ViT Saliency Prediction

How far can transformers be pushed?

Ignacy Alwasiak
Kacper Marzol
Dominik Matuszek
Kiryl Vinahradau

Introduction

Attention rollout

Rollout prediction

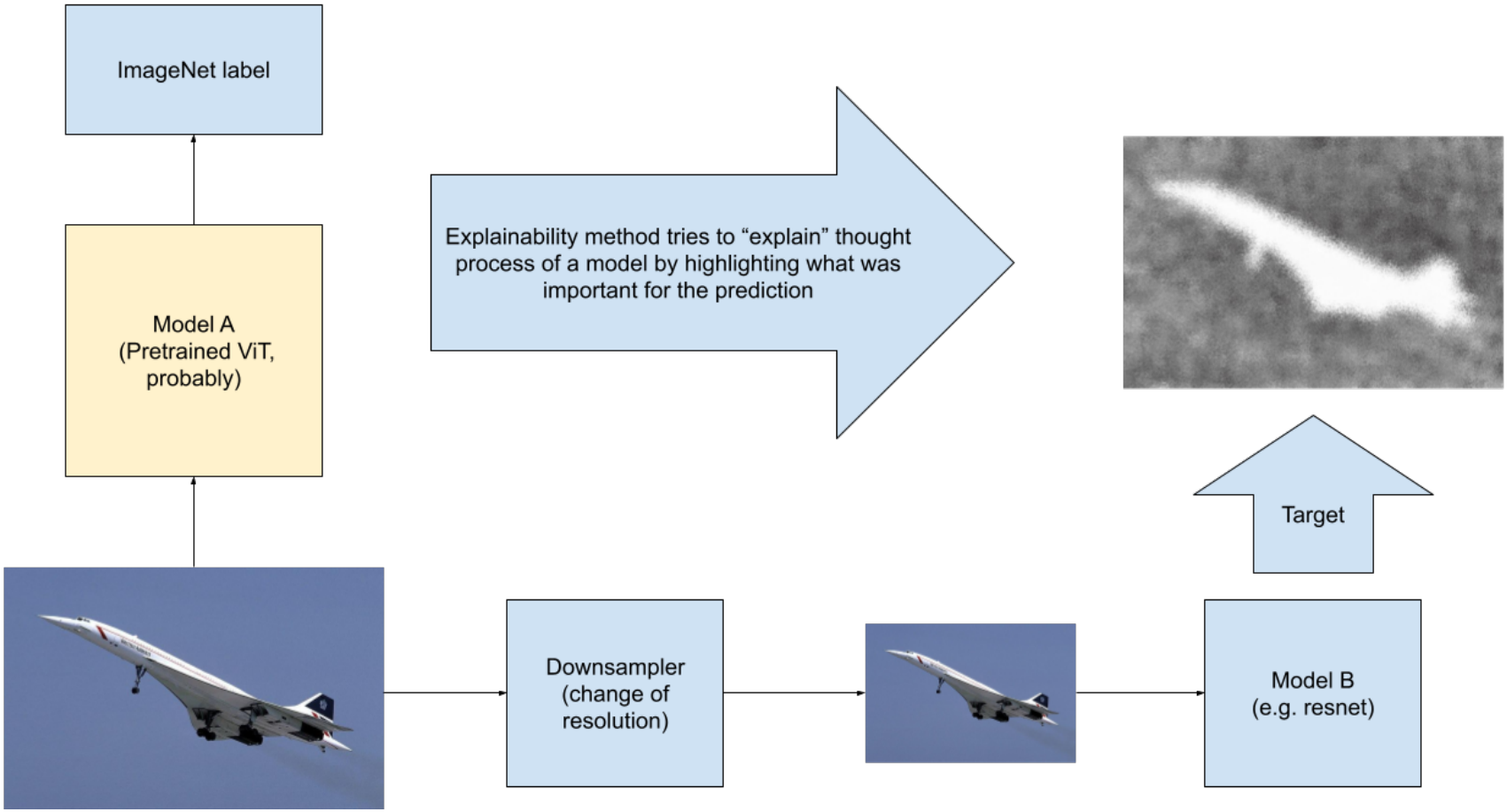
Ideal circumstances testing

Final results

Introduction

The task

Given a model A and an image I train a model B that predicts what is going to be important on the image I for the model A, *based on downsampled version of the image* I.



”British Airways Concorde G-BOAC 03.jpg” by Eduard Marmet; licensed on CC-BY-SA 3.0; `https://commons.wikimedia.org/wiki/File:British_Airways_Concorde_G-BOAC_03.jpg`

Why?

For the current transformer-based image classification models, the more patches are provided, the longer their inference time is.

We will predict (using a small, fast model) which patches are going to be useful and drop a given percentage of least useful patches.

Hopefully, it won't mess up the accucracy.

Attention rollout

Attention rollout

Attention rollout is an explainability method for transformers.
It can show which **patches** of an image were important for prediction of a transformer.
`https://github.com/jacobgil/vit-explain`

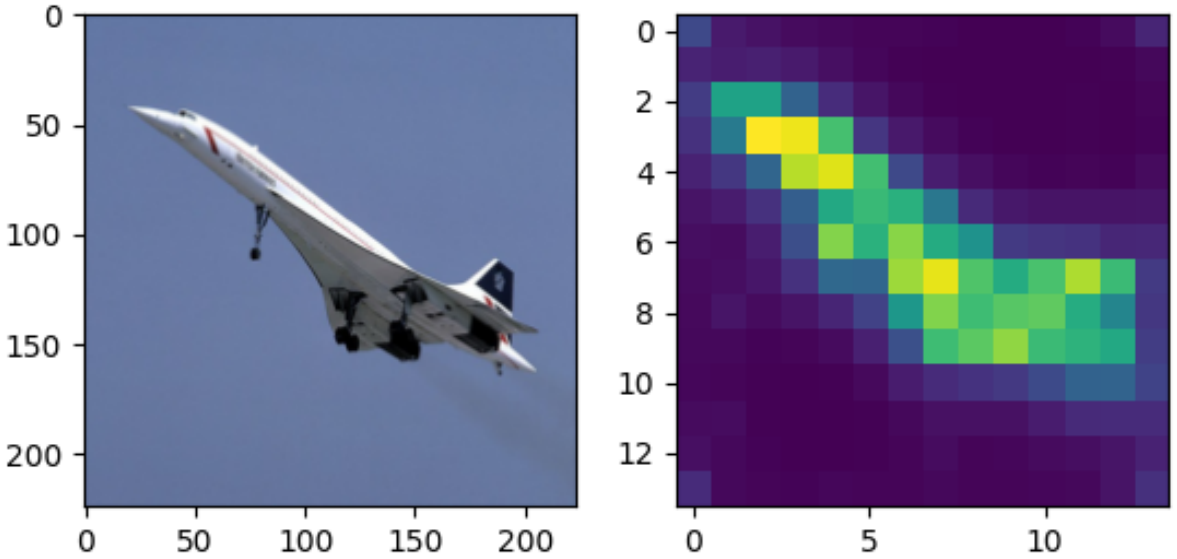


Figure: Example attention rollout for deit_tiny_patch16_224

Rollout prediction

Variants

To train a model that predicts attention rollout, we've decided to check different rollout variants:

- Default rollout
- Gradient-based rollout

We've also decided to check prediction power of different model architectures:

- Convnets
- Transformers

Work split

The search for a good student model was split as follows:

- Ignacy – Gradient rollout, transformer
- Kacper – Default rollout, convnet
- Kiryl – Gradient rollout, convnet
- Dominik – Default rollout, transformer

Work was to be done in the Implementation II stage of the project.

