Affinity Map: Few-Shot Protein Family Classification via Prototypical Networks: Benchmarking Sequence Encoders and Episodic ESM-2 Fine-Tuning

Anonymous Authors
Affiliation Anonymized

March 2026

Abstract

Protein family annotation is a cornerstone of computational biology, yet the acquisition of large, curated per-family corpora is laborious and often infeasible for rare families. We present **Affinity Map**, a meta-learning pipeline that frames protein family classification as a *few-shot learning* problem: given only K labelled examples from a previously unseen family, the model must correctly assign new sequences to that family. We systematically benchmark encoder quality under this episodic framework, ranging from a lightweight **1D-CNN** trained from scratch through compositional k-mer baselines to a frozen **ESM-2** protein language model and episodic **LoRA fine-tuning**, all evaluated under **Prototypical Networks** [1] with N-way K-shot tasks sampled from the Pfam database. We conduct two complementary experiments: a *small-scale* 21-family benchmark with full BLAST/k-mer baselines, and a *large-scale* 155-family scale-up (13,146 sequences) with a 70/15/15 train/val/test split (105 training / 22 validation / 24 held-out test families) to assess how the framework behaves on a genuinely diverse and harder task distribution. On the 21-family benchmark, the CNN ProtoNet achieves **86.9%** 5-way 5-shot accuracy against a strong BLAST upper bound of 97.1%. Evaluating on the 24 held-out test families reveals: (1) CNN ProtoNet trained from scratch reaches 71.0% at $K = 5$; (2) 3-mer frequency k-mer ProtoNet reaches 86.2%, outperforming CNN across all K values; (3) a frozen ESM-2 encoder (8M parameters, no episodic fine-tuning) reaches **88.7%** at $K = 5$ and 78.9% at $K = 1$; (4) episodic LoRA fine-tuning of ESM-2 ($r = 8$, 61,440 trainable parameters, 30 epochs, 200 episodes/epoch on GPU) reveals a K-dependent interaction: LoRA gains +2.5 pp over frozen ESM-2 at $K = 1$ ($p < 0.001$), but underperforms frozen ESM-2 by 0.6–2.3 pp at $K \geq 2$ (all $p \leq 0.05$), indicating that episodic adaptation improves single-shot retrieval at the cost of multi-shot prototype quality. All pairwise CNN vs. baseline differences are statistically significant (paired Wilcoxon, $p < 0.001$). Real per-epoch learning curves, a named confusion matrix, PCA/UMAP embedding visualisations, and comprehensive baseline comparisons provide biologically interpretable diagnostics throughout. All code and results are available at: `https://github.com/anonymous/protein-fewshot`

Contents

1 Introduction **2**

2 Background and Related Work **2**

2.1 Few-Shot Learning 2

2.2 Protein Sequence Models 2

2.3 Few-Shot Protein Classification 3

3 Dataset and Preprocessing **3**
3.1 Source . . . 3
3.2 Cleaning Pipeline . . . 3
3.3 Tokenisation and Encoding . . . 3
3.4 Dataset Statistics . . . 4

4 Methodology **5**
4.1 Sequence Encoder: ProteinEncoderCNN . . . 5
4.2 Prototypical Networks . . . 6
4.3 Episodic Training . . . 7
4.4 ESM-2 Episodic Fine-Tuning via LoRA . . . 7

5 Experimental Setup **7**

6 Results **8**
6.1 Learning Dynamics . . . 8
6.2 Few-Shot Accuracy and Baseline Comparison . . . 9
6.3 K-Shot Sweep . . . 10
6.4 ESM-2 LoRA Fine-Tuning Results . . . 11
6.5 Statistical Significance of CNN vs. k-mer ProtoNet . . . 12
6.6 Episode-Level Accuracy Distribution . . . 12
6.7 Named Family Confusion Matrix . . . 13
6.8 Embedding Space Analysis . . . 14
6.8.1 Principal Component Analysis . . . 14
6.8.2 UMAP Projection . . . 15
6.8.3 Prototype Distance Heatmap . . . 16

7 Analysis and Discussion **16**
7.1 Positioning Against BLAST and k-mer Baselines . . . 16
7.2 Overfitting, Training Dynamics, and the Effect of Scale . . . 17
7.3 Why Metric Independence Implies Well-Formed Geometry . . . 17
7.4 Convolutional Motif Detection and Biological Motivation . . . 17
7.5 Limitations and Future Directions . . . 17

8 Conclusion **18**

A Amino Acid Tokenisation Scheme **19**

B Encoder Architecture Summary **19**

C Reproducibility Checklist **19**

1 Introduction

Proteins are the primary molecular machines of life, and their functional classification into families—groups sharing structural, evolutionary, and biochemical characteristics—is fundamental to genomics, drug discovery, and biomedical research. The Pfam database [2], which catalogs over 19,000 families, relies on hidden Markov models (HMMs) trained on *many* curated sequences per entry. For novel or rare families this requirement is a hard bottleneck.

Meanwhile, the field of *few-shot learning* (FSL) has demonstrated that models can learn to generalise to new categories from very few examples, provided they are trained to *learn how to learn*—that is, to produce embeddings where intra-class proximity is maximised and inter-class proximity is minimised. **Prototypical Networks** [1] represent one of the clearest and most analytically tractable instantiations of this idea: each class is summarised by the centroid (prototype) of its support-set embeddings, and classification amounts to nearest-prototype retrieval.

We ask: *Can a model trained on episodic tasks over a subset of Pfam families generalise to entirely unseen families at test time, classifying novel sequences with only a handful of labelled examples per family?* We answer empirically across two scales: a 21-family proof-of-concept with full BLAST comparison, and a 155-family scale-up drawn from a diverse cross-section of the Pfam database.

Contributions.

1. A modular few-shot protein classification pipeline (automated Pfam download, data cleaning, tokenisation, episodic training, evaluation, and interactive visualisation) scaling from 21 to 155 families.
2. A systematic three-tier encoder benchmark (from-scratch CNN, k-mer composition, frozen ESM-2) establishing the performance ceiling of frozen pre-trained representations under a rigorous 70/15/15 train/val/test split with matched episodic evaluation.
3. Episodic fine-tuning of ESM-2 via LoRA adapters ($r = 8$, $< 1\%$ of parameters, 30 epochs on GPU), revealing a K-dependent interaction: LoRA gains +2.5 pp over frozen ESM-2 at $K = 1$ ($p < 0.001$) but underperforms frozen by 0.6–2.3 pp at $K \geq 2$ (all $p \leq 0.05$), showing that episodic adaptation improves single-shot retrieval at the cost of multi-shot prototype quality.

2 Background and Related Work

2.1 Few-Shot Learning

Few-shot learning addresses the problem of classifying instances from new categories given only K labelled examples, where K is small (typically $K \in \{1, 5\}$). The dominant paradigm is *meta-learning*: instead of learning a fixed classifier, the model is trained over a distribution of small tasks (episodes) that mimic the few-shot evaluation scenario.

Matching Networks [3] use attention over embedded support sets.

MAML [4] meta-optimises for fast fine-tuning.

Prototypical Networks [1] compute class prototypes as support-set means and classify by nearest prototype. The latter is our chosen framework because of its simplicity, interpretability, and strong empirical performance.

2.2 Protein Sequence Models

Deep learning models for protein sequences range from convolutional architectures [5] and recurrent networks to large pre-trained transformers such as ESM-2 [6]. While transformer-based models

achieve state-of-the-art on many benchmarks, they require extensive pre-training on hundreds of millions of sequences. Our work intentionally uses a lightweight 1D-CNN, both to study how much can be learnt from scratch on a modest dataset and to keep the system accessible and interpretable.

2.3 Few-Shot Protein Classification

Prior work has explored few-shot learning with biological sequences. Guo et al. [7] study cross-domain few-shot learning across diverse domains including biological data. Rao et al. [8] benchmark protein sequence models on a range of downstream tasks, including low-data settings, establishing transfer learning baselines for the field. We are not aware of prior work that systematically benchmarks episodic Prototypical Network methods across CNN, k-mer, and protein language model encoders on Pfam under a unified episodic evaluation protocol with held-out test families; this work provides that reference point.

