

Exploration of Unsupervised Segmentation as a Foundation for Vision Graph Neural Networks in Individual Animal Re-Identification

LifeCLEF at CLEF 2026

Sinan Pourazari¹

¹University of Siegen, Department of Electrical Engineering and Computer Science, Siegen, Germany

Abstract

Image-based individual animal re-identification is essential for non-invasive population monitoring. While grid-based architectures such as Vision Transformers (ViTs) achieve state-of-the-art performance, they rely on compute-intensive parameter spaces to resolve the non-rigid nature of an animal's morphology. In this paper, we explore a resource-efficient alternative by proposing a Spatial-Topological Vision Graph Attention Neural Network (ST-VGANN), an exploratory proof-of-concept with 1.75 million active inference parameters. The model represents animal morphology as an irregular topological graph, extracts superpixel-based texture primitives with an unsupervised Convolutional Autoencoder (CAE), and routes them through GATv2 layers. For the AnimalCLEF 2026 challenge, we present an ablation study across 38 configurations that maps the architectural search space, revealing insights into topological routing, latent dimensionality, and regularization redundancy. The standalone model achieved a Private Leaderboard ARI of 0.082 (Public: 0.072, Rank 211/230); an uncalibrated late-fusion ensemble with MegaDescriptor-L-384 reached 0.196 Private ARI (Public: 0.171, Rank 170/230). The Source code is available on GitHub.

Keywords

Wildlife Re-identification, Graph Neural Networks, GNN, Superpixels, Open-set Recognition, Geometric Deep Learning, Feature Disentanglement, VGNN, Vision-Graph, Vision Graph Neural Network

1. Introduction

Wildlife re-identification (Re-ID) relies on recognizing unique natural markings, such as the spot patterns of a leopard or the facial scales of a sea turtle. While grid-based architectures such as Vision Transformers (ViTs) [1, 2, 3, 4] frequently score high in competitive benchmarks in this domain, they typically handle the non-rigid, articulated nature of animals through large parameter spaces and compute-intensive scaling. In this paper, presented as part of the AnimalCLEF 2026 competition [5, 6] within the broader LifeCLEF evaluation campaign [7], we investigate the performance of an Spatial-Topological Vision Graph Attention Neural Network (ST-VGANN), a graph-based alternative that models biological geometry to explore a parameter-efficient approach. This framework aims to empower environmentalists and conservation institutions that lack access to high-performance computing resources, enabling them to train and deploy specialized models according to their local conservation needs. The specific configuration of the hardware used is detailed in Appendix A.

The AnimalCLEF 2026 challenge [6] targets the discovery step of wildlife monitoring: autonomously grouping images of individuals so that each cluster corresponds to a single animal, thereby assigning identities to previously unseen subjects. Unlike traditional closed-set verification that matches inputs against a known catalog, this open-set clustering task requires algorithms to build and maintain an identity reference structure under realistic, highly variable conditions. The competition features a morphologically diverse set of Animals, including Loggerhead Sea Turtles, Eurasian Lynxes, Texas Horned Lizards and Fire Salamanders, and introduces severe real-world challenges such as unconstrained

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

✉ sinan.pourazari@student.uni-siegen.de (S. Pourazari)

id 0009-0001-8174-0259 (S. Pourazari)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

poses, background clutter, and long-tailed class imbalance. To evaluate how effectively a system can handle these unconstrained environments without a predefined catalog, overall performance is measured using the Adjusted Rand Index (ARI). This metric assesses pairwise cluster consistency against ground-truth identities, penalizing both false merges and fragmented clusters.

We model each animal as an undirected graph where superpixel nodes are connected via spatial adjacency. This topological approach intends to short-circuit the resource-intensive filter-learning phase of standard convolutional networks. While grid-based architectures require large parameter spaces and extensive training datasets to hierarchically learn low-level edge, color, and texture filters, our framework deterministically supplies these primitives through unsupervised segmentation and structural descriptors.[8, 9, 10] By offloading low-level extraction, the subsequent Graph Neural Network focuses exclusively on macroscopic relational reasoning. This structural bottleneck lowers data dependency. Consequently, the architecture maintains a profile of 3.2 million active parameters during training and 1.75 million parameters for inference, presenting a parameter-efficient alternative to compute-heavy baseline architectures that trains entirely on consumer-grade hardware.

To ensure full reproducibility, our complete source code and experimental framework are publicly available on GitHub¹. Every performance metric and ablation benchmark reported in this paper can be independently reproduced on standard consumer hardware by following the repository’s documentation.

2. Related Work

Image based individual animal re-identification is enabling non invasive wildlife monitoring. Unlike human biometrics, wildlife identification relies on highly variable natural phenotypes (e.g. fur rosettes, facial scales) that are subject to severe perspective distortion and environmental occlusion, necessitating highly robust extraction pipelines.

Global and Local Descriptors. State-of-the-art approaches typically employ deep global descriptors, local keypoint matchers, or hybrid ensembles. Foundation models like MegaDescriptor-L-384 [1] extract global semantic embeddings through large-scale, multi-taxonomic pre-training. Recent frameworks like WildFusion [11] demonstrate the effectiveness of late-fusion ensembling, combining these global priors with calibrated local correspondences to filter false matches across shifting viewpoints.

Metric Learning and Open-Set Retrieval. Open-set ecological retrieval requires robust metric learning. Architectures frequently employ the ArcFace angular-margin penalty [12] to maximize inter-class separability, a highly resilient approach against the extreme few-shot and long-tail constraints of datasets like WildlifeReID-10k [13].

Vision Graph Neural Networks. To address the limitations of traditional convolutional and attention-based architectures, Vision Graph Neural Networks (VGNNs) have emerged within the broader framework of Geometric Deep Learning [8]. Foundational models, such as Vision GNN (ViG) [14], conceptualize images as graph structures by partitioning visual data into nodes and propagating information via anisotropic message passing. While models like ViG [14] demonstrate the potential of Vision Graph Neural Networks, their architecture relies on uniform grid-based patching to generate graph nodes. We build upon their findings but pivot away from this grid-based paradigm, as uniform patches may not optimally align with the non-rigid, highly irregular morphology of natural animal markings. Instead, we utilize superpixel segmentation (SEEDS) to allow the graph topology to natively conform to the natural contours and boundaries of the animal. We complement this geometric approach with metric learning techniques and specialized node feature formulations.

¹<https://github.com/Sinan-Pourazari/Wildlife-Re-ID/tree/AnimalCLEF2026>

3. Evaluation Metric: Adjusted Rand Index (ARI)

Because individual re-identification operates in an open-set scenario, where test identities are disjoint from the training set and their total number is unknown, we frame the evaluation as a clustering problem and utilize the Adjusted Rand Index (ARI).

