Proto4DME: Interpretable Cell Counting via Additive Prototype Density Decomposition and Optimal-Transport Coverage

Abdurahman Ali Mohammed ABDU@IASTATE.EDU
Iowa State University, USA

Wallapak Tavanapong TAVANAPO@IASTATE.EDU
Iowa State University, USA

Abstract

Cell counting via density map estimation predicts a per-pixel density. Summing the density yields the final count, a common readout in clinical diagnostics and disease monitoring. Yet these models are often hard to audit when errors occur. We present **Proto4DME**, an interpretable density map estimator with faithful explanations by construction. The predicted density (and thus the count) is an additive, non-negative combination of contributions from learned visual patterns (prototypes). Prior prototype-based counting uses signed aggregation, which permits cancellation. In contrast, Proto4DME provides non-canceling attributions, in which increasing a prototype's activation can only increase the predicted density. So prototype heatmaps correspond to positive contributions for the count. Proto4DME learns spatial prototype activation maps from backbone features and selects a compact set of prototypes using sparsity-inducing Hard-Concrete gates. To encourage diverse foreground coverage and prevent prototype collapse, we introduce an entropically-regularized optimal-transport coverage objective. It allocates ground-truth density mass across prototypes under capacity constraints and induces competition among prototypes. Across three microscopy benchmarks (MBM, ADI, and DCC), Proto4DME achieves competitive mean absolute error (MAE) while producing compact, auditable explanations that support error analysis and model debugging.

Data and Code Availability

All datasets used in this study are publicly available: MBM Kainz et al. (2015), ADI Paul Cohen et al. (2017), and DCC Marsden et al. (2018). We use the official dataset releases and the splits described in Sec. 3.1. Our implementation of Proto4DME is publicly available at https://github.com/NRT-D4/Proto4DME to facilitate reproducibility.

Institutional Review Board (IRB) This research does not involve data from live humans. It does not require IRB approval.

1. Introduction

Cell counting is the task of estimating how many cells are present in a microscopy field of view Guo et al. (2022). These counts are a routine quantitative readout in biomedical studies (e.g., measuring proliferation, comparing treatment conditions, or quantifying infection and immune response), and they often feed directly into downstream statistical analyses where small systematic biases can change conclusions.

Automating cell quantification is challenging due to the nature of microscopy images. Images can range from sparsely populated fields to highly crowded regions with substantial overlap, and cells can vary widely in morphology and intensity, and appear under uneven illumination or staining Xie et al. (2018); Kainz et al. (2015). In many datasets, supervision is limited to sparse point annotations of cell centers rather than full instance masks, making classical detection or segmentation pipelines brittle. In addition, background texture and debris can mimic cellular structure, and acquisition settings can shift across batches.

Density map estimation (DME) is therefore a common strategy for counting in crowded scenes Lempitsky and Zisserman (2010). Instead of predicting discrete cell instances, a model predicts a density map whose sum approximates the object count. This formulation is well-suited to biomedical imaging because it can be trained from point annotations, it tolerates overlap when boundaries are am-

biguous, and it produces a spatial output that can be inspected visually Lempitsky and Zisserman (2010); Zhang et al. (2016). Modern density estimators build on fully convolutional backbones and multi-scale context. Crowd-counting architectures such as CSRNet Li et al. (2018) are widely adopted. Vision Transformer (ViT) backbones have also been adapted to microscopy counting, but typically rely on fine-tuning large pretrained models and can be comparatively resource-intensive to train and deploy Mohammed et al. (2025a); Lin et al. (2022).

Even when density map estimators are accurate on average, they can be difficult to trust and debug. When predicted counts deviate from expectations, practitioners need to know whether the model is mistaking debris for cells, missing dim or small cells, or failing in particular tissue regions. This requirement is not only scientific but also operational. In automated image-analysis pipelines, practitioners often need actionable feedback Wang et al. (2024) about failure modes before a model can be deployed in practice Rudin (2019).

Recently, *CountXplain* Mohammed et al. (2025b) introduced prototype-based explanations for microscopy density map estimation. However, aspects of its formulation can hinder faithful attribution in counting. Prototype coefficients may be negative, so highlighted regions can represent negative contributions and reduce the predicted density via cancellation. Moreover, its extreme-point supervision provides limited coverage of diverse cell appearances, and the prototype set size must be specified in advance rather than selected automatically.

We address these challenges with **Proto4DME**, an interpretable density map estimator with faithful explanations by construction. Our main contributions are:

- **Optimal-transport prototype coverage:** an entropically regularized optimal-transport objective that allocates ground-truth density mass across prototypes under capacity constraints, encouraging diverse foreground coverage and preventing collapse.

- **Non-negative density head:** a positivity constrained 1×1 density head that maps prototype similarity maps to a final density map. This yields an exact additive, non-cancelling decomposition of the predicted density (and total count).

- **Automatic prototype selection for density estimation:** an adaptation of Hard-Concrete gates Louizos et al. (2017) to density map estimation, enabling the model to automatically select a compact, data-driven set of prototypes for each dataset.

We evaluate Proto4DME on three microscopy benchmarks, MBM Kainz et al. (2015), DCC Marsden et al. (2018), and ADI Paul Cohen et al. (2017). Proto4DME achieves competitive mean absolute error (MAE) while producing compact, auditable explanations that support error analysis and model debugging.

In the remainder of the paper, we describe the Proto4DME architecture and training objective, present quantitative results and ablations, and demonstrate how the learned prototypes support global summarization of counting concepts as well as localized, faithful explanations of individual predictions.

2. Related Work

Prior OT-based counting methods improve accuracy by *matching predicted and target spatial mass distributions* (including unbalanced variants for mass mismatch) in crowd and cell counting Wang et al. (2020); Babu Sam et al. (2022); Ma et al. (2021); Ding et al. (2023). In our work, OT is only an auxiliary regularizer; the core contribution is a prototype-based density estimator with gated prototype selection and per-prototype similarity maps that yield faithful local explanations and compact concept sets.

Post hoc interpretability methods offer partial insight but have important limitations in this setting. Class Activation Mapping Zhou et al. (2016), Grad-CAM Selvaraju et al. (2017), and pixel-wise attribution methods Bach et al. (2015); Shrikumar et al. (2017); Lundberg and Lee (2017) can fail sanity checks Adebayo et al. (2018) and typically yield heatmaps that neither decompose the final integral count nor map cleanly to reusable biological concepts (cell morphologies, imaging artifacts, or background patterns).

