Jaguar Re-Identification Challenge Report

Applied Hands-On Computer Vision

Josef Pribbernow Vincent Eichhorn

January 2026

Abstract

ABSTRACT

Contents

1 Introduction

The challenge is to identify individual jaguars based on their unique spot patterns. Therefore this task is a fine-grained classification task which is critical for conservation efforts, as it automates the process of tracking populations across vast areas like the Pantanal.

1.1 Data Challenges

The jaguar re-identification task presents several data challenges. Intra-class variation occurs as individual jaguars appear differently across images due to changes in lighting, posture, or camera angle. This requires the model to generalize features effectively for the same individual under diverse conditions. Inter-class similarity further complicates the task, as different jaguars may share similar spot patterns or physical traits. This increases the risk of misidentification, demanding highly discriminative feature learning. Class imbalance in the dataset, characterized by a long-tail distribution, means some jaguars are well-represented with numerous images, while others have limited samples. This imbalance can bias models toward frequently seen individuals, reducing performance on rare ones. Additionally, the images contain the background of the jaguars environments which poses the risk of spurious correlations. The model can rely on background features to identify a jaguar. However the images are segmented using ther Segment Anything Model 3 (SAM). Therefore this risk can be minimized so the model relies only minor on irrelevant environmental cues rather than the jaguar's unique patterns, in the case of removed overlaying foreground.

1.2 Conservation Impact

By automating jaguar identification, this model supports calable population monitoring, enabling researchers to track movement patterns, territory use, and population trends. The methods developed here can also be adapted to other spotted species, contributing to broader wildlife conservation efforts.

2 Baseline

The basline model leverages metric learning to map images into an embedding space where similar individuals are clustered together, while dissimilar ones are separated. The model uses a foundation model to embed the images of jaguars in order to extract features of the jaguars appearance. To adapt the general extracted features to resemble the concrete identity information of the individual jaguars.

2.1 Backbone Embedding Model

The MegaDescriptor model is used as a fixed feature extractor. It is pre-trained on diverse wildlife datasets and excels at capturing fine-grained visual features, making it ideal for distinguishing subtle differences in jaguar spot patterns.

Jaguar identification relies on unique rosette patterns of the jaguars skin, which are visually complex and require high-dimensional representations. MegaDescriptor provides dense, discriminative embeddings that encode these patterns effectively. To optimize efficiency, features are pre-extracted and cached, reducing the computational overhead during training and further experimentation. This approach ensures that the model focuses on learning task-specific adaptations rather than recalculating generic features repeatedly.

2.2 Projection Head

The experiment baseline uses a Multi-Layer Perceptron (MLP) attached to the backbones output. This projection head adapts the generic features to the jaguar re-ID task.In contrast to the frozen backbone the projection head is trainable. In concrete it consists of linear layers to map backbone features to a hidden dimenstion and furthermore to the output embedding dimension, batch normalization and ReLU for stability and non-linearity and a dropout layer to prevent overfitting. By introducing non-linearity and regularization, it ensures the embeddings are optimized for discriminative power. The L_2 normalization of the output embeddings places them on a hypersphere, which is essential for angular-based loss function ArcFace. This normalization ensures that similarity is measured by angular distance, aligning with the intuition that visually similar jaguars should have embeddings that are close in angular space.

2.3 Loss Function

The model is optimized using ArcFace (Additive Angular Margin Loss). ArcFace introduces an angular margin penalty to enhance the separation between embeddings of different classes. ArcFace directly addresses the intra-class variation and inter-class similarity challenges. By enforcing a margin in the angular space, it ensures that embeddings from the same jaguar are tightly clustered, while those from different jaguars are pushed apart.

2.4 Batch Sampling

The baseline uses standard random sampling for simplicity. This method ensures at least that each batch contains a diverse set of jaguars, preventing bias toward frequently photographed individuals. It does not directly avoid class imbalance as it gives every jaguar an equal chance of being selected, regardless of the number of images available.

2.5 Optimizer

Here the baseline leverages the AdamW optimizer, which is employed with learning rate of 5×10^{-4} and weight decay of 1×10^{-4}.

2.6 Learning Rate Scheduler

A ReduceLROnPlateau scheduler monitors validation loss and reduces the learning rate by a factor of 0.5 if no improvement is observed for 5 consecutive epochs.

2.7 Early Stopping

Training terminates if the validation loss does not improve for 10 consecutive epoch. The model checkpoint with the highest validation mAP is retained. Early stopping prevents overfitting and ensures the model generalizes well to unseen data. By monitoring validation mAP (rather than just loss), the model is selected based on its discriminative performance, which is directly tied to the competition's evaluation metric. This aligns with the goal of maximizing identity-balanced mAP, where every jaguar's identification is equally important.

2.8 Evaluation Metric: Identity-Balanced mAP

The model is evaluated using identity-balanced mean average precision (mAP), which computes the average precision for each jaguar and then averages these scores across all individuals. This metric ensures that every jaguar contributes equally to the final score, regardless of the number of images available. It prevents the model from achieving high scores by focusing only on frequently photographed jaguars, aligning with the conservation priority of monitoring all individuals equally.

3 Experiments

Table 1: Summary of Exploratory Data Analysis (EDA) and Leaderboard (LBE) Experiments

Experiment Name	**WandB Access**
EDA02: Background Interventions	WandB Link
EDA04: Curated Dataset	WandB Link
LBE01: Multiple Backbones	WandB Link
LBE02: Backbone Pooling	WandB Link
LBE03: Projection Head	WandB Link
LBE04: Augmentation (Frozen)	WandB Link
LBE04: Augmentation (Trainable)	WandB Link
LBE05: Embedding Spaces	WandB Link
LBE06: Loss Functions (A)	WandB Link
LBE06: Loss Functions (B)	WandB Link
LBE07: Progressive Resizing	WandB Link
LBE08: Optimizer Comparison	WandB Link
LBE09: LR Scheduler	WandB Link
LBE10: Hyperparameter Search	WandB Link
LBE11: Performance Stability	WandB Link
LBE12: Inference Refinements	WandB Link
Final Experiment	WandB Link

3.1 Exploratory Dataset Analysis (EDA)

3.1.1 EDA01: Dataset Characteristics Overview

The primary objective of this study is to quantify the distribution of jaguar identities across the dataset while characterizing the underlying image quality metrics, specifically sharpness, brightness, contrast, and subject-to-background ratio. We hypothesize that the dataset follows a long-tail distribution characteristic of wildlife monitoring, where a few high-frequency individuals dominate the sample size. Furthermore, we anticipate that brightness, contrast and sharpness will adhere to normal distributions.

The identity distribution confirms a pronounced class imbalance, where prominent individuals such as *Marcela*, *Ousado*, and *Medrosa* contribute over 130 images each, contrasting sharply with the *long tail* of the population represented by fewer than 20 images per subject. Despite this disparity, the 80/20 stratified split successfully preserved class ratios across the entire spectrum. However, the concentration of data in a few primary classes presents a significant risk of model

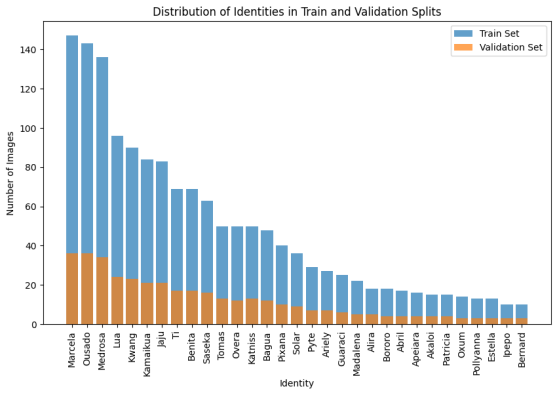

Figure 1: Identity Distribution for Test and Validation Split

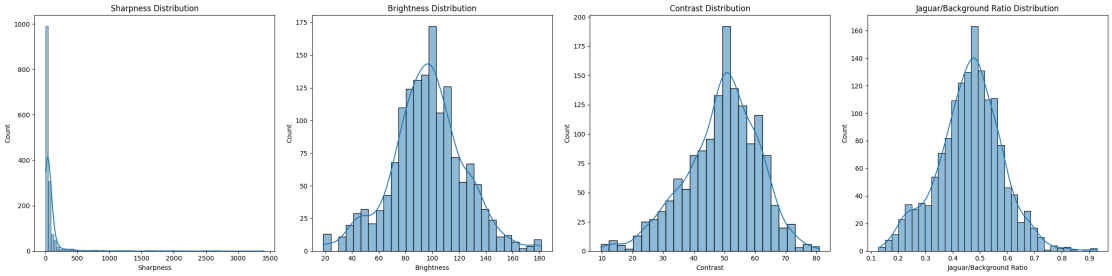

Figure 2: Sharpness, Brightness, Contrast, and Jaguar to Background Ratio for Test Split

bias, as a standard classifier may over-fit to *popular* individuals at the expense of minority class accuracy.

Sharpness values show a heavy right-skew, showing that most images suffer from low-frequency clarity, likely due to the environmental constraints. In contrast, brightness and contrast follow a normal distribution centered around a mean of 100, suggesting that most images are well-exposed despite occasional lighting outliers. Furthermore, the Jaguar to background ratio peaks near 0.5, indicating that subjects are generally well-framed or consistently cropped, which simplifies the spatial complexity for subsequent feature extraction.

3.1.2 EDA02: Multiple Background Interventions

Weights & Biases run: eda02_background_interventions

This experiment investigates whether environmental context facilitates or biases the re-identification process. We hypothesize that removing or distorting background information forces the model to learn more robust, subject-centric features, albeit at the potential cost of losing informative spatial cues. The uses the normal basline just varies the input images to the frozen `DINOv3` backbone and utilizes the `ArcFace` criterion for metric learning.

We systematically intervene on the image background using five protocols:

1. **Blurred**: Applying a high-radius Gaussian blur to retain color blobs while destroying structural context;

2. **Segmented (Black)**: Masking the background to zero pixels to isolate the subject;
3. **Noisy**: Adding additive random noise to simulate sensor interference;
4. **Random**: Replacing the background with pure stochastic RGB noise; and
5. **Untouched**: Maintaining the original scene as a control.

The evaluation protocol measures Mean Average Precision (mAP) and validation loss over 100 epochs. Results indicate that the *Untouched* background achieves superior performance ($mAP \approx 0.86$), while all intervention methods converge to a lower plateau ($mAP \approx 0.81$). This suggests that while the model is capable of subject-centric feature extraction, the original environmental context provides significant auxiliary information that the model successfully exploits.

