MODEL EXTRACTION ATTACK FOR VIDEO CLASSIFICATION

Objective

Develop an efficient strategy and implementation to extract the video based models in the black box and grey box for the following models:

1. Model Extraction for Swin-T Model for Action Classification on Kinetics-400 dataset
2. Model Extraction for MoViNet-A2-Base Model for Video Classification on Kinetics-600 dataset

BACKGROUND

Data Collection

Data available

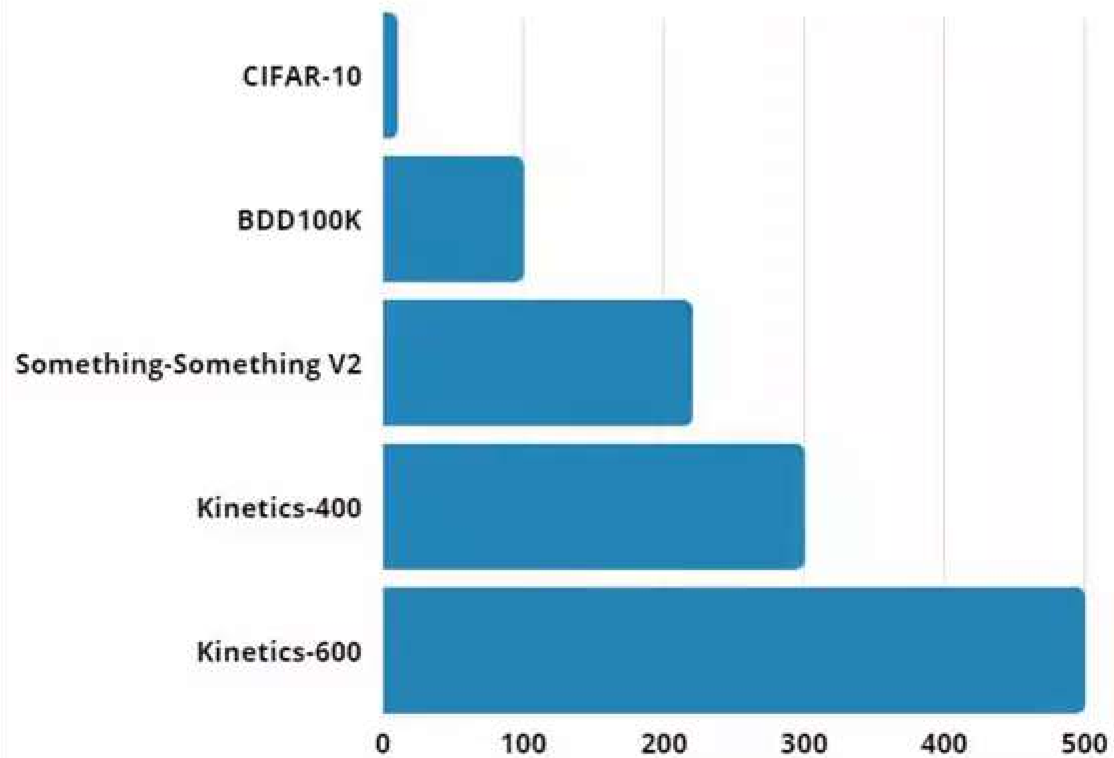
5 million video clips of 10 seconds each

Training Data: 4 million video clips
↓
5%- 20,000 videos
32 videos of each class to maintain class balance
↓
Converting each clip into 16 frames
16*20,000=320,000

Validating Data: 30,0000 video clips
↓
100%- 30,000 videos
↓
Converting each clip into 16 frames
16*30,000=480,000

Data used in grey box

Data size comparison



famous datasets

Closely related classes

kicking soccer ball

kicking field goal

catching or throwing softball

catching or throwing baseball

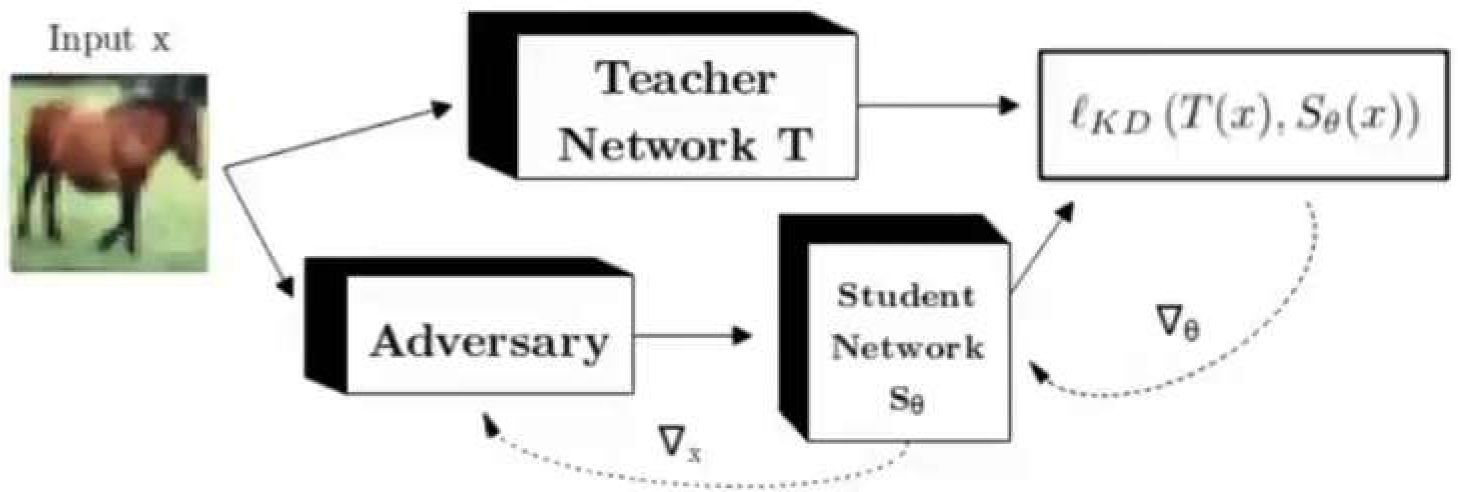
Settings

Black Box : Full dataset abstraction

Grey Box : Only 5% of Kinetics dataset accessible

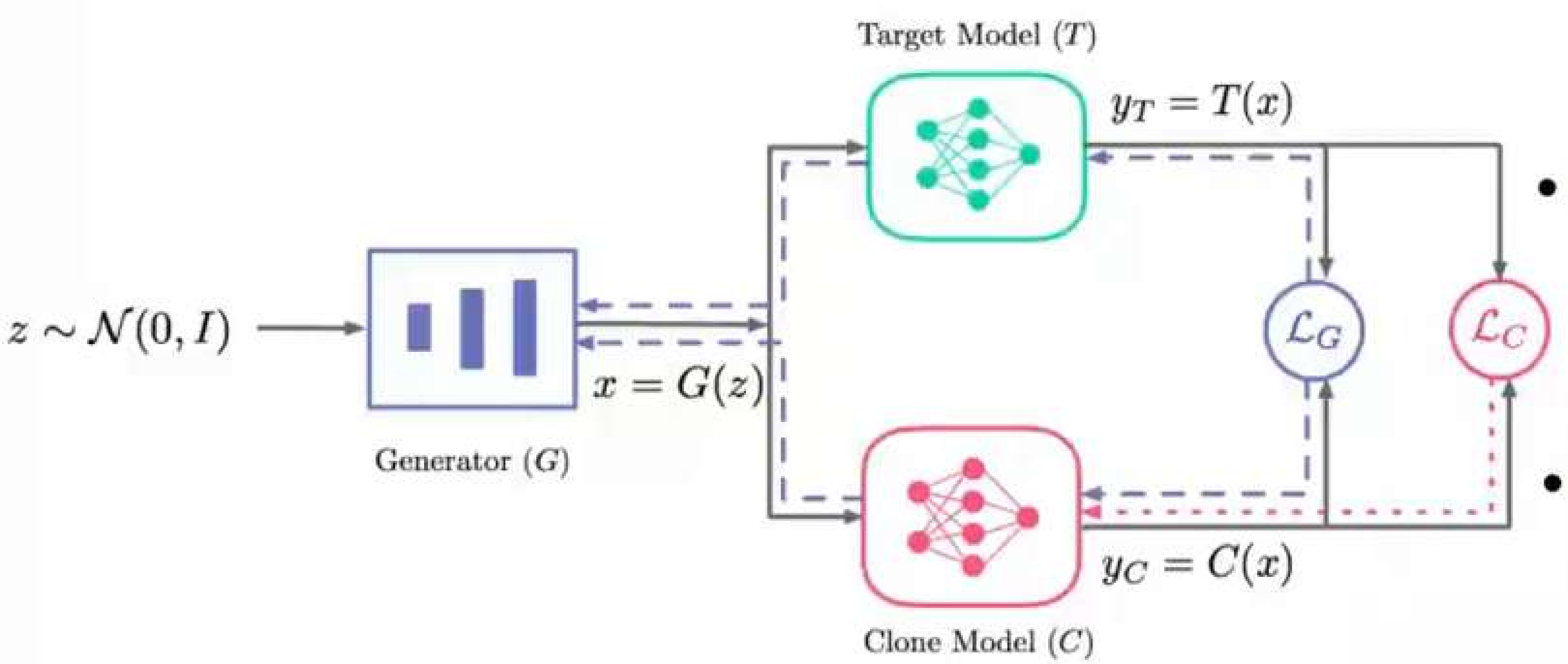
Literature Survey

Robust Knowledge Distillation



- Showed that knowledge distillation using only natural images can **preserve much of the teacher's robustness** to adversarial attacks.
- Introduced **Adversarial Robust Distillation (ARD)** for producing small student networks **robust to adversarial attacks**.