Partial results

It was confirmed that both transformers and convnets can learn attention rollouts.

Furthermore, we've observed that gradient-based rollout method is more computationally expensive. In addition to this, attention rollouts for smaller models seemed to be much more consistent with our human intuition than rollouts of big transformers.

Results for convnet

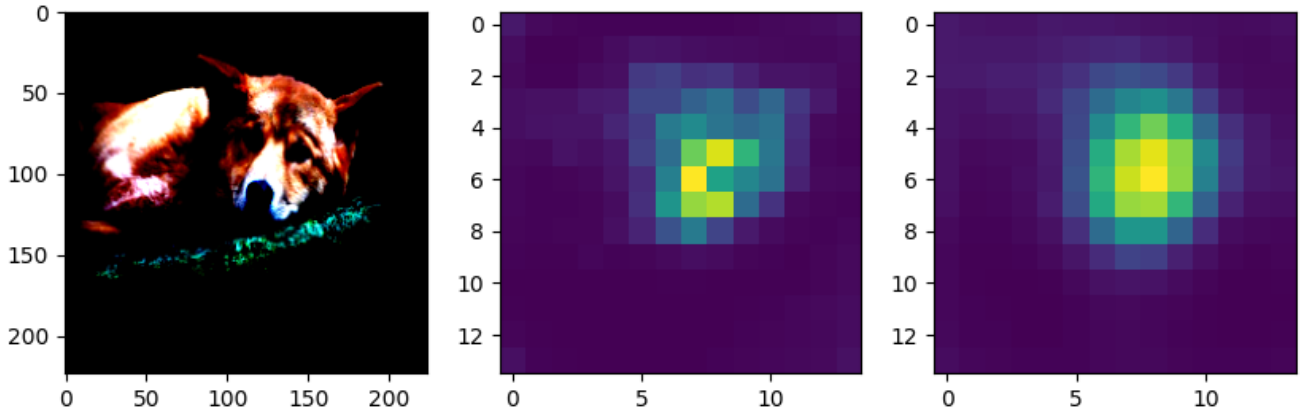


Figure: Middle: attention rollout; Right: predicted rollout

Results for convnet

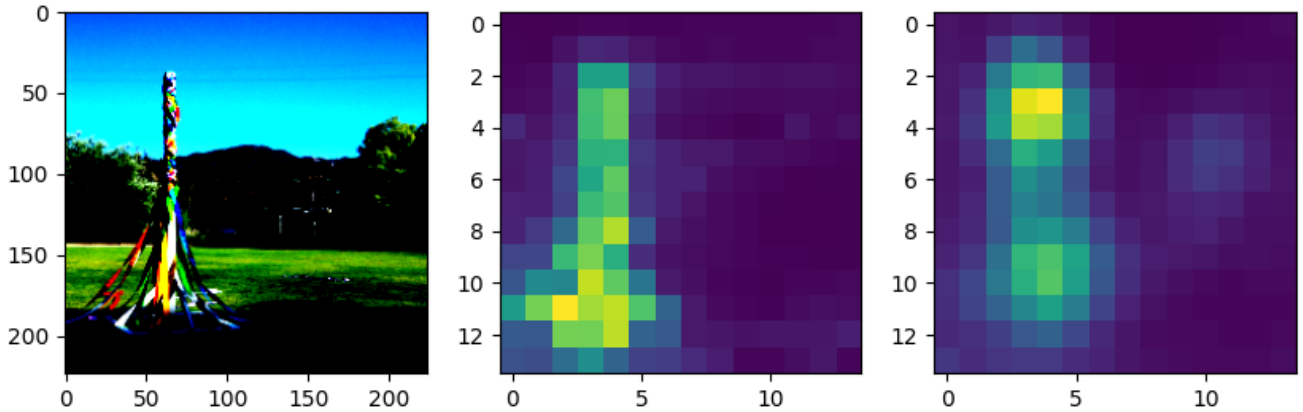


Figure: Middle: attention rollout; Right: predicted rollout

Unhelpful attention rollout

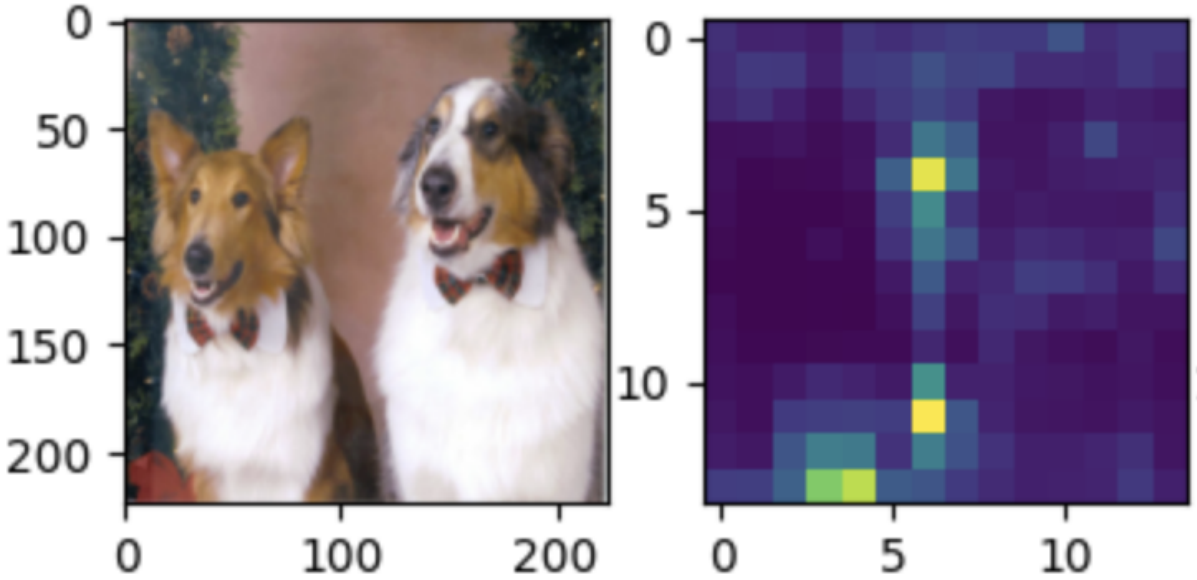


Figure: Attention rollout for a bigger transformer

Split

At that point, we've decided to split. Kacper and Kiryl were to enhance models for predicting rollouts, while Dominik and Ignacy were supposed to test how a model would perform while making predictions based on incomplete data (more on that in a second).

