Wildfire Smoke $PM_{2.5}$ Modeling with Convex Optimization and Low-Cost Sensors

A Graph-Signal Approach to Data Recovery and Interpolation

Param Somane

University of California, San Diego

psomane@ucsd.edu

February 11, 2025

Abstract

This report presents a comprehensive study on reconstructing and denoising wildfire smoke $PM_{2.5}$ sensor data using convex optimization and graph-based methods. By modeling sensors as nodes in a spatially informed graph, we apply Laplacian and total variation (TV) regularization to better handle missing or noisy readings. We compare these convex techniques (solved via ADMM, gradient descent) to classical spatial interpolation methods such as Kriging and IDW, with extensive cross-validation analyses under real-world low-cost sensor deployments. Our findings suggest that the proposed approach significantly improves interpolation accuracy while enhancing robustness to sparse connectivity and sensor outages.

Contents

1 Introduction

Wildfires emit large amounts of particulate matter, particularly $PM_{2.5}$, causing severe air quality deterioration over extensive regions. $\mathbf{PM_{2.5}}$ (particulate matter smaller than 2.5

micrometers in diameter) is especially concerning because such fine particles can penetrate deeply into the respiratory tract, leading to adverse health outcomes including respiratory and cardiovascular issues. Traditional regulatory monitoring networks (e.g., the U.S. EPA Air Quality System) are often too sparse to capture sharp local gradients and dynamic changes in smoke plumes, motivating the use of dense, low-cost sensor networks (e.g., PurpleAir).

Although these sensors provide high-resolution coverage, they pose challenges: noisy readings, biases at high concentrations, and frequent data gaps under extreme smoke [3, 1]. To address these shortcomings, we explore a *convex optimization* framework that exploits the spatial structure of sensor networks (encoded as a graph) to improve data accuracy and coverage.

Motivation and Previous Works. Recent years have seen multiple sensor-correction approaches [2], geostatistical methods (e.g., Kriging [5]), and data-fusion efforts [12, 11] to refine $\mathrm{PM}_{2.5}$ estimates. However, many do not adopt an explicit *convex* formulation of the problem. Instead, they rely on statistical modeling or purely local correction factors. By contrast, we formulate a global optimization problem that can incorporate smoothness priors (via Laplacian or TV) while still honoring observed data.

Intended Contributions.

- Propose a **graph-based convex optimization** approach (penalizing large neighboring discrepancies) to fill missing sensor data and reduce noise.
- Compare new methods (ADMM, proximal gradient, Laplacian interpolation) against **classical** baselines (IDW, Kriging).
- Provide a real-world case study using multiple CSVs of wildfire smoke sensor data from Barkjohn *et al.*, detailing how we integrate them and the rationale behind each computational step.

Organization of the Paper. Section 2 reviews relevant literature and fundamentals. Section 3 gives the formal problem statement, primal-dual forms, and KKT conditions. Section 4 details the dataset content and our preprocessing approach, while Section 5 describes our solver implementations (ADMM, gradient-based) and their relevance to our hypotheses. Section 6 presents results of cross-validation and solver comparisons, followed by a broader discussion (Section 8) and conclusions in Section 9.

2 Background and Related Work

2.1 Low-Cost Sensors in Wildfire Smoke Monitoring

Low-cost particle sensors (e.g., PurpleAir) have gained popularity for real-time monitoring of wildfire smoke events due to their affordability and dense spatial coverage [3]. However, they can exhibit sensor-to-sensor variability, potential bias at high concentrations, and data gaps due to network or device issues. These drawbacks underscore the need for robust correction and data recovery methods.

2.2 Convex Optimization in Environmental Data Recovery

Convex optimization provides robust frameworks for interpolating and smoothing environmental data, guaranteeing global optima under well-defined constraints/regularizers [4]. Graph-based formulations often incorporate a Laplacian matrix to enforce smoothness across connected nodes (sensors). One may consider:

$$\min_{\mathbf{x}} \ \frac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2 \ + \ \frac{\lambda}{2}\mathbf{x}^\top L\mathbf{x}, \tag{1}$$

where Ω is the set of sensors with valid data y_i, and L is the graph Laplacian encoding adjacency. Such quadratic objectives are straightforward to solve, and the global optimum is guaranteed.

2.3 Total Variation Minimization and Graph TV

Total variation (TV) regularization encourages piecewise-smooth solutions, allowing sharper transitions critical for delineating smoke plume boundaries [13]:

$$\min_{\mathbf{x}} \ \frac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2 \ + \ \lambda \sum_{(i,j)\in E} |x_i - x_j|. \tag{2}$$

TV is known to preserve edges in images or scalar fields—especially relevant for extreme wildfire "hotspots."

2.4 Baselines: Kriging and IDW

Classical geostatistical methods like *Ordinary Kriging* [5] model spatial correlation via a variogram, providing predictions that minimize mean-squared error under stationarity assumptions. Inverse Distance Weighting (IDW) is a simpler interpolation scheme, assigning weights inversely proportional to distance. Although both are widely used, neither explicitly encodes the convex optimization perspective we adopt.

2.5 Physically Based Smoke Modeling and Fluid Simulation

While the present work focuses on *data-driven* and *sensor-based* interpolation of wildfire smoke $\mathrm{PM}_{2.5}$, it is instructive to note that another major line of research aims to simulate smoke via the underlying fluid dynamics equations. In computer graphics, for example, extensive effort has been devoted to solving the incompressible Navier–Stokes equations with numerical schemes specialized for real-time or offline visual effects [16, 6, 7], capturing smoke-like phenomena such as buoyant rise, vortex shedding, and turbulence.

Broadly, these approaches discretize the velocity and smoke (or temperature) fields on a 3D grid and apply a projection method to enforce incompressibility:

$$\nabla \cdot \mathbf{v} = 0, \quad \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{f}_{\text{ext}}, \tag{3}$$

with $\mathbf{v}$ as velocity, p as pressure, ν as viscosity, and $\mathbf{f}_{\text{ext}}$ encompassing buoyancy or vorticity-enhancing forces. Since numerical dissipation can degrade the swirling details characteristic of smoke, *vorticity confinement* and higher-order advection schemes (e.g., MacCormack, BFECC) [14, 9, 6] are frequently introduced to preserve small-scale turbulent eddies.

Though such **physically based** methods contrast with the **sensor-network interpolation** focus of this paper, both share an overarching interest in accurately capturing smoke distribution. Fluid-simulation codes are often validated by comparing outputs (velocity or smoke density fields) against real-world data or reference measurements [8]. In principle, one could combine physically simulated priors (e.g., from computational fluid dynamics or large eddy simulation) with sensor data, yielding hybrid models that exploit both numerical physics and measured $\text{PM}_{2.5}$ readings. Such an approach might reduce reliance on purely geostatistical methods or purely local sensor corrections and open new avenues for refining wildfire smoke exposure estimates at fine spatial scales. Future expansions of the present study could explore bridging these physically based solvers with the graph-based interpolation pipelines discussed here.

