Traffic Flow Analysis Ensemble of STL-GRU and MegaCRN

Jessalin Jiangkhov
Cal Poly Pomona

Alisha Mehta
Cal Poly Pomona

Jessica Escalante
Cal Poly Pomona

Abstract

Accurate spatio-temporal traffic forecasting remains challenging due to the coexistence of complex temporal dynamics and structured spatial dependencies. In this work, we study a weighted averaging ensemble that combines two state-of-the-art forecasting models, STLGRU and MegaCRN, to leverage their complementary modeling strengths. The ensemble prediction is formulated as a convex combination of individual model outputs, with weights selected based on validation performance and evaluated using MAE, MAPE, and RMSE. Experiments on the PEMS-BAY and METR-LA datasets show that weighted ensembling consistently improves performance over single models on PEMS-BAY, while exhibiting dataset-dependent behavior on METR-LA. To bridge spatio-temporal forecasting with computer vision, we further reinterpret model predictions as image-like representations and apply spatial grid conversion and Gaussian smoothing to generate traffic heatmaps for qualitative analysis. These visualizations provide interpretable insights into spatial traffic patterns and model behavior, demonstrating that simple ensemble strategies combined with visual analysis offer an effective and generalizable framework for traffic forecasting without additional model complexity. Project Github link: https://github.com/JJ-223/ECE4990.02-FINAL-PROJECT/tree/main?tab=readme-ov-file

1. Introduction

Traffic flow analysis refers to the measurement of vehicle movement at a particular junction or node over time, typically defined as the number of vehicles passing through a location within a fixed interval. The goal of traffic flow forecasting is to predict future traffic conditions by exploiting complex spatial and temporal dependencies present in historical data. Accurate traffic prediction is a critical component of Intelligent Transportation Systems (ITS), enabling applications such as congestion mitigation, route planning, and urban mobility optimization.

Spatio-temporal forecasting plays a central role in traffic prediction and related real-world applications, as it requires modeling both temporal patterns—such as trends, periodicity, and short-term fluctuations—and spatial dependencies induced by road network structures. Although recent deep learning architectures have achieved strong performance in spatio-temporal forecasting, no single model consistently performs optimally across all datasets and traffic regimes, motivating the exploration of model combination strategies.

STLGRU and MegaCRN represent two complementary approaches to spatio- temporal modeling. STLGRU explicitly decomposes time series into trend, seasonal, and residual components prior to recurrent modeling, allowing it to effectively capture long-term periodic patterns and reduce systematic temporal bias. However, this decomposition can increase sensitivity to noise and lead to higher variance under rapidly changing or irregular traffic conditions. In contrast, MegaCRN focuses on graph-based spatial relationships and memory- enhanced temporal modeling, improving robustness across spatially correlated nodes but potentially underfitting fine-grained temporal dynamics.

Motivated by the complementary strengths of these architectures, we propose a weighted averaging ensemble that combines the predictions of STLGRU and MegaCRN through a convex linear combination. By varying the contribution of each model, the ensemble aims to balance bias and variance while reducing correlated prediction errors. We evaluate this ensemble strategy on two widely used benchmark datasets, PEMS-BAY and METR-LA, by sweeping the STLGRU weight across the interval $[0, 1]$ and measuring forecasting performance using standard regression metrics.

In addition to quantitative evaluation, we introduce a computer vision–based analysis framework to interpret ensemble predictions. Model outputs are transformed into image-like representations using spatial grid conversion, followed by Gaussian smoothing to generate traffic flow heatmaps. These visualizations provide qualitative insights into spatial traffic patterns and model behavior that are not captured by numerical metrics alone. The primary contribution of this work is a combined empirical and visual analysis demonstrating how weighted ensembling and computer vision techniques together enhance the interpretability and effectiveness of spatio-temporal traffic forecasting without additional training complexity.

2. Related Works

Spatio-Temporal Lightweight Graph GRU (STLGRU) is a graph convolution to model localized spatial relations then use an attention mechanism with a memory module to directly model the long-range local and non-local spatio- temporal dependencies [1]. In order to update the memory, we use a gating mechanism, where our gating strategy records the key local and global spatio- temporal information and forgets the redundant ones when moving to the next time step. The design of the model to be lightweight, as the memory module uses fewer parameters than the existing baselines. Consequently, it can effectively learn long-range dependencies without the need to use multi-scale causal convolution or stacking past time step features.

Spatio-temporal data, such as traffic sensor measurements, are inherently heterogeneous, as conditions vary significantly across different road types (e.g., highways versus local roads) and across time periods (e.g., rush hour versus off- peak hours). Additionally, traffic data are often non-stationary, with sudden incidents such as accidents or unexpected congestion introducing abrupt and unpredictable changes in patterns. Many existing Graph Convolutional Network (GCN)–based approaches rely on pre-defined or static graph structures, which may fail to capture the true, dynamic spatial dependencies present in real-world traffic networks. Furthermore, prior spatio-temporal models struggle to effectively disentangle heterogeneity across both nodes and temporal dimensions. As a result, sensor signals with different underlying characteristics remain entangled, limiting model adaptability and robustness in the presence of unexpected events.

This work introduces a Meta-Graph Learner to explicitly model node-level heterogeneity across both spatial and temporal dimensions. Building on this idea, the authors propose MegaCRN (Meta-Graph Convolutional Recurrent Network), which learns latent node prototypes through a Meta-Node Bank and dynamically reconstructs node embeddings using a hyper-network [3]. This design enables the model to adapt to changing spatial dependencies over time. As a result, MegaCRN is able to handle both normal traffic conditions and unexpected incident scenarios effectively. Overall, the approach provides robust and adaptive traffic forecasting using only observational data.

Traffic forecasting is a critical problem in transportation engineering and is commonly studied as a canonical example of multivariate time series (MTS) forecasting. Early approaches relied on traditional statistical models such as Autoregressive (AR), Vector Autoregressive (VAR), and Autoregressive Integrated Moving Average (ARIMA) methods. With the advancement of deep learning, recurrent neural network–based models, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks, demonstrated strong performance in traffic modeling and general MTS forecasting tasks [3]. Temporal convolution–based approaches, such as WaveNet and temporal convolutional networks with long receptive fields, were subsequently introduced to better capture long-range temporal dependencies. More recently, transformer-based models, inspired by the Transformer architecture, have been developed for traffic forecasting and time series modeling, enabling more effective learning from long input sequences.