Prototype-based interpretability provides a natural bridge. Self-explainable prototype models justify predictions via similarity to learned prototypical patterns, and prior work (e.g., ProtoPNet) shows that prototypes can ground human-understandable reasoning in representative parts Chen et al. (2020);

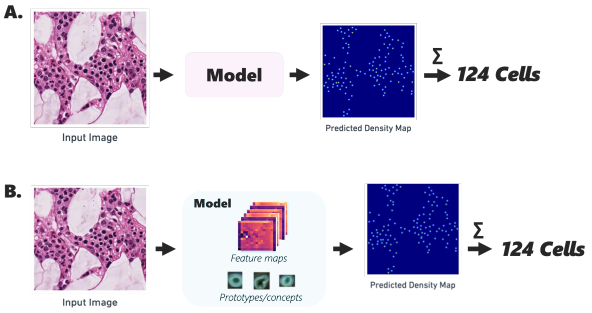

Figure 1: (A) Black-box vs (B) Interpretable Density Map Estimation models

Rymarczyk et al. (2021); Chen et al. (2019); Barnett et al. (2021); Mohammadjafari et al. (2021); Singh and Yow (2021a,b); Djoumessi et al. (2024). While regression variants exist Hesse and Namburete (2022); Hesse et al. (2024), density map estimation adds a spatial requirement. Explanations must specify where attributions appear and how they aggregate into the final count, while remaining faithful by construction and compact so each prediction can be communicated by a small number of dominant concepts Molnar (2022).

CountXplain Mohammed et al. (2025b) is the first prototype-based approach for interpretable density map estimation in microscopy. It learns cell and background prototypes and predicts density as a learned linear combination of prototype similarity maps.

While CountXplain establishes an important direction, several design choices weaken the link between prototype visualizations and faithful, actionable counting explanations. First, its density head is unconstrained and can assign negative coefficients, which enables cancellation. A prototype may strongly activate in a region yet reduce the predicted density and the final count, so prototype heatmaps do not allow a simple additive interpretation of contribution.

Second, its prototype-to-feature alignment is supervised only at two extreme spatial locations per image, namely the max-density location for "cell" and the min-density location for "background". In dense and heterogeneous scenes, this provides limited signal and does not encourage prototypes to collectively cover the full foreground density mass.

Third, the number of prototypes is fixed a priori with no mechanism to deactivate redundant prototypes during training. Redundant concepts can therefore persist and dilute interpretability.

Table 1: Dataset statistics.

Dataset	Image Size	N_{train}/N_{total}	Cell Count
MBM Kainz et al. (2015)	600×600	15/44	126 ± 33
ADI Paul Cohen et al. (2017)	150×150	50/200	165 ± 44
DCC Marsden et al. (2018)	varied	100/176	34 ± 22

3. Methods

3.1. Datasets

We evaluate Proto4DME on three microscopy cell counting benchmarks: **MBM**, **DCC**, and **ADI**. Each dataset provides microscopy images paired with sparse cell annotations (point marks at cell centers or equivalent) from which ground-truth density maps can be constructed. Together, these benchmarks span heterogeneous imaging conditions, diverse cell morphologies, and a wide range of cell densities.

MBM. We use the Modified Bone Marrow (MBM) dataset introduced by Cohen et al. Paul Cohen et al. (2017), derived from the BM dataset of Kainz et al. Kainz et al. (2015). The original BM data consist of real bone marrow images from healthy individuals, with standard staining that depicts nuclei in blue while other constituents appear in shades of pink/red.

ADI. The human subcutaneous adipose tissue (ADI) dataset was constructed from the Genotype-Tissue Expression (GTEx) Consortium Lonsdale et al. (2013) and contains densely packed adipocyte cells. Regions of interest (ROIs) are sampled from high-resolution histology slides using a sliding window and then downsampled to a suitable scale by Paul Cohen et al. (2017). Adipocytes vary substantially in size (approximately 20–200 μm) and are often tightly packed with few gaps, making ADI a challenging test case for automated cell counting.

DCC. Marsden et al. Marsden et al. (2018) built the Dublin Cell Counting (DCC) dataset to represent a wide range of cells, including embryonic mouse stem cells, human lung adenocarcinoma, and human monocytes. Image sizes range from 306×322 to 798×788, increasing dataset variation.

3.2. Model Architecture

Proto4DME is a prototype-based density map estimator built on top of a fully convolutional counting

backbone. This choice provides spatial inductive biases and computational efficiency, promoting robust generalization in the data-scarce regime common in cell counting. Given an input image $I \in \mathbb{R}^{H\times W\times C_{\text{in}}}$, where C_{in}, W, and H are the channel, width, and height, respectively, the model produces (i) a predicted density map $\hat{D}$ and (ii) a set of per-prototype activation maps that provide a faithful, spatially localized explanation of the prediction.

Backbone feature extractor. We adopt a CSRNet-style architecture with a VGG Simonyan and Zisserman (2015) like front-end and a dilated convolution back-end to preserve spatial resolution while providing a large effective receptive field. The backbone maps the image to a mid-level feature tensor

$$F \in \mathbb{R}^{C\times H'\times W'} \tag{1}$$

$$\tilde{F} = \sigma(W_{\text{add}} * F + b_{\text{add}}) \in \mathbb{R}^{C\times H'\times W'} \tag{2}$$

Here, C is the number of backbone feature channels and (H', W') are the backbone output spatial dimensions. We use the feature channels produced by the CSRNet back-end and apply a lightweight 1×1 "add-on" transformation (feature enhancer) to obtain a bounded feature representation. In the add-on, $\sigma(\cdot)$ denotes the sigmoid nonlinearity (so $\tilde{F}$ is bounded elementwise in $[0, 1]$), $*$ is convolution, and W_{add} and b_{add} are the 1×1 convolution weights and bias. This add-on stack stabilizes prototype matching by constraining feature magnitudes.

Prototype bank and distance maps. Proto4DME maintains a bank of K learned prototypes $\{\mathbf{p}_k\}_{k=1}^{K}$, where each $\mathbf{p}_k \in \mathbb{R}^C$ represents a recurrent visual pattern in the backbone feature space. Because background prototypes are typically not used by the counting head and are redundant under our non-negative additive formulation, we use only cell prototypes. Background is then implicitly defined as regions with uniformly low cell-prototype evidence (i.e., low activation across all prototypes). For every spatial location (i, j), we compute the squared Euclidean distance between the local feature vector and each prototype:

$$\Phi_k(i,j) = \left\|\tilde{F}_{:,i,j} - \tilde{\mathbf{p}}_k\right\|_2^2, \qquad \tilde{\mathbf{p}}_k = g_k\,\mathbf{p}_k, \tag{3}$$

where $g_k \in [0, 1]$ is the gate value defined below yielding a distance tensor $\Phi \in \mathbb{R}^{K\times H'\times W'}$. In practice, distances are computed efficiently using an L2-convolution following Chen et al. (2019).

Similarity mapping. Following prior work Chen et al. (2019), we convert distances to prototype similarity maps $S_k(i, j)$ using a monotone log transform.

The backbone feature extractor, squared-ℓ_2 prototype matching, and distance-to-similarity mapping follow standard prototype-learning practice. We next introduce components specific to Proto4DME.

