Resolving Label Uncertainty with Implicit Posterior Models (Supplementary material)

Esther Rolf[*] [1,6] **Nikolay Malkin**[2,6] **Alexandros Graikos**[3,6] **Ana Jojic**[4] **Caleb Robinson**[5] **Nebojsa Jojic**[6]

[1]University of California, Berkeley, CA, USA
[2]Mila and Université de Montréal, Montreal, QC, Canada
[3] Stony Brook University, Stony Brook, NY, USA
[4] Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA
[5] Microsoft AI for Good, Redmond, WA, USA
[6] Microsoft Research, Redmond, WA, USA

A CODE

This paper is accompanied by a code repository at `github.com/estherrolf/implicit-posterior`. The repository contains three directories. Two of them illustrate our algorithms for partial-label learning and weakly supervised segmentation and are sufficient to reproduce predictions resembling those in Fig. 1. The third directory contains code for the land cover mapping experiments (§4.3, §4.4).

B PRACTICAL CONSIDERATIONS

Mini-batches: Figure 2 shows a PyTorch implementation of the QR and RQ loss functions, where loss is computed over *batches* of training data. Our experiments validate that so long as these batches are large enough to include enough diversity of $(x_i, p_i(l))$ pairs, our method works when Equation (2) and Equation (3) are applied directly to batches. As discussed in §4.4, handling batched input is important for leveraging the scale of large training datasets. As discussed in §2.1, should mini-batch training become an issue in future implementations, it may be beneficial to estimate the denominator of Equation (2) across multiple batches.

To illustrate the dependence of the algorithm on batch size, we ran the MNIST experiment with one negative label (§4.1) with differing batch sizes (Table B.1). The performance degrades at batch sizes 32 and smaller, when batches are likely to be missing samples of some classes.

Relative benefits/limitations of the QR and RQ loss formulations: The algorithm presented in §2.1 details two loss options: a **QR** option and an **RQ** option, both with unique strengths. The QR algorithm is guaranteed to converge as each step reduces loss (except for randomness in the learning algorithm). The RQ algorithm, on the other hand, has the appealing property that it reduces to standard minimization of cross entropy loss in the case of hard labels. In §D, we discuss connections between QR option and variational auto-encoders (VAEs), and between the RQ option and the wake-sleep algorithm. Ultimately, though, we find that which option works better may depend on the application, with RQ working across all applications we tried but sometimes being slightly beaten by QR.

Comparing performance across these varied learning settings can shed light on the performance of the proposed **QR** and **RQ** methods under different conditions. Future research could systematize and formalize settings where one variant would be superior to the other; results in this work show that both can be effective ways to resolve uncertainty in non-“ground-truth" labels.

Simple ways to avoid degenerate solutions: As discussed in §2.1, minimizing Equation (1) can lead to degenerate solutions. However, avoiding these solutions can be quite simple, and in most of our experiments we did not make any

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

Table B.1: Peak test accuracies (following the same experiment settings as in §4.1) and standard deviations over 10 random seeds with different training batch sizes. The last two columns show properties of the distribution over the number of distinct classes in a randomly sampled batch: the likelihood that all ten MNIST classes occur at least once and the expected number of distinct classes that occur.

	peak test acc %			
batch size	RQ	NLL	$\mathbb{P}$[all 10 classes appear in batch]	$\mathbb{E}$[# distinct classes in batch]
256	95.96±0.24	94.57±3.12	100.00%	10.00
128	96.32±0.39	94.83±3.21	100.00%	10.00
64	96.66±0.21	96.15±0.25	98.82%	9.99
32	94.18±1.05	96.64±0.20	69.10%	9.66
16	93.35±3.21	96.85±0.22	7.03%	8.14
8	92.41±4.65	96.78±0.19	0	5.70
4	91.10±6.42	96.99±0.23	0	3.44
2	89.04±10.29	96.93±0.18	0	1.90

interventions to explicitly avoid such local minima. In a targeted experiment in Table E.1 we show that pre-training on hard labels (even out-of-domain) or using sharper learned priors can help break symmetries during early training phases. When hard labels are not available, one could similarly start the training process with a cross-entropy loss on the prior belief, and then switch to RQ or QR loss. The intuition is that first training to minimize cross-entropy breaks the symmetry at the start, while implicit posterior modeling sharpens the predictions in later iterations.

C ADDITIONAL RELATED WORK

There are several approaches to learning with uncertain, weak, or coarse labels under different assumptions and settings. Work on partial-label learning often employs loss functions that aim to decrease prediction entropy [Nguyen and Caruana, 2008, Yao et al., 2020, Yu and Zhang, 2016]. These approaches do not use a generative formulation in these loss functions, making them less suitable for problems with more varied forms of uncertainty encoded in priors. Another approach to learning with imprecise or fuzzy data is to learn a model which finds the best (deterministic) disambiguation of uncertain observations, often by generalizing traditional loss minimization techniques [Hüllermeier, 2014, Couso and Dubois, 2018, Cabannnes et al., 2020].

In §3, we discuss several opportunities to form prior beliefs from weak (e.g. coarse, imprecise, or uncertain) observations, including fusing multiple data sources. While these illustrative examples set the stage for experiments in §4 and §F, several alternative and additional techniques have been developed to model and utilize data from weak sources [Hernández-González et al., 2016, Zhou, 2018]. For example, data programming [Ratner et al., 2016, 2017] provides an opportunity to collect and learn from multiple weak user-provided labeling functions. Another line of work studies the generation and use of pseudolabels in learning settings. Specifically, Zou et al. [2020] relies on a domain-specific augmentation procedure for semantic segmentation with image-level labels, and, Zhang et al. [2021] studies unsupervised clustering applied to object re-identification. Application-specific solutions also include object detection in remote sensing images [Han et al., 2014] and change detection with multitemporal satellite imagery [Zheng et al., 2021, Bao et al., 2021, Li et al., 2021].

In our experimental setups, we chose a mix of baselines to both compare algorithm design and benchmark performance on certain tasks. To compare our approach on an *algorithmic basis*, we compare to the negative logarithm of the sum of likelihoods (NLL), which is used in prior works to handle multiple ambiguous labels [Jin and Ghahramani, 2002] and negative labels [Kim et al., 2019]. We compare to self-epitomic LSR [Malkin et al., 2020] as an algorithmic comparison by which to contrast our method with an "explicit" generative modeling approach. Our similar performance to self-epitomic LSR in regimes where self-epitomic LSR has been shown to perform well (super-resolution in land cover mapping (§4.3) and the tumor-infiltrating lymphocytes task (§F.2)) is an important validation of our motivation in §2.

To benchmark *performance* of our approach across tasks, we compare to state-of-the-art pseudo-labeling methods in supervised text classification (see §4.5), an established 1m resolution map of land cover predictions across the United States [Robinson et al., 2019] and best-performing published results for the land cover mapping tasks we study [Malkin et al., 2020] [Robinson et al., 2020], the best known published results for the tumor-infiltrating lymphocyte segmentation

Table D.1: Comparison of modeling forms for variational auto-encoders (VAE), wake-sleep algorithms (WS), expectation-maximization (EM), and our proposed implicit posterior (IP). Variational auto-encoders parametrize both a generative model p and a posterior model q. Here we distinguish between θ_p and θ_q as these models can differ in both architecture and parameters. The EM formulation parametrizes the generative model $p(x_i|\ell;\theta_p)$ and the posterior is instantiated as auxiliary matrix with entries $a_{i,\ell}$ calculated to maximize the objective given the estimated $p(x_i|\ell;\theta_p)$ on the observed instances i. In implicit posterior modeling, the posterior $q(\ell|x_i;\theta_q)$ is modeled and parametrized directly, with the generative link p instantiated as an auxiliary matrix with entries of the form $a_{i,\ell}$. Combining this auxiliary matrix with the prior beliefs $p_i(\ell)$ at each instance as in Eq. (3) yields a posterior model r_i *implied* by forward model $q(\ell;x_i,\theta_q)$ and weak prior beliefs on each instance $p_i(\ell)$.

	VAE/WS	EM	IP
generative p	$p(x\|\ell;\theta_p)$	$p(x\|\ell;\theta_p)$	$a_{i,\ell}$
posterior q	$q(\ell\|x;\theta_q)$	$a_{i,\ell}$	$q(\ell\|x;\theta_q)$

task [Malkin et al., 2019, 2020], and a host of comparisons for the video instance segmentation task (see Table F.3 for a full list).

As stated in §4.1, the NLL (union) objective and **RQ** are equivalent when $\sum_i q_i(\ell)$ is uniform over ℓ and the prior is uniform over all classes in the negative label sets, evidenced by the comparable performance between the two in Figure 3. In this case, the denominator in (3) is independent of ℓ, and thus

$$r_i(\ell) = \begin{cases} \frac{1}{C-|N_i|} q_i(\ell) & \ell \notin N_i \\ 0 & \ell \in N_i \end{cases},$$

where C is the number of classes and N_i is the negative label set for sample i. The **RQ** loss then simplifies as

$$\mathrm{KL}(r_i \| q_i) = \mathbb{E}_{\ell\sim r_i}\left[\log\left(\frac{r_i(\ell)}{q_i(\ell)}\right)\right] = \sum_{\ell\notin N_i} \frac{1}{C-|N_i|} q_i(\ell) \log \frac{1}{C-|N_i|},$$

which is a constant multiple of the NLL (union) loss $\sum_{\ell\notin N_i} q_i(\ell)$.