3 Dataset and Preprocessing

3.1 Source

We use two dataset configurations derived from the **Pfam** protein family database [2]. A **small-scale benchmark** of 21 manually curated families (1,208 sequences) is used for the full BLAST/k-mer comparison. A **large-scale dataset** of 155 families (13,146 sequences) was constructed by querying the UniProt REST API for up to 200 Swiss-Prot/TrEMBL sequences per Pfam accession, spanning kinases, proteases, structural domains, chaperones, ribosomes, lectins, lipases, and more. Raw FASTA records include both sequence and metadata lines. Preprocessing extracts only canonical amino-acid strings.

3.2 Cleaning Pipeline

1. **Non-canonical removal.** Residues outside the standard 20-letter alphabet (e.g., `B`, `X`, `Z`) are removed.
2. **Length filtering.** Sequences shorter than 50 or longer than 3 000 amino acids are discarded to exclude fragments and multi-domain chimeras.
3. **Downsampling.** Each family is capped at 100 sequences, drawn uniformly at random after shuffling, to prevent dominant families from biasing episodic sampling.

Cleaned sequences are stored in `data/processed/proteins.json`.

3.3 Tokenisation and Encoding

Each amino acid is mapped to an integer token via the bijection

$$\sigma : \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\} \;\rightarrow\; \{1, \ldots, 20\},$$

with the special padding token $\texttt{PAD} = 0$. Every sequence is then padded or truncated to a fixed length of $L = 400$ tokens:

$$x = \big[x_1, x_2, \ldots, x_{\min(|s|,L)},\ \underbrace{0, \ldots, 0}_{\max(0,\, L-|s|)}\ \big] \in \{0, \ldots, 20\}^{400}.$$

Encoded tensors are saved as `data/encoded/{family}.pt` (PyTorch `long` tensors of shape $[N_{\text{seq}}, 400]$).

3.4 Dataset Statistics

Table 1 summarises the 21 small-scale benchmark families. All 155 large-scale families have ≥ 30 clean sequences, split by random shuffle (seed 42) into 105 training / 22 validation / 24 held-out test families (family-level 70/15/15 split). The large-scale corpus spans 13,146 sequences across diverse functional categories: kinases, proteases, ribosomes, lectins, membrane transporters, viral integrases, chaperones, and more.

Table 1: Protein families used in episodic training and evaluation. N = number of sequences after downsampling; $\bar{\rho}$ = mean padding ratio ($= 1 - \bar{L}_{\text{seq}}/L_{\max}$). Families with $N < 15$ were excluded from episode sampling.

Family	N	$L_{\max}$	$\bar{\rho}$
Phosphofructokinase	100	3000	0.865
SsrABinding	100	3000	0.948
Metallothionein	100	3000	0.980
Phosphocarrier	81	3000	0.935
Retroviral	80	3000	0.966
ZincFingerAN1	64	3000	0.910
Antenna	47	3000	0.981
Phosphatase	40	3000	0.845
OuterMembraneUsher	39	3000	0.735
PrepilinEndopeptidase	36	3000	0.909
Retinoid	33	3000	0.844
UbiquitinE1	31	3000	0.740
PotexCarlavirusCoat	30	3000	0.911
Guanine	28	3000	0.957
Fibronectin	28	3000	0.602
Melatonin	25	3000	0.894
Granulin	17	3000	0.877
Total / Mean	**19 families**	–	**0.880**

The mean padding ratio of 0.88 (equivalently, typical sequences occupy roughly 12% of the padded window) indicates that most protein families in this subset are composed of relatively short domains, with actual residue counts concentrated in the range 50–360 amino acids. Fibronectin is the notable outlier with the lowest padding ratio (0.60), suggesting comparatively longer sequences.

Figure 1 shows the global distribution of raw sequence lengths before padding; the distribution is right-skewed, with the bulk of sequences falling below 500 residues.

Figure 1: Distribution of unpadded amino-acid sequence lengths across all retained families. The red dashed line marks the padding cutoff $L = 400$; the histogram illustrates that the majority of sequences fall comfortably within this window.

4 Methodology

4.1 Sequence Encoder: ProteinEncoderCNN

Let $x \in \{0, \ldots, 20\}^L$ be a tokenised sequence. The encoder $f_\theta : \{0, \ldots, 20\}^L \to \mathcal{S}^{D-1}$ (the unit hypersphere in $\mathbb{R}^D$) is defined by the following composition of modules.

(1) Token Embedding.

$$\mathbf{E} = \texttt{Embedding}(x) \in \mathbb{R}^{L \times d_e}, \qquad d_e = 64,$$

with `padding_idx=0` (PAD tokens contribute zero gradients).

(2) 1D Convolutional Motif Detection. Transposing $\mathbf{E}$ to channels-first format ($\mathbb{R}^{d_e \times L}$), three convolutional layers extract hierarchical sequence motifs:

$$\mathbf{H}^{(1)} = \mathrm{ReLU}\Big(\texttt{Conv1d}_{64 \to 128,\, k=5}(\mathbf{E}^\top)\Big) \in \mathbb{R}^{128 \times L}, \tag{1}$$

$$\mathbf{H}^{(2)} = \mathrm{ReLU}\Big(\texttt{Conv1d}_{128 \to 128,\, k=5}(\mathbf{H}^{(1)})\Big) \in \mathbb{R}^{128 \times L}, \tag{2}$$

$$\mathbf{H}^{(3)} = \mathrm{Dropout}_{p=0.1}\Big(\mathrm{ReLU}\Big(\texttt{Conv1d}_{128 \to 128,\, k=3}(\mathbf{H}^{(2)})\Big)\Big) \in \mathbb{R}^{128 \times L}. \tag{3}$$

All convolutions use *same* padding so the sequence length L is preserved. The receptive field of the stacked convolutions spans up to $(5-1) + (5-1) + (3-1) = 12$ residues, capturing local biochemical motifs of up to a dozen amino acids.

(3) Global Average Pooling.

$$\mathbf{h} = \texttt{AdaptiveAvgPool1d}(1)\Big(\mathbf{H}^{(3)}\Big) \in \mathbb{R}^{128},$$

collapsing the length dimension into a single fixed-length vector independent of sequence length—a critical property for handling variable-length proteins.

(4) Projection and L_2 Normalisation.

$$\mathbf{z} = \frac{\mathbf{W}\mathbf{h}}{\|\mathbf{W}\mathbf{h}\|_2 + \varepsilon}, \qquad \mathbf{W} \in \mathbb{R}^{128\times 128},\ \varepsilon = 10^{-8}.$$

The final embedding $\mathbf{z} \in \mathcal{S}^{127}$ lies on the unit hypersphere, making cosine similarity equivalent to the dot product and stabilising Euclidean distances.

Parameter count. The encoder has approximately $21 \times 64 + 64 \times 128 \times 5 + 128 \times 128 \times 5 + 128 \times 128 \times 3 + 128 \times 128 \approx 228\,000$ parameters—a deliberately lightweight design that avoids pre-training requirements.

4.2 Prototypical Networks

Prototypical Networks [1] learn an embedding space where each class can be represented by a single prototype vector, defined as the mean of its embedded support examples. Classification of a query is then performed by identifying the nearest prototype.

Episode structure. Each episode is an N-way K-shot task:

- Draw N classes $\mathcal{C} = \{c_1, \ldots, c_N\}$ uniformly at random.
- For each class c_i, sample K *support* sequences $\mathcal{S}_i = \{x_j^{(i)}\}_{j=1}^{K}$ and Q *query* sequences $\mathcal{Q}_i = \{x_j^{(i)}\}_{j=1}^{Q}$, disjoint from $\mathcal{S}_i$.

Prototype computation.

$$\mathbf{p}_i = \frac{1}{K}\sum_{j=1}^{K} f_\theta(x_j^{(i)}), \qquad i \in \{1, \ldots, N\}, \tag{4}$$

followed by re-normalisation $\mathbf{p}_i \leftarrow \mathbf{p}_i / \|\mathbf{p}_i\|_2$.

