Paper Review : Modality-agnostic Automated Data Augmentation in the Latent Space (MODALS)

Anton Zaitsev | Othmane Mahfoud

University of Luxembourg

Problem: Data

- Deep learning models need a lot of **good quality training data**.
- Labelled data is **scarce and expensive**.

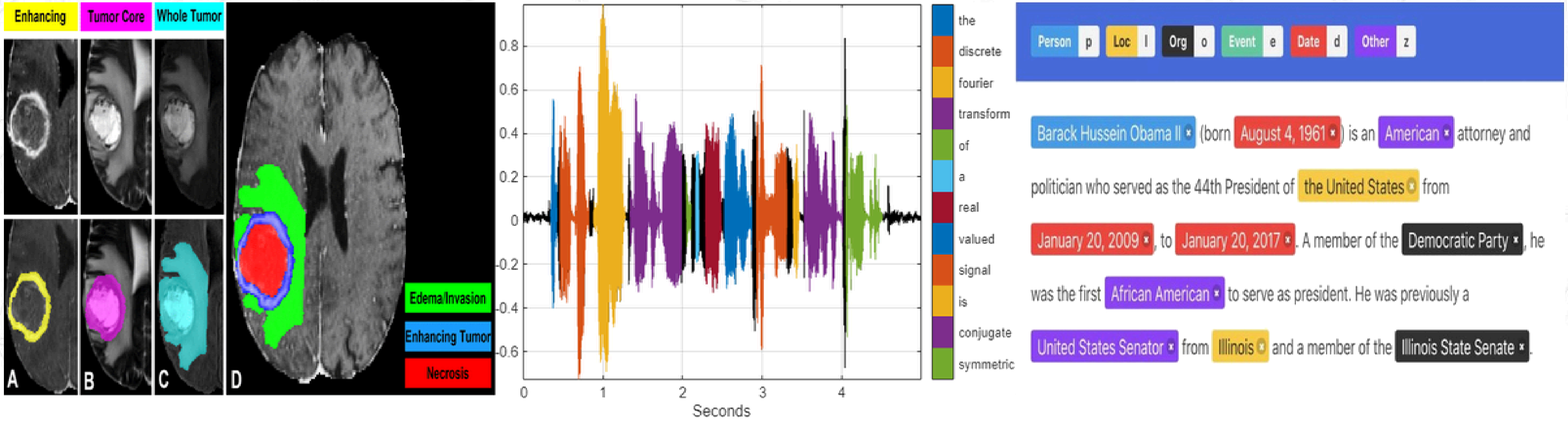


Figure 1: Domains with imbalanced and scare data.

Solution: Data Augmentation

- **Data augmentation** is a way to **increase the size of training datasets**.

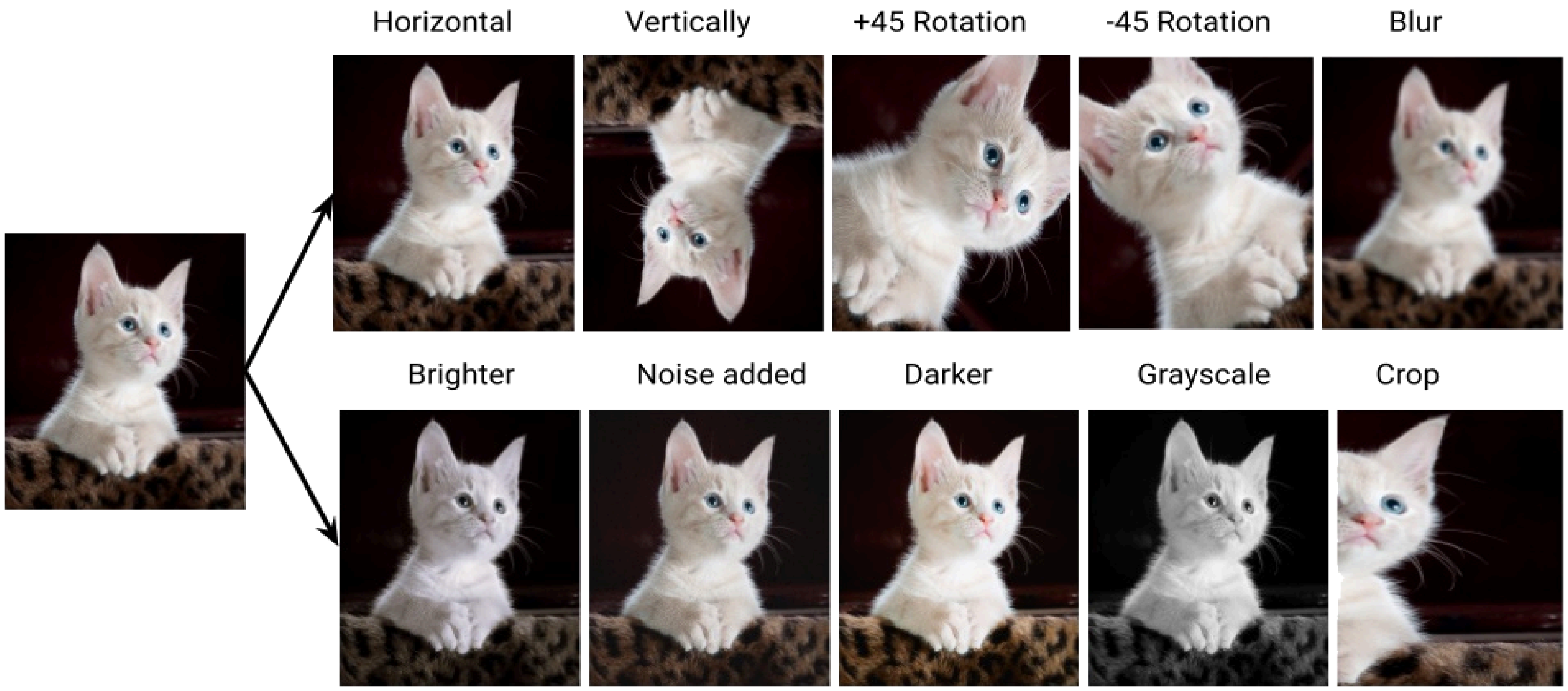


Figure 2: Various types of **image** augmentation techniques.