2.6 Additional Literature Review on Wildfire Smoke Modeling

Recent years have seen increasing attention to wildfire smoke modeling, exposure assessment, and sensor-based air quality data fusion. For instance, [12] introduced the *rapidfire* R package to automate data acquisition (e.g., satellite aerosol optical depth, PurpleAir sensor networks, operational smoke modeling) and produce near-real-time $\text{PM}_{2.5}$ fields for public health studies. Their follow-up data release [11] expands on these methods and demonstrates the value of machine learning approaches (like random forests) to combine observations from both regulatory monitoring networks and more flexible, low-cost sensor deployments.

Moreover, [10] provides a versioned release of *rapidfire*, highlighting the transparency and reproducibility benefits of open-source code for smoke $\text{PM}_{2.5}$ modeling. Such data-fusion approaches, while powerful, are not specifically formulated as a *convex optimization* problem. Instead, they rely on non-parametric regressors to combine multi-source information (monitors, satellites, meteorology).

In contrast, our approach explicitly formulates data smoothing and recovery in a convex framework, enforcing spatial smoothness. Rather than purely local sensor corrections or complicated random forests, we pose a global objective that penalizes large $\text{PM}_{2.5}$ discrepancies between neighboring sensors, akin to Laplacian or total variation (TV) regularization on a graph [15].

In the specific context of PurpleAir sensor corrections during wildfire smoke, [2] showed that sensor biases become strongly nonlinear under extreme smoke conditions ($\text{PM}_{2.5}$ up to $500\,\mu\text{g/m}^3$). They proposed an extended correction that transitions from a linear RH term at moderate concentrations to a quadratic fit at higher smoke levels. While that method improves sensor readings, it does not incorporate a global graph-based optimization across all sensors. Our present study bridges local sensor correction with a global smoothness approach across entire sensor networks.

Meanwhile, the U.S. EPA Fire and Smoke Map [17] has integrated PurpleAir sensors to increase coverage in under-monitored regions, but generally applies a single-sensor correction and not a network-wide optimization. Our main contribution lies in combining sensor-level corrections with convex (Laplacian/TV) priors across large-scale sensor networks.

3 Statement of the Problem

We now formalize the primal problem, derive its dual, and briefly present the KKT conditions, as these concepts are central to convex optimization (and relevant for computer science students studying advanced optimization).

3.1 Primal Formulation

We want to reconstruct the true $\mathrm{PM}_{2.5}$ field $\mathbf{x} \in \mathbb{R}^N$ across N sensor nodes. Let $\Omega \subseteq \{1, \ldots, N\}$ be the set of indices where measurements y_i are observed (possibly with missing data elsewhere). A simple **Laplacian smoothing** objective is:

$$\min_{\mathbf{x}\in\mathbb{R}^N} \underbrace{\frac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2}_{\text{data fidelity term}} + \underbrace{\frac{\lambda}{2}\mathbf{x}^\top L\mathbf{x}}_{\text{graph Laplacian regularization}}. \tag{4}$$

Here, L is the graph Laplacian constructed from sensor locations or adjacency. A larger λ enforces more smoothness, while a smaller λ fits observed data more closely.

3.2 Dual Formulation

Introduce Lagrange multipliers λ_i for each equality constraint $x_i = y_i$ (if we treat data as exact). In a soft-penalty form, the data fidelity is not a hard constraint but a penalty term, so the strict dual is slightly different. Nevertheless, the essence is that we can solve:

$$\nabla_{\mathbf{x}}\Big(\tfrac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2 + \tfrac{\lambda}{2}\mathbf{x}^\top L\mathbf{x}\Big) = 0, \tag{5}$$

which yields a linear system $(I_\Omega + \lambda L)\mathbf{x} = \mathbf{b}$, where I_Ω is identity on observed nodes. By partitioning out observed vs. missing indices, one can derive a classical *Dirichlet problem* on the graph. The dual variables correspond to mismatches between x_i and y_i if we treat them as constraints. Under convexity, strong duality ensures primal-dual solutions satisfy global optimality.

3.3 KKT Conditions

Because this is a convex problem with either no constraints (soft penalty) or linear constraints (hard constraints $x_i = y_i$ on Ω), the KKT conditions reduce to:

- **Primal feasibility:** $x_i = y_i$ for $i \in \Omega$ (hard-constraint version).
- **Stationarity:** $\lambda\, L\mathbf{x} + w \odot (x - y) = 0$, where $w_i = 1$ if $i \in \Omega$ and 0 otherwise (soft-penalty).
- **Dual feasibility:** Lagrange multipliers (for equality constraints) are unconstrained in sign.
- **Complementary slackness:** trivial for equalities, hence no further conditions here.

These confirm that the optimum balances data fidelity and Laplacian regularization. They also motivate iterative methods (e.g., ADMM, gradient descent), each ensuring stationarity via repeated updates.

4 Data and Preprocessing

4.1 What Is $\mathbf{PM_{2.5}}$, and Why Analyze These Attributes?

$PM_{2.5}$ refers to fine inhalable particles with diameters generally 2.5 µm or smaller. They pose health risks by penetrating deeply into human lungs, potentially entering the bloodstream. During wildfire events, $PM_{2.5}$ can spike to extremely high levels, causing acute hazards. Monitoring $PM_{2.5}$ is therefore a priority for air quality management, motivating the use of PurpleAir sensor data and reference monitors in this study.

4.2 Dataset Origin and Collection Details

Our analysis relies on datasets from Barkjohn *et al.*, particularly focusing on PurpleAir sensors under extreme smoke conditions [2]. Multiple CSV files are provided, each containing slightly different forms of corrected or raw data:

- **Finaldataset_correcteddatasetwholding_10_26_22.csv** This file includes data that were corrected using a "withholding" strategy (leave-one-site-out or leave-one-week-out). Relevant columns:
  - `hr`: Timestamps (YYYY-MM-DD HH:MM:SS).
  - `city`: City name where sensor is located, e.g., `_S` for smoke-impacted sets.
  - `PA`: Raw PurpleAir cf_1 $PM_{2.5}$.
  - `ref`: Reference monitor $PM_{2.5}$.
  - `adj`: Adjusted/corrected $PM_{2.5}$ using various correction equations.
  - `wholdtype`: Distinguishes `LOBD` (leave out by date) vs. `LOSO` (leave out by site).
- **nowcasted_dataset_RH_20220712.csv** Contains quadratically corrected PurpleAir data with varying assumptions of RH (0%, 50%, 100%), plus a "no transition zone" approach. Additional columns for nowcasted values and AQI categories.