Lastly, it is worth noting that the similar term “implicit generative model" has been used in prior literature to refer to amortized sampling procedures for nonparametric (or not specified) energy functions, such as generative adversarial models (e.g., Mohamed and Lakshminarayanan [2017]). Although we do not make an explicit connection with such models, our formulation also does not assume a parametrization of the data distribution, and one can understand the term “implicit posterior” as referring to a function that is a posterior for an implicit (i.e., uninstantiated, unparametrized) generative model. However, we assume tractability of sampling from a posterior over certain distinguished latents (classes) conditioned on observed data (features, e.g., images), rather than directly sampling latents.

D RELATIONSHIPS WITH EM, VAE, AND WAKE-SLEEP ALGORITHM

As discussed in §2.1, the **QR** loss guarantees continual improvements in the free energy (1). On the other hand, option **RQ** is equivalent to performing a gradient step on the cross-entropy of q_i and r_i and a gradient step on the *negative* entropy of r_i. In the case that the priors $p_i(\ell)$ are hard (supported only on one ground truth label), the same is true of r_i, and the **RQ** loss is equivalent to cross-entropy. This option reverses the KL distance in a manner reminiscent of the training procedure in the wake-sleep algorithm [Hinton et al., 1995], where parameter updates for the forward and reverse models are iterated, but the KL distance optimized always places the probabilities under the model being optimized in the second position in the KL distance (inside the logarithm), so that the generative and the inference models each optimize log-likelihoods of their predictions. The wake-sleep algorithm, however, also trains a generative model rather than treating it as an auxiliary distribution as we do, and that requires sampling. As opposed to VAEs, the wake-sleep algorithm samples the generative model, not the posterior.

It is interesting to contrast our approach to the expectation-maximization (EM) formulation. In standard EM, the q distributions are considered auxiliary, rather than parametrized as direct functions of the inputs x. The $q_i(\ell) = a_{i,\ell}$ is simply a matrix of numbers normalized across ℓ. Its dependence on the data x arises through the iterative re-estimation of the minimum of the free energy, where the link $x - \ell$ is modeled directly in the parametrized forward distribution $p(x|\ell)$ (see

Table D.1). We instead model forward probabilities $p(x_i|\ell)$ as auxiliary parameters, a matrix of numbers $a_{i,\ell}$ normalized across i that we fit to minimize the free energy at each data point, and optimize only the parameters of the q model which explicitly models the link $x - \ell$. This allows us to capture nonlinear (and 'deep') structure and benefit from inductive biases inherent to training deep models with SGD, but without the cost of training an actual parametrized generative model and other problems associated with deep generative model fitting. The resulting q network approximates the posterior in a generative model – which (locally) maximizes the log likelihood of the data – and it is usually highly confident (as seen in Fig. 1).

The implicit modeling of the posterior in EM does not lead to overfitting of the generative model. But, given that degenerate solutions to optimization with implicit posterior models are possible when the prior is constant across all data points (§2.1), we can imagine that our approach of implicit posterior modeling might lead to degenerate solutions. As demonstrated in Fig. 1 and in our experiments, avoiding degenerate solutions is not too hard. We address this point further in §B.

E EXPERIMENT DETAILS

E.1 LAND COVER MAPPING

E.1.1 Datasets

Imagery Data Our land cover mapping experiments use imagery from the National Agriculture Imagery Program (NAIP), which is 4-channel aerial imagery at a $\leq$ 1m/px resolution taken in the United States (US).

Chesapeake Conservancy land cover dataset The Chesapeake Conservancy land cover dataset consists of several raster layers of both imagery and labels covering parts of 6 states in the Northeastern United States: Maryland, Delaware, Virginia, West Virginia, Pennsylvania, and New York [Robinson et al., 2019][1]. The raster layers include: high resolution (1m/px) NAIP imagery, high resolution (1m/px) land cover labels created semi-autonomously by the Chesapeake Conservancy, low resolution (30m/px) Landsat-8 mosaics imagery, low resolution (30m/px) land cover labels from the National Land Cover Database (NLCD), and building footprint masks from the Microsoft Building Footprint dataset. The dataset is partitioned into train, validation, and test splits per-state, where each split is a set of $\approx$ 7km $\times$ 6km *tiles* containing the aligned raster layers.

EPA EnviroAtlas data The EnviroAtlas land cover data consists of high resolution (1m/px) land cover maps over 30 cities in the US, and is collected and hosted by the US Environmental Protection Agency (EPA) [Pickard et al., 2015]. A detailed description of the dataset and its land cover definitions is provided by Pilant et al. [2020]. As with most high-resolution land cover datasets (including the Chesapeake Conservancy land cover labels), the EnviroAtlas land cover labels are themselves derived by remote sensing and learning procedures, and thus are not themselves a perfect "ground truth" representation of land cover. For example, the estimated accuracy of the provided labels is 86.5% in Pittsburgh, PA, 83.0% in Durham, NC, 86.5% in Austin, TX, and 69.2% in Phoenix, AZ [Pilant et al., 2020].

The high-resolution label files were aligned to match the extent of the NAIP tiles from the closest available years to the years that the EnviroAtlas labels were collected: for Pittsburgh, PA and Phoenix, AZ, we used data from 2010 and for Durham, NC and Austin, TX, we used data from 2012. We chose these four cities to get a wide coverage across the United States (US), and due to a mostly consistent set of classes being used between the four cities.

National Land Cover Database (NLCD) The National Land Cover Database is produced by the United States Geological Survey (USGS) and uses 16 land cover classes. Maps are generated every 2-3 years, with spatial resolution of 30m/px. Data and more information can be found at: `https://www.usgs.gov/centers/eros/science/national-land-cover-database`.

Microsoft Building Footprint dataset The Microsoft Building Footprint dataset consists of predicted building polygons over the continental US from Bing Maps imagery. As of the time of writing, the most updated Microsoft Building Footprints dataset in the US can be accessed at: `https://github.com/Microsoft/USBuildingFootprints`.

[1] Dataset can be downloaded from: `https://lila.science/datasets/chesapeakelandcover`.

Open Street Map (OSM) data Open Street Map (`https://www.openstreetmap.org/`) is an ongoing effort to make publicly available and editable map of the world, generated largely from volunteer efforts. The data is available under the Open Database License. From the many different sources of information provided by OSM [Haklay and Weber, 2008], we download raster data for road networks, waterways, and water bodies, using the OSMnx python package [Boeing, 2017].

Data splits and data processing For experiments using the Chesapeake Conservancy dataset (Table 1), we used established train, test, and validation splits. In particular, we used the 20 test tiles in New York (NY) and the 20 test tiles in Pennsylvania (PA) on which to conduct our experiments. Here a *tile* matches the extent of a NAIP tile, roughly 7km × 6km. To facilitate comparison of our results with previous published results on this dataset, we condensed the labels into four classes: (1) water, (2) impervious surfaces (roads, buildings, barren land), (3) grass/field, and (4) tree canopy.

For experiments with the EnviroAtlas dataset (Table 2), we aligned the high resolution land cover data, NLCD, OSM, and Microsoft Building Footprints data with NAIP imagery tiles, matching years as closely as possible to the EnviroAtlas data collection year for NLCD and NAIP. We instantiated a split of 10 train, 8 validation, and 10 test tiles in Pittsburgh, and 10 test tiles in Durham, NC, Austin, TX, and Phoenix, AZ. For Pittsburgh we assigned tiles to splits randomly from the set of 28 tiles that had no missing labels. There were not enough such tiles in Durham to follow the same procedure, so we chose the ten evaluation tiles at random from a set with no number of missing labels per tile. For Austin and Phoenix, we chose the 10 evaluation tiles at random from the tiles in each city that had no agriculture class (as it is not present in Pittsburgh or Durham) and no missing labels. We set aside 5 separate tiles in each city for use in "learning the prior" (in Pittsburgh these 5 tiles are a subset of the 8 validation tiles). As above, each tile corresponds to one NAIP tile. The tiles in these constructed sets for Pittsburgh, Durham, and Austin contain five unique labels: (1) water, (2) impervious surfaces (roads, buildings), (2) barren land, (4) grass/field, and (5) trees. Phoenix additionally has a "shrub" class; when forming the prior we merge this class with trees, and we ignore the shrub class when evaluating in Phoenix. We cropped all data tiles to ensure no spatial overlap in any tiles between or within the train/val/test splits.

E.1.2 Forming the priors

To form the priors for the land cover classification tasks, we first spatially smooth the NLCD labels by applying a 2D Gaussian filter (with a standard deviation of 31 pixels) across every channel in a one-hot representation of the NLCD classes. The main reason for applying this smoothing is to reduce artifacts due to the $30m^2$ boundaries of the NLCD data, to undo the blocking procedure induced by the aggregation to 30m × 30m extents, to incorporate the spatial correlations between nearby NLCD blocks, and to remove erroneous sharp differentials between inputs that can cause artifacts during later training stages.

We then remap the blurred NLCD layers to the classes of interest by multiplying by a matrix of cooccurrence counts between the (unblurred) NLCD data and the high resolution labels in each region. For the Chesapeake region, we use the train tiles provided with the Chesapeake Conservancy land cover dataset to define cooccurrence matrices in NY and PA. For EnviroAtlas, we compute cooccurrences using the entire city (excluding tiles with agriculture in Phoenix AZ, and Austin, TX). The cooccurrence matrices for each region we study are shown in Figure E.1.

