Supplemental Information — scNym: Semi-supervised adversarial neural networks for single cell classification

Jacob C. Kimmel
Calico Life Sciences, LLC
South San Francisco, CA, 94080
`jacob@calicolabs.com`

David R. Kelley
Calico Life Sciences, LLC
South San Francisco, CA, 94080
`drk@calicolabs.com`
* Lead Contact

Supplemental Methods

Semi-supervision with MixMatch

We train semi-supervised scNym models using the MixMatch framework [Berthelot et al., 2019], treating the target dataset as unlabeled data $\mathcal{U}$. At each iteration, MixMatch samples minibatches from both the labeled dataset $(\mathbf{X}, \mathbf{y}) \sim \mathcal{D}$ and unlabeled dataset $\mathbf{U} \sim \mathcal{U}$. We generate "pseudolabels" [Lee, 2013] using model predictions for each observation in the unlabeled minibatch. While generating pseudolabels, we run the model in typical inference mode–deactivating dropout layers, not updating batch normalization running statistics, and not propagating gradients. We use the running estimates of feature mean and variance computed during training for batch normalization during pseudolabeling.

$$u_i \sim \mathbf{U}$$
$$z_i = f_\theta(u_i)$$

Our procedure differs slightly from standard MixMatch in that we do not average across several augmented versions of unlabeled observations, but rather pseudolabel a single, unmodified observation. This choice was motivated by empirical performance. Augmentation ensembling may be unable to deliver improved accuracy because only simple augmentations are available for gene expression data, relative to the rich set of transformations available for the imaging data originally studied with MixMatch.

We next "sharpen" the pseudolabels using a scaling procedure, often referred to as "temperature scaling" [Hinton et al., 2015, Guo et al., 2017].

$$\text{Scale}(z_i, T) = z_i^{\frac{1}{T}} / \sum_{j=1}^{J} z_j^{\frac{1}{T}}$$

where z_i is a probability vector and T is a temperature parameter. Values of $T > 1$ increase the entropy of the probability vector ("smoothing"), while values of $T < 1$ decrease the entropy of the probability vector ("sharpening"). For our experiments, we set $T = 0.5$ such that the transformation acts as a form of entropy minimization on the pseudolabels. This entropy minimization encourages unlabeled examples to belong to one of the described classes.

We then randomly mix each observation in both the labeled and unlabeled minibatches with another observation using MixUp [Zhang et al., 2018]. We allow labeled and unlabeled observations to mix together during this procedure. As in standard MixUp training, we mix both the observed features and the labels/pseudolabels between the two randomly paired observations.

$$\lambda \sim \text{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1 - \lambda)$$
$$w_m = \text{Mix}_{\lambda'}(w_i, w_j)$$
$$q_m = \text{Mix}_{\lambda'}(q_i, q_j)$$

where (w_i, q_i) is either a labeled observation and ground truth label (x_i, y_i) or an unlabeled observation and the pseudolabel (u_i, z_i).

We set the MixUp parameter $\lambda' = \max(\lambda, 1 - \lambda)$ such that the first example in the MixUp operation always dominates the resulting mixed sample. This allows us to maintain the labeled or unlabeled identity of each observation in the mixed minibatch. This procedure yields a minibatch $\mathbf{X}'$ of mixed labeled observations and a minibatch $\mathbf{U}'$ of mixed unlabeled observations.

We introduce a semi-supervised interpolation consistency penalty during training in addition to the standard supervised loss. For observations and pseudolabels in the mixed unlabeled minibatch U', we penalize the mean squared error (MSE) between the mixed pseudolabels and the model prediction for the mixed observation.

$$L_{\text{SSL}}(\mathbf{U}', f_\theta) = \mathbb{E}_{u_m, z_m \sim \mathbf{U}'} \| f_\theta(u_m) - z_m \|_2^2$$

This encourages the model to provide smooth interpolations between observations and their ground truth or pseudolabels, generalizing the decision boundary of the model. We use MSE on the model predictions and mixed pseudolabel rather than the cross-entropy because it is bounded and therefore less sensitive to incorrect predictions [Berthelot et al., 2019].

We weight this unsupervised loss relative to the supervised cross-entropy loss using a weighting function $\lambda_{\text{SSL}}(t) \rightarrow [0, 1]$.

$$L(\mathbf{X}', \mathbf{U}', f_\theta, t) = L_{\text{CE}}(\mathbf{X}', f_\theta) + \lambda_{\text{SSL}}(t) L_{\text{SSL}}(\mathbf{U}', f_\theta)$$

We initialize this coefficient to $\lambda_{\text{SSL}} = 0$ and increase the weight to a final value of $\lambda_{\text{SSL}} = 1$ over 100 epochs using a sigmoid schedule. We set λ_{SSL} based on recommendations from the MixMatch authors. We adopt a sigmoid scaling function that is common in the semi-supervised learning field [Verma et al., 2019].

$$\lambda_{\text{SSL}}(t) = \exp(-5 * (t/100 - 1)^2)$$

Domain Adaptation with Domain Adversarial Neural Networks

We use domain adversarial networks (DAN) as an additional approach to incorporate information from the target dataset during training [Ganin et al., 2016]. DANs are one of a family of domain adaptation approaches that attempt to transfer labels from a training set to a target set, where the target distribution is systematically different than the training distribution. We can readily appreciate the application to single cell identity classification, as the training and target domains are likely to differ across technologies, conditions, batches, and laboratories.

The DAN method encourages the classification model to embed cells from the training and target dataset with similar coordinates, such that training and target datasets are well-mixed in the embedding. By encouraging the training and target dataset to be well-mixed, we take advantage of the inductive bias that cell identity classes in each dataset are similar, despite technical variation or differences in conditions. This bias allows the model to adapt across domains, producing an embedding where training and target data are well-mixed.

If cell type proportions are different between training and target domains, this inductive bias may be ill-suited and the domain adversary could conceivably harm performance. For this reason, we use a low weight for the adversarial penalty. Empirically, we find that the adversary improves performance even when the cell type distributions are imbalanced. For example, we find that in our Tabula Muris 10x Chromium to Mouse Cell Atlas transfer experiment, the adversary improves performance even though the cell type distributions differ drastically (Fig. S6).

We introduce this technique into scNym by adding an adversarial domain classification network g_ϕ. We implement g_ϕ as a two-layer neural network with a single hidden layer of 256 units and a rectified linear unit activation, followed by a classification layer with two outputs and a softmax activation. This adversary attempts to predict the domain of origin d from the penultimate classifier embedding v of each observation. For each forward pass, it outputs a probability vector $\hat{d}$ estimating the likelihood the observation came from the training or target domain.

We assign a domain label d to each molecular profile based on the experiment of origin. For experiments with only a single training dataset and single target dataset, we set cells from the training dataset to $d = 0$ and cells from the target dataset to $d = 1$. For multi-domain training experiments in the mouse kidney, we assigned a unique domain label to each sequencing protocol in the training and target datasets. We likewise assigned a unique domain label for each experimental method used in multi-domain cortex nuclei task (Fig. S9). We one-hot encode all domain labels.

During training, we pass a minibatch of labeled observations $x \in \mathbf{X}$ and unlabeled observations $u \in \mathbf{U}$ through the domain adversary to predict domain labels.

$$\hat{d} = g_\phi(v) = g_\phi(f_\theta(x)^{(l-1)})$$

where $\hat{d}$ is the domain probability vector and $v = f_\theta(x)^{(l-1)}$ denotes the embedding of x from the penultimate layer of the classification model f_θ.

We fit the adversary using a multi-class cross-entropy loss:

$$L_{\text{adv}}(\mathbf{X}, \mathbf{U}, f_\theta, g_\phi) = \mathbb{E}_{(x,d)\sim\mathbf{X};(u,d)\sim\mathbf{U}} \left[-\sum_{j=1}^{J} d_j \log(\hat{d}_j) \right]$$

where $j \in J$ are domain indicator indices and J is the set of source domains.

To make use of the adversary for training the classification model, we use the "gradient reversal" trick at each backward pass. We update the parameters ϕ of the adversary using the gradients computed during a backward pass from L_{adv}. At each backward pass, this optimization improves the adversarial domain classifier. Our update rule for the parameters ϕ is therefore a standard gradient descent:

$$\phi_t = \phi_{t-1} - \eta \frac{\partial L_{\text{adv}}}{\partial \phi}$$

where η is the learning rate.

We update the parameters θ of the classification model using the *inverse* of the gradients computed during a backward pass from L_{adv}. Using the inverse gradients encourages the classification model f_θ to generate an embedding where it is difficult for the adversary to predict the domain. Our update rule for the classification model parameters therefore becomes:

$$\theta_t = \theta_{t-1} - \eta \left(\frac{\partial L_{\text{CE}}}{\partial \theta} + \lambda_{\text{SSL}}(t) \frac{\partial L_{\text{SSL}}}{\partial \theta} - \lambda_{\text{adv}}(t) \frac{\partial L_{\text{adv}}}{\partial \theta} \right)$$

We increase the weight of the adversary gradients from $\lambda_{\text{adv}} \to [0, 0.1]$ over the course of 20 epochs during training using the sigmoid schedule above. By scaling the adversary's gradients, we allow both the classification model f_θ and the adversary g_ϕ to find an approximate fit before introducing the adversarial penalty. We scale the adversarial *gradients* flowing to θ, rather than the adversarial loss term, so that full magnitude gradients are used to train a robust adversary g_ϕ.

We set λ_{adv} based on the authors' experiments in the natural image domain [Ganin et al., 2016]. This setting appears sufficient to promote domain adaptation (Fig. S17) while also training a robust classifier.

Incorporating both MixMatch and the domain adversary, our full loss function becomes:

$$L(\mathbf{X}, \mathbf{U}, \mathbf{X}', \mathbf{U}', f_\theta, g_\phi, t) = L_{\text{CE}}(\mathbf{X}', f_\theta) + \lambda_{\text{SSL}}(t) L_{\text{SSL}}(\mathbf{U}', f_\theta) + L_{\text{adv}}(\mathbf{X}, \mathbf{U}, f_\theta, g_\phi, t)$$

Pseudolabel Thresholding for New Cell Type Discovery

Entropy minimization and domain adversarial training enforce an inductive bias that all cells in the target dataset belong to a class in the training dataset. For many cell type classification tasks, this assumption is valid and useful. However, it is violated in the case where new, unseen cell types are present in the target dataset. We introduce an alternative training configuration to allow for quantitative identification of new cell types in these instances.

We adopt a notion of "pseudolabel confidence thresholding" introduced in the FixMatch method [Sohn et al., 2020]. We set a minimum pseudolabel confidence $\tau = 0.9$ to identify unlabeled observations with high and low confidence pseudolabels, marked by a binary indicator $c_i \to \{0, 1\}$, where $c_i = 1$ is a high confidence pseudolabel. We determine pseudolabel confidence *before* entropy minimization (label sharpening) is applied. We chose τ based on performance in other problem domains [Sohn et al., 2020].

$$
\begin{aligned}
z_i &= f_\theta(u_i) \\
c_i &= \mathbb{1}(z_i > \tau)
\end{aligned}
$$

where $\mathbb{1}(\cdot)$ is the indicator function.

We have observed that new cell types will receive low confidence pseudolabels, as they do not closely resemble any of the classes in the training set (Fig. S12). We wish to exclude these low confidence pseudolabels from our entropy minimization and domain adversarial training procedures, as these methods both encourage the classifier to provide more confident labels from the training dataset for unlabeled examples. In the case of new cell types, these procedures may pathologically cause the model to label new cell types with incorrect but high confidence scores for the nearest cell type in the training set.

Pseudolabels influence the model parameters θ through each of the three components of our loss – supervised, semi-supervised interpolation consistency, and adversarial. We make two modifications to the training procedure to prevent low confidence pseudolabels from contributing to any component of the loss.

First, we use only high confidence pseudolabels in the MixUp operation of the MixMatch procedure. This prevents low confidence pseudolabels from contributing to the supervised classification loss through MixUp with labeled samples. It also prevents low confidence pseudolabels from mixing with high confidence pseudolabels, allowing us to compute the semi-supervised loss exclusively on high confidence pseudolabels.

$$
\begin{gathered}
\mathbf{W} = \mathbf{X} :: \mathbf{U}_{c=1} \\
w_m = \text{Mix}_\lambda(w_i, w_j); q_m = \text{Mix}_\lambda(q_i, q_j)
\end{gathered}
$$

where $\mathbf{W}$ is a row-wise concatenation of the labeled minibatch and confident pseudolabeled minibatch and $\mathbf{U}_{c=1}$ are the observations in $\mathbf{U}$ with confident pseudolabels.

We scale the interpolation consistency loss based on the fraction of confident pseudolabels as in FixMatch [Sohn et al., 2020]. The semi-supervision penalty therefore scales with the number of high confidence observations available.

$$
L_{\text{SSL}}(\mathbf{U}', f_\theta) = \frac{\sum_i^{N'} c_i}{N'} \mathbb{E}_{(u_m, z_m) \sim \mathbf{U}'_{c=1}} \| f_\theta(u_m) - z_m \|_2^2
$$

where N' is the number of elements in the unlabeled minibatch.

Second, we use only unlabeled examples with high confidence pseudolabels to train the domain adversary. This prevents unlabeled examples with low confidence pseudolabels from contributing to the adversarial loss. These low confidence unlabeled examples can therefore occupy a unique region in the model embedding, even if they are easily discriminated from training examples.

Our adversarial loss is then only slightly modified to penalize domain predictions only on confident samples in the pseudolabeled minibatch, $\mathbf{U}_{c=1}$.

$$
L_{\text{adv}}(\mathbf{X}, \mathbf{U}, f_\theta, g_\phi) = \mathbb{E}_{(x,d) \sim \mathbf{X}; (u,d) \sim \mathbf{U}_{c=1}} \left[- \sum_{j=1}^{J} d_j \log(\hat{d}_j) \right]
$$

where $j \in J$ are domain indicator indices and J is the set of source domains.

We found that this pseudolabel thresholding configuration option was essential to provide accurate, quantitative information about the presence of new cell types in the target dataset (Fig. S13). However, this option does modestly decrease performance when new cell types are not present. We therefore enable this option when the possibility of new cell types violates the assumption that the training and target data share the same set of cell types. We have provided a simple toggle in our software implementation to allow users to enable or disable this feature.