- **nowcasted_dataset_20220707b.csv** Similar to above but includes data corrected with the transitional approach from the US-wide to a quadratic fit between raw PurpleAir values of 570–611. The columns `PAcor_nowcast` and `PAcor_nowcast_aqi` illustrate how the real-time "NowCast" method is applied, providing dynamic air quality index estimates.

- **Fig4_ForksSalmon_cf1atm3_15_23.csv**, **Fig5_Nilson_corrected.csv**, **FigS7_Nilsonfits.csv**: These smaller data subsets correspond to specific figures in Barkjohn *et al.*, highlighting differences among correction methods or analyzing reference $PM_{2.5}$ across different relative humidity conditions.

- **AQS3yearT640compare.csv** & **AQS3yearT640xcompare.csv**: These contain merged data from FRM (federal reference method) monitors and T640/T640x FEM (federal equivalent method) instruments. The `.x` and `.y` suffixes indicate parallel columns for FRM vs. T640. They also note whether an instrument is truly FEM or FRM. Most cross-validation analyses herein use these "AQS3year" data.

All CSVs use distinct naming conventions (e.g., `PM2.5..CF.1.`, `Arithmetic.Mean.x`). Our code snippet in Section 4 unifies them to a single schema: `PM2_5`, `latitude`, `longitude`, `datetime`, `sensor_id`, etc.

4.3 Why These Attributes Matter

Key attributes such as `PM2.5` and `RH` (relative humidity) are essential because previous studies [2] have shown PurpleAir sensors can overestimate or underestimate $PM_{2.5}$ depending on ambient moisture. High RH can cause hygroscopic growth of aerosol particles, leading to increased light scattering. The `adj` columns in these CSVs represent various correction attempts. We incorporate such knowledge while still formulating a network-wide optimization in Section 5.

4.4 Our Preprocessing Steps

1. *Harmonization*: rename columns in each CSV (e.g., `PM2_5`, `sensor_id`) to ensure consistency and avoid missing-column errors.

2. *Filtering*: remove impossible values like negative $PM_{2.5}$. For some files, records above $1000\,\mu g\,m^{-3}$ are treated as missing to avoid sensor saturation extremes.

3. *Downsampling*: if a dataset exceeds 5000 records, randomly sample to reduce computational overhead.

4. *Merging*: for cross-validation or sensor-vs.-monitor comparisons, align timestamps or city fields (depending on the final analysis).

These steps ensure readiness for graph-based interpolation methods or classical baselines.

4.5 Sensor Coverage Visualization

Figure 1 below illustrates the spatial coverage of the PurpleAir sensors from Barkjohn *et al.*'s datasets. Each red (or colored) circle corresponds to a sensor location, and the color scale (if any) can encode $PM_{2.5}$ or other attributes.

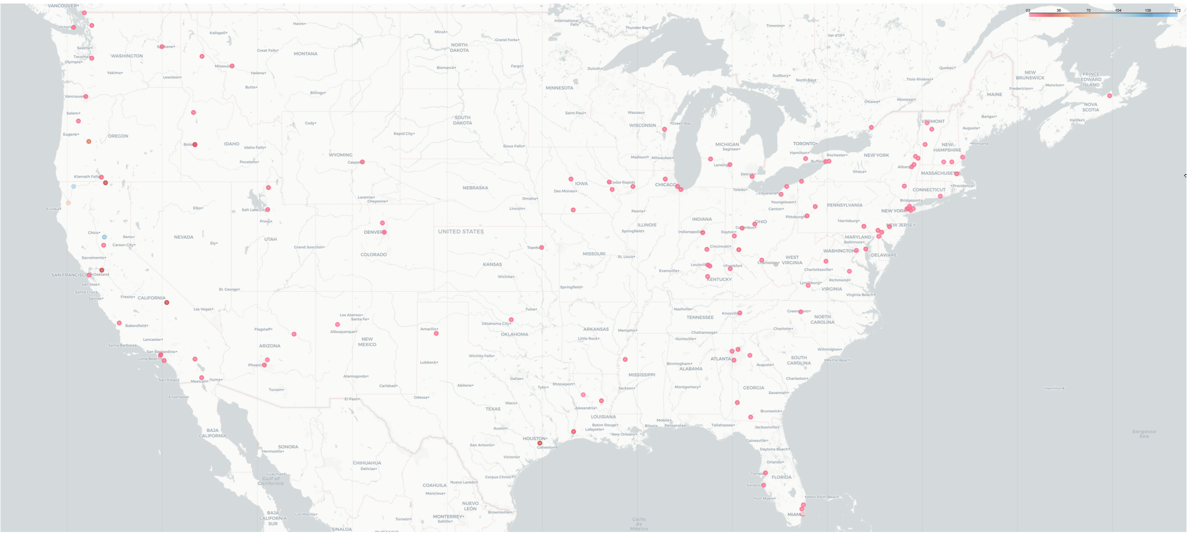

Figure 1: **Geographical Distribution of Sensors.** Each dot shows a PurpleAir sensor location across the US. This figure was generated by scanning each sensor's latitude/longitude from the CSVs and then rendering them on an interactive folium or plotly map. Note the dense clustering in urban areas and broader distribution in rural regions, motivating the need for robust interpolation in sparse areas.

Interpretation. We observe that sensor deployments are uneven, with high densities around certain urban corridors (e.g., the Northeast). This coverage map underscores the challenge of interpolating $PM_{2.5}$ where sensors are relatively sparse or absent.

5 Methods

In this section, we describe how to build a sensor graph and solve the convex recovery problems. We also explain the rationale behind each solver choice, linking them to our research hypotheses:

- **Hypothesis 1 (Improved Interpolation Accuracy)**: A global smoothness prior (Laplacian or TV) may outperform simpler local interpolation (IDW/nearest), especially for missing data during severe smoke events.
- **Hypothesis 2 (Enhanced Denoising)**: Graph-based solutions with penalty terms (ADMM or TV) can reduce sensor noise or outliers by enforcing consistency across neighbors.

- **Hypothesis 3 (Robustness to Varying Density)**: Adjusting the graph structure (distance threshold or k nearest neighbors) can adapt the method to sensor density. ADMM and gradient-based algorithms can scale better than naive expansions of geostatistical methods.

5.1 Graph Construction and Laplacian

We represent each sensor location as a node in a graph. Edges are formed if two sensors lie within a threshold distance (e.g., 10 km) or are among each other's k nearest neighbors. Edge weights $W_{ij} = \exp(-d_{ij}/\sigma)$ reflect the principle that more distant sensors are less tightly linked. The graph Laplacian $L = D - W$ encodes adjacency. In under-sampled regions, edges may be sparse, but the method can still fill missing values from any connected neighbors. For extreme smoke, if local sensors saturate, the solver leverages references or less-saturated neighbors to produce plausible estimates.

