Joint Triangulation and Mapping via Differentiable Sensor Fusion

Jonathan P. Chen* Uber AI Labs
Fritz Obermeyer* Uber AI Labs
Vladimir Lyapunov Uber ATG
Lionel Gueguen Uber ATG

Noah Goodman
Uber AI Labs
Stanford University
{jpchen, fritzo, vl, lgueguen}@uber.com

Abstract

The challenge of sensor fusion is prevalent in route planning, robotics, and autonomous vehicles. We leverage automatic differentiation (AD) and probabilistic programming to develop an end-to-end stochastic optimization algorithm for sensor fusion and triangulation of a large number of unknown objects. Our algorithm uses a generative model to train a Expectation Maximization (EM) clustering solver. We validate our method on street sign detections extracted from noisily geo-located street level imagery without depth information by jointly estimating the number and location of objects of different types, together with parameters for sensor noise characteristics and prior distribution of objects. We find that our model is more robust to upstream misclassifications than current methods and generalizes across sign types.

1 Introduction

One of the most challenging problems in building autonomous vehicles and route planning is the construction of accurate maps. These maps are often algorithmically constructed from a combination of sources of information including satellite imagery, government data, street view imagery, and human labeling [16, 11]. These sources trade off cost and scalability with accuracy. Therefore, improving the accuracy of automatic systems that perform robust mapping in the presence of noisy detection and classification of objects would save both cost and effort.

Our objective is to extract object locations from millions of street-level images of road signs. We employ an upstream neural detector and classifier to noisily extract rays (an origin and direction) from images and GPS data. To triangulate the location of objects from noisy rays, we propose a probabilistic model that performs bundle adjustment at scale, learning measurement parameters, *e.g.*observable radius and GPS error, while simultaneously predicting the probable locations of signs through an agglomerative clustering algorithm. The clustering algorithm performs expectation maximization (EM) using message passing and Newton's method. Critically, the clustering solver is differentiable, so sign-specific parameters can be learned jointly using stochastic variational inference (SVI).

Our system is able to incorporate additional sources of prior information to improve estimation. As an example, we show that triangulation can be further improved with domain-specific priors by incorporating road networks. Since in the real world, this data is not ubiquitously available, we learn

*Equal contribution

Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

Figure 1: Sample sign detections from photos taken from cameras on vehicles.

maximum a posteriori (MAP) estimates of geographic parameters offline with limited data, then generalize to signs in different cities.

Our contributions are as follows:

- We implement a differentiable soft clustering algorithm that uses loopy belief propagation (BP) to solve a data association problem and a differentiable Newton solver to predict cluster locations.
- We propose a generative model and a variational model that are used to learn model parameters end-to-end on partially-labeled data.
- We learn parameters unique to each type of road sign, and incorporate prior information in the form of road networks, learning *e.g.*where each type of road sign is typically located w.r.t intersections.
- We develop heuristics such as sparsification, eccentricity pruning, and locally sensitive hashing to scale to data batches of over 10,000 rays and 1000 objects on each compute node.
- We evaluate our method against state-of-the-art methods used by companies to build autonomous vehicles and maps at scale.

2 Background

The input data consists of rays denoted by the origin GPS position, the direction of the detection as a unit vector, and a confidence score from the DNN classifier. The collected data is only in 5x5 block areas, which have partially labeled ground truth. The data in San Francisco, CA consists of roughly 10 non-contiguous 5x5 blocks scattered throughout the city.

There are two main components involved in clustering: cluster prediction and data assignment [19, 17]. A typical clustering algorithms predict clusters given rays, then assign rays to clusters. Some evaluation metric (*e.g.*Euclidean distance) is used to determine how well the predicted clusters explain the observed data, and then the process is repeated.

K-means clustering is a divisive algorithm used for clustering data that come from various sources [19, 3]. The algorithm partitions data by assigning nearby objects to the nearest cluster using a distance metric. There has been work in developing heuristic algorithms (*e.g.*Lloyd's Algorithm [9]) for improving clustering in certain domains. In sensor fusion problems, it suffers from overconfidence about false detections, as it fails to incorporate prior information and uncertainty about observation parameters such as observer location.