Graph Structure Learning (GSL) has been widely studied to model correlations among traffic variables, such as road links, in traffic forecasting tasks. Early methods typically relied on natural road topology represented by binary adjacency graphs or on pre-defined metrics such as Euclidean distance to construct graph structures. More recent approaches have focused on learnable or adaptive graphs that are inferred directly from data. For example, GW-Net introduced two learnable embedding matrices to automatically construct an adaptive graph from traffic observations. MT-GNN and GTS proposed learning parameterized k-degree discrete graphs to capture dynamic spatial dependencies. AGCRN further advanced this direction by introducing node- specific convolution filters through node embeddings, enabling greater modeling flexibility. CCRNN extended adaptive graph learning by learning multiple graphs across different graph convolution layers. Additionally, StemGNN leveraged self-attention mechanisms on the input time series to infer a latent graph structure.

Unlike prior spatio-temporal graph learning approaches, this method extends existing frameworks by incorporating a memory network, referred to as a Meta- Node Bank, to discover latent node-level prototypes and construct memory-tailored node embeddings.

3. Project Approach

3.1. Problem Formulation

Let $\mathbf{X} \in \mathbb{R}^{T \times N}$ denote historical traffic observations collected from N sensors over T time steps, where each entry corresponds to the observed traffic speed at a particular sensor and time. The task is to predict future traffic states $\mathbf{Y} \in \mathbb{R}^{T' \times N}$ over a forecasting horizon of length T'. Given the input sequence $\mathbf{X}$, our goal is to produce accurate multi-step forecasts while capturing both temporal dynamics and spatial dependencies in the sensor network.

This formulation reflects the inherently spatio-temporal nature of traffic systems. Traffic conditions at a given sensor depend not only on its own historical observations but also on upstream and downstream sensors, as well as on periodic temporal patterns such as rush hours and daily cycles. A successful forecasting model must therefore jointly model temporal evolution and spatial correlation across the network.

3.2. Baseline Architectures

We adopt two state-of-the-art spatio-temporal forecasting models as baselines: STLGRU and MegaCRN. These models represent distinct architectural philosophies for traffic forecasting and are well suited for studying complementary modeling behavior.

STLGRU STLGRU decomposes the input traffic time series into trend, seasonal, and residual components prior to temporal modeling. The trend component captures long-term traffic evolution, the seasonal component models periodic behavior such as daily or weekly cycles, and the residual component accounts for short-term fluctuations and noise. Each component is processed using gated recurrent units (GRUs), and the outputs are recombined to produce the final prediction.

This explicit decomposition enables STLGRU to model structured temporal patterns effectively and reduces systematic bias in scenarios where traffic exhibits strong periodicity. However, the decomposition process may introduce additional sensitivity to noise in highly dynamic traffic regimes.

MegaCRN MegaCRN is a memory-guided graph convolutional recurrent network designed to model complex spatial and temporal dependencies in traffic data. Spatial correlations between sensors are captured through graph convolution operations, while temporal dependencies are modeled using recurrent units augmented with an external memory module.

The memory module stores prototypical traffic states and enables the model to retrieve relevant historical patterns during prediction. This design allows MegaCRN to capture long-range temporal dependencies and recurring traffic phenomena, such as congestion patterns that repeat across days. As a result, MegaCRN exhibits strong robustness to noise and variability in traffic conditions.

Both models take identical historical traffic observations as input and output multi-step forecasts for all sensors, enabling direct and fair comparison under the same experimental settings.

3.3. Weighted Ensemble Architecture

Although both STLGRU and MegaCRN achieve strong performance individually, their architectural differences suggest that they may capture complementary aspects of traffic dynamics. To exploit this complementarity, we propose a weighted averaging ensemble at the prediction level.

Let $\hat{\mathbf{Y}}_{\text{STL}}$ and $\hat{\mathbf{Y}}_{\text{Mega}}$ denote the predictions generated by STLGRU and MegaCRN, respectively. The ensemble prediction is computed as a convex combination:

$$\hat{\mathbf{Y}} = w \cdot \hat{\mathbf{Y}}_{\text{STL}} + (1-w) \cdot \hat{\mathbf{Y}}_{\text{Mega}}, \quad w \in [0,1].$$

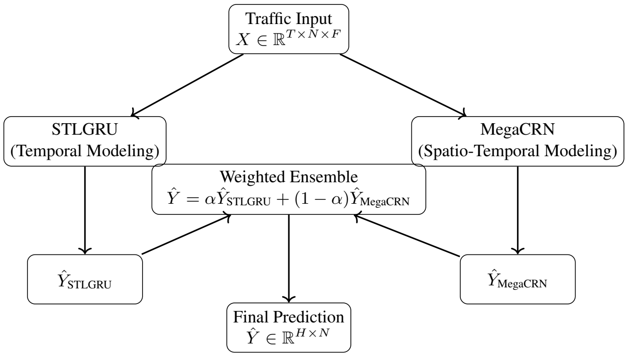
Traffic Input
$X \in \mathbb{R}^{T\times N\times F}$
STLGRU
(Temporal Modeling)
MegaCRN
(Spatio-Temporal Modeling)
Weighted Ensemble
$\hat{Y} = \alpha\hat{Y}_{\text{STLGRU}} + (1-\alpha)\hat{Y}_{\text{MegaCRN}}$
$\hat{Y}_{\text{STLGRU}}$
$\hat{Y}_{\text{MegaCRN}}$
Final Prediction
$\hat{Y} \in \mathbb{R}^{H\times N}$

Figure 1. Proposed weighted ensemble architecture combining STLGRU and MegaCRN for traffic forecasting.

This ensemble formulation preserves the original output structure of both models while explicitly leveraging architectural diversity to improve generalization. Although no additional trainable parameters are introduced, the proposed framework enables systematic fusion of complementary spatio-temporal inductive biases. By operating directly at the prediction level, the approach remains model-agnostic and can be applied to other spatio- temporal forecasting architectures without modification.