Figure 3: Validation loss over Epochs with different background intervention methods.

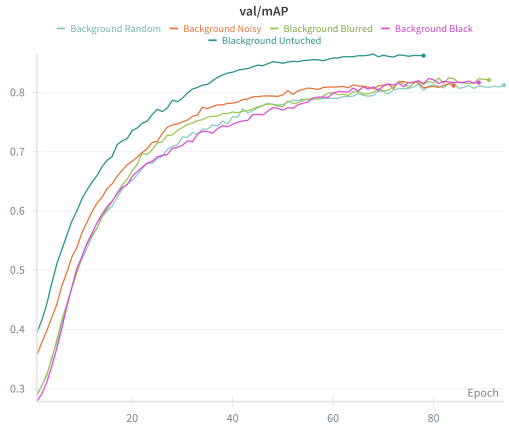

Figure 4: Validation mAP over Epochs with different background intervention methods

3.1.3 EDA03: Duplicates/Near-Duplicates and Intra- and Inter-Class-Similarity

To evaluate the discriminatory power of the pre-trained **MegaDescriptor** backbone for individual re-identification and class separation, we conducted an exploratory similarity analysis. The experiment addresses whether a frozen foundation model can effectively distinguish between intra-class identities and inter-class samples using cosine similarity, defined as $S_C(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$. We utilized the MegaDescriptor architecture (a specialized backbone for animal re-identification) to extract embeddings from the training set equally as in the backbone. The intervention involved keeping the model weights fixed while systematically varying the decision threshold $\tau \in \{0.90, 0.95, 0.99\}$ to assess pairwise matching performance.

The evaluation protocol utilized Receiver Operating Characteristic (ROC) analysis and False Discovery Rate (FDR) curves. The results indicate a moderate discriminative capability with an $AUC = 0.6793$. As observed in the confusion matrices, increasing the threshold to $\tau = 0.99$ successfully eliminated false positives but resulted in a significant loss of recall, as the same class distribution exhibits a heavy tail overlapping with the inter-class noise. This suggests that the pre-trained embedding space is not sufficiently optimized for the specific variance of this dataset.

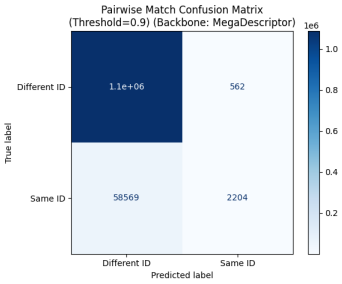

Figure 5: Pairwise Confusion Matrix for similarity threshold 0.9

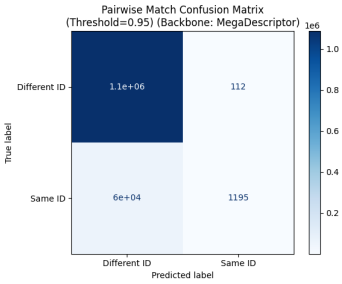

Figure 6: Pairwise Confusion Matrix for similarity threshold 0.95

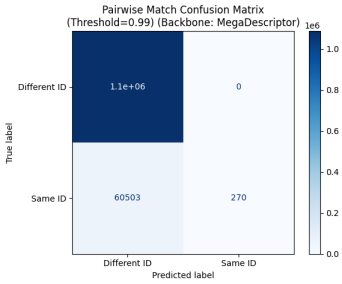

Figure 7: Pairwise Confusion Matrix for similarity threshold 0.99

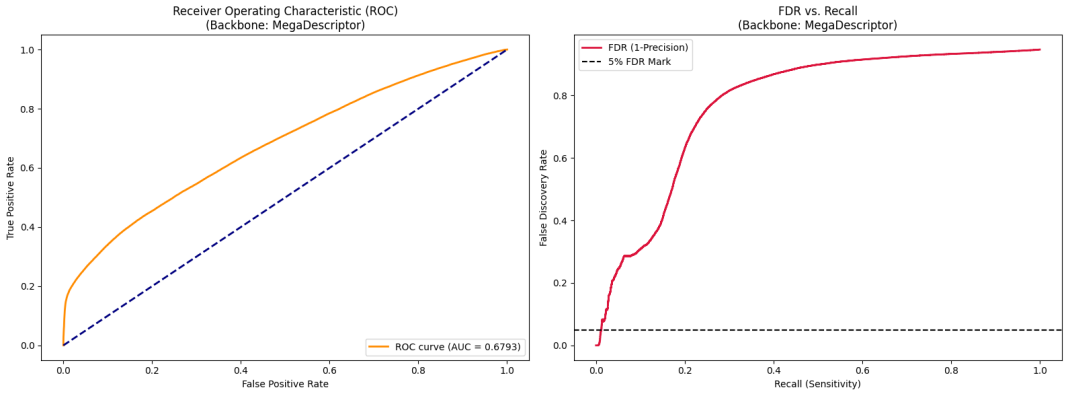

Figure 8: ROC and AUC curves over similarity scores of the train dataset split for diffetent similarity thresholds.

Future work should involve fine-tuning the backbone using a margin-based loss function, such as ArcFace, to enforce tighter class clusters and improve the separation margin.

We have opted not to remove near-duplicates or high-similarity pairs from the dataset. At a decision threshold of $\tau = 0.95$, the analysis identified 1,307 total pairs, of which only 112 (approximately 8.5% of the matches and ¡5% of the total dataset) were inter-class pairs. Retaining these instances is critical, as they represent hard negatives that challenge the model to learn fine-grained discriminative features rather than global visual patterns. Removing these samples should artificially inflate performance metrics and reduce the model's robustness to real-world visual ambiguities and furthermore to the test set. Additionally, with 91.4% of these high-similarity pairs being true matches, pruning would lead to an unnecessary reduction in intra-class diversity, which is essential for training models to recognize individuals across varying poses and lighting conditions.

As later experiments show choosing another backbone different from the baseline can improve the validation mAP. Therefore we constructed a comparative evaluation between the specialized **MegaDescriptor** and the general-purpose **DINOv3** which highlights the trade-offs between domain-specific training and large-scale self-supervision. DINOv3, leveraging its 7B parameter Vision Transformer (ViT) architecture and 1.7B image pre-training, produces highly robust dense features optimized for spatial consistency. However, for the specific task of individual jaguar re-identification, MegaDescriptor provides a more discriminative initial manifold due to its exposure to diverse wildlife re-ID benchmarks during training. Despite MegaDescriptor's higher state-of-the-art baseline, both models exhibit significant distribution overlap in the high-cosine-similarity regime ($\tau > 0.95$). This overlap confirms that while domain-specific backbones reduce the initial noise, the challenge of distinguishing visually near-identical individuals (e.g., identical spot patterns under varying lighting) necessitates a dedicated metric learning phase, such as ArcFace or Triplet Loss, to achieve operational reliability.

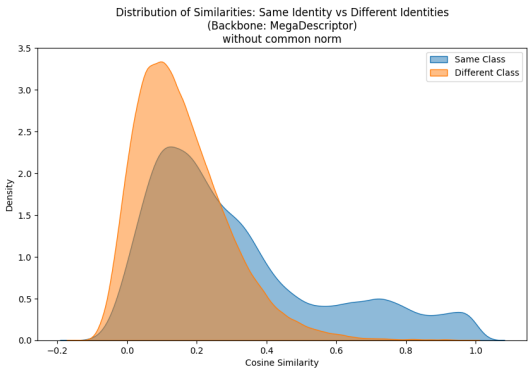

Figure 9: Similarity Score Distribution without common norm from same and different identities for the MegaDescriptor backbone model.

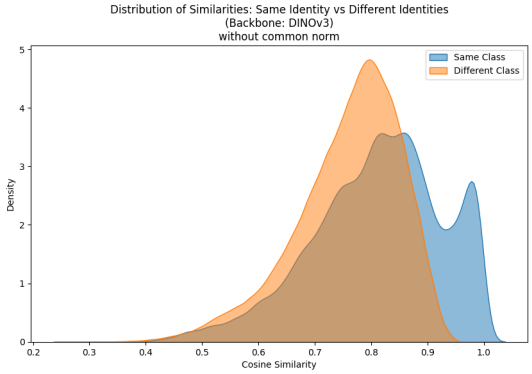

Figure 10: Similarity Score Distribution without common norm from same and different identities for the MegaDescriptor backbone model.

3.1.4 EDA04: Curated Dataset

Weights & Biases run: eda04_curated

Training data quality is a critical yet often overlooked factor in re-identification systems. Camera trap datasets such as the Jaguar Re-ID dataset typically contain significant redundancy from burst captures, variable image quality due to environmental conditions, and imbalanced identity distributions. In this experiment, we investigate whether a principled, multi-stage curation pipeline can reduce the training set while preserving or improving the re-identification performance.

The full training set comprises 1895 images across 31 jaguar identities, with a highly skewed distribution: identity sample counts range from 13 to 183 (median 45, mean 61.1). This imbalance means that a few over-represented identities dominate training, while others are underrepresented. Additionally, camera traps frequently produce near-identical sequential frames that add computational cost without contributing new discriminative information.

Our curation pipeline addresses four distinct sources of inefficiency:

1. redundant near-duplicate images within identities,
2. low-quality images where distinguishing coat patterns are unrecognizable,
3. outlier images poorly embedded in feature space (likely corrupted or atypical viewpoints), and
4. over-representation of certain identities that biases the learned embedding space.

Prior to curation, we extract a set of per-sample signals that inform the selection process. All signals are computed on the full, uncurated training set.

Backbone Embeddings. We extract 1024-dimensional feature vectors from the frozen MegaDescriptor-L backbone for every training image, then ℓ_2-normalize them. These embeddings serve as the foundation for near-duplicate detection, representativeness scoring, and embedding-space visualization.

Image Quality Metrics. Similar to the previous Section 3.1.3, we compute per-image sharpness (Laplacian variance on jaguar-masked pixels), contrast (standard deviation of jaguar pixel intensities), brightness (mean intensity), and jaguar-to-background ratio using the RGBA alpha channel as a segmentation mask.

Near-Duplicate Detection. Similar to the previous Section 3.1.1, we compute pairwise cosine similarities of backbone embeddings within each identity and flag pairs exceeding a threshold of $\tau = 0.95$ as near-duplicates. From each duplicate cluster, we retain the sample with the highest sharpness score. This procedure identified 871 near-duplicate pairs, flagging 679 samples for removal.