Interpretability Comparison

We compared the biological relevance of features selected by scNym and SVM as a baseline by computing cell type specific Gene Ontology enrichments. We trained both scNym and an SVM to transfer labels from the Tabula Muris 10x

Genomics dataset to the Tabula Muris Smart-seq2 dataset. We then extracted feature importance scores from the scNym model using integrated gradients and from the SVM model based on coefficient weights. We selected cell type markers for each model as the top $k = 100$ genes with the highest integrated gradient values or SVM coefficients.

We selected a set of 19 cell types in the Tabula Muris for which cell type specific Gene Ontology terms were available. We collected all Gene Ontology terms that contained (1) the cell type in the name or (2) the name of the relevant tissue (e.g. "mammary" for "luminal epithelial cell of mammary gland"). We then computed the enrichment of the relevant cell type specific Gene Ontology terms in scNym-derived and SVM-derived cell type markers using Fisher's exact test. We removed gene sets that had an Odds-Ratio of zero for both methods from further analysis. These gene sets may not accurately capture gene expression features of the target cell type, and their removal leads to no loss of information about relative performance as both methods have zero recovery. We present a sample of the gene sets used (Table S4).

Fisher's exact test yields an Odds-Ratio, the odds that a gene in the cell type marker set is in the Gene Ontology term relative to a gene *not* in the cell type marker set. We compared the mean Odds-Ratio across relevant Gene Ontology terms between scNym-derived markers and SVM-derived markers. To determine statistical significance of a difference in these mean Odds-Ratios, we performed a paired *t*-test across cell types.

We performed the procedure above using $k \in \{50, 100, 150\}$ to determine the sensitivity of our results to this parameter. We found that scNym integrated gradients had consistently stronger enrichments for relevant Gene Ontology terms across cell types for all values of k.

Baseline Methods

scmap-cluster and scmap-cell: We trained scmap-cluster models using the 1,000 most variable genes selected using "M3Drop" [Kiselev et al., 2018, Andrews and Hemberg, 2018]. scmap-cell models were trained with $k = 10$ nearest neighbors and the 1,000 most variable genes selected with "M3Drop" using cosine similarities as a distance metric. We constructed the scmap-cell nearest neighbor index using the default number of sub-centroids, $k_{\text{sub}} = \sqrt{N}$ where N is the number of cells. We set scmap hyperparameters to maximize performance based on optimizations in the original publication [Kiselev et al., 2018]. We set the minimum similarity parameter threshold $= 0.0$ to allow scmap methods to predict a class for all cells. However, we found that even with this setting, a small number of cells were given the "unassigned" label. To allow for a more direct comparison, we manually extracted the nearest neighbor graph and called cell types for cells predicted as "unassigned" by scmap-cell.

scmap-cell-exact: We also implemented a classifier based on scmap-cell with an exact *k*-nearest neighbors search algorithm, rather than the approximate search in the original scmap-cell implementation. We refer to this baseline as "scmap-cell-exact". We implemented the scmap-cell-exact classifier using scikit-learn "KNeighborsClassifier" with a cosine distance metric. We fit scmap-cell-exact models on the top 1,000 most variable genes selected using the M3Drop procedure with $k = 10$ nearest neighbors, mirroring the settings for the original scmap-cell implementation.

SVM: Our SVM baseline models were trained using the scikit-learn "LinearSVC" implementation with an l_2 regularization term weighted with $\lambda_{\text{reg}} = 1$. We fit models using a squared hinge loss objective and used the "one-versus-rest" multiclass training strategy. This approach and its hyperparameter settings were suggested as the result of a single-domain benchmarking study [Abdelaal et al., 2019]. We estimated SVM prediction confidence using Platt Scaling [Platt, 1999] with 5-fold cross-validation to fit the logistic scaling parameters, as implemented in "CalibratedClassifierCV" in scikit-learn.

scPred: We trained scPred models [Alquicira-Hernandez et al., 2019] per the authors' recommendations, with one exception. We set threshold $= 0$ so that scPred predicted a class for all observations. This allowed us to compute accuracy scores comparable to other methods, whereas scPred would otherwise report "Undetermined" for many observations.

singleCellNet: We trained singleCellNet models [Tan and Cahan, 2019] per the authors' recommendations. We used the parameter settings nRand $= 70$, nTopGenes $= 10$, nTrees $= 1000$, nTopGenePairs $= 25$ as recommended.

CHETAH: CHETAH models [de Kanter et al., 2019] were trained per the authors' recommendations. As with scPred models, we set threshold $= 0$ so that CHETAH models reported a class label for each observation, rather than "Undetermined." This allowed us to compare CHETAH accuracy to other methods on a fair basis.

Harmony-SVM: We used the "harmony" unsupervised domain adaptation procedure as a pre-processing step for SVM training [Korsunsky et al., 2019]. We fit harmony embeddings using the default parameters in the "harmonypy" implementation to integrate training and target datasets. We found that harmony integration converged for all benchmarking tasks. We then trained SVM classifiers as described above using the harmony embedding latent variables as input features for each cell, rather than gene expression values for each gene.

LIGER-SVM: We also used the "LIGER" unsupervised domain adaptation procedure as a pre-processing step for SVM training [Stuart et al., 2019]. We fit the LIGER integrated NMF embedding using a rank $k = 20$ based on the authors' suggestion in their documentation. As for the "harmony-SVM" method, we trained SVM classifiers using the LIGER latent variables rather than gene expression values.

scANVI: We fit scANVI models [Lopez et al., 2018, Xu et al., 2019] following the authors' instructions in their tutorial using their package "scvi-tools". We fit scANVI models using a set of highly variable genes selected using a heuristic procedure suggested by the scANVI authors (top 4000 genes based on normalized dispersion). We first optimized scANVI models for up to 400 epochs in an unsupervised fashion, followed by 100 epochs using cell annotations for semi-supervision. The exact number of unsupervised epochs is chosen based on the dataset size by the "scvi-tools" software. We provided domain labels as "batch" labels to the scANVI model for domain adaptation during optimization. We parameterized scANVI models with single hidden layer encoders and decoders, each containing 128 units. We used 10 latent dimensions for the autoencoder latent space. We used zero-inflated negative binomial likelihoods as a reconstruction loss for unsupervised optimization.

Performance Benchmarking

For all benchmarks, we computed the mean accuracy across cells ("Accuracy"), Cohen's κ-score, and the multiclass recevier operating characteristic (MCROC). We computed the MCROC as the mean of ROC scores across cell types, treating each cell type as a binary classification problem.

For the Rat Aging Cell Atlas [Ma et al., 2020] benchmark, we trained scNym models on single cell RNA-seq from young, *ad libitum* fed rats (5 months old) and predicted on cells from aged rats (*ad libitum* fed or calorically-restricted). We converted the cell type annotations provided by the authors into "cell ontology" terms for consistency with common nomenclature [Diehl et al., 2016]. We removed genes expressed in < 20 cells to filter uninformative features. We also filtered cells with < 100 or > 5000 expressed genes to remove low-quality profiles and likely multiplets.

For the human PBMC stimulation benchmark, we trained models on unstimulated PBMCs collected from multiple human donors and predicted on IFNB1 stimulated PBMCs collected in the same experiment [Kang et al., 2017]. Prior to training and prediction, we removed doublet cells as annotated by the authors and cells without valid cell type labels ("nan").

For the Tabula Muris cross-technology benchmark, we trained models on Tabula Muris 10x Genomics Chromium platform and predicted on data generated using Smart-seq2. We limited our analyses to tissues with similar cell type labels between the two technologies: Bladder, Heart and Aorta, Kidney, Limb Muscle, Liver, Lung, Mammary Gland, Spleen, Thymus, and Tongue. We amended the cell ontology class labels of the Smart-seq2 data to match the degenerate names provided in the 10x Genomics data. We removed cell types in the Smart-seq2 data that were not present in the 10x Chromium data. We also removed cells with the ambiguous "kidney cell" label from the 10x Genomics data which are not found in the Smart-seq2 data. We also corrected labels for a set of "kidney loop of Henle epithelial cells" (*Slc12a1*+, *Umod*+) that were incorrectly labeled as "kidney collecting duct epithelial cells".

For the Mouse Cell Atlas (MCA) [Han et al., 2018] benchmark, we trained models on single cell RNA-seq from lung tissue in the Tabula Muris 10x Chromium data [Tabula Muris Consortium, 2018] and predicted on MCA lung data. Due to the different cell annotation ontologies in these datasets, we manually re-annotated the Mouse Cell Atlas using the cell ontology class specification so that classes matched the Tabula Muris. During evaluation, we merged "classical monocyte" and "non-classical monocyte" classes predicted by each model to form a single "monocyte" class to match annotations available for the MCA.

For the spatial transcriptomics benchmark, we trained models on spatial transcriptomics from a mouse sagittal-posterior brain section and predicted labels for another brain section. Sagittal-posterior brain section datasets from the 10x Genomics Visium Spatial Transcriptomics demonstration data were downloaded from `https://www.10xgenomics.com/resources/datasets/`. We manually annotated "region" identities based on the predominant cell type markers in each transcriptional cluster of cells. We clustered cells by selecting highly variable genes, performing principal component analysis (PCA), and then applying Leiden community detection to the PCA latent space. We considered *Neurod1, Cpne7, Ntng1* and *Ddn* to be neuronal markers, *Slc6a11* to be an astrocyte marker, *Plp1* to be an oligodendrocyte marker, and *Car8* and *Sbk1* to be oligodendrocyte precursor cell markers.

For the single cell to single nucleus benchmark in the mouse kidney, we trained scNym models on all single cell data from six unique sequencing protocols and predicted labels for single nuclei from three unique protocols [Denisenko et al., 2020]. For the single nucleus to single cell benchmark, we inverted the training and target datasets above to train on the nuclei datasets and predict on the single cell datasets. We set unique domain labels for each protocol during training in both benchmark experiments. To evaluate the impact of multi-domain training, we also trained models on only one single cell or single nucleus protocol using the domains from the opposite technology as target data.

For the multi-domain cross-technology benchmark in mouse cortex nuclei, we generated four distinct subtasks from data generated using four distinct technologies to profile the same samples [Ding et al., 2020]. We trained scNym and baseline methods to predict labels on one technology given the remaining three technologies as training data for all possible combinations. Each combination in this regime represents one of four unique tasks. We used each technology as a unique domain label for scNym.

For the cross-species mouse to rat demonstration, we selected a set of cell types with comparable annotations in the Tabula Muris and Rat Aging Cell Atlas [Ma et al., 2020] to allow for quantitative evaluation. We selected kidney endothelial cells, kidney loop of Henle epithelial cells, liver hepatocytes, and T cells from the kidney and liver data of both animals. For the rat data, we included B cells in the kidney and liver, while for the Tabula Muris we added B cells and T cells from the spleen because no B cells were captured in the kidney or liver data. We subset the data to expressed genes with unambiguous orthology relationships across species based on Ensembl BioMart. We trained scNym with mouse data as the source domain and rat data as the target domain. We used the new identity discovery configuration to account for the potential for new cell types in a cross-species experiment.

For the cross-species mouse to human demonstration, we similarly selected a set of cell types with comparable cell annotation ontologies in the Tabula Muris l0x lung data and human lung cells from the IPF Cell Atlas [Habermann et al., 2020]. We trained an scNym model using mouse data as the source domain and human data as the target, as for the mouse to rat demonstration.

Hyperparameter Optimization Experiments

We performed hyperparameter optimization across four tasks for the top two baseline methods, the SVM and scmap-cell-exact. For the SVM, we optimized the regularization strength parameter C at 12 values: $C \in 10^k \; \forall k \in [-6, 5]$. For scmap-cell-exact, we optimized (1) the number of nearest neighbors ($k \in \{5, 10, 30, 50, 100\}$), (2) the distance metric ($d(\cdot, \cdot) \in \{\text{cosine}, \text{euclidean}\}$), and (3) the number of features to select with M3Drop ($n_f \in \{500, 1000, 2000, 5000\}$). We optimized scNym for two of the four tasks, due to computational expense and superiority of default parameters relative to baseline methods. For scNym, we optimized (1) weight decay ($\lambda_w \in 10^{-5}, 10^{-4}, 10^{-3}$), (2) batch size ($M \in \{128, 256\}$), (3) the number of hidden units ($h \in \{256, 512\}$), (4) the maximum MixMatch weight ($\lambda_{\text{SSL}} \in \{0.01, 0.1, 1.0\}$), and (5) the maximum DAN weight ($\lambda_{\text{Adv}} \in \{0.01, 0.1, 0.2\}$). We did not optimize weight decay for the PBMC cross-stimulation task. We performed a grid search for all methods.

Hyperparameter optimization is non-trivial in the context of a domain shift between the training and test set. Traditional optimization using cross-validation on the training set alone may overfit parameters to the training domain, leading to suboptimal outcomes. This failure mode is especially problematic for domain adaptation models, where decreasing the strength of domain adaptation regularizers may improve performance within the training data, while actually decreasing performance on the target data.

In light of these concerns, we adopted a procedure known as reverse cross-validation to evaluate each hyperparameter set [Zhong et al., 2010]. Reverse cross-validation uses both the training and target datasets during training to account for the effect of hyperparameters on the effectiveness of transferring labels across domains. Formally, we first split the labeled training data $\mathcal{D}$ into a training set, validation set, and held-out test set $\mathcal{D}', \mathcal{D}^v, \mathcal{D}^*$. We use 10% of the training dataset for the validation set and 10% for the held-out test set. We then train a model $f_\theta : x \rightarrow \hat{y}$ to transfer labels from the training set $\mathcal{D}'$ to the target data $\mathcal{U}$. We use the validation set $\mathcal{D}^v$ for early stopping with scNym and concatenate it into the training set for other methods that do not use a validation set. We treat the predictions $\hat{y} = f_\theta(u)$ as pseudolabels for the unlabeled dataset and subsequently train a second model $f_\phi : u \rightarrow \tilde{y}$ to transfer annotations from the "pseudolabeled" dataset $\mathcal{U}$ back to the labeled dataset $\mathcal{D}$. We then evaluate the "reverse accuracy" as the accuracy of the labels $\tilde{y}$ for the held-out test portion of the labeled dataset, $\mathcal{D}^*$.

We performed this procedure using a standard 5-fold split for each parameter set. We computed the mean reverse cross-validation accuracy as the performance metric for robustness. For each method that we optimized, we selected the optimal set of hyperparameters as the set with the top reverse cross-validation accuracy.