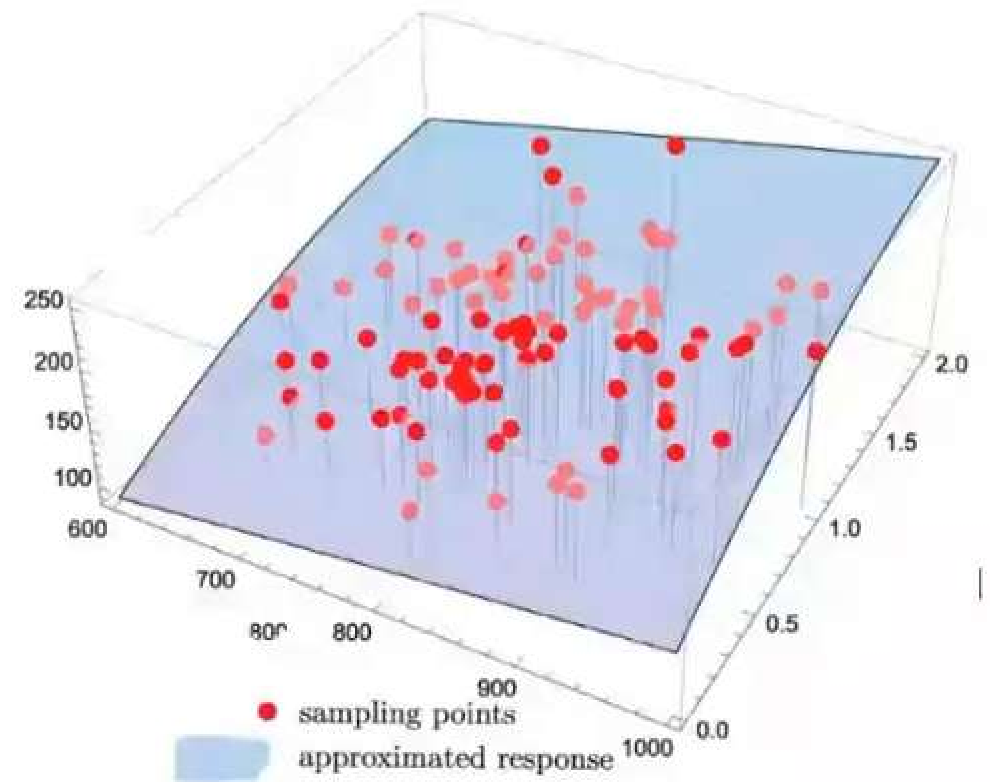
DataFree MAZE

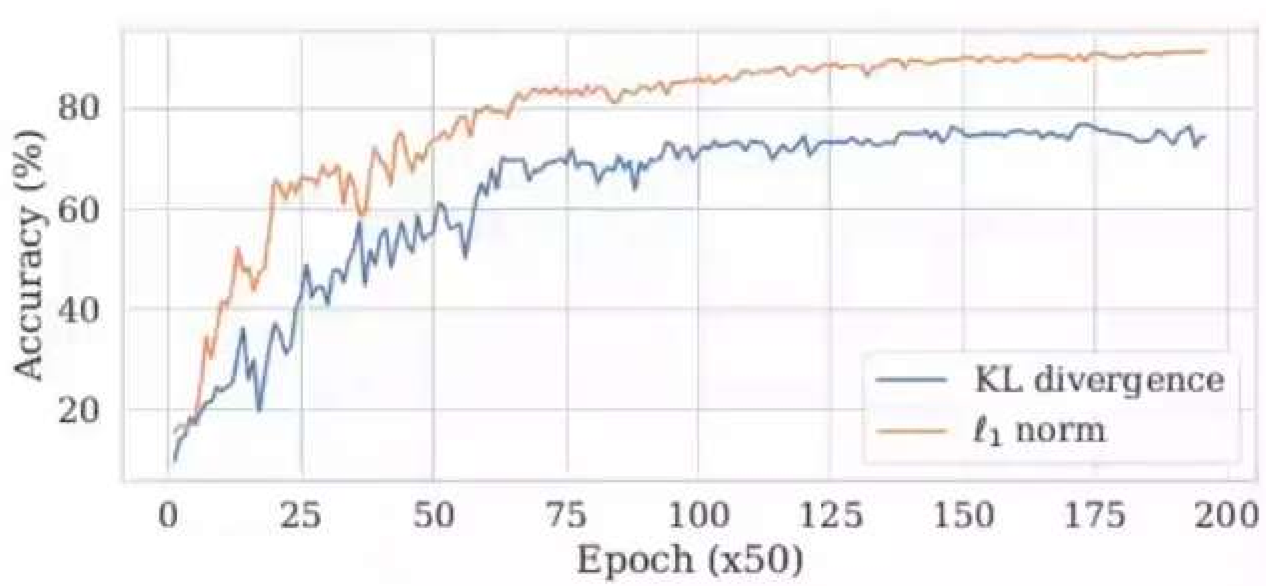


- In this paper, the proposed approach does not require any proprietary data and **can instead create synthetic data** using a **generative model**.
- Relies on **zeroth-order gradient estimation** to approximate the gradients of the teacher model.

Literature Survey

DataFree Model Extraction





- This subsequent work utilises the **forward differences** method for the zeroth-order gradient estimation.

$$\nabla_{\mathrm{FWD}} f(x) = \frac{1}{m} \sum_{i=1}^{m} d \frac{f(x + \epsilon \mathbf{u_i}) - f(x)}{\epsilon} \mathbf{u_i}$$

- Introduces the use of L1 loss as an alternative for KL Divergence loss to **prevent vanishing gradients** and **faster convergence**.

$$\mathcal{L}_{\ell_1}(x) = \sum_{i=1}^{K} |v_i - s_i|$$

Used when logits are **accessible**.

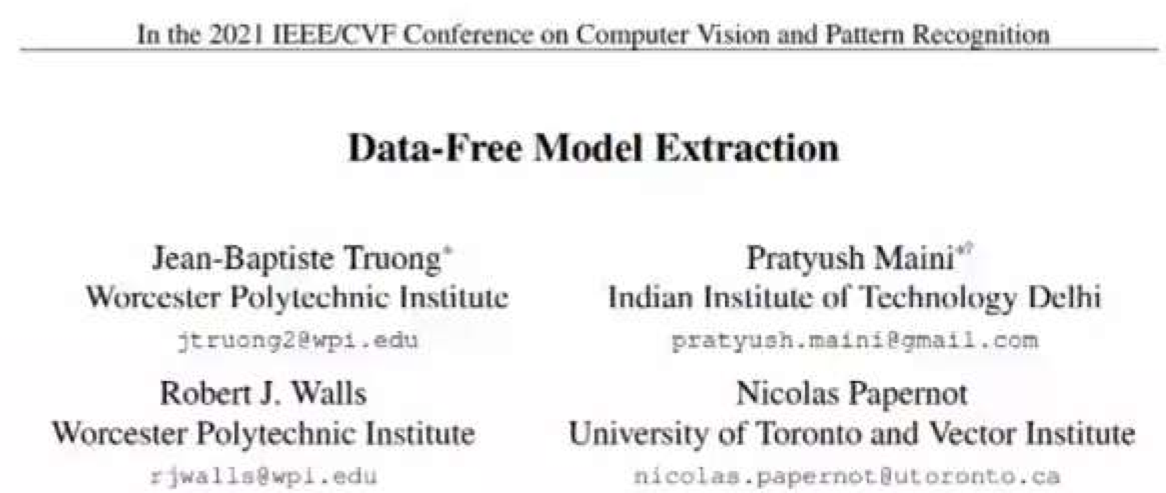
$$\mathcal{L}_{\mathrm{KL}}(x) = \sum_{i=1}^{K} \mathcal{V}_i(x) \log\left(\frac{\mathcal{V}_i(x)}{\mathcal{S}_i(x)}\right)$$