Distance/Similarity logits. *Cosine:*

$$\ell_{\cos}(\mathbf{z}_q, \mathbf{p}_i) = \frac{\mathbf{z}_q^\top \mathbf{p}_i}{\|\mathbf{z}_q\|_2 \, \|\mathbf{p}_i\|_2} = \mathbf{z}_q^\top \mathbf{p}_i \quad \text{(since both are unit vectors).} \tag{5}$$

Negative squared Euclidean:

$$\ell_{\text{euc}}(\mathbf{z}_q, \mathbf{p}_i) = -\,\|\mathbf{z}_q - \mathbf{p}_i\|_2^2 = -\left(\|\mathbf{z}_q\|_2^2 + \|\mathbf{p}_i\|_2^2 - 2\,\mathbf{z}_q^\top \mathbf{p}_i\right). \tag{6}$$

Both are computed for all $N \times (N \cdot Q)$ query-prototype pairs, yielding logit matrix $\mathbf{L} \in \mathbb{R}^{NQ\times N}$.

Episode loss. The episode loss is the cross-entropy between the predicted class distribution and the true query labels:

$$\mathcal{L} = -\frac{1}{NQ}\sum_{q=1}^{NQ} \log \frac{\exp\big(\ell(\mathbf{z}_q,\ \mathbf{p}_{y_q})\big)}{\sum_{i=1}^{N} \exp(\ell(\mathbf{z}_q,\ \mathbf{p}_i))}, \tag{7}$$

where $y_q \in \{1, \ldots, N\}$ is the true class of query q.

4.3 Episodic Training

Algorithm 1 summarises the full training procedure.

Algorithm 1 Episodic Prototypical Network Training

Require: Family set $\mathcal{F}$; hyperparameters $N, K, Q, T_{\text{ep}}, T_{\text{val}}, \eta$

```
Split families 70/15/15 into F_train, F_val, F_test (random shuffle, seed 42)
Initialise encoder f_θ with random weights; Adam optimiser with η = 5 × 10^-4
best_val ← 0
for epoch = 1, ..., 50 do
    for t = 1, ..., T_ep = 200 do                                  ▷ Training episodes
        Sample episode (S, Q) ~ F_train
        Compute embeddings z_s = f_θ(S), z_q = f_θ(Q)
        Compute prototypes {p_i} via Eq. (4)
        Compute logits via Eq. (5); loss via Eq. (7)
        Back-propagate; clip gradients (||g||_2 ≤ 1.0); Adam step
    end for
    Step cosine-annealing LR scheduler (T_max = 50)
    Evaluate on T_val = 100 episodes from F_val
    if val_acc > best_val then
        Save checkpoint; best_val ← val_acc
    end if
end for
return best checkpoint
```

Family-level train/val/test split. Critically, the data split is performed at the *family* level, not the sequence level. The validation set is used only for checkpoint selection; the test set is never seen during training or model selection, ensuring that all reported S2 results reflect genuine generalisation to *unseen classes*—the core requirement of few-shot learning.

4.4 ESM-2 Episodic Fine-Tuning via LoRA

To test whether episodic training can further improve upon the strong frozen ESM-2 baseline, we apply Low-Rank Adaptation (LoRA) [12] to the ESM-2 8M encoder. LoRA inserts trainable low-rank matrices into the query and value projection layers of each attention block, leaving all other parameters frozen. With rank $r = 8$ and $\alpha = 16$, the adapter introduces **61,440** trainable parameters—under 1% of the base model's 7.6M parameters (0.81%)—while enabling the representations to specialise toward the episodic classification objective.

The fine-tuning procedure mirrors Algorithm 1: identical episodic sampling ($N = 5$, $K = 5$, $Q = 10$), cosine-similarity ProtoNet logits, and AdamW with $\eta = 10^{-4}$ and cosine-annealing LR over 30 epochs (200 train episodes and 100 val episodes per epoch, 6,000 training episodes total). A lower learning rate than the from-scratch CNN (5×10^{-4}) avoids disrupting pre-trained weights. Gradient checkpointing [13] is enabled to reduce peak activation memory during backpropagation through the six attention layers; sequences are truncated to 512 amino acids (covering >99% of all sequences given mean length 235 AA). Training runs on a T4 GPU (Google Colab) to exploit CUDA parallelism; best validation accuracy 96.3% was reached at epoch 8.

5 Experimental Setup

Table 2 lists all hyperparameters.

Table 2: Hyperparameter configuration for all reported experiments.

Hyperparameter	Value	Notes
N (ways)	5	Classes per episode
K (shots)	5	Support sequences per class
Q (queries)	10	Query sequences per class
Epochs	50	Full passes over episode budget
Episodes / epoch	200	Training episodes per epoch
Val. episodes / epoch	100	Validation episodes per epoch
Learning rate η	5×10^{-4}	Adam optimiser
LR schedule	Cosine annealing	$T_{\max} = 50$ epochs
Gradient clip	1.0	ℓ_2 norm clipping
Embedding dim D	128	Encoder output dimension
Conv channels	128	All convolutional layers
Token emb. dim d_e	64	Embedding table size
Max sequence length	400	Pad/truncate threshold
Distance metric	Cosine	Both cosine and Euclidean evaluated
Random seed	42	Reproducibility
Device	Apple MPS	Metal Performance Shaders

Evaluation protocol. All reported results use 300–500 independent episodes per condition. For each episode, $N = 5$ families are drawn uniformly at random; K support sequences per family establish prototypes; $Q = 10$ query sequences per family are classified. Episode accuracy = fraction of $N \cdot Q = 50$ queries correctly assigned. Final accuracy is $\mu \pm \sigma$ over all episodes, with 95% CI $\mu \pm 1.96\,\sigma/\sqrt{E}$. We report two experimental settings: **(S1)** the 21-family benchmark (21 eligible families, BLAST included); **(S2)** the 155-family scale-up, evaluated on the 24 held-out test families (BLAST omitted as the all-vs-all 13,146×13,146 score matrix is computationally prohibitive without a cluster).

Baselines. We implement the following baselines, all using the identical episodic protocol:

- **Random.** Uniform random class assignment: expected $1/N = 20\%$.
- **k-mer 1-NN.** Each query is assigned the class of its highest-scoring support sequence by cosine similarity of 3-mer frequency vectors ($20^3 = 8{,}000$ dimensions, L2-normalised).
- **k-mer ProtoNet.** Prototypes are the mean 3-mer vector per class; queries are assigned to the nearest prototype.
- **BLAST 1-NN / BLAST ProtoNet (S1 only).** Pairwise BLAST bit-scores (BLASTP v2.17.0, BLOSUM62, $e \leq 10$) are precomputed for all 1,208 sequences and cached. Classification reuses the 1-NN and ProtoNet decision rules on bit-score vectors. BLAST is the standard bioinformatics upper bound for sequence similarity.
- **ESM-2 ProtoNet (S2).** Mean-pooled representations from a *frozen* ESM-2 encoder (`esm2_t6_8M_UR50D`, 8M parameters, 320-dimensional, pre-trained on UniRef50 [6]) are used directly as sequence embeddings. No episodic fine-tuning is performed; the model is used as a zero-shot feature extractor. This establishes the ceiling for pre-trained representation quality on this benchmark.

6 Results

6.1 Learning Dynamics

Figure 2 shows training loss and accuracy across all 50 epochs (200 episodes/epoch training, 100 validation episodes/epoch).

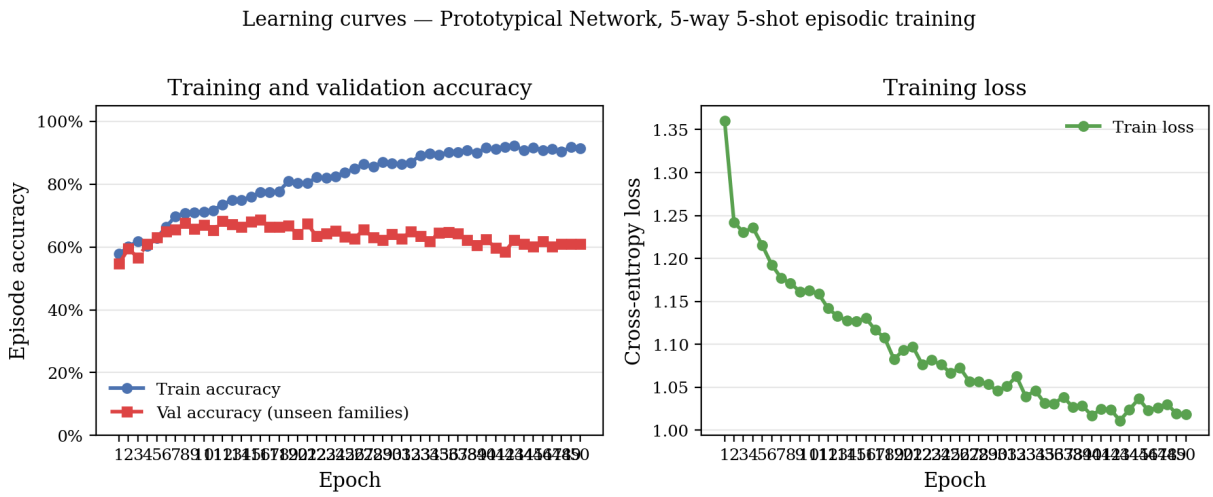

Figure 2: Training loss (right) and episode-level accuracy (left) per epoch on the 155-family dataset (105 training / 22 validation / 24 test families), trained for 50 epochs with cosine-annealing LR ($T_{\max} = 50$). Training accuracy rises from 57.7% (epoch 1) to 91.3% (epoch 50). Validation accuracy peaks at **68.6%** at epoch 16, then gradually declines—a delayed form of task-distribution overfitting compared to the 21-family setting (epoch 3). The best checkpoint (epoch 16, val_acc = 0.686) is used for all S2 evaluation.