Supplemental Figures

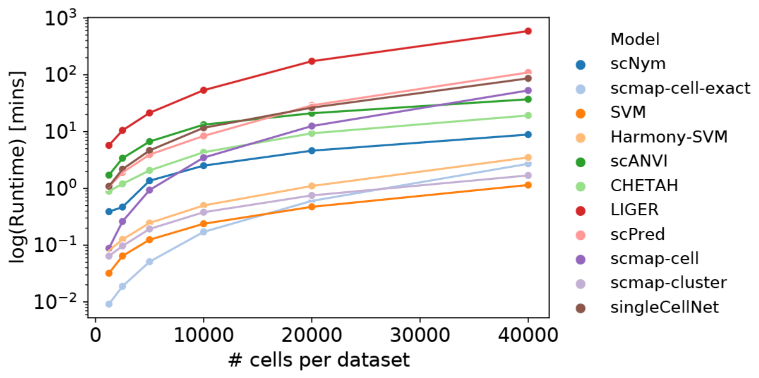


Figure S1: **scNym runtime performance is competitive with baseline methods.** We evaluated the runtime performance of all methods on datasets of increasing size. Each dataset was constructed using a subsample of n cells each for the training and target dataset from a mouse kidney single cell atlas. All methods were run on the same hardware (Intel Xeon CPU, 64 GB RAM, 1 Nvidia Titan RTX consumer GPU). Only scANVI and scNym are implemented to take advantage of GPU hardware. scNym runtime performance was competitive with baseline methods and completed even the largest annotation transfer task in roughly 10 minutes.

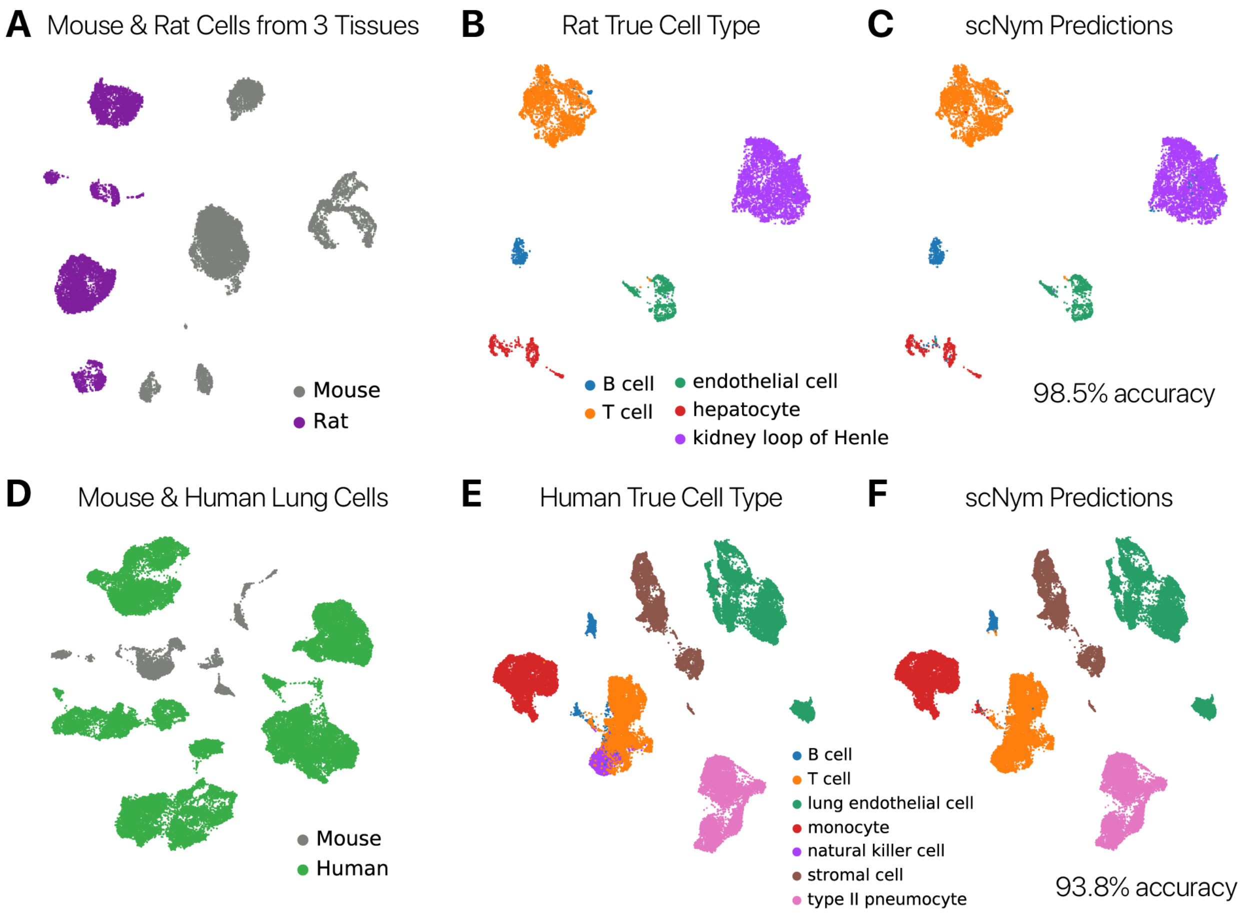


Figure S2: **scNym transfers cell type annotations across species.** We trained scNym to transfer annotations from the Tabula Muris to the Rat Aging Cell Atlas or Human IPF Cell Atlas. We subset each of the source and target datasets to cell types where the annotation ontologies were well-matched to allow for quantitative evaluation. **(A)** Mouse and rat cells from the kidney, liver, and spleen projected in a common latent space with UMAP. Species-specific and experimental differences lead to a segregation of mouse and rat data in the latent space. **(B)** Ground truth cell type labels for the rat data. **(C)** scNym predictions for rat cells after training on mouse cells. scNym predictions are highly accurate (98.5% accuracy). **(D)** Mouse and human cells from the lung segregate in a common latent space, reflecting species-specific differences. **(E)** Ground truth cell type labels for the human data. **(F)** scNym predictions for the human data are accurate for most cell types. Most errors are confusion between NK cells and T cells.

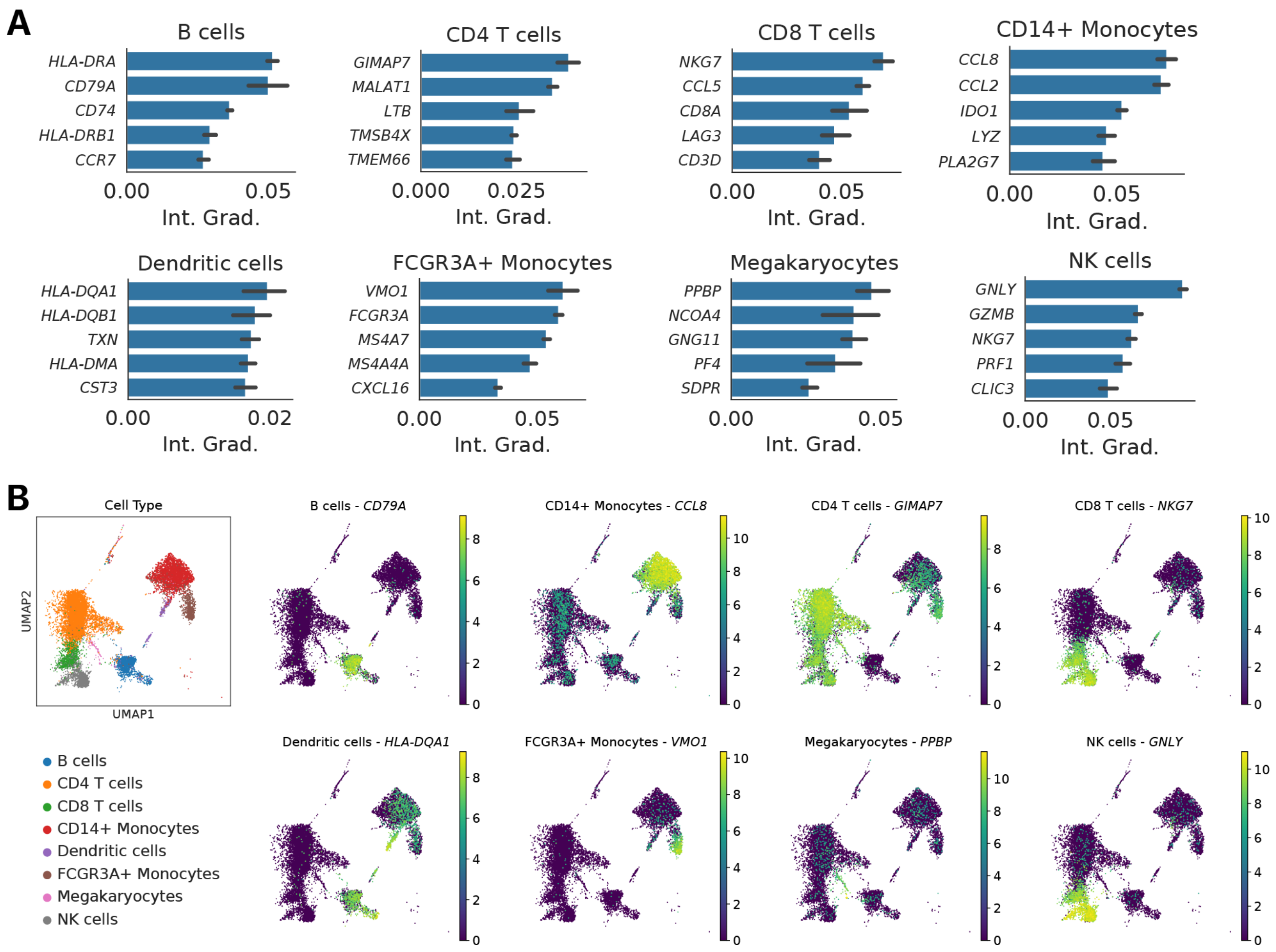


Figure S3: **Integrated gradient analysis of scNym models reveals genes that drive cell type classification decisions.** We used integrated gradient analysis to identify salient genes for each cell type in the human PBMC cross-stimulation task [Kang et al., 2017]. **(A)** Top 5 salient genes for each cell type, computed as the mean across a set of 300 correctly classified cells. **(B)** Expression of the most salient gene for each cell type in the UMAP embedding.

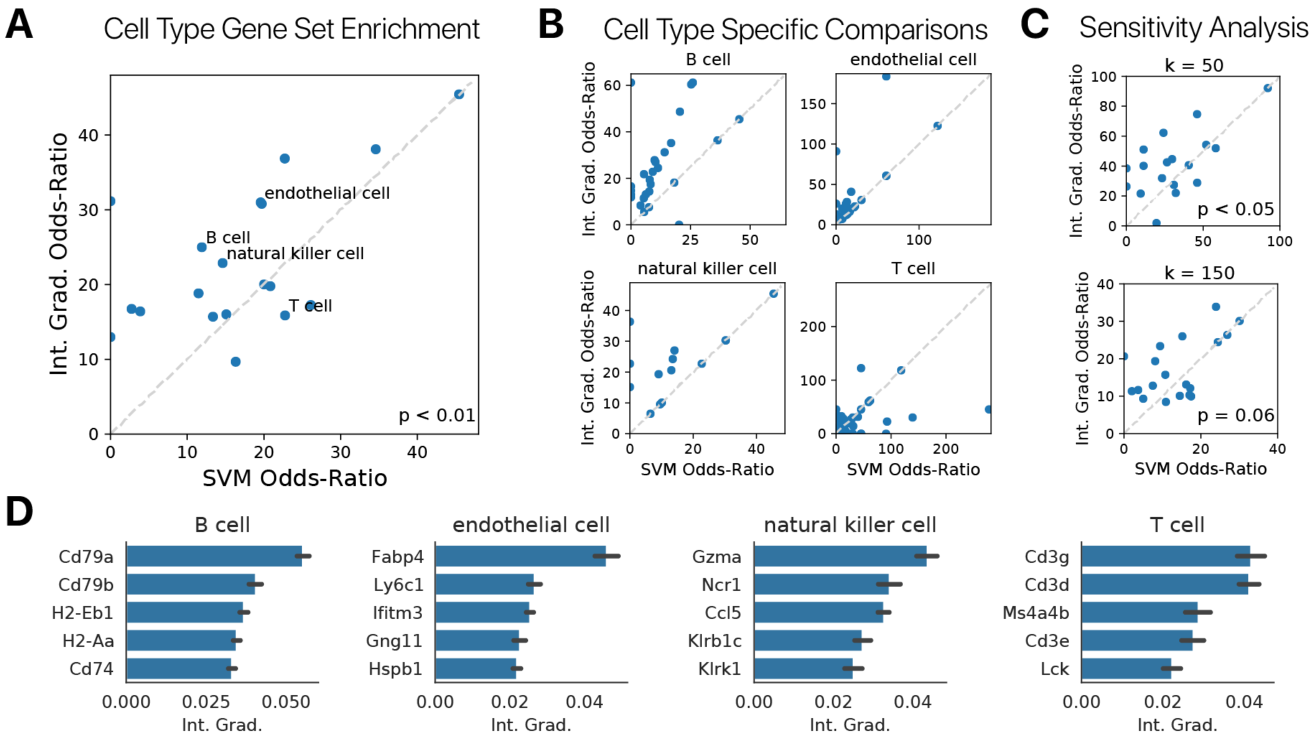
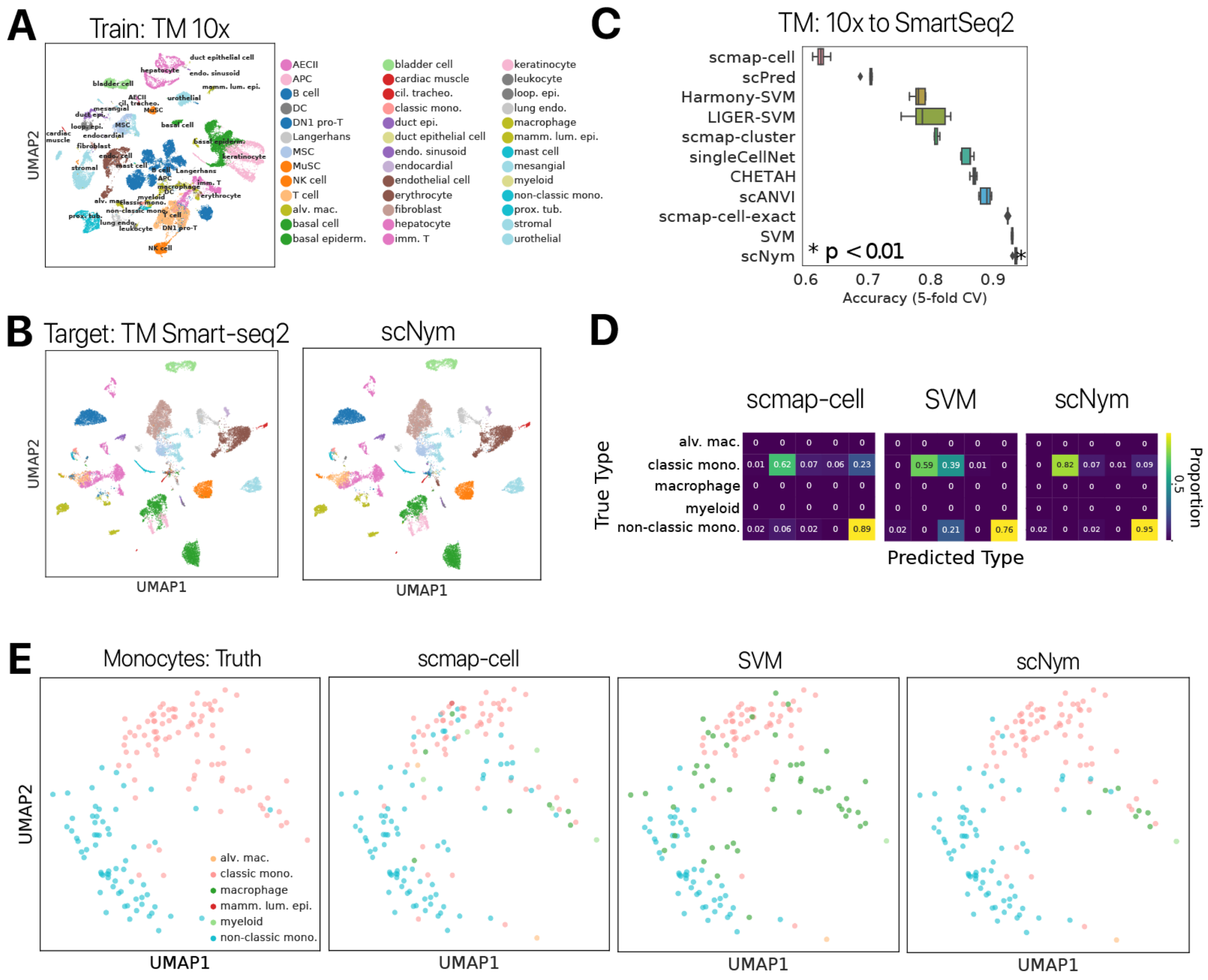


