Resolving Label Uncertainty with Implicit Posterior Models

Abstract

We propose a method for jointly inferring labels across a collection of data samples, where each sample consists of an observation and a *prior belief* about the label. By implicitly assuming the existence of a generative model for which a differentiable predictor is the posterior, we derive a training objective that allows learning under weak beliefs. This formulation unifies various machine learning settings; the weak beliefs can come in the form of noisy or incomplete labels, likelihoods given by a different prediction mechanism on auxiliary input, or common-sense priors reflecting knowledge about the structure of the problem at hand. We demonstrate the proposed algorithms on diverse problems: classification with negative training examples, learning from rankings, weakly and self-supervised aerial imagery segmentation, co-segmentation of video frames, and coarsely supervised text classification.

1 INTRODUCTION

In prediction problems, coarse and imprecise sources of input can provide rich information about labels. Negative labels (what an instance is *not*), rankings (which of two instances is larger), or coarse labels (aggregated by taxonomy or geography) give clues on what the ground truth label of an instance *might* be, but not what it *is* directly. We consider a collection of data samples, indexed by i, consisting of observations (features) x_i and corresponding sample-specific *prior beliefs* about their latent label variables, $p_i(\ell)$. This paper proposes algorithms to **resolve the uncertainty in these prior beliefs** by jointly inferring an assignment of target labels ℓ_i and a model that predicts ℓ_i given x_i.

Partial or aggregate annotations and auxiliary data sources are often more widely available and convenient to collect than "ground-truth" or high-resolution labels, but they are not readily used by discriminative learners. Supervision from probabilistic targets can result in uncertain predictions (§2). Most approaches to resolve these uncertainties involve iterative generation of hard pseudolabels [Zhang et al., 2021] or loss functions promoting low entropy of predictions [Nguyen and Caruana, 2008, Yu and Zhang, 2016, Zou et al., 2020, Yao et al., 2020]. Typically, these approaches are application-specific [Han et al., 2014, Zheng et al., 2021, Bao et al., 2021, Li et al., 2021]. In many settings, fusing weak input data into a probability distribution over classes is a more natural alternative to transforming the weak input into hard labels [Mac Aodha et al., 2019]. Further connections and comparisons to prior work are made throughout this paper and synthesized in §C and §D.

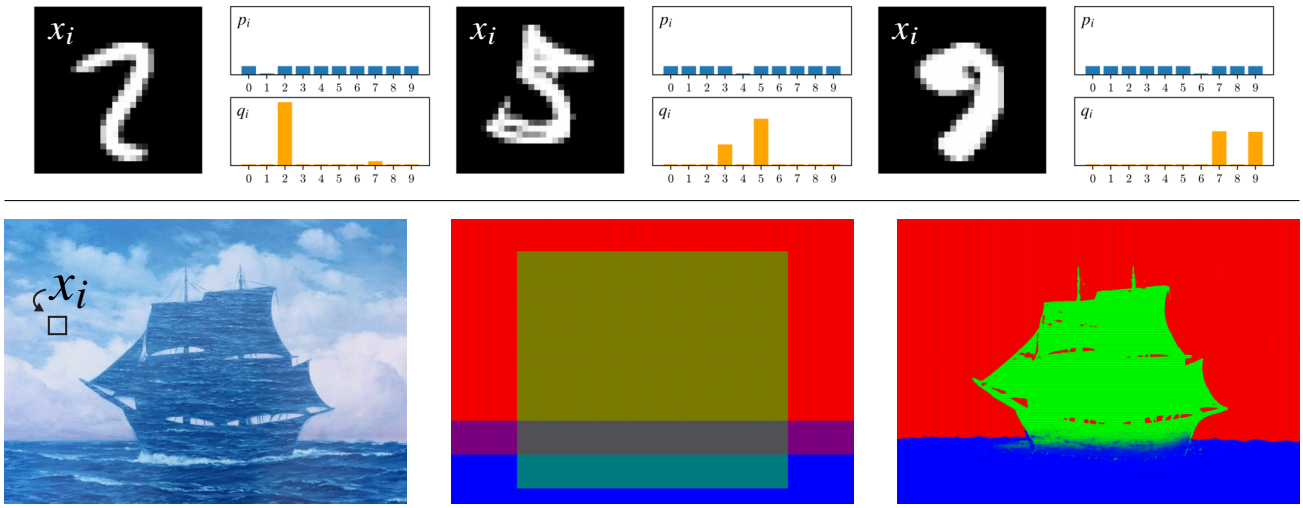
Our key modeling insight (§2.1) is to identify the output distribution of a discriminative model, a feed-forward neural network q, with an approximate posterior over latent variables in an *generative* model of features, of which the given prior belief is a part. Bayesian reasoning about the generative model and its posterior makes it possible to learn the inference network *without instantiating the full generative model*, while reaping the benefits of generative modeling: high certainty in the posterior under soft priors and rich opportunities to model structure in the prior beliefs.

Prior beliefs about labels can arise from many sources (§3). We validate the effectiveness of our approach with experiments (§4, §F) on multiple domains and data modalities that highlight: prior beliefs as a natural way to fuse weak inputs, graceful degradation of performance with increasingly noisy or incomplete inputs, and comparison with explicitly generative modeling approaches.

2 BACKGROUND AND APPROACH

Two motivating examples. Two illustrative examples are shown in Fig. 1. In the first example, the x_i are 784-dimensional vectors representing 28×28 MNIST digits. We aim to infer the digit classes $\ell_i \in \{0, 1, ..., 9\}$ for all images in the given collection based on data in which we are given

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.



(a) $\{x_i\}$: *Le séducteur*, René Magritte (b) $p_i(\ell)$: *Boat Prior*, anonymous artist (c) $q_i(\ell)$: Inferred segmentation

Figure 1: **Above:** Inference of latent MNIST digit classes with negative label supervision using a small CNN trained on the **RQ** criterion (§2.1). **Below:** (a) Joint inference of latent pixel classes in an image. (b) Prior beliefs $p_i(\ell)$ over three classes – sky (red), boat (green), water (blue) – are manually set. (c) A small CNN trained on $(x_i, p_i(\ell))_i$ infers the posterior classes.

just one *negative* label per sample, i.e., the prior beliefs $p_i(\ell)$ (top row) are uniform over all classes except for one incorrect class. The procedure described in this paper produces inferred distributions over labels (bottom row) that are usually peaky and place the maximum at the correct digit 97% of the time (see Fig. 3 and §4.1).

In the second example, the observations $\{x_i\}_{i \in \text{pixels}}$ are image patches centered around each pixel coordinate i in a Surrealist painting, with patch size (11×11) equal to the receptive field of a 5-layer convolutional neural network used in our inference procedure. The prior beliefs $p_i(\ell)$ are distributions over 3 classes (sky, boat, water) depending on the coordinate i. The joint inference of all labels in this image yields a feasible segmentation despite the high similarity in colors and textures (see §F.4 for more details).

These examples illustrate the problem of training on weak beliefs, which is often encountered in some form in machine learning. Weak supervision, semi-supervised learning, domain transfer, and integration of modalities are all settings where coarse, partial, or inexact sources of data can provide rich information about the state of a prediction instance, though not always a "ground truth" label for each instance. An inference technique that uses weak beliefs as the sole source of supervision needs to estimate statistical links between observations x_i and corresponding latents ℓ_i. These links should simultaneously be highly confident (i.e., lead to low entropy in the posterior distributions) and explain the varying prior beliefs, which typically have low confidence (high entropy in the prior distributions).

Supervised learning on prior beliefs. Supervised learning models, including many neural nets, are typically trained to minimize the cross-entropy $-\sum_i \sum_\ell p_i^d(\ell) \log q_i(\ell)$ between a "hard" distribution over labels with $p_i^d(\ell) \in \{0, 1\}$ and the distribution $q_i(\ell) = q(\ell|x_i; \theta)$ output by a predictor q using data features x_i. This is equivalent to minimizing the KL divergence $\sum_i \mathrm{KL}(p_i^d \| q_i)$, minimized when the two distributions $p_i^d(\ell)$ and $q_i(\ell)$ are equal. Thus, when $p_i^d(\ell)$ is a "softer" prior over latent labels, $p_i(\ell)$, the trained model q will reflect this, and also be highly uncertain.