Used when logits are **not accessible**

IDEATION

Inspiration

- Our model extraction strategy is inspired by the work **'Data-Free Model Extraction'** published in CVPR 21.
- This paper proposes a data-free model extraction approach for **static images,** achieving high accuracy with reasonable query complexity: **0.99×and 0.92× the victim model accuracy** on **SVHN** and **CIFAR-10** datasets given **2M** and **20M** queries respectively**.**
- Extrapolated this strategy to video-based models, taking into account **space and time tokens of videos.**



In the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition

Data-Free Model Extraction

Jean-Baptiste Truong*
Worcester Polytechnic Institute
jtruong2@wpi.edu

Pratyush Maini*†
Indian Institute of Technology Delhi
pratyush.maini@gmail.com

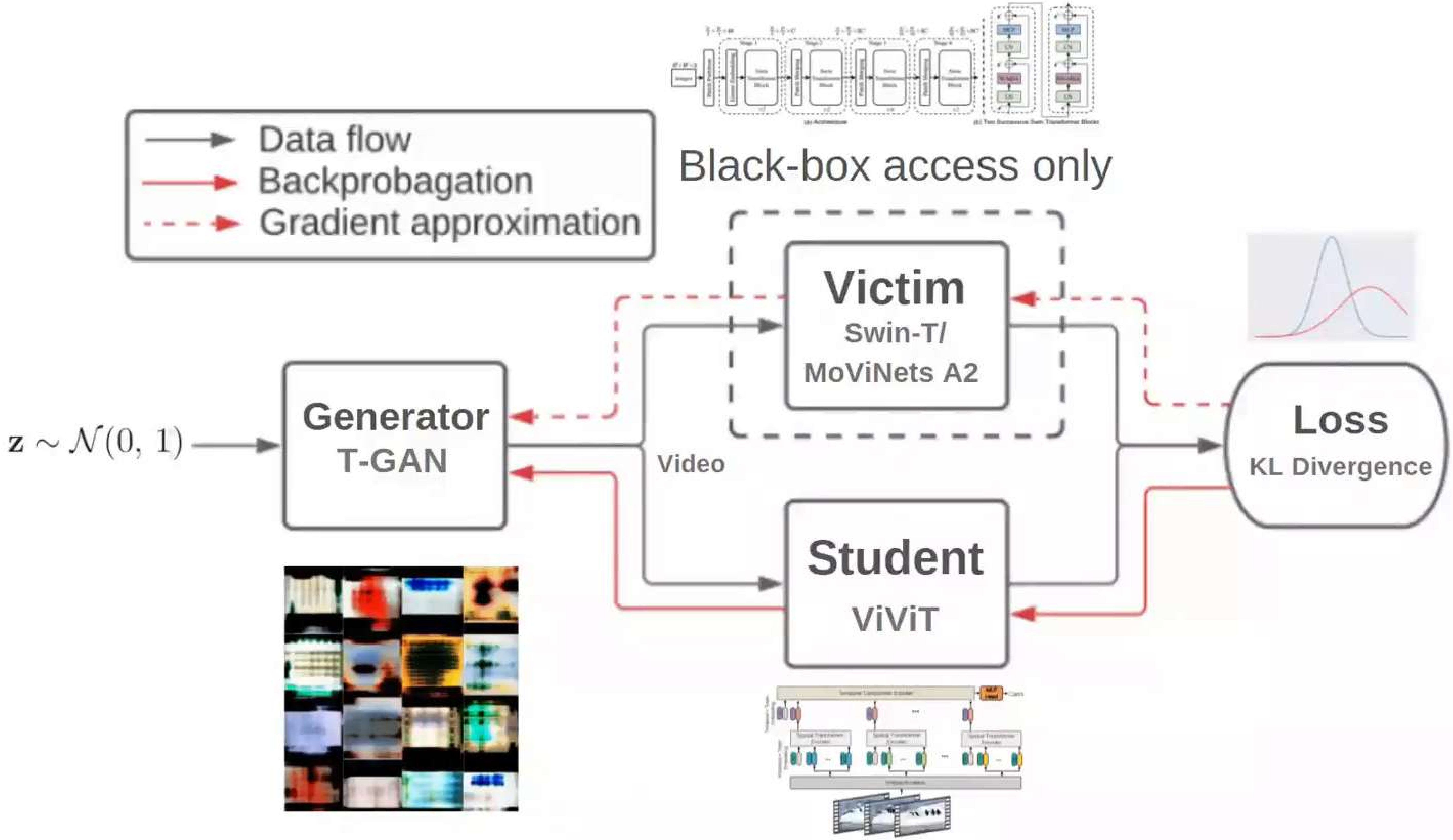
Robert J. Walls
Worcester Polytechnic Institute
rjwalls@wpi.edu

Nicolas Papernot
University of Toronto and Vector Institute
nicolas.papernot@utoronto.ca

Goal

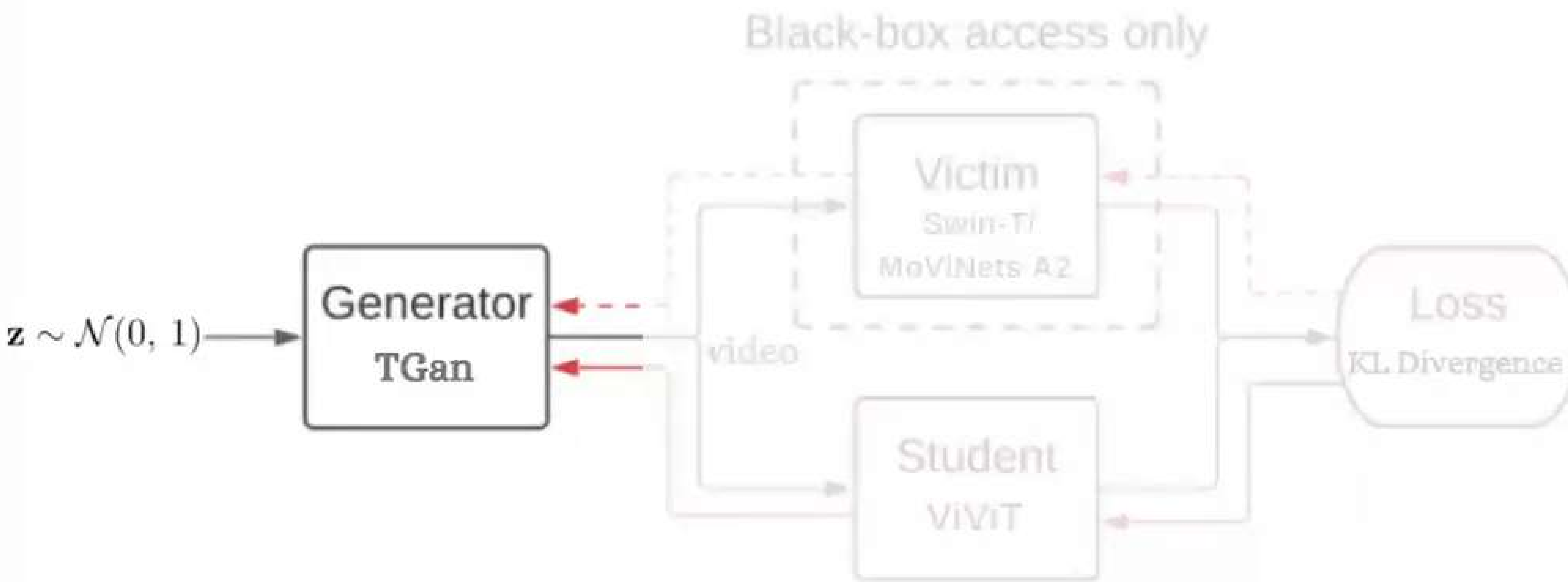
- **Minimize the number of queries** made to the model-to-be-extracted (Swin-T/ MoViNet-A2-Base) with a **novel query generation process.**
- Maximise the student model's accuracy upon the victim's test set .