Density-based spatial clustering of applications with noise (DBSCAN)[2] is one of the most widely-cited clustering algorithms. It clusters points based on a neighborhood radius and separates non-reachable points as outliers. Unlike k-means, DBSCAN does not require the user to specify the number of clusters upfront. However it suffers from limitations in the *a priori* knowledge of the data; the size and shape of objects in the real world are not always known.

Triangulation from bundle adjustment has a long history in object tracking as well. The PHD filter [10] and labeled multi-Bernoulli filter [13] were earlier methods to represent collections of unknown number of objects with unknown positions. To track multiple objects with unlabeled detections, Williams and Lau [17] solve the data association problem by using loopy belief propagation [12] to produce an approximate soft assignment of detections to objects. Turner et al. [15] describe a similar loopy BP-based multi-object tracking system, though they do not perform sensor fusion.

3 Method

Our method is a nested optimization algorithm consisting of a clustering solver in the inner loop, and SVI training in the outer loop. For each SVI step, the clustering solver runs multiple iterations of loopy BP, takes a Newton step, and merges and prunes clusters. The global parameters used by the clustering solver will be trained by our generative model.

We employ an initialization scheme similar to probabilistic space carving, whereby we rasterize the projection of rays onto 2D space, and initialize candidate clusters along ray intersections. The number of possible assignments grows quadratically with the number of clusters, so we can reduce the computation time of the data association algorithm by being prudent with respect to intialization.

3.1 Joint probabilistic data association

The data assignment of rays to clusters is solved approximately with an EM algorithm, iteratively alternating between computing expectations of assignments and maximizing object location probabilities at each step. During the *E*-step, we use loopy belief propagation to compute the marginal association probabilities of assignments from detections (rays) to objects.

Formally, for each object $i \in \{1, ..., n\}$, let $e_i \in \{0, 1\}$ be the existence variable which is 1 if the detection exists and 0 otherwise. Similarly, let $a_{ij} \in \{0, 1\}$ be the assignment variable which is 1 if measurement i is assigned to detection j and 0 otherwise. False detections are incorporated in a_{ij} by assigning measurements to a ghost cluster if they associate with no detections. We can think about this setup as a bipartite graph with detections and clusters as nodes and assignments as edges. Then the joint probability of the existence and assignment logits can be factored as follows:

$$P(e, a) \propto \gamma(e, a) \prod_i \psi_i^{e_i} \prod_{ij} \psi_{ij}^{a_{ij}} \tag{1}$$

where

$$\psi_i = \frac{p_e(x_i)}{1 - p_e(x_i)} \tag{2}$$

$$\psi_{ij} = \delta_{i,j} \frac{f(z_j \mid x_i)}{f_{\mathrm{FD}}(z_j)} \tag{3}$$

$$\gamma(e, a) = \begin{cases} 1, & \text{if } a_{i,j} \leq e_i \forall i, j. \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

and $\delta_{i,j}$ is the detection logit, x_i is the inferred location of the cluster in 2D coordinate space, $f(z_j \mid x_i)$ is the likelihood of ray j for detecting a cluster at x_i, $p_e(x_i)$ is the log probability of existence, and f_{FD} is the likelihood of false detections. $\gamma(e, a)$ is a Kronecker delta for agreement, only introducing terms in the joint probability if assignments agree with detections. The derivations for this formulation can be found in Williams and Lau [17].

Computing the pairwise marginals for each edge in the graph is exponentially expensive but can be approximated by loopy BP. Loopy BP generalizes the message passing in belief propagation to graphs with cycles. We define μ_{ij} to be the message passed from $e_i \rightarrow a_{ij}$ and ν_{ij} to be the message

passed in the reverse direction: $a_{ij} \to e_i$. The messages being passed are as follows:

$$\mu_{ij} = \frac{\psi_i \prod_{k \neq j} \nu_{kj}}{1 + \psi_i \prod_{k \neq j} \nu_{kj}} \tag{5}$$

$$\nu_{ij} = 1 + \frac{\psi_{ij}}{1 + \sum_{l \neq i} \psi_{lj} \mu_{lj}} \tag{6}$$

$$\overline{e_i} = \frac{\psi_i \prod_j \nu_{ij}}{1 + \psi_i \prod_j \nu_{ij}} \tag{7}$$

$$\overline{a_{ij}} = \frac{\psi_{ij} \mu_{ij}}{1 + \sum_k \psi_{kj} \mu_{kj}} \tag{8}$$