Transforming soft labels into hard training targets, (e.g. training on $\mathbb{1}[\ell = \arg\max_\ell p_i^d(\ell)]$), can introduce the opposite bias. In these cases, the cost would be minimized by predictions with zero entropy, but learning such a prediction function faces difficulty with overconfident labels which are often wrong, and the possibility that certain labels often receive substantial weight in the prior, but never the maximum. These issues are illustrated in Fig. E.3.

Generative modeling resolves the prior's uncertainty. The approach to classification problems through *generative* modeling, instead of targeting the conditional probability of latents given the data features, assumes that there is a forward (generative) distribution $p(x_i|\ell)$ and optimizes the log-likelihood of the observed features, $\sum_i \log(x_i) = \sum_i \log \sum_\ell p(x_i|\ell) p_i(\ell)$, with respect to the parameters of that distribution. The posterior under the model $q(\ell|x_i) \propto p(x_i|\ell) p_i(\ell)$ is then used to infer latent labels for individual data points [Seeger, 2002]. The generative modeling approach does not suffer from uncertainty in the posterior distribution over latents given the input features, even when the priors $p_i(\ell)$ are soft. (Recall that the posterior distributions in a mixture of high-dimensional Gaussians are often peaky even when the priors are flat.)

However, expressive generative models are typically harder and more expensive to train compared to supervised neural networks, as they often require sampling (e.g., sampling of the posterior in variational auto-encoders [VAEs; Kingma and Welling, 2014] and sampling of the generator in GANs [Goodfellow et al., 2014]). Furthermore, the modeling often requires doubling of parameters to express both the forward (generative) model *and* the reverse (posterior) model. And, in case of GANs, the learning algorithms may not even cover all modes in the data, which would prevent joint inference for *all* data points. (See §D for further discussion.)

2.1 OPTIMIZING IMPLICIT POSTERIOR MODELS

Suppose that there exists a generative model $p(x|\ell)$ of observed features conditioned on latent labels. Optimization of the log-likelihood of observed features, $\sum_i \log p(x_i) = \sum_i \log(\sum_\ell p(x_i|\ell)p_i(\ell))$, can be achieved by introducing a variational posterior distribution $q(\ell|x_i)$ over the latent variable for each instance x_i and minimizing the free energy (a negated evidence lower bound (ELBO)), defined as

$$-\sum_i \sum_\ell q(\ell|x_i) \log \frac{p(x_i|\ell)p_i(\ell)}{q(\ell|x_i)} \geq -\sum_i \log p(x_i). \quad (1)$$

Minimizing the free energy involves estimating both the forward distributions $p(x_i|\ell)$ and the posteriors $q(\ell|x_i)$.

One could parametrize both $p(x|\ell)$ and $q(\ell|x)$ as functions $p(x|\ell, \theta_p)$ and $q(\ell|x, \theta_q)$ using neural networks, as done by VAEs (although VAEs use continuous latent variables ℓ and do not involve sample-specific priors). However, in our algorithms, we only parametrize $q(\ell|x;\theta)$ as a neural network taking input x and producing a distribution over ℓ. The generative conditional $p(x_i|\ell)$ is defined only on data points x_i and is calculated by minimizing (1) for fixed $q(\ell|x)$, subject to the constraint that $\sum_i p(x_i|\ell) = 1$ for all ℓ. The optimum is achieved by:

$$p(x_i|\ell) = a_{i,\ell} = \frac{q(\ell|x_i)}{\sum_j q(\ell|x_j)} \,. \quad (2)$$

Here the generative conditional $p(x|\ell)$ is not fully specified for all values x. Rather, it is represented as a matrix of numbers $a_{i,\ell}$ describing the conditional probabilities of different values of x_i given different latent labels ℓ. The probabilities $p(x_i|\ell)$ are greater for the data points i for which $q(\ell|x_i)$ is more certain, relative to how popular assignment to class ℓ is across data points (denominator in (2)).

In our formulation, q plays the role of a variational posterior, but *implicitly*, in a generative model consisting of varying instance-specific priors $p_i(\ell)$ and a complex conditional $p(x|\ell)$ that is never fully estimated, but is instead maximized for the data points studied. The full link between x and ℓ is left entirely to the neural network q to capture explicitly.

In variational methods, the free energy (1) is usually rewritten as $\sum_i \mathrm{KL}(q(\ell|x_i)\|r_i(\ell))) - \log p(x_i)$, where r is the posterior of the forward model, i.e., for the points i, $r_i(\ell) \propto p_i(\ell)p(x_i|\ell)$. The minimization of free energy then reduces to minimizing the KL divergence between r and q.

We define $q_i(\ell) = q(\ell|x_i;\theta)$. After our reduction of $p(x_i|\ell)$ to the auxiliary matrix in (2), the posterior r has the form

$$r_i(\ell) = c_i \cdot p_i(\ell)p(x_i|\ell) = c_i \frac{p_i(\ell)q_i(\ell)}{\sum_j q_j(\ell)} \,, \quad (3)$$

where c_i are scalars making $\sum_\ell r_i(\ell) = 1$. For each instance i we have two outputs: the direct model outputs of the variational posterior q_i and their *implied posterior* r_i, which is computed by multiplying the renormalized model outputs with the provided prior at each instance as in (3). Using these two outputs, we can optimize a single set of model parameters θ to minimize (1):

$$\min_\theta \sum_i \mathrm{KL}(q_i\|r_i) = \quad (4)$$
$$\min_\theta \sum_i \mathrm{KL}\Bigg(\underbrace{\Big(q(\ell|x_i;\theta)\Big)_\ell}_{\text{model output with input } x_i} \Bigg\| \Bigg(c_i \cdot \overbrace{p_i(\ell)}^{\text{per-instance priors}} \; \overbrace{\frac{q(\ell|x_i;\theta)}{\sum_j q(\ell|x_j;\theta)}}^{\text{model output normalized per-class as in Eq. (2)}} \Bigg)_\ell \Bigg) \,.$$

While (4) optimizes the free energy (1) by minimizing $\mathrm{KL}(q_i\|r_i)$, minimizing $\mathrm{KL}(r_i\|q_i)$ would also find solutions for which the direct model and its implied posterior are close. We propose to optimize either of these two objectives with respect to the model parameters θ by gradient steps. We iterate over data instances x_i with priors $p_i(\ell)$:

(1) Calculate the distributions r_i in terms of q_i as in (3).

(2) Update the parameters of q with a gradient step:
- Option **QR**: $\theta \leftarrow \theta - \eta\nabla_\theta \sum_i \mathrm{KL}(q_i\|r_i)$.
- Option **RQ**: $\theta \leftarrow \theta - \eta\nabla_\theta \sum_i \mathrm{KL}(r_i\|q_i)$.

Gradients of the objectives are propagated to the expression of r_i through q_i (see (4) and Fig. 2). Both losses have a stable point when $q_i = r_i$, and **RQ** reduces to the cross-entropy loss in the case of priors which put all mass on one label (e.g. $p_i(\ell) = \mathbb{1}[\ell = \ell_i]$). A discussion of the relative benefits and limitations of the **QR** and **RQ** losses is given in §B, along with practical considerations for implementation.

By defining the conditional model $p(x|\ell)$ as an auxiliary matrix of probabilities $a_{i,\ell}$ that is fit to the reverse model q during learning, we avoid parametrizing both directions of the link $\ell - x$ with highly nonlinear models.[1] We thus manage to keep the problem in the realm of training a single feed-forward network q as a predictor of variables ℓ, but in a way that treats the instance-specific priors $p_i(\ell)$ as they would be in generative modeling.

[1] Note that the use of an auxiliary matrix $a_{i,\ell}$ is also found in expectation-maximization [EM; Dempster et al., 1977], which also minimizes the free energy. However, in EM, it is the variational posterior $q(\ell|x_i)$ which is optimized as a matrix of numbers $a_{i,\ell}$ only on data points, while the *generative* model p is fully parametrized (see Table D.1).

```python
# log_q : ( batch_size, n_classes ) log-likelihoods from model
# prior : ( batch_size, n_classes ) prior likelihoods

def ce_loss(log_q, prior):
    return -(log_q * prior).sum(1)

def qr_loss(log_q, prior):
    log_r = (log_q.log_softmax(0) + prior.log()).log_softmax(1)
    return (log_q * log_q.exp()).sum(1) - (log_r * log_q.exp()).sum(1)

def rq_loss(log_q, prior):
    log_r = (log_q.log_softmax(0) + prior.log()).log_softmax(1)
    return (log_r * log_r.exp()).sum(1) - (log_q * log_r.exp()).sum(1)
```

Figure 2: Cross-entropy and implicit **QR** / **RQ** losses in PyTorch. The normalization in (2) is done within batches, rather than across the entire dataset. In practice, this may be sufficient if batches are large and representative of the diversity in the data. Otherwise, the denominator in (3) may need to be updated in an online fashion.