Data Augmentation Problems

- Data augmentation methods are **manually designed** and evaluated **for different modalities** separately.
- **Modalities are different types of data**, such as images, text, or audio.
- We **cannot use** the **same data augmentation techniques for different modalities**, e.g., images and text.

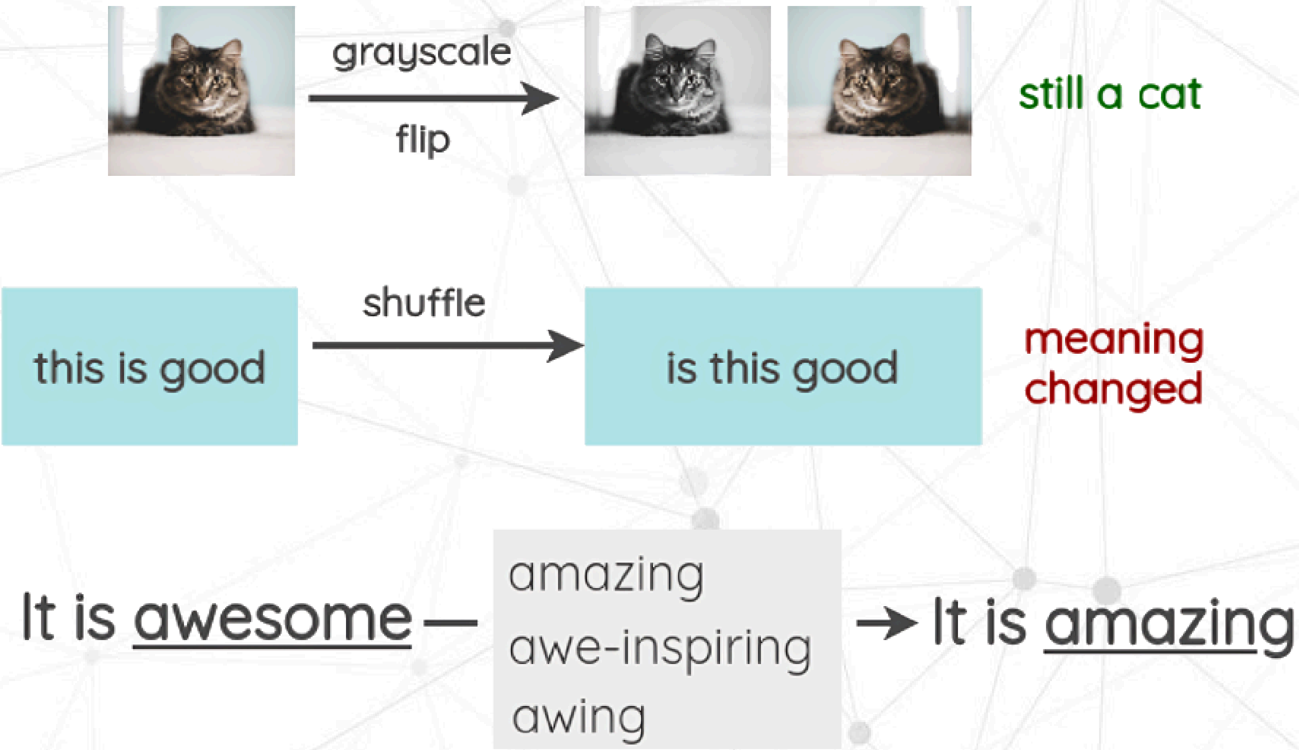


Figure 3: Same data augmentation techniques cannot be used for different data modalities.

Solution: MODALS

- **MODALS: Modality-agnostic Automated Data Augmentation in the Latent Space.**
- MODALS transforms data in the **latent space**.
- MODALS works with **any data modality**.