Ideal circumstances testing

Establishing the baseline

We've implicitly assumed that reducing number of image tokens won't cause the model to predict poorly if we sample them based on rollouts...

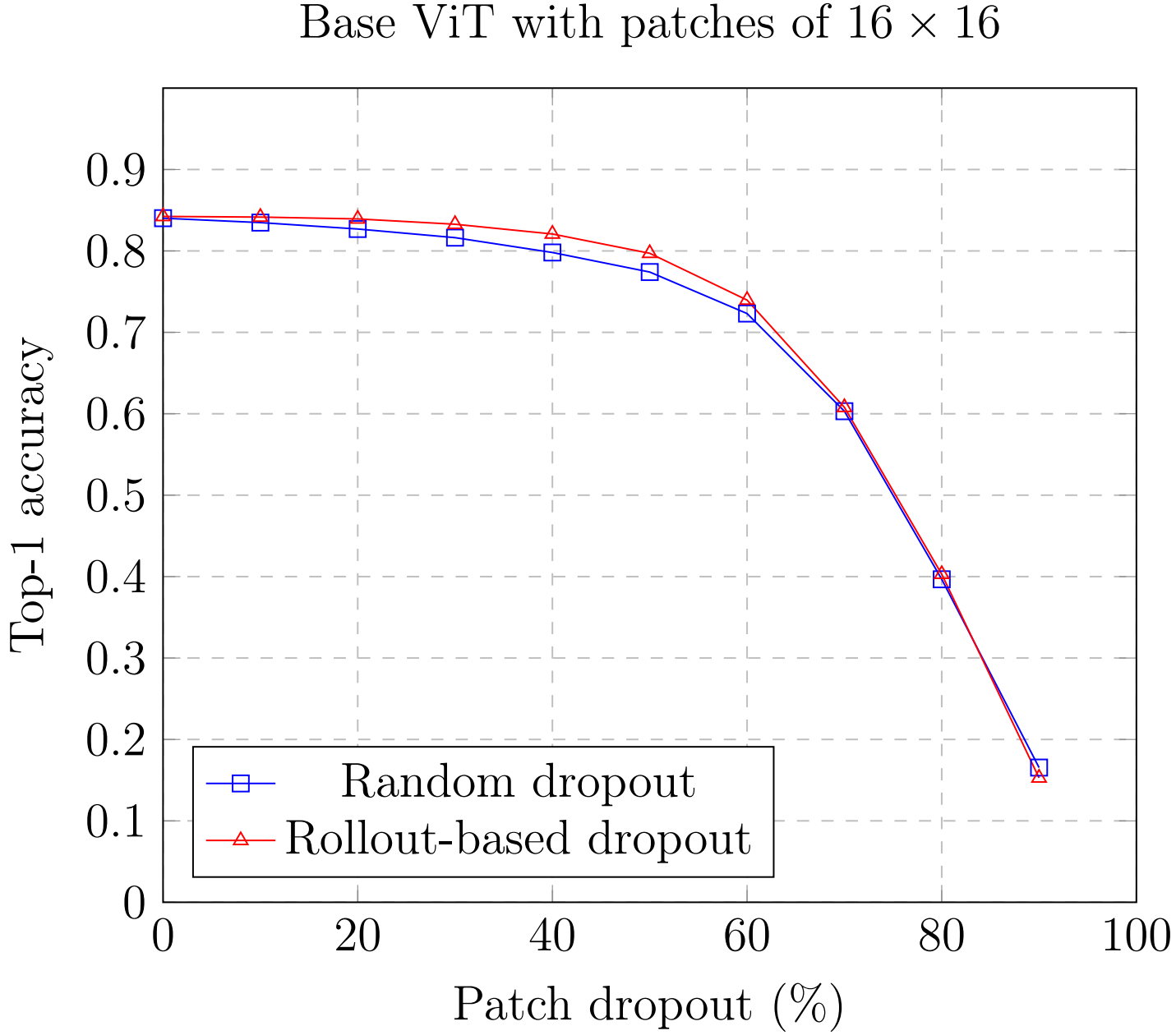
...but is that true?

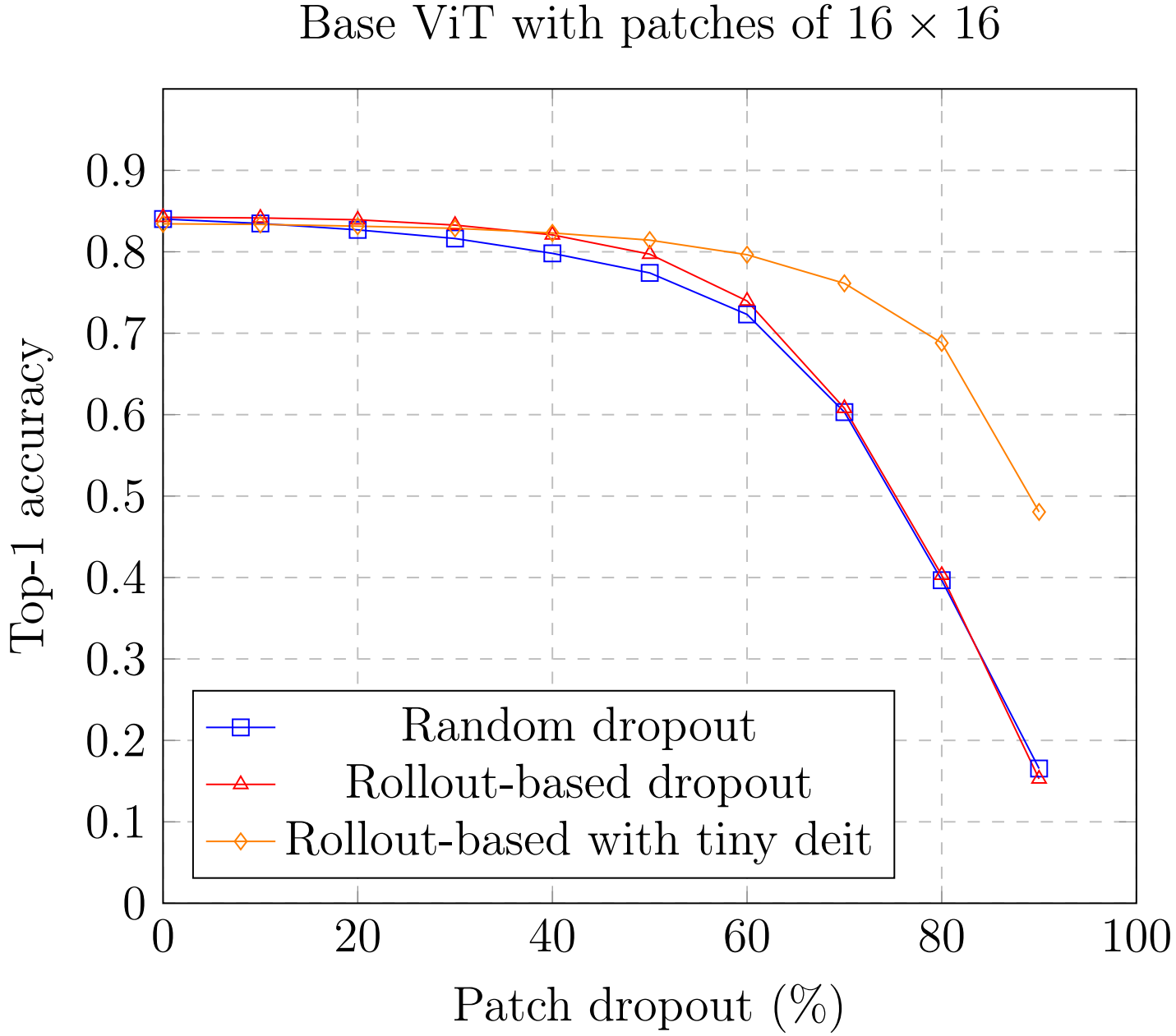
Experiment

We will perform an experiment by plotting the transformer's performance after dropping $k\%$ of image patches randomly or according to a rollout of a transformer.