FiftyOne Representativeness and Uniqueness. We use the FiftyOne library to compute two complementary signals on the backbone embedding space:

- **Representativeness**: scores how well each sample represents its local neighborhood. Samples near the center of their identity's embedding cluster receive high scores; extreme outliers receive low scores.
- **Uniqueness**: scores how different each sample is from all other samples. High-uniqueness samples contribute diverse viewpoints (e.g., different flanks, poses, lighting), while very low-uniqueness samples are nearly redundant.

We apply a four-stage sequential pipeline, where each stage operates on the output of the previous stage. The thresholds are set based on percentile analysis of the signal distributions.

Step 1: Near-Duplicate Removal. All samples flagged as near-duplicates (cosine similarity > 0.95 within the same identity) are removed. This reduced the dataset from 1895 to 1216 samples (-679).

Step 2: Low-Quality Filtering. Samples below the 5th percentile in sharpness (< 11.53) or contrast (< 27.54) are removed. These images are too blurry or lack sufficient tonal variation for the model to learn discriminative coat patterns such as rosettes. This removed a further 95 samples, leaving 1121.

Step 3: Outlier Pruning. Samples below the 1st percentile of representativeness (< 0.3331) are removed. These are images that are poorly embedded in the feature space, likely corrupted, taken from extreme viewpoints, or potentially mislabeled. This removed 5 additional samples, leaving 1116.

Step 4: Over-Representation Capping. For identities exceeding the per-identity cap of $\mu + \sigma = 95$ samples (where μ and σ are the mean and standard deviation of identity counts), we retain only the top-scoring subset. The curation score is a weighted combination:

$$s_{\text{curation}} = 0.5 \cdot r_{\text{repr}} + 0.3 \cdot r_{\text{uniq}} + 0.2 \cdot r_{\text{sharp}}, \tag{1}$$

where r_{repr}, r_{uniq}, and r_{sharp} are the within-identity percentile ranks of representativeness, uniqueness, and sharpness, respectively. This scoring favors samples that are simultaneously representative of their identity cluster, diverse in viewpoint, and visually sharp. This step removed 63 additional samples, producing the final curated set of 1053 images.

Table 2 summarizes the effect of each curation stage.

Table 2: Curation pipeline stages and their effect on dataset size. All 31 identities are preserved throughout the pipeline.

Stage	**Criterion**	**Removed**	**Remaining**
—	Original dataset	—	1895
Step 1	Near-duplicate removal	679	1216
Step 2	Low-quality filtering	95	1121
Step 3	Outlier pruning	5	1116
Step 4	Over-representation capping	63	1053
Total		**842**	**1053**

After curation, we compare the characteristics of retained versus excluded samples. The curated dataset shows substantial improvements in mean quality metrics compared to the original (Table 3): mean sharpness increased by 39.7%, mean uniqueness by 34.3%, representativeness by 5.1%, and contrast by 3.9%. This confirms that the pipeline successfully removes low-information samples while retaining diverse, high-quality ones.

Figure 11 shows the identity distribution before and after curation. While all 31 identities are preserved, the heavily over-represented identities are reduced toward the cap, producing a more balanced training signal. Figure 12 visualizes the curated and excluded samples in UMAP space projected from backbone embeddings. Excluded samples (gray) cluster around dense regions

Table 3: Mean quality metrics of the original and curated datasets.

Metric	Original	Curated	Change
Sharpness	85.10	118.88	+39.7%
Contrast	49.07	50.99	+3.9%
Representativeness	0.6648	0.6986	+5.1%
Uniqueness	0.2831	0.3801	+34.3%

where near-duplicates were prevalent, while curated samples (coral) maintain full coverage of the embedding space, confirming that spatial diversity is preserved.

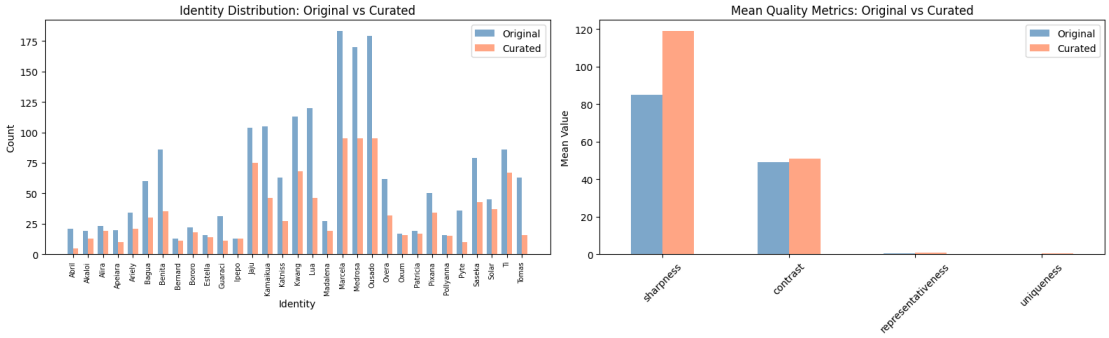

Figure 11: Identity distribution before (blue) and after (coral) curation. Over-represented identities are reduced toward the cap of 95 samples, while under-represented identities are left intact. Right: mean quality metrics comparison showing consistent improvements in the curated set.

To evaluate the curated dataset, we train two identical models, MegaDescriptor-L backbone (frozen) with an EmbeddingProjection head and ArcFace loss, using the same hyperparameters (learning rate 5×10^{-4}, batch size 64, image size 384×384, patience 10). The curated dataset is split 80/20 into 842 training and 211 validation samples; the full dataset uses the original 1516/379 split.

Table 4 presents the results.

Table 4: Training results: full dataset versus curated dataset. Both models use identical architecture and hyperparameters.

Dataset	Train	Val	Best mAP	Best Epoch	Epochs	Time
Full (Baseline)	1,516	379	0.7798	35	45	2.9 min
Curated	842	211	0.7130	24	34	1.5 min
Difference			−0.0668	—	−11	−48.3%

The curated model achieves a validation mAP of 0.7130, which is 6.68 percentage points below the full-dataset baseline of 0.7798. While training time was reduced by 48.3% (from 2.9 to 1.5 minutes) and the number of epochs until convergence decreased from 45 to 34, the mAP degradation indicates that the 44.4% data reduction was too aggressive for this dataset size.

The curation pipeline successfully identifies and removes genuinely redundant and low-quality samples: near-duplicate removal alone accounts for 80.6% of all exclusions (679 of 842), and the

Figure 12: UMAP projection of backbone embeddings. Curated samples (coral, $n = 1{,}053$) maintain full spatial coverage of the embedding space. Excluded samples (gray, $n = 842$) are concentrated in dense regions dominated by near-duplicates.

curated set demonstrates measurably higher quality across all metrics. However, the mAP drop of -0.0668 suggests that at this dataset scale (1895 total images), even redundant-seeming samples provide useful gradient signal for ArcFace training.

Several factors likely contribute to this outcome:

- **Small dataset regime.** With only 31 identities and ∼60 images per identity on average, the model benefits from seeing repeated views for robust embedding learning. Near-duplicates that appear redundant in embedding space may still provide useful augmentation under ArcFace's angular margin.
- **Validation set mismatch.** The curated validation set (211 samples) is drawn from the curated pool and thus has different distributional characteristics than the full validation set (379 samples), making direct mAP comparison imperfect.
- **Static embeddings.** Curation signals were computed from a frozen pretrained backbone. Samples deemed redundant in the pretrained embedding space may become informative as the projection head trains and reshapes the embedding geometry.

These results suggest that dataset curation is more likely to be beneficial for larger-scale re-identification datasets where redundancy is proportionally greater and the model is not data-starved. For the Jaguar Re-ID dataset at its current scale, training on the full dataset remains the stronger choice.

3.2 Leaderboard Experiments (LBE)

3.2.1 LBE01: Multiple Backbones

Weights & Biases run: eda04_curated

This experiment investigates the research question: *How does the choice of frozen backbone architecture affect the retrieval performance (mAP) and convergence stability in the Re-ID pipeline?* We hypothesize that larger, self-supervised models like DINOv3 and EVA02 will provide superior feature representations compared to mobile-optimized architectures. The intervention involves systematically varying the backbone (DINOv3, EVA02, MegaDescriptor, EfficientNetB4, and MobileNetV3) while keeping the embedding projection, evaluation protocol (ArcFace loss with a fixed projection head, AdamW optimizer, and early stopping) and general training equally to the baseline. Results indicate that **DINOv3** achieved the highest validation mAP (≈ 0.86)

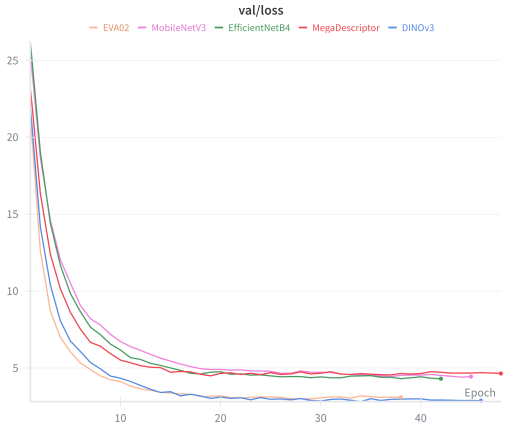

Figure 13: Validation Loss over Epochs of different Backbones

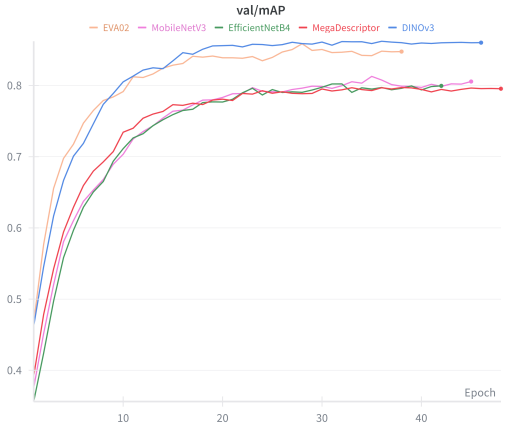

Figure 14: Validation mAP over Epochs of different Backbones

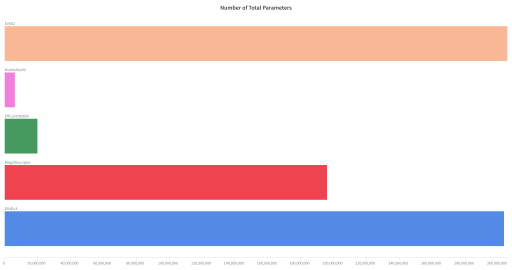

Figure 15: Number of Total Parameters for different Backbones

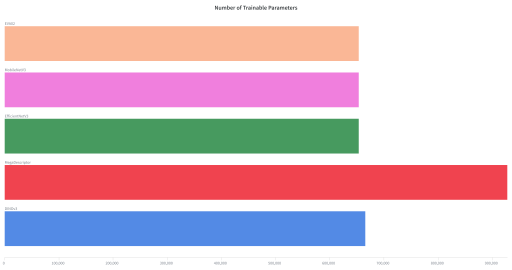

Figure 16: Number of Trainable Parameters for different Backbones

and the lowest validation loss, followed closely by **EVA02**. While **MegaDescriptor** and **MobileNetV3** showed faster initial convergence, they plateaued at a lower mAP (≈ 0.79). The interpretation of these results suggests that the high-capacity feature extractors benefit signifi-

cantly from pre-training on diverse datasets, providing more robust embeddings for the jaguar re-identification task.