Figure S4: **scNym integrated gradient feature attribution reveals cell type markers more effectively than SVM coefficients.** We computed integrated gradients for an scNym model and extracted SVM coefficients for 19 cell types after training both models on the Tabula Muris 10x Genomics dataset. Cell types were chosen based on the availability of cell type specific Gene Ontology terms. We computed enrichment scores within cell type specific GO terms using the top $k = 100$ ranked genes for each cell type from each method (Methods). **(A)** The mean enrichment score (Odds-Ratio) across cell type specific Gene Ontology terms for the 19 cell types. Most cell types showed stronger enrichments using integrated gradient markers (above grey line), and scNym integrated gradients yielded significantly stronger enrichments overall (paired t-test for Odds-Ratios across cell types, $p = 0.011$). scNym models are therefore more readily interpretable than the best baseline method. **(B)** Odds-Ratios for individual Gene Ontology terms in a subset of cell types. Each point is the Odds-Ratio for a single Gene Ontology term specific to a single cell type. scNym models show stronger enrichments for most relevant Gene Ontology terms in most cell types. **(C)** We performed a sensitivity analysis to evaluate the effect of the marker gene set size k on our comparative Gene Ontology enrichments. scNym integrated gradients showed stronger enrichments than SVM coefficients for smaller ($k = 50$) and larger ($k = 150$) gene set sizes. **(D)** Top five cell type markers extracted using scNym integrated gradients for a subset of cell types. Integrated gradients capture well-characterized cell type markers across a range of cell types (e.g. *Cd79a/b* in B cells, *Fabp4* and *Ly6c1* in endothelial cells, *Cd3d/g* in T cells).



True Type	scmap-cell					SVM					scNym				
alv. mac.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
classic mono.	0.01	0.62	0.07	0.06	0.23	0	0.59	0.39	0.01	0	0	0.82	0.07	0.01	0.09
macrophage	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
myeloid	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
non-classic mono.	0.02	0.06	0.02	0	0.89	0.02	0	0.21	0	0.76	0.02	0	0.02	0	0.95

Figure S5: **scNym transfers cell identity annotations between 10x and Smart-seq2 sequencing technologies in the Tabula Muris. (A)** UMAP projections of ground truth cell type labels in the *Tabula Muris* 10x training data. We limited our analyses to tissues with similar cell type labels between the two technologies: Bladder, Heart and Aorta, Kidney, Limb Muscle, Liver, Lung, Mammary Gland, Spleen, Thymus, and Tongue. **(B)** Ground truth labels and scNym predictions overlaid on a UMAP projection. scNym labels match ground truth for the large majority of cells (>92%). **(C)** Comparison of classification models on the 10x to Smart-seq2 transfer task. scNym models are significantly more accurate than baseline methods (Wilcoxon Rank Sums on accuracy scores, $p < 0.01$). **(D)** Confusion matrices for scmap-cell, SVM, and scNym on cells with a ground truth annotation containing "monocyte". scmap-cell confused classical monocytes and non-classical monocytes, while an SVM confuses monocytes with macrophages. scNym shows less confusion overall. **(E)** Ground truth annotations (left), scmap-cell predictions (left center), SVM predictions (right center) and scNym predictions (right) for monocytes in the Tabula Muris. scNym more accurately labels both types of monocytes.

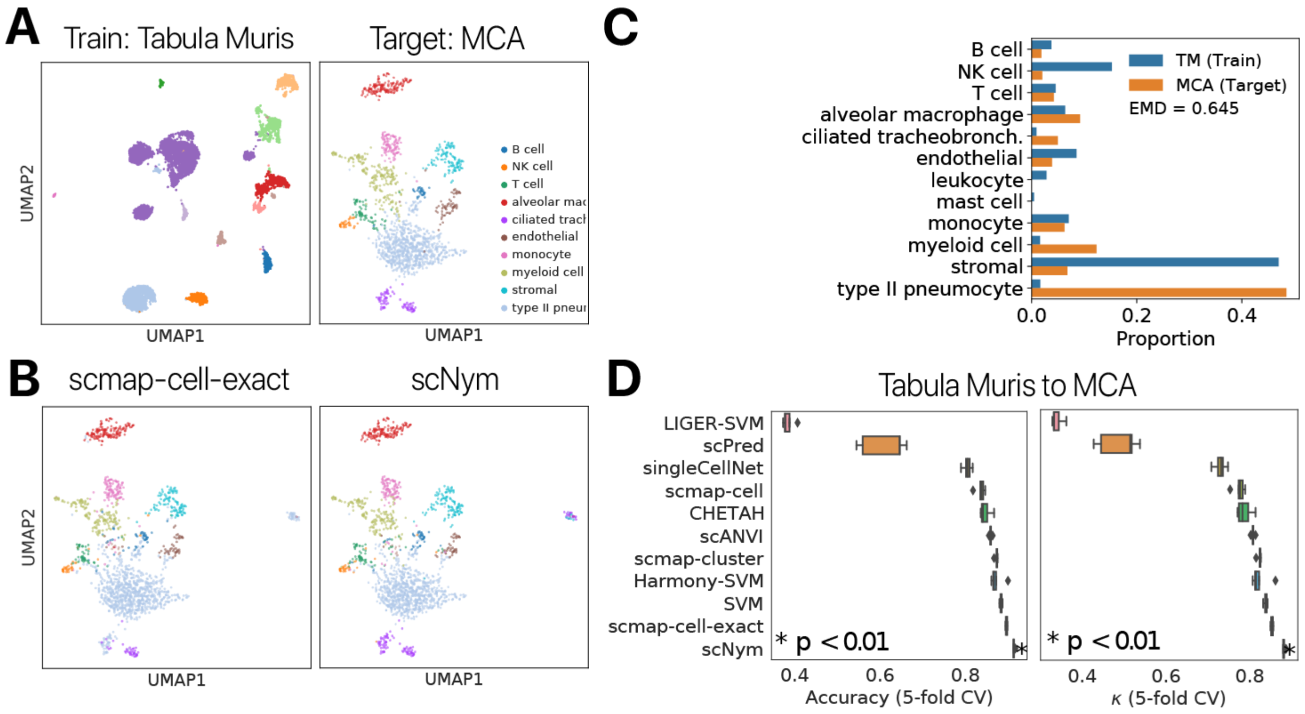


Figure S6: **scNym transfers cell identity annotations in the mouse lung between 10x Chromium data from the Tabula Muris and Microwell-seq data from the Mouse Cell Atlas.** **(A)** Ground truth annotations for the Tabula Muris training data and MCA target data are displayed in a UMAP projection. **(B)** Predictions from scNym and the next best model scmap-cell-exact are likewise shown in the UMAP projection. scNym predictions are highly concordant with ground truth labels. **(C)** Comparison of cell type distributions between the training set (Tabula Muris, TM) and the target set (Mouse Cell Atlas, MCA). There is a large class imbalance between the two data sets. We quantified this imbalance using the Earth Mover Distance (EMD) and found the EMD to be roughly 0.65, indicating the majority of probability mass would need to shift to match the distributions. Despite this class imbalance, scNym classifiers achieved high performance. **(D)** Accuracy and κ score comparison of scNym to seven other cell type classifiers. scNym models show significantly greater performance than the baseline methods (Wilcoxon Rank Sums on accuracy scores, $p < 0.01$).

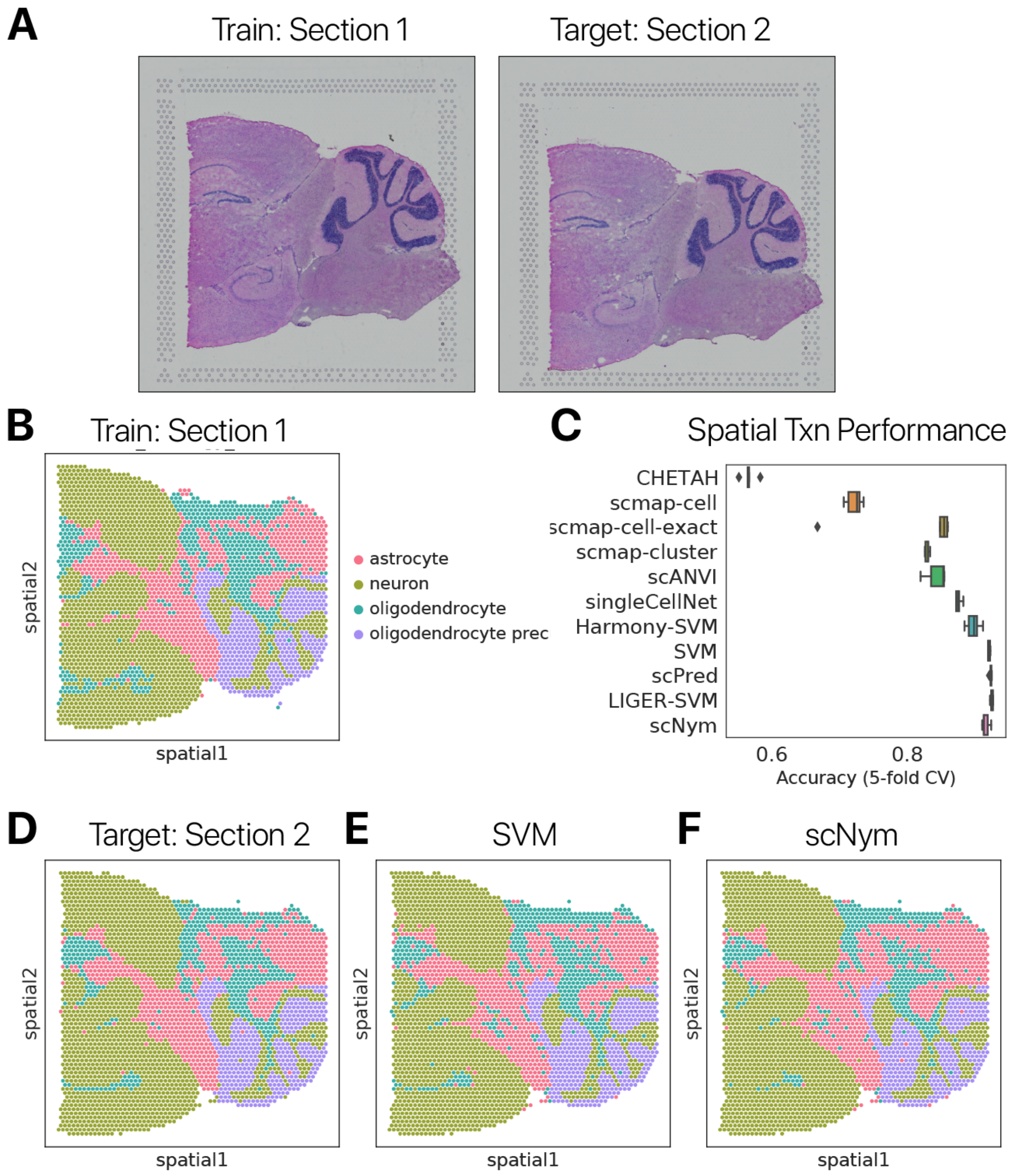


Figure S7: **scNym transfers regional identity annotations across spatial transcriptomics data sets.** **(A)** Images of training and target mouse brain sagittal-posterior sections profiled with spatial transcriptomics. **(B)** Ground truth regional identity annotations for the training section. **(C)** Accuracy score comparison of scNym to seven other state-of-the-art cell type classifiers. scNym performs competitively relative to baseline methods, modestly less accurate than scPred and SVM approaches. **(D)** Ground truth regional identity annotations in the target section, alongside **(E)** SVM predictions and **(F)** scNym predictions.