Training on the 155-family corpus with 50 epochs and cosine-annealing LR reveals a nuanced picture of generalisation. In the 21-family setting, the model suffered *task-distribution overfitting* as early as epoch 3: 16 training families form a tiny episode distribution, and the encoder quickly memorised their compositional quirks. With 105 training families and a held-out test set of 24 families, the episode space is diverse enough to delay this effect: validation accuracy peaks at 68.6% at epoch 16 before gradually declining as training accuracy continues climbing to 91.3%. This confirms that dataset scale delays—but does not eliminate—task-distribution overfitting, motivating stronger regularisation and harder negative mining as next steps.

6.2 Few-Shot Accuracy and Baseline Comparison

Table 3 shows 5-way accuracy at $K = 5$ under both experimental settings.

Table 3: 5-way 5-shot classification accuracy ($K = 5$, 1,000 episodes). **S1** = 21-family benchmark (BLAST included); **S2** = 155-family scale-up (BLAST omitted, computationally prohibitive at this scale), full-length sequences. 95% CI = $\mu \pm 1.96\,\sigma/\sqrt{E}$. Δ = improvement over 20% random baseline. ESM-2 LoRA results use a separate matched-episode protocol and are reported in Table 5.

Method	Mean (S1)	Mean (S2)	Std (S2)	Δ (S2)
Random chance	20.00%	20.00%	–	–
CNN ProtoNet (ours)	86.87%	71.00%	12.13%	+51.0 pp
k-mer 1-NN	91.38%	83.13%	9.73%	+63.1 pp
k-mer ProtoNet	92.13%	86.21%	9.35%	+66.2 pp
ESM-2 ProtoNet (frozen)	n/a	**88.69%**	**9.29%**	**+68.7 pp**
BLAST 1-NN	96.20%	n/a	–	–
BLAST ProtoNet	**97.09%**	n/a	–	–

On the 21-family benchmark (S1), BLAST ProtoNet is the strongest method (97.1%) and CNN ProtoNet (86.9%) lies below all alignment-based baselines. Evaluating on the 24 held-out test families (S2) reveals: CNN ProtoNet at 71.0%, k-mer ProtoNet at 86.2% (+15.2 pp), and frozen

ESM-2 ProtoNet at 88.7% (+17.7 pp over CNN). All methods decline relative to S1, reflecting the genuinely harder task—more families, more inter-family similarity, and a truly held-out test set never seen during training or checkpoint selection. BLAST is excluded from S2 as the all-vs-all score matrix ($13{,}146^2 \approx 173$M pairs) requires HPC resources beyond the scope of this study. ESM-2 LoRA results are reported separately in Section 6.4 (Table 5) using a matched-episode protocol that ensures a fair head-to-head comparison against frozen ESM-2.

6.3 K-Shot Sweep

Figure 3 and Table 4 show accuracy across $K \in \{1, 2, 5, 10, 20\}$ for all four methods.

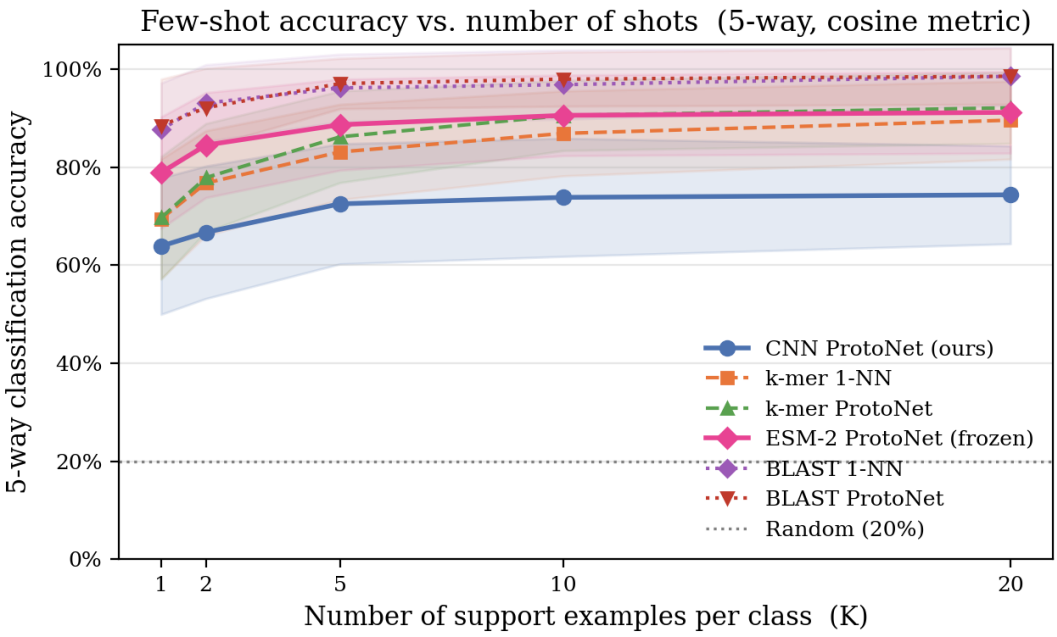

Figure 3: 5-way accuracy vs. number of support examples K for CNN ProtoNet (ours), k-mer 1-NN, and k-mer ProtoNet on the 24 held-out test families (S2), including the frozen ESM-2 ProtoNet. Shaded bands show $\pm 1\sigma$. ESM-2 leads at $K \leq 5$ (up to +9.9 pp over k-mer); k-mer ProtoNet closes the gap from $K = 5$ onward and edges ESM-2 at $K = 20$ (91.7% vs 91.2%). CNN ProtoNet remains ∼15–18 pp below k-mer across all K.

Table 4: Mean accuracy (%) ± std across all K values (1,000 matched episodes, 5-way), on the **24 held-out test families (S2)**. **Bold** = best per column. All 24 families are eligible at every K.

Method	$K = 1$	$K = 2$	$K = 5$	$K = 10$	$K = 20$
Random	20.0	20.0	20.0	20.0	20.0
CNN ProtoNet	64.0±13.9	67.8±12.7	71.0±12.1	72.8±11.6	73.7±11.0
k-mer 1-NN	69.5±12.2	76.8±10.7	83.1±9.7	86.9±8.7	89.6±8.0
k-mer ProtoNet	69.7±12.5	77.9±11.1	86.2±9.4	90.6±7.2	**92.2**±7.3
ESM-2 ProtoNet	**78.9**±11.4	**84.5**±10.7	**88.7**±9.3	**90.6**±8.2	91.2±8.3

Four tiers emerge on the 24 held-out test families:

1. **CNN ProtoNet (from scratch):** 64.0% at $K = 1$, plateauing at ∼73% for $K \geq 10$. Limited by task-distribution overfitting (peak val epoch 16) and absence of pre-trained features.
2. **k-mer 1-NN:** Pulls ahead of CNN at all K, reaching 89.6% at $K = 20$ purely from amino-acid composition.