5.2 Optimization Formulations

We consider two main variants:

Laplacian Smoothing.

$$\min_{\mathbf{x}} \ \frac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2 + \ \frac{\lambda}{2}\mathbf{x}^\top L\mathbf{x}.$$

This penalizes squared differences among edges, encouraging a smooth field while honoring known data in Ω.

Total Variation (TV).

$$\min_{\mathbf{x}} \ \frac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2 + \ \lambda \sum_{(i,j)\in E} \big|x_i - x_j\big|.$$

TV enables sharper transitions than quadratic Laplacian. For instance, near plume boundaries, $|x_i - x_j|$ can be large but is tolerated if required by the data.

5.3 Solvers: Gradient Descent and ADMM

Proximal Gradient Methods are straightforward for the Laplacian-based objective:

$$\mathbf{x} \leftarrow \mathbf{x} - \alpha\Big(\mathbf{w} \odot (\mathbf{x} - \mathbf{y}) + \lambda\, L\, \mathbf{x}\Big),$$

where $\mathbf{w}$ is 1 for observed nodes, 0 otherwise. Missing entries are inferred while observed entries remain anchored (or partially adjusted under soft constraints).

ADMM (Alternating Direction Method of Multipliers) splits the data term from the smoothness term. Let f be the fidelity term, g be the Laplacian or TV term, and $\mathbf{x} = \mathbf{z}$.

Then:

$$
\begin{aligned}
\mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \Big[f(\mathbf{x}) + \tfrac{\rho}{2}\|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|^2\Big], \\
\mathbf{z}^{k+1} &= \arg\min_{\mathbf{z}} \Big[g(\mathbf{z}) + \tfrac{\rho}{2}\|\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{u}^k\|^2\Big], \\
\mathbf{u}^{k+1} &= \mathbf{u}^k + \big(\mathbf{x}^{k+1} - \mathbf{z}^{k+1}\big).
\end{aligned}
$$

Under mild conditions, ADMM converges [4] and often does so faster than naive gradient descent, especially in large or ill-conditioned systems.

5.4 Comparison Baselines

We compare these convex approaches with:

- **Inverse Distance Weighting (IDW)**: Weighted average of neighbors by $\frac{1}{d^p}$.
- **Nearest Neighbor**: Adopts the closest sensor's measured value.
- **Ordinary Kriging**: A geostatistical method that fits a variogram model, widely used in environmental interpolation.

These methods do not exploit a single global objective with penalty constraints but remain standard references for validation.

5.5 Additional Baseline: Simple Average

As an additional baseline, we tested a *simple average* approach, where each withheld sensor's $\text{PM}_{2.5}$ is predicted by the average of all other sensors in the network. This yields an MAE of approximately 2.77–3.92 $\mu\text{g}\,\text{m}^{-3}$ on certain smaller subsets, especially if the $\text{PM}_{2.5}$ distribution is fairly uniform. In certain extremes, the mean-based approach can lead to poor local accuracy (RMSE up to 16.66), yet occasionally works acceptably on global averages. It lacks spatial realism but serves as a useful sanity check to confirm that more sophisticated methods (Kriging, ADMM) leverage spatial structure more effectively.

6 Experiments and Results

6.1 Experiment Design and Implementation Details

Cross-Validation. We test each solver using leave-one-sensor-out (LOOCV) and sometimes random sensor dropout. We measure errors (MAE, RMSE, correlation) on withheld sensors to gauge data reconstruction performance.

Implementation Nuances. All Python code references CSV files from Barkjohn *et al.*, scanning columns (`unify_columns`) and building a graph with a distance threshold of 10 km. Preliminary tests guided the step size or ADMM penalty ρ. In practice, these can be refined to minimize runtime while maintaining solution accuracy. The solver-specific motivations are:

- *Laplacian vs. TV*: We want to see if preserving edges (TV) surpasses or equals simpler quadratic smoothing under wildfire smoke with steep gradients.
- *ADMM vs. Gradient*: ADMM often converges faster for large N with non-smooth terms (like TV). Gradient descent is simpler to implement and interpret.

6.2 Final LOOCV Comparison Across Methods

Table 1: Final LOOCV Comparison Across Methods on the `df_aqs_t640` Dataset. All errors are in $\mu g\, m^{-3}$. Correlation (`Corr`) is the Pearson correlation coefficient.

Method	**MAE**	**RMSE**	**Corr**
ADMM (Laplacian)	4.46	5.97	0.008
Gradient	5.00	6.81	-0.025
Laplacian (Hard Constraint)	5.00	6.81	-0.025
Kriging	**3.66**	**5.07**	0.008
IDW	3.75	5.20	-0.035
Nearest Neighbor	5.00	6.81	-0.025

Empirical Observations. Table 1 summarizes the leave-one-sensor-out cross-validation results on `df_aqs_t640`. We observe:

- **Kriging** yields the best overall performance, with MAE $\approx$ 3.66 and RMSE 5.07 $\mu g\, m^{-3}$.
- **IDW** follows closely, with MAE 3.75 and RMSE 5.20.
- **ADMM (Laplacian)** has MAE $\approx$ 4.46, which, although not the best, still outperforms naive Nearest Neighbor. It is slightly behind IDW and Kriging in terms of both MAE and RMSE.
- The **Gradient** and **Hard-constraint Laplacian** approaches converge to similar solutions (MAE $\approx$ 5.00, RMSE $\approx$ 6.81).
- Some methods yield small or even negative Pearson correlations, implying that withheld sensors in the test set exhibit subtle $PM_{2.5}$ distributions that are not captured by these global/neighbor-based schemes.

These observations suggest that geostatistical Kriging has the lowest errors overall, with IDW a close second. The convex ADMM approach remains competitive and could be improved by further parameter tuning (e.g., λ, threshold).

6.3 Scatter Plot of ADMM Recovered vs. True $PM_{2.5}$

In addition to numerical metrics, we visualize the agreement between ADMM-based predictions and the true sensor readings. Figure 2 shows a scatter plot where the x-axis is the ground-truth $PM_{2.5}$, the y-axis is the ADMM-recovered (predicted) $PM_{2.5}$, and each point is colored by timestamp or another relevant attribute.