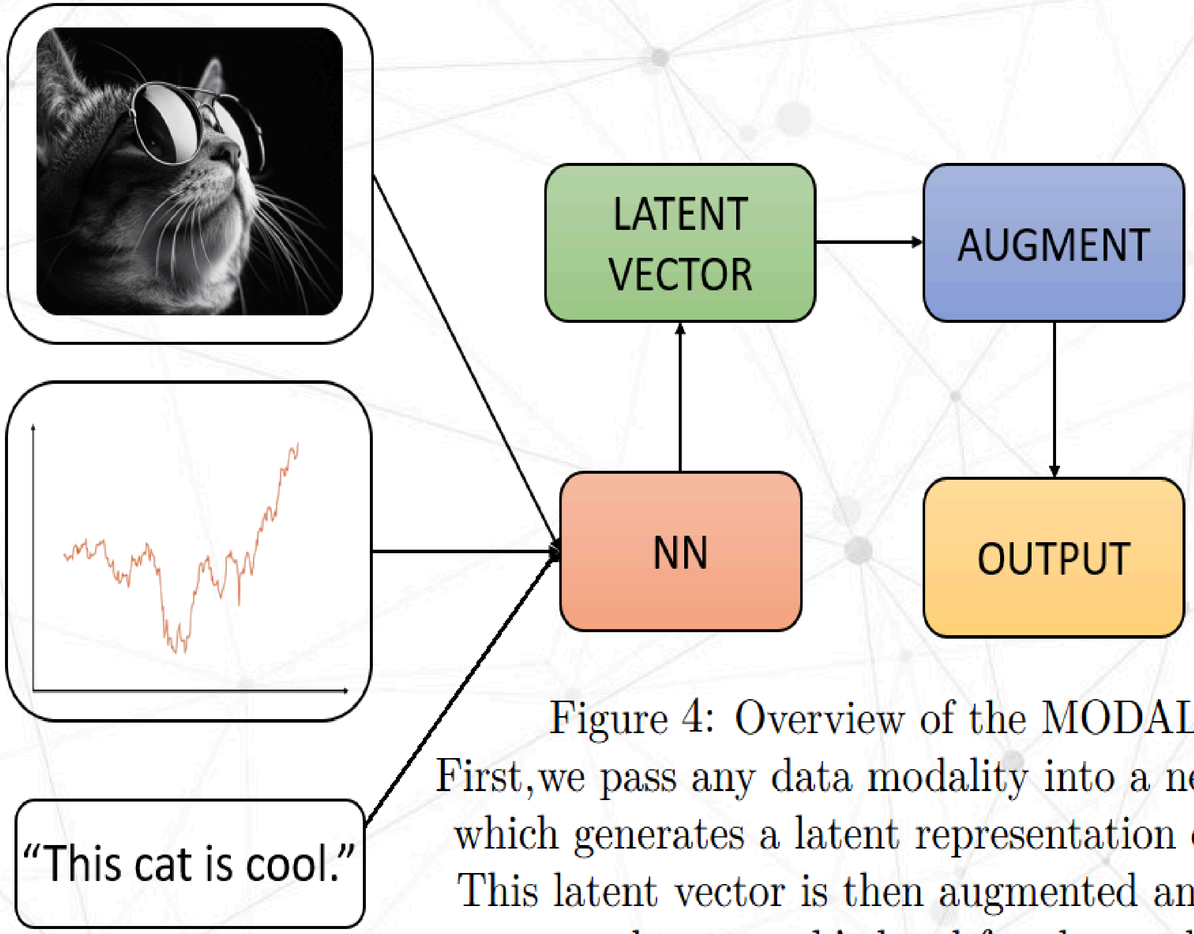


Figure 4: Overview of the MODALS idea. First,we pass any data modality into a neural network, which generates a latent representation of the input. This latent vector is then augmented and forwarded to the network's head for the prediction.

MODALS Results Teaser

- MODALS was tested on **text, tabular, time-series, and image** datasets.
- It **outperformed** baseline methods **in most cases**.

Dataset	w/o Aug.	Mixup	MODALS
HAR (s)	49.93	52.20	**57.71**
HAR (m)	85.44	86.05	**89.06**
HAR	88.64	91.60	**91.87**

Dataset	w/o Aug.	Mixup	MODALS
Malware (s)	79.24	81.14	**84.07**
Malware (m)	84.76	85.57	**86.24**
Malware	86.40	87.01	**87.12**

Table 1: Comparison of two baselines (training without augmentation and Mixup) with MODALS on two tabular datasets for different amounts of training examples (s: 10%, m: 50%)

MODALS Algorithm: Overview

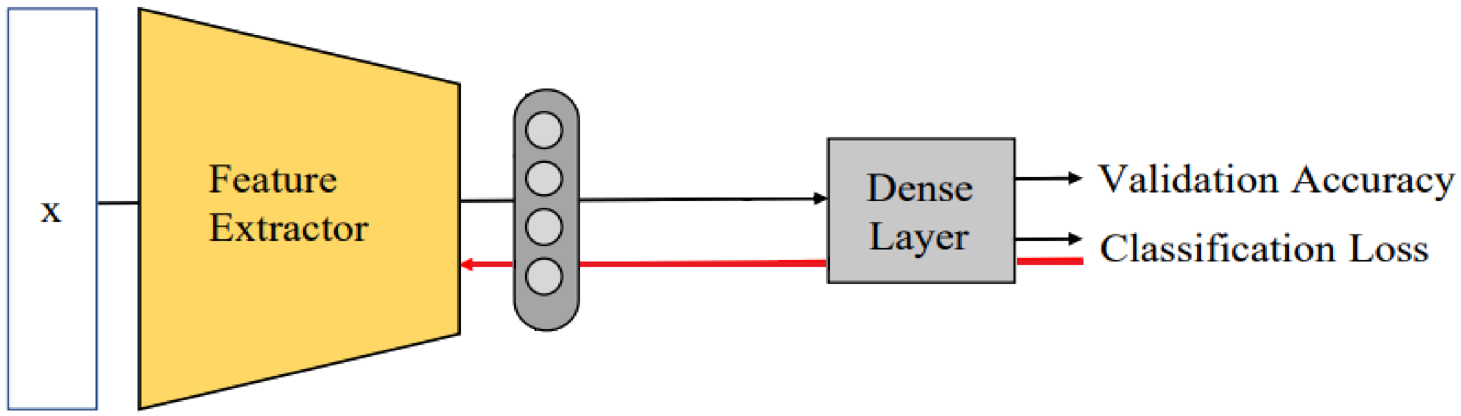


Figure 5: Simple machine learning pipeline.

