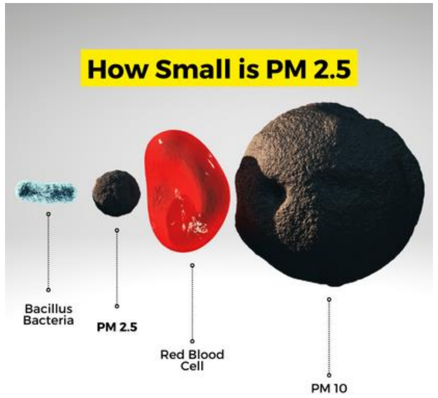
Spatially Adaptive $PM_{2.5}$ Estimation in Low-Sensor Regions using Variational Gaussian Processes

Authors: Shrey Gupta[1], Avani Wildani[2], Yang Liu[3]

[1]Boston College, [2]Cloudflare, [3]Emory University

DSAA, 2025, (11/09/2025)

Problem: Air Pollution Monitoring in Data-Scarce Regions



$PM_{2.5}$ (Particulate Matter) poses **public health risk**

- It contains particles < 2.5 microns → can **penetrate lungs and bloodstream**.
- Effects disproportionately **higher in densely populated regions**.

Need for ground sensors in high-density and rural regions.

However, this infrastructure can be unfeasible

- Dense sensor networks are **costly to install**.
- Developing regions **lack critical investment**.

Case Study: Lima, Peru

- **Second most polluted city** in the Americas.
- Only **10 ground sensors** for entire metropolitan area.
- Sensors **clustered in central Lima**, leaving vast areas unmonitored.

Lima, Peru

Key Challenges

1. Spatial Irregularities

- **Sparse sensor placement** across the region.
- **Uneven coverage** - dense in populated centers, absent elsewhere.

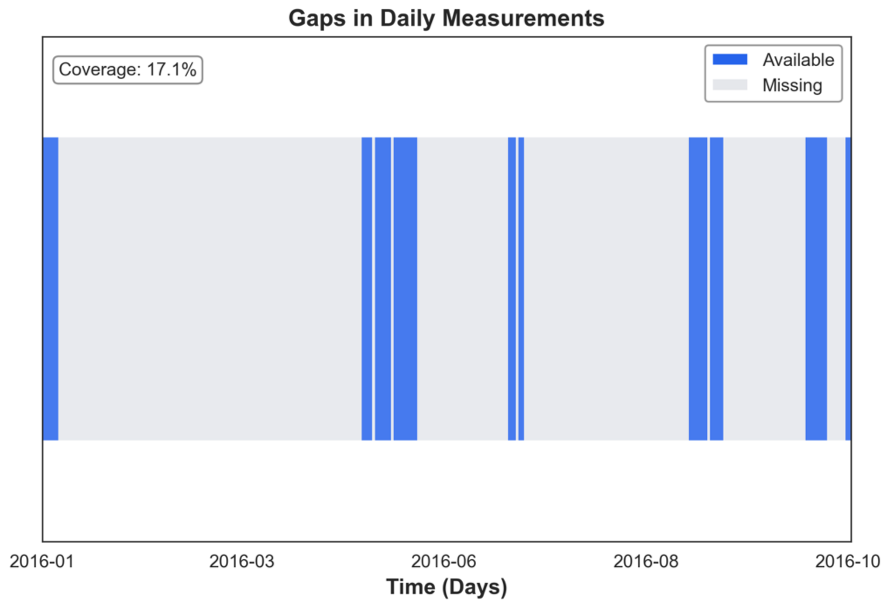
2. Temporal Irregularities

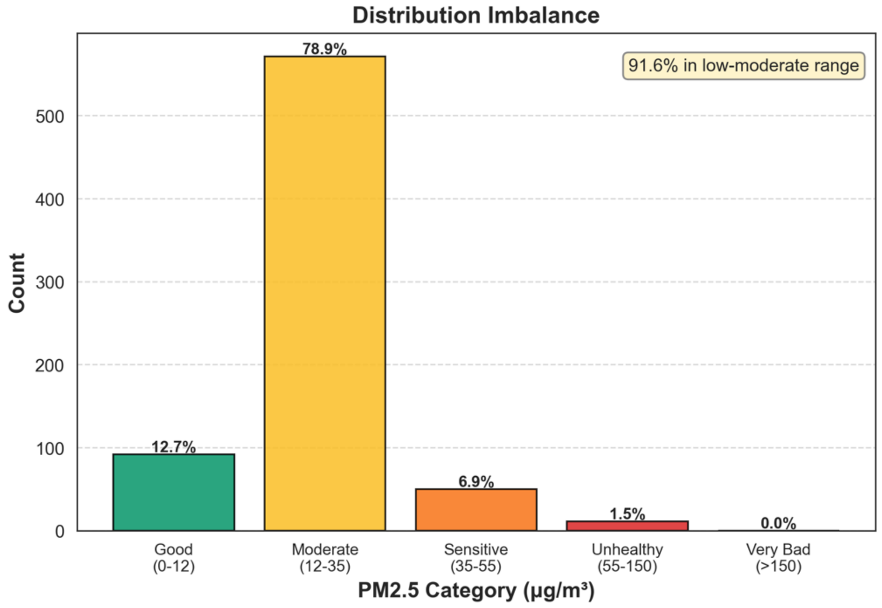
- **Temporal gaps** in the collected data (missing daily values).