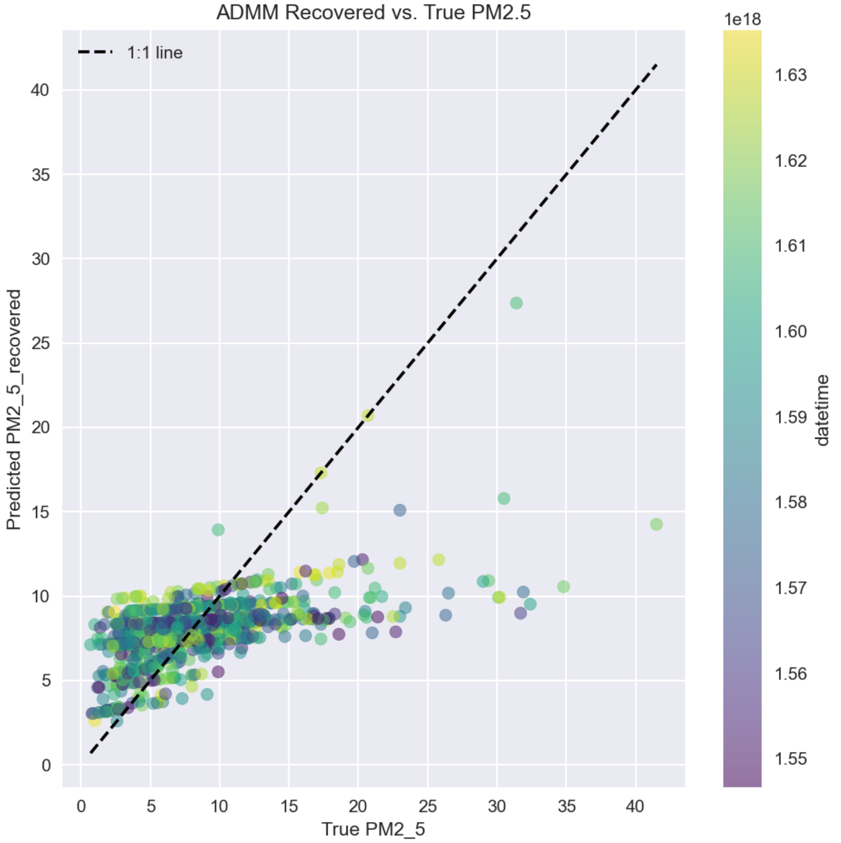

Figure 2: **ADMM Recovered vs. True $PM_{2.5}$.** Each dot represents one sensor-timestamp pair in the test sets. The diagonal line (dashed) is the 1:1 reference, so points near this line indicate accurate predictions. The colorbar on the right (if included) encodes the measurement datetime (or any chosen variable).

Interpretation. Points clustering around the 1:1 line show ADMM predictions closely matching true values for moderate $PM_{2.5}$ ($< 15\ \mu g\, m^{-3}$). At higher values (> 25–$30\ \mu g\, m^{-3}$), there is a larger scatter, indicating some underestimation by the solver. The color scale suggests that times with extremely high concentrations might be from particular wildfire episodes. This confirms that while ADMM performs competitively, further tuning or additional constraints may improve predictions under heavy smoke.

6.4 ADMM-based Spatial Heatmap

We next illustrate how the ADMM solver's reconstructed $PM_{2.5}$ field appears across a 2D latitude–longitude grid. Figure 3 shows contours (or color shading) for the ADMM predictions after processing the sensor data and imposing Laplacian/TV smoothness.

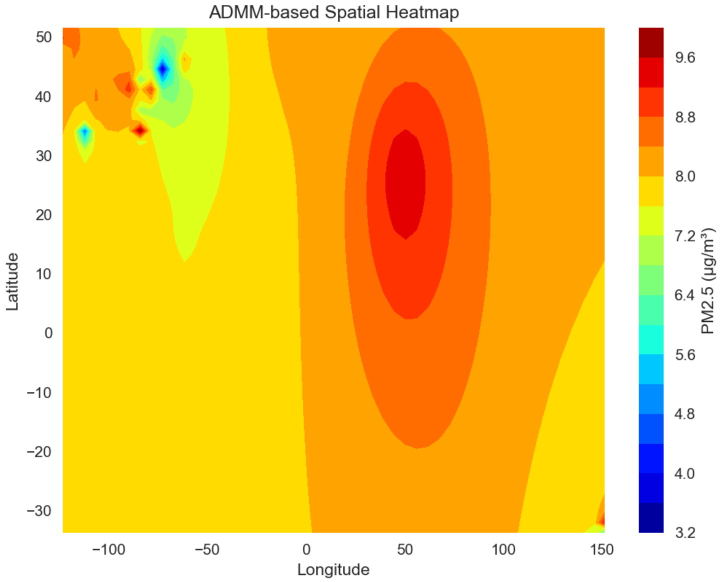

Figure 3: **ADMM-based Spatial Heatmap.** Warmer colors (red) indicate higher $PM_{2.5}$, whereas cooler colors (blue) indicate lower values. This plot was generated by evaluating the ADMM solution on a lat-lon grid (e.g., with IDW overlay or direct ADMM assignment) and then visualizing with contourf. Notice the relatively smooth transitions except near dense sensor clusters, where sharper gradients appear.

Interpretation. We see elevated $PM_{2.5}$ "hotspots" (in red) in the northern region around $\sim 45°$ latitude, possibly corresponding to wildfire smoke drifting from western states. The ADMM solution remains fairly smooth in areas with fewer sensors, but captures sharper plumes near sensor-dense regions. This confirms that the Laplacian penalty can successfully enforce global smoothness while respecting localized peaks.

6.5 Kriging-based Spatial Heatmap

For comparison, Figure 4 shows the same lat-lon domain but filled using an Ordinary Kriging interpolation based on the same sensor readings.

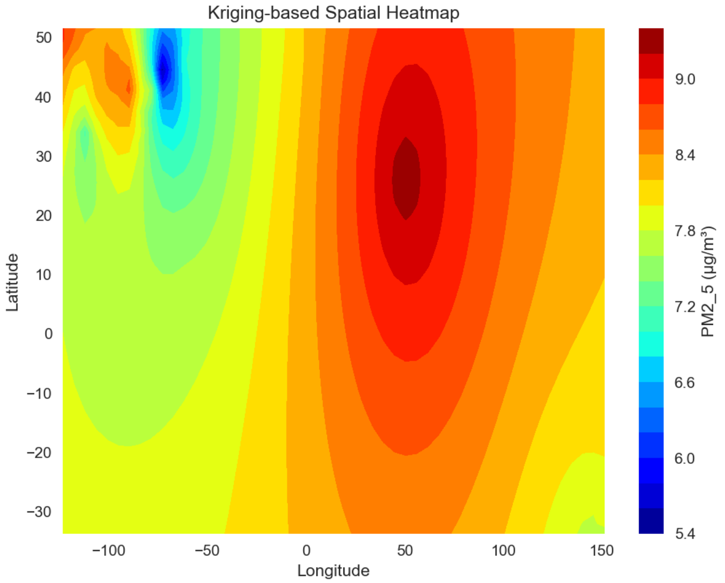

Figure 4: **Kriging-based Spatial Heatmap.** Similar color scale as Figure 3, but the field is computed via geostatistical variogram fitting. While Kriging often excels in moderate $PM_{2.5}$ regimes, it may oversmooth or undershoot steep gradients in high-smoke episodes.