The standard Rand Index (RI) measures the similarity between the predicted clustering and the ground-truth partition by evaluating all pairs of samples. Given a set of N total images, let a be the number of image pairs correctly grouped into the same identity cluster, and let b be the number of pairs correctly separated into different identity clusters. The RI is defined as the ratio of these correct decisions to all possible pairs:

$$RI = \frac{a + b}{\binom{N}{2}} \quad (1)$$

However, the standard RI fails to account for chance agreements. To establish a statistically reliable metric, the ARI corrects the RI by subtracting its expected value, $E[RI]$, assuming a hypergeometric distribution where cluster assignments are made randomly:

$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \quad (2)$$

where $\max(RI)$ represents the maximum theoretical value of the Rand Index (which is 1.0). The resulting ARI score is bounded between $[-1.0, 1.0]$. A score of 1.0 indicates that the predicted clusters match the true underlying animal identities. A score of 0.0 indicates that the clustering performance is no better than random chance, while negative scores indicate systematic misclassifications worse than random chance.

4. Methodology

ST-VGANN departs from traditional computer vision approaches by modeling an animal’s morphological structure as an irregular graph rather than a rigid pixel grid. Our methodology is a workflow divided into (1) geometric pre-processing, (2) graph construction, (3) feature extraction, (4) augmentation and caching, (5) relational reasoning, (6) metric learning and (7) clustering and reranking.

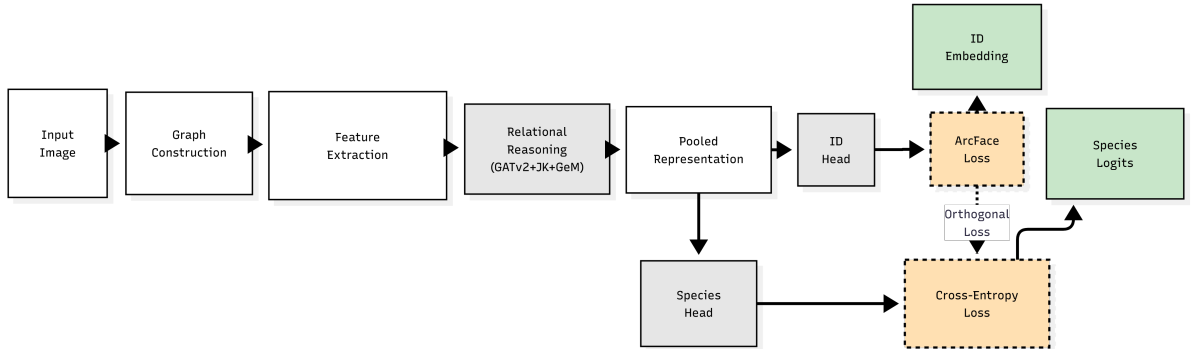


Figure 1: ST-VGANN architecture overview: input image, graph construction, feature extraction, GATv2 with Jumping Knowledge and GeM pooling, dual ID and Species heads, and ArcFace, Orthogonal, and Cross-Entropy losses.

Geometric Preprocessing To preserve biometric proportions, inference uses aspect-ratio-aware resizing with zero-padding. Training uses Random Resized Cropping for robustness. Our GNN handles padded regions natively, avoiding explicit background pruning.

Graph Construction We define the animal subject as an undirected graph $G = (V, E)$, where V denotes the set of superpixel nodes and E the edge set.

We utilize the SEEDS algorithm [15] to segment a given image. An input image is partitioned into N superpixels. Unlike standard grid-based patches, our goal is for these superpixels to adaptively conform to the natural contours and morphological boundaries of the animal. An example of the target output is shown in Figure 2.

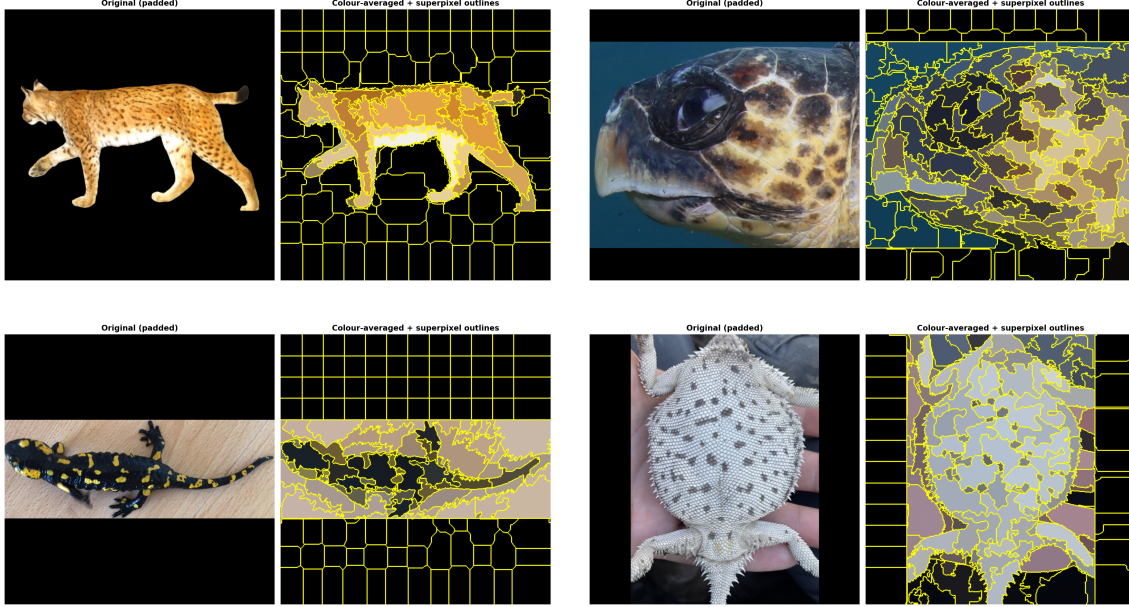


Figure 2: Examples from the evaluated datasets [1, 13, 16, 17, 18]. **Top left:** Eurasian Lynx. **Top right:** Sea Turtle. **Bottom left:** Salamander. **Bottom right:** Texas Horned Lizard. Individual identification relies on highly variable natural markings, ranging from fur patches to facial scale patterns that are subject to perspective distortion.

To evaluate the impact of graph connectivity on feature propagation, we study three distinct topological routing strategies in Section 6:

- **Spatial Adjacency:** Edges are defined by physical contiguity (sharing a pixel boundary), serving as our control for local-only inductive bias.
- **Latent Adjacency:** Edges are constructed via k -NN search in a projected latent feature space, connecting nodes based on morphological similarity rather than physical proximity.
- **Hybrid Topological Manifold:** The boolean union of spatial and latent edges, intended to preserve local textural continuity while facilitating global relational reasoning.

Feature Extraction To enable relational reasoning, each node v_i gets a feature vector x_i that concatenates several modalities. These features are explicitly chosen to capture both macroscopic geometry and localized biological motifs:

- **Position (pos):** The normalized centroid coordinates (x, y) of the superpixel. This provides spatial awareness, allowing the network to encode where a specific texture is located on the animal’s body.
- **Shape (shape):** Geometric descriptors of the superpixel boundary, including total area, perimeter, and eccentricity, which quantify the physical footprint of the segmented region.
- **Color (color):** The mean pixel values extracted using the CIELAB color space.
- **Texture (texture):** A statistical measure computed via Local Shannon Entropy over the grayscale image. This scalar acts as a roughness indicator, efficiently differentiating smooth backgrounds from textured surfaces.