This algorithm converges quickly in practice; in our experiments, we run loopy BP for only 5 iterations. We then perform the *M*-step with a regularized Newton solver and update cluster locations and merge nearby clusters within a certain radius. Instead of a least squares solver, we use a twice differentiable log likelihood and directly apply a regularized Newton step as in equation (9). This also allows us to add in other log-likelihood terms such as a geographic prior later on.

$$\mathbf{x}_{n+1} = \mathbf{x}_n - [\mathbf{H} + \lambda \mathbf{I}]^{-1} \nabla f(\mathbf{x}_n) \quad \forall\, n \geq 0. \tag{9}$$

$\mathbf{H}$ is the Hessian and λ is the regularization factor computed from the trust region radius. The matrix solve is inexpensive because $\mathbf{H}$ has block diagonal structure with blocks of size only 2 or 3 (for 2D or 3D mapping, respectively); hence we can run this jointly over a tensor of thousands of clusters. The optimum found by the Newton solver is differentiable with respect to the solver inputs [4] so we can backpropagate through the solution to later learn global parameters.

Algorithm 1 EM Clustering

input R
$x \leftarrow$ initialize(R)
while not converged **do**:
    $p_a, p_{ae} =$ LoopyBP$(\log p_d(x, R))$ ▷ E-step
    $loss = \sum_{i,j} p_{ae}(i,j) \log p_d(x_i, R_j)$
    $g \leftarrow \nabla_\theta(loss)$
    $H \leftarrow \nabla_\theta(g_1 \dots g_n)$
    $H_{\text{reg}} \leftarrow H + g/r - \lambda_{\min}$
    $x_{\text{next}} \leftarrow x - H_{\text{reg}}^{-1} g$ ▷ M-step (Newton step)
    $x \leftarrow$ merge(x_{next})
return x

Algorithm 2 LoopyBP

input p_a
$\mu_f, \mu_b = 0$
while not converged **do**:
    $\mu_f \leftarrow \log(1 - \exp(\mu_f - \sum(\mu_b) - p_a))$
    $p_{ae} \leftarrow \exp(\mu_f + p_a)$
    $\mu_b \leftarrow \log(1 + \exp(p_a - \log(1 + \exp(\sum(p_{ae}) - p_{ae}))))$
return p_a, p_e

For our experiments, we take one Newton step after loopy BP has converged and we run Algorithm 1 for 10 iterations. Our implementations are open source [2].

3.2 Variational Inference

To learn the parameters of the clustering solver, we posit a generative model and train them jointly using SVI. Variational inference [1] is an approximate inference technique that treats the problem of

[2] Link redacted for double blind review

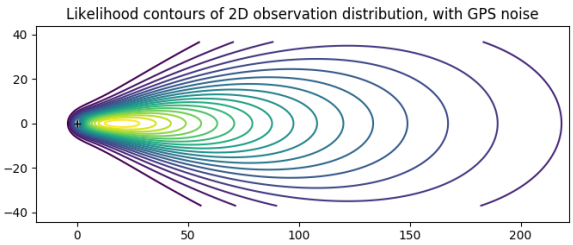

Figure 2: Observable distribution with GPS error and a radius of 50 meters. The most likely location of a sign observed by an observer standing at (0, 0) facing right is at (20, 0). Note that the incorporation of GPS noise allows (with low probability) for the sign to be behind the observer's location.

probabilistic inference $p(z|x) = p(x, z)/p(x)$ as an optimization problem by fitting an approximate distribution $q(z|x; \theta)$ to the model $p(x, z)$ by maximizing the evidence lower bound (ELBO)

$$\textbf{ELBO} = \arg\max_{\theta} \mathbb{E}_{\boldsymbol{q}(z)} \left[\boldsymbol{p}(x, z) - \boldsymbol{q}(z|x; \theta) \right] \tag{10}$$

When variational parameters are shared across data, variational inference is amenable to stochastic optimization via minibatching (stochastic gradient variational Bayes [8]) and random sampling of latent variables (stochastic variational inference [6]).