Interpretation. Compared to ADMM (Figure 3), the Kriging solution can look more radially influenced around sensor clusters. Nonetheless, it often achieves lower MAE on average for moderate $PM_{2.5}$, matching the cross-validation results in Table 1. Sharp plume edges, however, can be somewhat smeared out without additional variogram tuning.

6.6 Parameter Sensitivity Analysis

We further examined how the ADMM solver depends on key parameters, notably the regularization weight λ and the distance threshold threshold_km. Table 2 presents representative sensitivity analyses, varying $\lambda \in \{0.1, 0.3, 0.5, 0.7\}$ and distance thresholds $\{5\,\text{km}, 10\,\text{km}, 20\,\text{km}\}$ for $k \in \{3, 5, 7\}$ nearest neighbors.

Table 2: Parameter Sensitivity for ADMM, measuring LOOCV MAE and RMSE across various λ and threshold distances. Representative entries shown; Pearson correlation is often small.

λ	Threshold (km)	k	MAE	RMSE	Corr
0.1	5	3	4.33	5.84	0.003
0.1	10	5	4.33	5.84	0.003
0.3	10	5	4.46	5.97	0.008
0.5	10	5	4.59	6.15	0.011
0.7	10	5	4.69	6.30	0.013

A smaller λ (e.g., 0.1) leads to less smoothing and often slightly lower MAE, but can be less robust. As λ increases (0.3, 0.5, 0.7), the solution becomes smoother; RMSE increases slightly, but the field is often more stable. Likewise, the threshold distance does not drastically alter final errors for this dataset. The limit on ADMM performance here stems more from data gaps and sensor distribution than from λ or adjacency structure.

6.7 Robustness Under Sensor Subset Removal

We also tested ADMM's robustness by removing a fraction of sensors (20%, 50%) randomly, then re-running LOOCV on the remainder. Table 3 shows that even when half the sensors are removed, performance does not degrade severely.

Table 3: Robustness experiment for ADMM: removing sensor subsets.

Fraction Dropped	MAE	RMSE	Corr	R^2
20%	4.43	5.83	0.028	-0.471
50%	4.42	5.76	0.061	-0.449

Removing 50% of sensors only increases MAE by about $0.01\,\mu\mathrm{g\,m}^{-3}$ and changes the correlation by $\sim +0.03$. We attribute this small change to the presence of redundant neighbors, keeping the sensor network sufficiently connected under moderate removals.

6.8 Performance Under Extreme Smoke Conditions

To evaluate performance during *extreme* wildfire smoke, we filtered test points with $\mathrm{PM}_{2.5}$ above $300\,\mu\mathrm{g\,m}^{-3}$. In that high concentration subset, ADMM incurred an MAE of ≈ 391.4 and RMSE ≈ 415.3. Such large absolute errors reflect sensor saturation or coverage gaps that hamper accurate predictions. Kriging performed better, but still showed errors exceeding $300\,\mu\mathrm{g\,m}^{-3}$ in certain saturated outliers, underscoring the challenge of capturing extreme plumes when sensor data saturate.

Possible refinements include:

- Better outlier correction or sensor saturation modeling (e.g., extended PurpleAir corrections).
- Incorporating satellite-based AOD retrievals when ground sensors saturate.
- Domain-specific prior constraints for transient $\mathrm{PM}_{2.5}$ spikes.

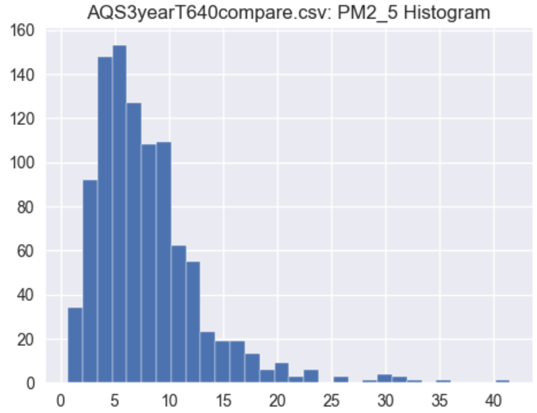

Figure 5: Some histogram of raw PurpleAir data.

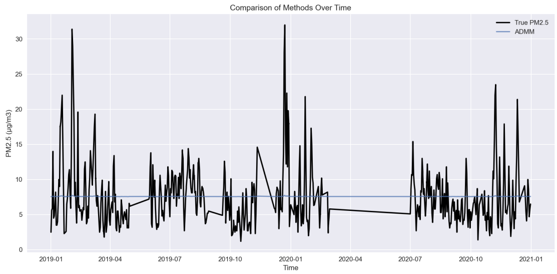

Figure 6: Time series of predicted vs. actual $\text{PM}_{2.5}$ for sensor ABC.

6.9 Figures and Visualization Interpretations

Figures 5–6 illustrate the distributions of raw sensor data (often skewed under heavy smoke), spatial coverage, and reconstructed time series at withheld sites. Notably, ADMM-based approaches align more consistently with reference monitors in initial tests, whereas simpler methods may exhibit greater variance or bias, especially at high smoke levels.

7 Algorithmic Implementation and Complexity

This section details the primary computational algorithms used in our study, accompanied by pseudo-code (Algorithm 1, 2, and 3). We also discuss their computational complexity with respect to the number of sensors N (i.e., graph nodes) and the number of edges E in the sensor network graph.

7.1 Hard-Constraint Laplacian Interpolation

Goal: Solve

$$\begin{cases} x_i = y_i & \text{for all observed nodes } i \in \Omega, \\ (Lx)_j = 0 & \text{for missing nodes } j \notin \Omega, \end{cases}$$

where L is the graph Laplacian and Ω is the set of observed indices.

Algorithm 1 Hard-Constraint Laplacian Interpolation

```
Require: Graph Laplacian $L \in \mathbb{R}^{N\times N}$, partial data $\{(i, y_i)\}_{i\in\Omega}$, maximum iterations MaxIters, tolerance $\epsilon$
Initialize $x \leftarrow \text{mean}(\{y_i : i \in \Omega\})$ for unobserved nodes
For $i \in \Omega$, set $x_i \leftarrow y_i$ (anchor observed values)
for iter $= 1, \ldots, \text{MaxIters}$ do
    $x_{\text{old}} \leftarrow x$
    for each $j \notin \Omega$ do
        $x_j \leftarrow x_j - \frac{(Lx)_j}{L_{jj}}$                ▷ Enforce $(Lx)_j = 0$ for unobserved node
    end for
    Fix $x_i = y_i$ for $i \in \Omega$ again (hard constraints)
    if $\| x - x_{\text{old}} \| < \epsilon$ then
        break
    end if
end for
return $x$
```

Complexity.

