This Probably Looks *Exactly* Like That

Zachariah Carmichael* · Timothy Redgrave* · Daniel Gonzalez Cedre* · Walter Scheirer

University of Notre Dame

PROTOTYPE NETWORKS

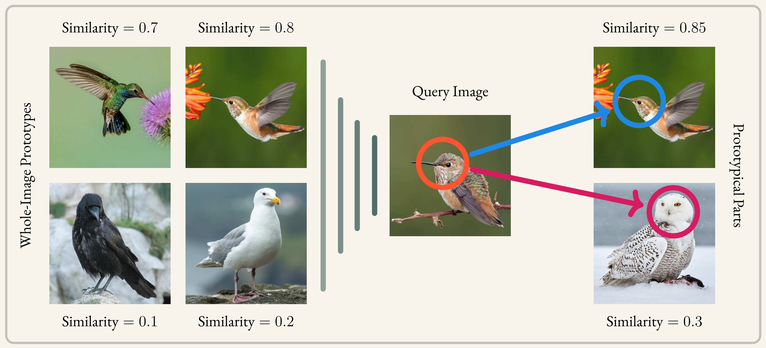

Figure 1. Examples of whole-image prototypes and prototypical parts explanations for classifying a bird according to species. (LEFT) Whole-image prototypes are traditionally images taken from the training data taken to be representatives for a particular class. A classification decision can be then explained by finding a prototype for that class with a high similarity to the query image and contrasting that with low similarity scores for prototypes from other classes. (RIGHT) Prototypical parts build on this idea by using *parts* of images from the training data and finding their similarity with *parts* of the query image when forming explanations.

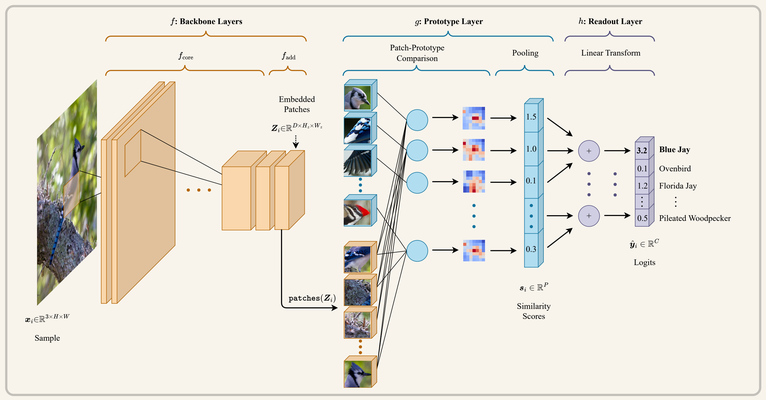

Figure 2. The PROTOPNET architecture enables *"This looks like that"* explanations. Predictions are a linear function of the similarity between learnable prototypes (corresponding to object parts) and embedded image parts. However, it is limited by its pixel space mapping (prototype projection) and prototype expressivity (latent point-based representation).

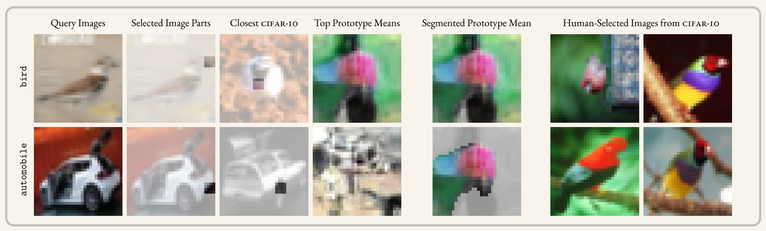

Figure 3. (a) *"This looks like that"*-style explanations of `bird` (TOP ROW) and `automobile` (BOTTOM ROW) image classification decisions. The third column shows the most-likely dataset image part for each prototypical distribution. Rather than using training samples as prototypes, *ProtoFlow* learns prototype distributions directly over the latent space, leading to the *"bird/car-adjacent"* images in the fourth column. (b) The mean point image of the `bird` prototype with a bird-like figure segmented from the background. (c) Human-picked images from CIFAR-10 that qualitatively match this prototype image.