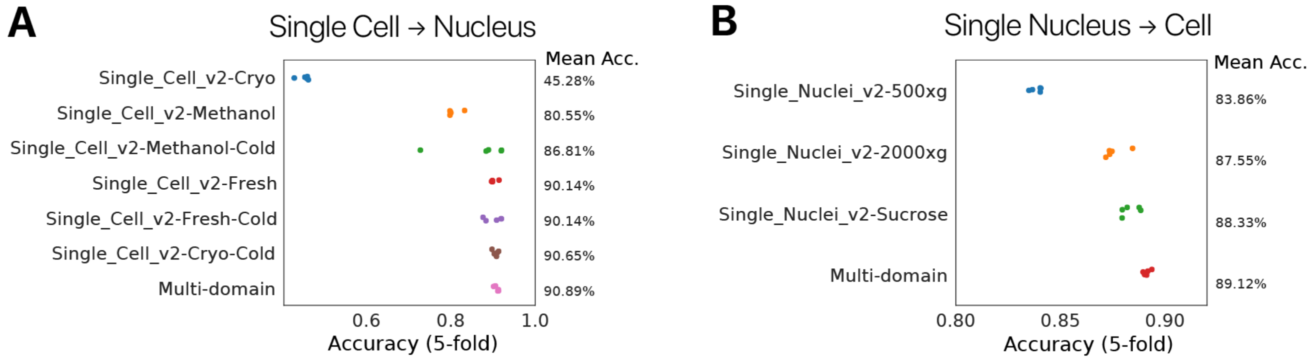


Figure S8: **Multi-domain training improves performance of cross-technology annotation transfer in the mouse kidney.** We trained scNym using each training domain individually or all domains through multi-domain training for both a single cell to single nucleus and single nucleus to single cell annotation transfer task. Distinct domains represent different sample preparation protocols. **(A)** Performance of different training approaches for the single cell to single nucleus task. Multi-domain models outperform models trained on any single domain. In a real world scenario, the benefit of multi-domain training is likely greater than the difference between the best single domain and multi-domain performance, as the best single domain cannot be determined *a priori*. **(B)** Similar results were obtained for single nucleus to single cell transfer tasks. Here, the benefit of multi-domain training is larger than in the single cell to single nucleus case (Wilcoxon Rank Sum $p < 0.01$ for Multi-domain vs. Single Nuclei v2 - Sucrose).

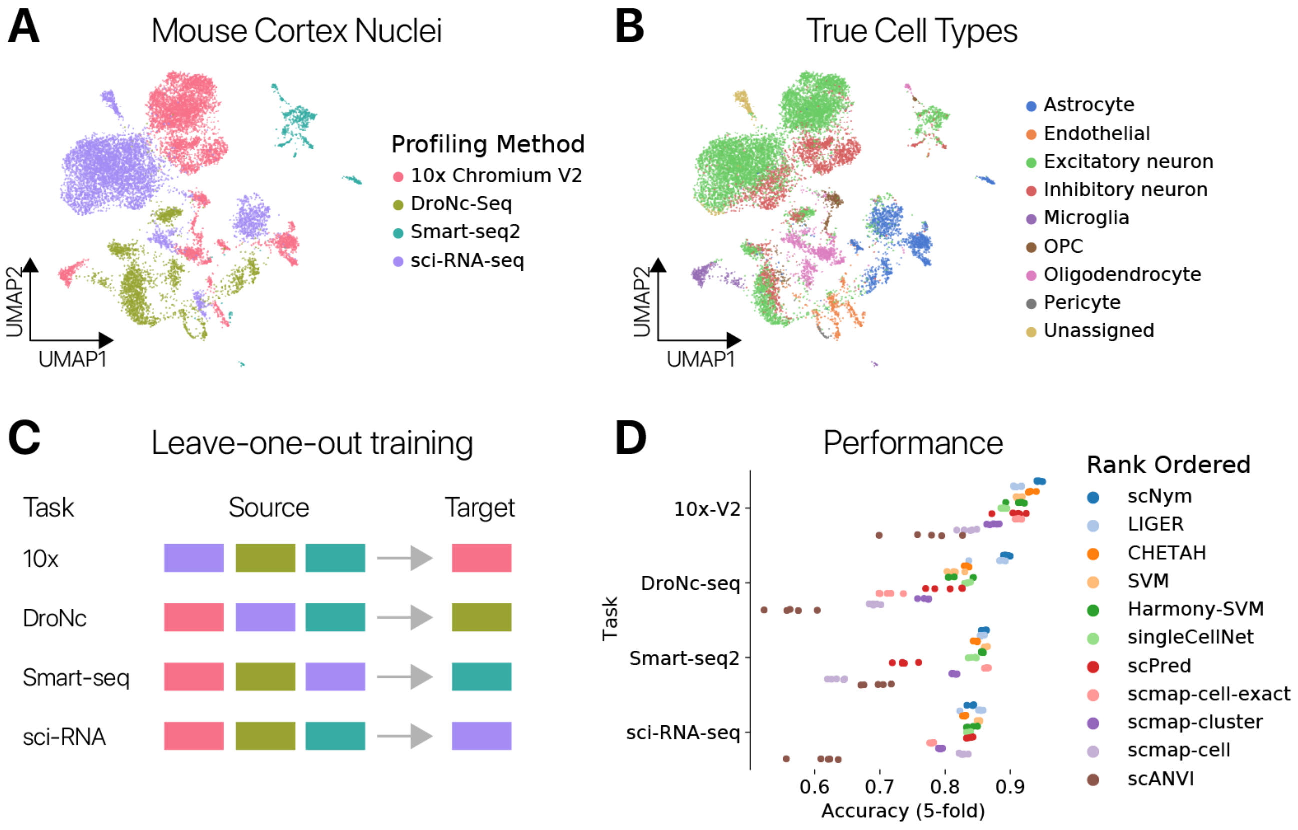


Figure S9: **scNym synthesizes information from multiple domains to transfer annotations across technologies in mouse cortex nuclei.** We trained scNym to transfer annotations in mouse cortex nuclei profiled with distinct technologies. **(A)** Mouse cortex nuclei profiled with four distinct sequencing technologies embedded in a latent space with UMAP. **(B)** True cell type annotations provided for the mouse cortex nuclei dataset. **(C)** Schematic representation of leave-one-out training procedure. We trained scNym using 3/4 domains at a time as source data, and the remaining domain as the target data in a leave-one-out fashion. This procedure yields four distinct transfer tasks. **(D)** scNym and baseline performance across transfer tasks, where the tasks are named based on the held-out target dataset. scNym offered superior performance to baselines on one task and was competitive for the remaining tasks. scNym was the top method after rank ordering by mean accuracy or the mean rank across tasks.

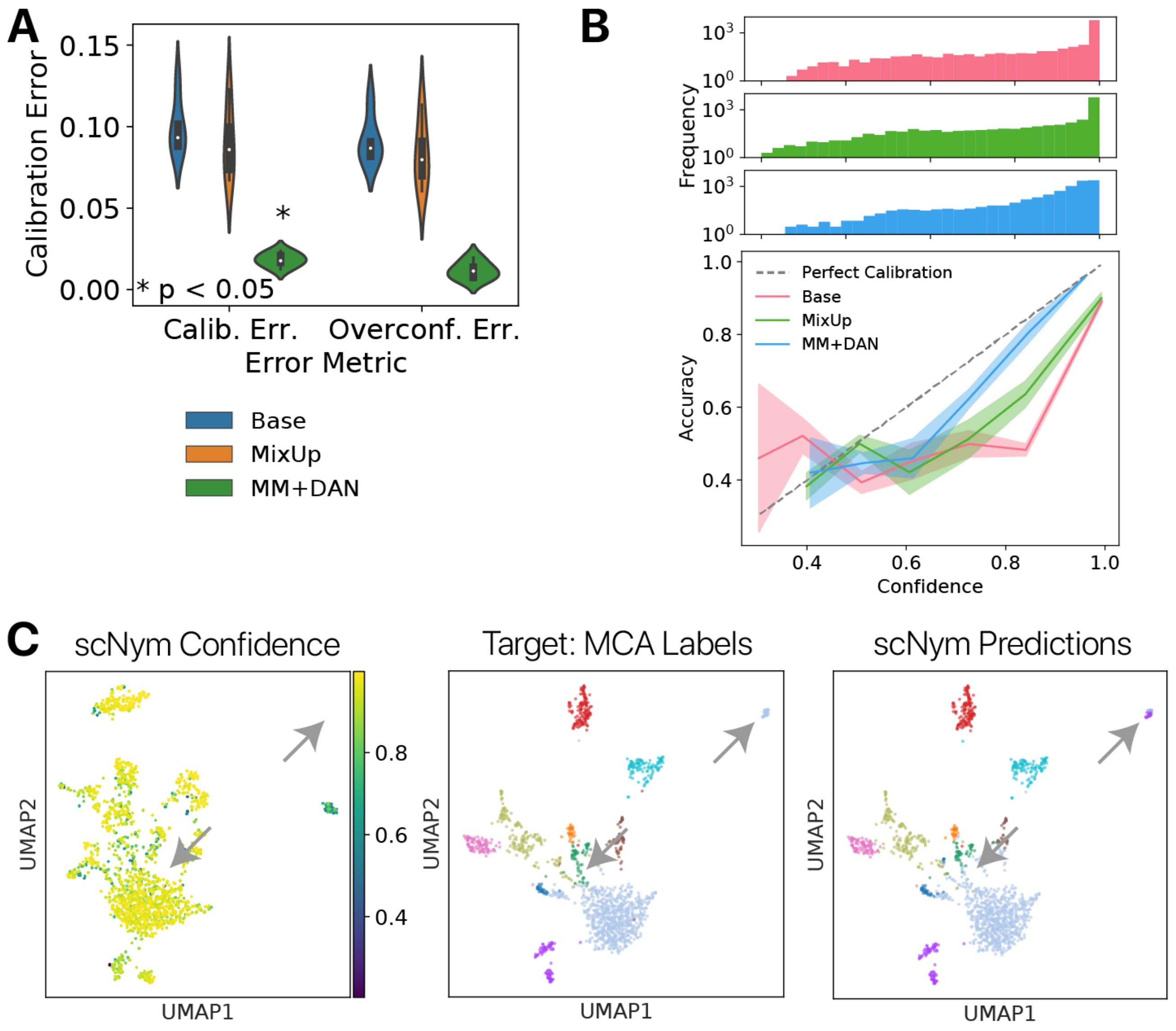


Figure S10: **Semi-supervision improves the calibration of scNym models.** **(A)** Expected calibration error (ECE) and overconfidence error (OE) scores for scNym trained on the Tabula Muris to Mouse Cell Atlas benchmark without MixUp or MixMatch (Base), with MixUp on labeled data only (MixUp), or with the full MixMatch (MM) and domain adversarial network (DAN) procedure (MM + DAN). The MixMatch + DAN procedure significantly reduces the expected calibration error ($p < 0.05$, Wilcoxon Rank Sums). **(B)** Calibration curves for different scNym model configurations comparing the model confidence scores to observed classification accuracies. scNym models with MixUp, MixMatch, and DAN ablated ("Base") show overconfident calibration at the high end. Confidence frequency histograms for each of the model configurations are shown above. **(C)** UMAP projection of model confidence scores, ground truth cell labels, and scNym predicted labels. We observed that incorrect predictions have lower confidence scores (arrows).

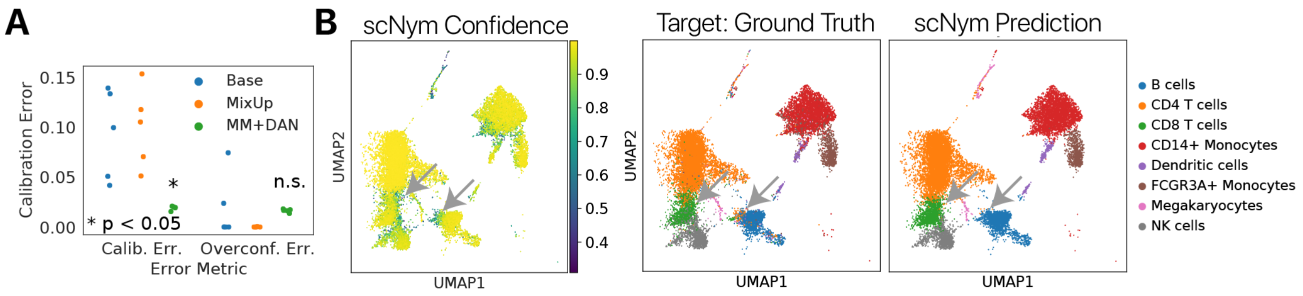


Figure S11: **scNym confidence highlights cells for manual curation.** **(A)** Expected calibration error (ECE) and overconfidence error (OE) scores for scNym models trained on the cross-stimulation benchmark without MixUp or MixMatch (Base), with MixUp on labeled data only (MixUp), or with the full MixMatch and domain adversarial network (DAN) procedure (MM + DAN). The MixMatch + DAN procedure significantly reduces the expected calibration error ($p < 0.05$, Wilcoxon Rank Sums). **(B)** Confidence of the scNym model predictions projected in the UMAP embedding, alongside ground truth cell type annotations and scNym predictions. Low confidence scores identify incorrectly classified cells for manual curation (arrows).