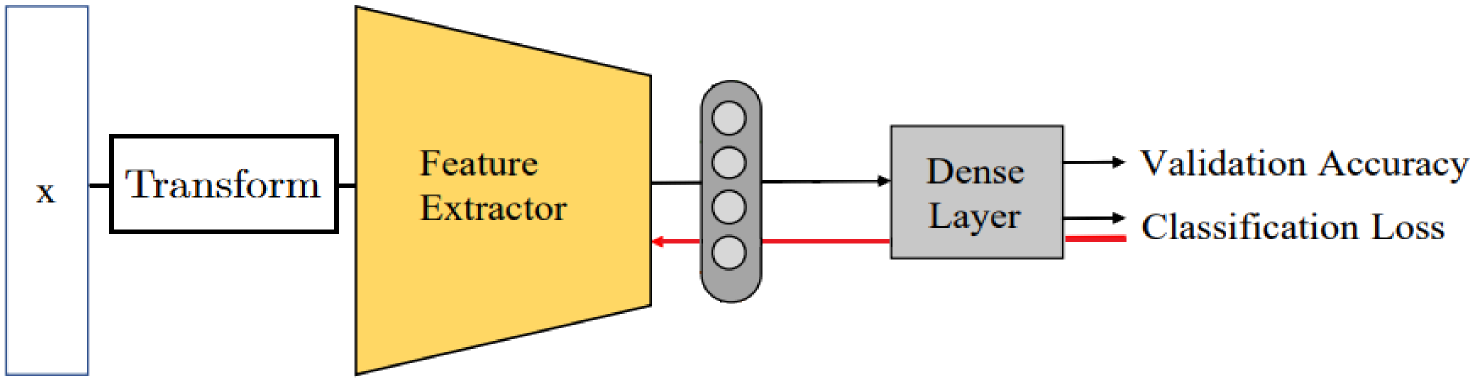


Figure 6: Simple machine learning pipeline with **transformed** (augmented) input data.

MODALS Algorithm: Overview

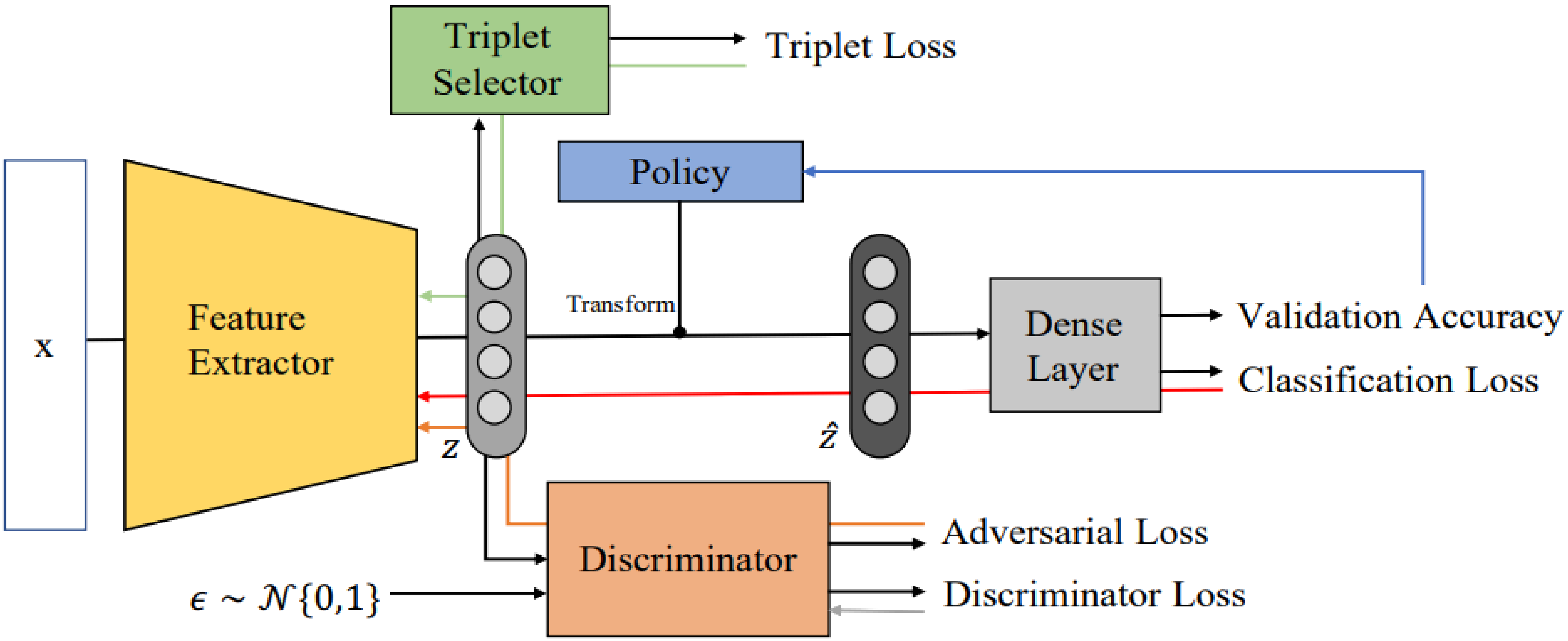


Figure 7: MODALS (z: seed latent representation; $\hat{z}$: augmented latent representation; black line: forward propagation; red line: gradient flow from L_{clf}; green line: gradient flow from L_{tri}; orange line: gradient flow from L_{adv}; blue line: reward signal; grey line: gradient flow from L_D.)

