Neural Point Processes for Pixel-wise Regression

Chengzhi Shi, Gözde Özcan, Miquel Sirera Perelló, Yuanyuan Li, Nina Iftikhar Shamsi, Stratis Ioannidis

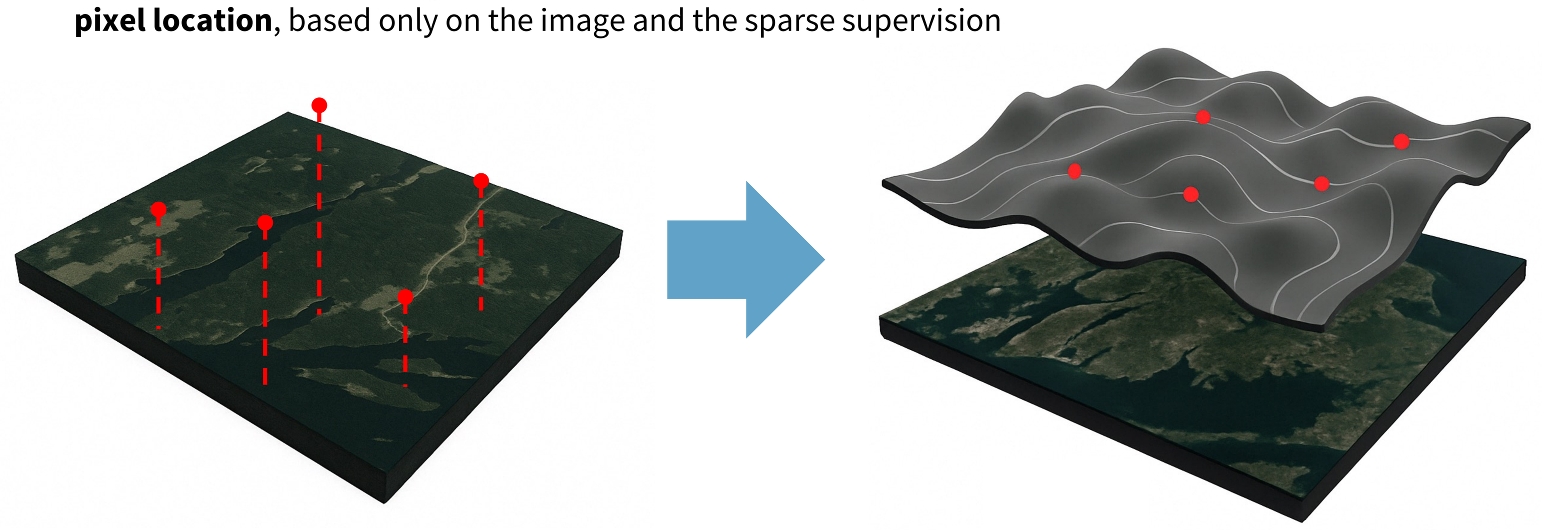
Northeastern University, Boston, MA, USA

Motivation

➔ Many real-world tasks only provide **labels at a small subset of pixels**. Some examples are:

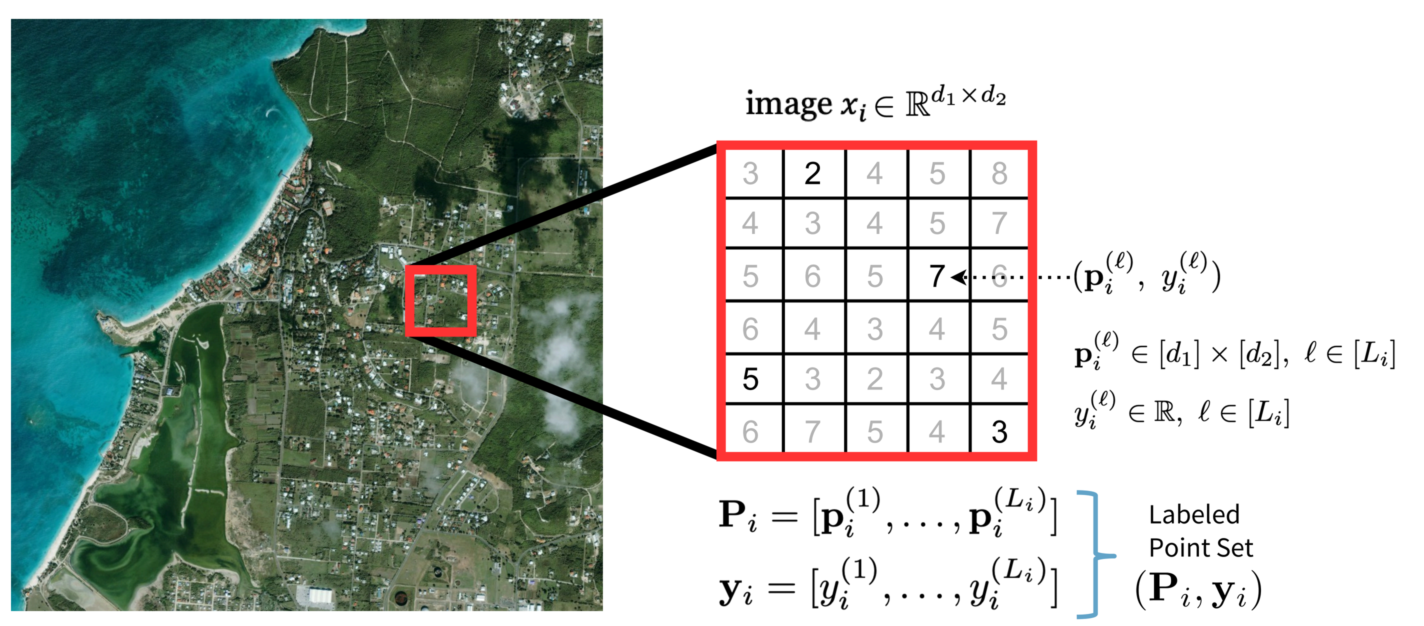
Challenge: how to **generalize from sparse supervision**?

➔ The goal is **pixel-wise regression**: to learn a model that predicts the continuous label value at **every pixel location**, based only on the image and the sparse supervision



Problem Formulation

➔ Let the **dataset** $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{P}_i, \mathbf{y}_i)\}_{i=1}^n$ be defined such that each sample consists of an image $\mathbf{x}_i \in \mathbb{R}^{d_1 \times d_2}$, where $i \in [n]$, and an associated set of labeled points $(\mathbf{P}_i, \mathbf{y}_i)$, detailed below:



➔ We fit a neural network (NN) $f(\mathbf{x};\theta) \in \mathbb{R}^{d_1 \times d_2}$, parametrized by $\theta \in \mathbb{R}^{m_\theta}$, that takes an input image and predicts values at all *possible pixel locations* $\mathbf{p} \in [d_1] \times [d_2]$

➔ Base approach: minimize the Euclidean error only at labeled points

$$\mathcal{L}_{\texttt{MSE}}(\theta, \mathcal{D}) = \sum_{i=1}^{n} \sum_{\ell=1}^{L_i} \left(f_{p_i^{(\ell)}}(\mathbf{x}_i;\boldsymbol{\theta}) - y_i^{(\ell)} \right)^2 = \sum_{i=1}^{n} \| f_{\mathbf{P}_i}(\mathbf{x}_i;\boldsymbol{\theta}) - \mathbf{y}_i \|_2^2$$

Problem: Ignores spatial relationships between nearby pixels

Acknowledgement

Research was sponsored by the United States Army Core of Engineers (USACE) Engineer Research and Development Center (ERDC) Geospatial Research Laboratory (GRL) and was accomplished under Cooperative Agreement Federal Award Identification Number (FAIN) W9132V-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of USACE EDRC GRL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Our Method: Neural Point Processes (NPPs)

Bringing structure to sparse pixel-wise regression

➔ NPP approach: we introduce spatial correlations by modeling labels are modeled as a Gaussian Process (GP) over the DNN output

- We assume that for every point $\mathbf{p}_i$, the corresponding label y_i is given by:

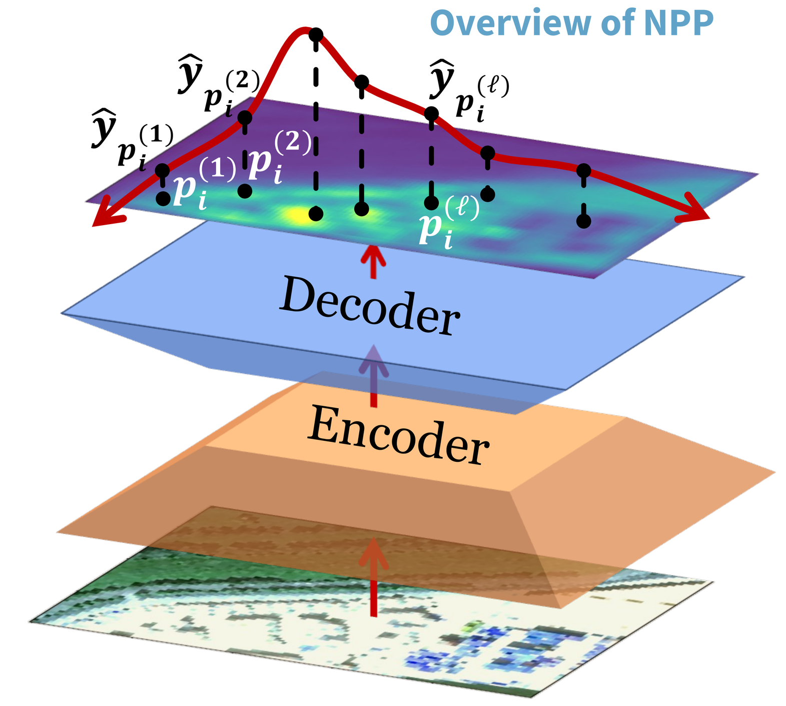
$$y_i = g_i(\mathbf{p}_i) + \varepsilon$$

where $\varepsilon \sim N(0, \sigma_0^2)$ is i.i.d noise, and $g_i(\cdot)$ is a GP:

$$g_i(\cdot) \sim \mathcal{GP}(m_i(\cdot), k_i(\cdot,\cdot))$$

with mean function $m_i(\mathbf{p}) = f_{\mathbf{p}}(\mathbf{x}_i;\theta) \in \mathbb{R}$ (output of the NN)
and kernel function $k_i(\mathbf{p},\mathbf{p}') = k(\mathbf{p},\mathbf{p}';\zeta_i)$ (parametric PSD kernel)

Overview of NPP



The NPP method:

- Encourage Smoothness
- Accounts for proximity

➔ With this setup, we can obtain our MLE of θ by minimizing:

$$\mathcal{L}_{\texttt{NPP}}(\boldsymbol{\theta},\zeta;\mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}\Big[\underbrace{\log\left|k(\mathbf{P}_i,\mathbf{P}_i;\zeta)+2\sigma_0^2\mathbf{I}\right|}_{\text{Kernel Regularization Term}} + \underbrace{(\mathbf{y}_i - f_{\mathbf{P}_i}(\mathbf{x}_i;\theta))^\top \big(\overbrace{k(\mathbf{P}_i,\mathbf{P}_i;\zeta)}^{\text{PSD Covariance matrix of the GP}} + \overbrace{\sigma_0^2\mathbf{I}}^{\text{Additional noise term}}\big)^{-1}(\mathbf{y}_i - f_{\mathbf{P}_i}(\mathbf{x}_i;\theta))}_{\text{Squared Mahalanobis distance}}\Big]$$

Squared Mahalanobis distance ⟶ enforces correlations between the values of points that are proximal

Key insight: since **NPPs** model outputs as a **Gaussian Process**, we enable **test-time updates**

➔ When partial labels $(\mathbf{P}^\dagger, \mathbf{Y}^\dagger)$ are available, we can compute the **posterior distribution** over predictions for the rest of the points we want to predict, namely $(\mathbf{P}^*, \mathbf{Y}^*)$