Hard-Concrete gates for prototype selection. To avoid committing to a fixed number of prototypes, we associate each prototype with a gate. Let $g_k \in [0, 1]$ denote the open probability of prototype k under a Hard-Concrete relaxation Louizos et al. (2017). During inference, we use the deterministic expectation g_k. As used in the distance computation, we gate prototype vectors via $\tilde{\mathbf{p}}_k = g_k\,\mathbf{p}_k$. We additionally gate each similarity channel so that prototypes with closed gates are silent:

$$\tilde{\mathbf{p}}_k = g_k\,\mathbf{p}_k \tag{4}$$

$$\tilde{S}_k(i,j) = g_k\,S_k(i,j) \tag{5}$$

Above-baseline evidence (excess activation). Raw similarity maps can contain a non-zero baseline due to feature normalization and the log transform. To make activations reflect above-baseline evidence, we center each prototype similarity map by its spatial mean and retain only positive deviations:

$$E_k(i,j) = \text{ReLU}\left(\tilde{S}_k(i,j) - \mu_k\right), \qquad \mu_k = \frac{1}{H'W'}\sum_{i,j}\tilde{S}_k(i,j) \tag{6}$$

The resulting maps E_k can be interpreted as prototype-specific heatmaps highlighting locations where prototype k matches more strongly than its image-level baseline.

Non-negative 1×1 density head. Finally, the predicted density map at the backbone resolution is obtained as a non-negative linear combination of per-prototype activation maps:

$$\hat{D}(i,j) \;=\; \sum_{k=1}^{K} w_k\,E_k(i,j) \qquad w_k \geq 0 \tag{7}$$

We enforce $w_k \geq 0$ by parameterizing the 1×1 convolution weights with a positivity constraint (e.g., softplus).

Faithful additive decomposition. The predicted count is obtained by summing the density map over space:

$$\hat{n} \;=\; \sum_{i=1}^{H'}\sum_{j=1}^{W'}\hat{D}(i,j) \tag{8}$$

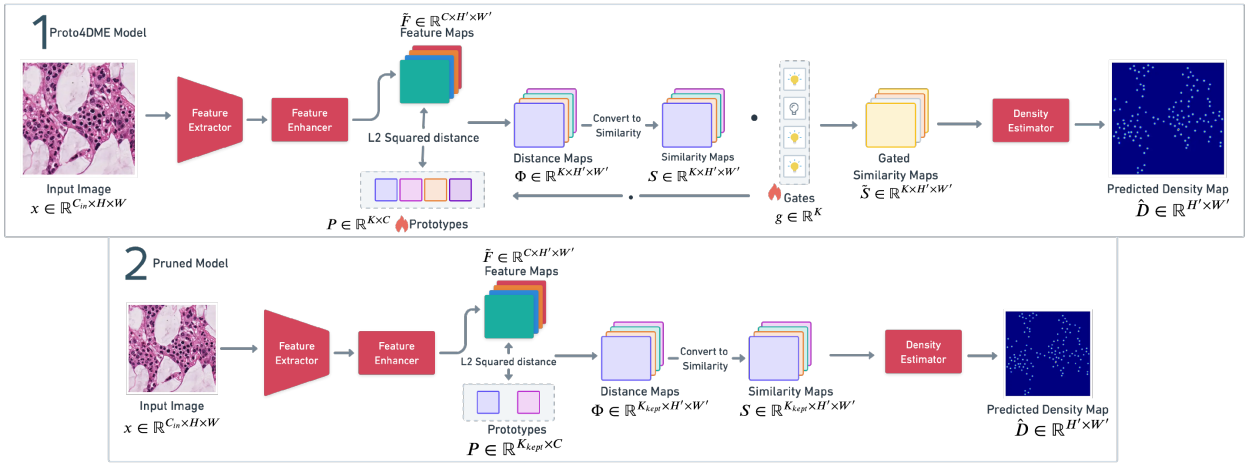

Figure 2: Overview of Proto4DME and pruning. The input image is encoded by a fully convolutional backbone and feature adapter. Prototypes are matched to spatial features via squared L2 distance to form distance maps that are converted to similarity maps. Hard-Concrete gates modulate prototype channels during training. During pruning, prototypes with gates below an automatically determined threshold are removed. A non-negative density head maps prototype similarities to the predicted density map.

Substituting the density-head expression yields an exact additive decomposition into per-prototype contributions:

$$\hat{n} = \sum_{i=1}^{H'} \sum_{j=1}^{W'} \sum_{k=1}^{K} w_k \, E_k(i,j) = \sum_{k=1}^{K} \underbrace{w_k \sum_{i=1}^{H'} \sum_{j=1}^{W'} E_k(i,j)}_{\text{prototype } k \text{ contribution}} \tag{9}$$

Because $w_k \geq 0$ and $E_k(i,j) \geq 0$, contributions are non-negative and cannot cancel. This makes explanations directly comparable across images.

3.3. Training Objective

Proto4DME is trained end-to-end with (i) a density regression loss, (ii) an image-level counting loss on global count, (iii) a prototype coverage and competition term via entropic optimal transport (OT), and (iv) an ℓ_0-style sparsity regularizer on prototype gates.

Density map regression (Huber) and image-level counting Given a ground-truth density map D, the model predicts a density $\hat{D} \in \mathbb{R}^{H' \times W'}$. We supervise the density map with a Huber (Smooth-ℓ_1) loss,

$$\mathcal{L}_{\text{dens}} = \frac{1}{H'W'} \sum_{i,j} \text{Huber}_{\beta}\left(\hat{D}_{ij} - D_{ij}\right) \tag{10}$$

where β is the Huber transition point. In addition, we explicitly constrain the total predicted count by penalizing the absolute error between summed densities:

$$\mathcal{L}_{\text{cnt}} = \left| \sum_{i,j} \hat{D}_{ij} - \sum_{i,j} D_{ij} \right| \tag{11}$$

This term directly optimizes image-level counting accuracy while $\mathcal{L}_{\text{dens}}$ preserves spatial supervision.

Prototype coverage and competition via balanced assignment Unlike CountXplain, which aligns prototypes using extreme points per image (maximum density for cell and minimum density for background), we use entropically regularized optimal transport for balanced assignment. For each image, we treat the ground truth density map $D(i,j)$ as a distribution of foreground mass over spatial locations by normalizing it to sum to one,

$$\mathbf{t}_{ij} = \frac{D(i,j)}{\sum_{u,v} D(u,v) + \delta} \tag{12}$$

where $\delta > 0$ is a small constant for numerical stability. We then flatten $\mathbf{t}_{ij}$ into a vector $\mathbf{t} \in \mathbb{R}^{H'W'}$ using a single spatial index $p \in \{1, \dots, H'W'\}$ corresponding to location (i, j), and denote the resulting entries by $\mathbf{t}_p$. Images with $\sum_{u,v} D(u, v) = 0$ provide no valid unit-mass target distribution, so we skip them (i.e., no OT loss is computed for that image).