MODALS Algorithm: Step 1

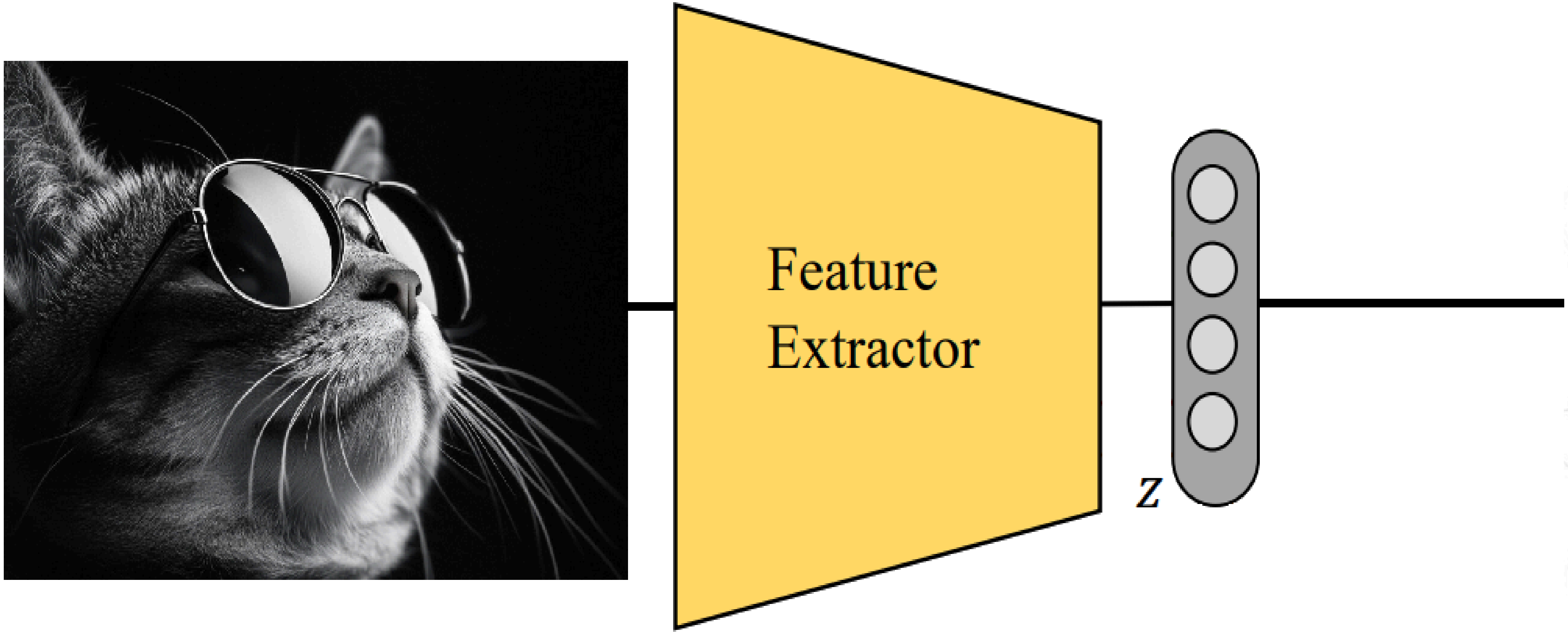


Figure 8: We pass data through a feature extractor, obtaining a latent representation vector z.