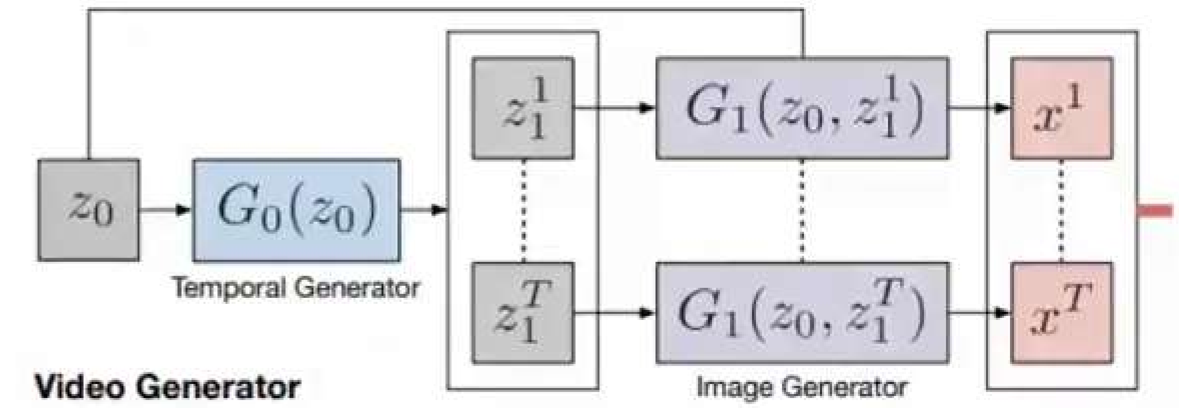
Our End-to-End Framework

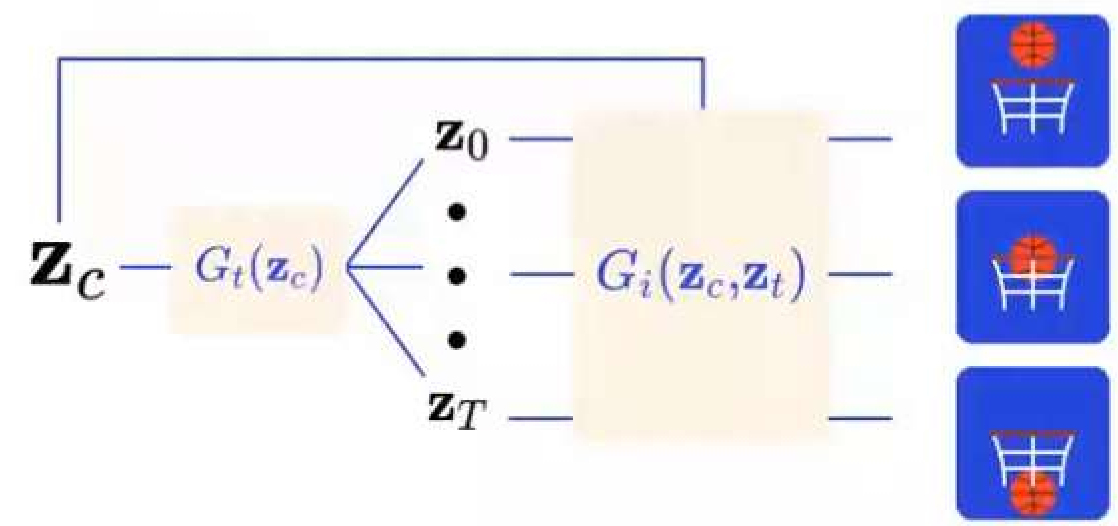


The Generator

Temporal GAN

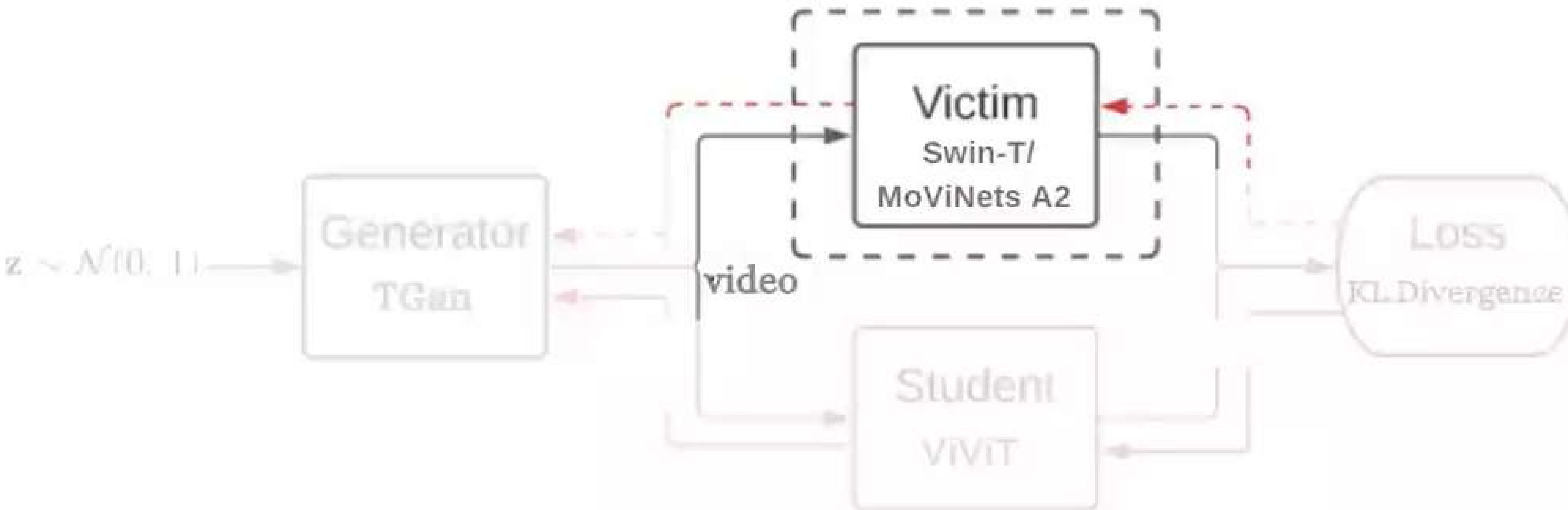




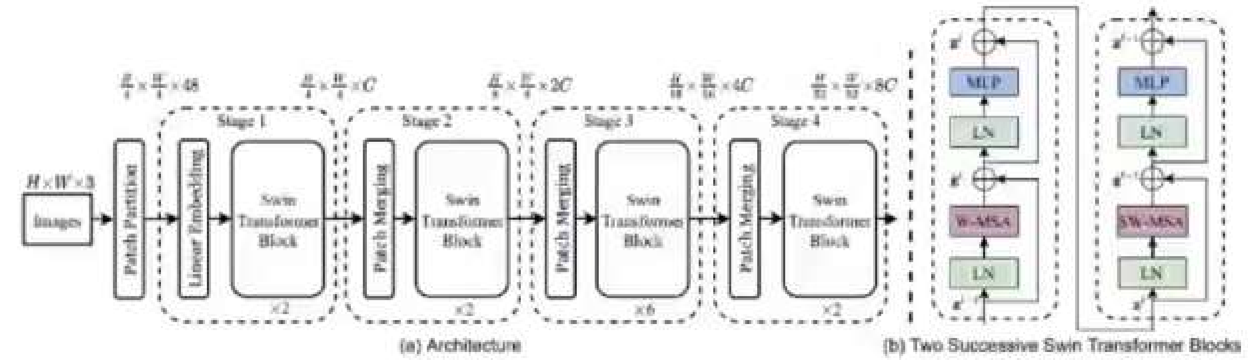


- The generator of Temporal-GAN consists of two sub-networks, namely the **temporal generator** and an **image generator**.
- Temporal generator first yields a set of **latent variables** for the image generator.
- It serves as an adversary to **maximize the disagreement** between Student and Victim model.

