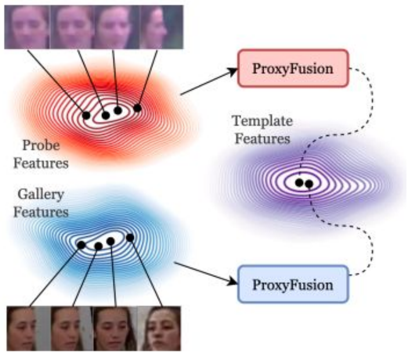
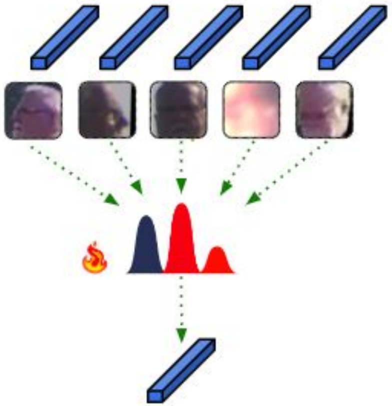
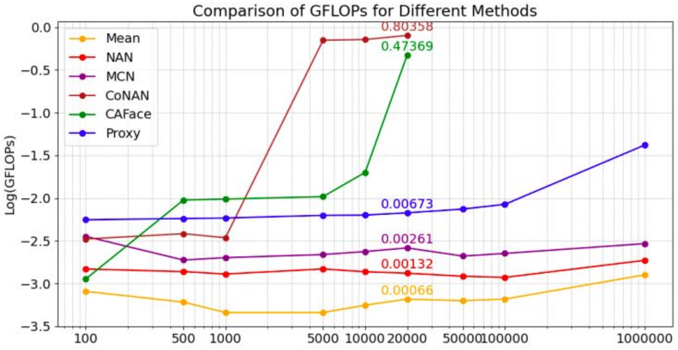
ProxyFusion

Face Feature Aggregation Through Sparse Experts

NeurIPS 2024



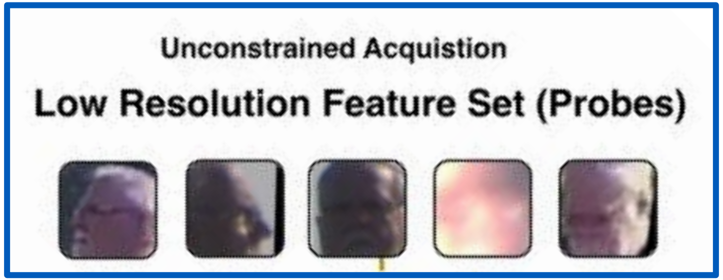


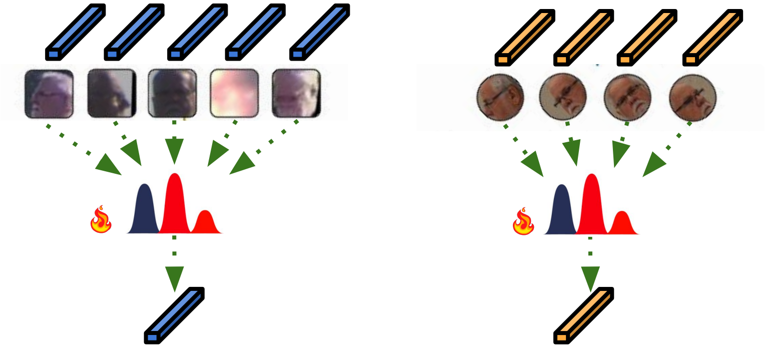


Bhavin Jawade, Alexander Stone, Deen Dayal Mohan,
Xiao Wang, Srirangaraj Setlur, Venu Govindaraju
Center for Unified Biometrics and Sensors, University at Buffalo

Face Feature Aggregation

Given a bunch of face features, how do you decide effective weightages (informativeness) to fuse these features for robust long range face recognition?