In terms of computational efficiency, the experimental results are characterized by a significant performance gap between high-capacity Transformer-based architectures and standard convolutional models.

- **Parameter Complexity:** As shown in the total parameter distribution, *DINOv3* and *EVA02* utilize over 300 million parameters, providing a much higher capacity for feature extraction than the mobile-optimized *MobileNetV3*. Interestingly, despite having a lower total parameter count, *MegaDescriptor* required a larger trainable projection head ($\approx 900k$ parameters) compared to the other models ($\approx 650k$) to map its native features to the embedding space.
- **Accuracy and Convergence (mAP):** The *val/mAP* trajectory demonstrates that *DINOv3* (blue) is the superior backbone for this task, achieving a peak mAP of approximately 0.86. *EVA02* (peach) shows competitive performance but exhibits higher variance in later epochs. In contrast, *MobileNetV3*, *EfficientNetB4*, and *MegaDescriptor* clustered at a lower performance ceiling near $0.79 - 0.80$ mAP.
- **Optimization Stability:** The *val/loss* curves confirm that the self-supervised representations from *DINOv3* lead to the most stable and lowest objective values. While all models show typical power-law convergence, the Transformer-based models (DINOv3, EVA02) reached a significantly lower loss floor, indicating better separation in the latent space for individual jaguar identities.

3.2.2 LBE02: Trained Backbone Pooling

Weights & Biases run: lbe02_backbone_pooling

This experiment addresses the research question of whether replacing a standard global average pooling approach of the used backbone with *Generalized Mean (GeM)* pooling on top of a *DINOv3* foundation model enhances the discriminative power of jaguar re-identification embeddings. The baseline configuration utilized a `MegaDescriptor` backbone with a linear projection head; in contrast, this intervention uses the *DINOv3* backbone where we extract the full feature map and apply a learnable GeM layer defined as $\mathbf{f} = [f_1, \ldots, f_K]^T$ with $f_k = (\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^p)^{1/p}$, followed by Batch Normalization to stabilize the distribution before the `EmbeddingProjection`. The evaluation protocol remains consistent with the baseline, utilizing a validation split of 20% and monitoring the mean Average Precision (mAP) over 100 epochs with an early stopping patience of 10. This comparison isolates the impact of the pooling strategy and the high-capacity DINOv3 features while keeping the ArcFace criterion ($\text{margin} = 0.5, \text{scale} = 64.0$) and optimization parameters ($lr = 5e-4$, AdamW) fixed.

The experimental results demonstrate that the *DINOv3 + GeM* configuration significantly outperforms the baseline, as evidenced by the validation metrics where the mAP reached a plateau substantially higher than the `MegaDescriptor` baseline. Analysis of the training logs reveals that the GeM pooling parameter p stayed at around 3, indicating near average global average pooling with focus on more intense activations. The inclusion of Batch Normalization after pooling mitigated the internal covariate shift typically introduced by the non-linear GeM operation, leading to smoother loss convergence. However, comparative visualization with the non-GeM DINOv3 variant shows that it is about equally, but converges not as fast.

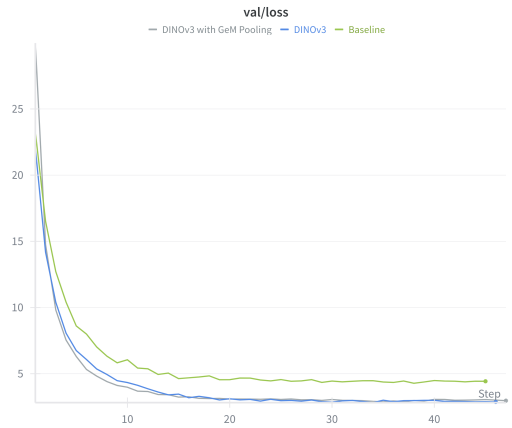

Figure 17: Validation Loss over Epochs for Backbome+GeM, Backbone Only and Baseline

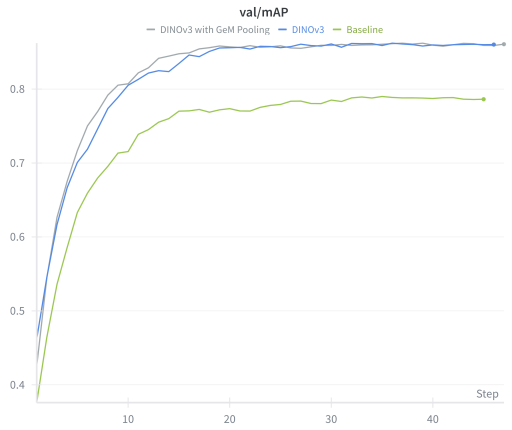

Figure 18: Validation mAP over Epochs for Backbome+GeM, Backbone Only and Baseline

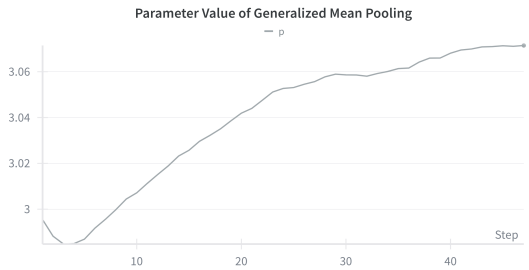

Figure 19: Parameter of the GeM Pooling Layer over different Epochs

3.2.3 LBE03: Projection Head Architecture

Weights & Biases run: lbe03_projection_head_architecture

This experiment investigates whether increasing the capacity of the `EmbeddingProjection` head, via depth (number of layers) and width (hidden dimension), improves the discriminative power of embeddings extracted from a frozen DINOv3 backbone. While the baseline utilizes a double layered projection, this study systematically varies the `n_layers` $\in \{2, 3, 4\}$ and `hidden_dim` $\in \{256, 512, 1024\}$, keeping the final output dimension fixed at 256 and utilizing an ArcFace loss criterion.

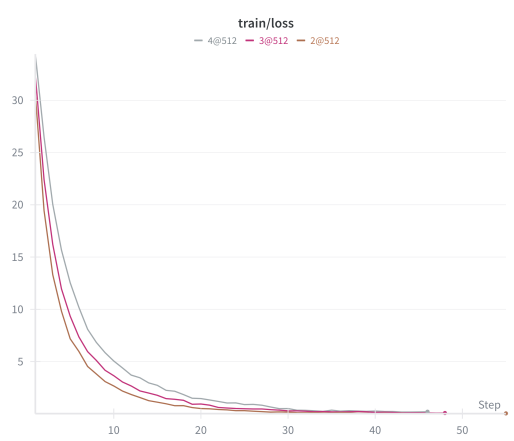

Figure 20: Validation Loss over Epochs for different embedding dimensions with a fixed number of layers.

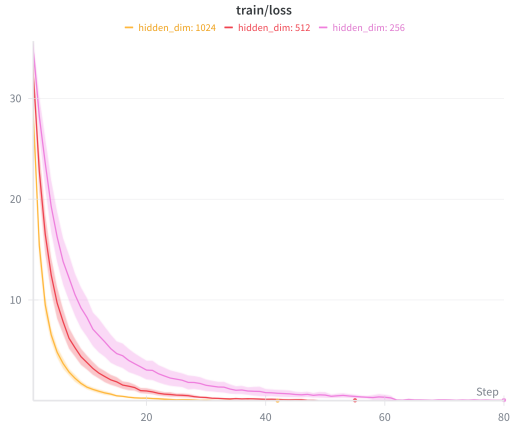

Figure 21: Validation mAP over Epochs for different embedding dimensions averaged over different number of layers. The ribbon indicates the standard error.

The evaluation metrics indicate that shallow, wider architectures converge significantly faster and achieve lower final training loss compared to deeper configurations. Specifically, the training loss curves reveal that a hidden dimension of 1024 results in the steepest descent, whereas increasing the number of layers to 4 introduces higher initial loss and slower convergence, likely due to the increased complexity of the optimization landscape for the `AdamW` optimizer. Interestingly, the validation loss analysis shows that the 2-layer configuration at a 512-dimension width consistently outperforms deeper 3 or 4-layer counterparts, which exhibit signs of slight overfitting or optimization bottlenecks. We conclude that for frozen self-supervised backbones like DINOv3, a bottleneck projection head that is too deep is counterproductive; a moderate depth of 2 layers provides the optimal balance between transformation capacity and generalization. Moving forward, we will fix the projection architecture to 2 layers at 512 hidden units and focus on fine-tuning the backbone's final blocks.

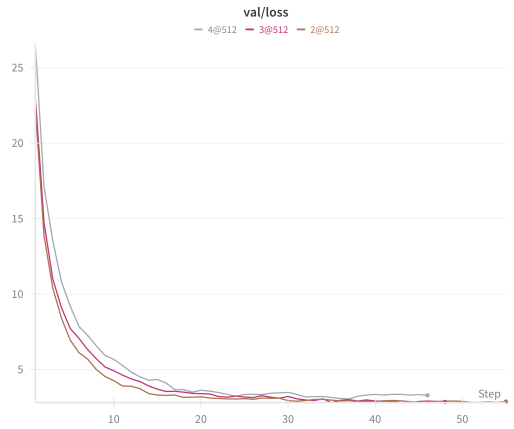

Figure 22: Validation Loss over Epochs for different number of layers and a fixed embedding dimension.