The priors for the Chesapeake Conservancy dataset are then generated by normalizing the blurred and remapped NLCD data so that summing over all five classes gives probability 1 for each pixel.

For the EnviroAtlas data, we augment this prior with publicly available data on buildings, road networks, water bodies, and waterways. We obtain building maps from the Microsoft Buildings Footprint database and road, water bodies, and waterways data from Open Street Map, using the OSMnx tool [Boeing, 2017] to download the data (see Appendix E.1.1). We apply a small spatial blur to each of these input sources to account for (a) vector representation of roads and waterways being unrealistically thin, and (b) possible data-image misalignment on the order of pixels. Where this results in probability mass on impervious surfaces or water, we add these probability masses to the blurred NLCD prior, and then renormalize to obtain a valid set of probabilities for each pixel.

In §4.4, we describe a method for "learning the prior," which uses a more sophisticated process to aggregate the individually weak and coarse inputs that we use in the handmade prior. In this method, we train a neural net to take as input the blurred, remapped NLCD representation (5 classes) concatenated with the 4 classes of additional data: buildings, roads, waterways, water bodies, and to predict high-resolution labels in each city. We train these networks using 5 tiles of imagery and high-resolution labels from the EnviroAtlas Dataset in each city which are distinct from the 10 test tiles in each city. The training procedure for these prior generation networks is described in in §E.1.3. To create the priors that we then train

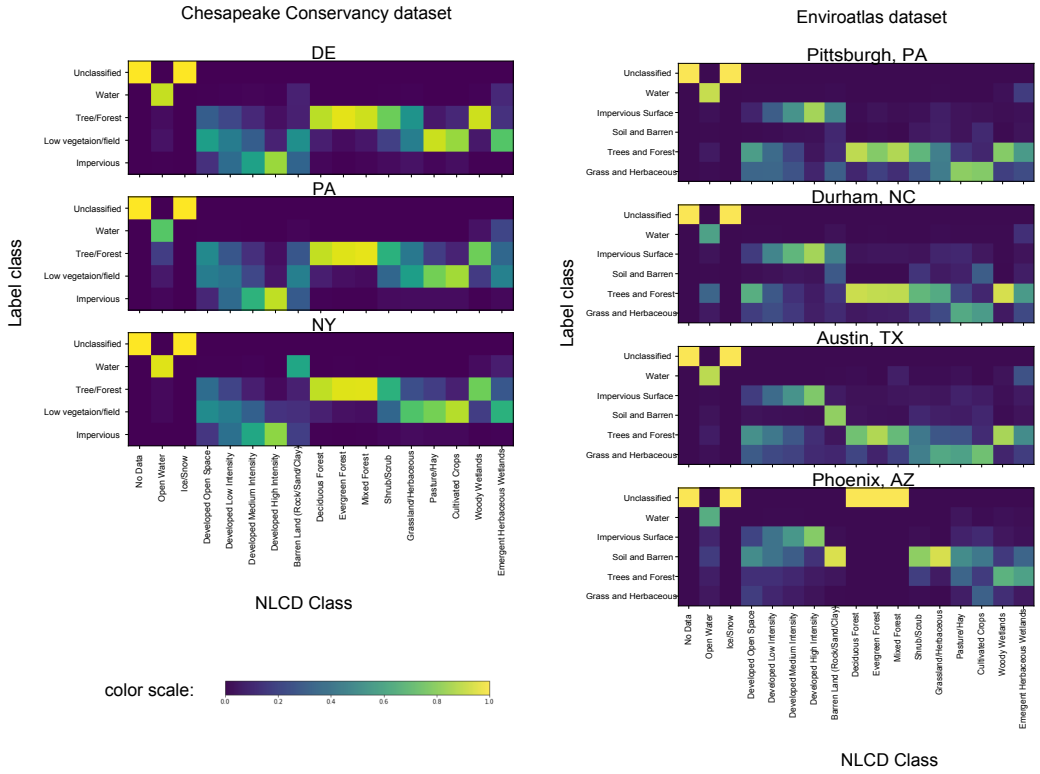

Figure E.1: Cooccurrence matrices between NLCD classes and high resolution land cover labels for each region we study.

our method on ('learned prior' rows in Table 2) we ran these learned models forward on (blurred and remapped NLCD, buildings, roads, waterways, and waterbodies) input for each of the 10 evaluation tiles in each city.

E.1.3 Experimental procedure

We use priors generated as described in Appendix E.1.2, with Gaussian spatial smoothing with standard deviation of 31 pixels, and cooccurrence matrix determined via the training splits in each city/state. We apply a pixel-wise additive smoothing constant of 1e-4 to the probability vectors output by the neural network as well as to the prior probability vectors used as the model supervision data. This additive smoothing constant ensures that there are no extremely low probability classes in either the prior or the predicted outputs during training.

Experiments summarized in Table 1 and Table 2 use a 5-layer fully connected network with kernel sizes of 3 at each layer, 128 filters per layer, and leaky ReLUs between layers. Note that the receptive field of this model is only 11×11 pixels. We use batch sizes of 128 instances during training, where each image instance is a cropped 128×128 pixels from a larger tile. Training and model evaluation is done within the torchgeo framework for geo-spatial machine learning [Stewart et al., 2021]. All models use the AdamW optimizer [Loshchilov and Hutter, 2017] during training and torchgeo defaults unless otherwise noted.

Comparison to previous label super-resolution for LC mapping To obtain the parameter setting used for the runs in New York (NY) and Pennsylvania (PA) in Table 1, we first perform a hyperparameter search with the 20 tiles test set in Delaware (DE) from the same overall dataset. We use a learning rate schedule that decreases learning rate when the validation loss plateaus, as well as early stopping to prevent over training of models. Of the grid of learning rates in {1e-3, 1e-4,1e-5}, we describe below, we pick learning rate as 1e-4 for both **QR** and **RQ** variants of our method, as this is the setting that minimizes the IoU of the q output on the 20 DE tiles for both variants.

When training on NY and PA jointly ("Chesapeake" in Table 1), we use the per-state cooccurrence matrices. This ensure that the cooccurrence matrices used are consistent between our method and the self-epitomic LSR benchmark across all columns in Table 1.

Generalization across cities. For the high-resolution model with NAIP imagery from Pittsburgh as input, we consider learning rates in $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and pick based on the best validation performance on the validation set in Pittsburgh. The chosen learning rate is 1e-3. We search over the same set of learning rates for the model with NAIP imagery and the prior concatenated as input; the chosen learning rate is also 1e-3. For this model with concatenated image and prior as input, only the number of input channels changes in the fully connected network model architecture. When training on the high-resolution land cover labels, we use a very small additive constant (1e-8) for the last layer of the model.

When training our methods, we initialize model weights using the best NAIP image input model from the Pittsburgh validation set runs, and then train using the priors and the training procedure described in the main text. We pick the learning rate for this training step using again the validation set in Pittsburgh; we search learning rates in $\{10^{-3}, 10^{-4}, 10^{-5}\}$, and pick 1e-5 as the learning rate for **QR** and 1e-3 as the learning rate for **RQ**, since these resulted in the best performance for the Pittsburgh validation set with the randomly initialized model. We discuss the results of a similar procedure using randomly initialized model weights in Appendix E.1.4.

For the learned prior, we use a 3 layer fully connected network is kernel sizes of 11,7, 5 respectively, 128 filters per layer and leaky ReLUs between layers. For each city, we train this model on the prior inputs (blurred and remapped NLCD, roads, buildings, waterways, and water bodies) using a validation set of 5 tiles separate from from the 10 evaluation tiles in each city. We considered learning rates in $\{10^{-3}, 10^{-4}, 10^{-5}\}$ for learning the prior in each city, and chose 1e-4 as it gave most often resulted in the highest accuracies of each validation set. For learning *on* this learned prior, we again initialize model weights using the best NAIP image input model from the Pittsburgh validation set runs, and set the learning rate to 1e-5 for **QR** evaluation runs and 1e-3 for **RQ** evaluation runs to match the other variants of the experiment.

Table E.1: Supplementary results to accompany Table 2.

		Pittsburgh, PA		Durham, NC		Austin, TX		Phoenix, AZ	
Train region	Model	acc %	IoU %	acc %	IoU %	acc %	IoU %	acc %	IoU %
Pittsburgh (supervised)	HR	89.3	69.3	74.2	35.9	71.9	36.8	6.7	13.4
	HR + aux	89.5	70.5	78.9	47.9	77.2	50.5	62.8	24.2
Same as test (random initialization)	**QR** (q)	80.5	56.8	78.3	44.4	79.2	50.5	75.2	29.5
	QR (r)	80.7	57.5	78.5	46.4	79.7	52.0	75.9	33.8
	RQ (q)	77.6	53.3	65.8	23.3	73.8	43.0	61.8	18.6
	RQ (r)	77.6	53.3	65.8	23.3	73.8	43.1	61.8	18.6
Same as test (pretrained in Pittsburgh)	**QR** (q)	80.6	58.5	78.9	47.7	76.6	49.1	75.8	45.4
	QR (r)	80.6	58.7	79.0	48.4	76.6	49.5	76.2	46.0
	RQ (q)	84.3	59.6	75.6	28.6	76.5	47.5	63.7	19.5
	RQ (r)	84.3	59.6	75.4	31.5	76.5	47.5	63.7	19.5
Same as test (learned prior)	**QR** (q)	82.4	63.7	79.0	48.7	79.4	51.3	73.4	42.8
	QR (r)	82.4	64.0	79.2	49.5	79.1	51.9	73.6	43.1
Full US* Robinson et al. [2019]	U-Net Lrg.	79.0	61.5	77.0	49.6	76.5	51.8	24.7	23.6

E.1.4 Additional Results

Extended results for generalizing across EnviroAtlas cities. The extended results for generalizing across cities with the EnviroAtlas datasets in Table E.1 contain the results of the **RQ** runs trained on the handmade prior in each city. Evaluation results in Pittsburgh, PA give further context for comparison of generalization across cities by each method.