MODALS Algorithm: Step 2

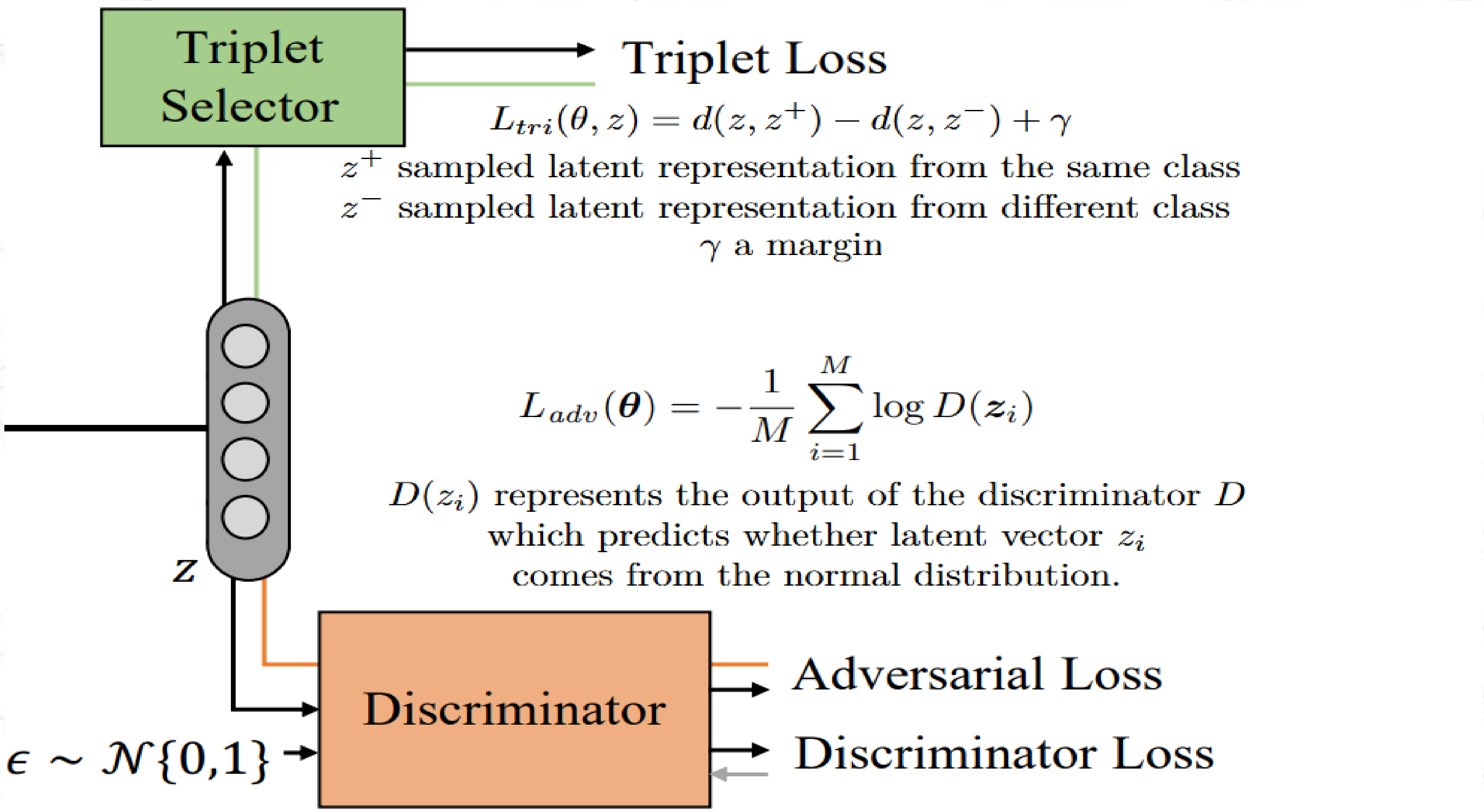
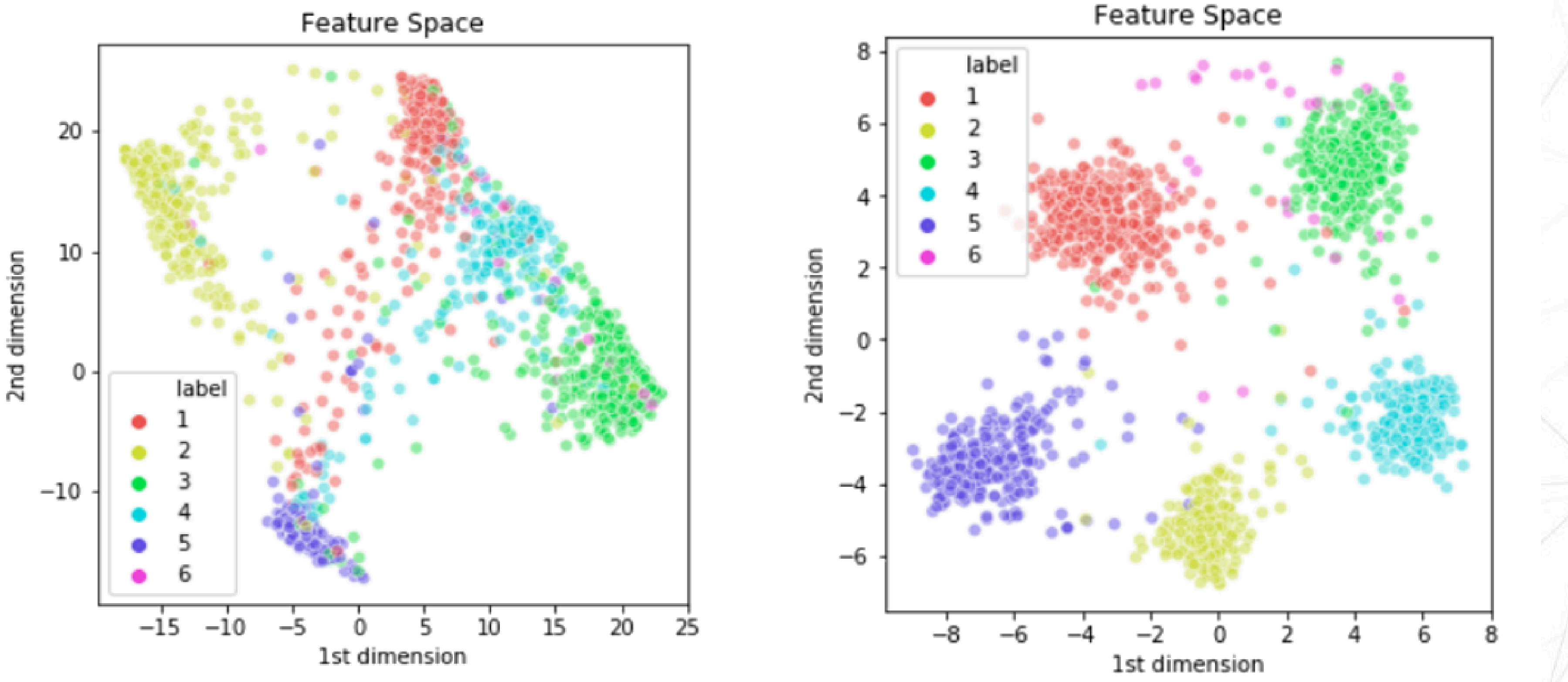


Figure 9: Latent vector z is passed through Triplet Selector and Discriminator.
This encourages the model to learn the best
latent representation of the input data for later augmentation.

MODALS Algorithm: Step 2, Triplet Loss

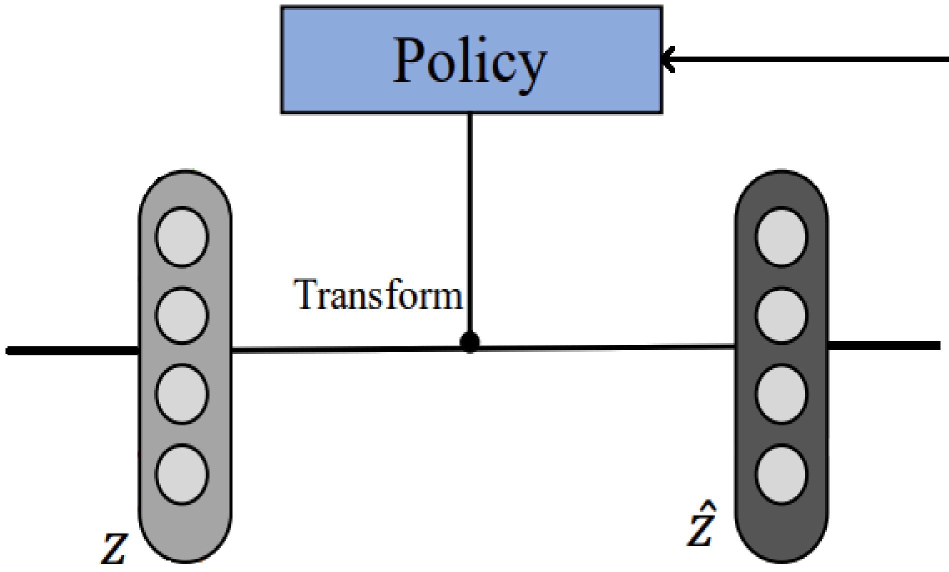


(a) Without triplet loss

(b) With triplet loss

Figure 10: Visualization of data points in the latent space for the TREC6 dataset.

