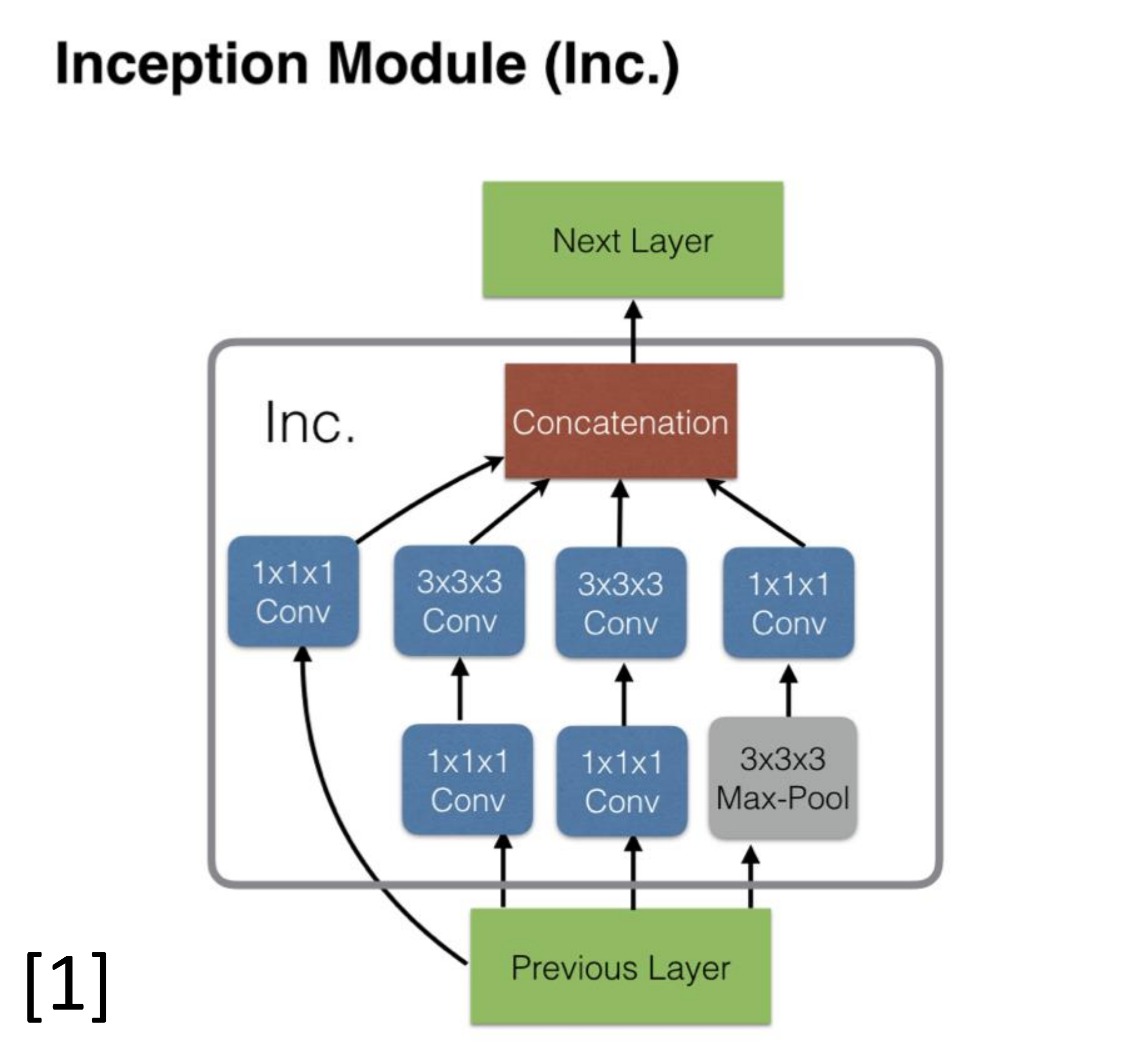


Other’s Approach

Inception architecture

Used for **ACTION RECOGNITION** like we saw in class

but without the action prediction part.