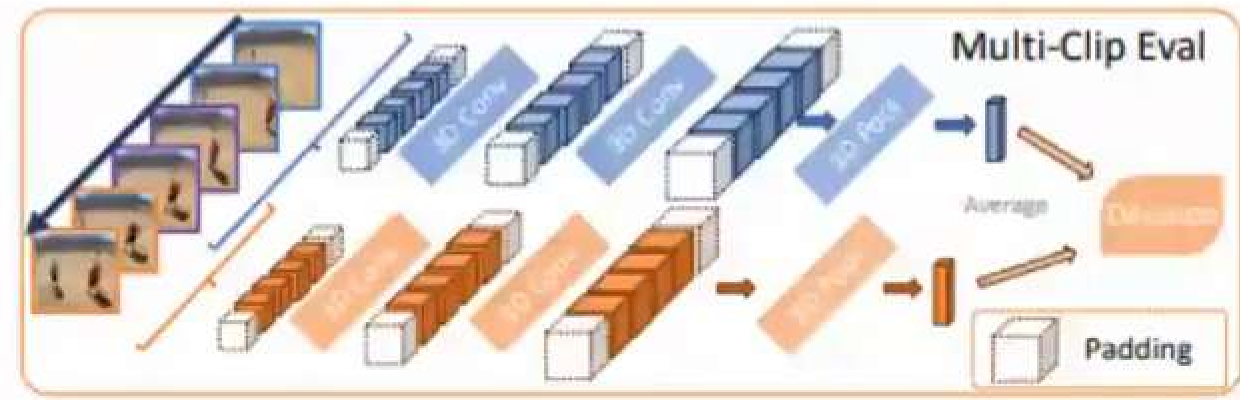
The Victim Model



SwinT



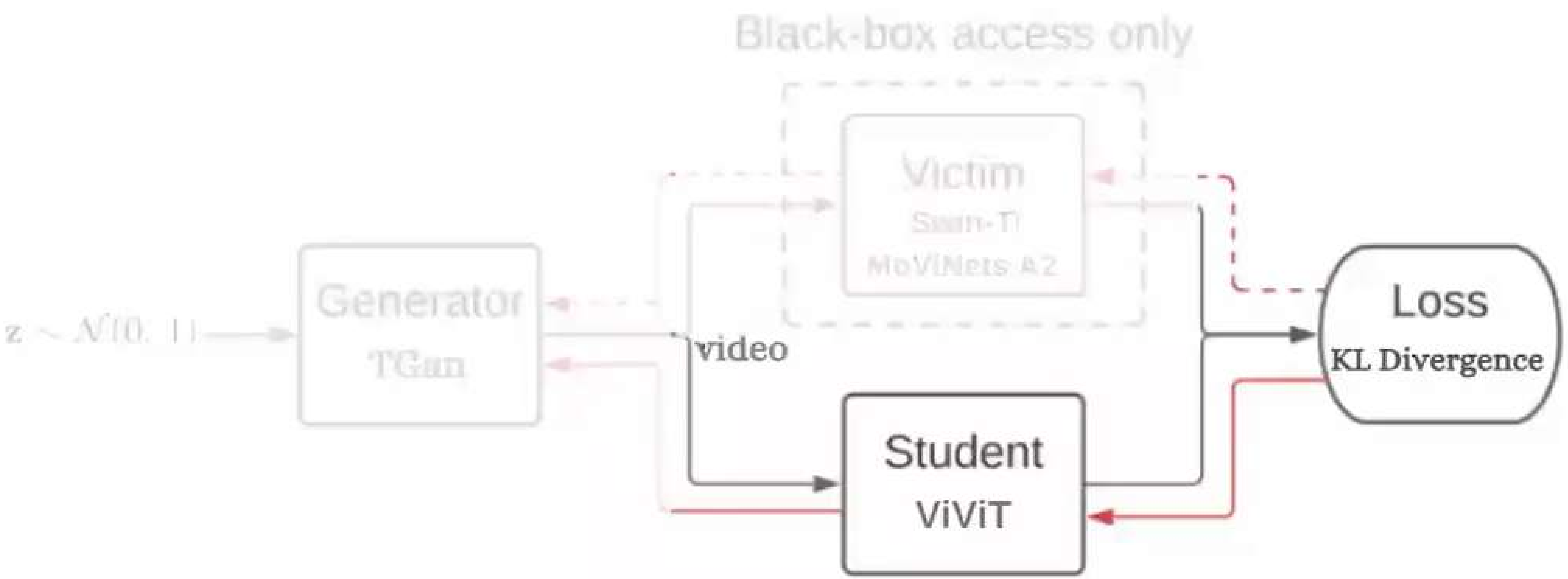
MoViNetA2 - Base

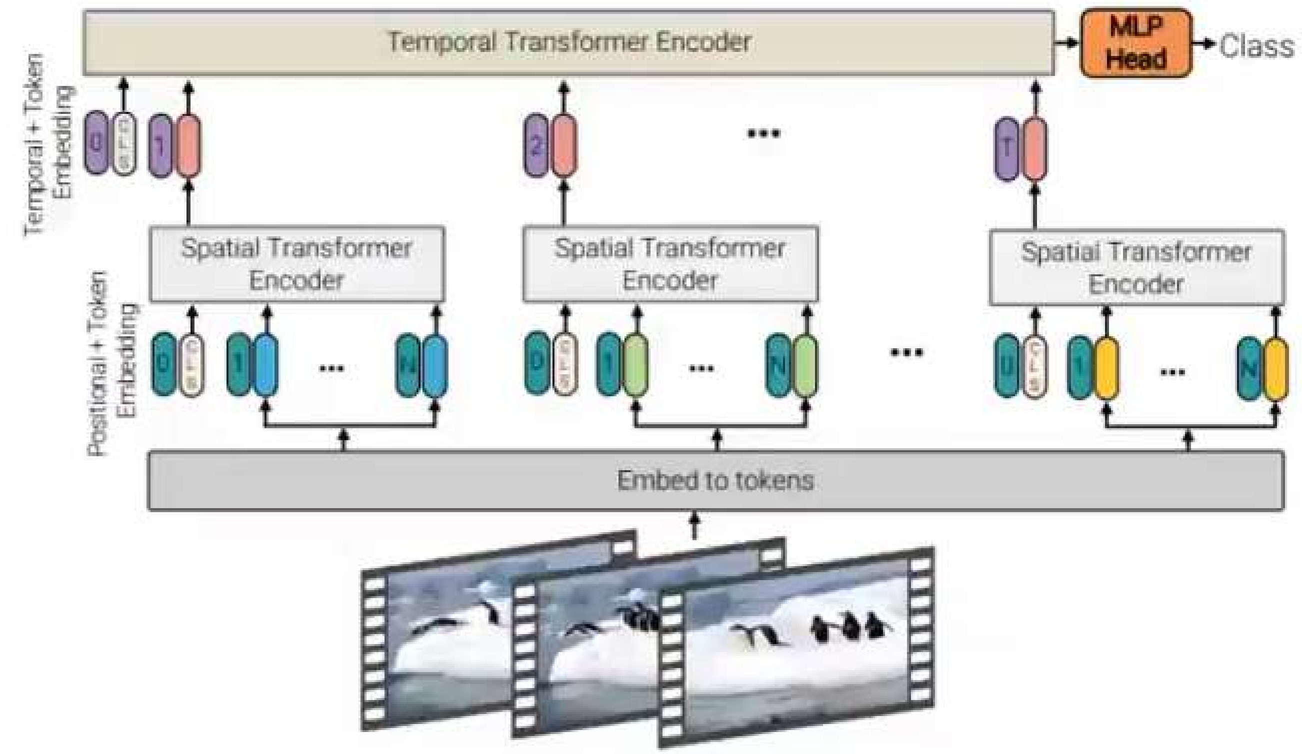


- The **Victim model** is inferred to **gain only a label for the query video.**
- Black box access of the victim model signifies that we **do not have access to the weights or gradients for backpropagation.**

The Student Model

Video Vision Transformer





- The ViViT model is computed on a sequence of **Spatio-temporal tokens** that we extract from the input video. The factorisations correspond to different attention patterns.
- It serves as an adversary to **minimize the disagreement** between Student and Victim model.

Our Approach

Black Box Setting

1. The generator of **Temporal GAN generates random videos,** used to infer/**attack the target model.**
2. The generated **input video and the obtained target label** from the victim model **are then used to update the student model.**
3. The loss is then calculated based on the output of the **student model**.
4. The **Generator** model tries to maximize the above loss to create better-exploiting queries so as to ensure the student model can learn efficiently, i.e in comparatively fewer queries.

Grey Box Setting

1. Similar to our black box setup, with just the **additional pre-training of this generator model** on the **UCF101** dataset and then using **5% of** the dataset (**Kinetics400/Kinetics600**).
2. **Introduced a threshold parameter:** It decides the **percentage** of the **real and the generated data in a batch.**
3. The pre-trained generator generates more quality queries and thereby reducing the number of queries needed to imitate the target model to a reasonable fidelity and the **use of threshold parameter prevents our model from over-fitting** on the 5% of the dataset.