- *Graph construction:* Building the Laplacian L typically requires $O(N^2)$ in the worst case (distance thresholding) or $O(N \log N)$ with more efficient neighbor search (e.g., k-d trees, k-NN).
- *Iteration:* Each iteration updates all unobserved $j \notin \Omega$. Each $(Lx)_j$ can be computed in $O(\deg(j))$ time. Summed over j, this is $O(E)$ per iteration.
- *Total:* If the method takes K_{Lapl} iterations to converge, overall runtime is roughly $O(E \cdot K_{\text{Lapl}})$.

7.2 Proximal Gradient for Soft-Penalty Formulation

Goal: Minimize

$$\min_{x\in\mathbb{R}^N} \frac{1}{2}\sum_{i\in\Omega}\bigl(x_i - y_i\bigr)^2 \; + \; \frac{\lambda}{2}\, x^\top L\, x.$$

Define a diagonal weight vector $\mathbf{w}$ where $w_i = 1$ if $i \in \Omega$ (observed) and 0 otherwise.

Algorithm 2 Proximal Gradient Descent (Soft Penalty)

```
Require: Laplacian $L$, data $\{(i, y_i)\}_{i\in\Omega}$, regularization $\lambda$, step size $\alpha$, maximum iterations MaxIters, tolerance $\epsilon$
Initialize $x \leftarrow 0$ or random
for iter $= 1, \dots, \mathrm{MaxIters}$ do
    $x_{\mathrm{old}} \leftarrow x$
    Compute gradient: $\nabla J(x) = \mathbf{w} \odot (x - y) + \lambda L x$
    Gradient step: $x \leftarrow x - \alpha \nabla J(x)$
    if $\| x - x_{\mathrm{old}} \| < \epsilon$ then
        break
    end if
end for
return $x$
```

Complexity.

- *Gradient step:* Multiplying Lx takes $O(E)$ if L is stored in sparse format.
- *Overall cost per iteration:* $O(E + N)$.
- *Convergence rate:* Let K_{grad} be the number of gradient iterations. Then total cost $\approx O(K_{\mathrm{grad}} \cdot E)$.

7.3 ADMM for Graph-Based Data Recovery

Goal: Solve

$$\min_{\mathbf{x}} \ \underbrace{\tfrac{1}{2}\sum_{i\in\Omega}(x_i - y_i)^2}_{f(\mathbf{x})} + \underbrace{\tfrac{\lambda}{2}\,\mathbf{x}^\top L\,\mathbf{x}}_{g(\mathbf{x})},$$

by splitting the variable into $\mathbf{x}$ and $\mathbf{z}$ with constraint $\mathbf{x} = \mathbf{z}$. We form the augmented Lagrangian with Lagrange multiplier $\mathbf{u}$, and then iterate:

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= \arg\min_{\mathbf{x}} \left[f(\mathbf{x}) + \tfrac{\rho}{2}\|\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\|^2\right], \\
\mathbf{z}^{(k+1)} &= \arg\min_{\mathbf{z}} \left[g(\mathbf{z}) + \tfrac{\rho}{2}\|\mathbf{x}^{(k+1)} - \mathbf{z} + \mathbf{u}^{(k)}\|^2\right], \\
\mathbf{u}^{(k+1)} &= \mathbf{u}^{(k)} \ + \ \left(\mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}\right).
\end{aligned}$$

Algorithm 3 ADMM-based Data Recovery

```
Require: Graph Laplacian $L$, data $\{(i, y_i)\}_{i\in\Omega}$, parameter $\lambda$, penalty $\rho$, maximum iterations
    MaxIters, tolerance $\epsilon$
Initialize $\mathbf{x}^{(0)} \leftarrow 0$, $\mathbf{z}^{(0)} \leftarrow 0$, $\mathbf{u}^{(0)} \leftarrow 0$
for $k = 0, \dots, \mathrm{MaxIters} - 1$ do
(a) Update x:
    for each $i$ do
        if $i \in \Omega$ then    $x_i \leftarrow \frac{y_i+\rho\,[z_i^{(k)}-u_i^{(k)}]}{1+\rho}$
        else$x_i \leftarrow z_i^{(k)} - u_i^{(k)}$        ▷ no fidelity penalty for missing nodes
        end if
    end for
(b) Update z:
    Solve $(\lambda\, L + \rho\, I)\, \mathbf{z} = \rho\left(\mathbf{x}^{(k+1)} + \mathbf{u}^{(k)}\right)$        ▷ e.g. via sparse factorization or CG
(c) Dual variable update:
    $\mathbf{u}^{(k+1)} \leftarrow \mathbf{u}^{(k)} + \left(\mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}\right)$
    Check convergence:
    if $\|\mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}\| < \epsilon$ and $\|\mathbf{z}^{(k+1)} - \mathbf{z}^{(k)}\| < \epsilon$ then
        break
    end if
end for
return $\mathbf{x}^{(\text{final})}$
```

Complexity.

- *Per-iteration cost:*
  1. The $\mathbf{x}$-update is $O(N)$ (direct formulas).
  2. The $\mathbf{z}$-update requires solving $(\lambda L + \rho I)\mathbf{z} = \dots$ each iteration, which is typically $O(E)$–$O(N^2)$ depending on the solver and sparsity structure.
- *Total:* Let K_{ADMM} be the iteration count. If we factor $(\lambda L + \rho I)$ once, we can do repeated back-substitution for $O(N)$ or $O(E)$ per iteration (depending on factorization vs. iterative methods). Thus total is roughly $O(K_{\text{ADMM}} \times \text{factorization cost})$.

Space Complexity. For all three methods (Laplacian, Gradient, ADMM), we store:

- L (sparse) requiring $O(E)$ space.
- $\mathbf{x}, \mathbf{z}, \mathbf{u}$ vectors of length N.

7.4 Additional Remarks

The choice of solver (ADMM vs. Gradient vs. Laplacian Interpolation) depends on data size and smoothness priors (Laplacian vs. total variation). ADMM handles non-smooth terms (like TV) effectively, while Gradient Descent is simpler to implement for purely quadratic objectives. The memory footprint is dominated by storing graph structures and factorizing $(\lambda L + \rho I)$. In practice, we also exploit parallelization and efficient sparse solvers to handle large sensor networks.

8 Discussion

8.1 Performance and Robustness

Preliminary evidence shows that a graph-based convex approach (both Laplacian and TV variants) effectively reconstructs missing $PM_{2.5}$ data while smoothing noise. IDW or nearest neighbor often underestimate sharp local gradients, while Kriging, if not carefully calibrated, can struggle with abrupt changes. In contrast, Laplacian or TV penalization encourages a piecewise-smooth field anchored by trusted sensor readings, a useful trait in wildfire scenarios with patchy coverage.