Comparison:

Higher expressing capabilities

Way more expensive

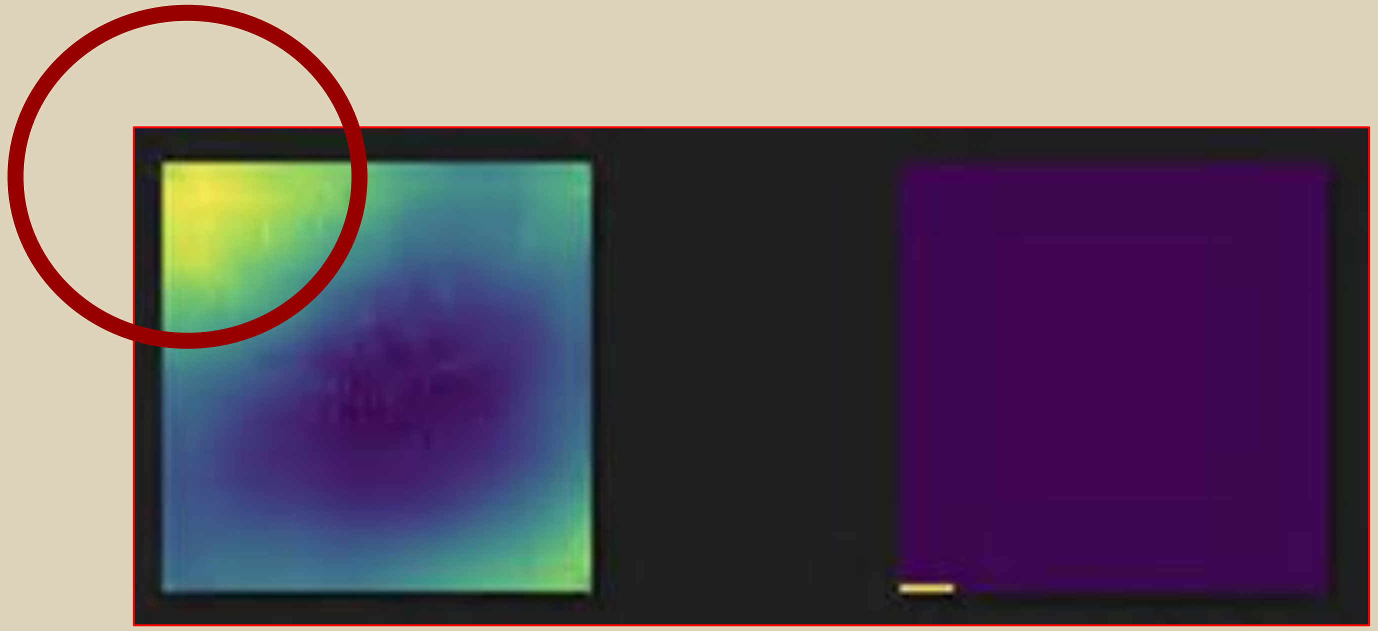
Z - Y - X

256 batch size 256 x 64 x 64

7 batch size 64 x 64 x 64
(ours 12GB vram, 16GB ram)

Model Uncertainty

Sometimes the model doesn't know it doesn't know...



Prediction | Ground truth

Improvement idea:

Monte Carlo Dropout

→ **more accurate measure of uncertainty in the model's predictions**

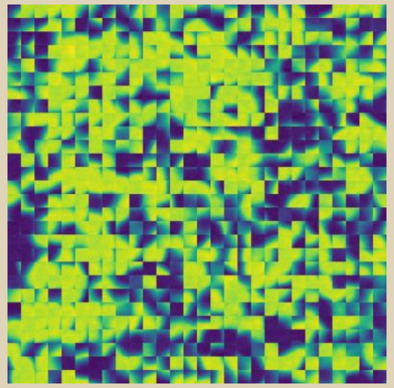
→ **a way to make a deep NN more "bayesian"**

