Generative AI for Data Challenges in Electric Power Engineering

Yi Hu

Assistant Professor

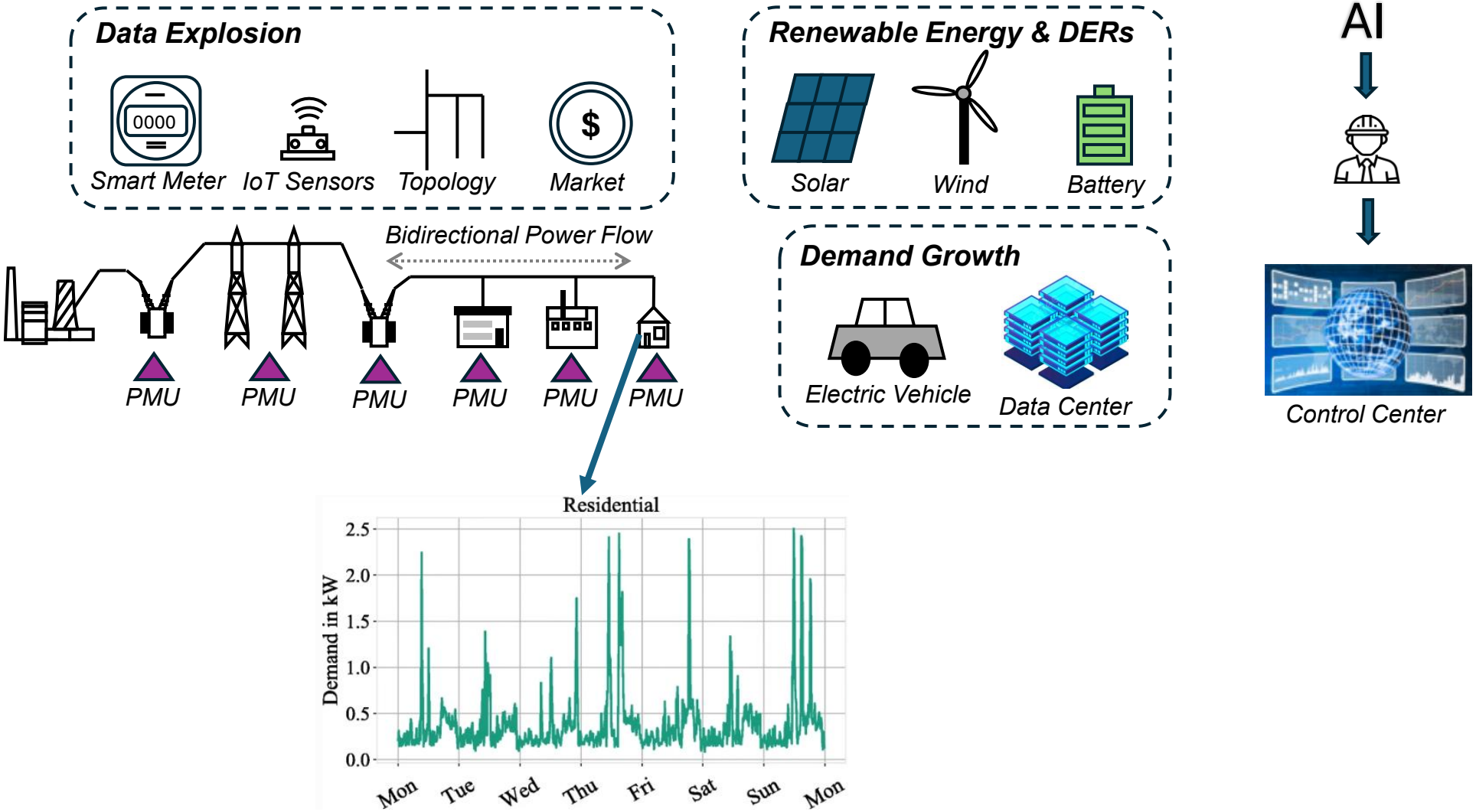
Electrical and Computer Engineering

Michigan Technological University

Houghton, MI

Data in Modern Power System

Modern Power System



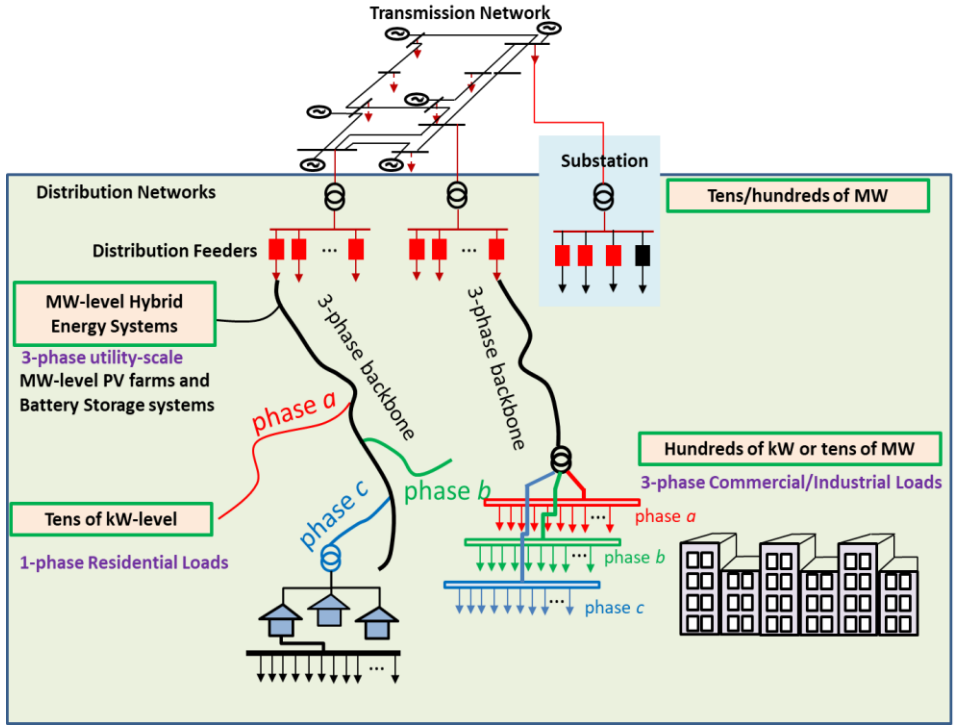
Time Series Data in Modern Power System

1-Minute Sub-metered data
- End use consumptions of appliances