In our case, the EM clustering solver is our variational approxmation $q(z|x)$. We will build the generative model $p(x, z)$ as follows: the generative process of the clusters is approximated with a multi-Bernoulli process [13]. We experiment with two prior densities of object locations: uniform (non-informative) density over the geographic region, and a Spike-and-slab distribution over the roads discussed in section 4.3. Our observation distribution incorporates two components: a radial component to model obstruction and invisibility of distant objects (modeled as an Exponential distribution), and an angular component to model a combination of orientation error of the sensor platform and segmentation error in the deep object detector. We account for GPS error by approximately convolving our radial-angular likelihood by a Gaussian. To make this easier to compute, we preserve the Exponential radial component and model the angular component as a radius-dependent Von Mises distribution. The resulting 3-parameter distribution is shown in Figure 2. This models the generative process of the clusters, which is used to learn the parameters of our clustering solver.

Armed with our generative model, we can now write the ELBO as:

$$\textbf{ELBO} = \mathbb{E}_{r,\phi\sim q}[\log \boldsymbol{p}_\theta(e, a, x) - \log \boldsymbol{q}_\theta(e, a)] \tag{11}$$

where $\boldsymbol{q}$ is the distribution produced by the clustering solver parameterized by θ, and e and a are existence and assignment variables respectively for clusters at location x. Note that because of the Newton solver's quadratic convergence rate, we can run without gradients in all but the final iteration, and propagate gradients back through only the final output of the loopy BP. This property is especially helpful because extra clusters at early iterations are often pruned or merged by the final iteration.

The assignment solver we use to produce the variational distribution generates soft assignments of the rays to clusters. These uncertainty estimates are useful when making predictions, especially in the context of building maps for autonomous vehicles. During training we make a mean-field approximation [18] that each object's assignment to rays is independent of other object's assignments, so that the assignment distribution factors into independent Categorical distributions. This approximation allows us to exactly marginalize out assignments, leading to lower-variance gradient estimates than Monte Carlo sampling.

Since our data is largely unlabeled, we train in a semisupervised manner. Specifically, in areas with labeled ground truth, we probabilistically assign detections to known objects, assuming all objects are accounted for in ground truth. In areas without ground truth, we predict candidate clusters via Algorithm 1. Note that even though the ground truth is known in certain areas, it only provides the candidate clusters; the data association problem is still unsupervised.

