Patched Forecasting with Gumbel-Based Selector for Sparse Mobile Crowd Sensing

Victor Li
Computer Science Department
Emory University
victor.li@emory.edu

Carson Lam
Computer Science Department
Emory University
carson.lam@emory.edu

Ting Li
Computer Science Department
Emory University
ting.li@emory.edu

***Abstract*—Mobile Crowd Sensing (MCS) is a promising paradigm for collecting large-scale data by using distributed sensors on smartphones and wearables. However, building comprehensive spatiotemporal fields of measurements with continuous sensing at all locations and time cycles is often impractical and costly. To address this, we leverage our proposed Learned Gumbel-Based Active Sparse MCS framework, utilizing an Encoder-Decoder Time Series Transformer that reconstructs the full spatiotemporal field for a given sensing cycle using the data collected by a small subset of available sensors. The framework incorporates a selector that actively employs the Gumbel distribution to dynamically pick the most effective sensors for the next sensing cycle. Rather than applying a single selector layer across all sensors, we introduce a patched forecasting approach to account for temporal patterns inherent in sensing cycles. Each patch forecasts its own subsequence of the time series while simultaneously selecting a subset of sensors to reuse in the next sensing cycle.**

***Index Terms*—Mobile Crowd Sensing, Active Learning, Time-Series Reconstruction, Deep Learning**

I. Introduction

Mobile Crowd Sensing (MCS) has transformed large-scale environmental and urban monitoring by using distributed sensors embedded in smartphones and wearables. Yet practical constraints—such as cost and bandwidth—make continuous monitoring of every sensor infeasible. By gathering data from a small subset of available sensors, Sparse MCS has been developed to address this challenge by predicting the remaining spatiotemporal field. Furthermore, Active Sparse MCS (AS-MCS) frameworks, such as our proposed Learned Gumbel-Based framework leveraging an Encoder-Decoder Time Series Transformer [4], can dynamically select sensors, ensuring that the most optimal sensors are chosen for reconstruction. To do so, we implement a learned layer that samples Gumbel noise from trainable weights, capturing historical importance, and combines it with encoder embeddings to provide global context.

In this work, we propose a modified approach that divides the sensing cycle into multiple patches, each with its own selector layer. Keeping the same Encoder-Decoder design, this patch-based approach allows the model to capture temporal patterns within the cycle. For instance, dividing a 24-hour cycle into morning, afternoon, and night enables the model to determine the most effective sensors for each part of the day.

II. Methodology

Our baseline Learned Gumbel-Based AS-MCS framework consists of three main components: an encoder that generates contextualized embeddings, a selector layer that performs active sensor selection, and a decoder that carries out the final inference. Our proposed model extends this baseline by implementing multiple selector layers and splitting up sequences into patches for encoding and decoding.

A. Data Preparation

We begin with complete historical data from N sensors and separate the static positional coordinates from the length-l, m-variable multivariate time series into two datasets $\Sigma \in \Re^{N\times 2}$ and $\mathbf{X} \in \Re^{N\times l\times m}$, respectively. To simulate the intervals at which sensors transmit their readings, the data is partitioned into sensing cycles of fixed length σ. A subset $\mathbf{B}$ of sensors such that $\text{len}(\mathbf{B}) \ll N$ is also created to simulate practicality constraints. Therefore, for each cycle w, the model receives $\mathbf{X}_{w,s}$ and Σ_s for $s \in \mathbf{B}$ plus the static coordinates Σ_r for missing locations $r \notin \mathbf{B}$.

B. Learned Gumbel-Based Active Sensing Framework

For our three-component Learned Gumbel-Based AS-MCS framework, we first pass sensed data $\mathbf{X}_{w,s}$ at coordinates Σ_s through the encoder to create contextualized embeddings for each sensor. Our encoder follows the original Transformer [3] architecture, but uses batch normalization instead of layer normalization introduced in the Multivariate Times Series Transformer [4]. In addition, we pre-process static and dynamic variables separately through an MLP and a linear layer, respectively, which is then added with fully learned positional encoding as opposed to classic sinusoidal encodings.

The selector layer then uses these embeddings to select k sensors for the next cycle. We introduce the parameter $k < \text{len}(\mathbf{B})$ to ensure selection diversity by only allowing the model's selector layer to choose a fraction of the sensors and fill the remaining $\text{len}(\mathbf{B}) - k$ indices at random. Our Learned Gumbel-Based Selector chooses these k sensors using a matrix of learnable weights $\alpha \in \Re^{N\times k}$, where each weight α_{ij} represents the probability of choosing the i-th sensor for

the j-th selected index. Gumbel noise is then sampled and added to α, leveraging its intrinsic connection to categorical sampling, which makes it particularly effective for top-k sensor selection tasks [2]. To enable gradient flow to these weights, α also serves as an embedding mask. The masked embeddings represent information solely from the k sensors, which are then used to reconstruct the $\text{len}(\mathbf{B}) - k$ other sensor readings known in the cycle.