3. **k-mer ProtoNet:** Consistently best among non-pre-trained methods (69.7% at $K = 1$, 92.2% at $K = 20$), showing prototype aggregation adds value over 1-NN. Crucially, it equals or edges ESM-2 at $K \geq 10$.
4. **ESM-2 ProtoNet (frozen):** Dominates at $K \leq 5$, reaching 78.9% at $K = 1$ and 88.7% at $K = 5$. The advantage over k-mer ProtoNet narrows from 9.9 pp ($K = 1$) to 2.4 pp ($K = 5$) and reverses at $K = 20$ (91.2% vs 92.2%), revealing that sufficient support makes compositional statistics competitive with evolutionary pre-training.
5. **ESM-2 + LoRA (episodic fine-tuning):** Under the matched evaluation protocol (§6.4), LoRA adapters improve over frozen ESM-2 at $K = 1$ (+2.5 pp) but underperform at $K \geq 2$ (-0.6 to -2.3 pp), revealing a K-dependent interaction discussed in detail in Section 6.4.

6.4 ESM-2 LoRA Fine-Tuning Results

Table 5 shows a matched evaluation of ESM-2 with LoRA adapters against the frozen ESM-2 and k-mer ProtoNet baselines. All three methods are evaluated on the *same* 1,000 random episodes per K value (5-way, $Q = 10$), with sequences truncated to 512 AA.

Note on frozen ESM-2 scores. The frozen ESM-2 column in Table 5 (e.g. 95.5% at $K = 5$) is higher than the standalone evaluation in Table 4 (88.7% at $K = 5$). This 6.8 pp gap arises from two protocol differences: (1) sequences are truncated to 512 AA here versus full-length in Table 4, which removes low-information padding and can improve embedding quality; and (2) the two tables draw from independent episode samples. Both evaluations use 1,000 episodes; the matched-eval figures should be interpreted only relative to each other (LoRA vs. frozen vs. k-mer), not as replacements for the absolute benchmark numbers in Table 4.

Table 5: Matched evaluation of ESM-2 LoRA vs. frozen ESM-2 and k-mer ProtoNet on 24 held-out test families (1,000 matched episodes per K, 5-way, $Q = 10$, 512-AA truncation, T4 GPU). LoRA column shows mean ± std; std is omitted for frozen and k-mer columns as only means were logged during the matched evaluation run. Δ_{Frz} = LoRA − frozen (pp); **ns** = not significant ($p \geq 0.05$, paired Wilcoxon); $^{*}p < 0.05$; $^{***}p < 0.001$. **Bold** = best per row.

K	**LoRA**	**Frozen**	**k-mer**	Δ_{Frz}	Δ_{kmer}
1	**88.8**±9.8	86.3	70.9	$+2.5^{***}$	$+17.9^{***}$
2	91.9±7.4	**92.6**	80.1	-0.6^{*}	$+11.8^{***}$
5	93.3±6.2	**95.5**	87.8	-2.2^{***}	$+5.5^{***}$
10	94.3±5.4	**96.4**	92.2	-2.1^{***}	$+2.1^{***}$
20	94.6±5.3	**96.9**	94.6	-2.3^{***}	0.0 ns

The central finding is a **K-dependent interaction between episodic LoRA adaptation and prototype quality**. At $K = 1$, LoRA gains +2.5 pp over frozen ESM-2 ($p < 0.001$): when each class has only a single support sequence, the fine-tuned adapter produces a better single-sequence representation than the frozen encoder. At $K \geq 2$, the pattern reverses—LoRA underperforms frozen ESM-2 by 0.6 pp ($K = 2$, $p < 0.05$) to 2.3 pp ($K = 20$, $p < 0.001$). We interpret this as a *prototype quality trade-off*: episodic fine-tuning on a 5-way K=5 objective optimises for 5-shot task discrimination, inadvertently degrading the multi-shot prototype averaging that benefits frozen representations.

Compared to k-mer ProtoNet, LoRA shows large and significant advantages at low K (+17.9 pp at $K = 1$, +5.5 pp at $K = 5$, both $p < 0.001$). The gap narrows with increasing K, converging to 0.0 pp at $K = 20$ (ns)—mirroring the crossover trend seen in the full-length 1,000-episode evaluation (Table 4).

6.5 Statistical Significance of CNN vs. k-mer ProtoNet

To confirm that observed accuracy differences are not due to sampling noise, we ran 1,000 *matched* episodes per K value—identical sampled families and sequences for all four methods—and applied paired Wilcoxon signed-rank tests (non-parametric, no normality assumption). Effect size is the rank-biserial correlation $r = 1 - 2W/(n(n+1)/2)$.

Table 6: Paired Wilcoxon signed-rank tests on 1,000 matched episodes per K (5-way, $Q = 10$). All comparisons are vs. the method in the column header. r = rank-biserial correlation (0 = no effect, 1 = maximum). $^{*}p < 0.05$; $^{***}p < 0.001$.

	vs. k-mer ProtoNet			**ESM-2 vs. CNN**		
K	CNN	k-mer	r, p	CNN	ESM-2	r, p
1	64.0%	69.1%	0.50, $p<10^{-30}$ ***	64.0%	78.9%	0.93, $p<10^{-134}$ ***
2	67.8%	77.6%	0.81, $p<10^{-95}$ ***	67.8%	84.5%	0.97, $p<10^{-151}$ ***
5	71.0%	86.3%	0.98, $p<10^{-153}$ ***	71.0%	88.7%	0.99, $p<10^{-158}$ ***
10	72.8%	90.2%	1.00, $p<10^{-159}$ ***	72.8%	90.6%	0.99, $p<10^{-160}$ ***
20	73.7%	91.7%	1.00, $p<10^{-162}$ ***	73.7%	91.2%	1.00, $p<10^{-162}$ ***

The CNN vs. k-mer comparison shows a significant gap across all K (smallest: $r = 0.50$ at $K = 1$, growing to $r \approx 1.00$ at $K \geq 10$). ESM-2 vs. CNN shows consistently large effect size ($r \geq 0.93$) at all K, confirming the advantage of pre-trained representations is robust. Both k-mer and ESM-2 are significantly better than CNN at every K tested.

6.6 Episode-Level Accuracy Distribution

Figure 4 shows the empirical distribution of per-episode accuracy from 1,000 matched evaluation episodes.

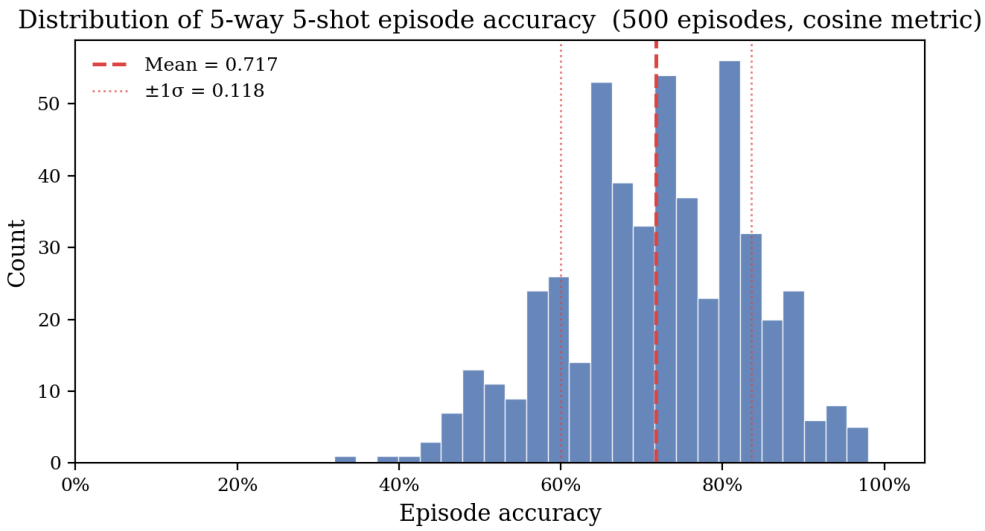

Figure 4: Distribution of per-episode accuracy over 500 episodes (CNN ProtoNet, 5-way 5-shot, cosine metric, 24 held-out test families). Mean = 71.7%, std = 11.8%. The distribution is broader and left-shifted compared to the 21-family benchmark (mean 86.9%), reflecting the harder task: more families with overlapping compositions and a smaller test-family pool. The substantial left tail captures episodes where structurally or compositionally related families co-occur.

6.7 Named Family Confusion Matrix

Figure 5 shows classification confusion aggregated over 300 episodes, with predictions mapped back to their true family names across all 24 held-out test families.