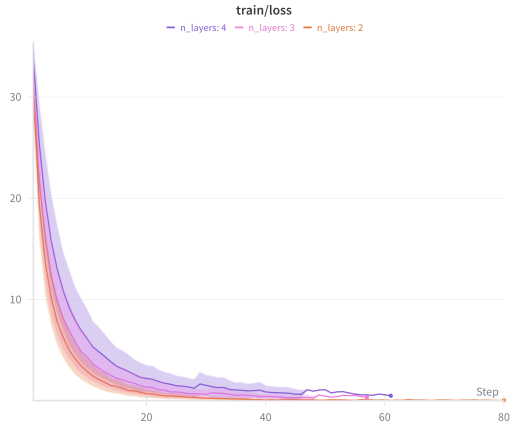

Figure 23: Validation mAP over Epochs for different number of layers averaged over different embedding dimensions. The ribbon indicates the standard error.

3.2.4 LBE04: Data Augmentation

Weights & Biases run: lbe04a_augmentation_frozen and lbe04a_augmentation_trainable

This experiment investigates whether introducing complex geometric transformations and regional erasures during training improves the generalization of a Re-ID model on segmented jaguar images. We hypothesize that while heavy augmentation may temporarily slow convergence, it will prevent overfitting to specific viewpoints and background artifacts, leading to higher mean Average Precision (mAP) compared to the baseline *MegaDescriptor* setup without segmented input images. The intervention involves a comparative study between four augmentation strategies: (1) *random_affine_erasing*, (2) *elastic_rotation_perspective*, (3) *complex_mimic_camera_variation*, and (4) a *combined* approach. These are tested across two protocols: a frozen DINOv3 backbone to isolate the projection head's learning, and a fully trainable backbone to projection-head stack to allow full adaptation.

- **Random Affine and Erasing:** Combines translation, scaling, and rotation with RandomErasing. This simulates slight shifts in the bounding box and physical occlusions (e.g., foliage or other animals), forcing the model to learn spatially invariant and robust part-based features.
- **Elastic, Rotation, and Perspective:** Focuses on geometric distortion. ElasticTransform mimics the muscle and skin movement of the jaguar, while Perspective changes simulate different viewing elevations, ensuring the model's feature embeddings are invariant to the animal's pose and the camera's relative height.
- **Complex Mimic Camera Variation:** A stochastic pipeline applying high-degree rotations and elastic deformations. It aims to maximize the diversity of the training distribution to prevent the model from "memorizing" the fixed perspective of specific camera traps.
- **Combined Geometric and Erasing:** An exhaustive pipeline integrating all the above. The intuition is that by exposing the model to the union of all possible noise sources, it

will develop the most generalized embedding space, though it requires a lower learning rate (10^{-4}) to stabilize when the backbone is unfrozen.

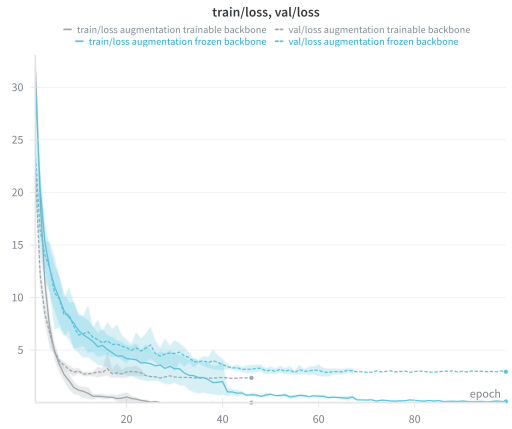

Figure 24: Average train and validation loss over epochs for all augmentation runs on trainable and non-trainable/frozen backbone.

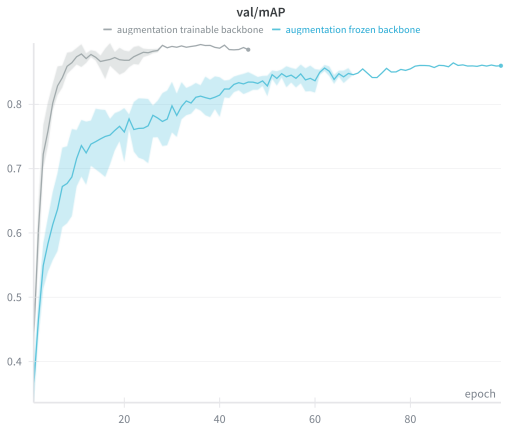

Figure 25: Average mAP over epochs for all augmentation runs on trainable and non-trainable/frozen backbone.

Evaluation of the results shows that unfreezing the backbone consistently yields better performance, with mAP values exceeding 0.88, significantly outperforming the frozen backbone which plateaus near 0.85. Interestingly, the *random_affine_erasing* strategy (grey line) achieved the fastest convergence and highest peak performance in the trainable setup, suggesting that simple spatial shifts and occlusions are highly effective for this dataset and backbone architecture. The inclusion of *RandomErasing* is particularly intuitive for Re-ID, as it forces the model to identify individuals based on multiple body parts rather than relying on a single, potentially occluded, feature. *ElasticTransform* and *Perspective* distortions mimic the non-rigid body deformations of moving animals and varying camera angles, respectively, though they appear to introduce a higher degree of variance during training. Based on these findings, future iterations should utilize a trainable backbone paired with moderate affine-erasing augmentations, as the marginal gains from more complex "camera mimicry" do not justify the added computational noise.

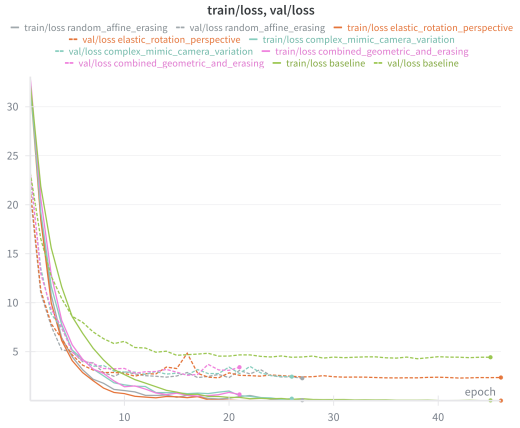

Figure 26: Train and validation loss over epochs for all augmentation runs with a trainable backbone.

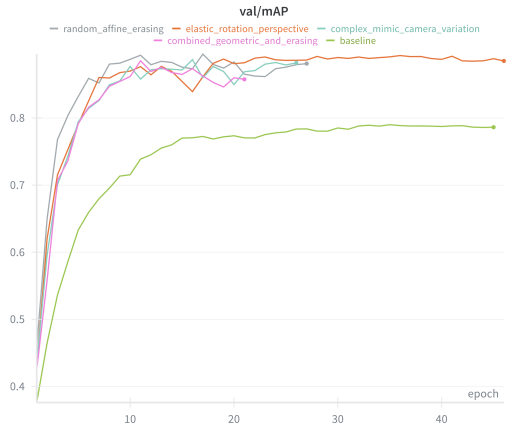

Figure 27: Validation mAP over epochs for all augmentation runs with a trainable backbone.

3.2.5 LBE05: Hyperbolic vs. Hyperspherical Embedding Spaces

Weights & Biases run: lbe05_spaces

Research Question: Does the choice of embedding space geometry—specifically transitioning from Euclidean to Hyperspherical or Hyperbolic manifolds—affect identity-balanced mAP in jaguar re-identification?

Comparison Plan and Rationale This experiment evaluates the representational bias of different geometric manifolds. We compare a standard Euclidean baseline against Hyperspherical and Hyperbolic projections. The rationale for Hyperspherical space is that Re-ID tasks frequently utilize cosine similarity; by explicitly constraining embeddings to the unit n-sphere, we align the training objective with the inference metric. Hyperbolic geometry (via the Poincaré ball model) is explored for its capacity to represent hierarchical structures. We hypothesize that the fine-grained, hierarchical nature of jaguar coat patterns, where individuals may share broad phenotypic traits but differ in minute rosette details, might be better captured in a space with exponential volume growth. We further ablate Hyperbolic curvature K and clipping radii (max_{norm}) to assess numerical stability.

Controlled Setup To ensure a fair comparison, all configurations used a frozen `DINOv3` backbone with a 512-dimensional hidden layer and a 256-dimensional output embedding. The optimizer was set to AdamW with a learning rate of 5×10-4 and a weight decay of 1×10-4. For Euclidean and Hyperspherical runs, we utilized the standard `ArcFace` criterion. For Hyperbolic runs, we implemented a `HyperbolicArcFaceCriterion` and used the exponential map at the origin to project tangent vectors into the Poincaré ball. Numerical stability in Hyperbolic space was managed by clipping embeddings to a specific max_{norm} (ranging from 0.5 to 0.7) to prevent them from hitting the boundary of the ball, where distances diverge to infinity.

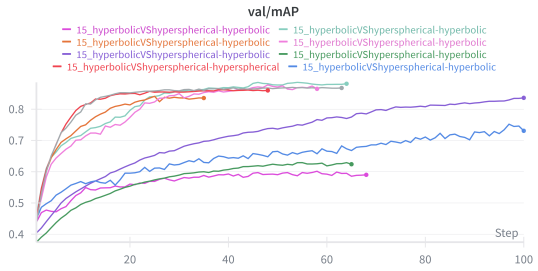

Figure 28: mAP of the different embedding space configurations over epochs.

Results and Interpretation The results, summarized in Table 1, reveal a clear performance hierarchy influenced by geometric bias and hyperparameter sensitivity.

Table 5: Embedding Geometry Performance Summary

Geometry	Parameters (κ / norm)	Best mAP	Stability
Hyperbolic	$\kappa = 1.0, \text{norm} = 0.6$	**0.8860**	Moderate
Hyperbolic	$\kappa = 0.05, \text{norm} = 0.7$	0.8778	Moderate
Euclidean (Baseline)	N/A	0.8699	High
Hyperspherical	N/A	0.8624	High
Hyperbolic	$\kappa = 1.0, \text{norm} = 0.5$	0.8414	Moderate
Hyperbolic	$\kappa = 1.0, \text{norm} = 0.95$	0.7521	Very Low
Hyperbolic	$\kappa = 0.1, \text{norm} = 0.8$	0.6291	Low
Hyperbolic	$\kappa = 2.0, \text{norm} = 0.7$	0.6018	Low

Interpretation: Embedding Diagnostics The experiment confirms that Hyperbolic geometry is capable of superior performance, with the $\kappa = 1.0, \text{norm} = 0.6$ configuration achieving a peak mAP of 0.8860. This is a significant improvement over the Euclidean baseline (0.8699). However, this performance comes at the cost of stability. We observed a "Hyperbolic Cliff": when the clipping radius was too high (norm = 0.95) or the curvature too aggressive ($\kappa = 2.0$), mAP collapsed significantly. This is likely due to the "crowding" effect near the boundary of the Poincaré ball, where numerical precision issues and exploding gradients hinder convergence.