Table E.1 also details the result of initializing the model weights randomly for the **QR** method. Table E.1 shows that the choice of model initialization can be important for our method – this is most apparent in Pittsburgh, PA (unsurprisingly since the high-resolution model was trained in Pittsburgh) and Phoenix, AZ. In Phoenix, much of the handmade prior is consistent across geographies and the randomly initialized model has trouble distinguishing between infrequent classes that most often occur together in the handmade prior. The results in Table E.1 suggest that using pre-trained models as a starting point for our method can help to break some of these symmetry issues in resolving the information in the prior. Results in Table 2 suggest that using a more detailed prior map may help with this as well.

Table E.2: Comparison of the Full US* U-Net Large [Robinson et al., 2019] map predictions when evaluated on the full 5 classes considered in Table 2 (water, grass/field, trees/shrub, impervious surfaces, and barren land) and evaluated on the four prediction classes predicted by the model (where barren land and impervious surfaces are merged as a single class), and when barren is post-facto assigned whenever the predicted class is “impervious surfaces" and the label class is “barren land".

	Pittsburgh, PA		Durham, NC		Austin, TX		Phoenix, AZ	
Classication Scheme	acc %	IoU %	acc %	IoU %	acc %	IoU %	acc %	IoU %
5 Classes	78.8	55.1	76.6	43.4	76.2	49.1	18.2	18.8
4 Classes	79.0	68.7	77.0	54.1	76.5	60.4	24.7	16.8
Barren reassigned	79.0	61.5	77.0	49.6	76.5	51.8	24.7	23.6

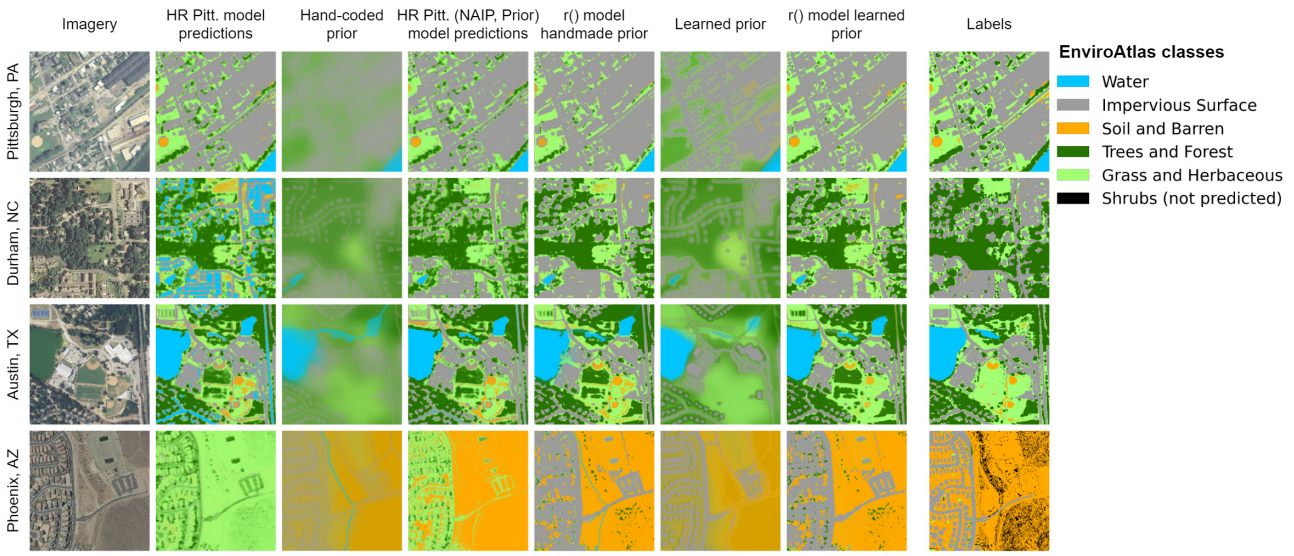

Figure E.2: Example predictions on the hand-coded and learned prior in each EnviroAtlas city we study.

Evaluating the Full US map from Robinson et al. [2019]. Recall that the row for the full US Map [Robinson et al., 2019] in Table 2 reflects the performance of the model evaluated on all 5 classes we consider in our experiments, where we give the map predictions the “benefit of the doubt" in that any prediction of “impervious surfaces" where the true label is “barren land" gets assigned a correct classification of “barren land." The results reported in Table 2 are thus a sort of upper bound on the predictive performance of the method that generated the predictive maps. It was important for us to keep the barren class while evaluating across cities, as it is the dominant class in Phoenix, AZ. In the remaining three cities, the barren class is challenging to predict as it is infrequent. In Table E.2, we compare this classification scheme with two alternatives: a 5 class scheme that will penalizes the map predictions for never predicts the barren class, and a 4 class scheme that merges the barren land and impervious surfaces classes in evaluation. Table E.2 shows that while the choice of evaluation scheme does not greatly effect accuracy (outside of Phoenix, AZ, where the accuracy of the Full US Map is low for both classification schemes), the average IoU drops significantly for all cities apart from Phoenix.

Comparing loss functions: qualitative results with land cover mapping. Figure E.3 compares predictions under different loss functions with an illustrative example. Here the prior is similar to the “hand-coded" prior described in Appendix E.1.2, but with the prior defined over all NLCD classes. We train each model (a slight variant on the network used in experimental results) on the single NAIP tile region encompassing the zoom-in in the figure for 2000 iterations with the Adam algorithm [Kingma and Ba, 2014], a batch size of 64, and a learning rate fixed at 1e-4 during training. Qualitative comparisons show that predictions made by the **QR** and **RQ** loss functions are more certain (sharper colors in plots) than training with cross entropy or squared-error loss on the soft priors, and, in in most places, arrive at better solutions than training with a standard cross entropy loss on the argmax of the prior.

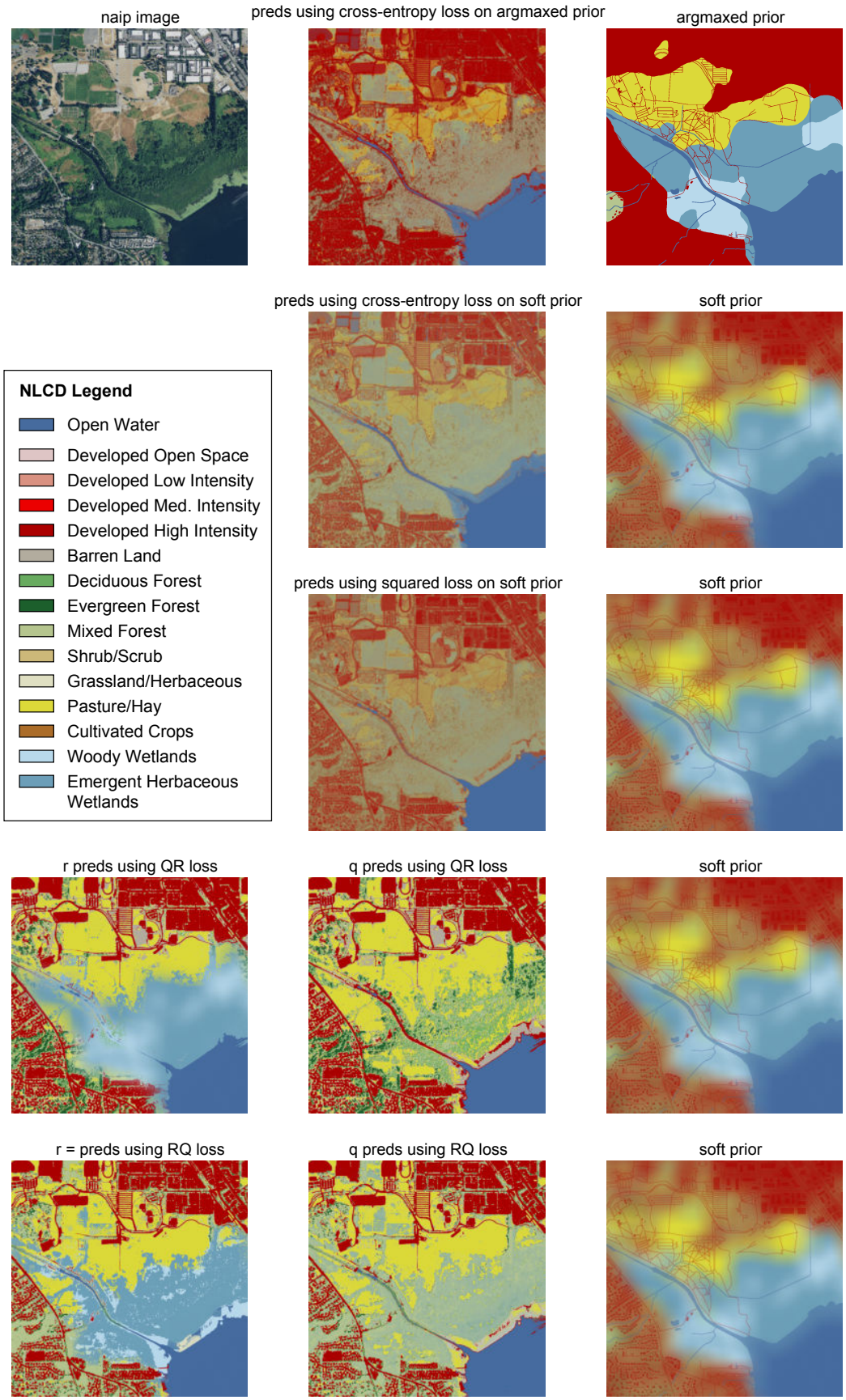

Figure E.3: Comparison of different loss functions on hard and soft prior.