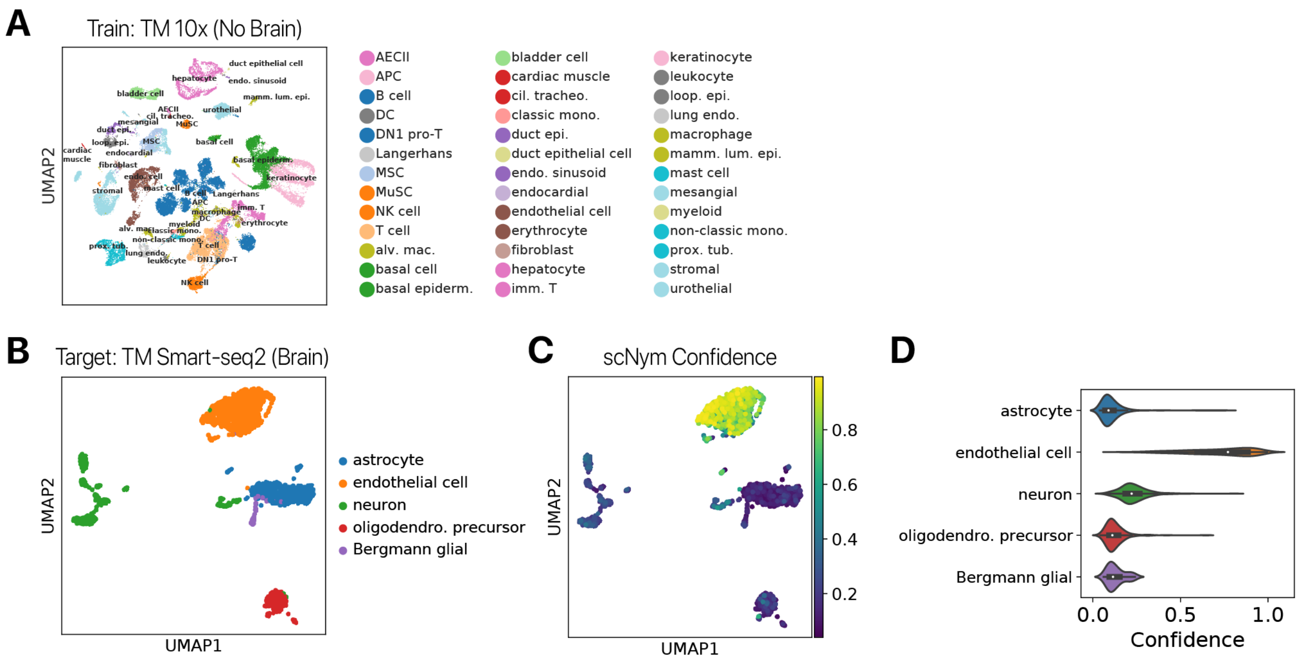


Figure S12: **Confidence scores from pre-trained scNym models highlight new cell type discoveries.** We trained scNym to transfer labels from the Tabula Muris 10x cells to the Smart-seq2 cells, excluding brain tissue from the training and target sets. We subsequently predicted cell types for cells in the Smart-seq2 Brain (Non-Myeloid) tissue. **(A)** UMAP projections of ground truth cell type labels in the *Tabula Muris* 10x training data. Note that the training data does *not* contain any neurons, astrocytes, or other glia. **(B)** Ground truth cell type annotations in the Tabula Muris brain tissue captured using Smart-seq2. Note that astrocyte, neuron, olidodendro. precursor, and Bergmann glia represent new, unseen cell types. **(C)** scNym confidence scores projected for the Tabula Muris brain data. New cell types receive low confidence scores, while endothelial cells are correctly predicted with high confidence because they are present in the training data. **(D)** Distribution of confidence scores in **(C)** for each ground truth cell type.

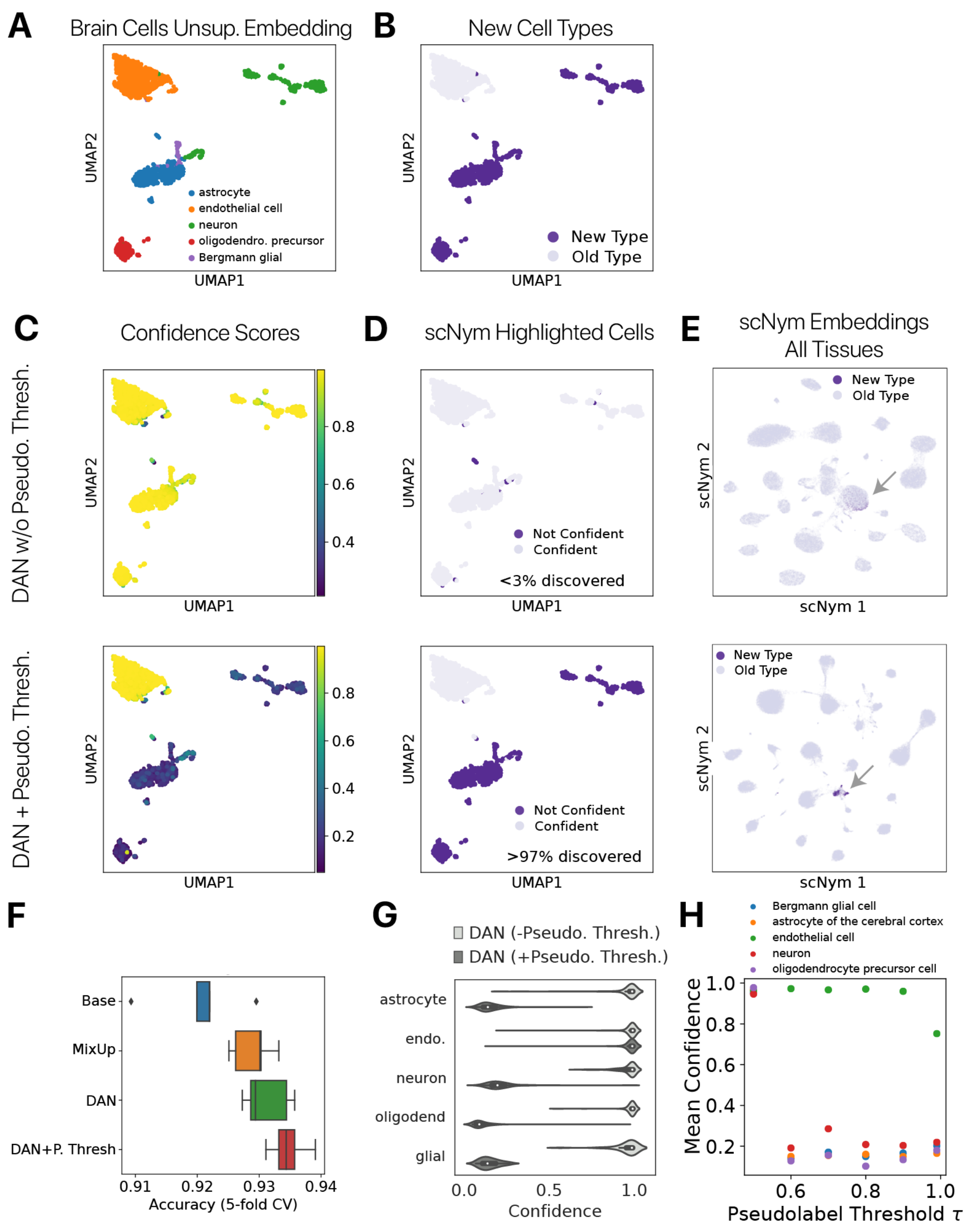


Figure S13: **scNym confidence scores highlight new cell type discoveries after semi-supervised training.** We trained scNym with and without pseudolabel thresholding to transfer labels from the Tabula Muris 10x cells to the Smart-seq2 cells, *including* Brain tissue in the target set but not the training set. **(A)** Ground truth cell type annotations in the Tabula Muris brain tissue captured using Smart-seq2. Note that astrocyte, neuron, olidodendro. precursor, and Bergmann glia represent new, unseen cell types. **(B)** New cell types "discovered" in this simulated experiment are highlighted in purple. **(C)** scNym confidence scores displayed in the UMAP projection. **(D)** Cells highlighted as potential discoveries by thresholding scNym confidence scores (< 0.5). Models with pseudolabel thresholding correctly identify the majority ($> 97\%$) of cells with novel types. **(E)** Embeddings from scNym's penultimate layer for all target tissues, projected with UMAP. Cell types not in the training set are labeled. Pseudolabel thresholding allows new cell types to occupy a distinct region in the embedding (arrows). **(F)** scNym models' cell type prediction accuracies. Pseudolabel thresholding improves performance when new cell types are present. All predictions for new cell types are counted as incorrect for this accuracy metric. **(G)** Distribution of confidence scores for each ground truth cell type for scNym models. Models trained with pseudolabel thresholding correctly provide low confidence scores to new cell types, while models without pseudolabel thresholding incorrectly provide uniformly high confidence. **(H)** Average confidence for each cell type across pseudolabel threshold values τ. Results are robust to a wide range of τ values, with a critical effectiveness range for $\tau > 0.5$.

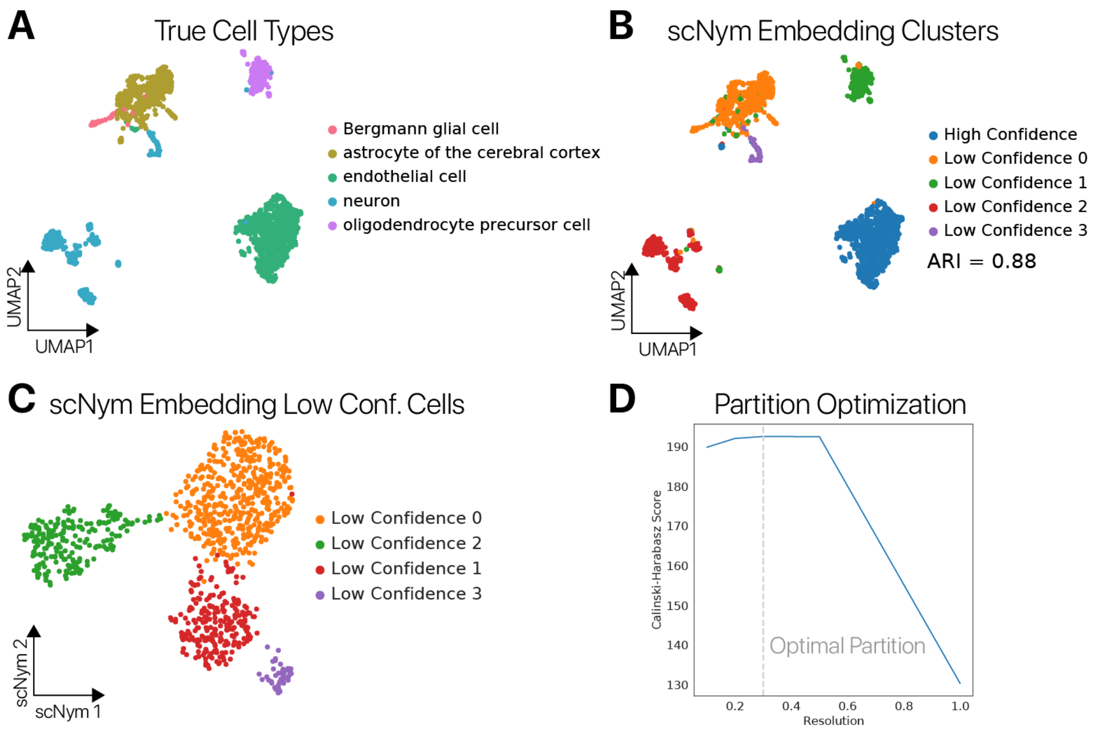


Figure S14: **Clustering low confidence cells in the scNym embedding reveals new distinct cell states.** We developed a clustering procedure in the scNym embedding to describe the distinct cell states within a set of low confidence cells. We applied this procedure to brain cells in the Tabula Muris Smart-seq2 data, which are a new cell type not found in the training set. **(A)** True cell type labels for the Tabula Muris Smart-seq2 brain data. All but the "endothelial cell" class are new cell types not present in the training data. **(B)** New cell state clusters suggested as a result of our clustering procedure. We found these clusters closely matched the true cell type labels – unseen by the scNym model – with an adjusted Rand index (ARI) of 0.88. **(C)** scNym embedding of low confidence cells, projected with UMAP. Low confidence cells form four distinct clusters in the embedding. **(D)** Cluster optimization results for cells in **(C)**. We evaluated the Calinski-Harabasz cluster validity index for a range of Leiden community detection resolutions and selected the top scoring resolution (grey line).

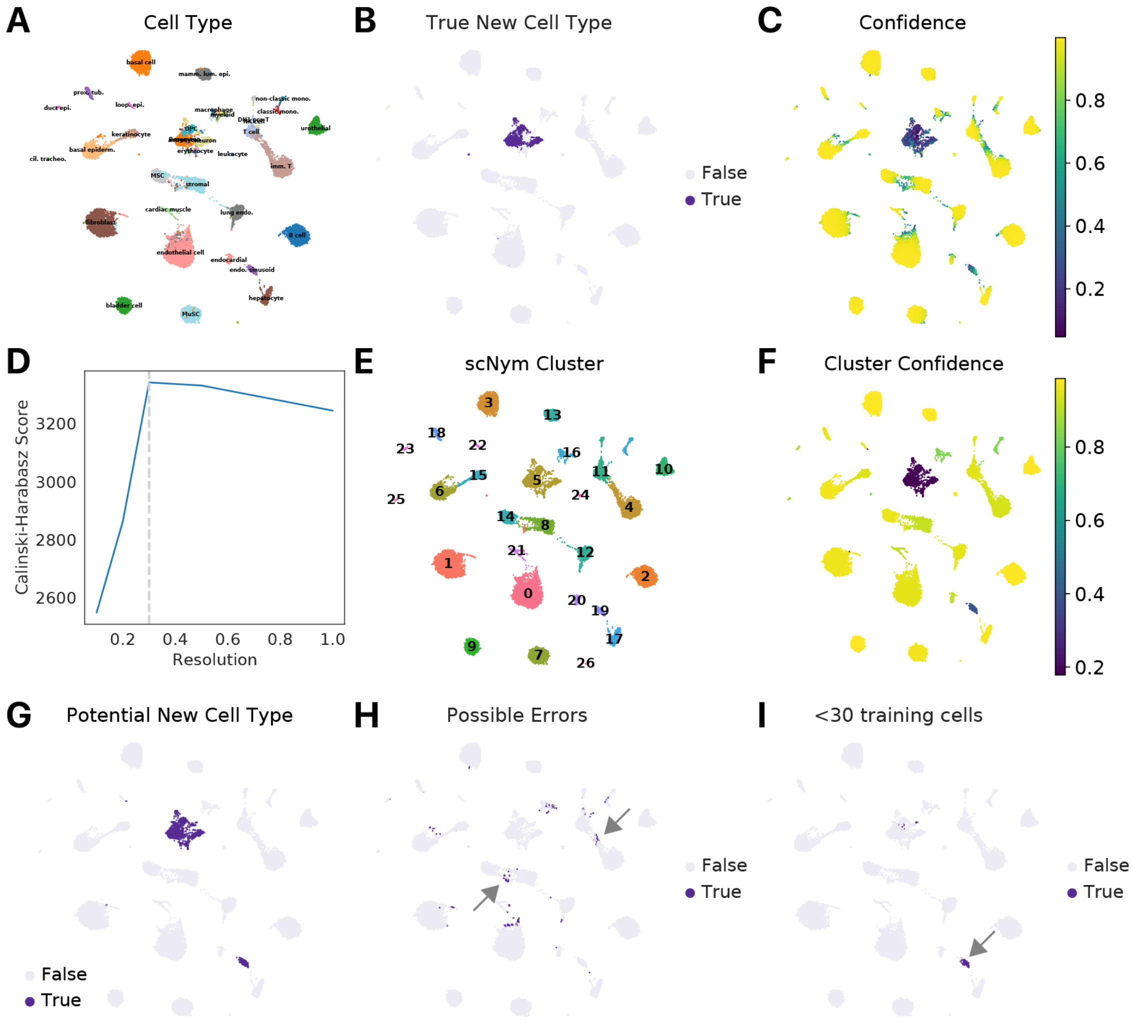


Figure S15: **Clustering in the scNym embedding discriminates potential new cell types from low confidence predictions on cell type boundaries.** We developed a clustering procedure using the scNym embedding for target cells to generate a metric of average confidence per cluster. We found that low confidence clusters represented either new cell types, or previously seen cell types with very little training data. Low confidence cells in high confidence clusters represented boundaries between cell types. **(A)** Cell type labels and **(B)** new cell types highlighted for the Tabula Muris Smart-seq2 data in the scNym embedding projected with UMAP. **(C)** scNym prediction confidence across cells. **(D)** Leiden clustering results for cells in **(A)** in the scNym embedding across a range of resolution parameters. **(E)** Optimal cluster partition in the scNym embedding. **(F)** Average confidence per cluster. **(G)** Clusters with low average confidence may represent new cell types. Here, we found that new cell types all had low cluster confidence. **(H)** Cells with low confidence within a high confidence cluster appear to represent cell type boundaries (arrows). **(I)** One cell type had low cluster confidence, even though it was not a new cell type. We found that this cell type had very few cells in the training data (<30 cells), suggesting another scenario that might lead to low cluster confidence.