Next, we discuss the consequences of implicitly modeling the generative model p with an auxiliary distribution. Option **QR** uses the KL distance in the direction it appears in (1) and thus guarantees continual improvements in free energy and convergence to a local minimum (with the exception for the effects of stochasticity in minibatch sampling). Substituting r_i from (3), the free energy (1) becomes:

$$F = \sum_{i,\ell} q_i(\ell) \log \left(\sum_j q_j(\ell) \right) - \sum_{i,\ell} q_i(\ell) \log \left(p_i(\ell) \right) \quad (5)$$

This criterion does not encourage entropy of individual q_i distributions, but of their *average*. The second term alone would be minimized if q could put all the mass on $\arg\max_\ell p_i(\ell)$ for each data point, but the first term promotes diversity in assignment of latents (labels) ℓ across the entire dataset. Thus a network q can optimize (5) if it makes different confident predictions for different data points.

To illustrate this, consider the case when all data points have the *same* prior, $p_i(\ell) = p(\ell)$. Then (5) and the **RQ** objective are minimized when $\frac{1}{N}\sum_i q_i(\ell) = p(\ell)$. This can be achieved when q learns a constant distribution $q(\ell|x_i;\theta) = p(\ell)$. But both objectives are also minimized if q predicts only a single label for each data point with high certainty, but it varies in predictions so that the counts of label predictions match the prior.

As demonstrated in Fig. 1 and in our experiments, avoiding degenerate solutions is not hard. We attribute this to two factors. First, the situations of interest typically involve uncertain, but varying priors $p_i(\ell)$ which break symmetries that could lead to predictors ignoring the data features x_i. Second, the neural networks used to model q, and their training algorithms, come with their own constraints and inductive biases. In fact, as discussed in §3 and §F.1, even unsupervised clustering is possible with suitably chosen priors that break symmetry, allowing this approach to be used for self-supervised training. See also §C, §D for more on relationships with other approaches.

3 SOURCES OF LABEL PRIORS

Having detailed our approach for learning from prior beliefs as weak supervision in §2, we now describe a range of machine learning settings where priors $p_i(\ell)$ emerge. These settings are illustrated in experiments in §4 and §F.

Negative or partial labels (§4.1). When we are given a set of equally possible labels L_i for each point data point i, instead of a single label ℓ_i, then we set the prior $p_i(\ell) = \frac{1}{|L_i|}\mathbb{1}[\ell \in L_i]$. An extreme example is when one negative label is given and hence can be "ruled out" (Fig. 1).

Joint labels and learning from rankings (§4.2). Priors may also come in the form of joint distributions over labels of multiple instances. For example, *ranking supervision* – the knowledge of which example in a pair is greater with respect to an ordering of the labels – gives prior beliefs about *pairs* of labels. Suppose our data is organized into pairs of images of digits $T_j = \{x_{j,1}, x_{j,2}\}$, and for each pair we are told which image represents the digit (0–9) which is greater (or equal). This sets a prior $p(\ell_1, \ell_2)$ over pairs of labels in each pair, represented by either an upper or a lower triangular matrix, depending on which digit in the pair is known to be greater, with all nonzero entries equal to $1/55$.

We assume the underlying generative model has the form $p(x_1, x_2|\ell_1, \ell_2) = p(x_1|\ell_1)p(x_2|\ell_2)$. We aim to fit its posterior model $q(\ell|x;\theta)$. For each pair T_j, we have two outputs of the predictor network, $q(\ell_1|x_{j,1})$ and $q(\ell_2|x_{j,2})$, for the two images in the pair. The joint posterior under the generative model is

$$r_j(\ell_1, \ell_2) \propto p(\ell_1, \ell_2)p(x_{j,1}|\ell_1)p(x_{j,2}|\ell_2) \propto \\ \propto \frac{p(\ell_1, \ell_2)q(\ell_1|x_{j,1})q(\ell_2|x_{j,2})}{\sum_j q(\ell_1|x_{j,1}) \sum_j q(\ell_2|x_{j,2})}, \quad (6)$$

and we can now use **QR** or **RQ** loss to fit $q(\ell_1|x_{j,1})$ to the marginal $r_j(\ell_1)$ and $q(\ell_2|x_{j,2})$ to $r_j(\ell_2)$.

Coarse data in weakly supervised segmentation (§4.3, §F.2, §F.4). We often have side information z associated to each instance i that allows setting the priors $p_i(\ell) = p(\ell|z_i)$ for each point directly by hand. These include situations when we have beliefs about labels for different points, as in the *Seducer* example (Fig. 1). Interesting weak supervision settings also arise in remote sensing (§4.3) and medical pathology (§F.2) applications. For example, in a task of segmenting aerial imagery into land cover classes, we often have coarse labels c associated to large *blocks* of pixels, but not the target labels ℓ for individual pixels. If the conditional $p(\ell|c)$ is known, it sets a belief about the high-resolution labels ℓ for pixels in a block of class c.

Fusing models and data sources (§4.4, S4.5). Auxiliary information z may not always come with a known correspondence $p(\ell|z)$. In the land cover mapping problem, auxiliary information includes different modalities and resolutions (road maps, sparse point labels, etc.). While these sources can be fused into a prior by hand-coded rules, the prior may

be more accurately set as the output of a model $p(\ell|z_i)$ *trained* on a separate dataset of points (ℓ_i, z_i). This is especially useful when the data x_i (imagery) is informative about the latents ℓ_i but is prone to domain shift problems, while the auxiliary data z_i does not suffer from domain shift issues but is not sufficient on its own to predict the labels. In a text classification problem, z_i might be the encoding of text x_i by a pretrained language model, and $p(\ell|z_i)$ a noisy distribution over labels given by their likelihoods under the language model as continuations of a prompt.

Priors for self-supervision (§F.1). In §2.1 we discussed the pitfalls of using a constant prior $p_i(\ell) = p(\ell)$ for all data points in training models under the **QR** loss as a potential method for unsupervised clustering. However, in §F.1 we give an example of *joint* learning of the posterior model q and an energy model (Markov random field) on the latent labels ℓ_i that expresses local structure of labels in an image. This results in unsupervised clusterings that are useful in downstream segmentation tasks. Such an approach is an example of a benefit of generative modeling – the possibility of learning of a parametrized distribution over latents – being inherited by implicit posterior models.

Priors with latent structure (§F.3). Implicit posterior modeling allows building hierarchical latent structure into the prior (another benefit of classical generative models), as we demonstrate in §F.3 on a video segmentation task. The prior is an admixture of possible segmentations with a structure similar to Jojic et al. [2009], but using a set of mask proposals $p(\ell_i|m)$ from a Mask R-CNN model [He et al., 2017], indexed by a latent m. The prior is $p_i(\ell) = \sum_m p(\ell_i|m)p(m)$, where $p(m)$, a probabilistic selection of the masks for the admixture in the given frame, is estimated by minimizing the free energy.

4 EXPERIMENTS

4.1 PARTIAL LABELS IN MNIST AND CIFAR-10

In this experiment, we compare algorithms for learning with partial labels on two 10-class image classification datasets, MNIST and CIFAR-10. To each training example x_i, we randomly assign a set N_i of k negative labels, chosen from the 9 labels distinct from the ground truth. The prior $p_i(\ell)$ is set to be uniform over $\ell \notin N_i$ and 0 for $\ell \in N_i$. We vary k from 1 (one negative label per example) to 9 (one-hot prior, full supervision). The data of k negative labels carries $-\log_2(1 - k/10)$ bits of label information; if $k = 1$, 22× less label information than in the fully supervised setting.

For both datasets, the base model q is taken to be a small convolutional network, with four layers of ReLU-activated 3×3 convolutions with stride 2 and a linear map to the 10 output logits (~33k learnable parameters for MNIST, ~34k for CIFAR-10). We experiment with four training losses:

- **CE:** cross-entropy between predictions $q(\ell|x_i;\theta)$ and the prior $p_i(\ell)$.
- **NLL (union):** negative logarithm of the sum of likelihoods assigned by q to labels in $\ell \notin N_i$, or, equivalently, $\log \sum_\ell p_i(\ell)q(\ell|x_i;\theta)$, as done, e.g., by Jin and Ghahramani [2002], Kim et al. [2019].
- The **QR** and **RQ** losses defined in §2.1.