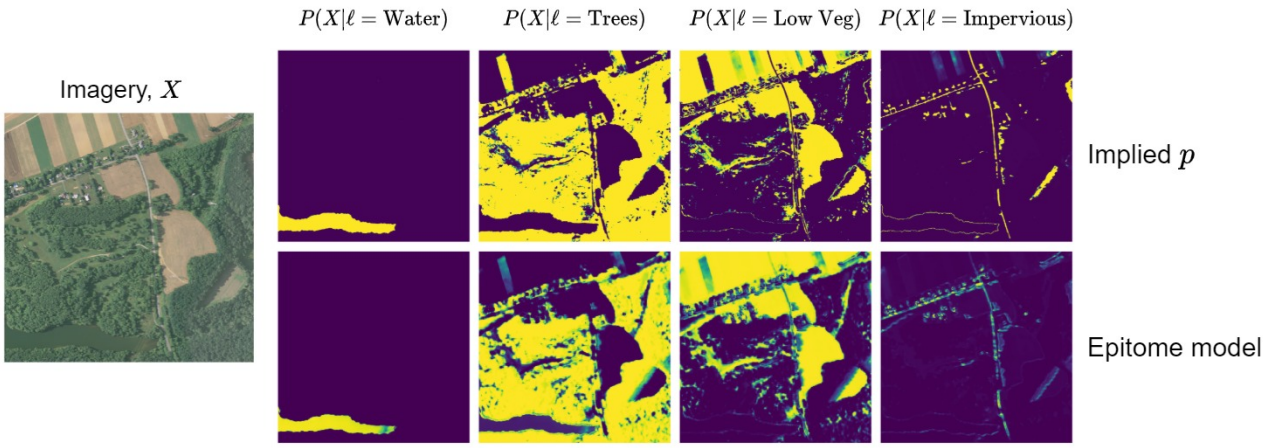

Figure E.4: Comparison of forward model likelihoods under the generative model trained with **QR** loss (above) and the likelihood under an epitome model [Malkin et al., 2020] for part of a test tile from §4.3.

F ADDITIONAL EXPERIMENTS

F.1 SELF-SUPERVISION FOR UNSUPERVISED IMAGE CLUSTERING

Neural networks are usually trained on large amounts of hard-labeled data $\{x_i, \ell_i\}$, yet, due to the biases induced by the typical architectures and learning algorithms, much of the modeling power of these networks seem to focus on correlations in the input space [Shwartz-Ziv and Tishby, 2017]. This means that a network trained for one application, i.e., for one label space $\ell \in L_1$, can be adopted to another application, i.e., a different labels space $\ell \in L_2$, as long as the input features are in a similar domain. The canonical example of this is the use of lower levels of the networks pre-trained on ImageNet as part of the networks solving a completely different set of image classification problems. Pretrained networks require smaller training sets in fine tuning, as long as they have learned to represent the variation in the input space well. Self-supervised models attempt to go a step further and learn these representations without *any* labels. In our framework, self-supervision can simply be seen as the appropriate choice of subset priors $p(\ell_T)$ over appropriately chosen tuples of labels.

To discuss the pitfalls and opportunities, consider again the **QR** loss (5)

$$F = -\sum_{i,\ell} q_i(\ell) \log p_i(\ell) + \sum_{i,\ell} q_i(\ell) \log \left(\sum_j q_j(\ell) \right) . \tag{F.1}$$

If we were to simply set $p_i(\ell)$ to a constant (e.g., uniform) distribution $p(\ell)$ for all data points i, then the optimal solution would be any function $q_i(\ell) = q(\ell|x_i)$ such that $\frac{1}{N} \sum_i q(\ell|x_i) = p(\ell)$. Thus simply using the uniform prior may not lead to appropriate unsupervised clustering (or self-supervised learning of the network q). The inductive biases in the network architecture and training may not help, because one solution is $q(\ell|x) = p(\ell)$, which can be achieved by zeroing out all weights except for biases in a final softmax layer that outputs probabilities for labels ℓ. As the softmax bias vector is the closest to the top in back-propogation with gradient descent, it will quickly be learned to match $\log p(\ell)$. This will not only slow down the propagation of gradients into the network, but could eventually stop it completely, as this solution is a global optimum. Another optimal solution would be a function satisfying $\frac{1}{N} \sum_i q(\ell|x_i) = p(\ell)$, but where individual entropies for each data point are small: $-\sum_\ell q(\ell|x_i) \log q(\ell|x_i) < \epsilon$, which motivates an alternative cost criterion:

$$F = -\sum_{i,\ell} q_i(\ell) \log q_i(\ell) + \sum_{i,\ell} q_i(\ell) \log \left(\sum_j q_j(\ell) \right) . \tag{F.2}$$

where the first term promotes certainty in predictions $q(\ell|x_i)$ for each point i and the second is promoting the diversity of the predictions across the different inputs, i.e., a high entropy of the average $\frac{1}{N} \sum_h q_i(h)$. This prevents learning a network with a constant output $q(h) = p(h)$ and forces the model to find some statistics in the input data that break it into clusters indexed by labels ℓ. The result will be highly dependent on the inductive biases associated with the network architecture and SGD method used, as we can imagine degenerate solutions here as well. For example, we can ignore completely some

subset of features and still train a network that is certain in its modeling of the remaining ones, and achieves a high diversity of predicted classes across the dataset. This may be dangerous if the features omitted end up being the most important ones for the downstream task. However, due to the stochastic gradient descent training as well as their architecture, it has been difficult to prevent neural networks from learning statistics involving all the input features. For example, training a neural network using a weak generative model as a teacher corresponds to using a simpler mixture model, whose posterior is used as a target $p_i(\ell)$ and then learning a neural network that can approximate it. The inductive bias then leads to networks that do not match $p_i(\ell)$ exactly but learn more complex statistics instead.

Equation (F.2) can be seen as a degenerate example of using a tuple prior where the tuple has the same data point repeated and the prior simply expects the two predictions to be the same. In many applications, there are natural constraints involving multiple data points that are easily modeled with priors over tuples or over the entire collection of labels. Consider unsupervised image segmentation, for an example. It is usually expected that nearby pixels should belong to the same class (or a small subset of classes), and that faraway pixels are more likely to belong to a different subset of classes. This belief is typically modeled in terms of Markov random field models of joint probabilities of labels in the image,

$$p(\{\ell_i\}) \propto \exp \sum_i \phi(\ell_i, \{\ell_j\}_{j \in N_j}). \tag{F.3}$$

We experimented with potentials of the form

$$\phi(\ell_i = \ell, \{\ell_j\}_{j \in N_j}) = \gamma_\ell + \alpha_\ell \frac{1}{|S_i|} \sum_{j \in S_i} \mathbb{1}[\ell = \ell_j] + \beta_\ell \frac{1}{|L_i|} \sum_{j \in L_i} \mathbb{1}[\ell = \ell_j], \tag{F.4}$$

where for pixel i, S_i is a small (5×5) neighborhood around it and L_i is a larger (50×50) neighborhood. If we set $\alpha_\ell = 1$, $\beta_\ell = -1$ for all ℓ, then we consider this a contrastive prior, as it favors labels ℓ_i to match the labels found more concentrated in its immediate neighborhood than in the larger scope. On the other hand α_ℓ, and β_ℓ can be estimated based on the current statistics in the label distribution using logistic regression. We refer to this as a self-similarity prior $p(\{\ell_i\}; \alpha_\ell, \beta_\ell, \gamma_\ell)$ with parameters which are periodically fit to the current statistics in the predictions $\sum_{j \in S_i} q(\ell | x_j)$, and $\sum_{j \in L_i} q(\ell | x_j)$ to promote similar label patterns across the image. The criterion (F.2) can also be seen as a degenerate version of this setting with S being 1×1 and L being infinite (or the whole image).

The contrastive version of this prior relies on the insight previously pursued in image self-supervision, e.g., Jean et al. [2019]. In our formulation, contrasting is accomplished without sampling triplets, but considering all the data jointly, by expressing the goal of contrasting with far away regions within the prior in our framework.

As an example of self-supervised pretraining in our framework, in Fig. F.1 we show an example of clustering a large tile of aerial imagery into 12 classes using 5 layer FCN as network q of the architecture used in §4.4. The clustering is achieved by updating the prior every 50 steps of gradient descent on batches of 200 256×256px patches. The prior is initialized to a contrasting prior, and then updated through gradient descent. After 7 iterations, the result is sharpened by continuing training using (F.2).