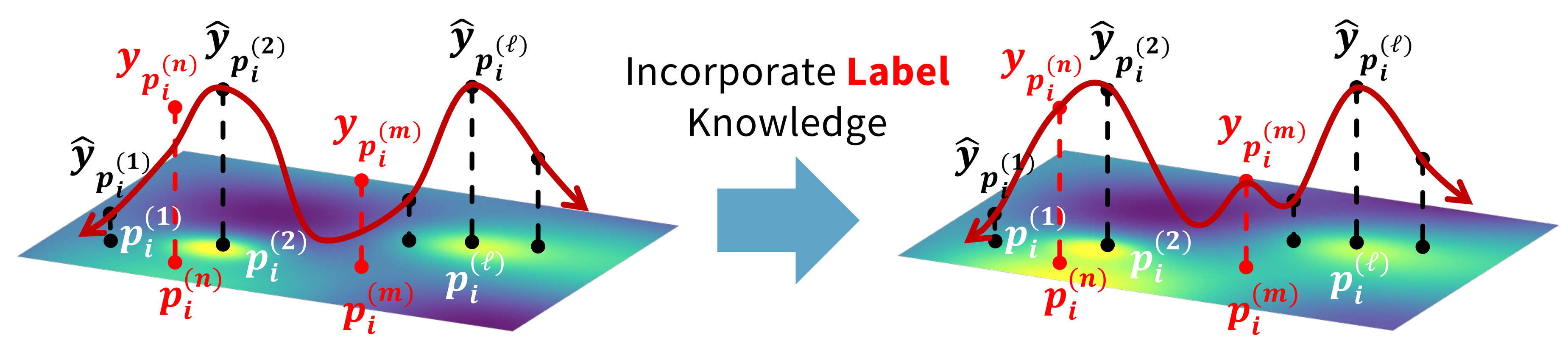
- Conditioned on observations $\mathbf{Y}^\dagger$, the posterior distribution of $\mathbf{Y}^*$ is Gaussian with the following mean and covariance:

$$\mathbb{E}[\mathbf{Y}^*|\mathbf{Y}^\dagger] = m(\mathbf{P}^*) + k(\mathbf{P}^*,\mathbf{P}^\dagger)\mathbf{K}_{\dagger,\dagger}^{-1}(\mathbf{y}^\dagger - m(\mathbf{P}^\dagger))$$
$$\mathrm{cov}(\mathbf{Y}^*) = k(\mathbf{P}^*,\mathbf{P}^*) - k(\mathbf{P}^*,\mathbf{P}^\dagger)\mathbf{K}_{\dagger,\dagger}^{-1}k(\mathbf{P}^\dagger,\mathbf{P}^*)$$