3. Distribution Imbalance

- **Mostly moderate $PM_{2.5}$ levels** in collected data
- **Few high pollution** (extreme) episodes.
- Imbalance creates **non-IID** data distribution.

Traditional machine learning models struggle with such data characteristics

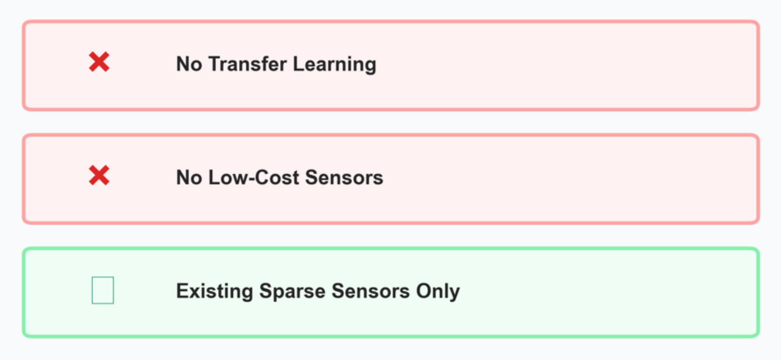




Research Question: Can We Build Self-Reliant Prediction Models?

Q1 Can we **avoid leveraging auxiliary technologies**: data or sensors?

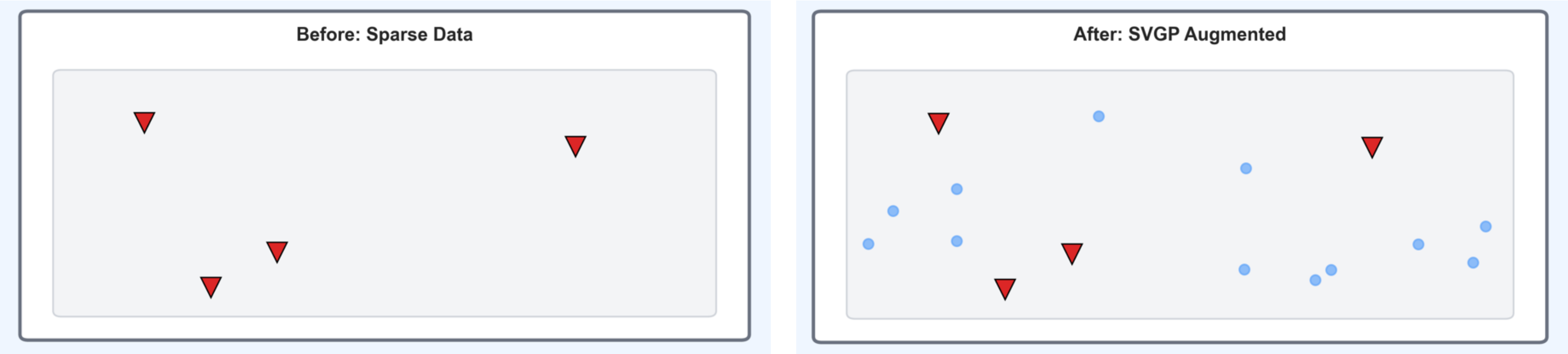
- No transfer learning from other regions (Requires: Large-scale models).
- No low-cost sensor deployments (Requires: Policy intervention).
- Use only existing sparse ground sensor data.



Q2 Does **sensor placement** affect ML model performance?

- Can model performance be improved through **strategic placement**?
- How do models **adapt to different spatial configurations** for such spatiotemporal settings?

Our Approach: Use Sparse Variational Gaussian Processes (SVGPs) to generate data points that spatially adapt to the region.



Why Sparse Variational Gaussian Processes?

Gaussian Processes (GPs):

- Non-parametric: Don't assume fixed data structure.
- Adapt to complex, non-IID data.