It will validate whether we can do something interesting with the rollout prediction.

(Dominik)


Base ViT with patches of 16 × 16
Top-1 accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Patch dropout (%)
Random dropout
Rollout-based dropout


Base ViT with patches of 16 × 16
Top-1 accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Patch dropout (%)
Random dropout
Rollout-based dropout
Rollout-based with tiny deit

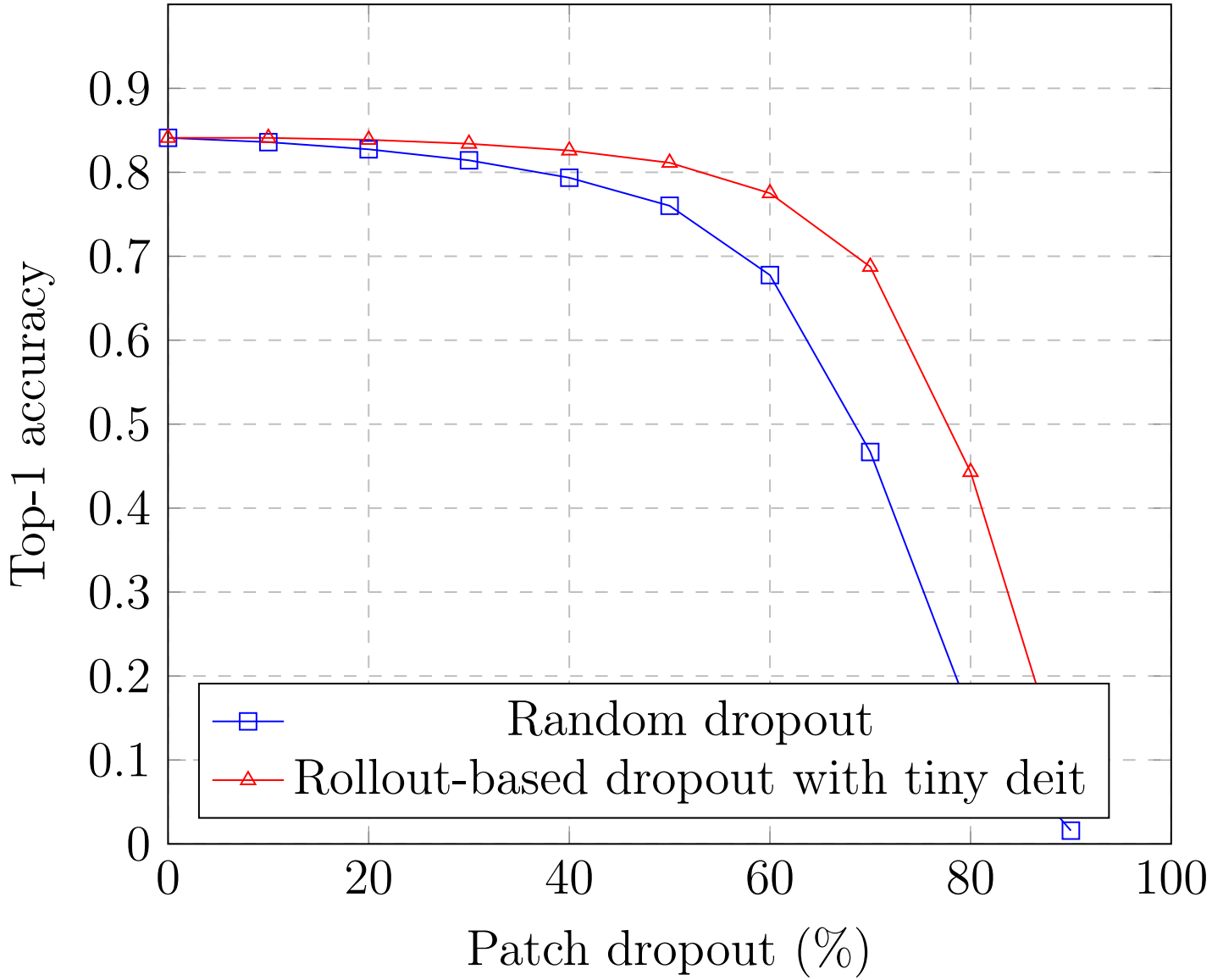
Wait, what?

It turns out that you can remove 90% of ViT inputs, and it will retain around 48% accuracy on ImageNet validation set.

Basic ViT was *not* trained with dropping image patches in mind.