This tile was recently used in testing the fine-tuning of a pretrained model with minimal amount of new labels in a new region [Robinson et al., 2020]. Both the pre-training region, the state of Maryland, and the testing region, the tiles in New York State, come from the 4-class Chesapeake Land Cover dataset (§4.3). Yet, the slight shift in geography results in reduction of accuracy from around 90% in Maryland down to around 72.5% in New York. In Robinson et al. [2020], various techniques for quick model adaptation are studied, on labels acquirable in up to 15 minutes of human labeling effort per tile. In Table F.1 we compare the tunability of our self-supervised models on the four 85km^2 regions tested in Robinson et al. [2020] with active learning approaches to tuning a pre-trained Maryland model with 400 labeled points. We show in the table the accuracy and mean intersection over union from Robinson et al. [2020] for tuning the pretrained model's last 64×4 layer with different active learning strategies for selecting points to be labeled. For example, random selection of 400 points for which the labels are provided yields an average accuracy improvement from 72.5% to 80.6%.

On the other hand, recall that we have created an unsupervised segmentation into 12 clusters, with posteriors over the clusters $q_i(\ell)$. To investigate how well these clusters align with ground truth land cover labels, we compute a simple assignment of clusters to land cover labels. Given a set of labeled points $\{(i, c_i)\}_{i \in I}$, we infer a mapping from clusters to four target labels,

$$p(c|\ell) \propto \sum_{i \in I : c_i = c} q_i(\ell).$$

The label of any point j can now be inferred as $\hat{\ell}_j = \arg\max_c \sum_\ell q_i(\ell) p(c|\ell)$. This procedure, using 400 randomly selected labeled points, yields an average accuracy of 81.1% (averaged over 50 random collections of labeled points), which is above

Figure F.1: Unsupervised clustering using implicit **QR** loss (middle) of a NAIP tile (left). On the right, we show the assignment of the 12 clusters to 4 land cover labels: water (blue), tall vegetation (darker green), low vegetation (lighter green) and impervious/barren (gray).

the performance of the pretrained model tuned on as many randomly selected points, and on par with the more sophisticated methods for point selection and the use of the pretrained model (Table F.1). (Note that the large model pretrained was trained on a large similar dataset in a nearby state).

Table F.1: Finetuning a pre-trained model by gradient descent [Robinson et al., 2020] versus implicit **QR** clustering + label assignment in low-label regimes.

	pretrained model in Robinson et al. [2020]				Implicit QR
Query method	No tuning	Random	Entropy	Min-margin	Random
Tuned parameters	0	64×4	64×4	64×4	12×4
Accuracy %	72.5	80.6	73.6	81.1	81.1
IoU %	51.0	60.8	50.1	60.8	59.8

F.2 TUMOR-INFILTRATING LYMPHOCYTE SEGMENTATION

The setup of this experiment mimics that of the land cover label super-resolution experiment in §4.3. The training data consists of 50,000 240×240px crops of H&E-stained histological imagery at 0.5μm/px resolution, paired with coarse estimates of the density of tumor-infiltrating lymphocytes (TILs) created by a simple classifier, at the resolution of 100×100 blocks. The goal is to produce models for high-resolution TIL segmentation. Models are evaluated on a held-out set of 1786 images with high-resolution point labels for the center pixel.

The coarse density estimates c belong to one of 10 classes, from 0 (no TILs) to 9 (highest estimated TIL density). We use an estimated conditional likelihood $p(\ell|c)$ of the likelihood of the positive TIL label at pixels with each low-resolution class c to construct a prior $p_i(\ell)$ over the TIL label probability. Notice that this prior is the same for all pixels in any given low-resolution, coarsely labeled block.[2]

We train a small CNN with receptive field 11×11 (five ReLU-activated convolutional layers with 64 filters) under the **RQ** loss against this prior for 200 epochs with learning rate 10^{-5}, then evaluate on the held-out testing set. Inspired by Malkin et al. [2020], we apply a spatial blur of 11 pixels to the predicted log-likelihoods (again correcting for the model's small receptive field and the dataset bias).

The AUC scores of this model and of the baselines are shown in Table F.2. Interestingly, the best-performing models – **RQ**

[2]We experimented with setting $p_i(\ell|c)$ to conditional likelihoods estimated from a held-out set and with simply setting $p_i(\ell = 1|c = 0) = 0.05$, $p_i(\ell = 1|c = 1) = 0.15$, …, $p_i(\ell = 1|c = 9) = 0.95$. The latter gave better results, perhaps due to the bias of the evaluation set, in which every image is known to be centered on a cell of some kind.

Table F.2: Area under ROC curve for various predictors on the TIL segmentation task.

	fully supervised			weakly supervised		
Model	SVM[a,b]	CNN[b]	CSP-CNN Hou et al. [2019]	U-Net[c]	Epitome[d]	**RQ**
AUC	0.713	0.494	0.786	0.783	0.801	0.802

[a]Zhou et al. [2017] [b]Hou et al. [2019] [c]Malkin et al. [2019] [d]Malkin et al. [2020]

and epitomic super-resolution (a generative model) – both have receptive fields of 11×11, much smaller than those of the U-Net and fully supervised CNNs. This means that prediction of TIL likelihood is possible using only *local* image data, but the challenge is learning to resolve highly uncertain label information. Unlike U-Nets and deep CNN autoencoders, small models are not able to learn and overfit to *distant* spurious clues to the classes of nearby pixels.

F.3 VIDEO SEGMENTATION WITH A STRUCTURED PRIOR

To demonstrate the use of priors with latent structure, we set up the problem of video segmentation as follows. Given a frame t, we tune networks $q_t(\ell_{i,t}|x_{i,t})$ predicting one of L pixel classes for a pixel at coordinate i in frame t. The prior in each frame comes from a Mask R-CNN model [He et al., 2017] pre-trained on still images in the COCO dataset [Lin et al., 2014]. The Mask R-CNN model finds several possible instances of objects of different categories and outputs the soft object masks in form of confidence scores for each pixel. We convert this into a probability distribution over the index f (foreground/background) of the form $p(f_{i,t}|m_t)$, where m_t are different detected instances by the model, and the distributions $p(f_{i,t}|m_t)$ are the soft masks for these instances converted to probability distributions, i.e. value of the probability of foreground differs for each pixel and each instance based on the Mask R-CNN confidence scores. Although the COCO dataset may not have had instances of object of interest in our frame x_t, we assume that some admixture (i.e., mixture with sample-dependent weights) of detected instances (likely involving unrelated types of objects) does model reasonably well the foreground segmentation in the frame. Mathematically, $p(f_{i,t}) = \sum_{m_t} p(f_{i,t}|m_t)p(m_t)$, where $p(m_t)$ expresses the probabilistic selection of the foreground masks for different instances from which the foreground is constructed. (One can think of instances m_t as akin to topics in topic models, which are also admixture models). To complete the prior, we fix the distribution $p(\ell|f)$ as fixed binary $L \times 2$ matrix assigning a subset of L pixel classes to foreground and the rest to the background. (For example, we assign first 3 classes to foreground and the remaining 5 to the background for a total of L=8 pixel classes). Therefore,

$$p(\ell_{i,t} = \ell) = \sum_f p(\ell|f) \sum_{m_t} p(f_{i,t} = f|m_t)p(m_t) \ . \tag{F.5}$$

We can now select the instances m_t in each frame by optimizing the free energy with this prior over $p(m_t)$. The procedure involves standard variational inference of the posterior distribution over possible instances m_t for each pixel i in frame t which involves the posterior $q_t(\ell_{i,t}|x_{i,t})$. In practice we found that it is enough to do this inference once, using the network q_{t-1} estimated in the previous frame.

This requires the inference of m_t for each pixel i:

$$s_i(m_t) \propto \exp\left(\sum_i \sum_{\ell,f} p(\ell|f) q_t(\ell_{i,t} = \ell|x_{i,t}) \log p(f_{i,t} = f|m_t)p(m_t)\right), \tag{F.6}$$

and then optimizing p_{m_t} as the count of times each instance is used,

$$p(m_t) \propto \sum_i s_i(m_t) \ . \tag{F.7}$$

Selection of instances m_t in frame t therefore involves comparing the predictions from the network $q_t(\ell_{i,t} = \ell|x_{i,t})$ grouped into foreground/background segmentation with the foreground/background segmentation for different instances from Mask R-CNN, and making a selection of a subset (probabilistically in $p(m_t)$) based on which instances most overlap with the predictions from network q_t. While the above two equations should in principle be iterated, and iterated with updates to network $q_t(\ell_{i,t} = \ell|x_{i,t})$, we found that in practice it is sufficient to just select the instances m_t based on their intersection with the network predictions once, at the very beginning, to make a soft fixed prior, and leave it to optimizing the prediction network with the **RQ** loss to find confident segmentation (Fig. F.2).

We tested the approach on the DAVIS 2016 dataset [Perazzi et al., 2016]. The dataset is comprised of 50 unique scenes, accompanied by per-pixel foreground/background segmentation masks. The objective is to produce foreground segmentation masks for all frames in a scene, given only the ground truth annotations of the first frame (Semi-Supervised). We evaluated our method on the 20-scene validation set at 480p resolution.