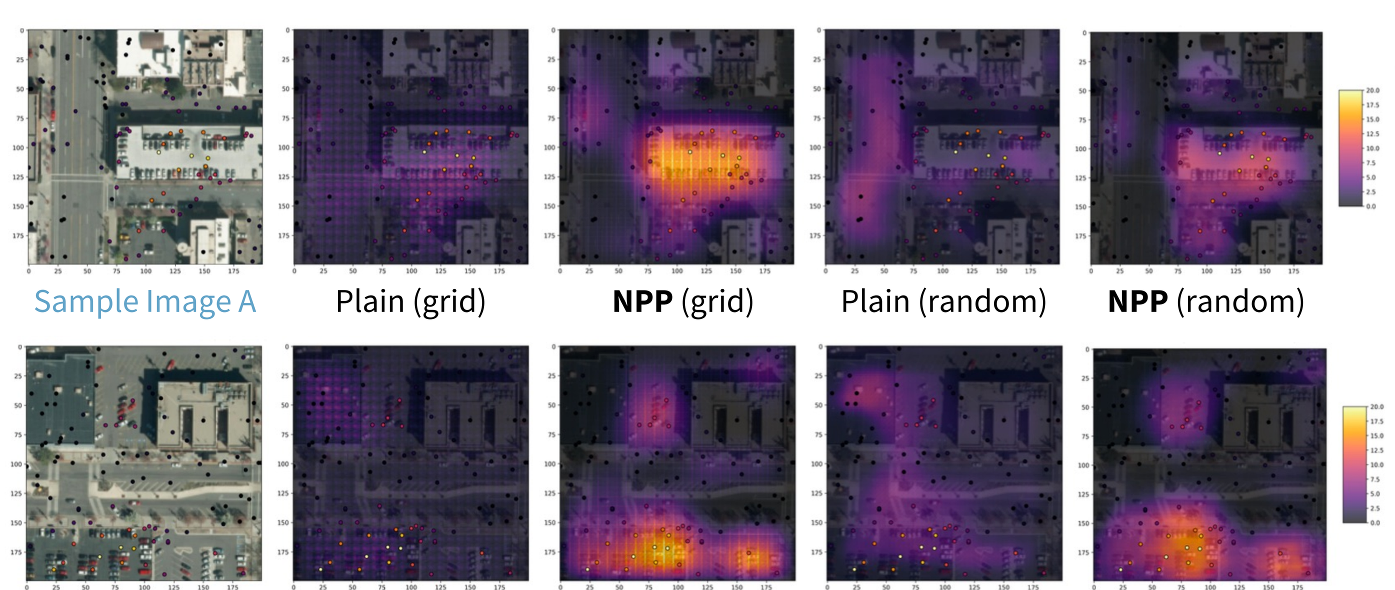
where $\mathbf{K}_{\dagger,\dagger} \equiv k(\mathbf{P}^\dagger,\mathbf{P}^\dagger) + \sigma_0^2\mathbf{I}$

A quick note on complexity
- Our method's complexity will scale with the number of labeled points, not with the image size! $(L \ll d_1 \times d_2)$

- This provides **refined estimates** and quantifies **uncertainty** — a full **Bayesian posterior**
- Check our Partial Label Revelation Experiments in the results



A Visual Comparison



➔ We compare predictions from different models on two sample test images from the COWC dataset (car counts per pixel)

- Heatmaps from **NPP** better align with car-dense areas like parking lots and building edges

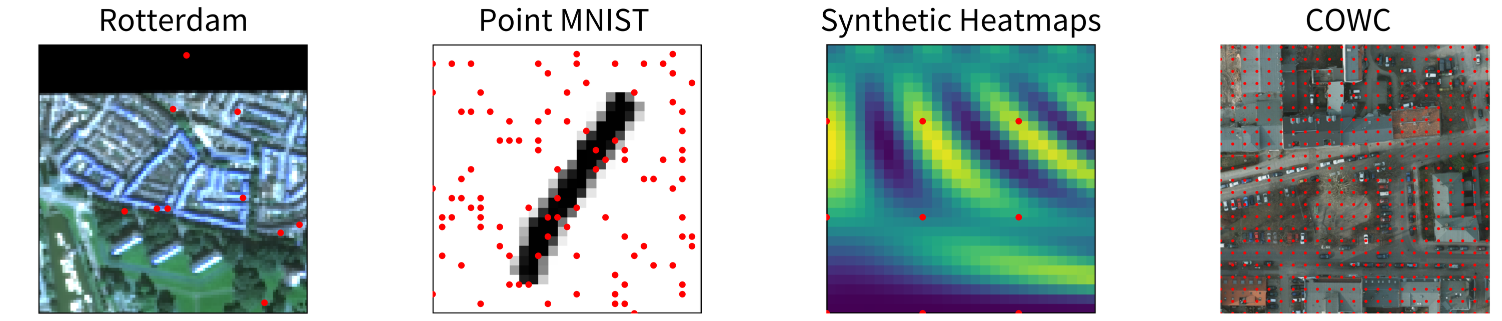
Experiments

Baselines Compared
- **Plain**: Standard MSE-trained network
- **NP** (Neural Processes) [Garnelo et al., 2018]
- **ConvNP** (Convolutional Neural Processes) [Gordon et al., 2020]
- **NPP** (Ours): Gaussian Process regularized training
- **NPP-GP** (Ours): NPP + posterior update with partial labels

Architectures (Plain, NPP, NPP-GP methods)
- **AE** (Autoencoder)
- **DDPM + AE** (Denoising Diffusion Probabilistic Model + AE)