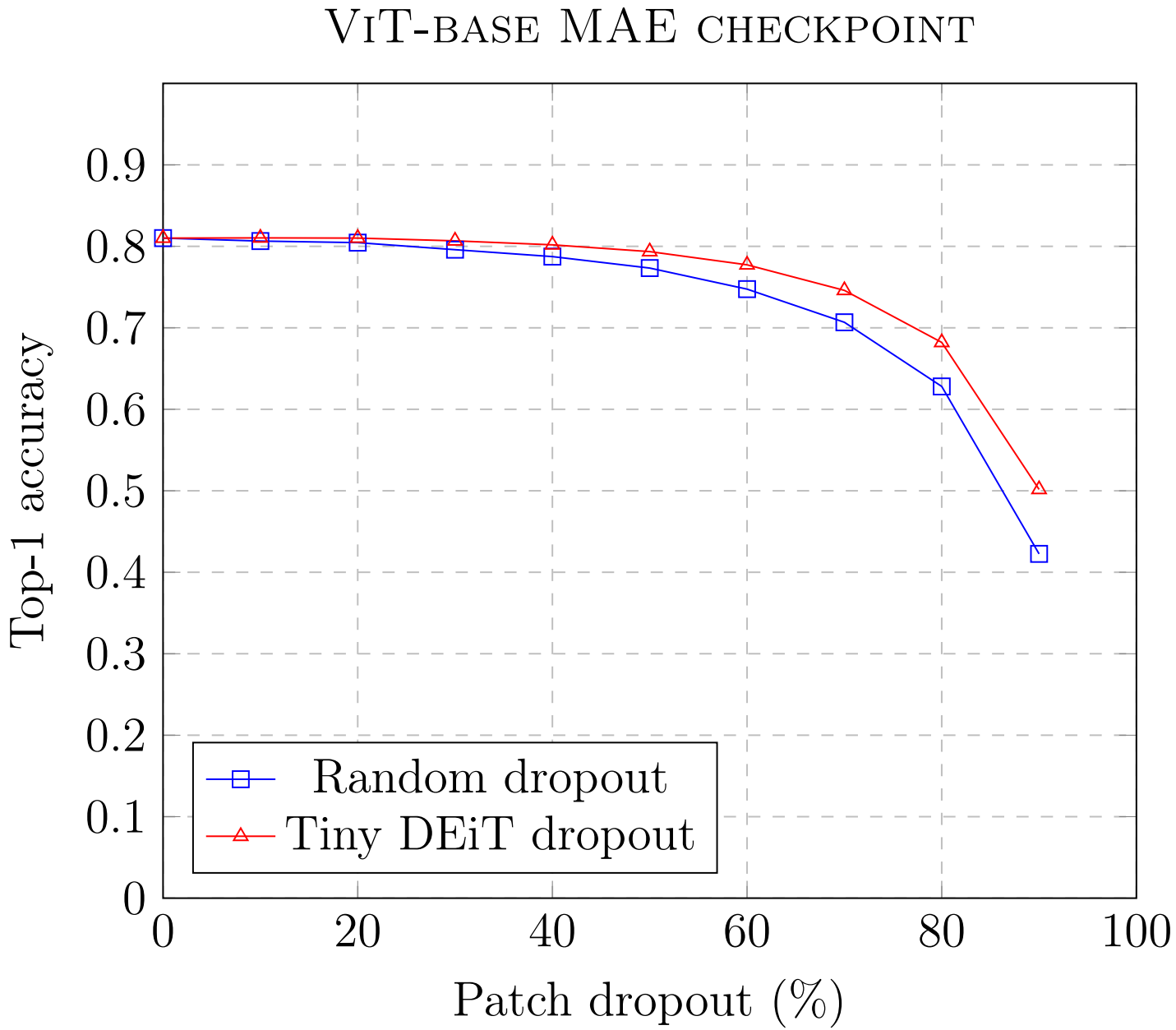
Larger ViTs collapse when provided with only 10% of an image, though.

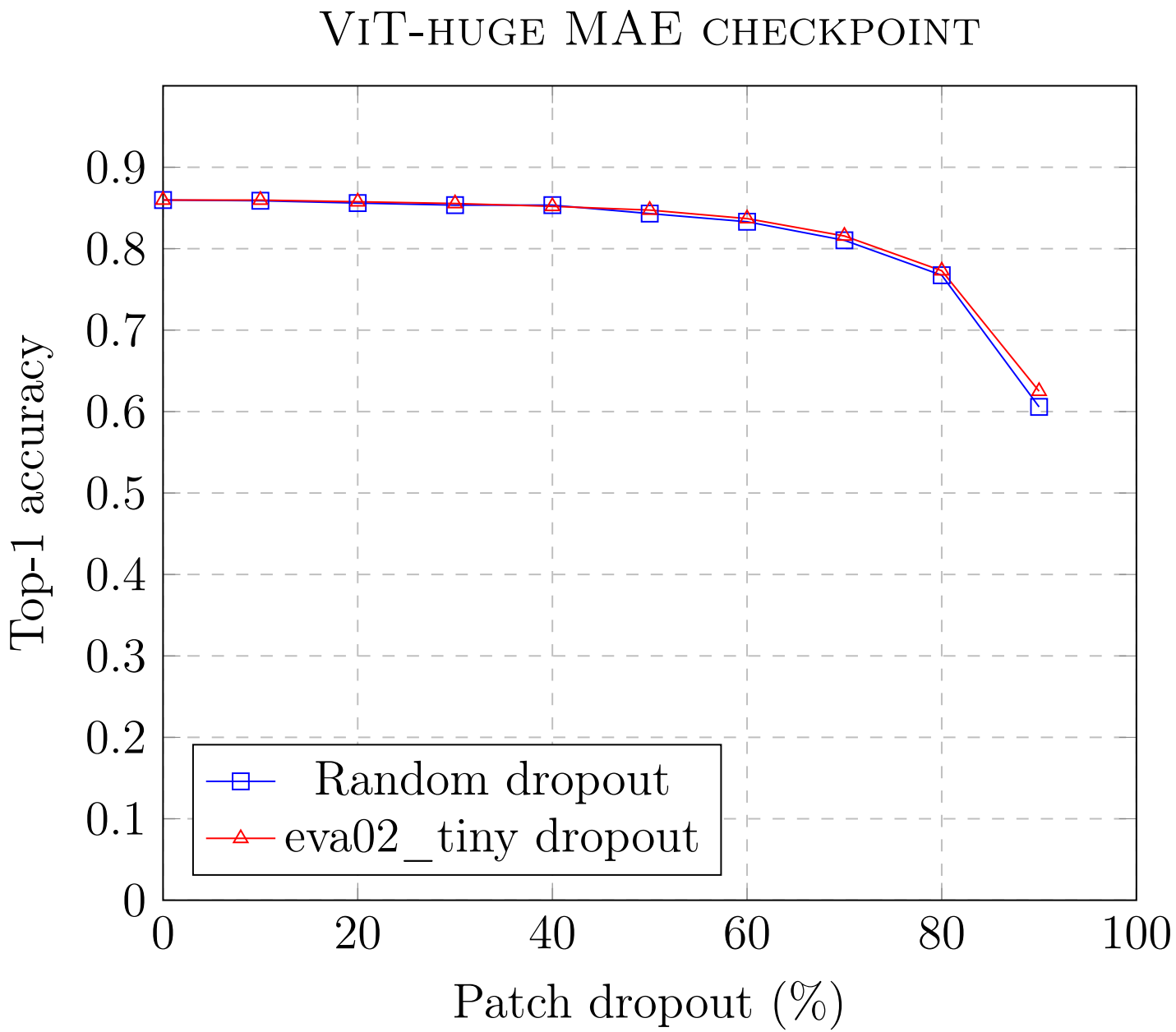
vit_large_patch16_224.augreg_in21k_ft_in1k



