# Stitch

AML Challenge Report 2025/26

Davide Perniconi
Matricola: 1889270

Daniele Marretta
Matricola: 1985747

Olga Corovencova
Matricola: 2249558

Leonardo Lavezzari
Matricola: 1984079

## 1 Proposed Method

We used a *contrastive approach* in order to learn a mapping function $T : R^{1024} \implies R^{1536}$ able to correctly align the space of image embeddings (`DINOv2-giant`) with text embeddings (`roberta-large-nli-stsb-mean-tokens`).

- **Architecture:** Our model is a MLP network with the following configuration:

  1. *Input dimension:* 1024
  2. *Hidden layers:* $[1256, 2048]$
  3. *output dimension:* 1536
  4. *Activation:* Gelu
  5. *Dropout:* 0.5
  6. *Learnable Temperature parameter*
  7. *Inizialization:* Xavier uniform
  8. *Normalization:* The final embedding is L2-normalized

- **Loss Function:** We augmented the InfoNCE loss with a memory bank, drawing inspiration from the **MoCo**[1]. Unlike standard MoCo, a query encoder is trained, and a momentum-updated key encoder provides key representations. In our setup, this dynamic is adapted for multimodal retrieval:

  1. **Query Encoder:** This role is fulfilled by our **MLP**. This is the only component updated by backpropagation.
  2. **Key Encoder:** This role is fulfilled by the pretrained **DINO** image model. We treat its embeddings as the key targets.
  3. **Memory Bank:** A FIFO queue that stores key vectors from prior mini-batches. This bank provides a large, diverse set of negative samples without requiring a prohibitively large batch size, which is the core benefit of the MoCo approach.

- **Training Details:** We trained the model with AdamW optimizer (learning rate 0.01, weight decay $1 \times 10^{-5}$) using batches of 256. A ReduceLROnPlateau scheduler, adjusted the learning rate based on validation loss. The size of the queue is 10,000 and we used early stopping when validation loss stagnated. Validation performance was tracked using MRR and Validation Loss. All hyperparameters, including the queue size, learning rate, hidden layer sizes, and dropout rate, were selected via hyperparameter optimization using Optuna [2].

## 2 Results and Discussion

Our final model achieved an MRR of 0.874 in the public leaderboard.
On the validation set, the model obtained:
MRR 0.942, Recall@1 0.906, Recall@3 0.973, NDCG 0.956
These results outperform the baseline provided by the challenge, which reports an MRR of approximately 0.462. This method effectively translates between text and image embedding spaces as the projection network maps text embeddings into the image space to make paired embeddings close. $L^2$-normalization stabilizes training and prevents scale differences from dominating the loss. The InfoNCE objective encourages alignment of positives and separation from a large set of negatives, with the memory queue providing diverse negative examples without large batches. However, InfoNCE embeddings are less precise than those from MSE or cosine similarity loss because they prioritize pushing positive pairs closer relative to negatives, rather than minimizing the absolute distance of a single positive pair.

## 3 Conclusion

Our method, using an MLP trained with InfoNCE loss achieved an MRR of 0.874 on the public leaderboard. This contrastive approach learned the mapping between the two spaces, balancing retrieval performance against the inherent precision trade-offs of the InfoNCE objective.
LINK TO THE REPO

# 4 What We Tried

During the competition we found models that did not outperform our final MoCo-based method, instead they provided valuable insights and contributed to our overall understanding of the problem.

## Method 1: Simple linear mapping with reference to zero shot translation

Our first approach, based on the principle of zero-shot translation [3], involved a simple linear mapping to align the text embeddings with the corresponding image embeddings. This baseline was used to directly translate between the two modalities in their latent spaces. However, the model performed poorly, suggesting the relationship between the text and image embedding spaces is highly non-linear and requires a more complex translation function.

## Method 2: Inverse Pseudo-Transform

Our second strategy leveraged the Inverse Pseudo-Transform (IPT) method [4], which first uses anchor points to project the data into a common, lower-dimensional intermediate space ( We then trained a linear transformation matrix ($W$) by solving a regularized least-squares problem ($\mathrm{X}_{\mathrm{rel}}^T \mathrm{X}_{\mathrm{rel}} + \lambda I$) to directly map the projected text embeddings ($\mathrm{X}_{\mathrm{rel}}$) to the projected image embeddings ($\mathrm{Y}_{\mathrm{rel}}$). The final translated embeddings were then reconstructed back to the original image space using the inverse of the anchor-based transform. In the end, the alignment of $\mathrm{X}_{\mathrm{rel}}$ and $\mathrm{X}_{\mathrm{rel}}$ didn't work very well, so the method performed badly.

## Method 3: MLP

We tried to train a MLP using Mean Squared Error Loss and/or cosine distance to minimize the difference between the predicted and actual image embedding vectors. We explored two primary architectures: a simple MLP and a deeper MLP incorporating residual connections to potentially capture more complex relationships between the two embedding spaces. This approach did not achieve good results, indicating that a simple direct mapping function is insufficient to capture the relationship between the two distinct embedding spaces.

## Method 4: Two-Head MLP with Memory Bank (Second best model)

We explored a more sophisticated approach inspired by vector decomposition. The model consists of two different heads, one for the direction and one for the magnitude of the output:

- a **Direction Head** predicting a unit-norm embedding
- a **Scale Head** predicting a positive scalar via Softplus

The output embedding is computed as

$$T(x) = \mathrm{normalize}(d(x)) \cdot s(x).$$

Training was performed using the same InfoNCE loss of the original model, with a large negative queue of 32,000 samples. The model achieved competitive retrieval performance (MRR $\approx 0.871$) on the public leaderboard. However, it did not surpass the simpler main approach, likely because scale modulation introduces additional instability in contrastive optimization.

## Method 5: Flow match, Vector field model with MLP

[5]
We attempted to use a Flow Matching approach, modeling the transformation from text to image embeddings with an MLP that defined a Vector Field.
However, this sophisticated model architecture only achieved a maximum MRR of 0.76, indicating that the predicted image embeddings were still not accurate enough for effective cross-modal retrieval.

# References

[1] Kaiming He et al. *Momentum Contrast for Unsupervised Visual Representation Learning*. 2020. arXiv: 1911.05722 [cs.CV]. URL: https://arxiv.org/abs/1911.05722.

[2] Takuya Akiba et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. 2019. arXiv: 1907.10902 [cs.LG]. URL: https://arxiv.org/abs/1907.10902.

[3] Valentino Maiorca et al. *Latent Space Translation via Semantic Alignment*. 2024. arXiv: 2311.00664 [cs.LG]. URL: https://arxiv.org/abs/2311.00664.

[4] Valentino Maiorca et al. *Latent Space Translation via Inverse Relative Projection*. 2024. arXiv: 2406.15057 [cs.LG]. URL: https://arxiv.org/abs/2406.15057.

[5] Yaron Lipman et al. *Flow Matching for Generative Modeling*. 2023. arXiv: 2210.02747 [cs.LG]. URL: https://arxiv.org/abs/2210.02747.