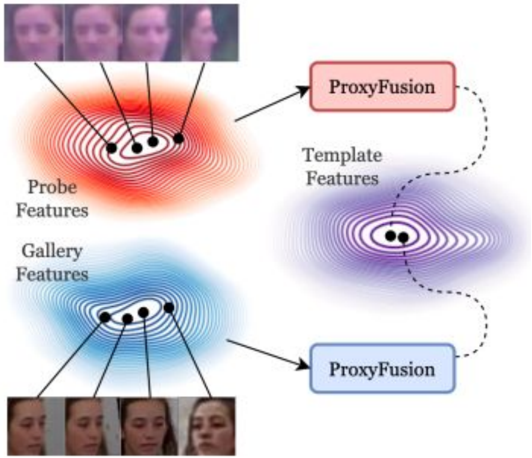




University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS

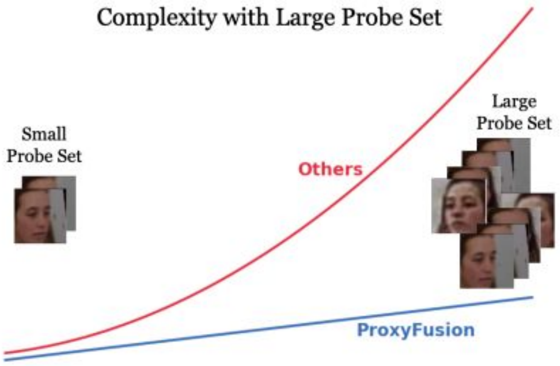
ProxyFusion



Cross-Distribution Matching

	Linear Complexity	Compatible with Legacy Templates	Cross-Domain Matching
Recurrent Methods [e.g. MARN]	✓	✗	✗
Intra-set Attention [e.g. CoNAN, RSA]	✗	✓	✓
Style Based Methods [e.g. CAFace, PFE]	✗	✗	✓
Metadata Approaches [e.g. TADPool, MFAN]	✗	✗	✗
ProxyFusion (Ours)	✓	✓	✓