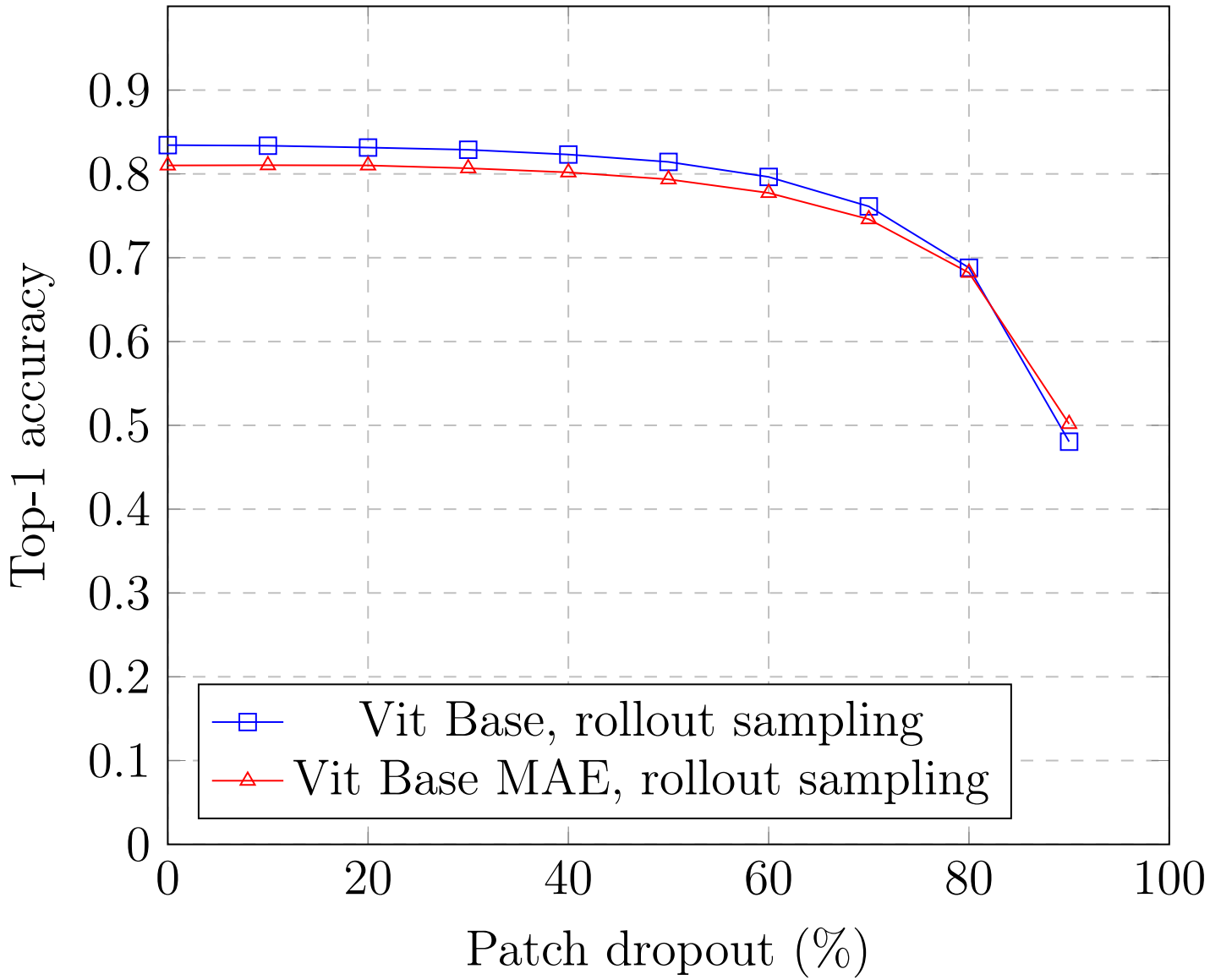
Let's push it further

Let's see what happens when we perform the same procedure, but on MAE. It was trained on incomplete images, so perhaps it'll be able to perform even better when we apply our procedure to it.


ViT-base MAE checkpoint
Top-1 accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Patch dropout (%)
Random dropout
Tiny DEiT dropout


ViT-huge MAE checkpoint
Top-1 accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Patch dropout (%)
Random dropout
eva02_tiny dropout

A very disappointing chart



Final results

Final results

Kacper trained ResNet-like model from scratch (we wanted to input images with small dimension into it, speeding up inference times).