- Not usually available
- Enabling technologies: IoT sensors

15-minute Smart Meter Data
- Average kWH, kVar, Voltage
- Sensitive information

Hourly
- Temperature, irradiance
- Average kWH

Super resolution

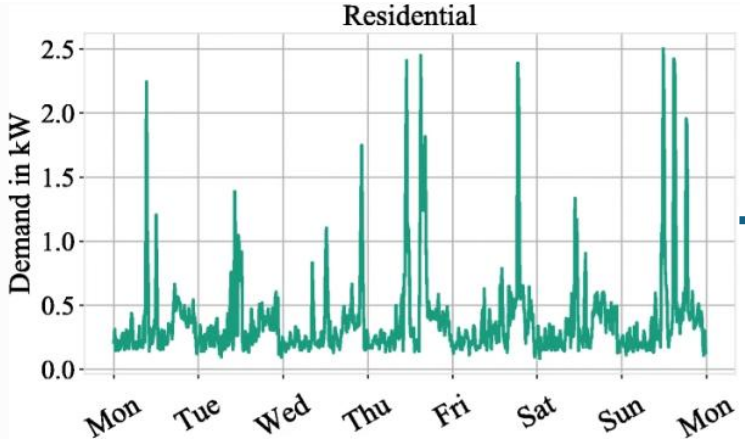


Daily
- Peak hour
- DR events

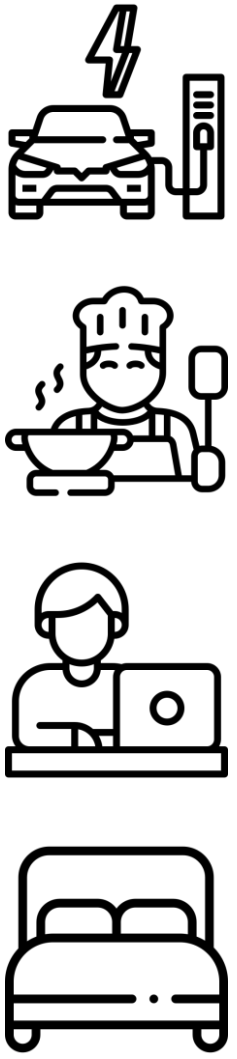
Monthly
- Utility billing information
- Peak day peak hour