The network q used in this experiment combines both the pixel intensities and spatial position information for its predictions. At each pixel location i, j, we augment the intensity information with learned Fourier features $[\sin(W[i,j]^T),\ \cos(W[i,j]^T)]^T$ [Tancik et al., 2020]. The image and spatial position are first processed separately; A 4-layer, 64-channel, fully-convolutional network with 3×3 kernels, ReLU activations and Batch Normalization produces the image features. A 3-layer, 16-channel, pixel-wise MLP with ReLU activations and Batch Normalization processes the learned Fourier features. These two are concatenated and passed through a single 3×3 convolution-ReLU-Batch Normalization layer before being mapped to output predictions. We also experimented with adding optical flow as another auxiliary input to the network.

For each scene, the network q_0 is trained on the first frame, using the given ground truth annotations split uniformly between 3 foreground and 5 background classes as prior, for 300 iterations. This network is then used to predict the foreground pixels in the next frame and after computing the intersection over union between the predicted foreground pixels and the Mask R-CNN output masks, we select masks that overlap more than a pre-specified threshold. The chosen masks are then summated, weighted by their Mask R-CNN confidence scores (0-1), to form the prior for the next frame. The process of selecting masks from the Mask-RCNN predictions and forming the prior for a frame is showcased in Figure F.3. The network q_0 is then fine-tuned for 10 iterations to obtain q_1 and this process repeats for all subsequent frames. We used the Adam optimizer, with a starting learning rate of 10^{-3} for the first frame, reduced to 10^{-5} for fine-tuning, and trained with batches of 128 64×64 patches.

To infer the foreground pixels we start with a Mask R-CNN pre-trained on the COCO dataset. Then, for each scene we only require ∼1min of training time on the ground truth-annotated first frame and ∼3s per every following frame for the entire process of forming the prior and inferring the foreground pixels. We do not train on any video data, in contrast to most video object segmentation methodologies that rely on both a pre-trained network on static image datasets (such as COCO) and additionally on offline training on video sequences. In Table F.3 we compare our results on the DAVIS 2016 validation set to other video object segmentation algorithms from 2017 - present.

Table F.3: Jaccard and F1 measures for various algorithms on the video instance segmentation task.

		J			F			
Model	J&F ↑	Mean ↑	Recall ↑	Decay ↓	Mean ↑	Recall ↑	Decay ↓	Year
OSVOS Caelles et al. [2017]	80.2	79.8	93.6	14.9	80.6	92.6	15	2017
MSK Perazzi et al. [2017]	77.55	79.7	93.1	8.9	75.4	87.1	9	2017
OnAVOS Voigtlaender and Leibe [2017]	85.5	86.1	96.1	5.2	84.9	89.7	5.8	2017
Lucid Khoreva et al. [2017]	82.95	83.9	95	9.1	82	88.1	9.7	2017
OSVOS-S Maninis et al. [2018]	86.55	85.6	96.8	5.5	87.5	95.9	8.2	2018
FAVOS Cheng et al. [2018]	80.95	82.4	96.5	4.5	79.5	89.4	5.5	2018
PReMVOS Luiten et al. [2018]	86.75	84.9	96.1	8.8	88.6	94.7	9.8	2018
OSMN Yang et al. [2018]	73.45	74	87.6	9	72.9	84	10.6	2018
AGAME Johnander et al. [2019]	81.85	81.5	93.6	9.4	82.2	90.3	9.8	2019
STM Oh et al. [2019]	89.4	88.7	97.4	5	90.1	95.2	4.2	2019
FEELVOS Voigtlaender et al. [2019]	81.65	81.1	90.5	13.7	82.2	86.6	14.1	2019
CFBI Yang et al. [2020]	89.4	88.3	-	-	90.5	-	-	2020
e-OSVOS Meinhardt and Leal-Taixe [2020]	86.8	86.6	-	-	87	-	-	2020
STCN Cheng et al. [2021]	91.7	90.4	98.1	4.1	93	97.1	4.3	2021
Ours	83.8	84	96.2	8.4	83.6	94.2	10.2	
Ours (+flow)	83.9	83.2	95.5	9.5	84.6	93.3	9.1	

Figure F.2: Example of inferring the foreground mask for a single frame.

F.4 IN-COLLECTION INFERENCE FOR MULTI-DOMAIN LEARNING: RETURN TO LE SÉDUCTEUR

One of the conclusions from our experiments on the EnviroAtlas landcover mapping task (§4.4) is that training a network with the goal of generalizing to new input data is often inferior to simply performing in-collection inference for each domain . In other words, given the collection of pairs $x_i, p_i(\ell)$, learning the posterior q under the implicit posterior model is optimized for resolving ambiguities in that collection, and possibly that collection alone. As pointed out in Malkin et al. [2020], which performs collection inference using large generative models to mine self-similarity among the examples in the collection, this is appropriate when we can expect our data x_i to always come paired with prior beliefs $p(\ell_i)$. It is interesting to reconsider the Seducer example from Fig. 1. The artist created several versions of that painting in differing styles. Fig. F.4 shows that collection inference applied separately to each of these paintings works equally well. However, using a learned q network from one image onto others yields inferior segmentations (Fig. F.5), as the learned network specialized for inference in the data it saw. (A fully generative model would be expected to similarly overtrain on the input data features x_i, as would a supervised neural network trained on hard-labeled pairs (x_i, ℓ_i) due to the domain shift.) Yet, if we know we will always be given collections with beliefs in the form of priors $p_i(\ell)$, local (collection) inference may be all we need.

References

Qianyue Bao, Yang Liu, Zixiao Zhang, Dafan Chen, Yuting Yang, Licheng Jiao, and Fang Liu. Mrta: Multi-resolution training algorithm for multitemporal semantic change detection. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021.

Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

Vivien Cabannnes, Alessandro Rudi, and Francis Bach. Structured prediction with partial labelling through the infimum loss. In *International Conference on Machine Learning*, pages 1230–1239. PMLR, 2020.

S. Caelles, K.K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Neural Information Processing Systems (NeurIPS)*, 2021.

J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Inés Couso and Didier Dubois. A general framework for maximizing likelihood under incomplete data. *International Journal of Approximate Reasoning*, 93:238–260, 2018.

Mordechai Haklay and Patrick Weber. OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing*, 7(4): 12–18, 2008.

Junwei Han, Dingwen Zhang, Gong Cheng, Lei Guo, and Jinchang Ren. Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing*, 53(6):3325–3337, 2014.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *International Conference on Computer Vision (ICCV)*, 2017.

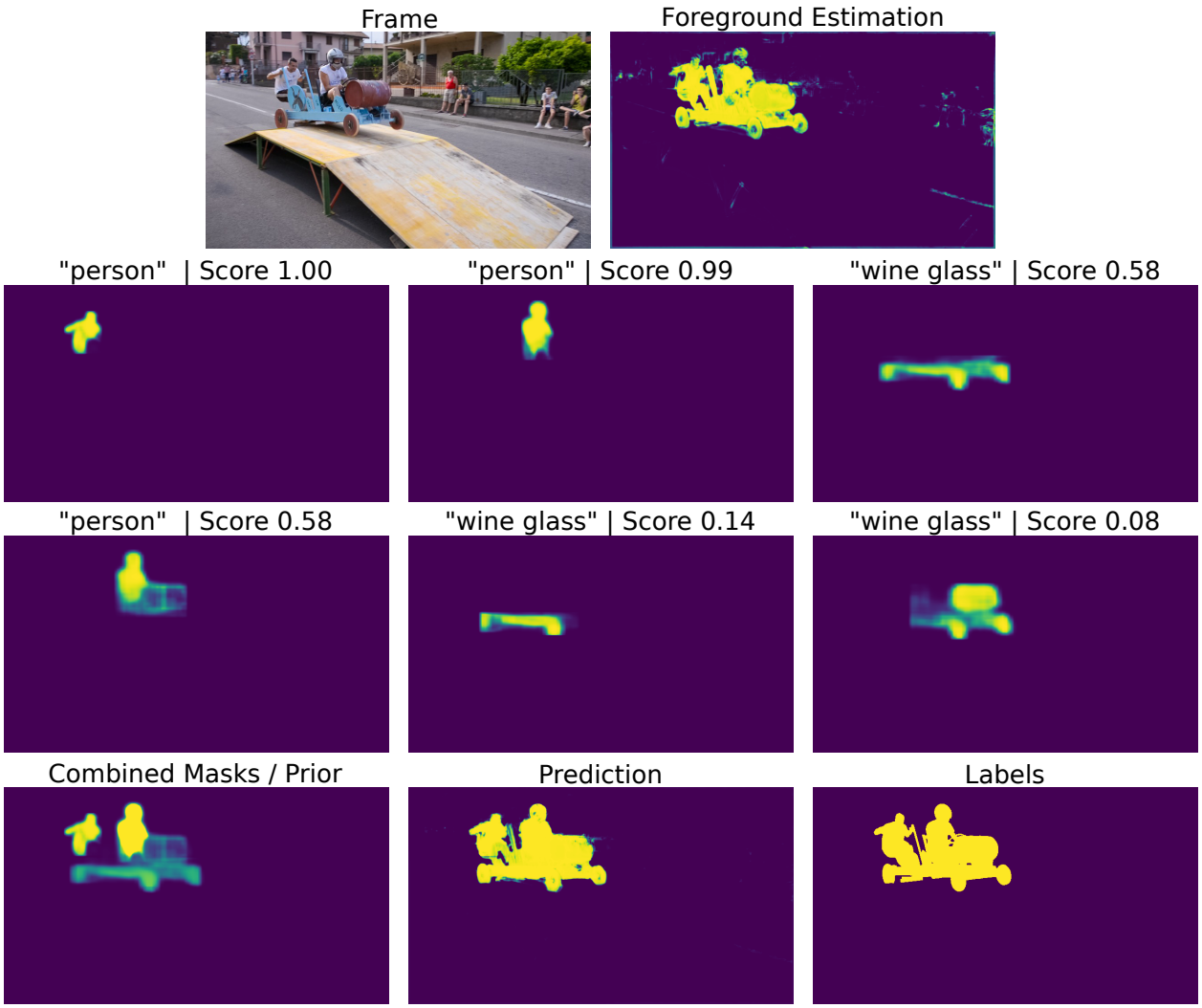

Figure F.3: Video frame segmentation procedure. Starting with a network q_{t-1} trained on frame $t-1$, we apply q_{t-1} on frame t to get a rough foreground estimation (top). By running the pre-trained Mask R-CNN model on frame t and selecting only the masks that overlap with the q_{t-1} prediction we get the candidate object masks (middle). The prior is constructed as the sum of the candidate masks, weighted by their corresponding Mask R-CNN scores (bottom), and q_{t-1} is finetuned on frame t with this prior to produce the predictions (bottom).