Data for GreyBox

Generator output trained on UCF101 dataset

(1 - threshold) %

+

Samples from actual Kinetics-600 Dataset

(threshold) %

Single Batch

```
#train_loop
batch_size = 16
threshold = 0.3
for epoch in range(epochs)):
    for kin_real, _ in tqdm(train_loader):
      mask = torch.rand(batch_size) > threshold
      z = torch.randn((batch_size, 100))
      ucf_fake = Gen(z)
      vid = mask * ucf_fake + (1-mask) * kin_real
      out = model(vid)
      with torch.no_grad():
          target = teacher(vid)
          pred = target.argmax(dim=1)
      loss = loss_func(out, pred)
      loss.backward()
      optimizer.step()
```

Implementation and Compute Details

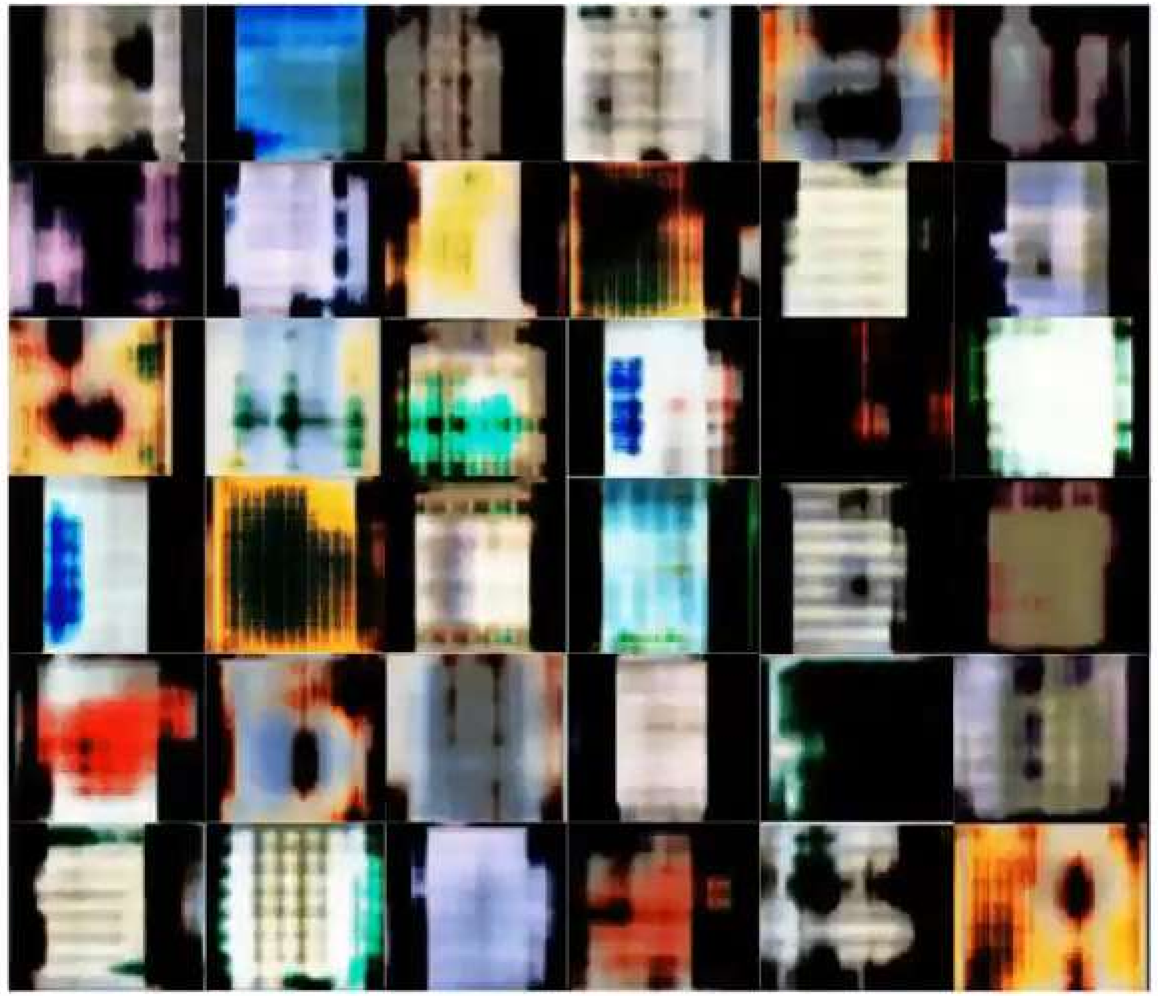
Model Training

- The student model is trained against the labels that the teacher model predicts on the query video generated by the generator.
- We customized the zeroth-order gradient estimator to work with an additional axis that represents the temporal characteristics of a sampled video.

Frameworks and Computation

- **3 Nvidia GPUs** with **11Gb VRAM, 92 Gb of RAM,** and an **Intel i9 CPU** with **16 cores.**
- For training the generator, we approximate gradients using zeroth-order estimations on top of the high number of iterations, thereby making it a computationally expensive process. In order to utilize the limited computing resources efficiently, our implementation was entirely done in **PyTorch lightning** to extensively parallelize our complete pipeline.

Generator Analysis

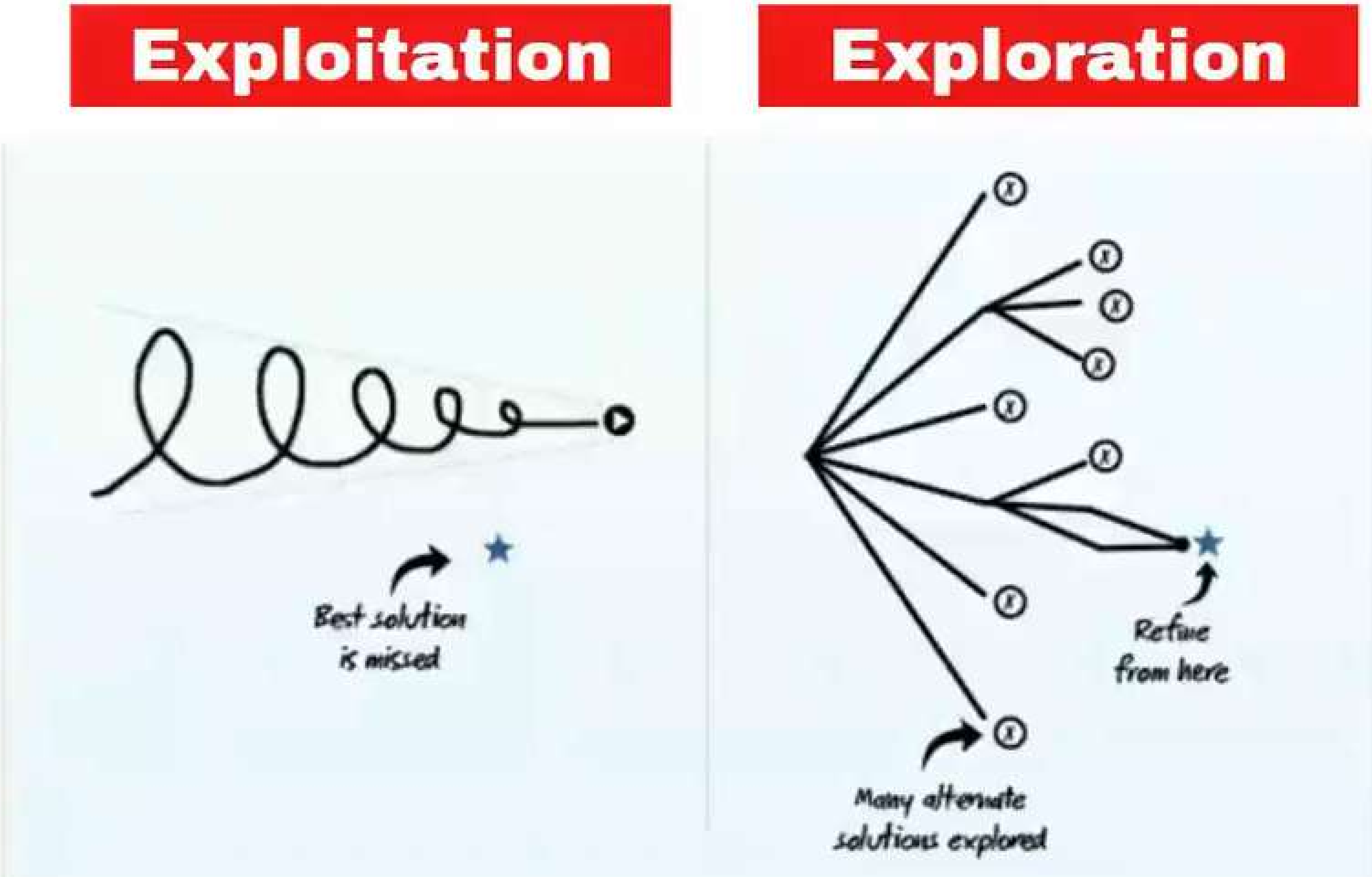


Trained using Black Box Setting (Kinetics 400)