Compatibility To Legacy Templates

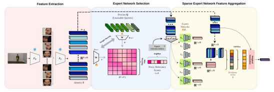


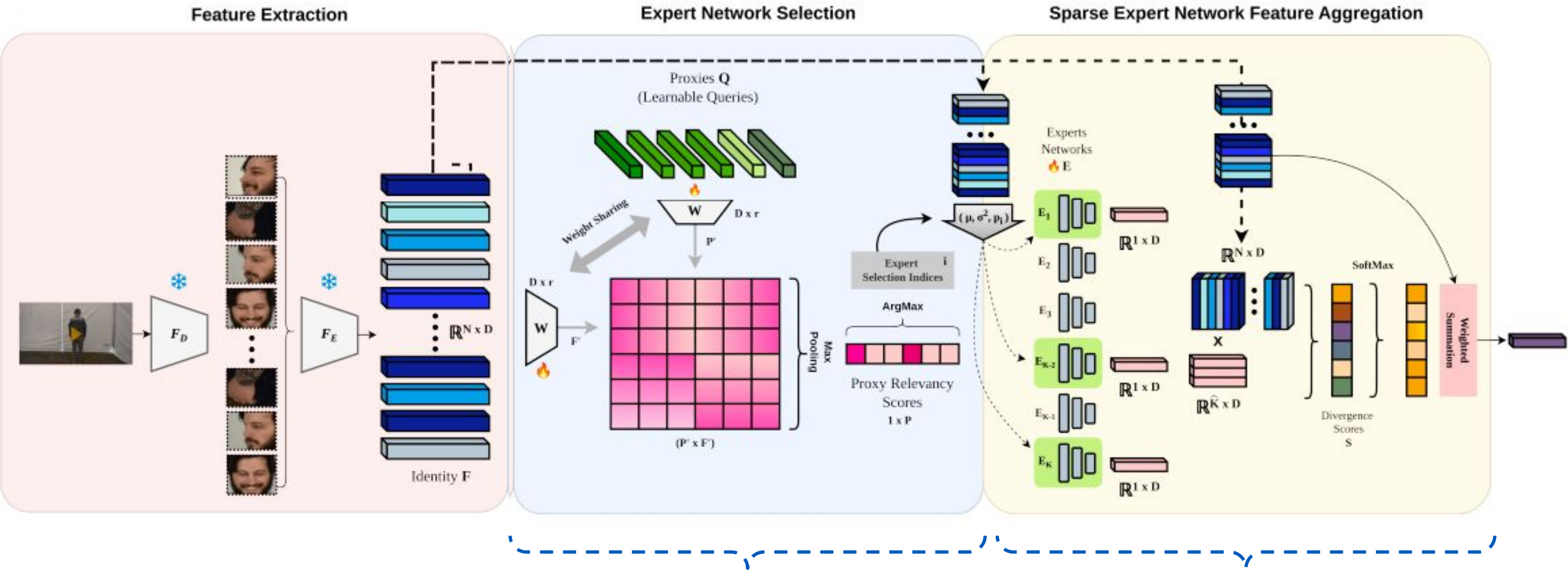
Time Complexity

University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS

Proposed Architecture





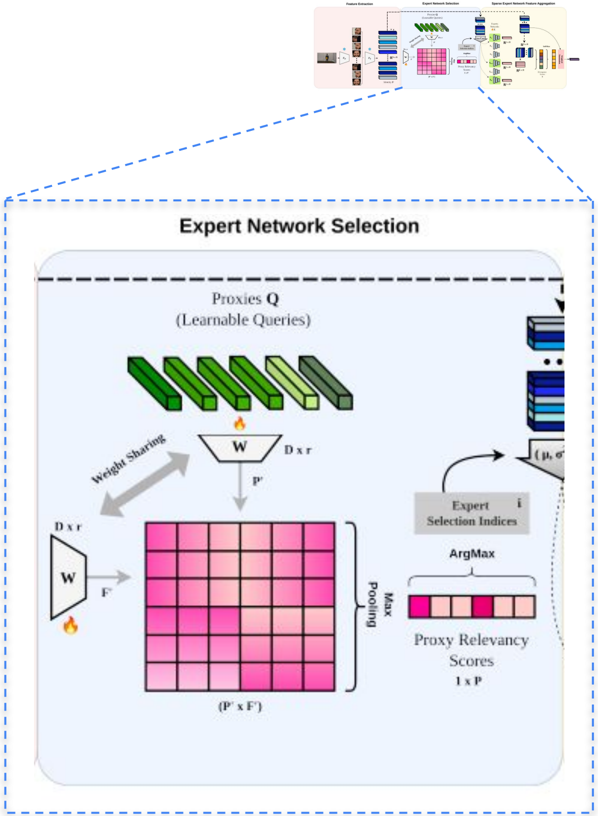
Stage 1

Stage 2

University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS

Expert Network Selection



- **Learnable proxies** for latent facial attributes
- Proxy **relevancy scores** to sparsely activate expert networks

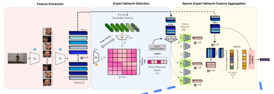
$$r_j = \sum_{i=1}^{N} (p'_j \cdot f'_i)$$

- Compute **Top-K** Indices using proxy relevance scores
- **Activate** the relevant experts

University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS

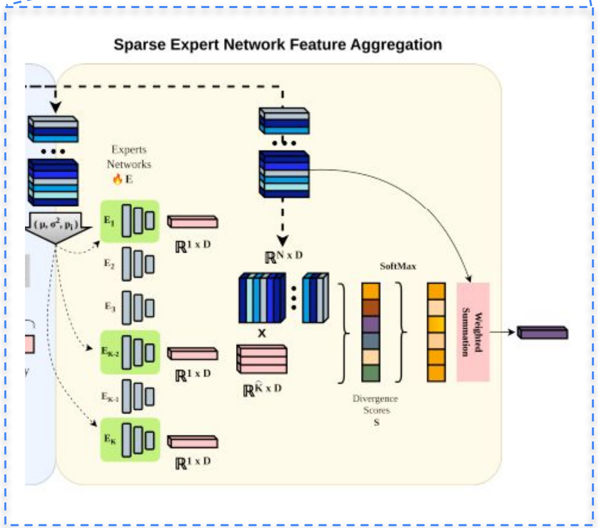
Sparse Expert Network Feature Aggregation



- **Aggregation** through selected experts
- Experts conditioned on of **mean, variance, and proxies**

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}_i, \quad \boldsymbol{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{f}_i - \boldsymbol{\mu})^2$$

$$\mathbf{x}_j = \left[\boldsymbol{\mu} \bigoplus \boldsymbol{\sigma}^2 \bigoplus \mathbf{p}_j\right]$$

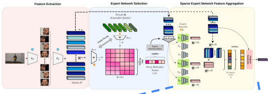


- The outputs of the expert networks - **set-centers**
- For each feature f in the feature set, compute the **divergence score** relative to each set center:

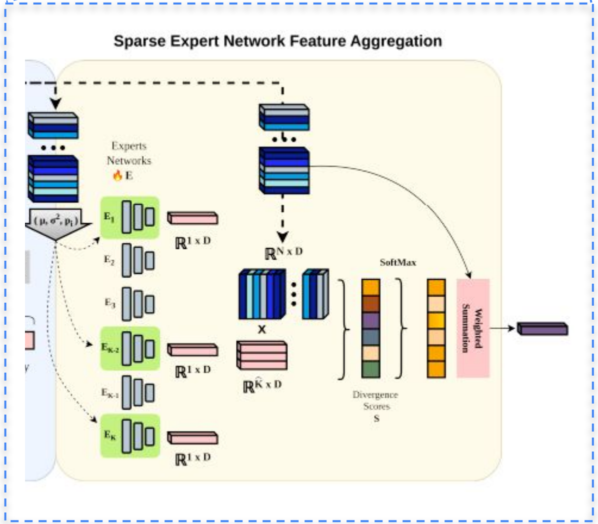
$$a_{ij} = \frac{\exp(\mathbf{c}_j \cdot \mathbf{f}_i)}{\sum_{k=1}^{N} \exp(\mathbf{c}_j \cdot \mathbf{f}_k)}$$