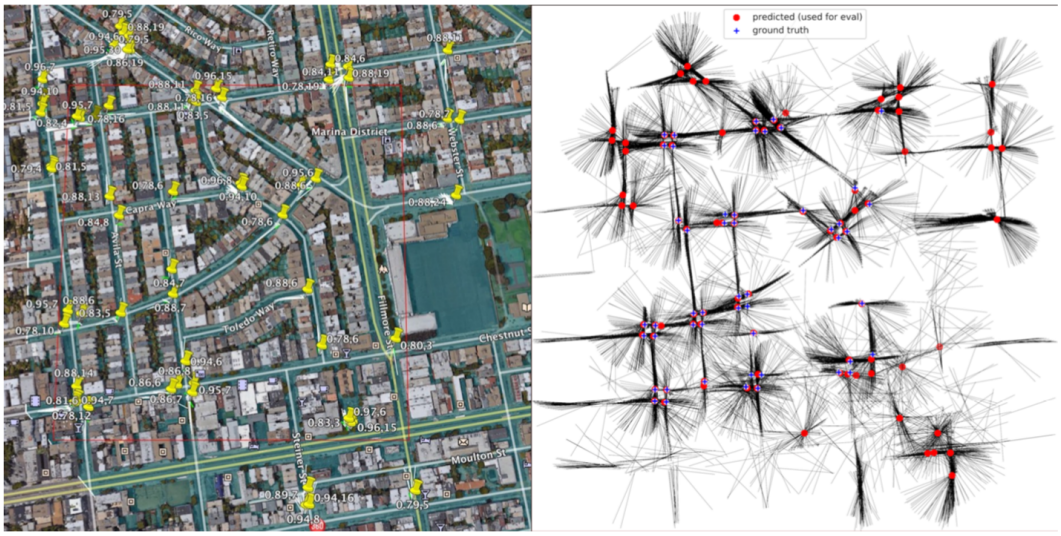

Figure 3: Predicted Stop sign locations (in yellow on the left and red on the right) for a 5x5 block region in San Francisco, CA. Note in the figure on the right that the ground truth labels (blue crosses) for Stop signs are incomplete in the region. The assignment solver has no knowledge of the ground truth clusters.

3.3 Heuristics

To eliminate false detections that arise when two rays "look past each other" (*i.e.*lie on the same line in opposite directions), we compute the eccentricity of the detection from the eigenvalues of its assigned rays. We then prune clusters based on an eccentricity threshold. The intuition is that a true detection (especially one that is not eclipsed by buildings) will be observable from multiple angles. Naturally, this varies per sign type since the visibility of signs are subject to their location and surroundings. We perform a similar thresholding with the covariance matrices of the clusters as well.

Because the ray-cluster assignment problem is very sparse, we implement sparse tensor versions of the EM algorithm, simultaneously assigning tens of thousands of rays to thousands of clusters in parallel. This is critical for scaling up training on GPUs. We also implement locally sensitive hashing to efficiently merge nearby clusters greedily.

4 Experiments

We compare two instantiations of the EM algorithm – one which is initialized by hand and one that is trained via SVI – to k-means and DBSCAN algorithms. The k-means implementation is a recursive algorithm with $k = 2$. First, a least squares solver determines candidate clusters from rays. Clusters outside a neighborhood threshold are divided into two clusters. Rays are then reassigned to one of the two clusters, and the process is repeated until convergence. DBSCAN requires a preprocessing step of computing the length of the ray (*i.e.*distance to the object) using the height of the object in image space and the focal length of the camera. The resulting "candidate locations" are then fed into the DBSCAN algorithm to perform density-based clustering. To understand the precision-recall tradeoffs, we compare the F1-scores and the Area under the Curve (AUC). A true positive is any cluster within a 10 meter radius of a ground truth object, and any additional clusters inside or outside the radius are considered false positives.

4.1 SVI training

We train 6 parameters for each sign type: the confidence weight, confidence bias, max confidence, angle error, gps error, and observable radius. As an ablation study, we train an additional 2 road network parameters discussed in section 4.3 Since sign-specific parameters are independent, the

		Stop	DoNotEnter	NoLeft	LeftYield	NavPullThru	NoRightCond	Highway
	Truth	228	54	50	2	4	4	5
	Detections	13087	6779	822	620	4836	741	926
k-means	F1	0.76	0.61	0.56	0.5	0.020	–	0.43
	AUC	0.74	0.55	0.44	0.57	0.34	0.0	0.43
DBSCAN	F1	0.63	**0.68**	0.55	–	0.020	0.083	0.4
	AUC	0.78	0.63	0.46	0.0	0.39	0.01	0.33
EM (ours)	F1	0.83	0.65	**0.60**	0.40	**0.039**	0.22	**0.89**
	AUC	0.83	**0.64**	**0.48**	0.81	0.06	0.03	**0.80**
SVI-EM (ours)	F1	**0.84**	0.65	**0.60**	**0.67**	0.020	**0.50**	0.66
	AUC	**0.86**	**0.64**	**0.48**	**0.88**	**0.57**	**0.31**	0.57

Table 1: Results on road signs for the city region in San Francisco comparing the F1 scores and Area under the curve. “EM" is the EM algorithm with hand initialized parameters. “SVI-EM" is the EM algorithm with parameters tuned by SVI. Note that the ground truth is incomplete so F1 scores should be higher for all four methods.

training scheme is completely parallelizable. We train only in one region that has partial ground truth. The incompleteness of the ground truth poses to be a problem as SVI will (incorrectly) attempt to explain away rays within a region that are not associated with a ground truth cluster. As such, in areas where the true clusters are sparse *e.g.*Highway shields, we notice that SVI training does not make improvements over the manual initialization scheme. The eccentricity and covariance pruning thresholds are used during training but are removed at test time. This again is beneficial for signs that lack enough detections for eccentricity and covariance to be useful.