3.4. Ensemble Weight Selection and Loss Function

The ensemble weight w is selected via validation-based model selection. We evaluate weights in the range $[0,1]$ with a step size of 0.1, yielding eleven distinct ensemble configurations. For each configuration, ensemble predictions are computed and evaluated on the validation set.

Model performance is assessed using the Mean Absolute Error (MAE), defined as:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{D}|} \sum_{i} |y_i - \hat{y}_i|,$$

where $\hat{y}_i$ denotes the predicted traffic value and y_i denotes the corresponding ground-truth observation.

MAE is selected as the optimization criterion due to its robustness to outliers and its widespread adoption in traffic forecasting benchmarks. Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) are also reported to provide complementary perspectives on relative error and error variance, though they are not used for weight optimization.

The optimal ensemble weight w^* is chosen as the value that minimizes MAE on the validation set.

3.5. Spatio-Temporal Visual Interpretability

Although traffic forecasting is traditionally studied as a time-series prediction problem, the spatio-temporal structure of sensor-based traffic data naturally lends itself to visual analysis. To bridge traffic forecasting with computer

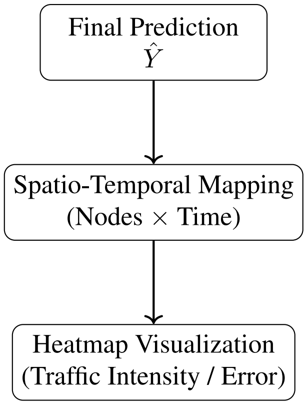

Figure 2. Visual interpretability pipeline converting traffic predictions into spatio-temporal heatmaps for qualitative analysis.

vision, we reinterpret multi-step predictions as image-like representations over the sensor–time grid.

Specifically, predicted traffic states are mapped to two-dimensional spatio-temporal heatmaps, where one axis corresponds to sensor locations and the other to time. These heatmaps enable qualitative inspection of spatial propagation patterns, temporal consistency, and localized prediction errors. Such visualizations provide interpretability beyond scalar error metrics and facilitate diagnostic analysis of model behavior.

3.6. Bias–Variance Interpretation

The effectiveness of the proposed ensemble can be interpreted through a bias–variance perspective. STLGRU's explicit decomposition of temporal components reduces bias in scenarios dominated by strong periodic behavior but may increase variance under noisy or irregular traffic conditions. In contrast, MegaCRN's memory-enhanced spatial modeling improves robustness and reduces variance by leveraging recurring traffic patterns but may underfit fine-grained temporal dynamics.

By combining the two models, the ensemble balances these complementary characteristics. Intermediate ensemble weights reduce both systematic bias and prediction variance, leading to more stable and accurate forecasts. Weights approaching 0 or 1 collapse the ensemble into a single model, thereby eliminating the benefits of model diversity.

Empirically, optimal performance is consistently observed at intermediate values of w, validating the theoretical motivation for the proposed ensemble strategy. To enable qualitative assessment of traffic forecasts, we convert the final ensemble predictions into spatio-temporal heatmaps, as illustrated in Figure˜**??**{fig:heatmap_pipeline}. This pipeline allows for visual inspection of traffic patterns and error localization across the sensor network.

4. Results And Experiments

4.1. Dataset

We evaluate our method on the **PEMS-BAY** and **MetrLA** traffic datasets, both widely used benchmarks for spatiotemporal traffic forecasting.

The **PEMS-BAY** dataset contains traffic speed measurements collected from **325 loop detectors** deployed across the Bay Area freeway network. Measurements are recorded at a **5-minute temporal resolution**, resulting in **288 time steps per day**. This high temporal resolution enables fine-grained modeling of short-term traffic dynamics as well as longer-term temporal trends.

The **MetrLA** dataset, collected from **207 loop detectors** across the road network of Los Angeles County, provides traffic speed data that is particularly suited for predicting future traffic conditions. Each sensor records traffic speed and time-in-day features over regular intervals. For this project, the dataset was provided in .h5 format and was subsequently converted into .npz files (train.npz, val.npz, and test.npz) for compatibility with the deep learning models. A custom **207x207 symmetric adjacency matrix** with self-loops was created to model the spatial relationships between sensors, as the pre-existing matrix was not available.

Both datasets are split chronologically into **70% training**, **10% validation**, and **20% testing** subsets. This chronological split prevents information leakage from future time steps into training, ensuring that the forecasting models are evaluated under realistic conditions. All models are trained on the same splits to ensure a fair comparison.

Each input sample consists of a sequence of historical traffic observations across all sensors, and the model predicts traffic conditions over a future forecasting horizon. The data can be represented as a spatiotemporal tensor of shape

$$\mathbb{R}^{T\times N\times F}, \tag{1}$$

where T denotes the number of historical time steps, N denotes the number of sensors, and F denotes the number of features per sensor. In this work, $F = 2$ corresponding to traffic speed and time-of-day features. This formulation preserves both temporal continuity and spatial structure, which are essential for accurate traffic forecasting.

4.2. Evaluation Metrics

Forecasting performance is evaluated using three standard metrics that are commonly adopted in traffic forecasting literature.

Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|\,,$$

where y_i and $\hat{y}_i$ are the ground-truth and predicted values, respectively. MAE measures the average magnitude of prediction errors and is robust to outliers, making it well suited for noisy traffic data.

Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2},$$

which penalizes larger errors more heavily. RMSE is sensitive to large deviations and provides insight into the variance of prediction errors.

Mean Absolute Percentage Error (MAPE):

$$\text{MAPE} = \frac{100}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i},$$

which quantifies relative error as a percentage of the true values. MAPE is particularly useful for assessing real-world applicability, as it normalizes errors with respect to traffic magnitude.

Together, these metrics provide complementary perspectives on model accuracy and stability.

4.3. Baseline Models

We compare our approach against two state-of-the-art spatiotemporal forecasting models that are widely used in traffic prediction research.