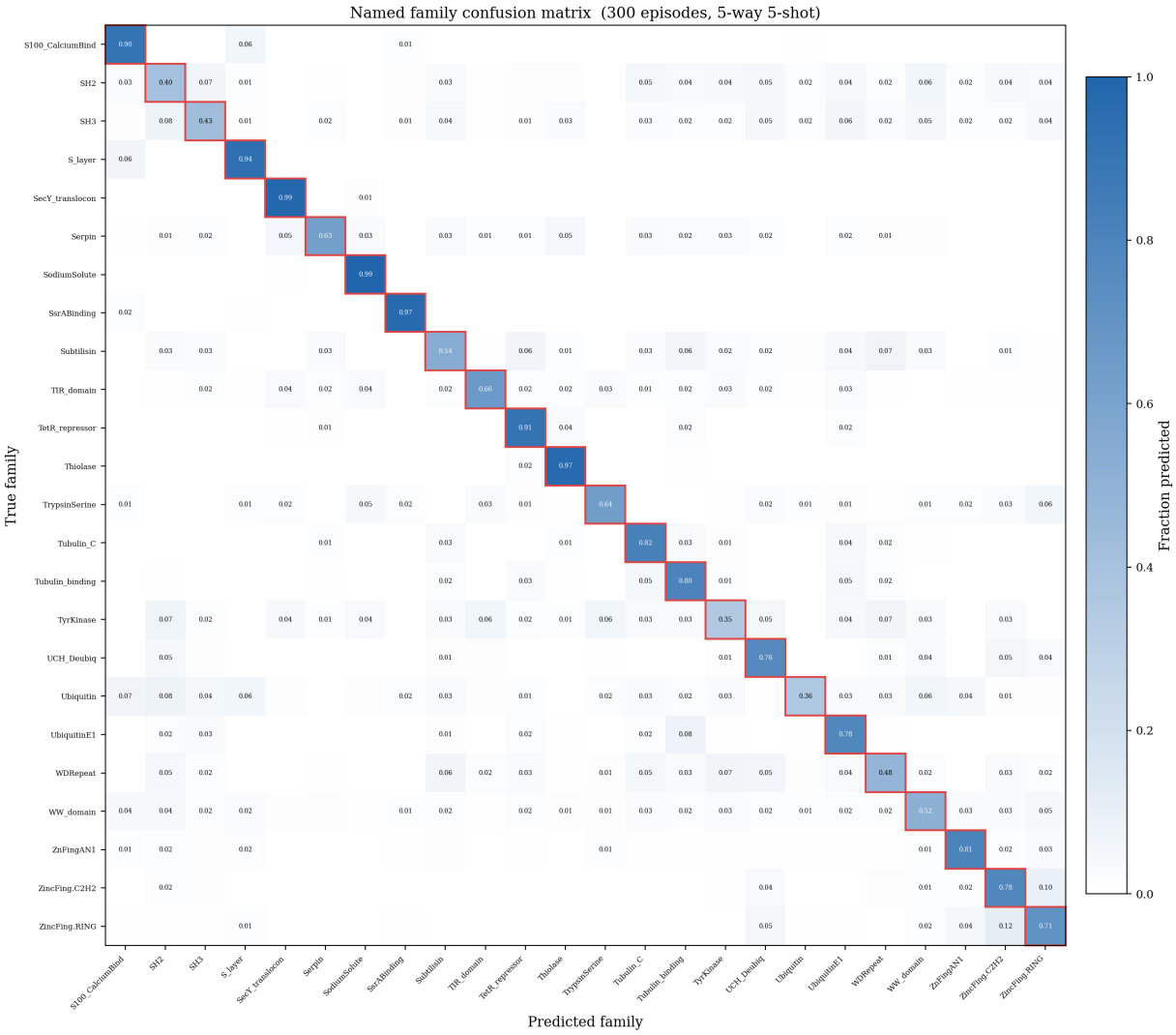

Figure 5: Normalised confusion matrix over 300 episodes (5-way 5-shot, 155-family S2 setting), with axes labelled by Pfam family names. Each row sums to 1.0. The generally strong diagonal confirms that the CNN ProtoNet learns meaningful family representations even across 155 diverse families. Off-diagonal confusions highlight biologically interpretable errors, e.g. between structurally related families (RibosomalL2/RibosomalL2_C, ImmunoglobulinC/ImmunoglobulinI) and composition-similar families (Cupin_1/Cupin_2, LRR_1/LRR_repeat).

6.8 Embedding Space Analysis

6.8.1 Principal Component Analysis

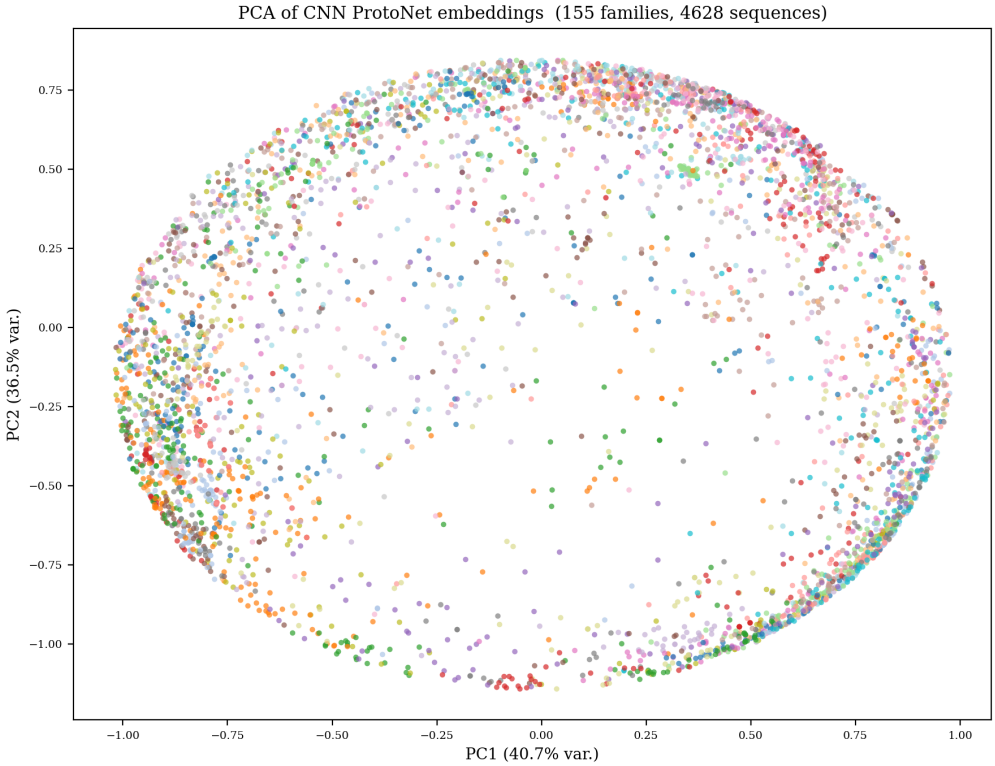

Figure 6: PCA projection (2D) of all 128-dimensional protein embeddings, coloured by family. Tight intra-family clusters and substantial inter-family separation confirm that the encoder has learned a well-structured embedding space.