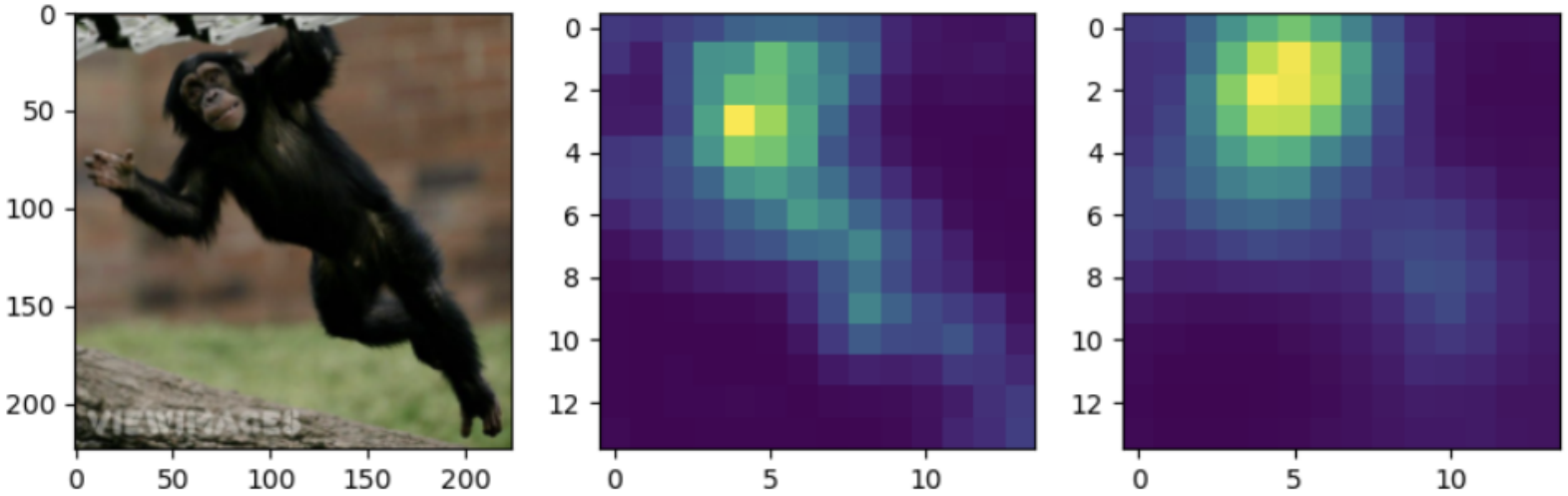
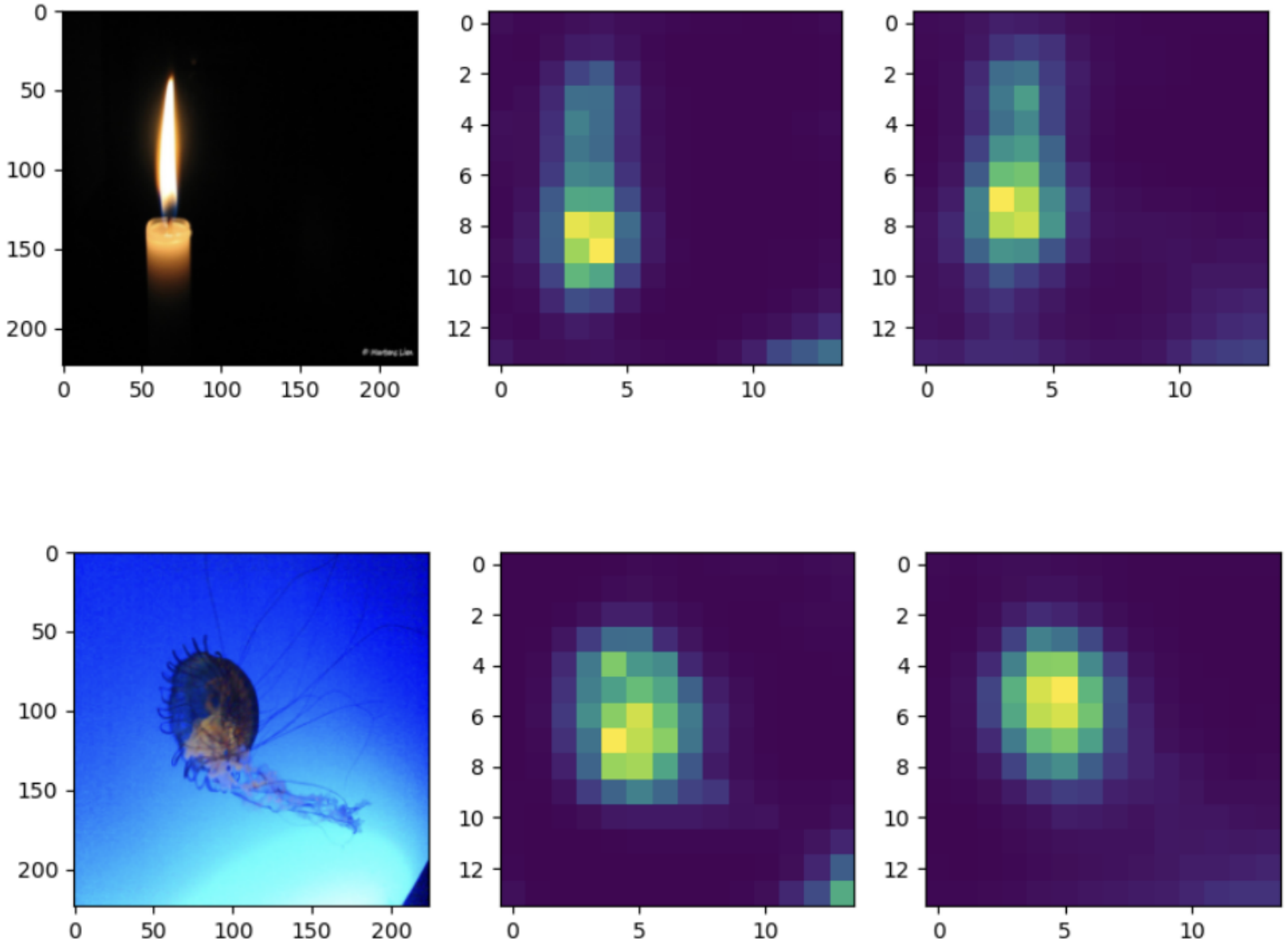
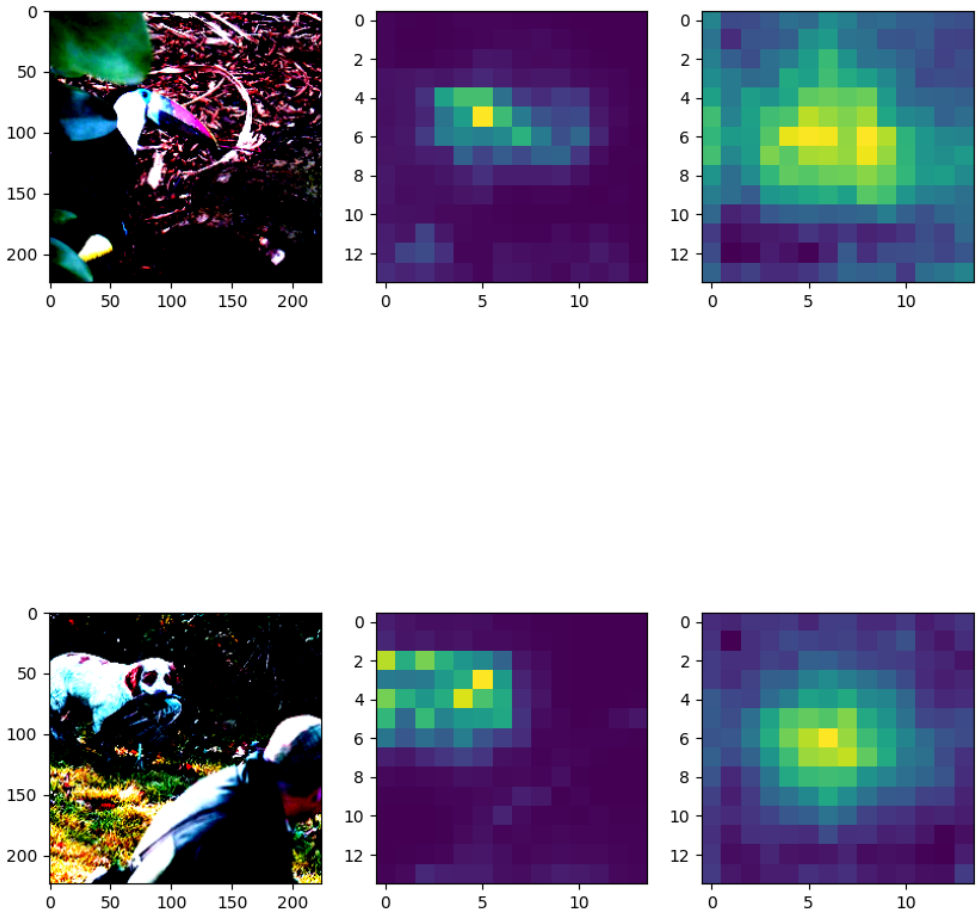


Figure: Ground truth in the center, prediction on the right


0
50
100
150
200
0
50
100
150
200
0
2
4
6
8
10
12
0
5
10
0
2
4
6
8
10
12
0
5
10
0
50
100
150
200
0
50
100
150
200
0
2
4
6
8
10
12
0
5
10
0
2
4
6
8
10
12
0
5
10