→ **achieved by sampling models and average them**

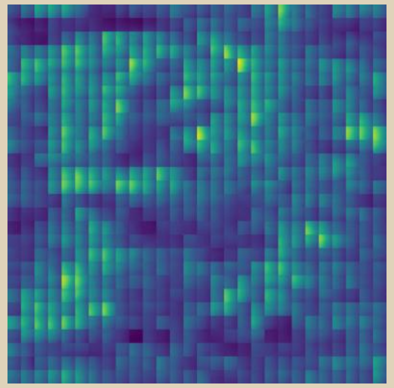
1. Train the network.
2. Extract the gating parameters, as well as the different expert kernels.
3. Add dropout on the input before each MoDE block (with dropout active also during evaluation). Also add random noise to the gating.
4. For inference, compute the predictions many times.
5. Average the results for each model.

Model Uncertainty

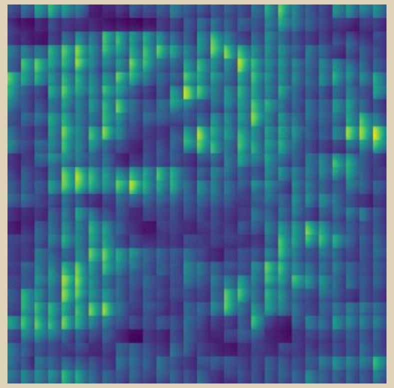
Single Prediction



Average from 60 models



Average from 360 models



Results

MODEL	LOSS ↓ average(BCE,DL)	DESCRIPTION
RepMode with 2D module	0,4513	Using all RAW training data, NO DO, NO augmentations
RepMode 2	0,377	Data normalized,custom dataset, augmentations
RepMode Buckets	0,2639	Data normalized, custom dataset, NO augmentations
RepMode Buckets v2	**0,2539**	Data normalized, custom dataset, NO augmentations, DO
2D decoder 16xRepMode	**0.52**	No normalization, custom dataset, augmentations, NO DO

- No official metrics, no benchmarks available. Evaluation is mostly done by human experts on some unlabeled sections.
- It is a challenge, so almost every team is secretly evaluating the models without sharing the results.
- Similarity metrics[13] don’t reflect the actual performance of the network

That's it!

Thank you for the attention

For more in depth information regarding the project refer to [12].

References

- [1] Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset https://arxiv.org/pdf/1705.07750v1.pdf
- [2] FReLU: Flexible Rectified Linear Units for Improving Convolutional Neural Networks https://arxiv.org/pdf/1706.08098.pdf
- [3] Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour https://arxiv.org/abs/1706.02677
- [4] Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning https://arxiv.org/pdf/1506.02142.pdf?
- [5] Dropout: A Simple Way to Prevent Neural Networks from Overfitting https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf
- [6] Mish: A Self Regularized Non-Monotonic Activation Function https://arxiv.org/pdf/1908.08681.pdf
- [7 - CODE] https://github.com/Correr-Zhou/RepMode
- [8] RepMode https://arxiv.org/pdf/2212.10066.pdf
- [9] Segmentation Models Pytorch - Losses https://smp.readthedocs.io/en/latest/losses.html
- [10] Vesuvius Challenge https://scrollprize.org/
- [11] Lossy image compression with compressive autoencoders https://arxiv.org/pdf/1703.00395.pdf
- [12] Our In-Depth version https://it.overleaf.com/read/sfshcpkrktnm#c38bff
- [13] Image Quality Assessment: From Error Visibility to Structural Similarity wang03-reprint.pdf (nyu.edu)

Papers

- Loss Functions for Image Restoration with Neural Networks https://arxiv.org/pdf/1511.08861.pdf
- Gradual warmup
- ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks https://arxiv.org/pdf/1809.00219.pdf
- FRELU
- Bayesian NN dropout
- Inception https://arxiv.org/pdf/1705.07750v1.pdf
- Masked Autoencoders Are Scalable Vision Learners https://arxiv.org/pdf/2111.06377.pdf

Resource Page

*** Don't forget to delete this page before presenting.

Use these design resources in your Canva Presentation.

Use these design resources in your Canva Presentation.

Titles: Playfair Display Black
Headers: Lato
Body Copy: Lato

You can find these fonts online too. Happy designing!