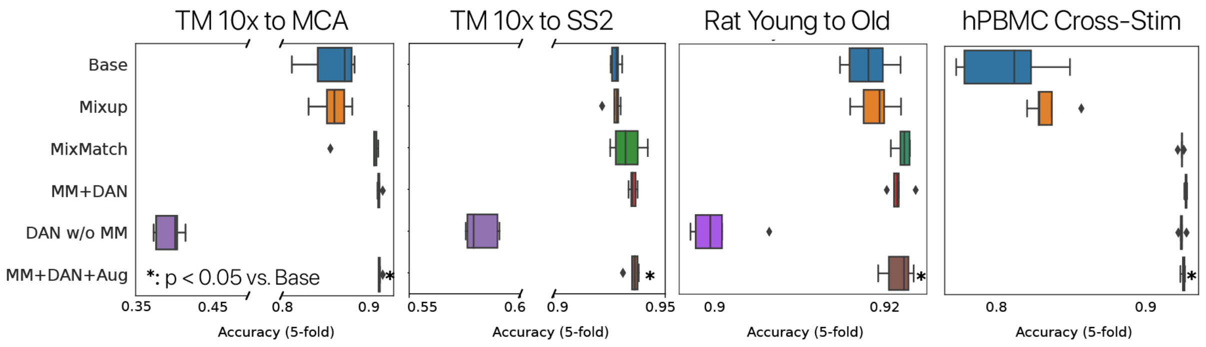


Figure S16: **Ablation experiments demonstrate that semi-supervision improves scNym performance.** We trained scNym models with and without semi-supervision components to complete cell annotation transfer tasks. In each case, we found that inclusion of MixUp augmentations, MixMatch training, and a domain adversary significantly improved scNym performance relative to a base model ($p < 0.05$, Rank Sums test).

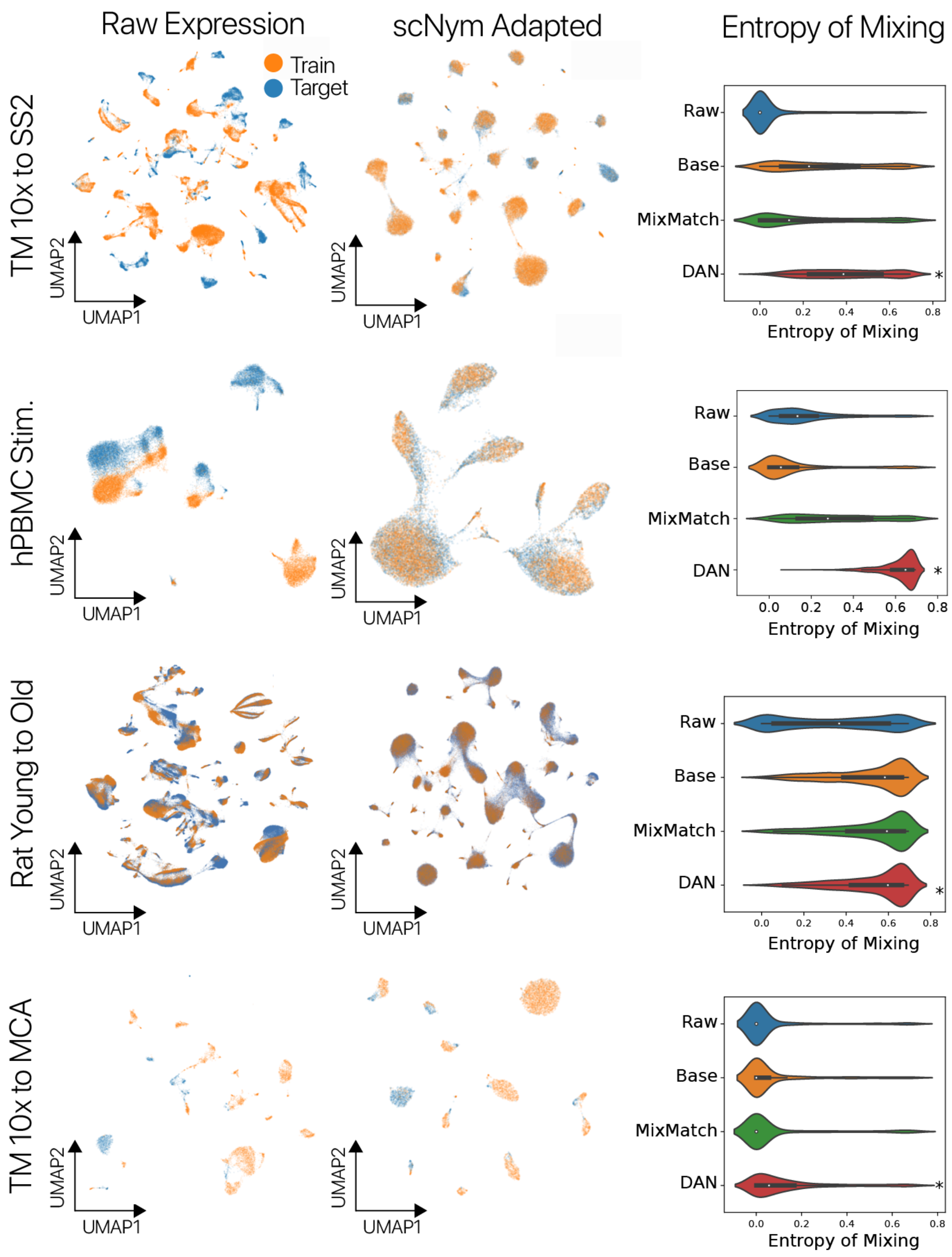


Figure S17: **Adversarial scNym models adapt training and target domains.** We trained scNym models with MixMatch semi-supervision and a domain adversary for each of our benchmark tasks. Raw gene expression embeddings derived using PCA and UMAP on normalized gene expression counts are shown on the left for each benchmark dataset. We have labeled the target and training domains with colors. In the center column, scNym embeddings from the penultimate neural network layer are projected using PCA and UMAP. Qualitatively, we found that training and target datasets are more intermixed in the scNym embeddings. We confirmed this observation quantitatively by computing the entropy of batch mixing (right) on raw gene expression embeddings (Raw, left column embeddings), base scNym model embeddings (Base), scNym models with MixMatch but no domain adversary (MixMatch), and our complete scNym model with MixMatch and a domain adversary (DANN, center column embeddings). scNym models with MixMatch and a domain adversary were significantly more mixed than raw gene expression embeddings for each benchmark task ($p < 0.05$ Wilcoxon Rank Sums). This indicates that scNym models are effectively integrating datasets to adapt across domains.

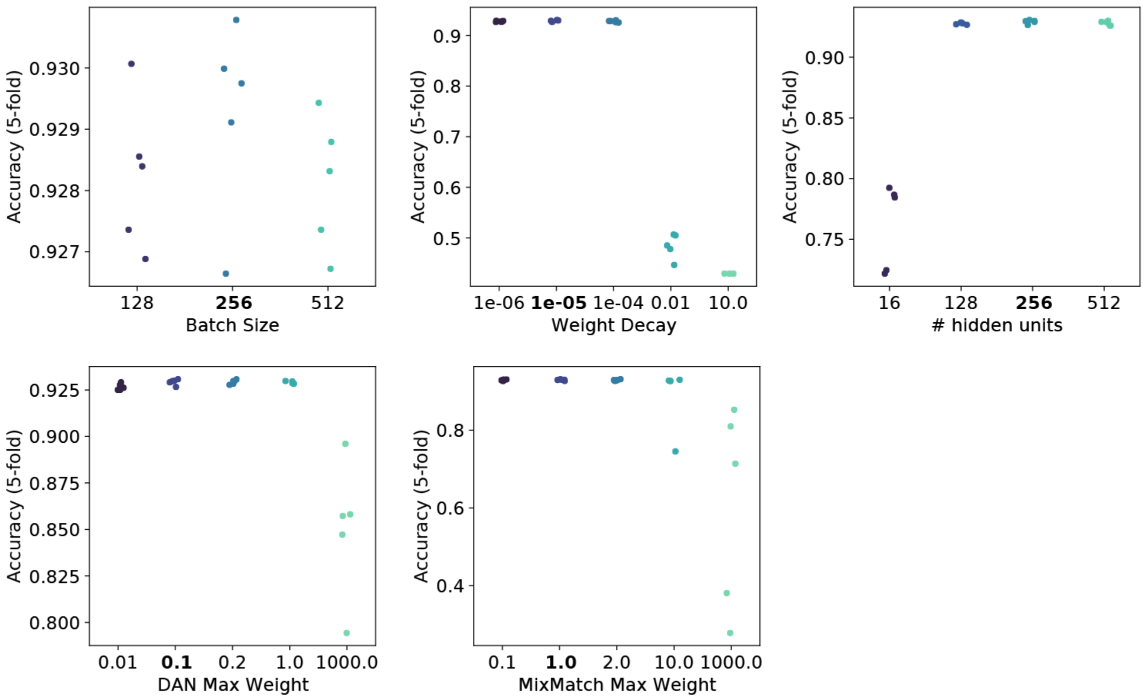


Figure S18: **scNym performance is robust across a range of hyperparameter values.** We evaluated the sensitivity of scNym to its hyperparameters by varying their values across a broad range for the human PBMC cross-stimulation task. For each panel, we held all but one hyperparameter constant at the default values and varied the parameter on the x-axis. Bolded values on the x-axis are the default parameter settings. We found that performance was robust to changes within an order of magnitude, suggesting that our default hyperparameters are not "overfit" to the tasks we use for benchmarking.

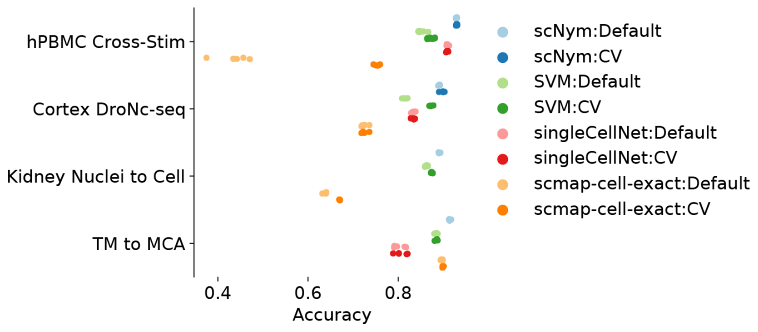
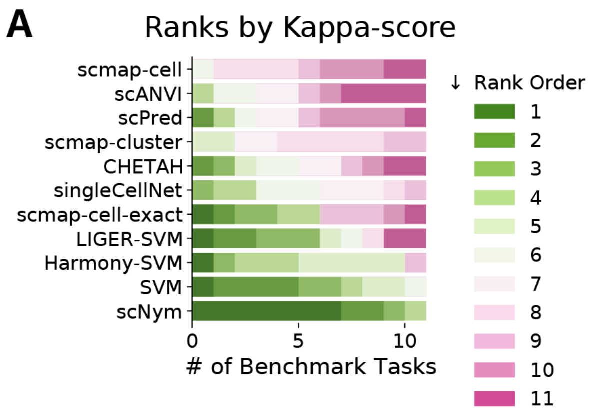


Figure S19: **scNym models with default hyperparameters are superior to baseline methods with tuned hyperparameters.** We optimized hyperparameters for the top three baseline models (SVM, singleCellNet, scmap-cell-exact) across four tasks using reverse 5-fold cross-validation. We optimized scNym hyperparameters for two of these tasks. All models were optimized using a grid search strategy. Across all four tasks, scNym with default parameters is superior to the baseline methods with optimized hyperparameters.