Discrete latent variables such as the Categorical variables in the assignment distribution produce high variance gradient estimates [14, 5]. To obtain lower-variance gradient estimates, we enumerate out discrete variables, performing exact inference for discrete latents in both our model and the variational approximation. This eliminates gradient estimator variance due to sampling latent variables, so that the only remaining source of variance during training is the random subsampling of data minibatches.

We train with the Adam optimizer [7] using a learning rate of 0.001 and anneal the learning rate with a decay factor of 0.7 every 50 epochs. We partition data into minibatches of approximately 5x5-block regions that contain anywhere from 40 to 7000 rays each depending on sign type. We run 5 iterations of loopy belief propagation and 10 EM iterations per SVI step.

4.2 Results

The results of various clustering algorithms run on signs in San Francisco are shown in Table 1. The AUC is our metric of interest, as it encompasses the precision-recall tradeoff that the algorithms make. Our algorithm outperforms the two baselines often used by autonomous vehicle mapping companies. Note that it tends to perform better when signs are sparse in a given region. In these scenarios, our model is able to better deal with false detections than the two baselines.

When trained with SVI, the algorithm tends to perform as well as or better than the hand initialized version; it has the highest AUC for all but one sign type. SVI training was only using data in one 5x5 block, and we expect the results to improve further if we were to train on the entire city. Crucially, the performance of SVI relies on the existence and accuracy of ground truth labels. When labels are noisy and sparse (*e.g.*for Highway signs), we see that SVI performs worse than hand-initialization.

SVI seems to successfully learn global parameter values even in the presence of unknown data associations and limited ground truth data. More importantly, it allows a single hierarchical generative model to generalize to different sign types by learning parameters for each sign type. As long as we are reasonably confident that our generative model is faithful to real world observations, we can reuse the same model to train a variety of clustering solvers for different traffic objects.

4.3 Geographic parameter training

One of the advantages of our Bayesian model is its ability to incorporate prior information as modular components. We demonstrate how to incorporate additional prior information in a city where we have access to the road network to improve triangulation results. These parameters can theoretically

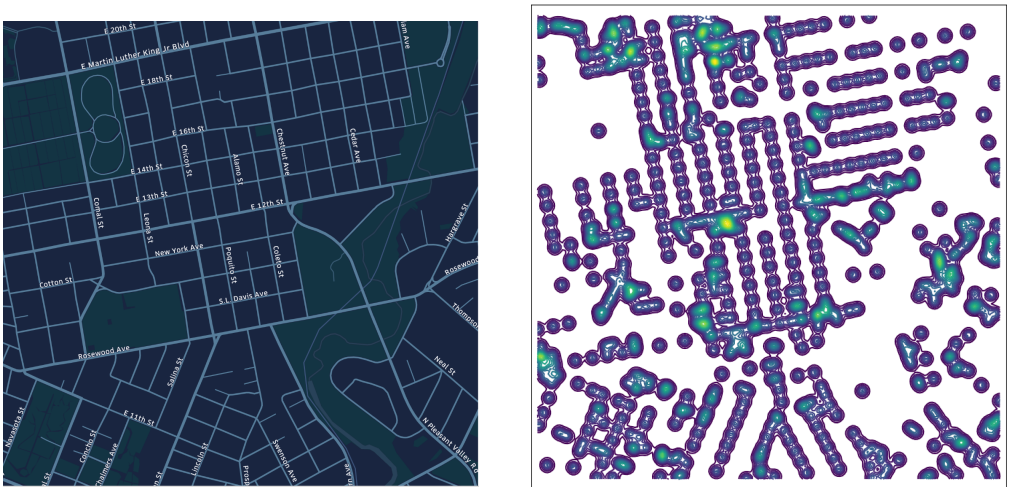

Figure 4: Region of roads in Austin, TX (left) and a Spike-and-slab prior distribution over the intersections with an affinity of 1 (right).