- **HOG (hog)**: Histograms of Oriented Gradients. This descriptor pools gradient magnitudes into angular bins, capturing macroscopic structural boundaries (e.g., the curve of a head or the edge of a facial scale) invariant to local illumination.
- **LBP (lbp)**: Local Binary Patterns. By thresholding local pixel neighborhoods, LBP captures micro-textural geometries such as corners and small spots. It provides a signal that is strictly invariant to monotonic grayscale transformations.
- **Latent Texture (cae)**: While statistical descriptors provide a computationally inexpensive baseline, they lack deep semantic capacity. To bridge this, a Convolutional Autoencoder is trained to reconstruct superpixel patches, producing a dense latent vector. During graph construction, the input patch is average-masked to prevent information leakage from neighboring nodes, ensuring the latent vector represents only the given Superpixel-node.

Augmentation and Caching To address computational bottlenecks, input images undergo pixel-level augmentations, including random horizontal flipping, scaling, and color jitter, before we implement state-persistent caching utilizing LMDB (Lightning Memory-Mapped Database). A deterministic hashing mechanism maps hyperparameters to stored graphs, drastically reducing per-epoch latency. Further details of this implementation can be found in Appendix B.

At runtime, we employ three distinct dropout regularization techniques: node dropout (randomly dropping nodes), edge dropout (randomly dropping edges), and modality dropout, which randomly zeroes out entire feature blocks for a given node to prevent over-reliance on a single dominant modality.

Relational Reasoning Relational reasoning is handled by a deep Graph Attention Network (GATv2) [19]. Unlike isotropic Graph Convolutional Networks, GATv2 performs anisotropic message passing, dynamically assigning importance to specific neighbors based on both morphological content and relative physical geometry.

Edge-Conditioned Message Passing We encode the physical geometry between connected nodes into a multi-dimensional edge attribute vector $\mathbf{e}_{ij}$ consisting of the normalized Euclidean distance and the sine/cosine of the directional angle. The GATv2 layer computes a dynamic attention coefficient α_{ij} conditioned on the source node, target node, and geometric edge attributes:

$$\alpha_{ij}^{(l)} = \text{softmax}_j \left(\mathbf{a}^T \text{LeakyReLU} \left(\mathbf{W}^{(l)} \vec{h}_i^{(l)} \parallel \mathbf{W}^{(l)} \vec{h}_j^{(l)} \parallel \mathbf{W}_e^{(l)} \mathbf{e}_{ij} \right) \right) \quad (3)$$

where l denotes the current layer index, and $\vec{h}_i^{(l)}$ and $\vec{h}_j^{(l)}$ represent the hidden feature representations of the source node i and target node j , respectively. The symbol $\parallel$ denotes vector concatenation. The matrices $\mathbf{W}^{(l)}$ and $\mathbf{W}_e^{(l)}$ are learnable linear weight transformations applied to the node features and edge attributes. Finally, $\mathbf{a}^T$ is a learnable attention weight vector, and softmax_j normalizes the resulting attention coefficients across all neighbors of node i .

To mitigate over-smoothing, where deep node representations become indistinguishable, we implement Dynamic Jumping Knowledge (JK) [20]. This mechanism aggregates intermediate representations from all hidden layers using an attention-weighted sum. As a result, the global pooling stage retains both localized textural features and broad structural semantics.

Metric Learning and Feature Disentanglement To isolate individual-specific traits from dominant species-specific characteristics, we employ a dual-headed optimization strategy refining the pooled graph representation vector. Following global graph pooling, the unified graph embedding is propagated through two parallel, independent Multi-Layer Perceptron (MLP) projection networks. These projection heads act as a structural bottleneck, transforming the high-dimensional global graph vector into two distinct latent vectors.

The main Re-ID head utilizes Sub Center ArcFace [21] to map the identity graph embeddings onto a hypersphere, handling severe intra-class pose variations. Simultaneously, the auxiliary species

Table 1

Kaggle Leaderboard Results (Team: Sinan Pourazari). Ranks are reported out of 230 participating teams.

System Variant	Phase	ARI Score	Rank
Standalone ST-VGANN	Public	0.07213	209
	Private	0.08206	211
ST-VGANN + WildFusion	Public	0.17070	172
	Private	0.19616	170

classification head provides a supervised species scaffold. To prevent these taxonomic features from leaking into the identity embedding, we apply an orthogonal loss regularization term across the projected representations:

$$\mathcal{L}_{ortho} = \|\mathbf{E}_{id}^T \mathbf{E}_{species}\|_F^2 \quad (4)$$

where $\mathbf{E}_{id}$ and $\mathbf{E}_{species}$ represent the batch matrices of the projected identity and species embeddings, T denotes the matrix transpose, and $\|\cdot\|_F^2$ denotes the squared Frobenius norm. This mathematically penalizes the correlation between the two latent spaces, forcing the identity head to ignore general species-level traits in favor of individual-specific markers. The learning rate was adjusted using a ReduceLROnPlateau scheduler that monitored total batch loss. To allow for larger batch sizes, which are required for effective SubCenter ArcFace hard-negative mining, we reduced the memory footprint by utilizing 8-bit quantized Adam optimization [22] and bfloat16 mixed-precision.

Open-Set Clustering and Reranking The initial pairwise distances between the graph embeddings are computed via standard cosine similarity. Because non-rigid biological subjects generate noisy manifolds, we apply k -reciprocal Jaccard reranking [23] (configured with a neighborhood size of $k_1 = 12$ and a penalty weight of $\lambda = 0.2$) to heavily penalize unidirectional similarities. The adjusted distances are clustered using HDBSCAN [24], which gracefully isolates morphological outliers as noise rather than forcing them into incorrect identity clusters. For the clustering phase, we fix the minimum samples parameter to 2. To generate the final competition submission, the resulting HDBSCAN cluster labels were serialized as string IDs in the form of `cluster_OriginDatasetName_ClusterID`. These predicted IDs were mapped to their corresponding `image_id` in a CSV format.

5. Implementation and Competition Results

For the AnimalCLEF 2026 Kaggle competition, we participated under the team name Sinan Pourazari. The competition evaluation was divided into a Public Leaderboard, calculated on approximately 52% of the test data, and a Private Leaderboard, calculated on the remaining 48% to determine final standings, using the Adjusted Rand Index (ARI) as the evaluation metric.

Table 1 details our competition submissions. Evaluated as a standalone framework, our ST-VGANN achieved a Private Leaderboard score of 0.08206 ARI. To contextualize this representation gap, the official AnimalCLEF baseline—a grid-based ensemble utilizing MegaDescriptor-L-384 (228.8M parameters) [1] and MIEWId [25] (51.11M parameters) via WildFusion [11]—achieved a Private Leaderboard score of 0.20646.

For our final submission, we deployed an uncalibrated late-fusion ensemble via WildFusion. Combining our topological features with MegaDescriptor-L-384’s global priors elevated our Private Leaderboard score to 0.19616 (Appendix C). The underlying embedding space explains this performance gap (Figure 3): while the standalone architecture successfully learns basic morphological distinctions, it struggles with fine-grained clustering.