Given prototype distance maps $\Phi_k(i, j)$ computed from the backbone features, we form the transport cost as the raw distances, flattened consistently as

$$Q_{k,p} = \Phi_k(i, j) \tag{13}$$

where p is the flattened index of (i, j). Thus $Q \in \mathbb{R}^{K \times H'W'}$ and $\mathbf{t} \in \mathbb{R}^{H'W'}$ share the same spatial indexing.

Prototype capacities are determined by the Hard Concrete gates. Recall $g_k \in [0, 1]$ is the expected open probability of prototype k. We convert these gate values into a probability distribution over prototypes using a temperature-controlled log softmax Hinton et al. (2015)

$$a_k = \begin{cases} \frac{1}{K}, & \text{if } \sum_{\ell=1}^{K} g_\ell < \epsilon_0 \\ \frac{\exp\big(\log(g_k + \epsilon_{gate})/\tau\big)}{\sum_{\ell=1}^{K} \exp\big(\log(g_\ell + \epsilon_{gate})/\tau\big)}, & \text{otherwise.} \end{cases} \tag{14}$$

where τ is the gate temperature, ϵ_{gate} prevents $\log(0)$, and ϵ_0 is a small threshold used to trigger a uniform fallback when all gates are effectively closed. All three are fixed positive constants. We set $\epsilon_{\text{gate}} = 10^{-8}$ purely for numerical stability and use $\epsilon_0 = 10^{-8}$ to detect the degenerate case where $\sum_k g_k$ is effectively zero, in which case we fall back to a uniform capacity allocation. The temperature $\tau > 0$ controls the softness of the capacity distribution; we use a moderate value ($\tau = 0.65$). This construction yields simplex marginals required by optimal transport (nonnegative and summing to one), allocates more capacity to prototypes that are more likely to be active, and still enforces competition because increasing mass for one prototype necessarily reduces it for others.

We then solve for a transport plan $\Pi \in \mathbb{R}^{K \times H'W'}$ via Sinkhorn iterations Cuturi (2013)

$$\begin{aligned} \Pi^\star = \arg\min_{\Pi \geq 0} \langle \Pi, Q \rangle + \varepsilon_{\text{ot}} \sum_{k,p} \Pi_{k,p} \big(\log \Pi_{k,p} - 1 \big) \\ \text{s.t.} \quad \Pi \mathbf{1}_{H'W'} = \mathbf{a}, \qquad \Pi^\top \mathbf{1}_K = \mathbf{t} \end{aligned} \tag{15}$$

where p indexes spatial locations, ε_{ot} is the entropic regularization coefficient, and the constraints ensure that all foreground mass $\mathbf{t}$ is assigned while prototypes compete under the capacity distribution $\mathbf{a}$. The OT loss is defined as the expected transport cost under the optimal plan,

$$\mathcal{L}_{\text{ot}} = \langle \Pi^\star, Q \rangle \tag{16}$$

Because $\mathbf{t}$ is derived from the ground truth density, this term forces coverage of foreground regions, and because $\mathbf{a}$ is gate-controlled, it discourages multiple prototypes from explaining the same locations and promotes specialization.

L0 sparsity regularization We regularize the expected number of active prototypes using the analytic expected L0 penalty induced by Hard Concrete gates,

$$\mathcal{L}_0 = \sum_{k=1}^{K} g_k \tag{17}$$

Total loss Our training objective combines standard density/count supervision with two new regularizers, an OT-style coverage/competition loss and an L_0 gate sparsity loss. We minimize the weighted sum:

$$\mathcal{L} = \lambda_{\text{dens}} \mathcal{L}_{\text{dens}} + \lambda_{\text{cnt}} \mathcal{L}_{\text{cnt}} + \lambda_{\text{ot}} \mathcal{L}_{\text{ot}} + \lambda_0 \mathcal{L}_0. \tag{18}$$

All coefficients in (18) are treated as hyperparameters and selected on a validation set.

Optional post-training pruning and head refitting The Hard-Concrete regularizer encourages the model to rely on a small subset of prototypes, but in practice, a few prototypes may retain small gate probabilities and contribute negligibly to the predicted density. For deployment and improved interpretability, we optionally apply a post-training pruning step that removes such weak prototypes and produces a compact model with fewer heatmaps to inspect. We rank prototypes by an effective contribution score r_k that combines the non-negative density-head weight and the expected gate openness. Recall that $w_k \geq 0$ is the density-head coefficient and $g_k \in [0, 1]$ denotes the expected gate openness for prototype k. We define

$$r_k = w_k g_k \tag{19}$$

We select the keep set using Algorithm 1, steps 1–5 (score computation, knee thresholding Satopaa et al. (2011), and safeguards for minimum retention).

We then construct the pruned model using Algorithm 1, steps 6–8 (dropping prototypes, rewiring the head, and transferring parameters so that retained prototypes preserve their learned influence).

Finally, because pruning can still introduce small deviations due to numerical effects and the removal of stochastic gating during training, we perform a short refitting stage as part of pruning in which the pruned model is distilled to match the unpruned model's density predictions on a calibration set. In this stage, we keep the backbone and prototypes fixed and optimize only the density head. This lightweight procedure typically preserves counting accuracy while improving interpretability and reducing inference cost.

To summarize, Proto4DME couples prototype explanations with density estimation in a single model. Prior work uses gating mainly for sparsity and model compression, such as pruning weights or channels Louizos et al. (2017). We instead apply Hard-Concrete gates to the prototypes, enabling the model to learn a compact, data-driven set of visual concepts. We also change how prototypes are trained. While optimal transport is often used for matching or distribution alignment, we use balanced OT assignment to the ground-truth density mass. This enforces coverage of density and competition between prototypes, reducing redundancy and collapse. Combined with a non-negative density head, Proto4DME provides faithful, non-canceling additive explanations of both the density map and the total count.

Algorithm 1: Optional post-training prototype pruning and head refitting

Input: Trained prototype model $\mathcal{M}$ with prototypes $\{\mathbf{p}\}_{k=1}^{K}$, non-negative head weights $\{w_k\}_{k=1}^{K}$, gate expectations $\{g_k\}_{k=1}^{K}$, minimum keep $K_{min} \geq 1$
Output: Pruned model $\mathcal{M}'$ with fewer prototypes and comparable predictions

1. Compute prototype scores $r_k \leftarrow w_k g_k$ for $k = 1, \ldots, K$
2. Normalize scores to $[0, 1]$, sort them in ascending order, and select a threshold γ using Kneedle algorithm Satopaa et al. (2011)
3. Define keep indices $\mathcal{K} \leftarrow \{k : r_k \geq \gamma\}$
4. If $|\mathcal{K}| = K$, set $\mathcal{K}$ to the indices of the top $\max(K_{\min}, 1)$ scores
5. If $|\mathcal{K}| < K_{\min}$, expand $\mathcal{K}$ to include the top $K_{\min}$ indices
6. Construct the pruned model $\mathcal{M}'$ by keeping only prototypes $\{\mathbf{p}_k\}_{k\in\mathcal{K}}$ and reinitializing the head to have $|\mathcal{K}|$ input channels, then transfer the corresponding head weights for $k \in \mathcal{K}$.
7. Set the gate parameters of $\mathcal{M}'$ so that its gate expectations match $\{g_k\}_{k\in\mathcal{K}}$
8. Freeze backbone and prototype parameters of $\mathcal{M}'$
9. For $\mathcal{Z}$ refit steps:
   (a) Sample a minibatch I
   (b) Compute teacher density $\hat{D}_T \leftarrow \mathcal{M}(I)$ and global count $\hat{n}_T \leftarrow \sum \hat{D}_T$
   (c) Compute student density $\hat{D}_S \leftarrow \mathcal{M}'(I)$ and global count $\hat{n}_S \leftarrow \sum \hat{D}_S$
   (d) Update only the head of $\mathcal{M}'$ to minimize $(\hat{n}_S - \hat{n}_T)^2$
10. Return $\mathcal{M}'$