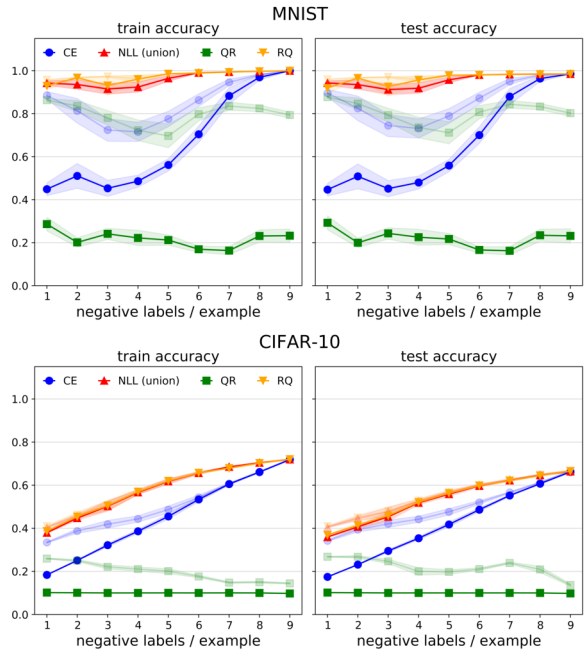


Figure 3: Accuracies of MNIST and CIFAR-10 classifiers trained with varying numbers of negative labels per example; the lighter variant of each color and marker shows the peak accuracy over 300 training epochs. (Average of 10 runs with standard error region.)

The **CE**, **NLL (union)**, and **RQ** loss objectives are equivalent when $k = 9$. The **RQ** and **NLL (union)** losses are equivalent when $\sum_i q_i(\ell)$ is uniform over ℓ, which approximately holds after a sufficient number of training epochs.

All models are trained for 300 epochs on batches of 256 images with the Adam optimizer [Kingma and Ba, 2014] and a learning rate of 10^{-4}. After each epoch, we compute the accuracy of the predictor q on the ground truth labels in the train and test sets. Fig. 4 shows the final train and test set accuracies, as well as the maximum accuracies achieved at any epoch. Reported results are averaged over 10 choices of partial label sets and random initializations.

Models trained on **RQ** loss perform best, with the greatest benefit over **CE** seen for very few negative labels. As expected, the performance of **RQ** and **NLL (union)** is very similar across k. We hypothesize that the small advantage of **RQ** over **NLL (union)** loss can be attributed to regularization in early training. Meanwhile, **QR** performs as well as **CE** for very uncertain priors at the peak epoch (light curves), but its predictions degenerate – usually toward uniform predictions – with longer training.

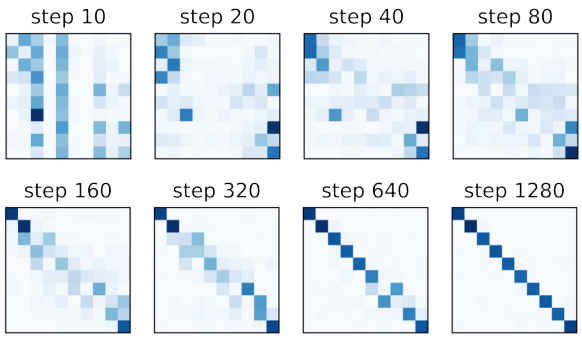


Figure 4: Confusion matrices of MNIST classifiers in the course of training on batches of 128 ranked pairs of digits. The trajectory of convergence to the diagonal shows that uncertainty is first resolved for the digits 0/9, then 1/8, etc.

Table 1: Pixel accuracy and class mean intersection over union on the Chesapeake Land Cover dataset. All models use only coarse NLCD labels as supervision. For our proposed methods, we evaluate both the trained predictor (q_i) and the posterior under the generative model (r_i). The score of the best overall model is **bolded**.

	PA		NY		Chesapeake	
Model	acc %	IoU %	acc %	IoU %	acc %	IoU %
Self-epitomic[a]	**86.2**	67.6	86.4	70.5	86.3	69.7
Hard naïve[b]	85.3	63.0	83.6	59.8	83.6	59.7
QR (q)	85.9	69.3	87.3	73.0	86.4	71.1
QR (r)	**86.2**	**69.9**	**87.9**	**74.4**	**86.8**	**72.1**
RQ (q)	81.5	63.1	77.4	60.2	79.8	62.2
RQ (r)	81.5	63.2	77.5	60.3	79.8	62.4

[a] [Malkin et al., 2020] [b] [Malkin et al., 2019]

4.2 MULTIPLE-INSTANCE SUPERVISION: LEARNING FROM RANKS

We train a CNN of the same architecture as in §4.1 on MNIST, but with the only supervision coming in the form of pairs of images in which it is known which image represents the greater digit. The training set of 60k images is divided into pairs that are fixed throughout the training procedure; each digit appears in exactly one pair. We optimize to match the predictor q with the implicit posterior model (6) using the **RQ** loss. Fig. 4 shows the confusion matrices at initial iterations of training. The learned classifier has 97% accuracy on both training and testing sets, which means that from pairwise comparisons alone, we can group the digit images and place them in order.

4.3 LABEL SUPER-RESOLUTION

We benchmark our method's performance on the Chesapeake Land Cover dataset a large 1m-resolution land cover dataset used previously for label super-resolution [Robinson et al., 2019, Malkin et al., 2019]. It consists of several aligned data layers, including: NAIP (4-channel high-resolution aerial imagery at about 1m/px), NLCD (16-class, 30m-resolution coarse land cover labels), and high-resolution land cover labels (LC) in four classes. The task is to train high-resolution segmentation models, in the four target classes, using only NLCD labels as supervision. The NLCD layer is at 30× lower resolution than the imagery and target labels and follows a different class scheme. Cooccurrence statistics of NLCD classes c and LC labels ℓ are assumed to be known (Fig. E.1).

To form a prior over land cover classes ℓ at each pixel position, we map the NLCD classes to probabilities over the target LC classes using these known cooccurrence counts and apply a spatial blur to reduce low-resolution block artifacts (Fig. 5, "Prior"). We then train small convolutional networks (receptive field 11×11) to predict high-resolution land cover from input imagery. We evaluate both the **QR** and **RQ** variants of our approach on the two states that comprise the "Chesapeake North" test set: Pennsylvania (PA) and New York (NY), and the two states combined, after picking hyperparameters based on an independent validation set in Delaware (details in §E.1.3). A depiction of the data and prediction results is given in Fig. 5.

Table 1 compares our algorithms against the algorithmic technique with the best published performance on the Chesapake dataset, self-epitomic LSR [Malkin et al., 2020] and the hard naïve baseline from Malkin et al. [2019]. Self-epitomic LSR, a generative modeling approach that explicitly produces likelihoods $p(x|\ell)$, analyzes small patches of data by making a large number of comparisons between sampled 7×7 image patches and *all other* image patches. It does not produce a trained feedforward inference model, and the inference procedure is at least an order of magnitude slower than evaluation of our convolutional model. The hard naïve baseline maps the NLCD classes to LC classes based on a given concurrence matrix, then trains a standard semantic segmentation model on these pseudo-labels.

Training on the **QR** loss outperforms (in once case, matches) performance of self-epitomic LSR (Table 1), and the generative model for $p(x|c)$ from (2) is largely consistent with the epitomic generative model (Fig. E.4). Moreover, our methods handle *batched input*, where self-epitomic LSR trains on one data tile at a time. Similar per-tile approaches have been shown to degrade in performance and exhaust computation capacity when training on multiple tiles [Malkin et al., 2020]). Optimization under an implied generative model has the computational advantage of scaling naturally to large training data while maintaining the benefits of leading generative modeling approaches. (See also §F.2.)