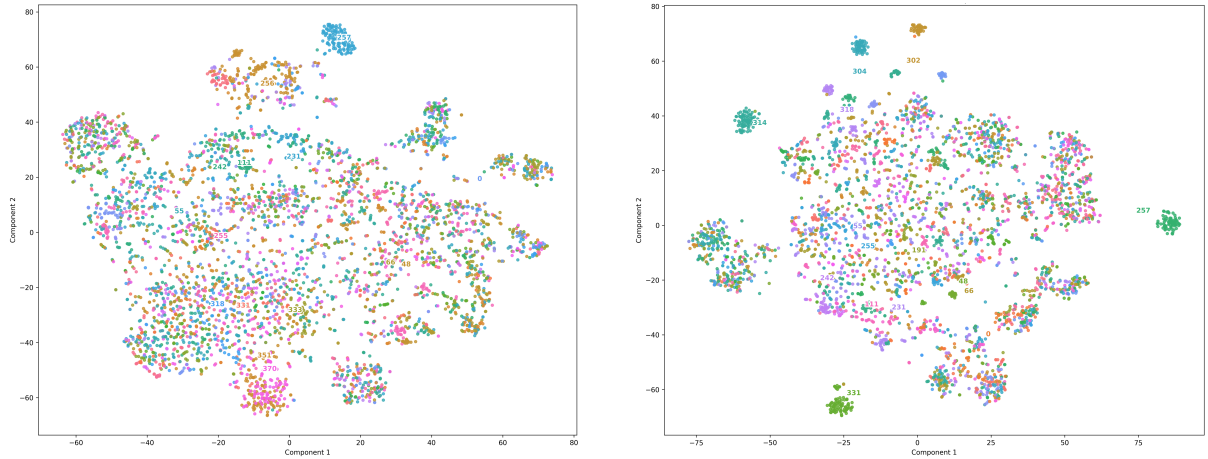


Figure 3: t-SNE of mixed test split embeddings with HyenaID2022 serving as the out-of-distribution (OOD) class. **Left:** Standalone ST-VGANN. **Right:** WildFusion ensemble.

5.1. Hardware and Parameter Footprint

To ensure deployability on consumer hardware, ST-VGANN maintains a highly constrained parameter space, offsetting the representation gap seen in the leaderboard results. As detailed in Table 2, textural features are extracted offline using a frozen Convolutional Autoencoder (1.82M parameters). During training, the active footprint is 3.29M parameters to accommodate species disentanglement and Sub-Center ArcFace metric learning. After training, these auxiliary branches are discarded, yielding a final inference model of 1.75M active parameters.

Table 2
ST-VGANN Parameter Distribution

Component	Parameters	Role
GNN + Identity Head	1,752,834	Active Inference
Species Disentanglement	527,875	Training Only
SubCenter ArcFace	1,013,760	Training Only
Texture Encoder (CAE)	1,821,181	Frozen Extractor

6. Ablation Study

We evaluated 38 configurations across graph topology, feature engineering, regularization, and data scaling. Table 3 reports the Known Mixed ARI (both native and oracle-tuned) alongside Kaggle Leaderboard scores. Two evaluation states are reported: **Native GNN** uses static HDBSCAN parameters without reranking, while **Oracle-Tuned** applies k -reciprocal Jaccard reranking with HDBSCAN parameters optimized to establish a theoretical performance ceiling. Finally, regarding network configurations, 1-layer shallow probes naturally omit the Jumping Knowledge (JK) mechanism, as a single-layer topology provides no intermediate representations to aggregate.

The Texas Horned Lizard dataset provided for this challenge did not include the ground-truth identity labels required for our specific local ablation splits. Consequently, the Texas Horned Lizard could not be included in the ablation evaluations in Table 3. To maintain a fair evaluation of the model’s capacity to handle out-of-distribution (OOD) taxa locally, we instead utilize the Hyena dataset [26] as a stand-in proxy for zero-shot generalization. However, because the official AnimalCLEF hidden test set contains these concealed identities, the Kaggle Leaderboard scores (Public/Private) reported in Table 3 do include the Texas Horned Lizard.

Table 3

Ablation results across all evaluation domains. **Baseline:** GATv2 3L, Spatial 1-Hop Edges, CAE Dim 90, Emb Dim 512, GeM Pooling, Ortho Loss, All Dropouts, {Color, Pos, CAE} features. **Nat** = Native GNN; **Opt** = Oracle-Tuned k -reciprocal upper bound; **K-Mix** = Known Mixed; **G-Mix** = Global Mixed (seen + unseen species); **Sal.** = Salamander; **Kgl.** = Kaggle (Pub / Priv). All scores are ARI. **Config notes:** Latent Only: spatial $\rightarrow$ latent k -NN edges; Hybrid: spatial $\cup$ latent; HOG/Color/CAE Only: single-modality feature; +All Hand.: adds HOG, LBP, Texture, Shape; Mask-T/S/All: U²-Net foreground mask for Turtle/Salamander/All; +Ext.Data: adds ATRW, Amvrakikos, Leopard.