Pretrained on UCF101 dataset

Samples from actual Kinetics-600 Dataset

Exploration Exploitation Analysis



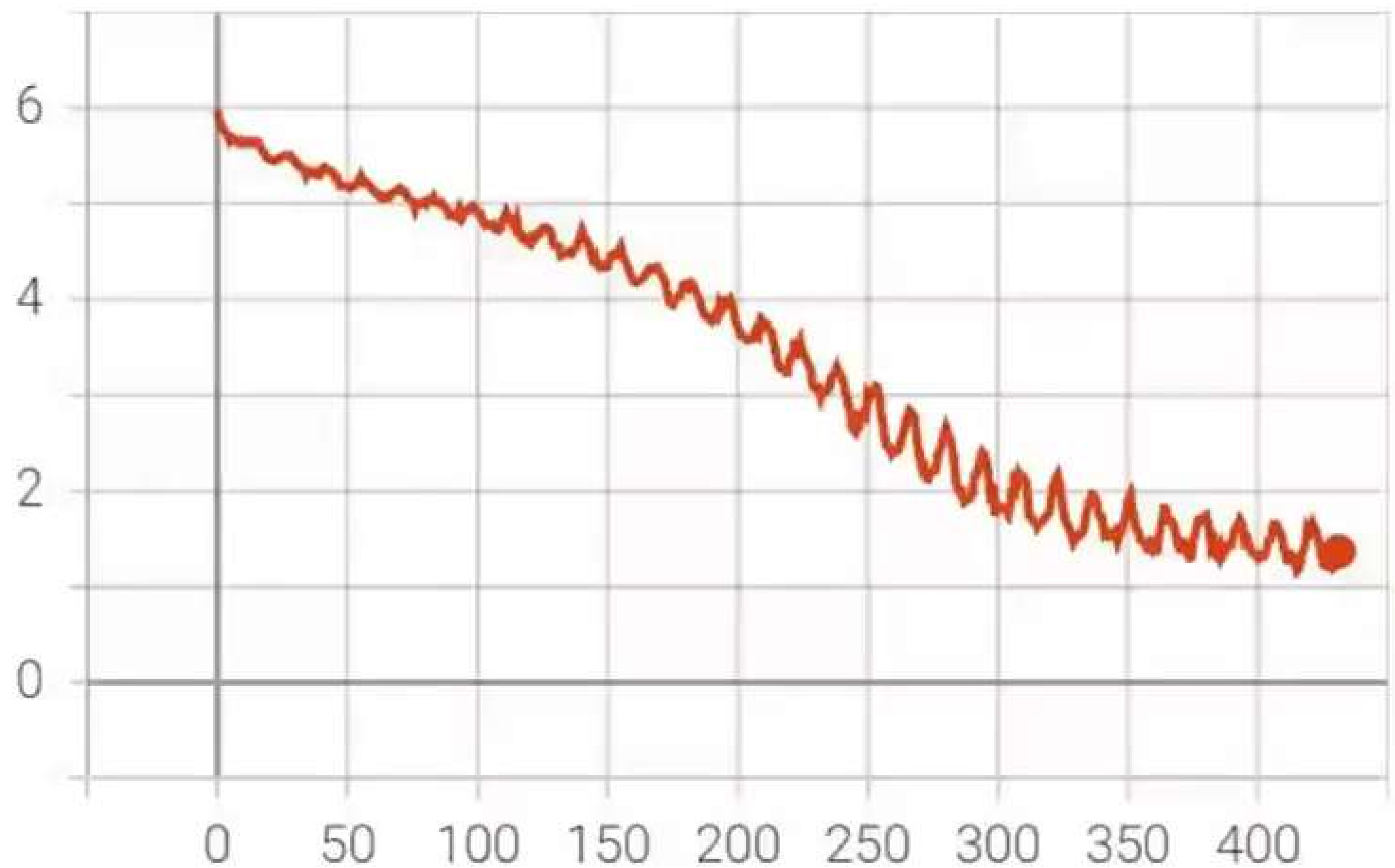
Threshold	Train Accuracy (Kinetics Train Set)	Validation Accuracy (Kinetics Valid. Set)
1	66.01%	5.67%
0.8	65.31%	13.41%
0.6	62.78%	27.85%
0.5	60.21%	57.57%

Threshold

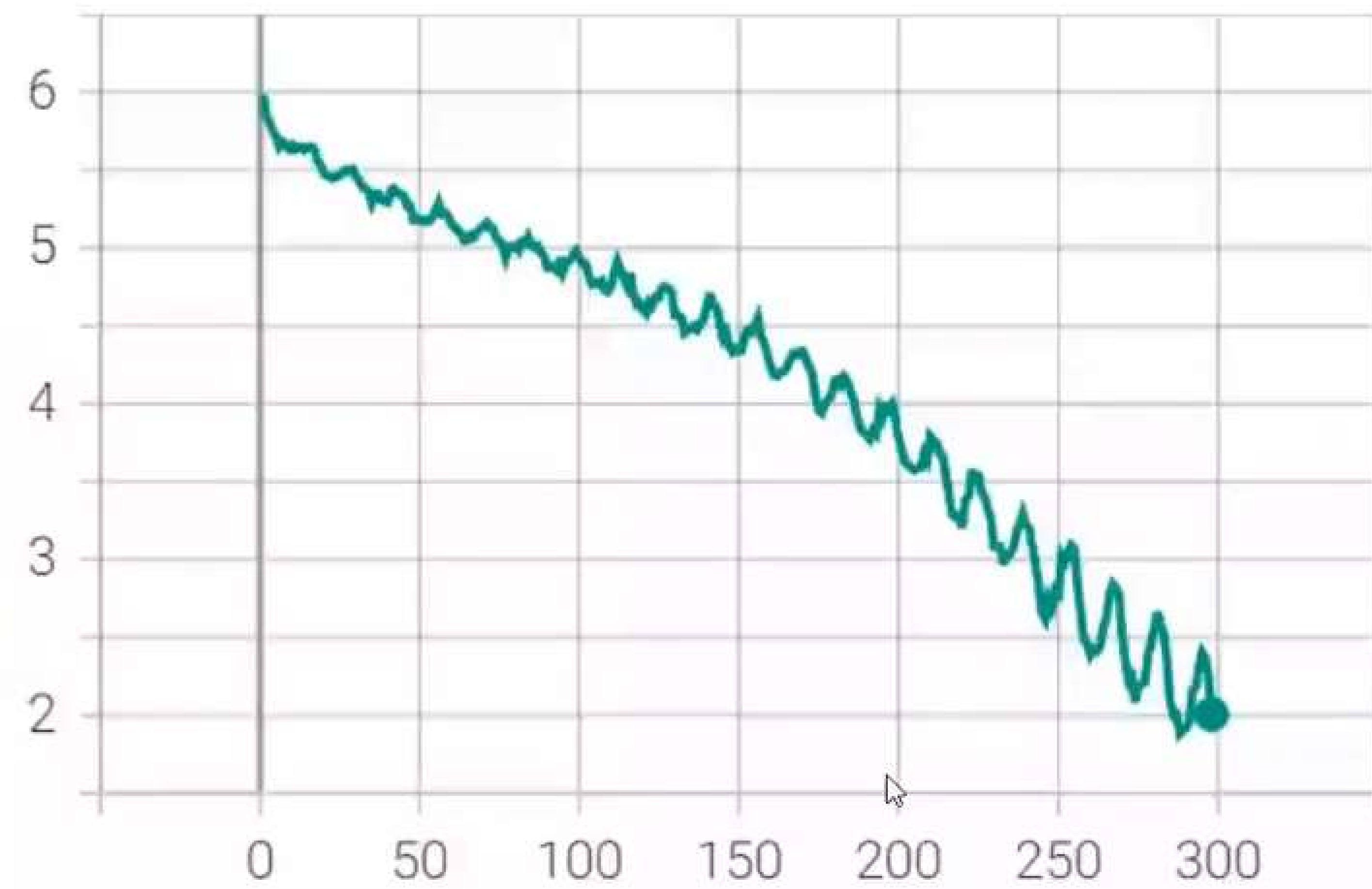
The proposed threshold parameter prevents our model from overfitting on the 5% of the training data available. Using the generator trained on the UCF-101 dataset we are able to generate more diversified queries.