Lastly, our decoder uses the encoder embeddings to predict the target variable at missing locations Σ_r for $r \notin \mathbf{B}$. Leveraging cross attention, our decoder pre-processes Σ_r using the same MLP and linear layer from the encoder to use as queries that attend over the encoder embeddings, acting as keys and values to generate predictions. The final layer produces a matrix, where each vector represents a specific sensor and timestamp. These vectors are subsequently projected into single interpolated values and concatenated across the full sequence length.

C. Patch-Based Approach

In our patched implementation, we introduce a parameter ρ and pass $X_{w,s}$ through a convolutional layer with a kernel size of σ/ρ to create ρ patches. Each patch is processed independently using the baseline Learned Gumbel-Based AS-MCS framework. The same encoder and decoder are used across patches, but each patch is assigned a dedicated selector layer responsible for selecting k/ρ sensors. Finally, each patch's selected sensor sets and their corresponding decoded outputs are concatenated together.

III. Results

A. Data

In our study, we utilize the Urban Air [5] and SensorScope's St-Bernard [1] datasets. The Urban Air dataset includes measurements from 437 weather stations across China, each recording hourly air quality data over the span of one year. On the other hand, the St-Bernard dataset consists of 31 temperature and humidity sensors located within a small region, providing much denser readings at two-minute intervals over two months. For both datasets, we chose a sequence length of 24 and created samples from long series using a sliding window. Additionally, we excluded sensors that are completely inactive during a cycle and linearly interpolated partially missing sensor readings.

B. Experimentation

In our experiments, we partitioned data chronologically, training on earlier time points and evaluating on subsequent recent observations. In Figure 1, we report preliminary mean squared error (MSE) losses for both models trained on both datasets with learning rates of 1e-3 and 3e-4. The patching approach yields marginal gains under *MSE@3e-4* on the U-Air dataset and *MSE@1e-3* on the St-Bernard dataset. However, these improvements are offset by substantial drops relative to the baseline across the remaining metrics, indicating that the method does not deliver consistently better generalization.

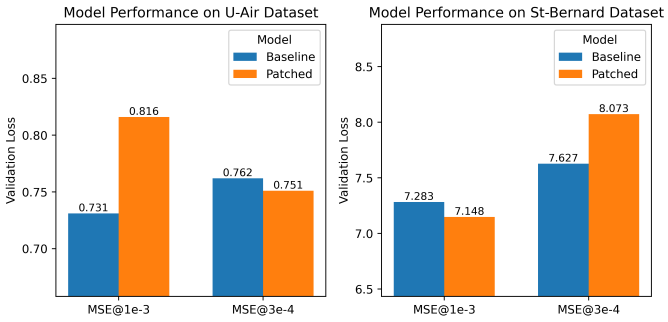

Fig. 1. Comparison of baseline and patched models across the U-Air and St-Bernard datasets trained with MSE under a learning rate of *1e-3* and *3e-4*.

We believe this performance drop is caused by the independent processing of each patch. By partitioning the sequence, the model forfeits the ability to leverage self- and cross-attention mechanisms across the entire sensed data, restricting contextualization within individual patches. This suggests that intra-cycle temporal patterns, while informative at times, are insufficient to compensate for the loss of cross-patch contextualization.

C. Future work

Our findings suggest that preserving cross-patch contextualization is key to realizing the benefits of multiple sensor selectors. Switching the Transformer architecture in our AS-MCS framework to alternative multivariate time series forecasting models, such as MultiPatchFormer, could be explored. MultiPatchFormer is well-suited for modeling cross-patch interactions through inter-patch attention, and its state-of-the-art performance could be further enhanced with our active sensing methodology. Finally, lightweight mechanisms such as cross-patch communication layers or Cross-Variate Patch Embeddings may help restore global context and improve stability.

References

[1] Guillermo Barrenetxea. *Sensorscope Data*. Zenodo. Apr. 2019. DOI: 10.5281/zenodo.2654726. URL: https://doi.org/10.5281/zenodo.2654726.

[2] T. Strypsteen and A. Bertrand. "End-to-End Learnable EEG Channel Selection for Deep Neural Networks with Gumbel-Softmax". In: *Journal of Neural Engineering* 18.4 (Aug. 2021), 0460a9. DOI: 10.1088/1741-2552/ac115d.

[3] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[4] George Zerveas et al. *A Transformer-based Framework for Multivariate Time Series Representation Learning*. 2020. arXiv: 2010.02803 [cs.LG]. URL: https://arxiv.org/abs/2010.02803.

[5] Yu Zheng et al. "Forecasting Fine-Grained Air Quality Based on Big Data". In: *Proceedings of the 21st SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2015)*. 2015.