Category	Config	L	K-Mix		G-Mix		Hyena		Lynx		Sal.		Turtle		Kgl. P/Pr	
			Nat	Opt	Nat	Opt	Nat	Opt	Nat	Opt	Nat	Opt	Nat	Opt	Nat	Opt
Baseline	3L Base	3	0.349	0.345	0.005	0.174	0.002	0.089	0.480	0.490	0.034	0.057	0.213	0.222	0.019/0.035	0.040/0.056
	1L Base	1	0.255	0.304	0.003	0.120	0.001	0.080	0.261	0.394	0.033	0.056	0.194	0.243	0.013/0.025	0.037/0.080
Edges	Latent Only	3	0.284	0.341	0.003	0.147	0.001	0.072	0.160	0.485	0.017	0.068	0.242	0.229	0.015/0.052	0.030/0.055
	Hybrid 3L	3	0.323	0.349	0.005	0.186	0.002	0.099	0.417	0.480	0.029	0.068	0.257	0.248	0.020/0.062	0.054/0.068
	Hybrid 2L	2	0.330	0.380	0.004	0.176	0.001	0.090	0.456	0.498	0.073	0.025	0.291	0.308	0.011/0.048	0.041/0.084
Depth	GCN 1L	1	0.106	0.315	0.001	0.144	0.000	0.073	0.208	0.459	0.011	0.053	0.076	0.178	0.018/0.025	0.040/0.076
	GCN 2L	2	0.247	0.358	0.003	0.141	0.001	0.075	0.224	0.494	0.011	0.067	0.193	0.217	0.024/0.029	0.044/0.074
	JK-Mean	3	0.355	0.359	0.005	0.191	0.002	0.104	0.475	0.525	0.038	0.074	0.219	0.262	0.033/0.057	0.042/0.083
	No JK	3	0.359	0.372	0.006	0.192	0.002	0.094	0.452	0.489	0.055	0.051	0.261	0.272	0.036/0.075	0.041/0.098
	GAT 2L	2	0.303	0.364	0.005	0.157	0.002	0.083	0.205	0.479	0.043	0.073	0.237	0.244	0.025/0.049	0.030/0.057
Features	Color Only	1	0.001	0.006	0.000	0.000	0.000	0.001	0.130	0.133	0.007	0.013	0.000	0.000	0.001/0.001	0.000/—0.002
	HOG Only	1	0.068	0.275	0.000	0.071	0.001	0.040	0.380	0.365	0.009	0.072	0.022	0.188	0.000/0.001	0.014/0.029
	No Color	1	0.073	0.149	0.009	0.050	0.015	0.034	0.118	0.235	0.009	0.027	0.021	0.031	0.019/0.027	0.014/0.029
	+HOG	1	0.292	0.300	0.003	0.108	0.001	0.064	0.438	0.432	0.057	0.069	0.199	0.250	0.012/0.024	0.038/0.053
	+Shape	1	0.257	0.294	0.006	0.161	0.003	0.063	0.323	0.421	0.012	0.037	0.210	0.205	0.002/—0.000	0.021/0.025
	CAE Only	1	0.135	0.168	0.026	0.053	0.026	0.033	0.151	0.212	0.025	0.030	0.067	0.081	0.005/0.019	0.002/0.003
	+All Hand.	1	0.312	0.347	0.003	0.191	0.001	0.088	0.443	0.497	0.041	0.076	0.284	0.274	0.029/0.055	0.038/0.057
CAE Dim	CAE-32	1	0.302	0.365	0.004	0.205	0.001	0.086	0.301	0.479	0.076	0.030	0.245	0.287	0.013/0.018	0.046/0.093
	CAE-64	1	0.276	0.303	0.003	0.167	0.001	0.084	0.421	0.440	0.062	0.021	0.190	0.222	0.013/0.035	0.040/0.058
	CAE-128	1	0.264	0.303	0.004	0.159	0.002	0.077	0.408	0.432	0.058	0.020	0.167	0.227	0.010/0.010	0.037/0.046
Emb Dim	Emb-32	1	0.034	0.254	0.001	0.019	0.001	0.007	0.072	0.152	0.005	0.014	0.032	0.112	0.000/0.000	0.003/0.050
	Emb-256	1	0.264	0.269	0.003	0.134	0.001	0.070	0.401	0.425	0.050	0.009	0.150	0.171	0.010/0.019	0.025/0.047
	Emb-1024	1	0.268	0.299	0.004	0.139	0.001	0.077	0.400	0.455	0.032	0.067	0.171	0.235	0.012/0.021	0.034/0.071
Pooling	Mean	1	0.092	0.353	0.000	0.136	0.001	0.061	0.407	0.453	0.005	0.056	0.026	0.265	0.013/0.003	0.040/0.094
	Max	1	0.123	0.181	0.003	0.071	0.004	0.050	0.219	0.346	0.031	0.025	0.049	0.162	0.001/0.012	0.000/0.003
Regul.	No Ortho	1	0.281	0.324	0.003	0.151	0.001	0.082	0.367	0.453	0.026	0.067	0.199	0.229	0.018/0.035	0.043/0.059
	No Drop	3	0.365	0.334	0.006	0.157	0.002	0.090	0.498	0.516	0.051	0.041	0.233	0.254	0.028/0.041	0.031/0.073
	No ModDrop	3	0.345	0.340	0.004	0.175	0.002	0.089	0.436	0.497	0.020	0.048	0.223	0.234	0.030/0.044	0.058/0.064
	No GrDrop	3	0.360	0.342	0.005	0.169	0.002	0.089	0.448	0.487	0.069	0.032	0.213	0.250	0.024/0.064	0.046/0.058
SP Count	50 SP	1	0.222	0.222	0.004	0.113	0.004	0.060	0.351	0.382	0.017	0.041	0.089	0.180	0.018/0.028	0.025/0.032
	500 SP	1	0.329	0.346	0.004	0.188	0.001	0.118	0.422	0.480	0.051	0.074	0.230	0.229	0.014/0.050	0.055/0.060
Masking	Mask-T	3	0.206	0.401	0.005	0.208	0.001	0.071	0.206	0.369	0.031	0.013	0.153	0.179	0.012/0.012	0.016/0.030
	Mask-S	3	0.011	0.043	0.001	0.007	0.001	0.008	0.166	0.424	0.145	0.121	0.000	0.000	0.000/0.001	0.008/0.031
	Mask-All	3	0.338	0.351	0.006	0.174	0.002	0.091	0.470	0.483	0.033	0.065	0.224	0.240	0.041/0.039	0.039/0.090
Domain	+Ext.Data	3	0.253	0.366	0.005	0.230	0.002	0.113	0.153	0.483	0.012	0.061	0.157	0.269	0.015/0.008	0.044/0.077
	Lynx Only	3	0.008	0.036	0.000	0.012	0.001	0.014	0.142	0.272	0.057	0.020	0.000	0.001	0.001/0.000	0.005/0.016
	Sal. Only	3	0.011	0.043	0.001	0.006	0.000	0.006	0.053	0.424	0.141	0.134	0.000	0.000	0.001/0.002	0.003/0.036
	Turtle Only	3	0.192	0.395	0.004	0.213	0.001	0.075	0.170	0.407	0.028	0.011	0.098	0.151	0.040/0.021	0.027/0.031

Graph topology and routing. Spatial adjacency consistently outperforms latent adjacency, which collapses native Eurasian Lynx clustering to 0.160 ARI by short-circuiting the structural hierarchy. This failure is particularly pronounced on the CzechLynx dataset, as latent routing struggles to map the soft, fuzzy boundaries of the Lynx’s fur and fails to generalize across the dataset’s unique inclusion of synthetic image samples. Three GATv2 layers with attention-based Jumping Knowledge (JK) is the strongest configuration: JK prevents over-smoothing by blending localized early-layer textures with deep structural semantics, and its removal degrades performance on texture-sensitive species despite marginally higher native scores on the Salamander. Shallow (1L) probes establish a consistent performance floor. A micro-granularity of 500 superpixels slightly elevates the oracle ceiling but incurs

disproportionate compute cost making 240 SP seem as the better practical trade-off however the optimal number of superpixels is highly dependent on the viewed animal species.

Feature engineering and dimensionality. The CAE latent vector is only effective in combination with Color, Position, and HOG; used alone it fails entirely. HOG contributes the largest marginal gain by providing illumination-invariant edge structure. Shape descriptors consistently degrade performance. Smaller CAE latent dimensions (32 vs. 90) improve the oracle ceiling by forcing the encoder to discard environmental noise rather than memorising it. The global embedding dimension exhibits a non-linear curve: Dim 256 and Dim 1024 both outperform the baseline Dim 512, suggesting a capacity-interference regime at intermediate sizes where noise is encoded but not orthogonalised.