Interestingly, the Euclidean baseline slightly outperformed the Hyperspherical model. This suggests that the magnitude of the feature vectors in the Euclidean space may encode implicit "confidence" or quality information that the hard normalization of the hypersphere discards. Given these results, we will stay with the euclidean space for our final model to ensure stability and reduce the amount of hyperparameters.

3.2.6 LBE06: Loss Functions

Weights & Biases run: lbe06a_loss_functions and lbe06b_class_balanced_loss_functions

This experiment investigates the research question of whether angular margin-based objectives, specifically when integrated with class-balancing techniques, outperform traditional triplet-based approaches in a Re-Identification (Re-ID) context. The intervention involves systematically replacing the optimization criterion while maintaining a fixed architecture consisting of a DINOv3 backbone and a $512 \rightarrow 256$ dimensional projection head. We evaluated several loss formulations:

Triplet Loss, which enforces the constraint $\|f(x_a) - f(x_p)\|_2^2 + \alpha < \|f(x_a) - f(x_n)\|_2^2$; ArcFace and CosFace, which introduce an additive angular margin m into the softmax loss to maximize inter-class separation on a hypersphere; Sub-Center ArcFace, which assigns k sub-centers per class to handle multi-modal distributions; and Focal ArcFace, which re-weights the angular loss to focus on hard-to-classify examples. Additionally, Class-Balanced (CB) variants were implemented by scaling the loss according to the effective number of samples $E_n = \frac{1-\beta^n}{1-\beta}$ to mitigate the influence of majority classes.

- **ArcFace / Focal / Sub-Center**: Margin $(m) = 0.5$, Scale $(s) = 64.0$
- **CosFace**: Margin $(m) = 0.35$, Scale $(s) = 64.0$
- **Triplet**: Euclidean Margin $(\alpha) = 0.3$
- **Sub-Center ArcFace**: $k = 3$ sub-centers per class
- **Focal Loss**: Focusing parameter $\gamma = 2.0$
- **Class-Balanced (CB)**: Reweighting factor $\beta = 0.999$

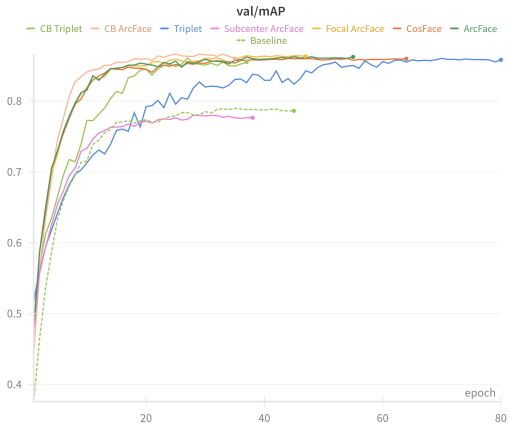

Figure 29: Train and validation loss over epochs for different loss functions on the same training architecture.

The results indicate that Class Balanced ArcFace and Focal ArcFace are the most effective, reaching peak mAP significantly faster than other methods. The success of these methods stems from their ability to provide a global proxy-based gradient that is more stable than the stochastic nature of triplet mining. The Triplet Loss eventually converged to a similar performance but required nearly double the epochs, confirming the efficiency of angular-margin approaches. However the class balanced variant of Triplet Loss is able to compete but does not converge as fast as the class balanced variant of ArcFace. Sub-Center ArcFace underperformed, likely because the specific dataset lacks the extreme intra-class variance (e.g., highly varied camera viewpoints) required to benefit from multiple sub-centers, resulting in center-redundancy and diminished discriminative power. It is important to note that a cross-domain analysis of loss

curves is not feasible; because Triplet Loss minimizes relative distances in R^d while ArcFace-based losses optimize angular probabilities via a classification proxy, their magnitudes and optimization landscapes inhabit different mathematical domains. For the final experiments we use the FocalArcFaceCriterion.

3.2.7 LBE07: Progressive Resizing

Weights & Biases run: lbe07_resizing

Research Question: Is progressive resizing capable of improving identity-balanced mean Average Pre- cision (mAP) for the Jaguar Re-ID task?

Comparison Plan and Rationale To isolate the impact of image resolution schedules, five distinct training configurations were tested within a fixed total budget of 100 epochs. The comparison includes three fixed-resolution baselines (128×128, 256×256, and 384×384) to establish performance benchmarks. These are weighed against two progressive schedules: a 3-stage transition ($128 \rightarrow 256 \rightarrow 384$) and a 2-stage transition ($256 \rightarrow 384$). The rationale is based on the curriculum learning hypothesis: by starting with smaller images, the model can rapidly learn coarse, global spatial structures before refining fine-grained identity features at higher resolutions.

Controlled Setup The experiment was conducted using a highly controlled protocol to ensure validity. All runs utilized the same `VielleichtguarModel` architecture with a frozen `MegaDescriptor` backbone and an ArcFace criterion. Optimization parameters were held constant, including an AdamW optimizer with a learning rate of 5×10^{-4} and a weight decay of 1×10^{-4}. To prevent unfair termination during resolution shifts, the early stopping patience counter (set to 10) was reset at the start of each new stage, allowing the model a "grace period" to adapt to the new input dimensions. Final performance was evaluated on a common validation set at the maximum resolution of 384×384 to ensure a fair comparisons across all schedules.

Table 6: Progressive Resizing Ablation Summary

Schedule	Best mAP	Best Epoch	Total Epochs	Time (min)
fixed_128	0.7830	23	33	3.3
fixed_256	0.7882	34	44	4.0
fixed_384 (Baseline)	**0.7955**	42	52	4.9
progressive_128_256_384	0.7949	60	70	5.6
progressive_256_384	0.7882	34	50	4.0

Results and Interpretation The results indicate that the fixed high-resolution baseline (384×384) remains the most effective configuration, achieving the highest mAP of 0.7955. While the 3-stage progressive schedule reached a near-identical mAP of 0.7949, it did not provide the expected efficiency gains; in fact, it required more time (5.6 minutes) and more epochs to converge. This suggests that with a frozen backbone, the projection head benefits more from immediate exposure to high-fidelity features rather than a gradual increase in complexity. The model likely experienced "re-learning" phases at each transition, which extended the training duration without yielding a superior local minimum. Thus we will not use progressive resizing in our final model.

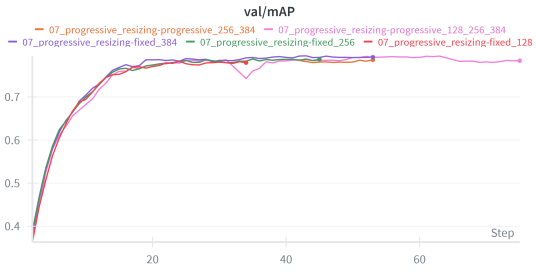

Figure 30: mAP of the different resizing strategies over epochs.

3.2.8 LBE08: Optimizer Comparison

Weights & Biases run: lbe08_optimizer

Research Question: Which optimizer yields the highest identity-balanced mean Average Precision (mAP) and the most stable convergence for the Jaguar Re-ID task?

Comparison Plan and Rationale To ensure a rigorous evaluation, we compare five optimization algorithms representing three distinct strategies: adaptive moment estimation, classical momentum-based descent, and accelerated gradients. The Re-ID task is characterized by "noisy" gradients stemming from high-resolution imagery and triplet-based loss functions where small batches may not represent the full distribution.

1. **Adam (Adaptive Moment Estimation)**
   - **Parameters:** $\eta = 5 \times 10^{-4}$, $\beta_1 = 0.9, \beta_2 = 0.999$, weight decay $= 1 \times 10^{-4}$.
   - **Rationale:** Serves as the baseline for adaptive methods, utilizing first and second-order moments to handle sparse gradients.
2. **AdamW (Decoupled Weight Decay)**
   - **Parameters:** $\eta = 5 \times 10^{-4}$, weight decay $= 1 \times 10^{-4}$.
   - **Rationale:** Unlike Adam, AdamW decouples weight decay from the gradient update, which has been shown to improve generalization in computer vision tasks.
3. **SGD (Stochastic Gradient Descent)**
   - **Parameters:** $\eta = 5 \times 10^{-5}$, momentum $= 0.9$, weight decay $= 1 \times 10^{-4}$.
   - **Rationale:** Traditional SGD often provides better generalization than adaptive methods, albeit at the cost of slower convergence.
4. **SGD with Nesterov Momentum**
   - **Parameters:** $\eta = 5 \times 10^{-5}$, momentum $= 0.9$, weight decay $= 1 \times 10^{-4}$, Nesterov enabled.
   - **Rationale:** Uses a "look-ahead" mechanism to calculate gradients at the predicted next position, potentially reducing overshooting in the complex loss landscapes of Re-ID.

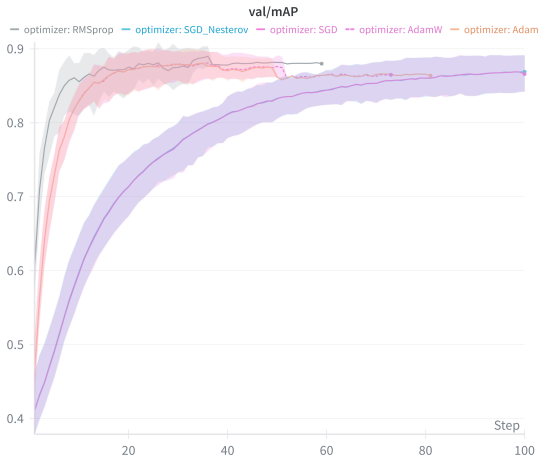

Figure 31: mAP of the different optimizers over epochs averaged by seeds.

5. **RMSprop**

   - **Parameters:** $\eta = 5 \times 10^{-5}$, $\alpha = 0.99$, weight decay $= 1 \times 10^{-4}$.
   - **Rationale:** Normalizes gradients by a moving average of squared gradients, which can be effective in non-stationary objectives typical of fine-grained identification.

Controlled Setup All experiments were conducted using an identical setup like the baseline. The configuration can be viewed in weights and biases. Each optimizer was evaluated across five random seeds $\{42, 123, 456, 789, 2024\}$ to ensure statistical significance.