4.4 DATA FUSION AND LEARNED PRIORS

In this set of experiments, we augment NLCD with information about the presence of buildings, road networks, and waterbodies/waterways from public sources (see Fig. 6 and

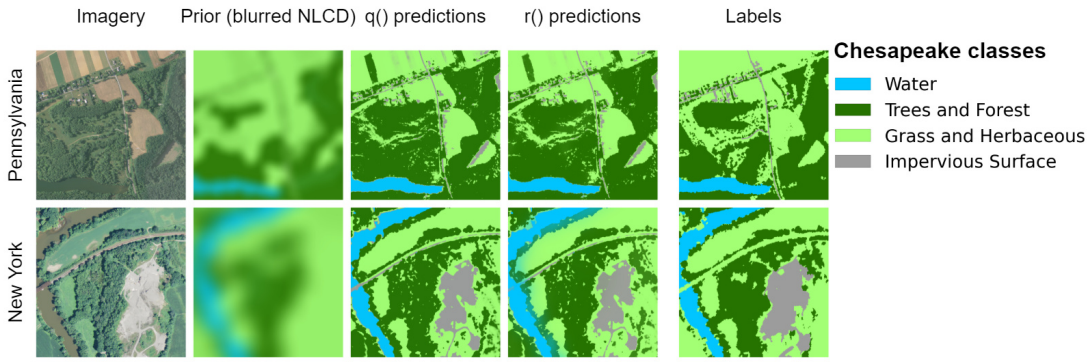


Figure 5: Predictions of models trained with **QR** loss on the NLCD-only prior in the Chesapeake region, shown on regions of 1000×1000 pixels in Pennsylvania and 500×500 pixels in New York.

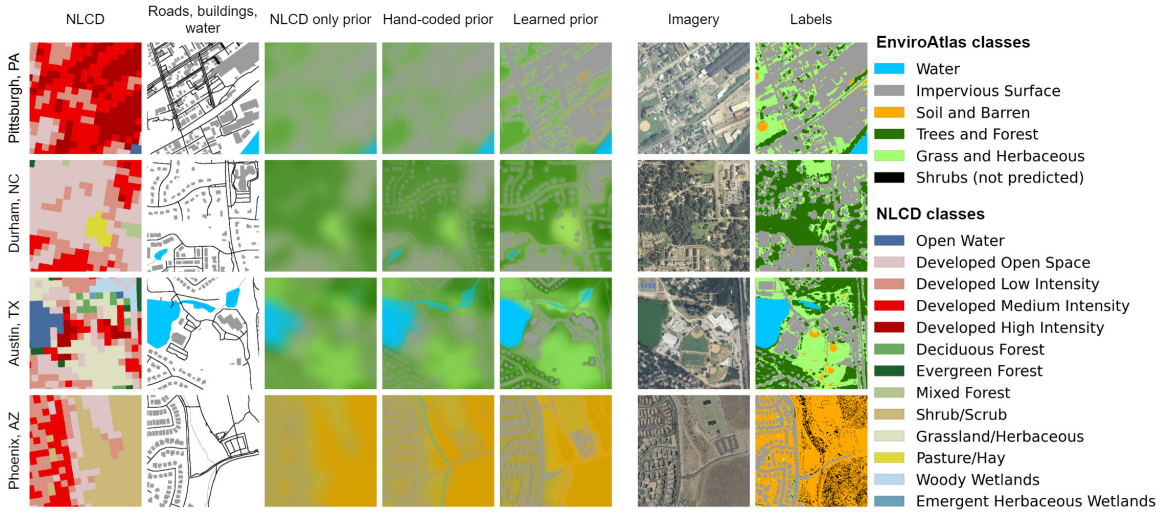


Figure 6: Prior generation for land cover mapping: "NLCD only prior" (§4.3) and "{Hand-coded, Learned} prior" (§4.4).

§E.1.1). To evaluate the ability of models to generalize to across regions, we use 1m 5-class land cover labels from the geographically diverse EnviroAtlas dataset [Pickard et al., 2015] in four cities in the US: Pittsburgh, PA, Durham, NC, Austin, TX, and Phoenix, AZ. The NLCD-based prior model from §4.3 is augmented with the auxiliary information to obtain a hand-coded prior for each image (see §E.1.2). These types of priors can be made everywhere in the United States, while hard 1m-resolution labels are rarely available.

An alternative to performing local inference under such priors is to simply apply supervised models trained on hard labels elsewhere, hoping that the domain shift is tolerable. Table 2 compares the performance of a model (of the same architecture as in §4.3) trained on Pittsburgh high-resolution data (HR) in each of the three other cities with that of models tuned on the hand-coded prior in each other city. The **QR** method trained on the local handmade prior outperforms the HR model in each evaluation city. This may be attributed to the extra data in each city given to our method in the form of prior beliefs. To isolate this effect, we also compare to a high-resolution model that consumes the prior belief to *input* data, concatenated with the NAIP imagery (HR + aux). While the HR + aux model does increase performance substantially from the HR model with NAIP imagery alone as input, the **QR** model remains the highest-fidelity approach in two of the three cities. These results illustrate that information that generalizes across domains may find its best use within a separate model – to build a prior in our setting – and then used to supervise local inference.

In practice, prior beliefs could be crafted by a domain expert to reflect the uniquities in geographic and structural features for each city. We emulate incorporating such context-specific knowledge by training (on a disjoint set of instances) a neural network that consumes the inputs to the handmade prior function (NLCD and auxiliary map data), and predicts high-resolution labels (Fig. 6, "Learned prior"). Alongside structural interactions between the inputs that generalize across cities (e.g., tree canopy supersedes rivers, roads su-

Table 2: Land cover classification experiments for generalizing across cities. In each column, the score of the best model not depending on auxiliary data as input is *italicized* and the score of the best overall model is **bolded**. (A larger set of experimental results is given in Table E.1.)

		Durham, NC		Austin, TX		Phoenix, AZ	
Train region	Model	acc	IoU	acc	IoU	acc	IoU
Pittsburgh (supervised)	HR	74.2	35.9	71.9	36.8	6.7	13.4
	HR + aux	78.9	47.9	77.2	50.5	62.8	24.2
Local (hand-coded prior)	**QR** (q)	78.9	47.7	76.6	49.1	*75.8*	*45.4*
	QR (r)	79.0	48.4	76.6	49.5	**76.2**	**46.0**
Local (learned prior)	**QR** (q)	*79.0*	*48.7*	***79.4***	*51.3*	73.4	42.8
	QR (r)	**79.2**	49.5	79.1	**51.9**	73.6	43.1
Full US[a]	U-Net Large	77.0	**49.6**	76.5	51.8	24.7	23.6

[a] [Robinson et al., 2019]

persede water), the learned prior captures region-specific knowledge (e.g., buildings in Durham tend to have grass surrounding them and trees farther out, while in Austin, this is reversed, and in Phoenix, riverbeds surrounded by barren land are likely to be dry). Using these tailored prior beliefs during **QR** training tends to increase scores (Table 2).

The final row in Table 2 benchmarks the performance of a high-resolution land cover model trained on imagery and labels over the entire contiguous US [Robinson et al., 2019]. This large model takes NAIP, Landsat 8 satellite imagery, and building footprints as inputs. Small, local models with priors created from only weak supervision outperform the US-wide model in all cities. (See §E.1.4 for details.)

4.5 TEXT CLASSIFICATION

This experiment follows the recent work of Mekala et al. [2021] and illustrates the effectiveness of learning on prior beliefs beyond computer vision. We work with a dataset of ∼12k New York Times news articles. Each article belongs to one of 20 fine categories (e.g., 'energy companies', 'tennis','golf'), which are grouped into 5 coarse categories (e.g., 'business', 'sports'). The goal is to train text classifiers that predict fine labels, but only the coarse label for each article is available in training.

Some external knowledge about the fine categories is necessary to resolve the coarse labels into fine labels. Past work on this problem [Meng et al., 2018, Mekala and Shang, 2020, Meng et al., 2020, Wang et al., 2021] has trained supervised models on pseudolabels created by mechanisms such as propagation of seed words and querying large pretrained models. On the other hand, Mekala et al. [2021] create training data by sampling additional *features* (articles) from a finetuned version of the large generative language model GPT-2 [Radford et al., 2019] conditioned on fine categories, then tune a classifier based on the almost equally large model BERT [Devlin et al., 2019] in a supervised manner.

We obtain comparable results using an elementary predictor, far less computation, and no finetuning of massive language models (Table 3). We form a prior $p_i(\ell)$ on the fine class ℓ of each article x_i by querying GPT-2 for the likelihood of each fine category name ℓ compatible with the known coarse label following the prompt "[article text] Topic: " and normalizing over ℓ. We then divide $p_i(\ell)$ by the mean likelihood of ℓ over all articles x_i and renormalize. We represent each article as a vector of alphabetic trigram counts (26^3 features, of which only 8k are ever nonzero) and train a logistic regression with the **RQ** objective against this 'GPT-2 prior'. After ten epochs of training (∼10s on a Tesla K80 GPU), the trained classifier nears or exceeds the performance of models requiring at least 100× longer to train, even excluding the time to generate any pseudo-training data.