University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS



Sparse Expert Network Feature Aggregation



- Experts conditioned on mean, variance, and proxies

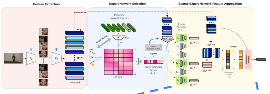
$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}_i, \quad \boldsymbol{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{f}_i - \boldsymbol{\mu})^2 \qquad \mathbf{x}_j = \left[\boldsymbol{\mu} \bigoplus \boldsymbol{\sigma}^2 \bigoplus \mathbf{p}_j\right]$$

- The outputs of the expert networks - set-centers
- For each feature f in the feature set, compute the divergence score relative to each set center:

$$a_{ij} = \frac{\exp(\mathbf{c}_j \cdot \mathbf{f}_i)}{\sum_{k=1}^{N}\exp(\mathbf{c}_j \cdot \mathbf{f}_k)}$$

- Using the divergence scores, compute the **weighted sum of the feature vectors for each expert:**

$$\mathbf{s}_j = \sum_{i=1}^{N} a_{ij}\mathbf{f}_i$$



Sparse Expert Network Feature Aggregation

- Experts conditioned on mean, variance, and proxies

$$\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}_i, \quad \boldsymbol{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{f}_i - \boldsymbol{\mu})^2 \qquad \mathbf{x}_j = \left[\boldsymbol{\mu} \bigoplus \boldsymbol{\sigma}^2 \bigoplus \mathbf{p}_j\right]$$

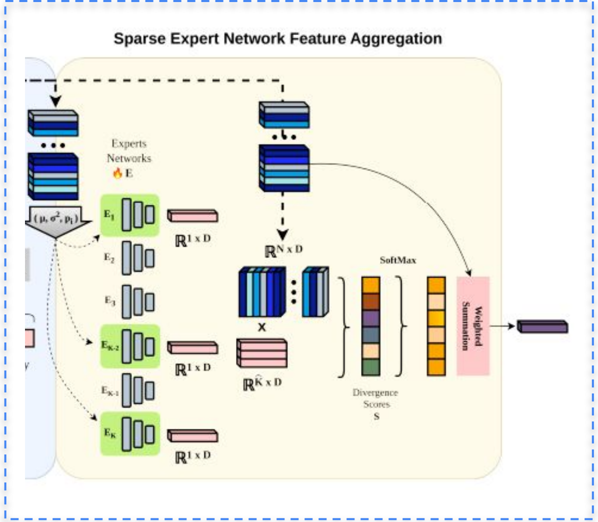
- The outputs of the expert networks - set-centers
- For each feature *f* in the feature set, compute the divergence score relative to each set center:

$$a_{ij} = \frac{\exp(\mathbf{c}_j \cdot \mathbf{f}_i)}{\sum_{k=1}^{N} \exp(\mathbf{c}_j \cdot \mathbf{f}_k)}$$

- Using the divergence scores, compute the weighted sum of the feature vectors for each expert:

$$\mathbf{s}_j = \sum_{i=1}^{N} a_{ij} \mathbf{f}_i$$

$$\mathbf{t} = \left[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{\widehat{K}}\right]$$



University at Buffalo
The State University of New York

NEURAL INFORMATION PROCESSING SYSTEMS

Optimization

- **Supervised contrastive loss** for identity matching

$$\mathcal{L}_{\text{id}} = \sum_{i \in \mathcal{B}} \frac{-1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \ln \frac{\exp(\mathbf{t}_i \cdot \mathbf{t}_p^\top / \tau)}{\sum_{j \in \mathcal{A}(i)} \exp(\mathbf{t}_i \cdot \mathbf{t}_j / \tau)}$$

- **Proxy loss** for decorrelation and diversity
    - K uniformly spaced equidistant vectors on the unit hypersphere

$$\mathbf{v}_i = \left(\mathbf{e}_i - \frac{1}{d}\sum_{j=1}^{d} \mathbf{e}_j\right)\sqrt{\frac{d}{d-1}},$$

$$L_{\text{Proxy}} = \frac{1}{K}\sum_{i=1}^{K}\left[\ln\left(1+\exp\left(-\alpha(s_{ii}-\lambda)\right)\right) + \frac{1}{|K-1|}\sum_{\substack{k \in K \\ k \neq i}} \ln\left(1+\exp\left(\beta(s_{ik}-\lambda)\right)\right)\right]$$

$$\Rightarrow$$

$$\mathcal{L} = \mathcal{L}_{\text{ID}} + \gamma \cdot \mathcal{L}_{\text{Proxy}}$$