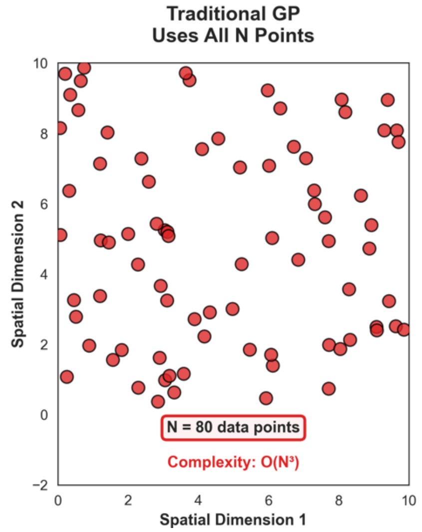
Sparse Variational GPs

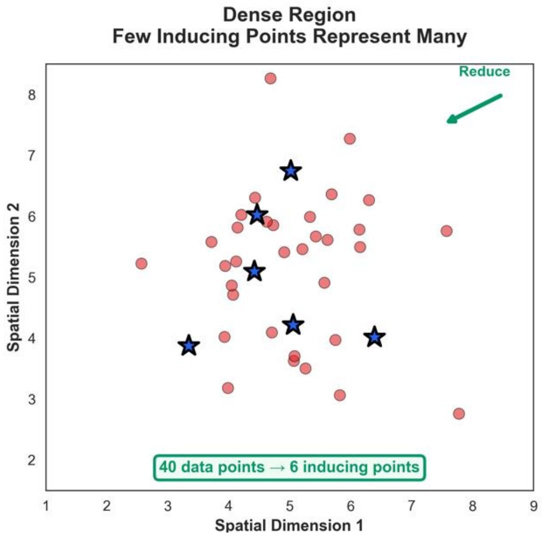
- **Sparse:** Use inducing points (M << N data points)
- Representative subset of the entire dataset.

Variational Inference: Approximate distributions via optimization

- Opimize inducing points during training.
- Adapt to underlying data structure.

Inducing points can serve as synthetic training data that generalize across sparse sensor networks





Central Hypotheses

Hypothesis 1

Spatial Adaptation: Well-initialized inducing points spread over the sparse sensing region.

What does "spatial adaptation" mean?

- Inducing points start near training sensors (K-means centroids).
- During optimization, they migrate across the region.
- Final positions capture spatial structure of $PM_{2.5}$ distribution.

Hypothesis 2

Strategic Placement Matters: Strategic placement of sensors enables improved spatial adaptation.

Why does this matter?

- Better spatial adaptation → better generalization.
- Optimizes the future sensor deployment strategies.
- Allows for limited sensing infrastructure.

SVGP Methodology

Gaussian Process

Defines a distribution over functions, specified by a mean function $m(x)$and a covariance (kernel) function $k(x, x')$.
$f(x) \sim GP\big(m(x), k(x, x')\big)$

Sparse Variational Gaussian Processes

SVGP approximates the GP using a smaller set of M << N inducing points, reducing computation.

Inducing Points

- Learnable points $Z = \{z_j\}_{j=1}^{M}$ in input space.
- Function values at these points: $u = f(Z)$.
- They summarize the dataset efficiently and allow sparse approximations.

Posterior approximation

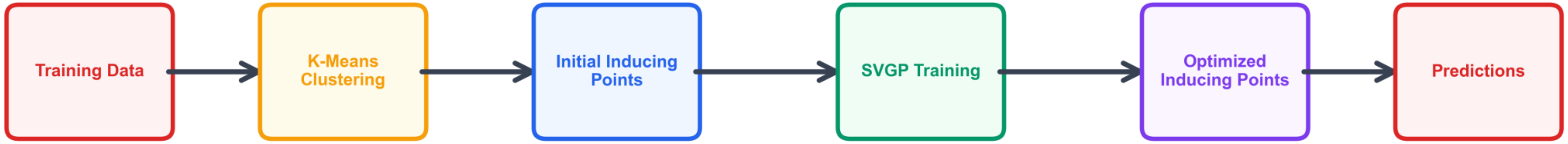
Instead of the full GP posterior $p(f \mid y)$,we use a variational distribution, $q(u)$ over the inducing points as

$q(f) = \int p(f \mid u), q(u), du$

SVGP Methodology

Optimization via ELBO

- Maximize Evidence Lower Bound (ELBO).
- ELBO $= \mathbb{E}_{q(f)}[log\, p\,(y \mid f)] - KL[q(u) \mid\mid p(u)]$
- Train for 1500 epochs; inducing points adapt during training.
- ELBO provides a tractable approximation of the full GP posterior.



SVGP Training Pipeline

Experimental Setup

Dataset:

- Daily averaged $PM_{2.5}$ values of Lima [year: 2016; Shape: (2419, 16)].
- 10 ground sensors in total.
- 16 features (meteorological, topographical, pollution, spatial, temporal).

Evaluation Strategy:

- 5 randomized train-test splits.
- 4 sensors for training, 6 sensors for testing.
- Tests model's ability to predict at unseen locations.