Table 3: F1-scores of various models on the coarsely supervised text classification task. The first five rows are taken from Mekala et al. [2021]. The last two rows use the GPT-2 prior defined in §4.5 as weak supervision with cross-entropy and **RQ** loss, respectively (mean of 10 random trials).

	Algorithm	Micro-F1 %	Macro-F1 %
pseudolabeling	WeSTClass[a]	76.23	69.82
	ConWea[b]	73.96	65.03
	LOTClass[c]	15.00	20.21
	X-Class[d]	91.16	81.09
pseudodata	C2F[e]	92.62	**87.01**
GPT-2 prior (trigram features)	prior argmax	86.33	77.61
	CE	87.18	77.90
	RQ	**93.18**	84.26

[a] Meng et al. [2018] [b] Mekala and Shang [2020] [c] Meng et al. [2020] [d] Wang et al. [2021] [e] Mekala et al. [2021]

5 DISCUSSION AND CONCLUSION

In summary, we found that the generative distribution in a free energy criterion can be left implicit to the minimization process in posterior (discriminative) model training. This allowed us to unite the training of neural networks $q(\ell|x_i;\theta)$ for prediction of labels ℓ from features x with the modeling of the prior $p_i(\ell)$, possibly with its own latent structure. Implicit modeling of the conditional generative distributions removes the burden of training accurate (and therefore large or deep) generative models, but still allows natural generative approaches to modeling priors.

Learning a discriminative network q and its implicit posterior model r via the **QR** and **RQ** methods can unify common supervised learning paradigms with realistic label supervision settings, enabling high-fidelity predictions from weak supervision sources carrying far less information. The additional experimental results in §F detail further results for weakly supervised image segmentation, self-supervised learning, and co-segmentation in video data.

Code is attached to this submission (see §A).

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.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Neural Information Processing Systems (NeurIPS)*, 2014.

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.

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.

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

Nebojsa Jojic, Alessandro Perina, Marco Cristani, Vittorio Murino, and Brendan Frey. Stel component analysis: Modeling spatial correlations in image class structure. In *2009 IEEE conference on computer vision and pattern recognition*, pages 2044–2051. IEEE, 2009.

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.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 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.

Oisin Mac Aodha, Elijah Cole, and Pietro Perona. Presence-only geographical priors for fine-grained image classification. *International Conference on Computer Vision (ICCV)*, 2019.

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.

Dheeraj Mekala and Jingbo Shang. Contextualized weak supervision for text classification. *Association for Computational Linguistics (ACL)*, 2020.

Dheeraj Mekala, Varun Gangal, and Jingbo Shang. Coarse2Fine: Fine-grained text classification on coarsely-grained annotated data. *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. Weakly-supervised neural text classification. *International Conference on Information and Knowledge Management*, 2018.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. Text classification using label names only: A language model self-training approach. *Empirical Methods in Natural Language Processing (EMNLP)*, 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.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. 2019.

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.

Matthias Seeger. Learning with Labeled and Unlabeled Data. 2002. URL `https://infoscience.epfl.ch/record/161327/files/review.pdf`.

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

Zihan Wang, Dheeraj Mekala, and Jingbo Shang. X-Class: Text classification with extremely weak supervision. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

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.

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.

A CODE

This submission comes with two code directories that illustrate our algorithms for partial-label learning and weakly supervised segmentation. The provided code is sufficient to reproduce results resembling those in Fig. 1.

We also attach code for land cover mapping experiments. Unfortunately, anonymity rules prevent us from releasing the data required to execute the code and reproduce results. The full data will be released upon publication.

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 the caption of Figure 2, should mini-batch training become an issue in future implementations, it may be beneficial to estimate the denominator of Equation (2) across multiple batches.

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 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, makes them less suitable for problems with more varied forms of uncertainty encoded in priors. Another line of work studies the generation and use of pseudo labels 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 temporal 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]. As discussed in §4.1, NLL and RQ are equivalent when $\sum_i q_i(\ell)$ is uniform over ℓ, evidenced by the comparable performance between the two in Figure 3. 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

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	$q(\ell\|x;\theta_q)$	$a_{i,\ell}$	$q(\ell\|x;\theta_q)$

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 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).

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][2]. 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`.

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 & 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 $\times$ 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.

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

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 $30\text{m} \times 30\text{m}$ 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 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.

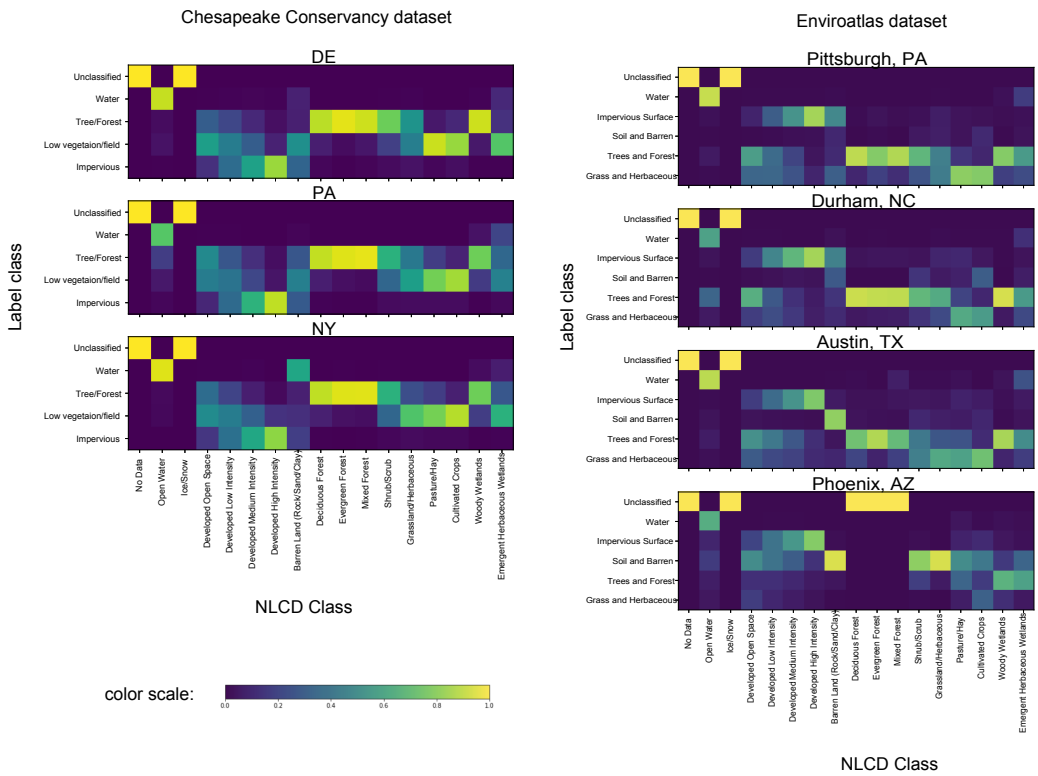


Figure E.1: Cooccurrence matrices between NLCD classes and high resolution land cover labels for each region we study.

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 & 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	HR	89.3	69.3	74.2	35.9	71.9	36.8	6.7	13.4
(supervised)	HR + aux	89.5	70.5	78.9	47.9	77.2	50.5	62.8	24.2
Same as test	**QR** (q)	80.5	56.8	78.3	44.4	79.2	50.5	75.2	29.5
(random	**QR** (r)	80.7	57.5	78.5	46.4	79.7	52.0	75.9	33.8
initialization)	**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	**QR** (q)	80.6	58.5	78.9	47.7	76.6	49.1	75.8	45.4
(pretrained	**QR** (r)	80.6	58.7	79.0	48.4	76.6	49.5	76.2	46.0
in Pittsburgh)	**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	**QR** (q)	82.4	63.7	79.0	48.7	79.4	51.3	73.4	42.8
(learned prior)	**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