4. Results

4.1. Experimental setup and evaluation metrics

We evaluate Proto4DME on three microscopy cell counting benchmarks: MBM, ADI, and DCC (Sec. 3.1). For each dataset, we report image-level counting error using mean absolute error (MAE) between the predicted count $\hat{n}$ and the ground-truth count n. Counts are obtained by summing the pixels of the predicted density map.

We compare Proto4DME to (i) a strong non-interpretable density regression baseline using the same counting backbone (CSRNet), and (ii) the closest prototype-based density estimator baseline (CountXplain). For each trial, we randomly sample train/test splits for each dataset and evaluate all methods on identical splits; we repeat this procedure 5 times and report the mean and standard deviation of MAE.

We use $K = 10$ cell prototypes and Hard-Concrete gating initialized with $p_0 = 0.8$ and temperature $\tau = 0.65$. We optimize with Adam (lr $= 5 \times 10^{-3}$, batch size 16) for up to 1000 epochs with early stopping (patience 150). The total loss uses $\lambda_{\text{cnt}} = 1$, $\lambda_{\text{dens}} = 10$, $\lambda_{\text{ot}} = 5$, and $\lambda_0 = 5 \times 10^{-4}$. OT is solved with Sinkhorn iterations (40) and entropic regularization $\varepsilon_{\text{ot}} = 0.08$. We set $\beta = 0.5$, $\delta = 10^{-12}$, and $K_{\min} = 2$, and we use $H' = \frac{1}{8}H$ and $W' = \frac{1}{8}W$. Loss weights and OT parameters were selected using a small validation-based search. The remaining training settings follow standard practice and were kept fixed.

4.2. Counting results

Table 2 reports mean MAE on MBM, ADI, and DCC. Proto4DME achieves the best mean MAE on **MBM** and **DCC**, improving over CSRNet by 3.7% and 21.8%, respectively, and over CountXplain by 14.1% and 21.2%. On **ADI**, Proto4DME improves over CSRNet (10.8 ± 2.77 vs. 12.37 ± 0.64) but remains behind CountXplain (7.71 ± 2.34).

Table 2: Counting results (MAE↓) (mean ± std.) on MBM, ADI, and DCC.

Method	MBM	ADI	DCC
CSRNet	6.19 ± 0.53	12.37 ± 0.64	2.61 ± 0.31
CountXplain	6.94 ± 1.74	$\mathbf{7.71 \pm 2.34}$	2.59 ± 0.23
Proto4DME	$\mathbf{5.96 \pm 0.68}$	10.8 ± 2.77	$\mathbf{2.04 \pm 0.26}$

Note: Proto4DME results are reported for the pruned model (Alg. 1).

While CountXplain attains a lower MAE on ADI, this MAE gap (10.80 vs. 7.71) must be weighed against the cost of explanation faithfulness. The performance gap likely reflects the dataset's inherent difficulty rather than a flaw in Proto4DME's non-negative head. Specifically, adipocytes in the ADI dataset are highly heterogeneous and tightly packed, ranging from 20 to 200 μm in size. They feature pale, near-empty interiors, and only their thin outer membranes are stained. This distinct morphology produces extremely subtle boundaries in dense scenes, making it challenging for spatial models to isolate individual cell instances without broader context. Consequently, overlapping local evidence creates significant ambiguity during density map estimation.

To handle this ambiguity, CountXplain relies heavily on a cancellation mechanism at the cost of faithfulness. As detailed in Table 3, CountXplain assigns negative coefficients to 60% of its prototypes on ADI, resulting in a negative-mass ratio of 0.433. This means that strong activation of a prototype can actually reduce the predicted count. As visible in Figure 3 (ADI row), CountXplain's prototype maps (e.g., Proto 1–Proto 3) tend to activate diffusely over broad non-cell tissue regions—areas where the ground-truth dot annotation marks only the sparse cell centers. The signed density head then compensates for these massive spurious responses by assigning negative coefficients to subtract the excess density. While this unconstrained regression approach recovers overall counting accuracy, it fundamentally breaks additive interpretability. For a practitioner auditing the model, a brightly highlighted heatmap region does not reliably correspond to a counted cell; instead, it may represent a background region where the model is aggressively suppressing its own overpredictions. Thus, the observed performance gap on ADI represents a faithfulness-accuracy tradeoff by design.

In contrast, Proto4DME maintains strict additive interpretability. The guarantee that $w_k \geq 0 \ \forall k$ ensures that every prototype heatmap is a valid positive attribution. This is a crucial structural requirement for auditable counting in biomedical applications, not merely an aesthetic choice. Forced to construct the density map through pure addition, Proto4DME produces more consistently cell-aligned activation patterns across its top prototypes. Because contributions are strictly non-negative, practitioners can read these maps directly as additive spatial components that build up the predicted count (Figure 3). Furthermore, regarding the observed instability on ADI, the dataset's densely packed cells and subtle boundaries create highly overlapping local evidence. This overlap makes learning a strictly non-negative spatial decomposition inherently more sensitive to different training runs. Notably, CountXplain also exhibits substantial variance on ADI (± 2.34) despite having the freedom to use negative cancellation. This shared instability suggests that the variance primarily reflects the intrinsic difficulty of the ADI dataset rather than a modeling deficiency.

Overall, these results indicate that Proto4DME can match or improve upon a strong density-regression baseline, and it remains competitive with the closest prototype-based baseline. In the following sections, we analyze Proto4DME components and explanation quality.

4.3. Local explanations: per-image prototype contribution breakdown

Figure 3 illustrates representative local explanations from Proto4DME by visualizing the input image alongside the top-K prototype *similarity* maps, which highlight regions most similar to each selected prototype.

Qualitatively, this decomposition enables targeted auditing: overcounting cases often coincide with strong activation from prototypes that respond to debris-like texture, while undercounting cases show missing activation on dim or small cells. Because contributions are non-negative, prototype heatmaps can be interpreted directly as attribution for density, and the stacked contribution totals correspond exactly to the model's predicted count.