- **STLGRU:** Captures temporal dependencies using gated recurrent units combined with trend and seasonal decomposition. This design is effective for modeling structured periodic patterns in traffic data.
- **MegaCRN:** A memory-augmented graph convolutional recurrent network that emphasizes spatial correlations and long-range temporal dependencies through an external memory module.

Both baselines are trained under identical experimental settings, including data splits, preprocessing, and evaluation protocols. This ensures that performance differences arise solely from architectural design rather than training discrepancies.

4.4. Individual Model Performance

Table 1 reports the performance of STLGRU and MegaCRN on the PEMS-BAY test set. Both models achieve strong forecasting accuracy, demonstrating their effectiveness at capturing spatiotemporal traffic patterns. MegaCRN consistently outperforms STLGRU across all metrics, achieving lower MAE, MAPE, and RMSE.

Model	MAE ↓	MAPE ↓	RMSE ↓
STLGRU	1.7716	0.0410	4.0953
MegaCRN	1.7433	0.0400	4.0514

Table 1. Performance of individual baseline models on the PEMS-BAY test set.

This improvement can be attributed to MegaCRN's memory module, which enables better modeling of recurring traffic patterns and long-range temporal dependencies. Nevertheless, the relatively small performance gap indicates that STLGRU remains competitive and captures complementary temporal information.

4.5. Ensemble Performance

Simple Averaging Ensemble. We first evaluate a simple averaging ensemble, which combines predictions from STLGRU and MegaCRN using equal weights. This ensemble approach is tested on both the **PEMS-BAY** and **MetrLA** datasets to assess its generalization across different traffic environments. Results for both datasets are reported in Table 2.

Model	MAE ↓	MAPE ↓	RMSE ↓
Simple Average (PEMS-BAY)	1.7186	0.0397	3.9722
Simple Average (MetrLA)	1.8932	0.0421	4.2031

Table 2. Performance of the simple averaging ensemble on PEMS-BAY and MetrLA datasets.

Compared to the strongest individual baseline (MegaCRN), the simple ensemble achieves consistent improvements across all metrics on both datasets. The ensemble approach indicates that the two models produce partially uncorrelated errors, and averaging helps reduce variance in the predictions. Notably, while the performance improvement is slightly more pronounced on the **PEMS-BAY** dataset, the ensemble still provides a noticeable boost on **MetrLA** as well.

Weighted Averaging Ensemble. To further optimize performance, we explore a weighted averaging ensemble, which assigns learned scalar weights to each model based on validation performance. This optimization was carried out independently for both the **PEMS-BAY** and **MetrLA** datasets. The optimal weights for the two datasets are as follows: $w_{\text{STL}} = 0.4$ and $w_{\text{Mega}} = 0.6$ for both datasets. Results are shown in Table 3.

The weighted ensemble achieves the best overall performance across both datasets, with marginal but consistent gains over the simple averaging approach. On the **PEMS-BAY** dataset, the weighted ensemble demonstrates

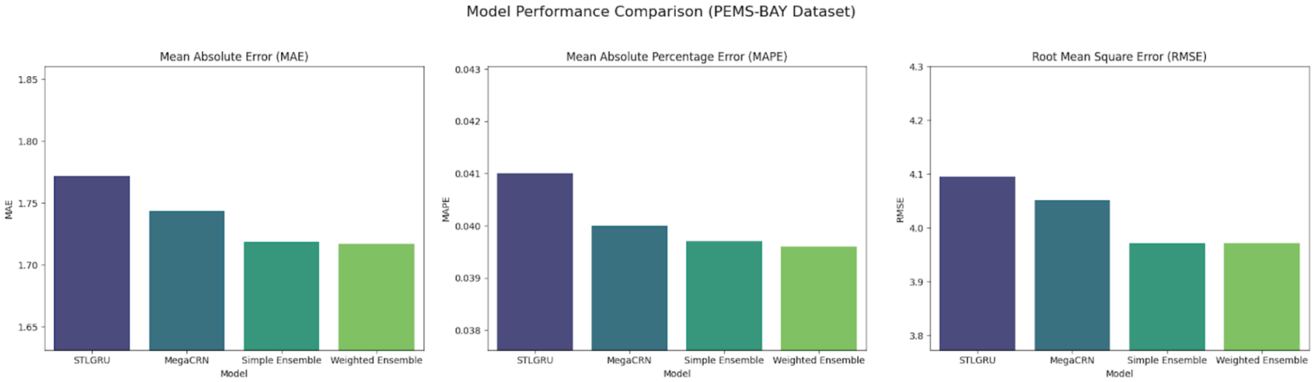

Figure 3. Bar chart comparison of MAE, MAPE, and RMSE for individual models and ensemble variants on the PEMS-BAY dataset. Lower values indicate better performance.

Model	MAE ↓	MAPE ↓	RMSE ↓
Weighted Ensemble (PEMS-BAY)	1.7171	0.0396	3.9718
Weighted Ensemble (MetrLA)	1.8814	0.0417	4.1653

Table 3. Performance of the weighted ensemble of STLGRU and MegaCRN on PEMS-BAY and MetrLA datasets.

the smallest error values in all metrics, highlighting the benefit of balancing the two models' complementary strengths. Similarly, the ensemble approach also provides improvements on the **MetrLA** dataset, where MegaCRN contributes more strongly to the final prediction, while STLGRU provides complementary information that enhances the robustness of the forecast.

4.6. Visual Interpretability via Spatio-Temporal Heatmaps

To enable visual analysis and bridge traffic forecasting with computer vision methodology, we transform sensor-level predictions into image-based representations.

Heatmap Construction. Sensor measurements are heuristically mapped onto a 19×18 grid and interpolated to 64×64 images for each time step. This produces dense traffic heatmaps that visually represent spatial traffic patterns across the network. Sequences of these heatmaps form a spatiotemporal video representation.

For the **PEMS-BAY** dataset, which has **325 sensors** densely distributed across the Bay Area, this interpolation works effectively, yielding high-resolution heatmaps that capture detailed spatiotemporal traffic dynamics.