Regularisation, pooling, and data. GeM pooling is strongly preferable to Mean or Max; Mean pooling underperforms on Sea Turtles (0.026 ARI) despite a reasonable oracle ceiling, indicating high hyperparameter sensitivity. Dropout of any kind produces negligible improvement, suggesting that superpixel segmentation natively acts as a structural bottleneck against overfitting. The Orthogonality Loss proves beneficial for feature disentanglement, albeit only slight in most instances; removing it degrades performance across most domains, notably causing a severe drop in the native Lynx ARI from 0.480 to 0.367. Single-species specialists underperform the generalist on all species except Salamander, where removing the large marine corpus frees representational capacity for high-contrast skin patterns. Finally, the persistent gap between native and oracle-tuned Kaggle scores across all configurations indicates that the primary bottleneck is clustering calibration rather than embedding quality.

Oracle Tuning and Reranking. For local splits, oracle HDBSCAN parameters were optimized directly on ground-truth labels. For the Kaggle Leaderboard (where test labels are hidden), we used a labeled training subset as a tuning proxy. Table 3 demonstrates that proxy-tuning consistently outperforms the static baseline, offering a viable real-world deployment strategy. Note, however, that the Oracle-Tuned performance gains stem from the combined benefits of both optimal HDBSCAN parameters and k -reciprocal Jaccard reranking.

7. Discussion and Future Work

While ST-VGANN establishes a theoretical foundation for parameter-efficient clustering on known species, it fundamentally fails to generalize to zero-shot domains, and the clustering manifold collapses when known and unknown species are mixed. Despite this, the architecture’s efficiency (8.26 s/epoch on consumer hardware) demonstrates that broad architectural search spaces are explorable without HPC access, yet with the current performance, this training efficiency does not yet outweigh the performance drop compared to foundation models like MegaDescriptor-L-384. Substantial refinement in feature extraction and routing remains necessary before the approach can reliably handle closed-set and especially open-set deployment.

For future work, several architectural bottlenecks remain to be addressed. First, we hypothesize significant potential in optimizing the graph construction phase. The impact of static parameters for the SEEDS segmentation proved to be inefficient across diverse morphological domains. As visualized in Figure 2, the quality of superpixel segmentation varies greatly from image to image. While standard superpixel boundaries can capture aspects of a Sea Turtle’s geometry, they frequently fail to cleanly isolate natural markings, often arbitrarily bisecting facial scales. Furthermore, these same static parameters severely struggle to resolve the fur textures and patterns of the Eurasian Lynx. Because of this high variance in segmentation quality, transitioning from static configurations to image-based or species-based hyperparameter tuning is essential. Alternatively, evaluating different superpixel algorithms may lead to more consistent node generation.

Second, while our approach relies heavily on local textural descriptors, it currently lacks a global feature perspective; integrating global semantic descriptors into the node features could further enhance

performance. Third, exploring alternative metric learning strategies, such as contrastive triplet loss, may yield more robust identity clustering.

Finally, the most significant limitation of our current pipeline is the static nature of the Convolutional Autoencoder (CAE). While the CAE provides efficient texture features, its weights are frozen prior to graph construction. Enabling end-to-end co-training of the CAE alongside the rest of ST-VGANN would allow the network to dynamically adapt its feature extraction to the metric learning objective. While this would increase training time and resource demands, it represents a natural next step for improving the generalizability of this architecture.

7.1. Suitability of Unsupervised Superpixel Segmentation for Re-ID

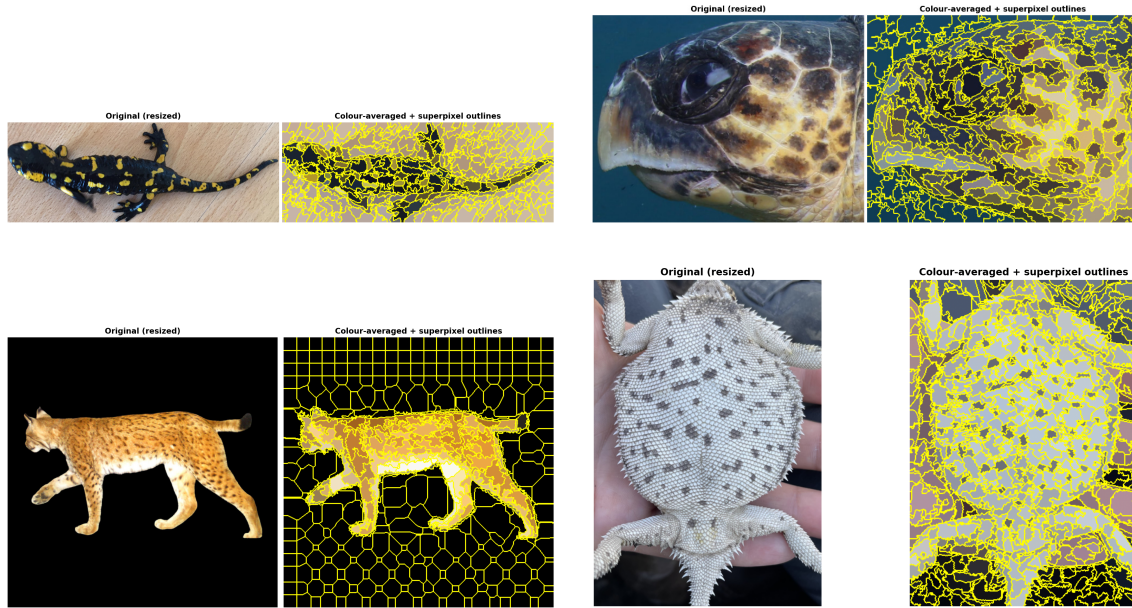


Figure 4: Examples from the evaluated datasets [1, 13, 16, 17, 18]. **Top left:** Salamander. **Top right:** Sea Turtle. **Bottom left:** Eurasian Lynx. **Bottom right:** Texas Horned Lizard. Individual identification relies on highly variable natural markings, ranging from fur patches to facial scale patterns that are subject to perspective distortion.

Visual analysis of the SEEDS outputs (Figure 2) raises valid questions regarding the suitability of unsupervised superpixels for fine-grained re-identification. Because SEEDS operates strictly on low-level color and intensity gradients, the resulting graph does not map the animal’s physical morphology (e.g., the physical ridges between scales), but rather its pigmentation topology.

Note that the colour-averaged visualisations in Figures 2 and 4 are illustrative only. Feature extraction operates directly on the pixel data within each superpixel boundary, so even if a single superpixel contains multiple spots or markings, they are fully encoded into the high-dimensional node embeddings via the CAE, HOG, and related descriptors.

The viability of this unsupervised graph construction is highly dependent on the visual contrast of the species’ natural markings. For animals with high-contrast, solid pigmentation motifs, the approach demonstrates higher efficacy. The algorithm isolates the bright yellow spots of the Fire Salamander against its black skin, and highlights the distinct dark dorsal dots of the Texas Horned Lizard as independent, localized nodes. For the Loggerhead Sea Turtle, while the superpixels do not trace the physical sutures of the facial scales, they isolate the distinct dark patches on the turtle’s face. However, this introduces a specific vulnerability: because the algorithm groups by intensity, specular reflections (glare from wet skin or scales) are frequently isolated into distinct, artificial nodes, introducing environmental noise into the biological graph.