4.4. Global interpretability: learned prototypes

We visualize learned prototypes to evaluate whether Proto4DME discovers recurring, human-interpretable patterns that align with microscopy

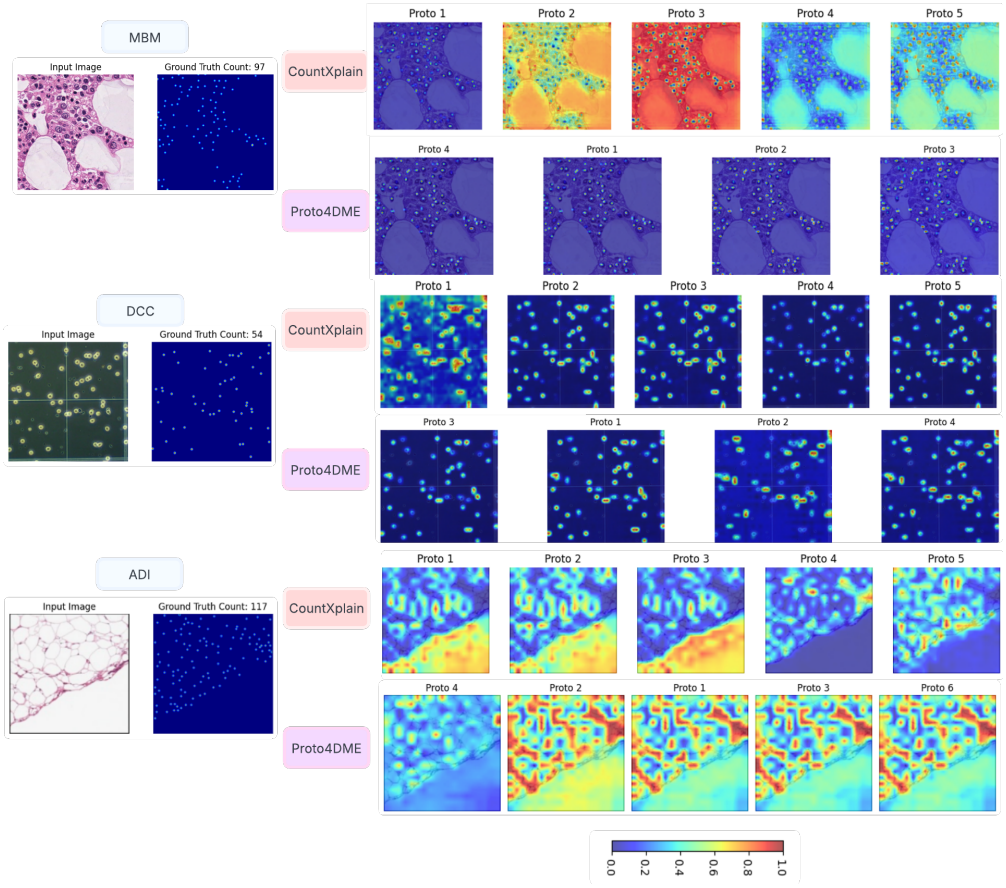

Figure 3: Local prototype explanations on three test images. Left: input and ground-truth density (with total count). Right: prototype similarity maps for CountXplain (top) and Proto4DME (bottom); brighter regions indicate higher similarity/activation (computed from distances in Eq. (3)).

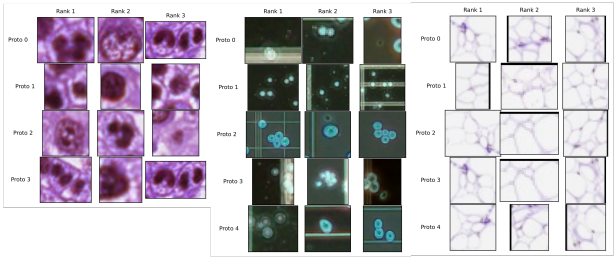

Figure 4: Model's global knowledge captured by prototypes. For each prototype, we display the three most representative image regions from the datasets (ranked by prototype similarity).

structure. For each prototype, we retrieve its top-3 global exemplars by scanning the training set and selecting the spatial locations that minimize the prototype-feature distance (equivalently, maximize similarity). We then extract an adaptive patch around the activated region by thresholding the upsampled prototype activation map and using connected components to obtain a tight bounding box. Figure 4 presents the resulting exemplar gallery (three patches per prototype), enabling direct inspection of the visual patterns each concept captures.

Across datasets, the learned prototypes correspond to distinct cell appearances and local contexts (e.g., compact bright nuclei, dim smaller cells, and clustered structures), while also capturing recurrent confounders such as textured background and debris-like

artifacts. Finally, the sparsity-inducing gate mechanism suppresses redundant prototype channels, yielding a compact set of concepts and a clearer global prototype gallery for practitioner inspection.

4.5. Ablations and analysis of Proto4DME components

Rather than treating ablations as purely accuracy-oriented variants, we evaluate each Proto4DME component against the specific principle it is designed to enforce. These principles include non-negative, cancellation-free attribution, prototype coverage and specialization through OT, and automatic concept set selection through sparsity-inducing gates. We therefore report targeted diagnostics alongside counting errors.

4.5.1. SIGNED AGGREGATION BREAKS ADDITIVE PROTOTYPE CONTRIBUTIONS

Proto4DME constrains the density head to be non-negative, so the predicted density decomposes additively into per-prototype contributions. In contrast, CountXplain aggregates prototype similarity maps using an unconstrained signed head. Inspecting the learned coefficients reveals that CountXplain assigns negative weights on all three datasets (Table 3), implying that stronger activation of some prototypes can reduce the predicted density. This enables cancellation and complicates the interpretation of prototype heatmaps, since high activation does not necessarily correspond to a positive contribution to the final count. Proto4DME avoids this behavior by construction via a non-negative density head.

Table 3: Signed vs. non-negative density-head coefficients reported as a percentage of negative weights and the negative-mass ratio $\sum_{w_i<0}|w_i|/\sum_i|w_i|$ reported.

	ADI	**DCC**	**MBM**
CountXplain: % $w_i<0\downarrow$	60.0	40.0	60.0
CountXplain: neg-mass $\downarrow$	0.433	0.315	0.569
Proto4DME: % $w_i<0\downarrow$	0.0	0.0	0.0
Proto4DME: neg-mass $\downarrow$	0.000	0.000	0.000

Table 4: Pruning compactness across datasets (5 runs). K_{kept} is after Alg. 1; MAE is reported pre→post pruning+refit. Numbers in parentheses indicate the min–max range of K_{kept} over runs.