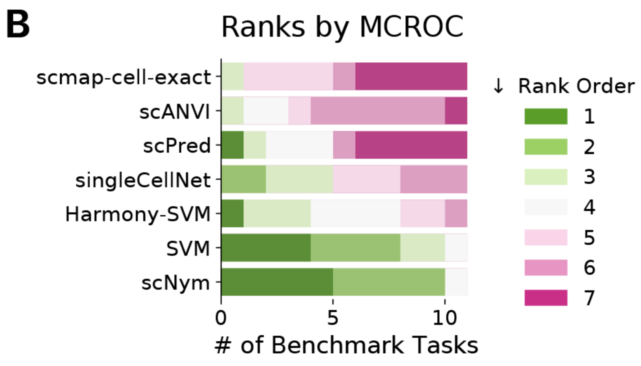


Figure S20: **scNym and baseline approaches ranked by alternative performance metrics. (A)** Models ranked by Cohen's κ-score. Model ranking is identical to the ranking derived from accuracy scores. **(B)** Models ranked by multi-class receiver operating characteristic (MCROC) score. We computed MCROC as the mean ROC score across cell types, treating each cell type as a binary classification problem. We could only generate MCROC scores for models that offer probabilistic outputs or a reasonable approximation (e.g. Platt Scaling for SVMs). Model ranking varies from the accuracy and κ-score ranks, but scNym remains the top ranked model and the SVM remains the top baseline method.

	scmap-cell	scmap-cell-exact	scmap-cluster	SVM	singleCellNet	scPred	CHETAH	Harmony-SVM	LIGER-SVM	scANVI	scNym
Young to Old Rat	0.838	0.864	0.777	0.912	0.885	OOM	0.695	0.859	OOM	0.812	0.917
hPBMC Cross-Stim	0.513	0.206	0.744	0.806	0.875	0.488	0.488	0.886	0.889	0.779	0.901
TM 10x to MCA	0.775	0.853	0.823	0.838	0.729	0.488	0.787	0.823	0.340	0.807	0.882
TM 10x to SS2	0.599	0.918	0.796	0.926	0.849	0.682	0.860	0.768	0.781	0.881	0.931
Spatial Txn	0.557	0.717	0.758	0.882	0.815	0.886	0.407	0.849	0.888	0.778	0.875
Kidney Cell to Nuc	0.704	0.856	0.718	0.839	0.808	0.551	0.810	0.884	0.865	0.240	0.873
Kidney Nuc to Cell	0.586	0.417	0.745	0.791	0.713	0.735	0.184	0.754	0.762	0.222	0.832
Cortex SS2	0.234	0.791	0.726	0.790	0.762	0.574	0.771	0.782	0.777	0.597	0.790
Cortex 10x	0.732	0.867	0.805	0.866	0.827	0.855	0.897	0.865	0.864	0.699	0.917
Cortex DroNc	0.504	0.606	0.684	0.725	0.760	0.706	0.765	0.743	0.821	0.500	0.851
Cortex sci-seq	0.640	0.511	0.615	0.732	0.703	0.717	0.698	0.709	0.718	0.463	0.728

Table S1: **Comparison of model performance across tasks using Cohen's κ-score.** Mean κ-score across 5-fold training split is reported. OOM indicates the model failed with an out-of-memory error on our hardware (256GB RAM).

	scmap-cell-exact	SVM	singleCellNet	scPred	Harmony-SVM	scANVI	scNym
Young to Old Rat	0.939	0.996	0.995	OOM	0.990	0.920	0.995
hPBMC Cross-Stim	0.703	0.979	0.909	0.909	0.986	0.973	0.982
TM 10x to MCA	0.933	0.984	0.927	0.903	0.973	0.934	0.987
TM 10x to SS2	0.955	0.990	0.982	0.928	0.973	0.939	0.988
Spatial Txn	0.942	0.982	0.977	0.979	0.973	0.980	0.981
Kidney Cell to Nuc	0.901	0.920	0.931	0.786	0.925	0.801	0.968
Kidney Nuc to Cell	0.687	0.842	0.843	0.838	0.822	0.702	0.866
Cortex SS2	0.968	0.973	0.957	0.930	0.965	0.961	0.968
Cortex 10x	0.984	0.994	0.990	0.992	0.991	0.979	0.994
Cortex DroNc	0.872	0.926	0.913	0.916	0.919	0.911	0.927
Cortex sci-seq	0.851	0.898	0.894	0.900	0.891	0.878	0.892

Table S2: **Comparison of model performance across tasks using the multi-class area under the Receiver Operating Characteristic (MCROC).** We computed the area-under the receiver operating characteristic curve for each cell type, treating the classification of each cell type as a binary classification problem. We then computed the multi-class ROC as the mean of ROC scores across cell types. MCROC could only be computed for models that offered a probabilistic output or where an established procedure for generating probabilistic predictions was available (e.g. Platt Scaling for SVM). Mean MCROC-score across 5-fold training split is reported. OOM indicates that a model failed with an out-of-memory error on our hardware (256GB RAM).

	Cortex DroNc-seq	Kidney Nuclei to Cell	TM to MCA	hPBMC Cross-Stim
scNym:Default	89.13	89.12	91.40	92.89
scNym:CV	89.85	Not Tested	Not Tested	92.95
SVM:Default	81.47	86.25	88.40	85.18
SVM:CV	87.32	87.46	88.45	87.27
singleCellNet:Default	83.42	OOM	80.21	90.95
singleCellNet:CV	83.39	OOM	80.61	90.86
scmap-cell-exact:Default	72.44	63.77	89.67	43.57
scmap-cell-exact:CV	72.44	67.02	89.88	75.34

Table S3: **scNym comparison to hyperparameter optimized baseline models.** We optimized the top three baseline models (SVM, scmap-cell-exact, singleCellNet) across four tasks using reverse 5-fold cross-validation. We optimized scNym for two of these tasks. All models were optimized using a grid search strategy. Across all four tasks, scNym with default parameters is superior to the baseline methods with optimized hyperparameters. Values presented are the percent accuracy. OOM indicates that models encountered an out-of-memory error on this task when provided with 128 GB of RAM. Not Tested indicates that we did not perform scNym tuning for this task due to superiority of the default parameters and computational expense.

Cell Type	Gene Ontology Terms
B cell	regulation of B cell receptor signaling pathway; B cell homeostasis; negative regulation of B cell activation; B cell proliferation; ...
T cell	regulation of T cell receptor signaling pathway; regulation of alpha beta T cell activation; regulation of T cell chemotaxis; T cell activation ...
cardiac muscle cell	regulation of cardiac muscle tissue development; ventricular cardiac muscle cell differentiation; relaxation of cardiac muscle; ...
classical monocyte	regulation of monocyte differentiation; monocyte chemotaxis; positive regulation of monocyte chemotaxis; regulation of monocyte chemotaxis; ...
dendritic cell	dendritic cell differentiation; myeloid dendritic cell activation; myeloid dendritic cell differentiation; dendritic cell cytokine production; ...
endocardial cell	endocardial cushion development; endocardial cushion morphogenesis; endocardial cushion formation; ...
endothelial cell	endothelial cell development; positive regulation of endothelial cell chemotaxis; positive regulation of endothelial cell differentiation; ...
hepatocyte	hepatocyte differentiation
keratinocyte	keratinocyte proliferation; regulation of keratinocyte proliferation; regulation of keratinocyte differentiation; ...
kidney loop of Henle ascending limb epithelial cell	negative regulation of kidney development; kidney epithelium development; regulation of epithelial cell differentiation involved in kidney develop...
luminal epithelial cell of mammary gland	mammary gland morphogenesis; regulation of mammary gland epithelial cell proliferation; mammary gland epithelium development; mammary gland development; ...
macrophage	positive regulation of macrophage activation; macrophage activation involved in immune response; regulation of macrophage activation; ...
mas T cell	regulation of mas T cell activation involved in immune response; mas T cell activation; positive regulation of mas T cell activation involved in immune...
mesangial cell	lomerular mesangial cell differentiation; lomerular mesangial cell proliferation; positive regulation of glomerular mesangial cell proliferation; ...
mesenchymal stem cell	mesenchymal stem cell differentiation; regulation of mesenchymal stem cell differentiation
natural killer cell	natural killer cell differentiation; negative regulation of natural killer cell mediated immunity; regulation of natural killer cell activation; ...
non-classical monocyte	monocyte chemotaxis; regulation of monocyte chemotaxis; negative regulation of monocyte chemotaxis
skeletal muscle satellite cell	regulation of skeletal muscle tissue development; skeletal muscle tissue regeneration; positive regulation of skeletal muscle tissue development; ...
type II pneumocyte	type II pneumocyte differentiation

Table S4: **Sample Gene Ontology terms used for model interpretability comparisons** We extracted cell type specific Gene Ontology terms for each of 19 cell types in the Tabula Muris (Methods). We compared the interpretability of scNym-derived salient genes and SVM-derived salient genes based on the enrichment of the top-k salient genes within cell type specific GO terms. A sample of GO Terms used for each cell type is shown above. Ellipses indicate some terms are not shown due to length.

References

[Abdelaal et al., 2019] Abdelaal T, Michielsen L, Cats D, Hoogduin D, Mei H, Reinders MJT, Mahfouz A, 2019. A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biology* **20**: 194. doi:10.1186/s13059-019-1795-z.

[Alquicira-Hernandez et al., 2019] Alquicira-Hernandez J, Sathe A, Ji HP, Nguyen Q, Powell JE, 2019. scPred: accurate supervised method for cell-type classification from single-cell RNA-seq data. *Genome Biology* **20**: 264. doi:10.1186/s13059-019-1862-5.

[Andrews and Hemberg, 2018] Andrews TS, Hemberg M, 2018. M3Drop: dropout-based feature selection for scRNASeq. *Bioinformatics* **35**: 2865–2867. doi:10.1093/bioinformatics/bty1044.

[Berthelot et al., 2019] Berthelot D, Carlini N, Goodfellow I, Papernot N, Oliver A, Raffel CA, 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32*, pp. 5049–5059.

[Denisenko et al., 2020] Denisenko E, Guo BB, Jones M, Hou R, de Kock L, Lassmann T, Poppe D, Clément O, Simmons RK, Lister R, et al., 2020. Systematic assessment of tissue dissociation and storage biases in single-cell and single-nucleus RNA-seq workflows. *Genome Biology* **21**: 130. doi:10.1186/s13059-020-02048-6.

[de Kanter et al., 2019] de Kanter JK, Lijnzaad P, Candelli T, Margaritis T, Holstege FCP, 2019. CHETAH: a selective, hierarchical cell type identification method for single-cell RNA sequencing. *Nucleic Acids Research* **47**: e95. doi:10.1093/nar/gkz543.

[Diehl et al., 2016] Diehl AD, Meehan TF, Bradford YM, Brush MH, Dahdul WM, Dougall DS, He Y, Osumi-Sutherland D, Ruttenberg A, Sarntivijai S, et al., 2016. The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. *Journal of biomedical semantics* **7**: 44–10.

[Ding et al., 2020] Ding J, Adiconis X, Simmons SK, Kowalczyk MS, Hession CC, Marjanovic ND, Hughes TK, Wadsworth MH, Burks T, Nguyen LT, et al., 2020. Systematic comparison of single-cell and single-nucleus RNA-sequencing methods. *Nature Biotechnology* **38**: 737–746. doi:10.1038/s41587-020-0465-8.

[Ganin et al., 2016] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V, 2016. Domain-Adversarial Training of Neural Networks. *arXiv:1505.07818 [cs, stat]* ArXiv: 1505.07818.

[Guo et al., 2017] Guo C, Pleiss G, Sun Y, Weinberger KQ, 2017. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, pp. 1–10.

[Habermann et al., 2020] Habermann AC, Gutierrez AJ, Bui LT, Yahn SL, Winters NI, Calvi CL, Peter L, Chung MI, Taylor CJ, Jetter C, et al., 2020. Single-cell RNA sequencing reveals profibrotic roles of distinct epithelial and mesenchymal lineages in pulmonary fibrosis. *Science Advances* **6**: eaba1972. doi:10.1126/sciadv.aba1972.

[Han et al., 2018] Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, Saadatpour A, Zhou Z, Chen H, Ye F, et al., 2018. Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell* **173**: 1307.

[Hinton et al., 2015] Hinton G, Vinyals O, Dean J, 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

[Kang et al., 2017] Kang HM, Subramaniam M, Targ S, Nguyen M, Maliskova L, McCarthy E, Wan E, Wong S, Byrnes L, Lanata CM, et al., 2017. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nature Biotechnology* **36**: 89–94.

[Kiselev et al., 2018] Kiselev VY, Yiu A, Hemberg M, 2018. scmap: projection of single-cell RNA-seq data across data sets. *Nature methods* **15**: 359–362. doi:10.1038/nmeth.4644.

[Korsunsky et al., 2019] Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh Pr, Raychaudhuri S, 2019. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods* **16**: 1289–1296. doi:10.1038/s41592-019-0619-0.

[Lee, 2013] Lee DH, 2013. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)* .

[Lopez et al., 2018] Lopez R, Regier J, Cole MB, Jordan MI, Yosef N, 2018. Deep generative modeling for single-cell transcriptomics. *Nature methods* pp. 1–11. doi:10.1038/s41592-018-0229-2.

[Ma et al., 2020] Ma S, Sun S, Geng L, Song M, Wang W, Ye Y, Ji Q, Zou Z, Wang S, He X, et al., 2020. Caloric Restriction Reprograms the Single-Cell Transcriptional Landscape of Rattus Norvegicus Aging. *Cell* pp. 1–41. doi:10.1016/j.cell.2020.02.008.

[Platt, 1999] Platt JC, 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, pp. 61–74. MIT Press.

[Sohn et al., 2020] Sohn K, Berthelot D, Li CL, Zhang Z, Carlini N, Cubuk ED, Kurakin A, Zhang H, Raffel C, 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *arXiv:2001.07685 [cs, stat]* ArXiv: 2001.07685.

[Stuart et al., 2019] Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck III WM, Hao Y, Stoeckius M, Smibert P, Satija R, 2019. Comprehensive Integration of Single-Cell Data. *Cell* pp. 1–37. doi:10.1016/j.cell.2019.05.031.

[Tabula Muris Consortium, 2018] Tabula Muris Consortium, 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562**: 367–372.

[Tan and Cahan, 2019] Tan Y, Cahan P, 2019. SingleCellNet: A Computational Tool to Classify Single Cell RNA-Seq Data Across Platforms and Across Species. *Cell Systems* pp. 1–31. doi:10.1016/j.cels.2019.06.004.

[Verma et al., 2019] Verma V, Lamb A, Kannala J, Bengio Y, Lopez-Paz D, 2019. Interpolation Consistency Training for Semi-supervised Learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pp. 3635–3641. AAAI Press.

[Xu et al., 2019] Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N, 2019. Harmonization and Annotation of Single-cell Transcriptomics data with Deep Generative Models. *bioRxiv* pp. 1–46. doi:10.1101/532895.

[Zhang et al., 2018] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D, 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

[Zhong et al., 2010] Zhong E, Fan W, Yang Q, Verscheure O, Ren J, 2010. Cross Validation Framework to Choose amongst Models and Datasets for Transfer Learning. In Balcázar JL, Bonchi F, Gionis A, Sebag M (editors), *Machine Learning and Knowledge Discovery in Databases*, Lecture Notes in Computer Science, pp. 547–562. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-15939-8_35.