METHODS

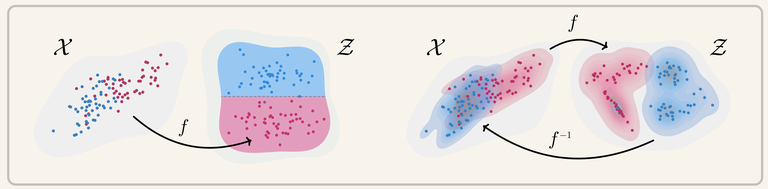

Figure 4. (LEFT) Existing prototypical networks rely on learning prototypes as latent points, making it hard to visualize latent points. (RIGHT) *ProtoFlow* learns prototypes as latent *probability distributions* with *exact inverses*.

$$p_{\mathcal{X}}(y \mid \boldsymbol{x}) = \frac{\sum_{k=1}^{K} \sigma(\pi_{y,k})\, \mathcal{N}(f(\boldsymbol{x})\,;\, \boldsymbol{\mu}_{y,k}, \boldsymbol{\Sigma}_{y,k})}{\sum_{c=1}^{C}\sum_{k=1}^{K} \sigma(\pi_{c,k})\, \mathcal{N}(f(\boldsymbol{x})\,;\, \boldsymbol{\mu}_{c,k}, \boldsymbol{\Sigma}_{c,k})} \tag{1}$$

$$\mathcal{L}_{\text{DIV}}(\mathcal{G}) = \frac{-2}{CK(K-1)} \sum_{c=1}^{C} \sum_{i=1}^{K-1} \sum_{j=i}^{K} \widetilde{\mathcal{H}}^2(\mathcal{G}_{c,i}, \mathcal{G}_{c,j}) \tag{2}$$

$$\mathcal{L}_{\text{CR}}(\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}) = -\log p_{\mathcal{X}}(\dot{\boldsymbol{x}} \mid \ddot{y}) = -\log p_{\mathcal{Z}}\big(f(\dot{\boldsymbol{x}}) \mid y = \ddot{y}\big) - \log \left| \det \left(\frac{\partial f}{\partial \dot{\boldsymbol{x}}} \right) \right| \tag{3}$$

$$\mathcal{L} = \mathcal{L}_{\text{CE}}\big(p_{\mathcal{X}}(y \mid \boldsymbol{x}), y\big) + \lambda_{\text{CR}} \mathcal{L}_{\text{CR}}(\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}) + \lambda_{\text{DIV}} \mathcal{L}_{\text{DIV}}(\mathcal{G}) \tag{4}$$

Equation Set 1. (1) The conditional class likelihood $p_{\mathcal{X}}(y \mid \boldsymbol{x})$ models the latent prototypes as Gaussian mixture models. (2) The diversity loss $\mathcal{L}_{\text{DIV}}$ penalizes prototypes with the squared Hellinger distance $\widetilde{\mathcal{H}}^2$. (3) Consistency regularization $\mathcal{L}_{\text{CR}}$ penalizes variance under perturbations and augmentations. (4) The full loss $\mathcal{L}$ is an affine combination of $\mathcal{L}_{\text{DIV}}$ and $\mathcal{L}_{\text{CR}}$ with cross-entropy.

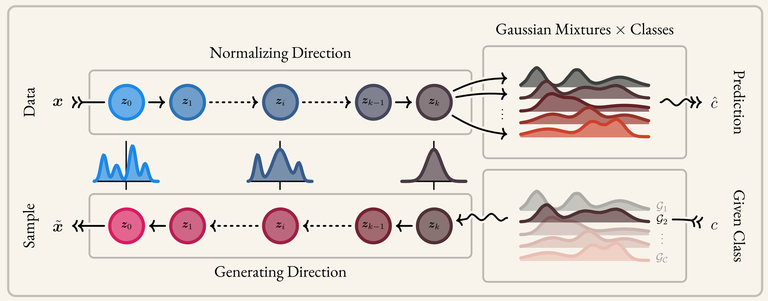

Figure 5. *ProtoFlow* is a composition of normalizing flows that learns latent Gaussian mixtures as prototypes. In the *normalizing* $\mathcal{X} \xrightarrow{f} \mathcal{Z}$ direction, the composition $f = f_k \circ \cdots \circ f_1$ *pulls back* the structured latent density $p_{\mathcal{Z}}$ to the complex data density $p_{\mathcal{X}}$. In the *generating* $\mathcal{X} \xleftarrow{f^{-1}} \mathcal{Z}$ direction, latent samples $\tilde{\boldsymbol{z}} \sim p_{\mathcal{Z}}$ get *pushed forward* along the inverse mapping f^{-1}.

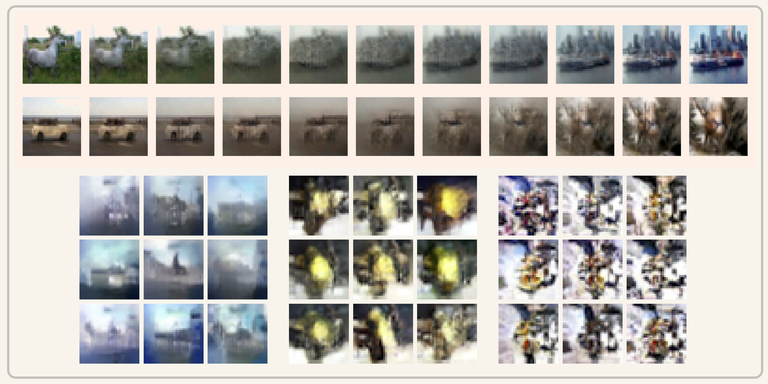

Figure 6. (TOP) Inter-class latent space interpolations using *ProtoFlow* trained on CIFAR-10. (BOTTOM) Prototype means (center of grids) and samples (surrounding) from prototypes learned on CIFAR-10 with truncation set to 1 and with consistency loss.