Dataset	$K_{\text{kept}}/K_{\text{total}}$	MAE (pre→post)
DCC	$5.0\pm0.7/10$ (4–6)	$1.93\pm0.30\rightarrow2.04\pm0.26$
ADI	$7.6\pm1.9/10$ (5–9)	$10.01\pm1.63\rightarrow10.80\pm2.77$
MBM	$3.4\pm0.5/10$ (3–4)	$5.83\pm0.72\rightarrow5.96\pm0.68$

4.5.2. BALANCED ASSIGNMENT (OT) IMPROVES COVERAGE AND PROTOTYPE DIVERSITY

We next study the effect of the balanced assignment objective. Removing OT does not change the explanation mechanism (prototypes and non-negative aggregation remain), but it weakens the pressure for prototypes to collectively cover the foreground density mass. To quantify this effect, we measure the mean minimum prototype distance on foreground pixels ($d^{\text{fg}}_{\min}$) and background pixels ($d^{\text{bg}}_{\min}$) across the validation set, where foreground is defined by nonzero ground-truth density. We summarize separation using $\Delta d_{\min}=d^{\text{bg}}_{\min}-d^{\text{fg}}_{\min}$, and additionally report a scale-invariant relative gap $\Delta_{\text{rel}}=\frac{d^{\text{bg}}_{\min}-d^{\text{fg}}_{\min}}{d^{\text{bg}}_{\min}+d^{\text{fg}}_{\min}}$ to account for potential global rescaling across ablations. With OT enabled, prototypes are closer to foreground than background ($d^{\text{fg}}_{\min}=3.17$, $d^{\text{bg}}_{\min}=3.55$, $\Delta d_{\min}=+0.38$, $\Delta_{\text{rel}}=+0.056$). In contrast, removing OT reverses this relationship ($d^{\text{fg}}_{\min}=20.18$, $d^{\text{bg}}_{\min}=15.40$, $\Delta d_{\min}=-4.78$, $\Delta_{\text{rel}}=-0.134$), indicating that prototypes drift toward background patterns without an explicit coverage and competition constraint. See Appendix A for qualitative results.

4.5.3. GATING PRODUCES COMPACT CONCEPT SETS.

We quantify how gating translates into a discrete, compact prototype set using the optional post-training pruning procedure (Alg. 1), which selects prototypes via the combined score $r_k=w_k g_k$ and refits only the counting head. Table 4 shows that pruning yields consistent compression across datasets, but with dataset-dependent retained set sizes. On DCC, half of the prototypes are kept on average out of five runs (5.0 ± 0.7 out of 10). ADI retains a larger subset (7.6 ± 1.9), suggesting higher concept diversity/coverage requirements. Conversely, MBM

prunes more aggressively (3.4 ± 0.5). Despite these reductions, MAE is generally preserved after refitting (small average change on DCC/MBM), while ADI exhibits higher variance with occasional larger degradation when fewer prototypes are retained. Because pruning uses $r_k = w_k g_k$, this mainly measures which prototypes have both higher gate probability (high g_k) and larger head weights (high w_k), rather than the gates alone.

5. Conclusion

We presented **Proto4DME**, an interpretable density map estimator for cell counting with faithful explanations by construction. A non-negative density head yields an exact, additive decomposition of both the predicted density and the final count into per-prototype contributions, while an entropically regularized optimal-transport coverage objective and Hard-Concrete gating promote diverse foreground coverage and compact concept sets.

Because prototypes are learned without concept labels, they may not align with biologically meaningful categories. So concept validation/curation remains future work. Moreover, Proto4DME inherits biases from density supervision (e.g., kernel choice and annotation noise), which can distort learned prototypes and explanations.

Acknowledgments

This work is partially supported by the National Science Foundation under Grant No. 2152117. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

Deepak Babu Sam, Abhinav Agarwalla, Jimmy Joseph, Vishwanath A Sindagi, R Venkatesh Babu, and Vishal M Patel. Completely self-supervised crowd counting via distribution matching. In *European Conference on Computer Vision*, pages 186–204. Springer, 2022.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7): e0130140, July 2015. ISSN 1932-6203.

Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *Nature Machine Intelligence*, 3(12):1061–1070, 2021.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782, December 2020. ISSN 2522-5839.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Yuanyuan Ding, Yuanjie Zheng, Zeyu Han, and Xinbo Yang. Using optimal transport theory to optimize a deep convolutional neural network microscopic cell counting method. *Medical & Biological Engineering & Computing*, 61(11):2939–2950, 2023.

Kerol Djoumessi, Bubacarr Bah, Laura Kühlewein, Philipp Berens, and Lisa Koch. This actually looks like that: Proto-bagnets for local and global interpretability-by-design. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 718–728. Springer, 2024.

Yinong Guo, Chen Wu, Bo Du, and Liangpei Zhang. Density Map-based vehicle counting in remote sensing images with limited resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 189:201–217, July 2022. ISSN 0924-2716.

Linde S. Hesse and Ana I. L. Namburete. INSightR-Net: Interpretable Neural Network for Regression Using Similarity-Based Comparisons to Prototypical Examples. In Linwei Wang, Qi Dou,

P. Thomas Fletcher, Stefanie Speidel, and Shuo Li, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022*, Lecture Notes in Computer Science, pages 502–511, Cham, 2022. Springer Nature Switzerland. ISBN 9783031164378.

Linde S. Hesse, Nicola K. Dinsdale, and Ana I. L. Namburete. Prototype learning for explainable brain age prediction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7903–7913, January 2024.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. URL http://arxiv.org/abs/1503.02531.

Philipp Kainz, Martin Urschler, Samuel Schulter, Paul Wohlhart, and Vincent Lepetit. You Should Use Regression to Detect Cells. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pages 276–283, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Victor Lempitsky and Andrew Zisserman. Learning To Count Objects in Images. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010.

Yuhong Li, Xiaofan Zhang, and Deming Chen. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1091–1100, June 2018. ISSN: 2575-7075.

Hui Lin, Zhiheng Ma, Rongrong Ji, Yaowei Wang, and Xiaopeng Hong. Boosting crowd counting via multifaceted attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19628–19637, 2022.

John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585, 2013.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Zhiheng Ma, Xing Wei, Xiaopeng Hong, Hui Lin, Yunfeng Qiu, and Yihong Gong. Learning to count via unbalanced optimal transport. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2319–2327, 2021.

Mark Marsden, Kevin McGuinness, Suzanne Little, Ciara E. Keogh, and Noel E. O'Connor. People, Penguins and Petri Dishes: Adapting Object Counting Models to New Visual Domains and Object Types Without Forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8070–8079, 2018.

Sanaz Mohammadjafari, Mucahit Cevik, Mathusan Thanabalasingam, and Ayse Basar. Using protopnet for interpretable alzheimer's disease classification. In *Canadian AI*, 2021.

Abdurahman Ali Mohammed, Catherine Fonder, Ying Wei, Wallapak Tavanapong, Donald S. Sakaguchi, Surya K. Mallapragada, and Qi Li. Cellfmcount: A fluorescence microscopy dataset, benchmark, and methods for cell counting. In *2025 IEEE International Conference on Data Mining (ICDM)*, 2025a. To appear.

Abdurahman Ali Mohammed, Wallapak Tavanapong, Catherine Fonder, and Donald Sakaguchi. Countxplain: Interpretable cell counting with prototype-based density map estimation. In *Medical Imaging with Deep Learning*, 2025b.