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

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

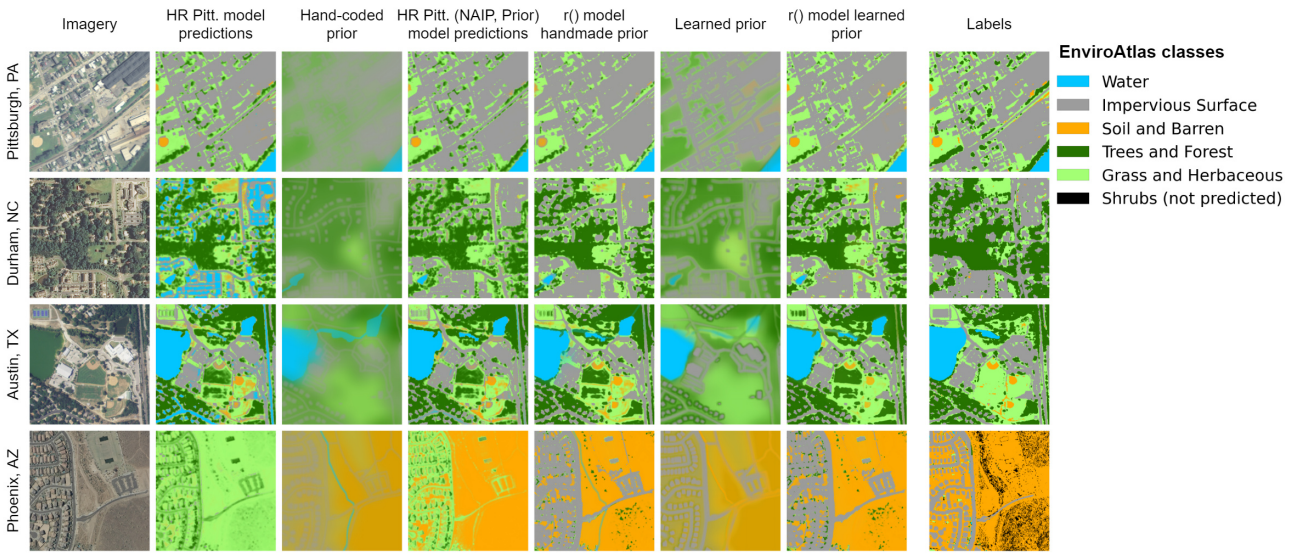


Figure E.2: Example predictions on the hand-coded and learned prior in each EnviroAtlas city we study.

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.

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.

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

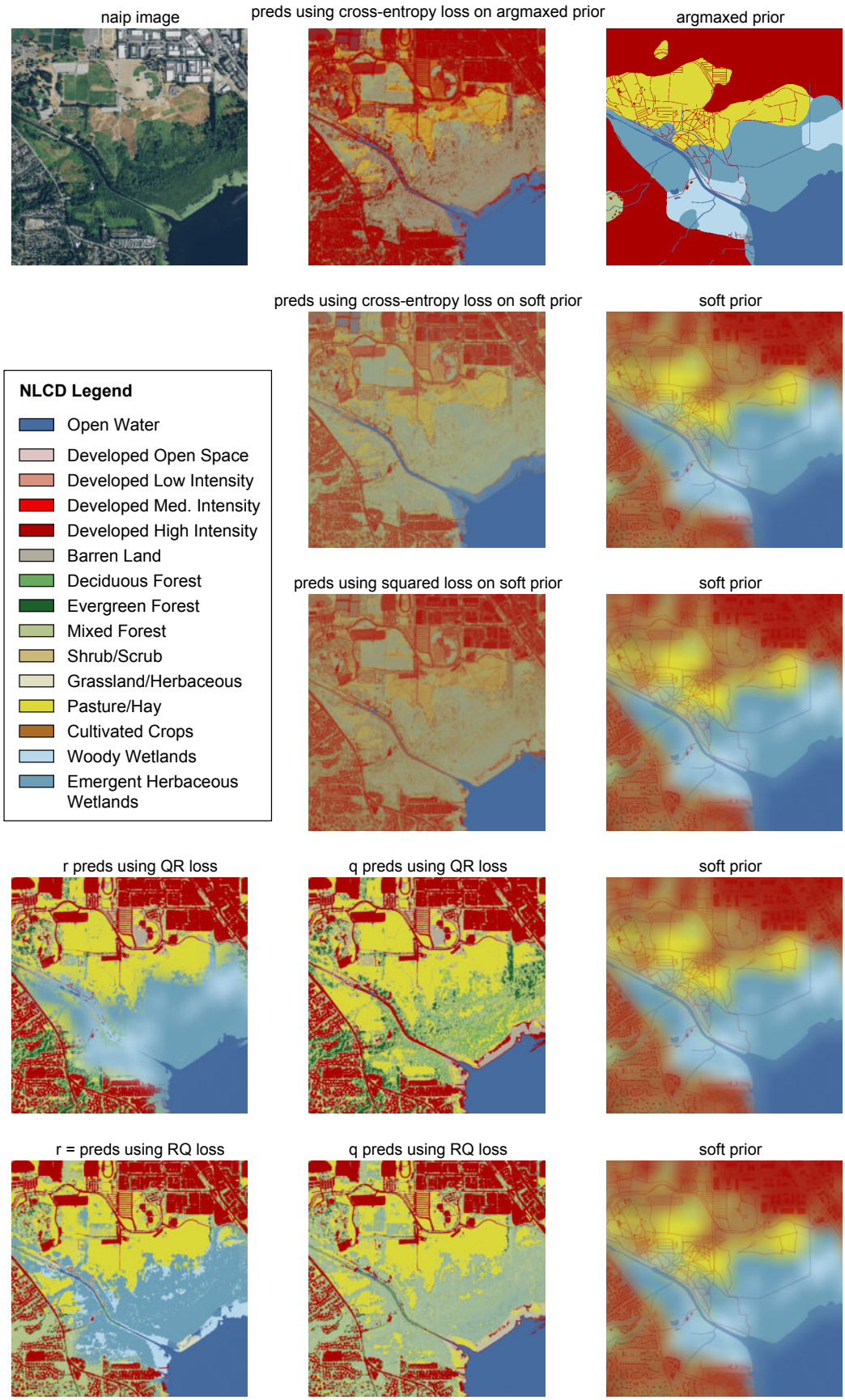


Figure E.3: Comparison of different loss functions on hard and soft prior.

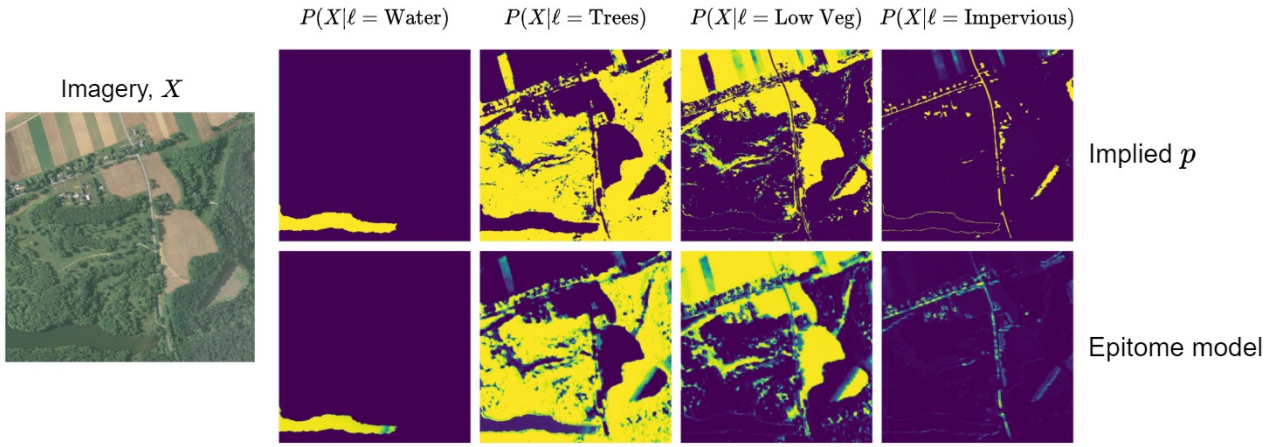


Figure E.4: Comparison of forward model likelihoods under the generative model trained with **QR** loss (above) and the likelihood under an epitome model [Malkin et al., 2020] for part of a test tile from §4.3.

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)

$$F = -\sum_{i,\ell} q_i(\ell) \log p_i(\ell) + \sum_{i,\ell} q_i(\ell) \log \left(\sum_j q_j(\ell) \right) . \tag{7}$$

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:

$$F = -\sum_{i,\ell} q_i(\ell) \log q_i(\ell) + \sum_{i,\ell} q_i(\ell) \log \left(\sum_j q_j(\ell) \right) . \tag{8}$$

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 (8) 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,

$$p(\{\ell_i\}) \propto \exp \sum_i \phi(\ell_i, \{\ell_j\}_{j \in N_j}). \tag{9}$$

We experimented with potentials of the form

$$\phi(\ell_i = \ell, \{\ell_j\}_{j \in N_j}) = \gamma_\ell + \alpha_\ell \frac{1}{|S_i|} \sum_{j \in S_i} \mathbb{1}[\ell = \ell_j] + \beta_\ell \frac{1}{|L_i|} \sum_{j \in L_i} \mathbb{1}[\ell = \ell_j], \tag{10}$$

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 (8) 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 (8).