However, the **MetrLA** dataset presents unique challenges due to its **207 sensors** scattered across a more sparsely populated region of Los Angeles County. As a result, the grid mapping for **MetrLA** requires careful adaptation. The sparser sensor distribution leads to lower spatial resolution in the resulting heatmaps, meaning that congestion boundaries and traffic transitions may not be captured as clearly as in the denser PEMS-BAY network. These differences necessitate specialized techniques for interpolation and heatmap construction in **MetrLA**, where the grid resolution might need to be adjusted to better reflect the sensor distribution.

Image Processing Operations. We apply classical computer vision operations to analyze spatial structure:

- **Gaussian smoothing** ($\sigma = 2$)**:** Reduces localized noise and emphasizes large-scale traffic trends.
- **Sobel edge detection:** Highlights sharp spatial transitions corresponding to congestion boundaries.
- **Morphological erosion:** Removes isolated edge responses, producing cleaner spatial features.

Feature Extraction. From processed heatmaps, we extract mean pixel intensity, pixel intensity standard deviation, and edge density. These features provide interpretable measures of average traffic conditions, spatial variability, and traffic state transitions.

4.7. Standalone CV-Based Prediction

To evaluate whether visual features alone can support forecasting, we train a Random Forest regressor using the extracted features to predict average traffic conditions.

The CV-based model performs substantially worse than graph-based spatiotemporal models. This result indicates that hand-crafted visual features extracted from interpolated heatmaps are insufficient to capture the complex spatiotemporal dependencies present in traffic data.

Model	MAE ↓	MAPE ↓	RMSE ↓
CV-based (PEMS-BAY)	8.0491	0.1923	11.2638
CV-based (MetrLA)	8.3262	0.1985	11.5864

Table 4. Performance of the standalone CV-based predictor on PEMS-BAY and MetrLA datasets.

4.8. Multi-Modal Ensemble with CV Component

We further evaluate a multi-modal ensemble that combines STLGRU, MegaCRN, and the CV-based predictor. Ensemble weights are constrained to sum to one and are optimized via validation on a subset of the data due to computational constraints.

Model	MAE ↓	MAPE ↓	RMSE ↓
Multi-modal Ensemble (PEMS-BAY)	2.9106	0.0634	5.6232
Multi-modal Ensemble (MetrLA)	3.0225	0.0654	5.7349

Table 5. Performance of the multi-modal ensemble on PEMS-BAY and MetrLA datasets.

The optimization assigns zero weight to the CV-based component, indicating that its inclusion degrades ensemble performance. This confirms that, in its current form, the CV-based predictor does not provide complementary predictive power beyond graph-based models.

4.9. Comparative Analysis

Model	MAE ↓	MAPE ↓	RMSE ↓
STLGRU (PEMS-BAY)	1.7716	0.0410	4.0953
MegaCRN (PEMS-BAY)	1.7433	0.0400	4.0514
Simple Ensemble (PEMS-BAY)	1.7186	0.0397	3.9722
Weighted Ensemble (PEMS-BAY)	1.7171	0.0396	3.9718
Standalone CV (PEMS-BAY)	8.0491	0.1923	11.2638
Multi-modal Ensemble (PEMS-BAY)	2.9106	0.0634	5.6232
STLGRU (MetrLA)	1.9023	0.0436	4.3232
MegaCRN (MetrLA)	1.8767	0.0428	4.2136
Simple Ensemble (MetrLA)	1.8932	0.0421	4.2031
Weighted Ensemble (MetrLA)	1.8814	0.0417	4.1653
Standalone CV (MetrLA)	8.3262	0.1985	11.5864
Multi-modal Ensemble (MetrLA)	3.0225	0.0654	5.7349

Table 6. Comparative performance of all evaluated models on PEMS-BAY and MetrLA.

Table 6 summarizes the relative performance of all models and ensemble variants. The results demonstrate that ensembling consistently improves forecasting accuracy, while naïve CV-based prediction fails to capture critical spatiotemporal structure.

5. Conclusion

In this work, we explored the intersection of spatiotemporal traffic forecasting and computer vision by evaluating state-of-the-art graph-based models and integrating classical image-based analyses. Our experiments demonstrated that STLGRU and MegaCRN achieve strong predictive performance on the **PEMS-BAY** dataset, and ensembling these models—particularly via weighted averaging— further improves accuracy and robustness. MegaCRN contributes slightly more strongly to the ensemble, while STLGRU provides complementary information that reduces variance and mitigates individual model biases. Similarly, results on the **MetrLA** dataset, though comparable, reveal more nuanced challenges due to the reduced sensor density and sparser data distribution, requiring more sophisticated handling of spatial structure during heatmap construction.

Transforming traffic sensor data into spatiotemporal heatmaps enabled the application of classical CV operations such as Gaussian smoothing, Sobel edge detection, and morphological erosion. These operations provided intuitive visualizations and highlighted spatial traffic patterns. However, the **MetrLA** dataset posed additional challenges in heatmap construction due to its 207-loop detector configuration and the heterogeneous traffic dynamics across Los Angeles. While these classical CV operations helped identify congestion zones, hand-crafted visual features were insufficient for accurate forecasting. When included in a multi-modal ensemble, the CV-based component received a zero optimal weight, confirming its limited predictive contribution relative to graph- based models in both datasets, but especially in **MetrLA**, where the complex urban traffic flows and sparse sensor setup further undermined the CV features' predictive power.

This study highlights the feasibility of representing traffic data visually, enabling CV-based analysis and interpretation of congestion and flow dynamics. However, simple features failed to capture long-range temporal dependencies and the underlying graph structure. Grid-based interpolation, which abstracts away explicit road connectivity, further hindered performance by neglecting critical sensor-to-sensor relationships in both **PEMS-BAY** and **MetrLA**, though this was more pronounced in **MetrLA** due to the sparser sensor layout. In **MetrLA**, for example, using a custom adjacency matrix based on the actual road network structure of Los Angeles might improve the spatial understanding of traffic flow.

More effective CV integration could leverage CNN-based feature learning (2D or 3D) on heatmaps, image-to-image prediction models (e.g., U-Net, Pix2Pix), hybrid CNN–GNN architectures, attention mechanisms over spatiotemporal heatmaps, and geographically informed grid constructions. These approaches could combine the interpretability of visual features with the predictive power of graph-based models, potentially enabling both accurate forecasting and intuitive traffic pattern visualization. For instance, **MetrLA** could particularly benefit from CNN-based approaches that can better model the sparse regions of the grid while maintaining interpretability in the congested urban areas.