Analysis from the loss curves

Grey Box Setting



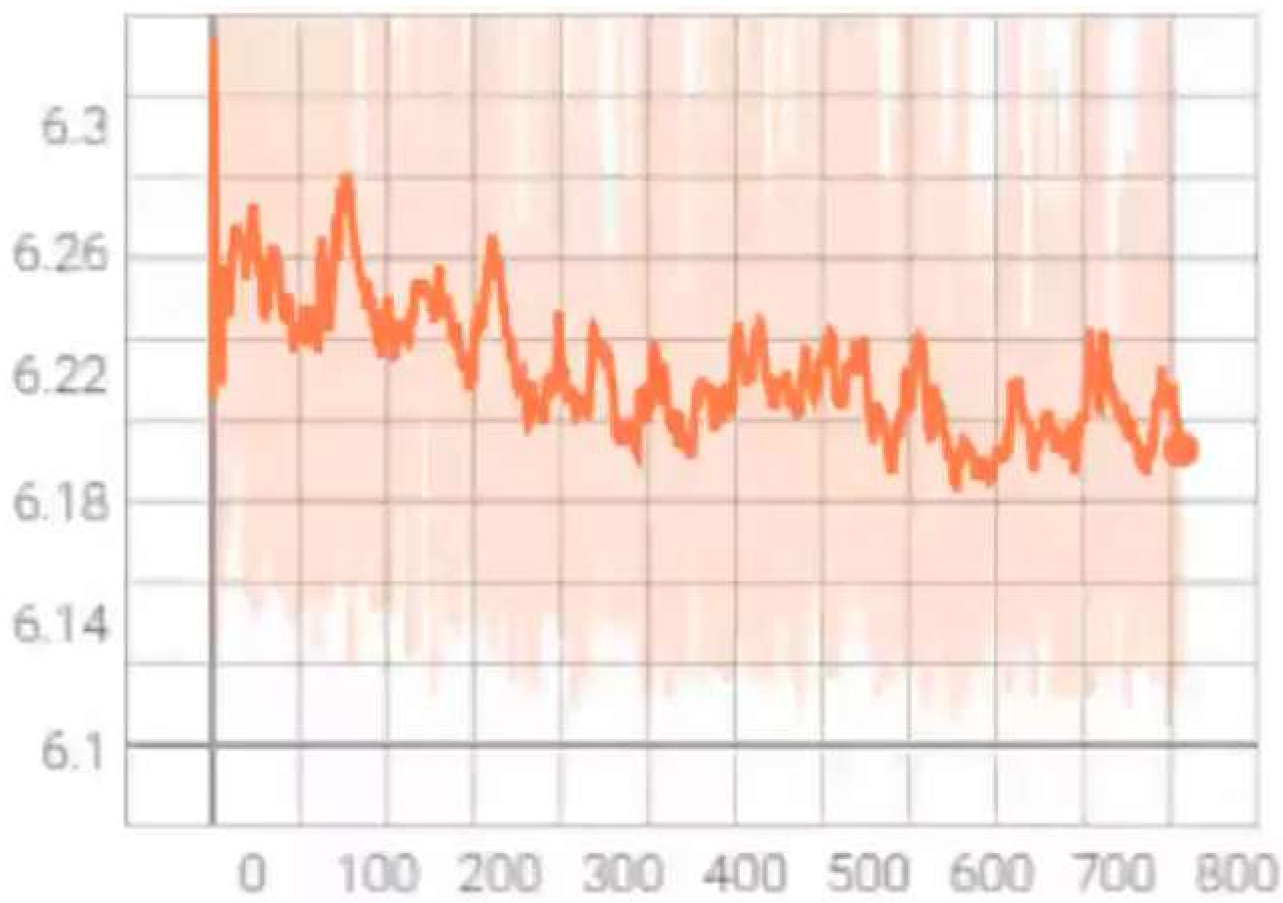
- KL(ViViT|| SwinT)



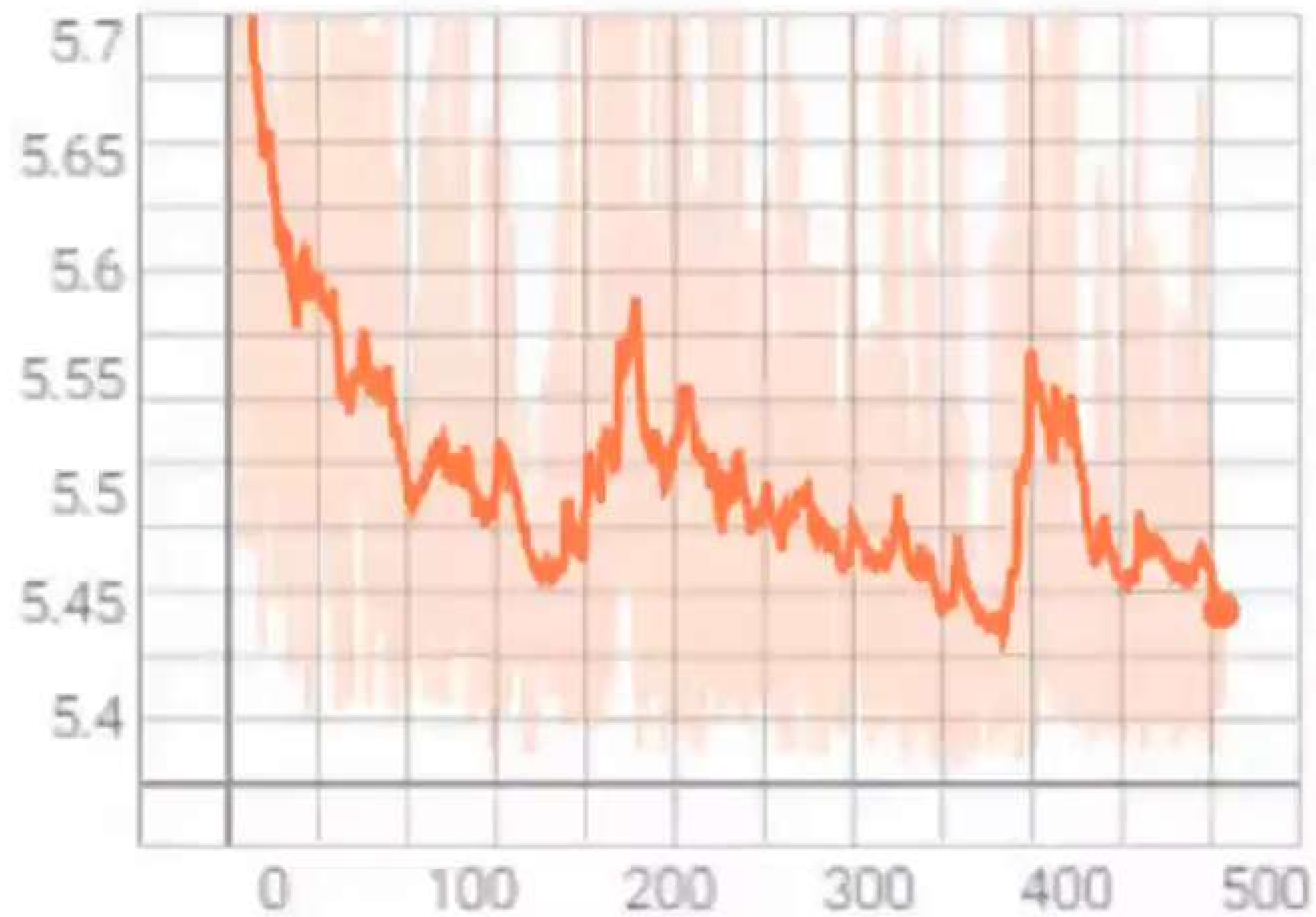
- KL(ViViT||MovieNet-A2)

Analysis from the loss curves

Black Box Setting



- KL(ViViT||SwinT)



- KL(ViViT||MovieNet-A0)

Limitations

Blackbox for Video is very expensive

- In the BlackBox setting, we had to train the generator from random noise in a vast search space. Additionally, the output of the generator was **uninterpretable** which emerged as a breakpoint in analyzing the progress of the training.
- In order to achieve an accuracy of 92%, a simple dataset like **CIFAR-10** required **20M queries**. Comparatively, we have a complex video dataset for classification with **400/600 closely related classes**.

Limitation of accessible compute

- Due to the high computational requirements, we were only able to train each model for **~400 epochs**, which ran for **~70 hrs** parallelly on the **3 GPU**s

Results

Model and Setting	Validation Accuracy	Number of Queries
P1. SwinT		
GreyBox, Student: ViViT	57.57%	0.72 M
Black Box, Student: ViViT	3.59%	1.12 M
P2. MoViNetA2 Base		
GreyBox, Student: ViViT	14.51%	0.80 M
Black Box, Student: ViViT	1.08%	1.20 M

Conclusion

We are **extremely grateful to Bosch** to have provided us with an **interesting problem statement**. We witnessed our **steep growth in the past month**.

Although the development of problem-specific attacks would have ensured a reasonable solution to the given task, we **invested significant time and efforts in ensuring the generality of our solution** and thereby solve the primary objective of this event: to address the challenge of securing AI models in general, which in turn **requires a versatile attack framework** to **exploit a diverse set of models.**

Our results on P1 testify that our proposed solution can indeed **provide exemplary results and a good accuracy** upon careful hyperparameter tuning. Hence, we are **highly confident** that we could **greatly improve our results in P2**, as we were only limited by the time required for further fine-tuning.