Lilo & Stitching

AML Challenge Report 2025/26

Ciro Ciaravolo
Matricola: 1938321

Daorui Dong
Matricola: 2244070

Lorenzo Musso
Matricola: 2049518

Giulia Pietrangeli
Matricola: 2057291

1 Proposed Method

Our final method is a **dual-submission strategy** based on weighted ensembles of three distinct adapter models, all mapping 1024-dim RoBERTa embeddings to the 1536-dim DINOv2-based image space.

Architecture: The final embedding z_{final} is a weighted average of the L2-normalized output of each component model (z_{RMLPA}, z_{MLP}, $z_{Stitcher}$). The resulting vector is L2-normalized again before retrieval.
The three component models are:

1. **RMLPA:** A Residual Bottleneck Adapter, which maps $D_{in} \to D_{bottle} \to D_{out}$ ($D_{bottle} = D_{in}/4$) with a residual shortcut connection.

2. **MLP (LatentMapper):** A single linear layer that maps centered text embeddings to the image space. It is initialized using Orthogonal Procrustes analysis on the training data.

3. **Stitcher:** A parallel-path model that sums the output of a direct linear projection with that of a deep, non-linear MLP path.

Loss Function: The models were trained independently using two different loss functions.

- The **RMLPA** model was optimized using a symmetric InfoNCE (contrastive) loss with a learnable temperature parameter.

- The **MLP** and **Stitcher** models were both optimized using a Triplet Margin Loss (margin $m = 0.2$) with in-batch hard negative mining.

Training Details: All models were trained using the AdamW optimizer and a Cosine Annealing learning rate scheduler. The optimal ensemble weights were determined on our internal validation set using two different methods: a **linear search** for the 2-model ensemble, and a **calculation based on the inverse of a pre-computed overfitting metric** for the 3-model ensemble. To prevent group leakage, all data splits used *GroupShuffleSplit* to ensure reliable validation metrics.

Submissions: We obtained two distinct submissions:

- **Submission 1 (2-Model):** A weighted average of RMLPA and Stitcher.

  $$z = \alpha \cdot z_{RMLPA} + (1 - \alpha) \cdot z_{Stitcher}$$

  (Optimal $\alpha = 0.20$).

- **Submission 2 (3-Model):** A weighted average of all three models.

  $$z = w_r \cdot z_{RMLPA} + w_m \cdot z_{MLP} + w_s \cdot z_{Stitcher}$$

  (Optimal weights: $w_r = 0.332, w_m = 0.057, w_s = 0.611$).

2 Results and Discussion

Our ensemble strategy proved to be effective, resulting in two distinct submissions. Both 2-model and 3-model ensembles were built on the strong performance of the individual models and submitted as highly competitive candidates. The internal test scores for the single models, calculated using a custom function which evaluates a sampled gallery for efficiency, were:

- **RMLPA:** 0.8625
- **MLP (LatentMapper):** 0.8840
- **Stitcher:** 0.8820

The success of this method comes from leveraging **model diversity**. By combining models with different architectures (residual, linear, parallel) and, crucially, different loss functions (InfoNCE vs. Triplet), the ensembles average out uncorrelated errors and produce a more robust mapping.

3 Conclusion

We present a dual-submission strategy based on two weighted ensembles. This approach highlights that combining adapters with diverse architectures and loss functions is an effective technique to improve generalization and achieve robust, competitive results in cross-modal retrieval.

Go to the GitHub repository

What We Tried

Method 1: RMLPA (Residual Adapter)

We implemented a Residual Bottleneck Adapter (bottleneck ratio $D_{in}/4$) to learn the mapping. It was trained using a symmetric InfoNCE loss with a learnable temperature. This model served as a strong baseline on our internal test set.

Method 2: MLP with Procrustes Initialisation

We explored a simpler approach: a single linear layer ('LatentMapper'). The key was to initialize its weights using the Orthogonal Procrustes analysis solution. This gave an excellent starting point, which was then fine-tuned with Triplet Loss.

Method 3: Stitcher (Parallel Adapter)

This model was designed to combine the strengths of a simple linear map and a deep, non-linear MLP. It sums the outputs of these two parallel paths and was trained with Triplet Loss.

Method 4: Ensemble Strategy (Final)

Recognizing that each model learned a slightly different mapping, our final strategy was to combine them. We saved the validation embeddings from each model and found the optimal weights. This led to our two submissions:

- a 2-model ensemble ($\alpha = 0.20$)
- e a 3-model ensemble ($w_r = 0.332, w_m = 0.057, w_s = 0.611$)

Other Approaches Explored

In addition to the models that formed our final ensemble, we investigated several other approaches that ultimately yielded limited results and were not included in the final submissions.

- **Simple Algebraic Mappings:** Inspired by work on latent space translation [1], we first experimented with simple, closed-form algebraic solutions. This included **linear** (Least Squares), **affine**, and **orthogonal** (Procrustes analysis) transformations to map RoBERTa embeddings directly to the DINOv2 space. These methods provided a weak baseline but failed to capture the complex non-linear relationship between the modalities.
- **Flow Matching:** We attempted to adapt the **Flow Matching** framework, typically used for generative modeling [2], for our cross-modal retrieval task. The goal was to learn a continuous vector field that transports the text embedding distribution to the image embedding distribution. However, this approach proved difficult to stabilize and did not converge to a competitive solution.
- **Transformer-based Adapters:** We also replaced our lightweight MLP adapters with heavier models, including a small **Transformer encoder** stack. We hypothesized that the self-attention mechanism could better refine the text embeddings for the image space [3]. These models failed to adapt effectively to the objective, showing suboptimal results and poor generalization compared to the simpler, parameter-efficient adapters.
- **Generative Text-to-Image Pipeline:** We attempted to use the **'flux' model** to generate images from text, which were then encoded by **DINOv2**. This yielded a low MRR score (0.7). Despite visual consistency, the resulting embeddings failed to align with the ground-truth, likely because the generative model is optimized for human vision, not for this specific latent space retrieval task.

References

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

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

[3] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.