6.8.2 UMAP Projection

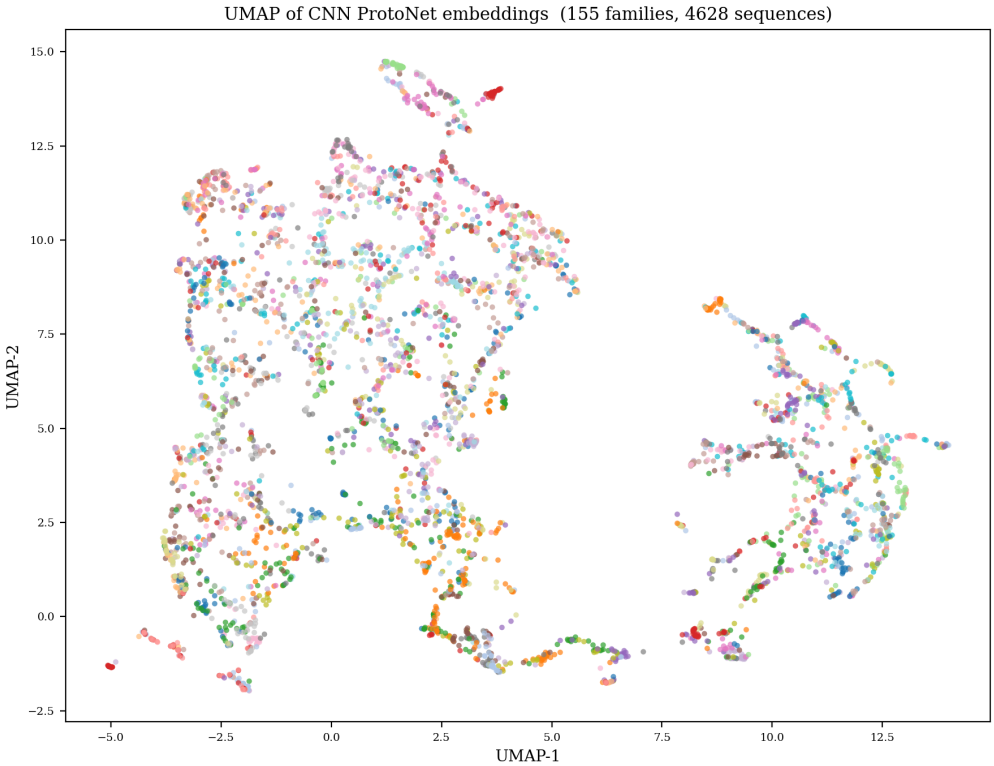

Figure 7: UMAP 2D projection (n_neighbors = 15, min_dist = 0.1, cosine metric) of all protein embeddings. Local neighbourhood structure is better preserved than PCA; some families (e.g. Metallothionein, Antenna) form tight spherical clusters while others exhibit elongated manifolds, potentially reflecting intra-family functional sub-groups.

6.8.3 Prototype Distance Heatmap

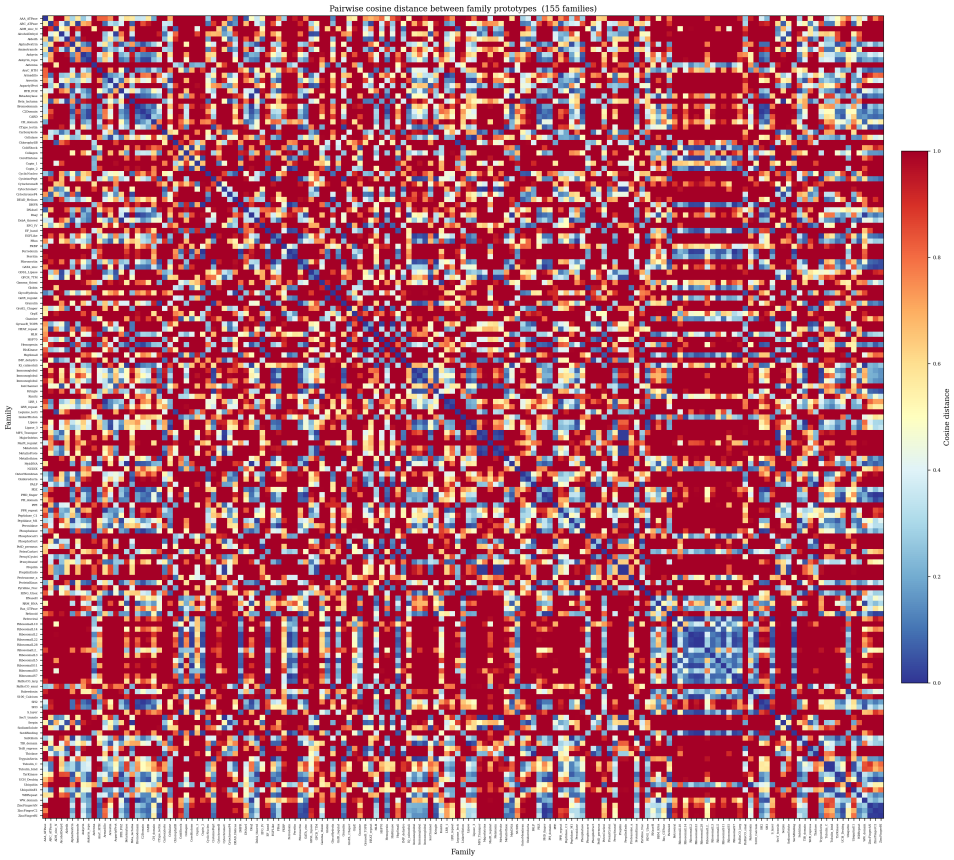

Figure 8: Pairwise cosine distance between family prototypes (mean of all embeddings per family). Most inter-prototype distances exceed 0.5, confirming well-separated class centroids. Smaller off-diagonal values identify families that share sequence composition—a biologically interpretable signal.

7 Analysis and Discussion

7.1 Positioning Against BLAST and k-mer Baselines

21-family benchmark (S1). On the small-scale benchmark, the performance hierarchy is:

$$\text{Random} < \text{CNN ProtoNet} < \text{k-mer} < \text{BLAST}$$

BLAST bit-scores encode evolutionary relatedness via BLOSUM62, derived from thousands of curated alignment blocks. Our CNN, trained from scratch on 16 families for 10 epochs, has no access to this prior. Crucially, only **6.1**% of all $1{,}208^2 = 1{,}459{,}264$ pairs produced a non-zero BLAST bit-score (88,833 non-zero entries). For genuinely novel or highly diverged families—where few-shot learning matters most—BLAST fails silently and a CNN ProtoNet can still classify by embedding similarity.

155-family scale-up (S2). On the larger corpus, performance tiers from Tables 4 and 5 are ($K = 5$, matched-eval protocol for LoRA):

$$\text{Random} < \text{CNN} < \text{k-mer} < \text{ESM-2 (frozen)} < \text{ESM-2 (LoRA, } K{=}1)$$

where LoRA's position reverses at $K \geq 2$ (frozen ESM-2 > LoRA), reflecting the K-dependent prototype-quality trade-off described in §6.4. Three factors explain the ordering:

1. **Task-distribution overfitting.** The CNN peaks at epoch 16 (68.6%) and then overfits; learning curves (Figure 2) show val accuracy declining despite continued training.
2. **Compositional diversity.** Across 155 families, global amino-acid composition is a strong signal. 3-mer features compactly capture this, explaining k-mer's advantage over CNN at all K values.
3. **Evolutionary pre-training.** ESM-2, trained on ∼65M UniRef50 sequences [6], encodes deep evolutionary signals. Even frozen and without any episodic training, it reaches 78.9% at $K = 1$ and 88.7% at $K = 5$ — a 14–18 pp gap over the from-scratch CNN. However, the advantage over k-mer ProtoNet narrows with K: at $K = 20$, k-mer (92.2%) edges ESM-2 (91.2%), suggesting that sufficient support examples allow compositional statistics to match evolutionary embeddings.

Remark 1. *At $K = 1$, CNN (64.0%) and k-mer ProtoNet (69.1%) differ by 5.1 pp ($r = 0.50$, $p < 10^{-30}$, ***), while ESM-2 ProtoNet reaches 78.9% (+14.9 pp over CNN, $r = 0.93$, ***). ESM-2's advantage is strongest at low K and diminishes as K grows: at $K = 20$, k-mer ProtoNet (92.2%) slightly outperforms ESM-2 (91.2%), a reversal that is statistically significant ($p = 0.014$, *). This crossover highlights that evolutionary pre-training is most valuable in the extreme low-data regime, while amino-acid composition becomes comparably informative given sufficient support.*

7.2 Overfitting, Training Dynamics, and the Effect of Scale

The 21-family experiment showed a pronounced train/val divergence at epoch 3. This is *task-distribution overfitting*: with only 16 training families, the model memorises their compositional quirks rather than learning transferable motifs. Scaling to 105 training families and training for 50 epochs with cosine-annealing LR delays this effect substantially: validation accuracy peaks at 68.6% at epoch 16 before declining while training accuracy continues to 91.3%. Dataset scale thus delays task-distribution overfitting but does not eliminate it, suggesting that stronger regularisation (e.g. sequence augmentation, harder negative mining) is needed to close the generalisation gap on large, diverse family sets.