ViT

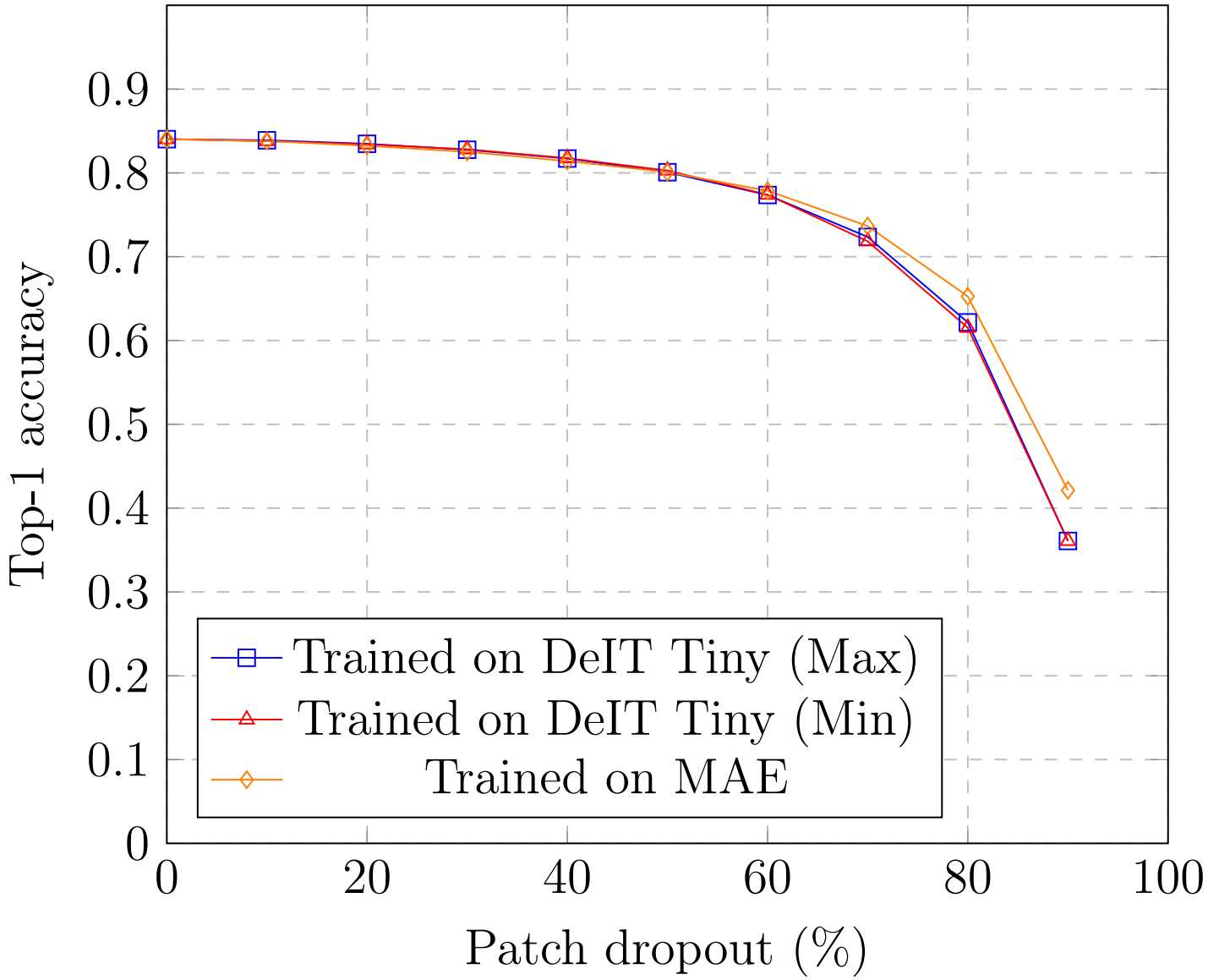
Furthermore, Kiryl trained a ViT-based rollout predictor.

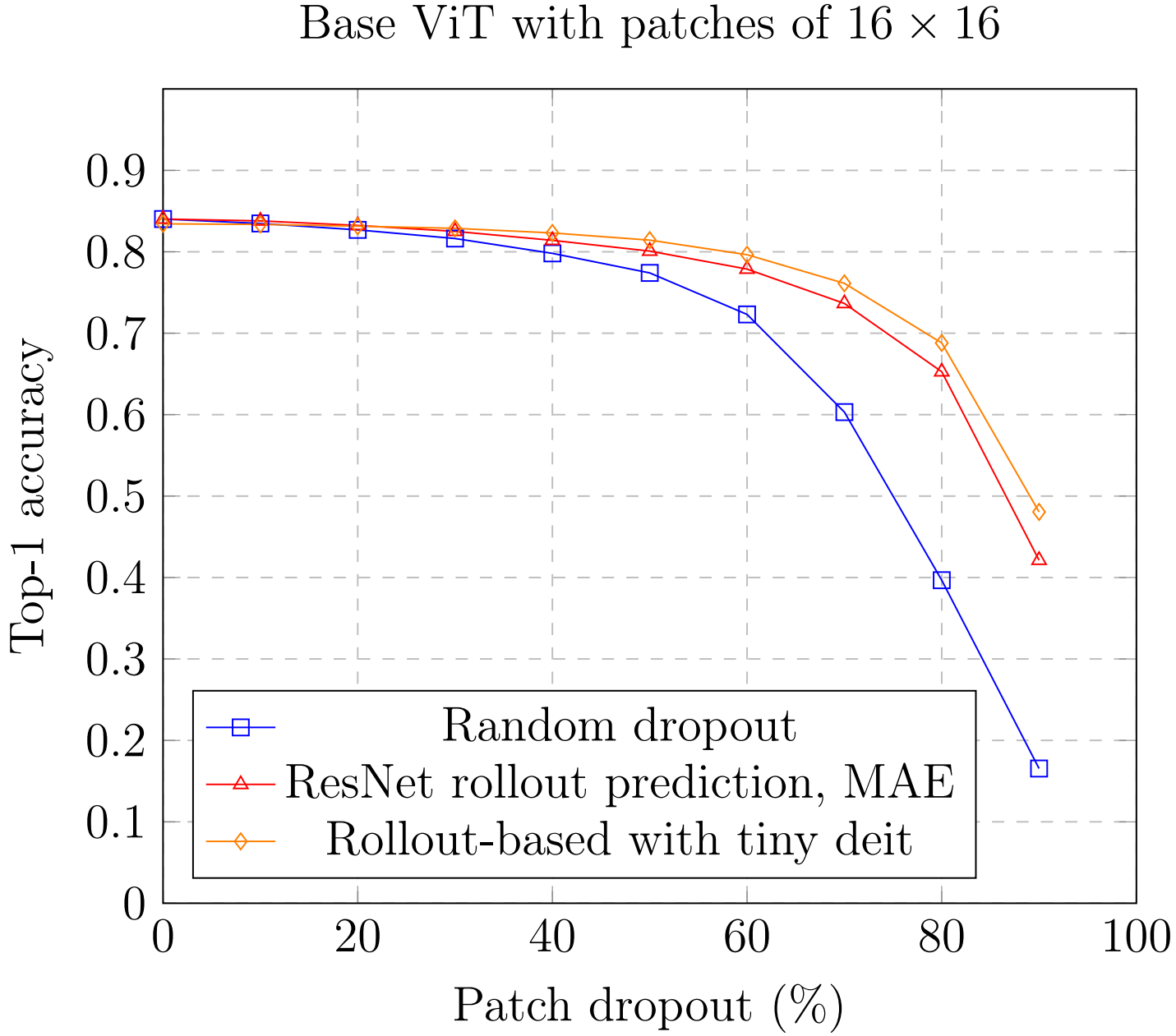


And how good is our sampling in practice?

Dominik verified that sampling from rollouts makes sense. But will sampling from our predictions of rollouts make sense? Ignacy prepared code to check that.

ResNet rollout comparison for base ViT with patches of 16×16




Base ViT with patches of 16 × 16
Top-1 accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
0
20
40
60
80
100
Patch dropout (%)
Random dropout
ResNet rollout prediction, MAE
Rollout-based with tiny deit