Datasets

Training:

1. BRIAR Research Set 3 (BRS 3)
2. WebFace 4M

Evaluation:

1. BTS 3.1
2. DroneSURF

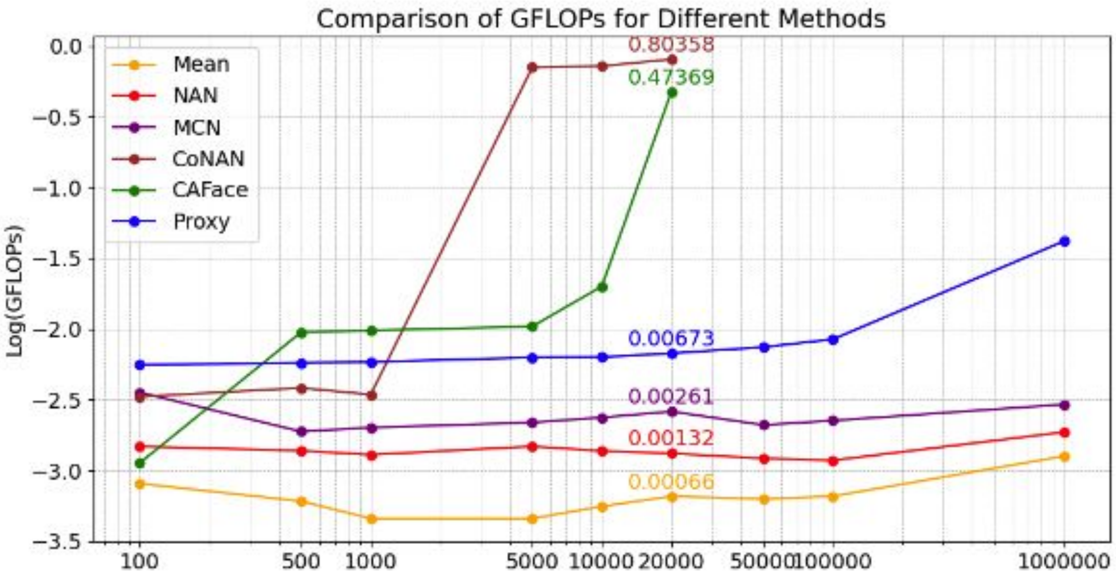
Dataset	Subjects/Identities	Media
BRS 3	170	49,429 clips/images: - 20,780 field clips
WebFace 4M	10,000	813,482 images
BTS 3.1	260 (treatment) 256 (control)	- 5,822 treatment probe videos - 1,914 control probe videos
DroneSURF	58 (34 training/validation, 24 test)	200 videos, 411,000 frames, 786,000+ face annotations

Comparison To SoTA

Verification Performance (TAR (%) @FAR=%) for face included treatment and control protocols of the BTS 3.1 dataset.

			Face Included Treatment				Face Included Control			
	Feature	Dataset	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	10^{-4}
GAP [11]	Adaface [8]	Briar	76.6	58.4	43.3	32.1	98.5	94.6	88.9	81.2
NAN [20]	Adaface [8]	Briar	78.5	61.2	46.8	33.4	98.5	95.3	89.3	84.8
MCN [19]	Adaface [8]	Briar	79.4	62.9	47.3	35.9	98.5	95.9	90.7	85.7
CoNAN [5]	Adaface [8]	Briar	81.3	64.3	49.6	36.8	98.6	96.2	91.8	86.1
ProxyFusion	Adaface [8]	Briar	**83.7**	**68.9**	**53.9**	**40.1**	**98.6**	**96.8**	**92.7**	**88.3**

Inference Time



Time complexity comparison of ProxyFusion approach against SoTA. On the Y-axis we plot the Log of GFLOPs with base 10, and X axis is the number of features in the feature set N

Visualizing Learned Weights

More Information

Codebase is available at: https://github.com/bhavinjawade/ProxyFusion

Project Page: https://bhavinjawade.github.io/proxyfusion_ub/

Reach out to:

Bhavin Jawade
PhD Candidate @ University at Buffalo
bhavinja@buffalo.edu

University at Buffalo
The State University of New York

NEURAL INFORMATION
PROCESSING SYSTEMS