MODALS Algorithm: Step 3



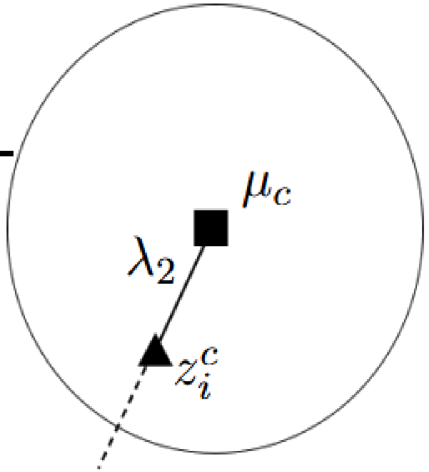


Figure 11: Hard example extrapolation.
Instead of extrapolating from random examples,
we extrapolate z_i^c from the class center $\mu_c = \frac{1}{m}\sum_{j=1}^{m} z_j^c$
with λ_2 as the scaling factor:

$$\hat{z}_i^c = z_i^c + \lambda_2(z_i^c - \mu_c)$$

MODALS Algorithm: Step 4

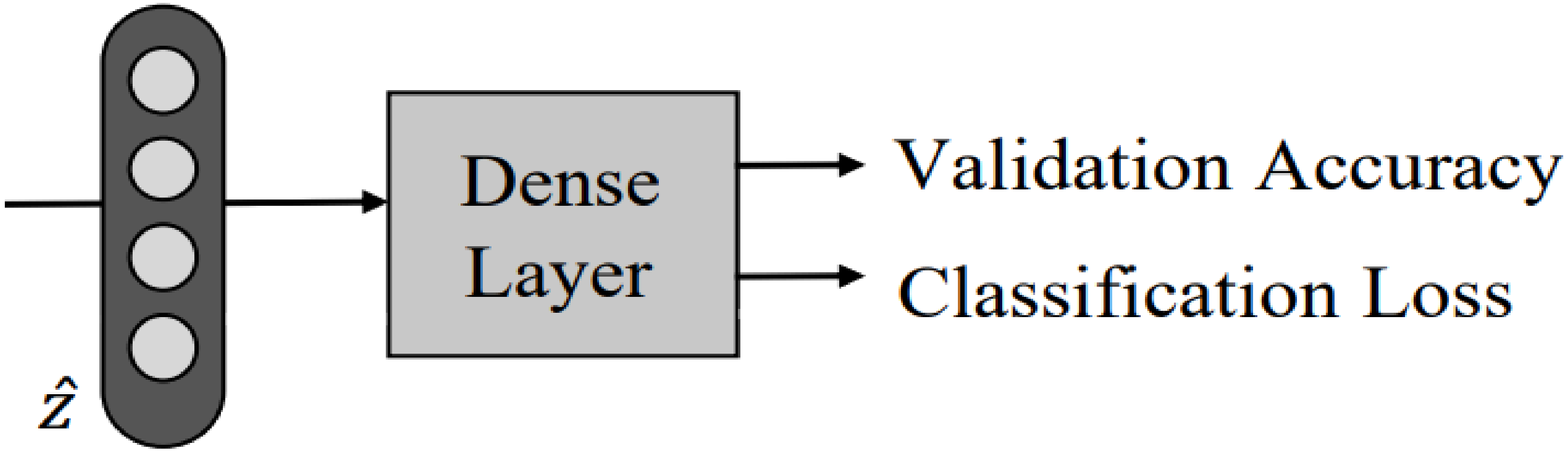


Figure 12: We pass $\hat{z}$ to a neural network head.