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).

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 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.[3]

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** 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

[3] We experimented with setting $p_i(\ell|c)$ to conditional likelihoods estimated from a held-out set and with simply setting $p_i(\ell = 1|c = 0) = 0.05$, $p_i(\ell = 1|c = 1) = 0.15, \ldots, p_i(\ell = 1|c = 9) = 0.95$. 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	SV[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]

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,

$$p(\ell_{i,t} = \ell) = \sum_f p(\ell|f) \sum_{m_t} p(f_{i,t} = f|m_t)p(m_t) \ . \tag{11}$$

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:

$$s_i(m_t) \propto \exp\left(\sum_i \sum_{\ell,f} p(\ell|f) q_t(\ell_{i,t} = \ell|x_{i,t}) \log p(f_{i,t} = f|m_t)p(m_t) \right), \tag{12}$$

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

$$p(m_t) \propto \sum_i s_i(m_t) \ . \tag{13}$$

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.

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

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 & 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 & 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	

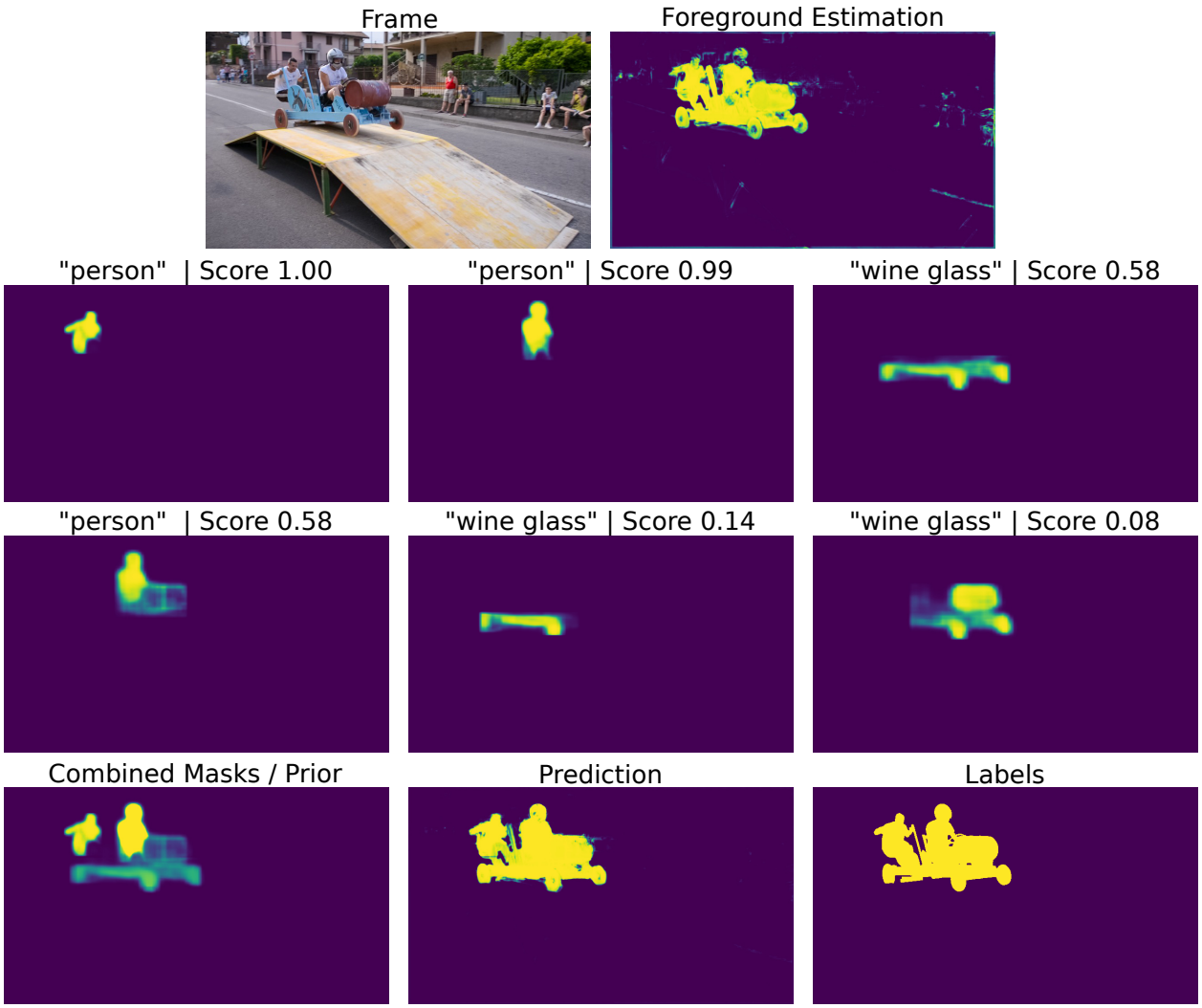


Figure F.3: Video frame segmentation procedure. Starting with a network q_{t-1} trained on frame $t-1$, we apply q_{t-1} on frame t to get a rough foreground estimation (top). By running the pre-trained Mask R-CNN model on frame t and selecting only the masks that overlap with the q_{t-1} prediction 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).

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.

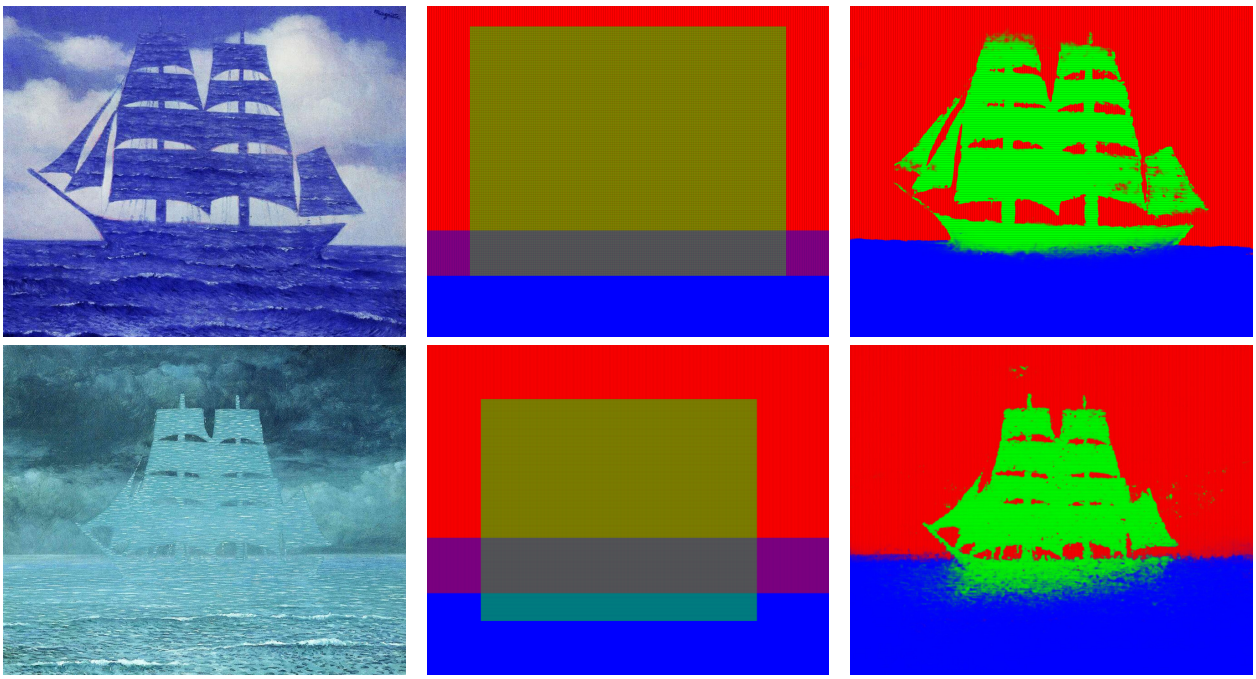


Figure F.4: Two additional versions of *Le séducteur* (left), hand-made priors (middle) and inferred segmentations (right).

(a) (b) (c)

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

References

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

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.

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

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.

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.

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.

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.

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.

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.

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.

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.

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.