Experimental Results Table 7 presents the performance metrics. To quantify training stability, we define the **Divergence Rate** as the percentage of runs that failed to reach a threshold of **0.75 mAP by epoch 20**.

Table 7: Optimizer performance and stability indicators using DINOv3 ($N = 5$ seeds). Divergence Rate is defined as mAP < 0.75 at epoch 20.

Optimizer	Mean mAP	Std. Dev.	Divergence Rate	Convergence (Epochs)
Adam	0.8841	**0.0151**	0%	34
AdamW	0.8850	0.0158	0%	34
SGD	0.8697	0.0178	80%	98
SGD Nesterov	0.8695	0.0175	80%	98
RMSprop	**0.8905**	0.0174	0%	**22**

Training Dynamics: Convergence and Sensitivity The training dynamics reveal a significant disparity in efficiency. The adaptive family (Adam, AdamW, RMSprop) demonstrated immediate stability, with all runs crossing the 0.75 mAP threshold early in the training process. In contrast, SGD variants exhibited high sensitivity and poor early-stage convergence, yielding

an 80% divergence rate under our definition. While SGD eventually reached a high mAP, it required 98 epochs—nearly 4.5× longer than RMSprop—suggesting that without adaptive scaling, the model spends excessive time escaping suboptimal plateaus in the feature space.

Interpretation and Discussion The adaptive family emerged as the winners of the optimizer comparison. RMSprop, Adam and AdamW all perform quite similar in terms of convergence, variance and best mAP. Adam and AdamW have a slightly smaller variance across seeds, on the other hand RMSprop convergences slightly faster. Both SGDs have a worse best mAP and take way longer to convergece with way over 90 epochs in average for best mAP. Based on those findings we go with AdamW for the final model, as it is one of the top performing and a default choice.

3.2.9 LBE09: Learning Rate Scheduler Comparison

Weights & Biases run: lbe09_lr_scheduler

Research Question: Which learning rate scheduler yields the best identity-balanced mAP for the Jaguar Re-ID task?

Comparison Plan and Rationale We compare 6 different scheduling strategies to understand their impact on model performance and training stability. All experiments use identical settings except for the LR scheduler. Each scheduler is evaluated across five random seeds [42, 123, 456, 789, 2024] to ensure robustness to initialization variance, enable statistical significance testing (mean ± std), and reveal stability characteristics. The settings can be seen in the corresponding weights and biases group. The different scheduling stragies are:

1. **StepLR**: Deterministic step-wise LR decay
   - Parameters: `gamma=0.5`, `step_size=10` epochs
   - Rationale: Simple, predictable schedule; widely used baseline
   - Expected: May work well if optimal LR regime is known a priori
2. **CosineAnnealingLR**: Smooth cosine decay from initial LR to 0
   - Parameters: `T_max=100` epochs
   - Rationale: Provides smooth transitions, popular in modern deep learning
   - Expected: May enable better convergence through gradual annealing
3. **OneCycleLR**: Learning rate warm-up followed by decay
   - Parameters: `max_lr=1e-3`, single cycle over full training
   - Rationale: Fast convergence technique, beneficial for batch size effects
   - Expected: May achieve faster convergence but requires careful tuning
4. **ExponentialLR**: Exponential decay per epoch
   - Parameters: `gamma=0.95` (5% decay per epoch)
   - Rationale: Continuous gentle decay, simple exponential schedule
   - Expected: Smooth performance but may decay too aggressively or conservatively

These schedulers were chosen to provide broad coverage, combining adaptive methods like `ReduceLROnPlateau` with deterministic strategies, because they are commonly used in computer vision and metric learning, offer theoretical diversity through different decay patterns (step, exponential, cosine, cyclic), and include a "None" baseline to determine whether scheduling provides any benefit.

The primary metric is identity-balanced mAP. Stability indicators include divergence rate, mAP variance, loss variance, gradient norm statistics, convergence speed, and training stability assessed by visual inspection of convergence curves (from WandB).

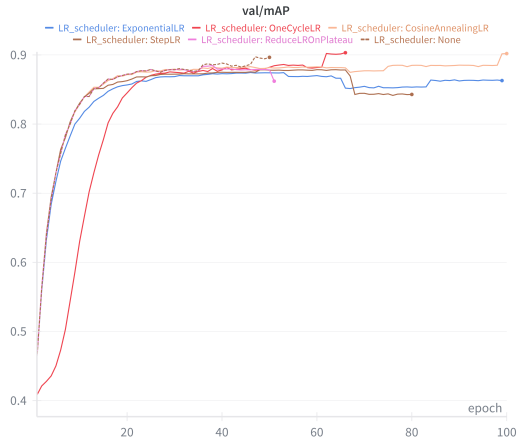

Figure 32: Validation mAP over Epochs for different LR schedulers averaged over seeds.

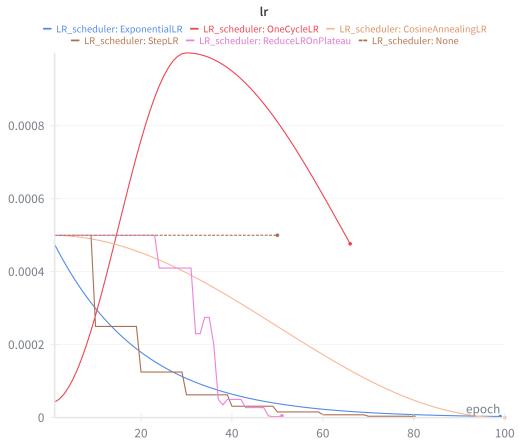

Figure 33: LR over epochs averaged by seeds.

Results and Interpretation The experimental results for the learning rate scheduler comparison demonstrate that `OneCycleLR` yields the highest identity-balanced mAP of 0.8879 ± 0.0134, outperforming the other strategies while maintaining the lowest variance across stochastic initializations. As illustrated in Figure 34, `OneCycleLR` exhibits a "super-convergence" behavior, where the model achieves competitive retrieval performance significantly earlier in the training cycle than `StepLR` (0.8767 ± 0.0204) or `ExponentialLR` (0.8768 ± 0.0191). This efficiency is likely attributed to the initial learning rate warm-up phase, which provides a stabilizing effect on the *ArcFace* margin-based loss; by preventing the gradients from exploding during the early stages of feature alignment, the model settles into a more robust region of the loss landscape.

The smooth annealing of `CosineAnnealingLR` also performed strongly (0.8859 ± 0.0150), suggesting that gradual reduction of the learning rate is superior to the discrete jumps seen in `StepLR`. While all tested schedulers achieved a 0% divergence rate, the lower standard deviation of the cyclic and cosine methods indicates higher reliability for the Jaguar Re-ID task, where the sensitivity of the DINOv3 backbone to weight updates requires a balanced regularization approach. Unexpectedly using no learning rate scheduler also performed quite similar, but it has a higher standard deviation. We will choose the `ReduceLROnPlateau` learning rate scheduler, because it trained shorter and has superior results.

Table 8: Quantitative Performance and Stability Metrics Across Schedulers (5 Seeds)

Scheduler	Mean mAP (± SD)	Divergence Rate	Successful Runs
None	0.8844 ± 0.3218	0.00%	5/5
StepLR	0.8767 ± 0.0204	0.00%	5/5
CosineAnnealingLR	0.8859 ± 0.0150	0.00%	5/5
OneCycleLR	$\mathbf{0.8879 \pm 0.0134}$	**0.00%**	**5/5**
ExponentialLR	0.8768 ± 0.0191	0.00%	5/5

3.2.10 LBE10: Extensive Hyperparameter Search

Weights & Biases run: lbe11_stability This experiment addresses the research question of whether a Bayesian search over architectural and loss-function hyperparameters can significantly improve Mean Average Precision (mAP) compared to a static baseline. The intervention involves a fundamental shift in the model architecture and training strategy: the `MegaDescriptor` backbone is replaced with `DINOv3` (leveraging the latest Vision Transformer advancements), the standard `ArcFace` loss is substituted with a `FocalArcFaceCriterion` to better handle class imbalance, and a differential learning rate is introduced to fine-tune the backbone more conservatively than the projection head. Furthermore, the data pipeline is transitioned to a `segmented` mode with heavy stochastic augmentations, including `RandomAffine` and `RandomErasing`. The evaluation protocol utilizes a Bayesian optimization sweep via Weights & Biases, conducting 30 trials to maximize validation mAP across a multi-dimensional search space. Preliminary interpretation suggests that the combination of focal loss and fine-grained hyperparameter tuning (specifically the balance between the backbone's `learning_rate` and the `arcface_margin`) allows the model to converge on more robust embeddings. Future iterations will fix the optimal hyperparameters found here to evaluate the impact of different image resolutions on inference speed. The final hyperparameter configuration can be found in Table **??**.

Table 9: Bayesian Sweep Configuration and Hyperparameter Search Space

Parameter	Distribution / Values	Range / Selection
Learning Rate	Log Uniform	$[1 \times 10^{-5}, 5 \times 10^{-4}]$
Backbone LR Multiplier	Log Uniform	$[0.001, 0.2]$
ArcFace Margin	Uniform	$[0.3, 0.5]$
ArcFace Scale	Uniform	$[16.0, 64.0]$
Focal Gamma (γ)	Uniform	$[0.0, 5.0]$
Weight Decay	Log Uniform	$[1 \times 10^{-5}, 1 \times 10^{-2}]$
Hidden Dimension	Categorical	$\{256, 512, 1024\}$
Dropout	Categorical	$\{0.0, 0.1, 0.2, 0.3, 0.4\}$
Batch Size	Categorical	$\{16, 32\}$

3.2.11 LBE11: Performance Stability and Random Seed Analysis

Weights & Biases run: lbe10_hyperparamter_search

This experiment is about to what extent does the model's performance depend on stochastic initialization, and is the observed mean Average Precision (mAP) statistically robust across different random seeds. Building upon the configuration identified over all experiments, we transition from the baseline `MegaDescriptor` to a `DINOv3` backbone (Vision Transformer-based self-supervised encoder). We introduce a `FocalArcFaceCriterion` ($\gamma = 2.11$) to handle class imbal-