		Stop	Crosswalk	Yield	StateRoute	TempParking	Merge	DoNotEnter
No road network	Prec	0.40	0.60	0.55	0.12	0.14	0.43	0.63
	Recall	**0.88**	**0.52**	**0.79**	**1.0**	**0.53**	**1.0**	**0.85**
Road network	Prec	**0.42**	**0.63**	**0.61**	**0.14**	**0.17**	**0.50**	**0.71**
	Recall	**0.88**	**0.52**	**0.79**	**1.0**	**0.53**	**1.0**	**0.85**

Table 2: Results on road signs with and without a prior distribution over the road network in various regions in Austin, TX. Regions are all approximately 4x4 blocks and were selected based on regions where there was the highest density of ground truth. The incorporation of an informative prior improves precision, even in small regions.

also be transferred across cities since traffic laws and signs are nearly universal across states, and even most countries (*e.g.*Stop signs are typically located at intersections and require a complete stop).

We use a Spike-and-slab prior over the entire area which places a Gaussian distribution at intersections and Uniform distribution everywhere else as in Figure 4.3. We train on a road network in Austin, TX, and compare clustering prediction results between those with a prior over the road network and those without (*i.e.*a Uniform prior across the entire region). The density function is given by:

$$p(x) = \frac{1-\alpha}{b-a} + \frac{\alpha}{K}\sum_{i=1}^{K}\mathcal{N}(x;\mu_i,\sigma) \tag{12}$$

where α is a Categorical logit that represents the affinity of clusters to intersections, b and a are the upper and lower bounds of the area of interest respectively, and K are the number of components. We train a MAP estimate of the intersection affinity for each sign type. This training can be done completely offline (*i.e.*outside of the SVI training loop), as it fits a spatial distribution directly to the road network.

We test in different regions per sign type since the ground truth is sparse across the city, with each region roughly consisting of 12-25 contiguous blocks in a rectangle. We manually select regions with higher concentrations of ground truth and run both models identically with the exception of the prior. As displayed Table 2, the streetmap prior seems to not affect the recall, but helps precision, giving strictly better results than the version without. This may be due to the fact that we use a sparse prior which is most effective in culling out outlier clusters. The prior provides weak information for a true cluster which has many detections associated with it. A false detection however, often comes from a few sporadic rays that don't necessarily converge at a single point. In the presence of the road network prior, it may be dropped, depending on the type of sign.

References

[1] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[2] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise.

[3] Mehmet Gönen and Adam A Margolin. Localized data fusion for kernel k-means clustering with application to cancer biology. In *Advances in Neural Information Processing Systems*, pages 1305–1313, 2014.

[4] Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

[5] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

[6] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

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

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[9] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[10] Ronald Mahler. Phd filters of higher order in target number. *IEEE Transactions on Aerospace and Electronic systems*, 43(4), 2007.

[11] Gellért Máttyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3611–3619, 2016.

[12] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[13] Stephan Reuter, Ba-Tuong Vo, Ba-Ngu Vo, and Klaus Dietmayer. The labeled multi-bernoulli filter. *IEEE Trans. Signal Processing*, 62(12):3246–3260, 2014.

[14] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2627–2636, 2017.

[15] Ryan D Turner, Steven Bottone, and Bhargav Avasarala. A complete variational tracker. In *Advances in Neural Information Processing Systems*, pages 496–504, 2014.

[16] Jan D Wegner, Steven Branson, David Hall, Konrad Schindler, and Pietro Perona. Cataloging public objects using aerial and street-level images-urban trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6014–6023, 2016.

[17] Jason Williams and Roslyn Lau. Approximate evaluation of marginal association probabilities with belief propagation. *IEEE Transactions on Aerospace and Electronic Systems*, 50(4):2942–2959, 2014.

[18] Eric P Xing, Michael I Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 583–591. Morgan Kaufmann Publishers Inc., 2002.

[19] Shi Yu, Leon Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039, 2012.