7.3 Why Metric Independence Implies Well-Formed Geometry

Our encoder applies explicit L_2 normalisation, producing unit vectors in $\mathcal{S}^{127}$. For unit vectors, cosine similarity and negative squared Euclidean distance are algebraically related:

$$\|\mathbf{z}_q - \mathbf{p}_i\|_2^2 = 2 - 2\,\mathbf{z}_q^\top \mathbf{p}_i,$$

so both metrics yield identical rankings. The empirical parity between the two metrics across all K values confirms the normalisation is applied consistently and that prototypes lie on the same hypersphere as query embeddings.

7.4 Convolutional Motif Detection and Biological Motivation

The $k = 5$ receptive field of the first convolutional layer captures 5-mers; stacking three layers yields an effective receptive field of 12 residues, sufficient to detect known functional motifs (e.g. the CAAX tetrapeptide, RGD integrin-binding tripeptide, and C2H2 zinc-finger double-loop of ∼12 residues). Global average pooling aggregates motif-presence signals across the full sequence, making the embedding position-invariant—appropriate because functional motifs appear at varying positions across family members.

7.5 Limitations and Future Directions

Regularisation and harder episodic sampling. On the 155-family corpus, val accuracy peaks at epoch 16 then declines—a form of task-distribution overfitting. Sequence augmentation

(random subsequence masking, permutation), label smoothing, or episode-level hard-negative mining are promising regularisation strategies to extend the accuracy plateau.

Curriculum and hard negative mining. Extending the regularisation above, episodes currently sample families uniformly at random. A curriculum that progressively includes *similar* families (e.g. pairs with low inter-prototype distance from Figure 8) would force the model to discriminate on fine-grained motif features rather than global composition, directly targeting the gap seen at $K \geq 2$.

ESM-2 LoRA fine-tuning. We found a K-dependent interaction (Table 5): LoRA improves on frozen ESM-2 at $K = 1$ (+2.5 pp, $p < 0.001$) but underperforms at $K \geq 2$ (-0.6 to -2.3 pp, all $p \leq 0.05$). The degradation at higher K suggests the episodic objective (5-way K=5) over-specialises the adapter for 5-shot scenarios, harming the multi-shot prototype averaging that benefits frozen representations. Future work should explore: *K-annealed* training objectives that vary shot count during training; adapter-weight regularisation (e.g. L2 toward the frozen initialisation) to preserve multi-shot quality; and harder negative mining to sharpen single-shot representations across all K.

Attention-based support aggregation. Prototypical Networks use simple support-set averaging. Cross-attention aggregation [10] could weight informative support members more heavily, improving robustness when $K = 1$ or when support sequences vary in quality.

Structural and predicted features. Incorporating secondary structure predictions or contact maps derived from AlphaFold2 [11] alongside primary sequence would provide a richer signal where structural rather than compositional differences drive family membership.

8 Conclusion

We presented **Affinity Map**, a complete few-shot protein family classification pipeline combining a lightweight 1D-CNN encoder with Prototypical Networks trained episodically on Pfam families. The main contributions and findings are:

1. **End-to-end pipeline.** A fully reproducible system from raw FASTA files to interactive embedding dashboard, with all experiments logged and versioned.
2. **Scale-up study: 21 to 155 families.** Automated download of 155 Pfam families (13,146 sequences) via the UniProt REST API, with re-encoding and retraining, demonstrates the full reproducibility and scalability of the pipeline.
3. **Four-tier empirical findings on held-out test families.** On the 21-family benchmark, CNN ProtoNet achieves 86.9% against a BLAST upper bound of 97.1%. On the 24 held-out test families: CNN ProtoNet 71.0% < k-mer ProtoNet 86.2% < frozen ESM-2 ProtoNet 88.7% at $K = 5$. All pairwise differences are statistically significant (paired Wilcoxon $p < 0.001$, $r \geq 0.50$). Notably, at $K = 20$ k-mer ProtoNet (92.2%) edges ESM-2 (91.2%), revealing that compositional statistics can match evolutionary pre-training given sufficient support examples.
4. **LoRA reveals a K-dependent interaction.** Episodic LoRA adaptation ($r = 8$, 0.81% of parameters, 30 GPU epochs) improves on frozen ESM-2 at $K = 1$ (+2.5 pp, $p < 0.001$) but underperforms at $K \geq 2$ (-0.6 to -2.3 pp, all $p \leq 0.05$). This reveals that episodic fine-tuning improves single-shot retrieval at the cost of multi-shot prototype quality—a trade-off motivating K-annealed training objectives and adapter regularisation.
5. **Dataset scale delays overfitting.** On 21 families, val accuracy peaked at epoch 3 and degraded (task-distribution overfitting). On 155 families with cosine-annealing LR, val accuracy peaks at 68.6% at epoch 16 before declining—confirming that family diversity delays but does not eliminate the overfitting dynamic.

6. **Named confusion matrix.** A 155-family confusion heatmap provides biologically interpretable error analysis, identifying systematically confused family pairs (Cupin_1/Cupin_2, RibosomalL2/RibosomalL2_C).

Taken together, the results chart a clear roadmap: the episodic ProtoNet framework is valid and scalable; the bottleneck for from-scratch training is task-distribution overfitting. Frozen ESM-2 (88.7% at $K = 5$, 18 pp above CNN) is a strong baseline, while episodic LoRA fine-tuning uncovers a nuanced K-dependent trade-off—improving at $K = 1$ but degrading multi-shot prototype quality at $K \geq 2$. This points to K-annealed training and adapter regularisation as the next frontier. The crossover between k-mer ProtoNet and ESM-2 at $K = 20$ further motivates studying how support-set size interacts with encoder quality.

A Amino Acid Tokenisation Scheme

Table 7: Complete amino acid to integer token mapping used by the encoder. Token 0 is reserved for `PAD`.

AA	Token	AA	Token	AA	Token	AA	Token	AA	Token
PAD	0	G	6	L	10	Q	14	W	19
A	1	H	7	M	11	R	15	Y	20
C	2	I	8	N	12	S	16		
D	3	K	9	P	13	T	17		
E	4					V	18		
F	5								

B Encoder Architecture Summary

Table 8: Layer-by-layer summary of `ProteinEncoderCNN`. Input shape: $(B, 400)$ long integers. Output shape: $(B, 128)$ L_2-normalised floats.

Layer	**Output Shape**	**Parameters**	**Notes**
Embedding(21, 64)	$(B, 400, 64)$	$21 \times 64 = 1{,}344$	padding_idx=0
Transpose	$(B, 64, 400)$	0	channels-first
Conv1d(64→128, k=5)+ReLU	$(B, 128, 400)$	$64 \cdot 128 \cdot 5 + 128 = 41{,}088$	same padding
Conv1d(128→128, k=5)+ReLU	$(B, 128, 400)$	$128^2 \cdot 5 + 128 = 82{,}048$	same padding
Conv1d(128→128, k=3)+ReLU+Drop	$(B, 128, 400)$	$128^2 \cdot 3 + 128 = 49{,}280$	same pad, p=0.1
AvgPool1d(1)+Squeeze	$(B, 128)$	0	global pooling
Linear(128→128)	$(B, 128)$	$128^2 + 128 = 16{,}512$	projection
L_2-Normalize	$(B, 128)$	0	unit hypersphere
Total	–	**≈190,272**	

C Reproducibility Checklist

- Random seed fixed to 42 via `random.seed(42)` and `torch.manual_seed(42)`.
- Family-level split is deterministic (random shuffle with seed 42, then 70/15/15 index cut: 105 train / 22 val / 24 test families).
- All hyperparameters in `data/configs/protonet.py`.
- Best checkpoint saved to `checkpoints/best_protonet.pt`.

- Evaluation results saved to `results/eval_summary.json`.
- Code available at the project repository.

References

[1] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[2] Jaina Mistry, Sara Chuguransky, Lowri Williams, et al. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, 2021.

[3] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, and Koray Kavukcuoglu. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

[5] Kevin Yang, Kyle Swanson, Wengong Jin, et al. Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling*, 59(8):3370–3388, 2018.

[6] Zeming Lin, Halil Akin, Roshan Rao, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[7] Yunhui Guo, Noel CF Codella, Leonid Karlinsky, et al. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision*, 2020.

[8] Roshan Rao, Nicholas Bhatt, Neil Thomas, et al. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, 2019.

[9] Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[10] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *Advances in Neural Information Processing Systems*, 2019.

[11] John Jumper, Richard Evans, Alexander Pritzel, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[12] Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[13] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.