Kernel Strategies (NPP, NPP-GP methods)
- **Static**: Fixed kernel (e.g., RBF), tuned via validation
- **Learnable**: Kernel params optimized during training
- **Context-aware**: parameters regressed from the input

Dataset	Point Distribution	# Samples	Width x Height	# Labels Sparse	# Labels Dense
Synthetic Heatmaps	grid	1000	28 x 28	9	100
	random	1000	28 x 28	10	100
Point MNIST	grid	1000	28 x 28	9	100
	random	1000	28 x 28	10	100
Rotterdam	grid	1000	100 x 100	16	121
	random	1000	100 x 100	10	100
COWC	grid	1000	200 x 200	81	529
	random	1000	200 x 200	100	500



Results

Metrics
- **MSE** ↓: Measures the average squared difference between predicted and true value
- **R²** ↑: Measures how well predictions explain the variance in the true data

➔ NPPs consistently outperform standard MSE baselines, NP, and ConvNP, especially in sparse label settings and when partial labels are available at inference time — a setup we refer to as *partial label revelation*, where some ground truth labels are revealed during inference to improve prediction accuracy across methods

	Datasets		Rotterdam				COWC				Partial label revelation
	Point pattern		Grid		Random		Grid		Random		
	Metric		MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	
Real-world	Sparse	Plain	1.79	−0.058	1.14	0.297	17.99	0.060	17.7	0.078	✗
		NPP (ours)	**1.76**	**−0.046**	**0.834**	**0.437**	**4.89**	**0.767**	**7.77**	**0.594**	
		NPP-GP (ours)	1.76	−0.046	**0.833**	**0.437**	**4.86**	**0.769**	7.65	0.600	✓
		NP	**1.44**	**0.047**	1.64	−0.027	14.0	0.263	15.9	0.171	
		ConvNP	0.725	−4.44	1.12	−8.10	16.6	−0.529	**5.68**	**0.768**	
	Dense	Plain	1.55	0.100	0.486	0.678	10.9	0.432	14.51	0.208	✗
		NPP (ours)	**1.25**	**0.286**	**0.453**	**0.700**	**5.67**	**0.704**	**5.31**	**0.710**	
		NPP-GP (ours)	1.25	**0.287**	**0.443**	**0.707**	**5.60**	**0.706**	5.02	0.726	✓
		NP	**1.20**	0.196	1.199	0.197	17.9	0.066	13.2	0.272	
		ConvNP	0.581	−7.27	0.344	0.379	19.7	−3.12	**3.47**	**0.922**	
	Datasets		PMNIST				Synthetic Heatmaps				Partial label revelation
	Point pattern		Grid		Random		Grid		Random		
	Metric		MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	
Synthetic	Sparse	Plain	**67.5**	**0.087**	0.456	0.992	1298	−15.0	14192	−173	✗
		NPP (ours)	72.0	0.026	**0.451**	**0.993**	**94.6**	**−0.164**	**27.1**	**0.666**	
		NPP-GP (ours)	**56.3**	**0.239**	**0.451**	**0.993**	**75.2**	**0.075**	27.5	0.661	✓
		NP	78.2	−0.072	44.8	0.404	111	−0.381	104	−0.587	
		ConvNP	63.2	−19.4	29.7	0.293	79.8	−5.28	**12.1**	**0.761**	
	Dense	Plain	0.467	0.994	0.181	**0.998**	104	−0.28	27.1	0.666	✗
		NPP (ours)	**0.307**	**0.996**	**0.128**	**0.998**	**94.6**	**−0.164**	**26.9**	**0.669**	
		NPP-GP (ours)	**0.300**	**0.996**	**0.120**	**0.999**	74.71	0.081	26.9	0.669	✓
		NP	64.6	0.121	27.2	0.627	59.2	0.269	43.2	0.467	
		ConvNP	12.7	0.765	10.0	0.861	**24.9**	**0.246**	**19.1**	**0.727**	

Partial Label Revelation Experiments: We evaluate model performance when a few point labels are revealed at inference time, enabling refined predictions