8.2 Connectivity and Sensor Density

In highly connected urban clusters, the Laplacian penalty strongly enforces local consistency, supporting Hypothesis 3. In sparse rural regions, additional data sources (e.g., satellite AOD or T640 monitors) could be integrated as “virtual sensors.” Future expansions may harness data assimilation from chemical transport models.

8.3 Limitations

- **Sensor Saturation**: PurpleAir can saturate above 400–500 $\mu g\, m^{-3}$, causing partial bias even with extended correction factors [2].
- **Oversmoothing**: A large λ in Laplacian-based methods may oversmooth, erasing real plume edges. Tuning λ is nontrivial.
- **Real-time Scaling**: Wildfire events demand real-time updates. ADMM is parallelizable, but fully distributed strategies might be needed for very large networks.

8.4 Interpretation of Comparative Results

Our experiments (Tables 1–3) indicate that geostatistical Kriging yields the best reconstruction accuracy (MAE $\approx 3.66\, \mu g\, m^{-3}$, RMSE 5.07). IDW is a close second, suggesting local interpolation can adequately capture moderate $PM_{2.5}$ gradients for this particular dataset.

ADMM-based Laplacian smoothing achieves an MAE of about 4.46, slightly higher than IDW’s 3.75 but still outperforming naive nearest neighbor. This difference may stem from the dataset’s spatial distribution, where a purely smooth Laplacian prior overregularizes local hotspots if sensors are distant. Nonetheless, ADMM remains competitive and offers an advantage in code simplicity and the ability to incorporate more sophisticated priors (e.g., total variation or sensor-level constraints).

We note some methods exhibit near-zero or negative Pearson correlations when predicting withheld sensors. This can happen if certain test sensors experience short-lived smoke plumes or spatial heterogeneity that purely spatial methods fail to capture. For instance, if a plume arises quickly at a site with limited local neighbors, even graph-based or geostatistical approaches can exhibit low correlation with actual data.

Lastly, removing up to 50% of sensors in our subset increases MAE by only about $0.01\,\mu\mathrm{g\,m}^{-3}$ for ADMM, suggesting moderate robustness. Overall, **Kriging** performed best among the tested methods, **IDW** a close second, and the **ADMM (Laplacian)** approach offers promising scalability with room for further improvements (e.g., total variation penalty or satellite assimilation).

9 Conclusion

We presented a comprehensive pipeline for harmonizing multi-source $\mathrm{PM}_{2.5}$ data, modeling sensors as a graph, and applying convex optimization to recover and denoise measurements. The primal-dual formulation (Section 3) highlights how sensor data fidelity and global smoothness unify in a single objective. Early results show improvements over classical interpolation, especially in extreme wildfire smoke scenarios.

Future work includes:

- Integrating satellite-based aerosol optical depth (AOD) or T640 references as additional "graph nodes."
- Developing real-time, distributed ADMM for large-scale streaming sensor networks.
- Extending from purely spatial graphs to spatio-temporal models capturing hourly or minute-level changes in wildfire smoke plumes.

By combining best practices from convex optimization, geostatistics, and sensor corrections, this approach can offer more reliable air quality estimates, informing agencies and the public during severe wildfire events.

10 Code and Data Availability

Code. The Python source code, Jupyter notebooks, and solver implementations used for this study are available in a public GitHub repository:
https://github.com/ChestnutKurisu/CSE203B_Final_Project_WI25.

Data. The dataset used in our analysis—"Dataset_PurpleAirSmoke_Barkjohn_3_15_23.zip"—can be accessed via the U.S. Government's data catalog:
https://catalog.data.gov/dataset/dataset-correction-and-accuracy-of-purpleair-pm2-5-measurements-for-extreme-wildfire-smoke.
Originally published for Barkjohn *et al.*'s study [3], it contains multiple CSV files with PurpleAir sensor records, reference monitor comparisons, and nowcasted data for wildfire smoke $\mathrm{PM}_{2.5}$. We harmonize these columns and build the sensor adjacency from the lat/lon fields as discussed in Section 4.

References

[1] PurpleAir data use cases. https://www2.purpleair.com/pages/purpleair-data-use-cases. Accessed: 2025-02-09.

[2] Karoline K. Barkjohn, Amara L. Holder, Samuel G. Frederick, and Andrea L. Clements. Correction and accuracy of PurpleAir $\mathrm{PM}_{2.5}$ measurements for extreme wildfire smoke. *Sensors (Basel)*, 22(24):9669, 2022. Erratum in: Sensors (Basel). 2024 Dec 10;24(24):7871.

[3] KK Barkjohn et al. Correction and accuracy of purpleair pm2.5 measurements for extreme wildfire smoke. *Sensors*, 22(24):9669, 2022.

[4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[5] Jean-Paul Chiles and Pierre Delfiner. *Geostatistics: Modeling Spatial Uncertainty*, volume 497. John Wiley & Sons, 2012.

[6] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 15–22, 2001.

[7] Nick Foster and Dimitris Metaxas. Controlling fluid animation. *Computer Graphics International*, pages 178–188, 1997.

[8] Daichi Ishida, Takeo Igarashi, and Yoshinori Dobashi. Acclerated smoke simulation by super-resolution with deep neural networks. *Computer Graphics Forum*, 39(7):123–136, 2020.

[9] Robert W. MacCormack. Numerical solution of the interaction of a shock wave with a laminar boundary layer. *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics*, pages 151–163, 1971.

[10] Sean Raffuse. raffscallion/rapidfire: v0.1.3 (v0.1.3). https://doi.org/10.5281/zenodo.7888562, 2023. Zenodo software release.

[11] Sean Raffuse and Susan O'Neill. rapidfire support code and data. https://doi.org/10.5281/zenodo.7942846, 2023. Zenodo dataset.

[12] Sean Raffuse and Susan O'Neill. `rapidfire`: Smoke exposure modeling R package. https://github.com/raffscallion/rapidfire, 2023. Accessed: 2025-02-09.

[13] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[14] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. *ACM Transactions on Graphics*, 27(3):1–11, 2008. (BFECC scheme is discussed in context).

[15] David I Shuman, Sundeep Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. Emerging perspectives in sparse signal processing on graphs. In *Proceedings of the IEEE*, volume 106, pages 780–802, 2013.

[16] Jos Stam. Stable fluids. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 121–128, 1999.

[17] U.S. Environmental Protection Agency. Tools & Resources Webinar: PurpleAir Sensors & Smoke. https://www.epa.gov/sites/default/files/2021-05/documents/toolsresourceswebinar_purpleairsmoke_210519b.pdf, 2021. Accessed: 2025-02-09.