Christoph Molnar. *Chapter 3.4 Evaluation of Interpretability*. 2 edition, 2022. URL https://christophm.github.io/interpretable-ml-book/evaluation-of-interpretability.html.

Joseph Paul Cohen, Genevieve Boucher, Craig A. Glastonbury, Henry Z. Lo, and Yoshua Bengio. Count-ception: Counting by Fully Convolutional Redundant Counting. In *Proceedings of the*

IEEE International Conference on Computer Vision Workshops, pages 18–26, 2017.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence 2019 1:5*, 1:206–215, 5 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x.

Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1420–1430, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, October 2017. ISSN: 2380-7504.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017.

Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, April 2015. arXiv:1409.1556 [cs].

Gurmail Singh and Kin-Choong Yow. An interpretable deep learning model for covid-19 detection with chest x-ray images. *Ieee Access*, 9:85198–85208, 2021a.

Gurmail Singh and Kin-Choong Yow. These do not look like those: An interpretable deep learning model for image recognition. *IEEE Access*, 9: 41482–41493, 2021b.

Alan Q Wang, Batuhan K Karaman, Heejong Kim, Jacob Rosenthal, Rachit Saluja, Sean I Young, and Mert R Sabuncu. A framework for interpretability in machine learning for medical imaging. *IEEE Access*, 12:53277–53292, 2024.

Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai Nguyen. Distribution matching for crowd counting. *Advances in neural information processing systems*, 33:1595–1607, 2020.

Yuanpu Xie, Fuyong Xing, Xiaoshuang Shi, Xiangfei Kong, Hai Su, and Lin Yang. Efficient and Robust Cell Detection: A Structured Regression Approach. *Medical image analysis*, 44:245–254, February 2018. ISSN 1361-8415. doi: 10.1016/j.media.2017.07.003. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6051760/.

Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, June 2016. ISSN: 1063-6919.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929. IEEE Computer Society, June 2016. ISBN 9781467388511.

Appendix A. OT ablation: qualitative results

Qualitative analysis (Fig. 5). Figure 5 shows the top-K Proto4DME similarity maps for a representative **MBM** test image produced by a model trained *without* the OT balanced assignment loss. Although several prototypes fire close to dark, nucleus-like structures, the responses are largely diffuse and highly overlapping across prototypes. Many heatmaps highlight similar locations and extend broadly over tissue regions rather than concentrating tightly on cell instances. This redundancy suggests limited prototype specialization and weak pressure to provide complementary contributions. In the absence of OT, prototypes can instead gravitate toward ubiquitous textures and staining micro-patterns that are easy to match throughout the image, which qualitatively manifests as widespread hotspots and reduced foreground–background selectivity. These visual patterns are consistent with our distance-based diagnostics: without OT, the average minimum prototype distance is smaller on background than on foreground (negative $\Delta d_{\min}$), indicating that prototypes are, on balance, better matched to background regions. Overall, Appendix Fig. 5 provides qualitative evidence that OT is important for promoting foreground coverage and discouraging prototype drift toward background cues.

Appendix B. Prototype contribution breakdown example

Prototype contributions toward the predicted count per image. For each image, Proto4DME produces a faithful breakdown of the predicted count into non-negative per-prototype terms. Concretely, the predicted count decomposes as $\hat{n} = \sum_k c_k$, where $c_k = \sum_{i,j} w_k E_k(i,j) \geq 0$, E_k is the prototype activation map, and $w_k \geq 0$ is the non-negative density-head coefficient (Eq. 9). Table 5 shows an example local explanation after pruning. All remaining prototypes are included, and their contributions sum exactly to the predicted count $\hat{n}$. In this example, the top three prototypes account for 65.5% of $\hat{n}$, providing a compact summary of the main drivers of the prediction. Because contributions are strictly additive and non-negative, prototype heatmaps can be interpreted directly as attributions for density, without cancellation effects.

Table 5: Example local explanation for a single image from the DCC dataset after pruning. Proto4DME decomposes the predicted count $\hat{n}$ into non-negative per-prototype contributions c_k. We report all remaining prototypes (post-pruning), and contributions sum exactly to $\hat{n}$.

Proto	g_k	c_k	**% of** $\hat{n}$
p_1	0.967	3.372	23.16%
p_4	0.575	3.269	22.45%
p_3	0.952	2.893	19.88%
p_0	0.846	2.619	17.99%
p_2	0.915	2.405	16.52%
Top-3 total		9.534	65.49%
All-prototypes total		14.557	100%
Predicted count $\hat{n}$		14.557	100%

Table 6: Example local explanation for a single image from the ADI dataset after pruning. Proto4DME decomposes the predicted count $\hat{n}$ into non-negative per-prototype contributions c_k. We report all remaining prototypes (post-pruning), and contributions sum exactly to $\hat{n}$.

Proto	g_k	c_k	**% of** $\hat{n}$
p_5	0.989	33.617	16.08%
p_4	0.961	26.310	12.58%
p_0	0.952	24.346	11.64%
p_6	0.956	23.503	11.24%
p_2	0.933	22.771	10.89%
p_7	0.951	22.399	10.71%
p_8	0.919	20.586	9.85%
p_1	0.936	18.113	8.66%
p_3	0.946	17.434	8.34%
Top-3 total		84.273	40.80%
All-prototypes total		209.079	100%
Predicted count $\hat{n}$		209.079	100%

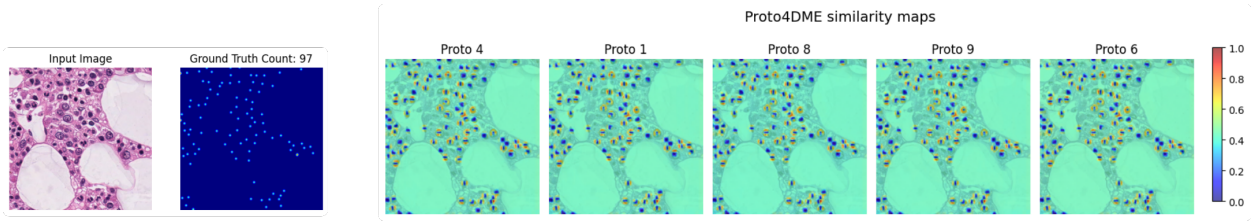

Figure 5: Prototype similarity maps obtained from a model that was trained without OT loss

Table 7: Example local explanation for a single image from the MBM dataset after pruning. Proto4DME decomposes the predicted count $\hat{n}$ into non-negative per-prototype contributions c_k. We report all remaining prototypes (post-pruning), and contributions sum exactly to $\hat{n}$.

Proto	g_k	c_k	**% of** $\hat{n}$
p_3	0.640	96.842	57.85%
p_4	0.894	31.455	18.79%
p_1	0.892	31.200	18.64%
p_2	0.835	7.565	4.52%
p_0	0.772	0.349	0.21%
Top-3 total		159.497	95.28%
All-prototypes total		167.411	100%
Predicted count $\hat{n}$		167.411	100%