Baseline Models:

Gaussian Process Regressor (GPR)

- RBF + Constant + White Kernel [$k_{\text{RBF}}(x, x') = \sigma_f^2 \exp\left(-\frac{|x-x'|^2}{2\ell^2}\right) + k_{\text{Const}}(x, x') = c + k_{\text{White}}(x, x') = \sigma_n^2, \delta_{x,x'}$]
- 10 optimizer restarts

Gradient Boosting

- Learning rate = 0.05, 1000 estimators

Lasso Regression

- $\alpha = 0.5$

Metrics: RMSE (Root Mean Squared Error)

Results

Split 1: Sparse Sensor Configuration

- Training sensors (red) widely distributed.
- Initial inducing points (yellow) clustered near sensors.
- Optimized inducing points (purple) spread northward.

Split 2: Linear Sensor Configuration

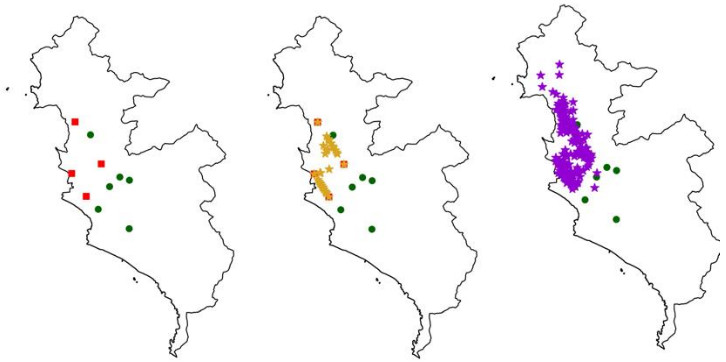
- Training sensors nearly linearly arranged.
- Limited spatial spread of inducing points.
- Growth restricted around sensor locations.

Key Observation: Sensor placement **directly affects** inducing point adaptation

- Sparse, well-distributed sensors → better spatial coverage.
- Linear/clustered sensors → limited adaptation.

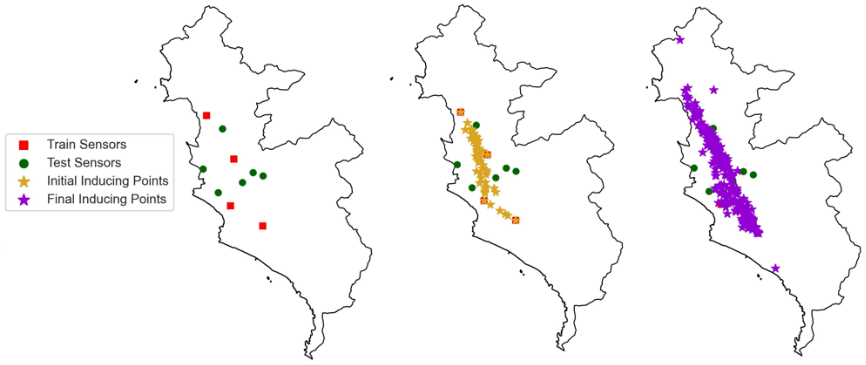
Validation: This confirms our hypothesis about strategic sensor placement.

Sparse Sensor Configuration



Split 1

Linear Sensor Configuration



Split 2

Model	**RMSE**
SVGP	**10.13**
Gaussian Process	11.24
Gradient Boosting	11.25
Lasso Regression	11.30

Future Directions

1. Ablation studies with ML models.

- Use the inducing points with alternative ML models to compare prediction accuracy.

2. Interpolation models to determine $PM_{2.5}$ values at inducing point locations.

- Interpolate PM2.5 using kriging, etc, to compare their performance to SVGPs.

3. Generative Modeling for Synthetic Data

- Use optimized inducing points with generative architectures.
- Synthesize additional training data to reduce spatial irregularities.

4. Low-Cost Sensors for Extreme Events

- Deploy targeted low-cost sensors in high-PM2.5 hotspots.
- Capture underrepresented extreme values.

5. Ground-sensor Placement

- Use inducing points to identify high-uncertainty regions.
- Active learning: where new data helps most. | Adaptive sensing: where new sensors help most.

Conclusions & Impact

Key Contributions

Spatial Adaptation Validated
- Inducing points spread across sparse sensing regions
- Capture underlying PM2.5 distribution structure

Strategic Placement Matters
- Well-distributed sensors enable better adaptation
- Informs future infrastructure deployment

Strong Performance
- 10% in RMSE
- No auxiliary data or additional sensors needed

Practical Impact:
- Scalable framework for developing regions with limited resources
- Reduces infrastructure costs while maintaining prediction accuracy
- Applicable to other environmental monitoring challenges

Thank you! Questions?

Contact: shrey.gupta@bc.edu

Web: shrey.gupta.github.io

Code Available: github.com/shrey-gupta/svgps-for-low-sensor