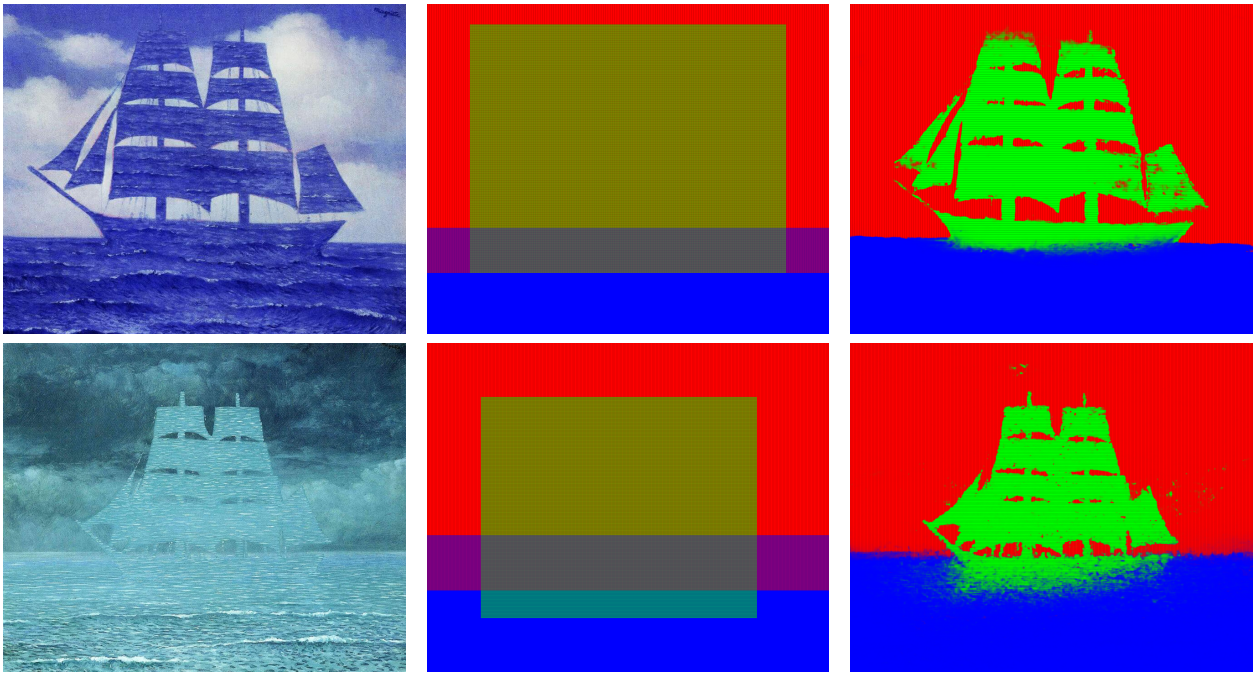

Figure F.4: Two additional versions of *Le séducteur* (left), hand-made priors (middle) and inferred segmentations (right).

(a) (b) (c)

Figure F.5: Result of applying a network q trained to infer (b), on all three *Le séducteur* versions.

Jerónimo Hernández-González, Inaki Inza, and Jose A Lozano. Weak supervision and other non-standard classification problems: a taxonomy. *Pattern Recognition Letters*, 69:49–55, 2016.

Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and R M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268 5214:1158–61, 1995.

Le Hou, Vu Nguyen, Ariel B Kanevsky, Dimitris Samaras, Tahsin M Kurc, Tianhao Zhao, Rajarsi R Gupta, Yi Gao, Wenjin Chen, David Foran, et al. Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern Recognition*, 2019.

Eyke Hüllermeier. Learning from imprecise and fuzzy observations: Data disambiguation through generalized loss minimization. *International Journal of Approximate Reasoning*, 55(7):1519–1534, 2014.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

Rong Jin and Zoubin Ghahramani. Learning with multiple labels. *Neural Information Processing Systems (NeurIPS)*, 2002.

Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2017.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. NLNL: Negative learning for noisy labels. *International Conference on Computer Vision (ICCV)*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Zhuohong Li, Fangxiao Lu, Hongyan Zhang, Guangyi Yang, and Liangpei Zhang. Change cross-detection based on label improvements and multi-model fusion for multi-temporal remote sensing images. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *European Conference on Computer Vision (ECCV)*, 2014.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. *Asian Conference on Computer Vision (ACCV)*, 2018.

Nikolay Malkin, Caleb Robinson, Le Hou, Rachel Soobitsky, Jacob Czawlytko, Dimitris Samaras, Joel Saltz, Lucas Joppa, and Nebojsa Jojic. Label super-resolution networks. *International Conference on Learning Representations (ICLR)*, 2019.

Nikolay Malkin, Anthony Ortiz, and Nebojsa Jojic. Mining self-similarity: Label super-resolution with epitomic representations. *European Conference on Computer Vision (ECCV)*, 2020.

Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

Tim Meinhardt and Laura Leal-Taixe. Make one-shot video object segmentation efficient again. *Neural Information Processing Systems (NeurIPS)*, 2020.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *International Conference on Learning Representations (ICLR)*, 2017.

Nam Nguyen and Rich Caruana. Classification with partial labels. *Knowledge Discovery and Data Mining (KDD)*, 2008.

Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *International Conference on Computer Vision (ICCV)*, 2019.

F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Brian R Pickard, Jessica Daniel, Megan Mehaffey, Laura E Jackson, and Anne Neale. EnviroAtlas: A new geospatial tool to foster ecosystem services science and resource management. *Ecosystem Services*, 14:45–55, 2015.

Andrew Pilant, Keith Endres, Daniel Rosenbaum, and Gillian Gundersen. US EPA EnviroAtlas meter-scale urban land cover (MULC): 1-m pixel land cover class definitions and guidance. *Remote Sensing*, 12(12), 2020.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*. NIH Public Access, 2017.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Neural Information Processing Systems (NIPS)*, 2016.

Caleb Robinson, Le Hou, Nikolay Malkin, Rachel Soobitsky, Jacob Czawlytko, Bistra Dilkina, and Nebojsa Jojic. Large scale high-resolution land cover mapping with multi-resolution data. *Computer Vision and Pattern Recognition (CVPR)*, 2019.

Caleb Robinson, Anthony Ortiz, Nikolay Malkin, Blake Elias, Andi Peng, Dan Morris, Bistra Dilkina, and Nebojsa Jojic. Human-machine collaboration for fast land cover mapping. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Adam J Stewart, Caleb Robinson, Isaac A Corley, Anthony Ortiz, Juan M Lavista Ferres, and Arindam Banerjee. Torchgeo: deep learning with geospatial data. *arXiv preprint arXiv:2111.08872*, 2021.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Neural Information Processing Systems (NeurIPS)*, 2020.

Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *British Machine Vision Conference (BVMC)*, 2017.

Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. FEELVOS: Fast end-to-end embedding learning for video object segmentation. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K. Katsaggelos. Efficient video object segmentation via network modulation. *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. *European Conference on Computer Vision (ECCV)*, 2020.

Yao Yao, Jiehui Deng, Xiuhua Chen, Chen Gong, Jianxin Wu, and Jian Yang. Deep discriminative CNN with temporal ensembling for ambiguously-labeled image classification. *Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

Fei Yu and Min-Ling Zhang. Maximum margin partial label learning. In *Asian conference on machine learning*, pages 96–111. PMLR, 2016.

Xiao Zhang, Yixiao Ge, Yu Qiao, and Hongsheng Li. Refining pseudo labels with clustering consensus over generations for unsupervised object re-identification. *Computer Vision and Pattern Recognition (CVPR)*, 2021.

Zhuo Zheng, Yinhe Liu, Shiqi Tian, Junjue Wang, Ailong Ma, and Yanfei Zhong. Weakly supervised semantic change detection via label refinement framework. *International Geoscience and Remote Sensing Symposium (IGARSS)*, 2021.

Naiyun Zhou, Xiaxia Yu, Tianhao Zhao, Si Wen, Fusheng Wang, Wei Zhu, Tahsin Kurc, Allen Tannenbaum, Joel Saltz, and Yi Gao. Evaluation of nucleus segmentation in digital pathology images through large scale image synthesis. In *Medical Imaging 2017: Digital Pathology*, volume 10140. International Society for Optics and Photonics, 2017.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53, 2018.

Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*, 2020.