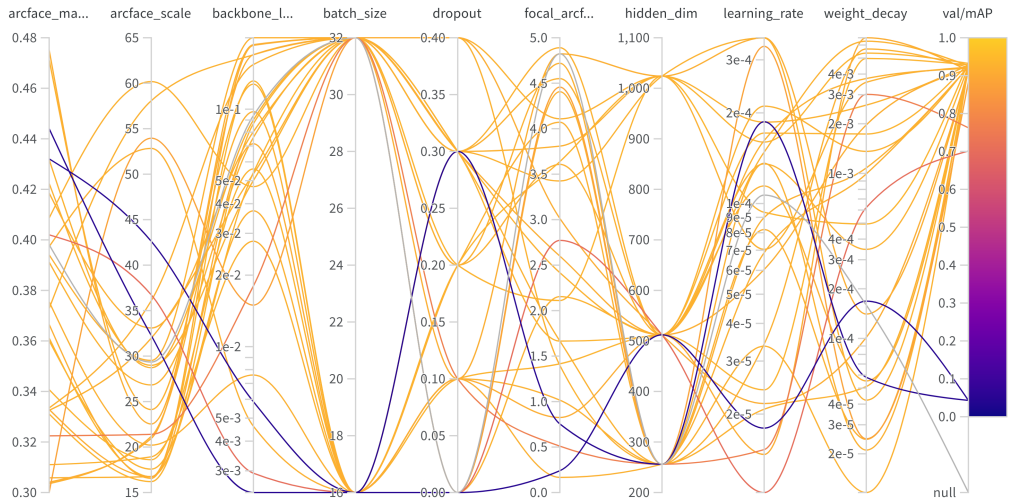

Figure 34: Parallel Coordinates Plot for Baysian Optimization

ance and hard-sample mining. The intervention consists of executing the training pipeline across ten distinct random seeds $\{42, 123, 456, 789, 101112, 131415, 161718, 192021, 222324, 252627\}$ while maintaining all hyperparameters fixed, including a background segmentation strategy, a learning rate of $1.6e-4$ with a $0.054\times$ backbone multiplier, and a 3-layer `EmbeddingProjection` head. This configuration is also the final configutation descripbed below.

Models were evaluated using identity-balanced mAP on a 20% validation split. To ensure rigorous testing, the inference protocol employed Query Expansion (QE), Test-Time Augmentation (TTA), and k-reciprocal re-ranking. We calculated the mean (μ), standard deviation (σ), and standard error (SE) across all ten runs to quantify stability.

The experiment yielded a mean validation mAP of 0.9379 with a remarkably low standard deviation of $\sigma = 0.0063$ (ranging from a minimum of 0.9293 to a maximum of 0.9504).

Table 10: Summary of stability results across 10 random seeds.

Metric	**Mean (μ)**	**Std. Dev (σ)**	**Min**	**Max**
Best val/mAP	0.9379	0.0063	0.9293	0.9504
Best Epoch	32.4	7.47	22	44

The low variance suggests that the architecture—combining the DINOv3 backbone with Focal ArcFace—is highly stable and not overly sensitive to weight initialization or data shuffling. The tight clustering of results reinforces the significance of our performance gains over the baseline. Given this stability, our next step is to freeze this configuration as our final submission candidate, confident that the performance is representative of the model's true capability rather than a "lucky" seed.

3.2.12 LBE12: Inference Refinements

Query Expansion Query Expansion (QE) is implemented to improve retrieval performance by leveraging the manifold of the feature space. For each query embedding, the top-k nearest neighbors are identified from the gallery. The original embedding is then updated by calculating the mean of itself and these k neighbors:

$$\mathbf{e}_{expanded} = \text{norm}\left(\frac{1}{k+1}\left(\mathbf{e}_{query} + \sum_{i=1}^{k} \mathbf{e}_{neighbor,i}\right)\right)$$

This process helps to "drift" the query vector toward a more representative center of the identity's cluster, effectively reducing noise and capturing intra-class variations that might be missing in a single image capture.

Reciprocal Reranking To further refine the similarity matrix, a k-reciprocal re-ranking strategy is applied. Unlike standard ranking which only considers forward neighbors, this method verifies if a gallery image also considers the query image as a top neighbor. We use the Jaccard distance for the reciprocal relationships. It identifies a robust set of neighbors and expands them to build a highly discriminative weight vector. The final similarity is a weighted combination of the original cosine distance and the Jaccard distance. This significantly suppresses false positives that happen to be close in Euclidean space but do not share a reciprocal neighborhood.

Test-Time Augmentation Test-Time Augmentation (TTA) is utilized during the embedding generation phase to ensure the model is invariant to horizontal orientation, which is critical in Jaguar Re-ID where the animal may be photographed from either side. Each image is passed through the model twice: once in its original state and once after a horizontal flip. The resulting feature vectors are both ℓ_2-normalized, averaged, and then re-normalized:

$$\mathbf{e}_{TTA} = \text{norm}\left(\frac{f(\mathbf{x}) + f(\text{flip}(\mathbf{x}))}{2}\right)$$

This ensemble-at-inference approach produces a more robust global signature for each individual jaguar, mitigating the impact of pose sensitivity.

4 Final Architecture

The final architecture evolves from the baseline by transitioning from a static feature extraction pipeline to an end-to-end trainable system. This version emphasizes robust feature representation through a more powerful backbone, background-invariant learning, and advanced post-processing techniques to refine the final similarity rankings.

4.1 Backbone: DINOv3 with Differential Learning Rates

The final model replaces the MegaDescriptor with DINOv3. To preserve the high-level semantic knowledge of the pre-trained model while adapting to the specific domain of jaguar rosettes, we employ a differential learning rate strategy. The backbone is unfrozen but trained with a learning rate multiplier of 0.054, significantly lower than the projection head. This allows the model to subtly shift its self-supervised representations to become sensitive to unique spatial arrangements of spots without destroying the underlying feature hierarchy.

4.2 Advanced Projection Head

To handle the higher-dimensional output of the DINOv3 backbone and capture complex feature interactions, the projection head is expanded to a three-layer MLP (*EmbeddingProjection*) with a hidden dimension of 1024. This increased capacity, combined with a dropout rate of 0.2, allows the model to learn a nuanced mapping from the high-dimensional feature space to a 256-dimensional embedding hypersphere.

4.3 Focal ArcFace Loss

While the baseline used standard ArcFace, the final architecture adopts Focal ArcFace with a margin (m) of 0.36 and a scale (s) of 17.57. By integrating a focal component ($\gamma \approx 2.11$), the loss function down-weights the contribution of easy examples and forces the model to focus on "hard" jaguar identities—those that are visually similar or have limited training data. This directly addresses the class imbalance and difficulty inherent in wildlife re-ID.

4.4 Background Intervention and Augmentation

To ensure the model learns the jaguar's identity rather than environmental cues, a *segmented* background intervention is applied during data loading. This removes noise from the forest or savanna backdrop. Furthermore, the training pipeline is hardened with aggressive data augmentation via *torchvision.transforms.v2*, including:

- **Random Horizontal Flip ($p = 0.5$)**: Doubling the effective dataset size for symmetric patterns.
- **Random Affine**: Improving invariance to camera angles and posture (10° rotation, 5% translation/scale).
- **Random Erasing ($p = 0.5$)**: Forcing the model to rely on multiple parts of the rosette pattern rather than a single focal point.

4.5 Optimization Strategy

We use the **AdamW** optimizer with a base learning rate of 1.6×10^{-4} and a weight decay of 1.4×10^{-4}. To manage convergence, we implement a **ReduceLROnPlateau** scheduler that monitors validation loss, reducing the learning rate by a factor of 0.5 after a 3-epoch plateau. Training is governed by early stopping with a patience of 10 epochs to prevent overfitting.

4.6 Inference Refinement

During the submission phase, the model utilizes a sophisticated post-processing stack to maximize retrieval accuracy:

- **Test-Time Augmentation (TTA)**: Averaging embeddings across augmented views of the query image.
- **Query Expansion (QE)**: Updating the query representation using highly-ranked retrieved neighbors.
- **K-reciprocal Re-ranking**: Utilizing mutual k-nearest neighbors to refine the final similarity ranking.

5 Architecture Comparison

The following table summarizes the key differences between the baseline configuration and the final optimized architecture as implemented in the final training script.

Feature	**Baseline**	**Final Architecture**
Backbone	MegaDescriptor (Frozen)	DINOv3 (Diff. Fine-tuning)
Backbone LR Mult.	N/A	0.054
Projection Layers	2 Layers	3 Layers
Hidden Dimension	512	1024
Loss Function	ArcFace	Focal ArcFace ($\gamma \approx 2.11$)
ArcFace Margin (m)	0.5	0.36
ArcFace Scale (s)	30.0	17.57
Image Size	384 px	256 px
Batch Size	64	32
LR Scheduler	ReduceLROnPlateau	ReduceLROnPlateau (Patience 3)
Augmentation	Basic	Flip, Affine, Random Erasing
Background	Standard	Segmented
Post-processing	None	TTA, QE, K-reciprocal

Table 11: Comparison of Baseline vs. Final Architecture

Table 12: Final Hyperparameter Configuration for Jaguar Re-ID

Hyperparameter	**Final Selected Value**
Architecture Details	
Backbone	DINOv3 (Unfrozen)
Projection Head	3-Layer MLP
Hidden Dimension	1024
Output/Embedding Dimension	256
Dropout Rate	0.20
Optimization & Training	
Epochs	100
Batch Size	32
Base Learning Rate	1.599×10^{-4}
Backbone LR Multiplier	0.054
Weight Decay	1.406×102^{-4}
LR Scheduler	ReduceLROnPlateau (Patience: 3, Factor: 0.5)
Early Stopping Patience	10 Epochs
Loss Function (Focal ArcFace)	
ArcFace Margin (m)	0.360
ArcFace Scale (s)	17.570
Focal Gamma (γ)	2.111
Data & Pre-processing	
Input Resolution	256×256
Background Mode	Segmented Intervention
Random Horizontal Flip	p=0.5
Random Affine	10°, Translate: 0.05, Scale: 0.05
Random Erasing	p=0.5, Scale: (0.02, 0.25)

6 Individual Contributions

Vincent Eichhorn

Responsible for the description of the **Baseline** model in this report. Conducted a comprehensive suite of ablation studies and hyperparameter tuning, specifically focusing on early-stage architectural iterations:

- *Experiments:* 3.1.1–3.1.3 and 3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.6, 3.2.10.

Josef Pribbernow

Led the development of the **Final Architecture** and managed the **Submission** pipeline. Conducted advanced experiments involving loss function optimization and specialized training regimes:

- *Experiments:* 3.1.4 and 3.2.5, 3.2.7, 3.2.8, 3.2.9, 3.2.11, 3.2.12.

7 Appendix

Project Code All source code can be accessed through this repository at GitHub https://github.com/vincenteichhorn/cs-ahocv-jaguar-reid/