RESULTS

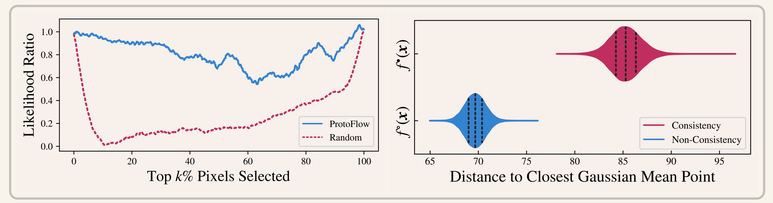

Figure 7. (LEFT) *Are we learning relevant prototypes?* Given a query image $\boldsymbol{x}$, we compute pixel-wise heatmaps of this image—for each prototype, for each class—based on the latent likelihood of patches of the query image conditioned on the prototypes (see paper for details). We quantify the relevance of our prototypes by comparing likelihoods from the top k% pixels from these heatmaps to those from random heatmaps. (RIGHT) *What is the impact of consistency loss?* Embedded points lie closer to distribution means *without* the consistency loss than *with* it, possibly explaining their lack of *"noise"* and more *"realistic"* appearance.

DATASET	RESOLUTION	MODEL	PROTO-BASED	FLOW-BASED	ACC ↑	BPD ↓	ECE ↓	MCE ↓
MNIST	28×28	*ProtoFlow*	✓	✓	99.36	0.535	0.006	0.587
		FlowGMM	✗	✓	99.63	—	0.004‡	—
		Fetaya et al.	✗	✓	99.30	1.00	—	—
		SCNF-GLOW	✗	✓	88.44	1.15	—	—
		SCNF-GMM	✗	✓	83.10	1.14	—	—
CIFAR-10	32×32	*ProtoFlow*	✓	✓	91.54	3.95	0.083	0.494
		IB-INN ($\gamma \to \infty$)	✗	✓	91.28	17.3	0.81	13.9
		IB-INN ($\gamma = 1$)	✗	✓	89.73	5.25	0.54	3.25
		FlowGMM	✗	✓	88.44	—	0.038‡	—
		Fetaya et al.	✗	✓	84.00	3.53	—	—
		ProtoPNet	✓	✗	84.9	—	—	—
		ProtoFlow	✓	✓	69.80	5.03	0.292	0.637
CIFAR-100	32×32	IB-INN ($\gamma \to \infty$)	✗	✓	66.22	18.4	0.62	16.8
		IB-INN ($\gamma = 1$)	✗	✓	57.43	4.93	0.58	7.04

Table 1. A selection of *ProtoFlow* results on joint generative & predictive modeling across different image classification datasets, achieving state-of-the-art accuracy while retaining highly competitive density estimation & calibration scores. See paper for more.

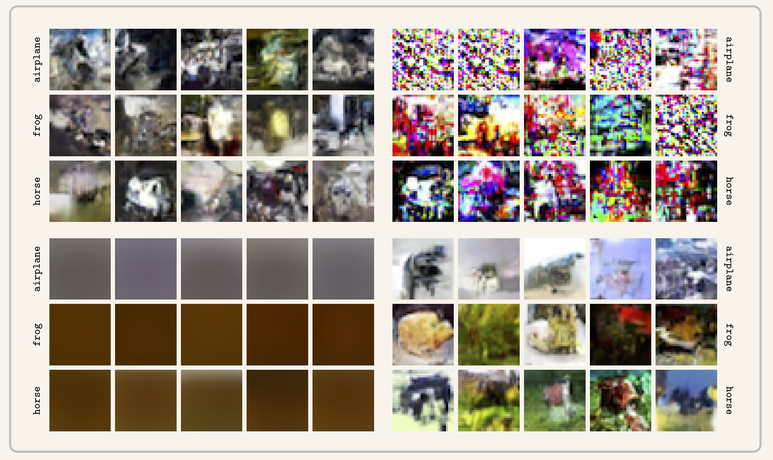

Figure 8. Mean points learned with consistency regularization (a) are more interpretable than the uninformative means (c) learned without it. However, samples with the consistency loss (b) are poor compared to those from obtained without it (d).

State-of-the-art performance on joint predictive and generative tasks.
Measurably improved prototype interpretability.
Improved uncertainty estimates & understanding of latent space.

*Equal contribution.

Timothy Redgrave's travel was supported by the U.S. Department of Defense (Contract No. W52P1J-20-9-3009). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Department of Defense or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation here on.

‡These results reported with temperature-based scaling of GMM variances.