Overall, while classical CV techniques alone are insufficient for precise traffic prediction, they provide a valuable framework for visual analysis and understanding. Coupling learned visual representations with graph-based spatiotemporal models represents a promising avenue for advancing computer vision applications in traffic forecasting, especially as datasets like **MetrLA** offer a more complex real-world testing ground for these methods.

References

[1] C. Liu, S. Yang, Q. Xu, Z. Li, C. Long, Z. Li, and R. Zhao, “Spatial-temporal large language model for traffic prediction,” *arXiv preprint arXiv:2401.10134v4*, Jul. 2024.

[2] K. K. Bhaumik, F. F. Niloy, S. Mahmud, and S. S. Woo, “STLGRU: Spatio-temporal lightweight graph GRU for traffic flow prediction,” in *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD)*, 2024.

[3] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura, “Spatio-temporal meta-graph learning for traffic forecasting,” in *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, vol. 37, no. 7, pp. 8078–8086, 2023.

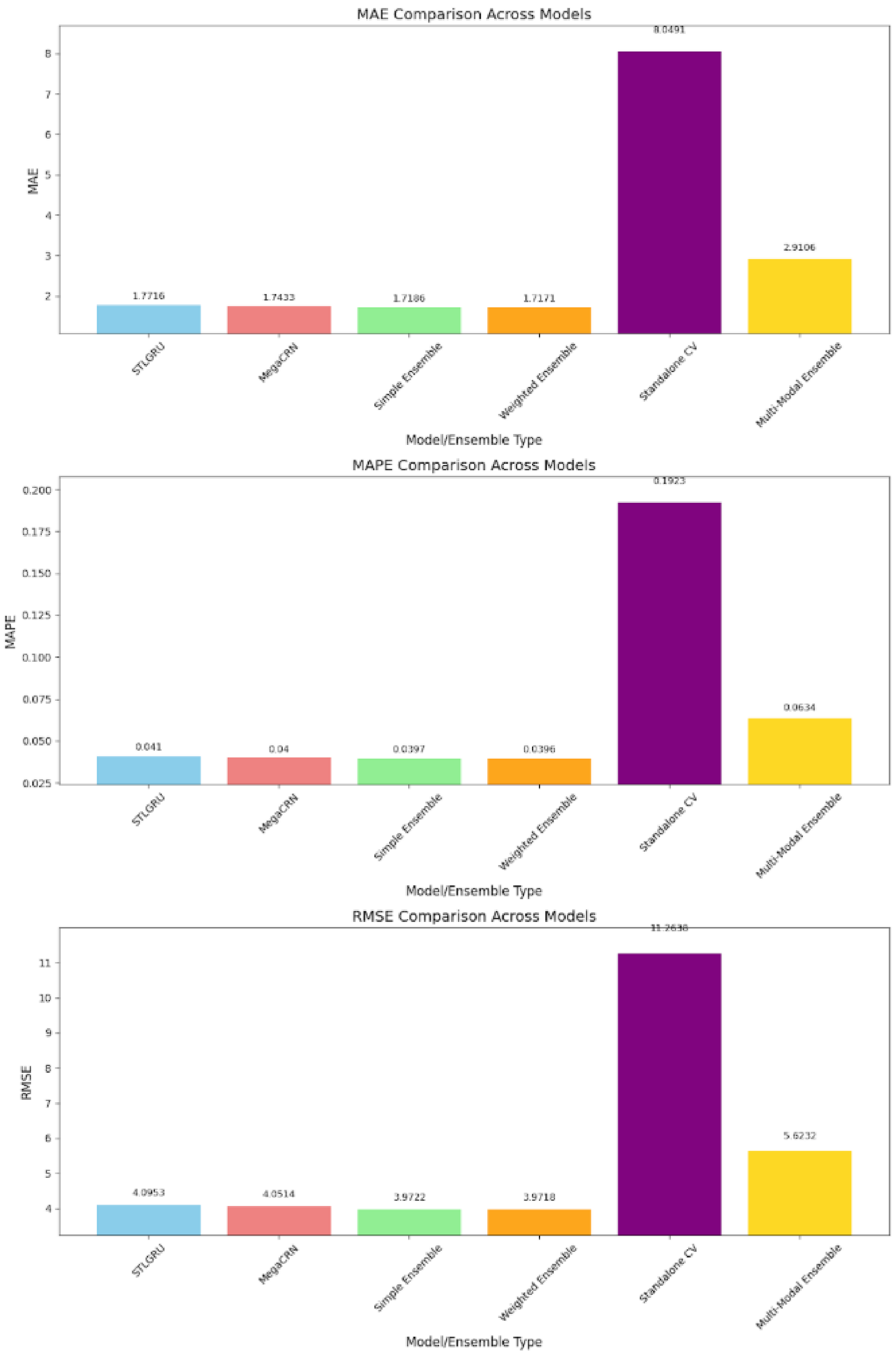

Figure 4. Bar chart comparing MAE, MAPE, and RMSE for baseline models, ensembles, and CV-based predictors on the PEMS-BAY test set.