MODALS Algorithm: Step 5

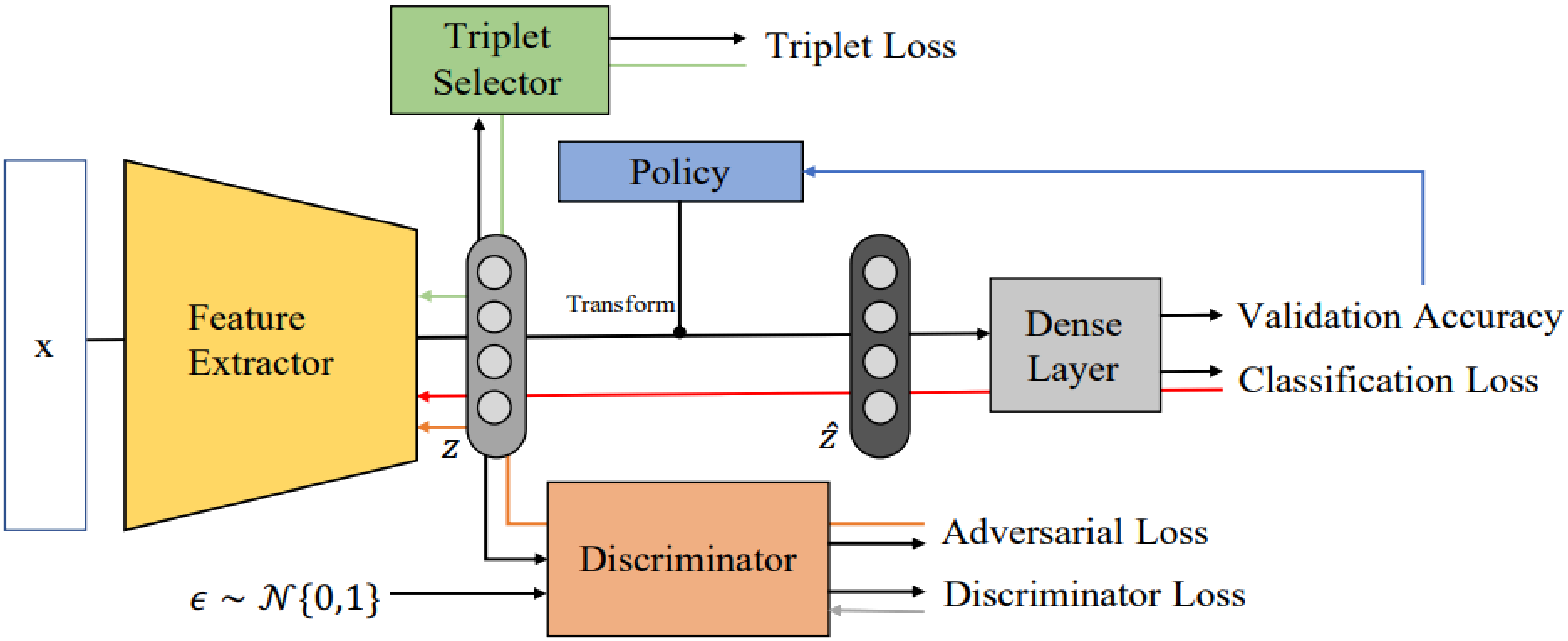


Figure 7: MODALS (z: seed latent representation; $\hat{z}$: augmented latent representation; black line: forward propagation; red line: gradient flow from L_{clf}; green line: gradient flow from L_{tri}; orange line: gradient flow from L_{adv}; blue line: reward signal; grey line: gradient flow from L_D.)

MODALS Results

Data Type	Metric Type	MODALS	Second Best	Percentage Change (%)
Tabular	Accuracy	84.64	83.17	**1.74**
Text	Accuracy	85.23	84.21	**1.26**
Time-Series	Accuracy	82.68	80.60	**3.14**
Image	Accuracy	87.39	87.86	**-0.55**

Table 2: Performance metrics for MODALS across different datasets and data types (averaged).

	L_{clf}	$L_{clf} + L_{adv}$	$L_{clf} + L_{tri}$	$L_{clf} + L_{adv} + L_{tri}$
w/ Aug.	86.35	88.18	88.37	**89.12** (MODALS)
w/o Aug.	84.71	86.64	86.74	87.69

Table 3: Comparison of average accuracy when trained under different loss settings and augmentation techniques

Conclusions

- MODALS is an **automatic data augmentation technique**.
- MODALS can be used for **any data type**, since it transforms data in the **latent space**.
- MODALS achieves **best results** on text, tabular, time-series data and **competitive results** on image data (compared to baseline augmentation techniques).
- MODALS can be **easily intergrated** into any **deep learning pipeline**.
- MODALS might be the **best solution** yet for:
  - Data modalities where input space **augmentation** is **difficult to define**.
  - **Scarce** and **imbalanced datasets**.

How MODALS can Impact the ML World?

- Enable deep learning for **new data modalities**.
- Advance deep learning for domains with **difficult data modalities** (e.g., medical imaging or NLP for low-resource languages).
- Boost efficiency in **multi-modal** applications.

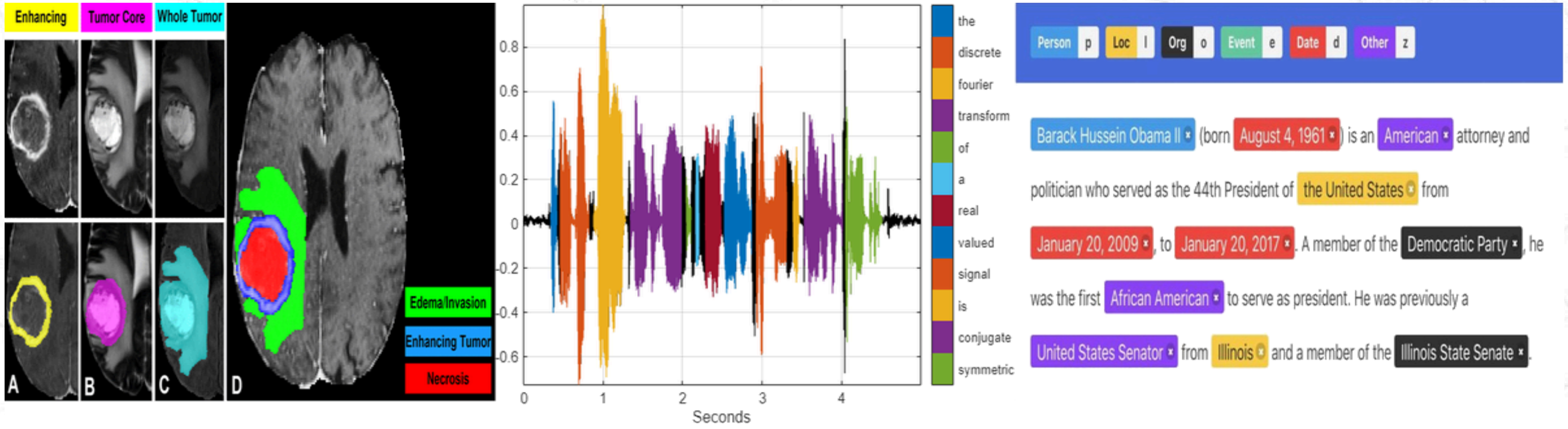


Figure 1: Domains with imbalanced and scare data.

Thank you