Conversely, the algorithm struggles significantly with the Eurasian Lynx. Unlike the solid, high-contrast rosettes of a leopard or Salamander, the spots on a lynx consist of soft, gradient-based color transitions embedded in dense fur. Consequently, the unsupervised segmentation cannot cleanly resolve these fuzzy boundaries, leading to arbitrary fragmentation of the animal’s legs and torso.

Furthermore, while isolated evaluation data for the Texas horned lizard is omitted in Table 3, Figures 2 and 4 reveal a critical structural constraint. Capturing the highly intricate surface topologies of such species requires generating a high density of small superpixels. Although scaling up the superpixel count preserves essential phenotypic details, it drastically inflates graph size and node cardinality, introducing a severe computational trade-off.

Ultimately, this visual analysis correlates directly with our ablation results (Table 3): ST-VGANN establishes positive theoretical upper bounds on high-contrast amphibians and reptiles, but struggles natively on mammals with soft textures. We concede that static, unsupervised SEEDS segmentation is insufficient as a standalone solution for diverse taxa. However, as demonstrated in Figure 4, applying alternative, highly detailed hyperparameters to the SEEDS algorithm significantly alters the resulting morphological mapping. These visualizations highlight the algorithm’s sensitivity to its configuration parameters, suggesting that our current static approach creates a severe structural bottleneck. To resolve this trade-off between capturing hyper-fine structural patterns and maintaining execution efficiency on consumer hardware, future work must move beyond rigid unsupervised parameters. We propose a transition toward adaptive, multi-scale segmentation strategies, image-aware tuning, or, more fundamentally, end-to-end learned semantic superpixels (such as Superpixel Sampling Networks or differentiable SLIC). Although our standalone architecture did not achieve state-of-the-art performance, it successfully extracted a fundamental morphological signal, albeit one that is currently noisy and insufficiently expressive. By resolving this structural bottleneck, coupling the topological reasoning capabilities of Graph Neural Networks with semantic segmentation provides a highly interesting direction for future research.

Acknowledgments

We extend our gratitude to Moritz Baumgart for valuable discussions that shaped the species disentanglement approach, ultimately inspiring the framework of scaffold learning, as well as for providing constructive feedback on the manuscript. We are equally grateful to all the individuals and organizations involved in the collection and annotation of the data that makes the training of these models and the development of animal re-identification tools possible.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini 3.5 Flash (29.06.2026) in order to check grammar and spelling, refine clarity in selected sections, and assist with LaTeX formatting. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] V. Čermák, L. Pícek, L. Adam, K. Papafitsoros, Wildlifedatasets: An open-source toolkit for animal re-identification, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024, pp. 5953–5963.
- [2] N. Semenova, Meta-algorithm for open-set animal re-id: Wildfusion, xgboost, and dual-backbone arcface, in: CEUR Workshop Proceedings, volume 4038, 2025, pp. 3154–3165.
- [3] J. Tan, A. Wang, Open-set animal re-identification via multilevel feature fusion, Working Notes of CLEF (2025).

- [4] R. Pakhomov, G. Demidov, K. Bogdan, S. Lansikh, D. Dinmuhametov, A. Khlopotnykh, Individual wildlife recognition via hybrid global-local matching and segmentation-aware filtering, in: Working Notes of CLEF 2025 - Conference and Labs of the Evaluation Forum, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_251.pdf.
- [5] L. Adam, L. Pícek, K. Papafitsoros, D. Williams, D. Biffi, Animalclef26 @ cvpr & clef, <https://kaggle.com/competitions/animal-clef-2026>, 2026. Kaggle.
- [6] L. Adam, K. Papafitsoros, D. A. Williams, D. Biffi, L. Pícek, Overview of AnimalCLEF 2026: Discovery and re-identification of individual animals, in: Working Notes of CLEF 2026 - Conference and Labs of the Evaluation Forum, 2026.
- [7] L. Pícek, L. Adam, S. Kahl, R. Bossy, L. Chrobak, H. Goëau, K. Papafitsoros, H. Klinck, W.-P. Vellinga, R. Planqué, T. Denton, K. Barnard, C. Nédellec, L. Deléger, M. Courtin, G. Martellucci, I. Moummad, F. Vinatier, P. Bonnet, A. Joly, Overview of LifeCLEF 2026: Ai challenges for biodiversity understanding and ecosystem management, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2026.
- [8] M. M. Bronstein, J. Bruna, T. Cohen, P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021. URL: <https://arxiv.org/abs/2104.13478>. arXiv: 2104.13478.
- [9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net, 2021. URL: <https://openreview.net/forum?id=YicbFdNTTy>.
- [10] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.
- [11] V. Cermak, L. Pícek, L. Adam, L. Neumann, J. Matas, Wildfusion: Individual animal identification with calibrated similarity fusion, in: A. Del Bue, C. Canton, J. Pont-Tuset, T. Tommasi (Eds.), Computer Vision – ECCV 2024 Workshops, Springer Nature Switzerland, Cham, 2025, pp. 18–36.
- [12] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
- [13] L. Adam, V. Cermak, K. Papafitsoros, L. Pícek, Wildlifereid-10k: Wildlife re-identification dataset with 10k individual animals, in: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) Workshops, 2025, pp. 2099–2109.
- [14] K. Han, Y. Wang, J. Guo, Y. Tang, E. Wu, Vision gnn: An image is worth graph of nodes, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 8291–8303. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/3743e69c8e47eb2e6d3afaea80e439fb-Paper-Conference.pdf.
- [15] M. Van den Bergh, X. Boix, G. Roig, L. Van Gool, Seeds: Superpixels extracted via energy-driven sampling, International Journal of Computer Vision 111 (2015) 298–314.
- [16] L. Adam, V. Čermák, K. Papafitsoros, L. Pícek, Seaturtleid2022: A long-span dataset for reliable sea turtle re-identification, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 7146–7156.
- [17] D. Biffi, M. R. Tucker, A. Ackel, D. A. Williams, Identification of individual texas horned lizards (*phrynosoma cornutum*) using genotypes and ventral spot patterns, Ecology and Evolution 15 (2025) e71167.
- [18] L. Pícek, J. Straka, M. Jirik, E. Belotti, M. Dul’a, J. Krausová, M. Bojda, V. Cermak, L. Bufka, R. Dvořák, et al., Czechlynx: A dataset for individual identification and pose estimation of the eurasian lynx, Scientific Data 13 (2026) 511.
- [19] S. Brody, U. Alon, E. Yahav, How attentive are graph attention networks?, in: International Conference on Learning Representations, 2022. URL: <https://openreview.net/forum?id=F72ximsx7C1>.
- [20] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, S. Jegelka, Representation learning on graphs with jumping knowledge networks, in: International conference on machine learning, pmlr, 2018,

- pp. 5453–5462.
- [21] J. Deng, J. Guo, T. Liu, M. Gong, S. Zafeiriou, Sub-center arcface: Boosting face recognition by large-scale noisy web faces, in: European Conference on Computer Vision, Springer, 2020, pp. 741–757.
 - [22] T. Dettmers, M. Lewis, S. Shleifer, L. Zettlemoyer, 8-bit optimizers via block-wise quantization, arXiv preprint arXiv:2110.02861 (2021).
 - [23] S. Bai, X. Bai, Sparse contextual activation for efficient visual re-ranking, IEEE Transactions on Image Processing 25 (2016) 1056–1069. doi:10.1109/TIP.2016.2514498.
 - [24] R. J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: Pacific-Asia conference on knowledge discovery and data mining, Springer, 2013, pp. 160–172.
 - [25] L. Otarashvili, Miewid, 2023. URL: <https://github.com/WildMeOrg/wbia-plugin-miew-id>. doi:10.5281/zenodo.13647526.
 - [26] Botswana Predator Conservation Trust, Panthera pardus csv custom export, Retrieved from African Carnivore Wildbook 2022-04-28 via LILA BC, 2022. URL: <https://lila.science/datasets/hyena-id-2022/>.