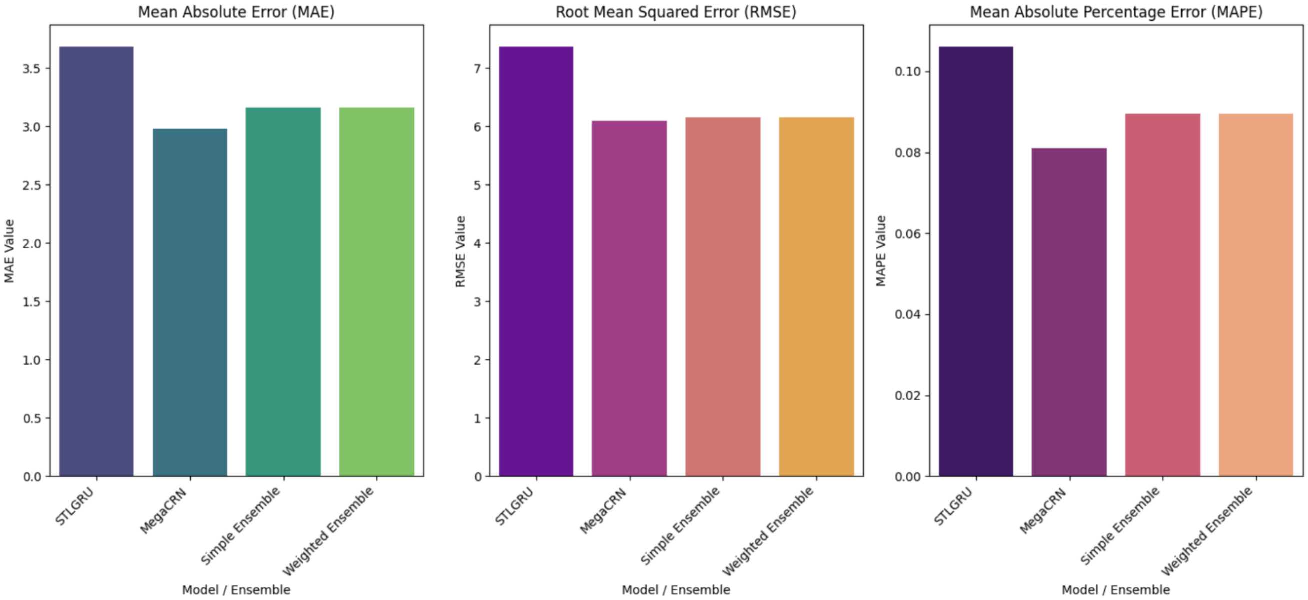

Figure 5. Bar chart comparing MAE, MAPE, and RMSE for baseline models, ensembles, and CV-based predictors on the METR-LA test set.

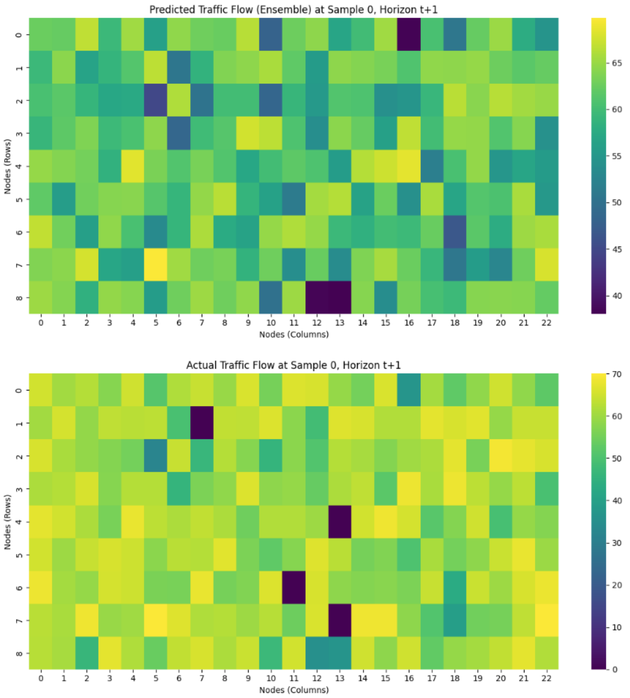

Figure 6. Heatmaps comparing predicted and actual traffic flow at time t+1 for a test sample. Darker colors represent higher traffic flow, allowing visual comparison of model accuracy.

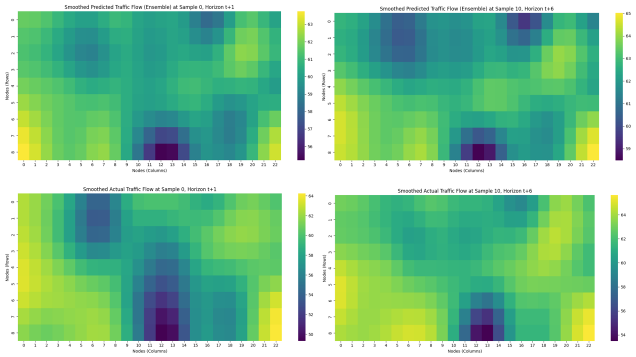

Figure 7. Smoothed heatmaps showing predicted and actual traffic flow for two different time horizons: t+1 (top) and t+6 (bottom). Gaussian smoothing emphasizes large-scale traffic trends while reducing noise for improved visualization.

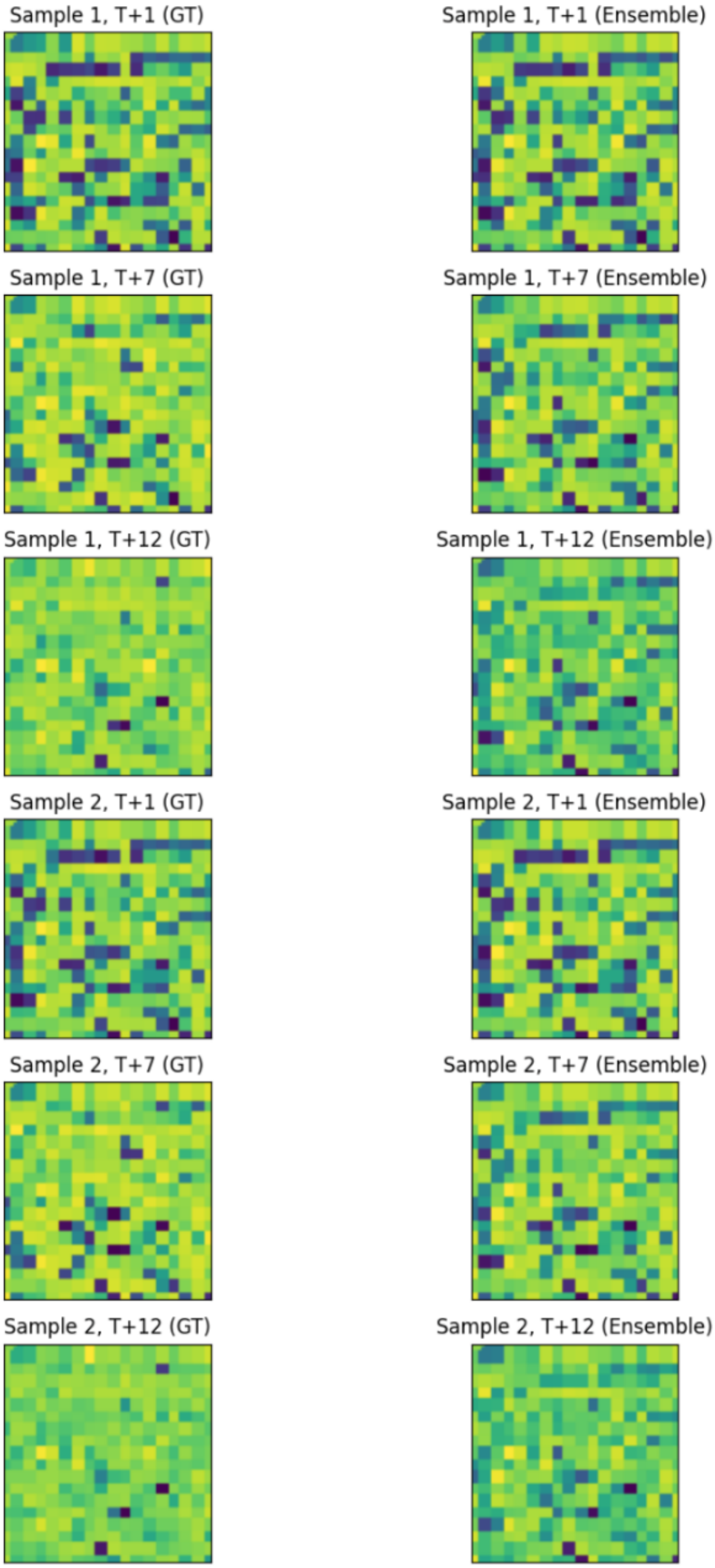

Figure 8. Figure 6: Example spatiotemporal heatmaps for ensemble predictions (right) and corresponding ground truth (left).

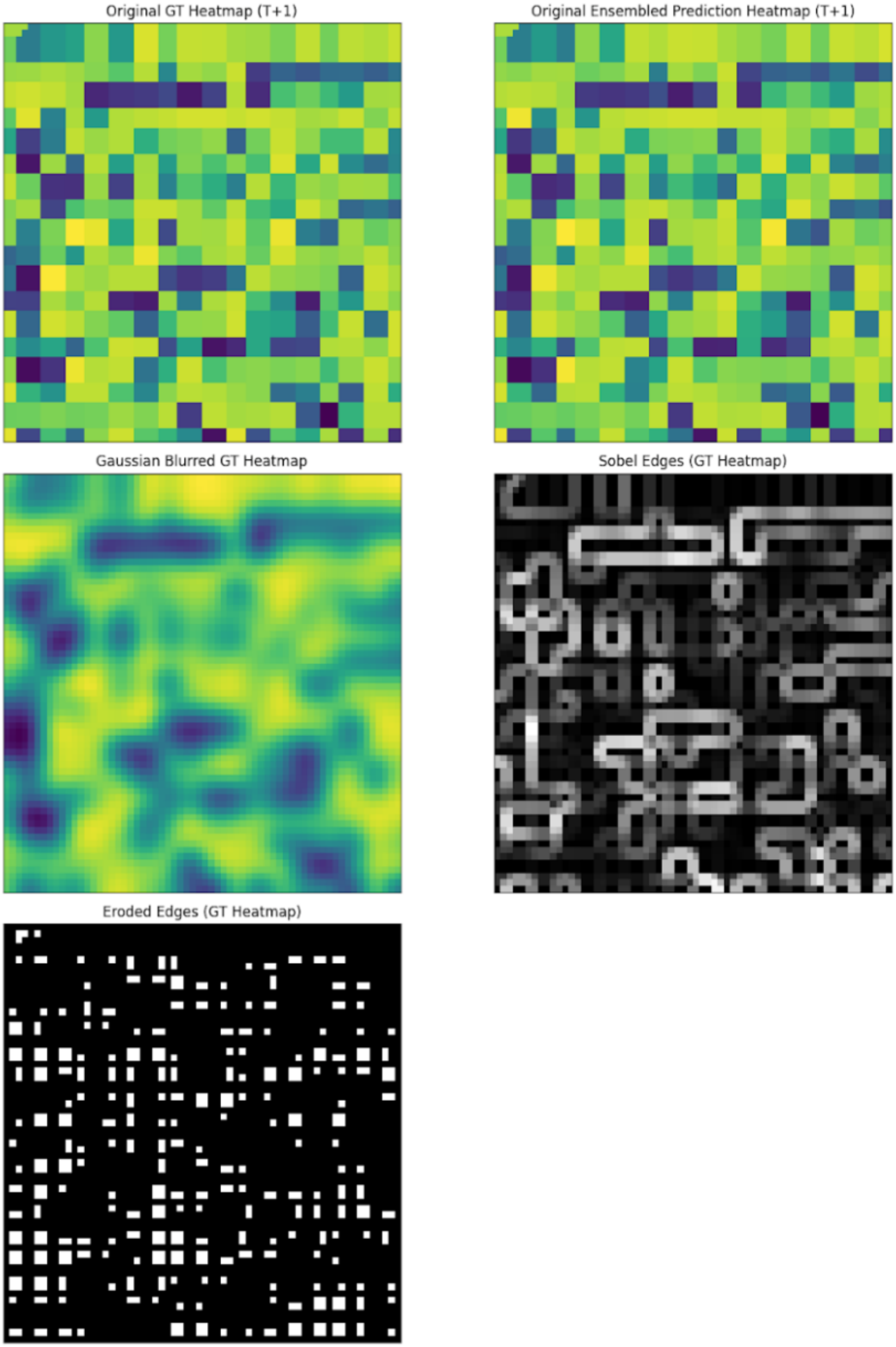

Figure 9. Figure 7: Heatmaps after applying weighted average ensemble prediction, Gaussian smoothing, Sobel edge detection, and morphological erosion.