A. Hardware

To ensure accessibility and reproducibility of our research, all benchmarks and training cycles were executed on a single consumer-grade workstation. This setup demonstrates that high-performance Vision Graph Neural Networks do not strictly require high-performance computing (HPC) clusters.

A.1. Environment Specifications

- **GPU:** NVIDIA RTX 5070 (12 GB VRAM)
- **CPU:** AMD Ryzen 7 7800X3D (8 Cores, 16 Threads)
- **System Memory:** 32 GB G.Skill Flare EXPO X5 (DDR5-6000)
- **Storage:** Samsung 970 EVO Plus NVMe M.2 SSD (PCIe Gen 3)
- **Operating System:** Windows 11 (Model is Linux Compatible)
- **Compute Platform:** NVIDIA CUDA Toolkit 13.0

A.2. Memory Optimization and Quantization

To facilitate the large batch sizes ($N = 512$) required for effective SubCenter ArcFace hard-negative mining, we implemented 8-bit quantized Adam optimization (`bitsandbytes`). This reduced the memory overhead of the optimizer states by approximately 75% without observed degradation in clustering accuracy. Furthermore, all graph representation tensors were cast to bfloat16 Mixed Precision. Unlike standard half-precision (float16), bfloat16 maintains the same dynamic range as float32, effectively preventing gradient underflow during deep message passing in the 3-layer GATv2 backbone.

B. LMDB Caching Logic

The transformation of raw pixels into topological Chimeras involves non-trivial computational latency. To eliminate data-loading bottlenecks, we utilized a Lightning Memory-Mapped Database (LMDB). Prior to training, a configuration-aware *Warmup Phase* verifies the presence of graphs in the cache using a **Deterministic Hash** key based on:

1. SEEDS parameters and input resolution.
2. Feature vocabulary (eg. HOG, LBP, CAE).
3. CAE latent dimensionality and versioning.

If a configuration mismatch is detected, the orchestrator triggers parallel re-computation. This approach allows multiple experimental runs to share pre-computed assets, maximizing GPU utilization.

C. Kaggle Submission Hyperparameters

To ensure full transparency and reproducibility of our AnimalCLEF 2026 Kaggle submission, Table 4 details the exact hyperparameter configuration used for the final training run of the ST-VGANN. This configuration corresponds to the 3-Layer Deep Baseline utilized in our primary ensembling pipeline.

Table 4
Hyperparameter Configuration for the Final Kaggle Submission

Category	Hyperparameter	Value
Graph Construction	Input Image Resolution	512×512
	SEEDS Max Superpixels (N)	240
	SEEDS Block Levels	4
	Edge Routing Strategy	Spatial
	Receptive Field (Hops)	1
Feature Extraction	Node Feature Vocabulary	Color, Pos, HOG, LBP, CAE, Shape
	CAE Latent Dimension	90
	CAE Pre-training Epochs	150
	HOG Angular Bins	9
GNN Architecture	Backbone Type	GATv2
	Message Passing Layers (L)	3
	Jumping Knowledge (JK) Mode	Attention
	Global Graph Pooling	GeM
	Final Embedding Dimension	512
Optimization & Loss	Main Training Epochs	200
	Batch Size	512
	SubCenter ArcFace Margin (m)	12.0
	Orthogonal Loss Weight (γ)	0.05
	Node / Edge Dropout Rate	0.10 / 0.15
	Random Seed	42
	Learning Rate Scheduler	ReduceLROnPlateau
	Scheduler Patience / Factor	10 epochs / 0.5

D. Training Time Table

The following data excludes the time needed for Graph generation and reflects solely the time needed to train the ST-VGANN model, the time for the graph construction is not included in Table 5.

Table 5

Runtime of ablation configurations

Ablation Category	Experiment Variant	Median Epoch (s)	Compute Time
Baselines (Control Group)	Baseline (3L Spatial)	13.50	0h 45m
Graph Construction	Edges Attn	18.50	1h 2m
Graph Construction	Edges Hybrid	36.50	2h 0m
Graph Construction	Edges Hybrid 2L	20.50	1h 5m
GNN Depth & Routing	GCN 1L	5.00	0h 16m
GNN Depth & Routing	GCN 2L	4.00	0h 13m
GNN Depth & Routing	GATv2 3L JK Attn	13.50	0h 45m
GNN Depth & Routing	GATv2 3L JK Mean	10.50	0h 33m
GNN Depth & Routing	GATv2 3L No JK	12.50	0h 40m
GNN Depth & Routing	Shallow Baseline	4.00	0h 13m
Node Feature Vocabulary	Color	4.00	0h 12m
Node Feature Vocabulary	CAE Pos Color	4.50	0h 15m
Node Feature Vocabulary	Texture Pos	4.50	0h 14m
Node Feature Vocabulary	Baseline Plus HOG	4.50	0h 14m
Node Feature Vocabulary	Baseline Plus Shape	4.50	0h 15m
Node Feature Vocabulary	PureTexture	4.50	0h 15m
Node Feature Vocabulary	all features	5.00	0h 16m
Compression	CAE Dim32	5.00	0h 16m
Compression	CAE Dim64	5.00	0h 15m
Compression	CAE Dim128	4.50	0h 15m
Compression	Emb Dim32	4.00	0h 13m
Compression	Emb Dim256	4.50	0h 15m
Compression	Emb Dim1024	5.00	0h 16m
Pooling	Pool Mean	4.50	0h 14m
Pooling	Pool Max	4.50	0h 14m
Orthogonality Constraint	No OrthoLoss	4.50	0h 15m
Superpixel Granularity	Macro 50SP	4.00	0h 12m
Superpixel Granularity	Micro 500SP	10.00	0h 33m
Training Data Scaling	Enriched Scaling	9.50	0h 32m
Single-Species Specialists	Lynx Only	2.50	0h 7m
Single-Species Specialists	Salamander Only	1.00	0h 2m
Single-Species Specialists	Turtle Only	7.50	0h 24m
Regularisation & Dropout	No Dropouts	15.50	0h 54m
Regularisation & Dropout	No ModalityDrop	11.00	0h 36m
Regularisation & Dropout	No GraphDrop	11.00	0h 37m
Background Removal (Cutouts)	Turtle Cutouts	7.50	0h 24m
Background Removal (Cutouts)	Salamander Cutouts	1.00	0h 2m
Background Removal (Cutouts)	All Species Cutouts	12.00	0h 39m