Time

Data Privacy Issue

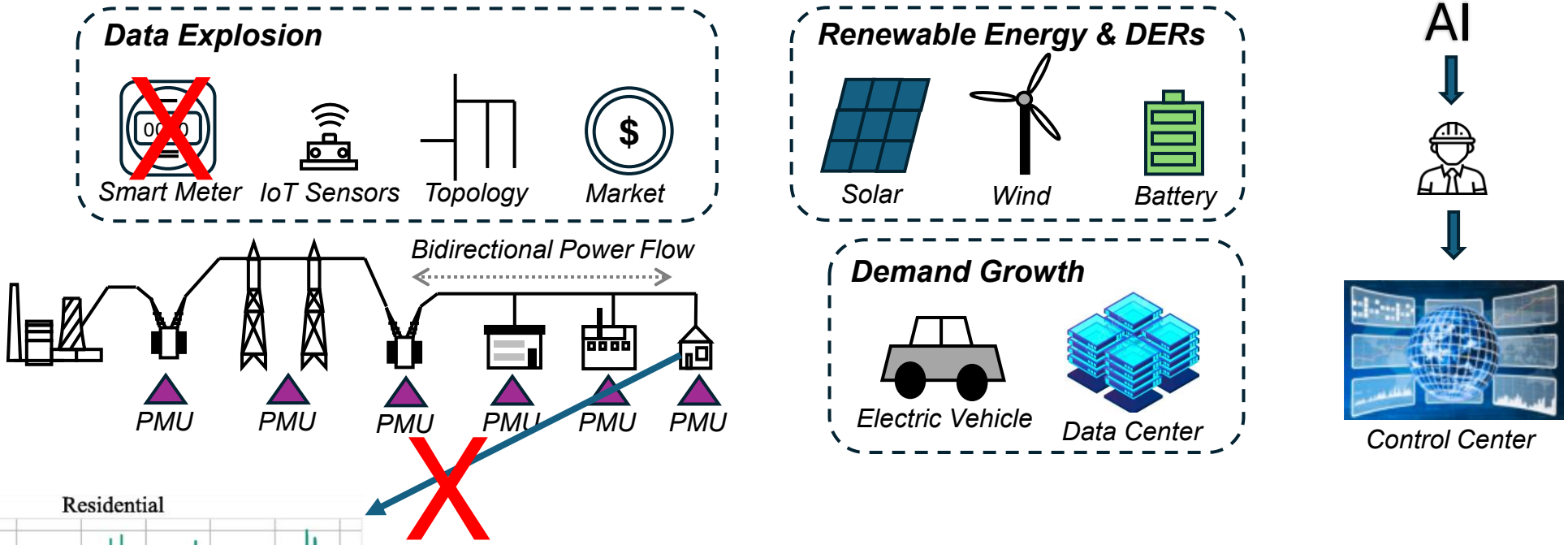


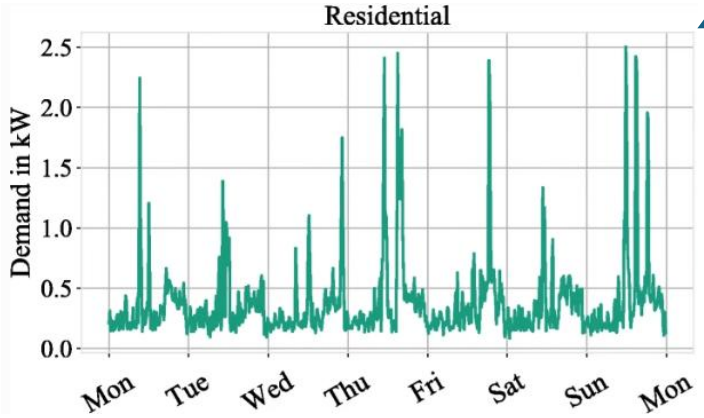
Daily activities



Bad Data in Power Engineering

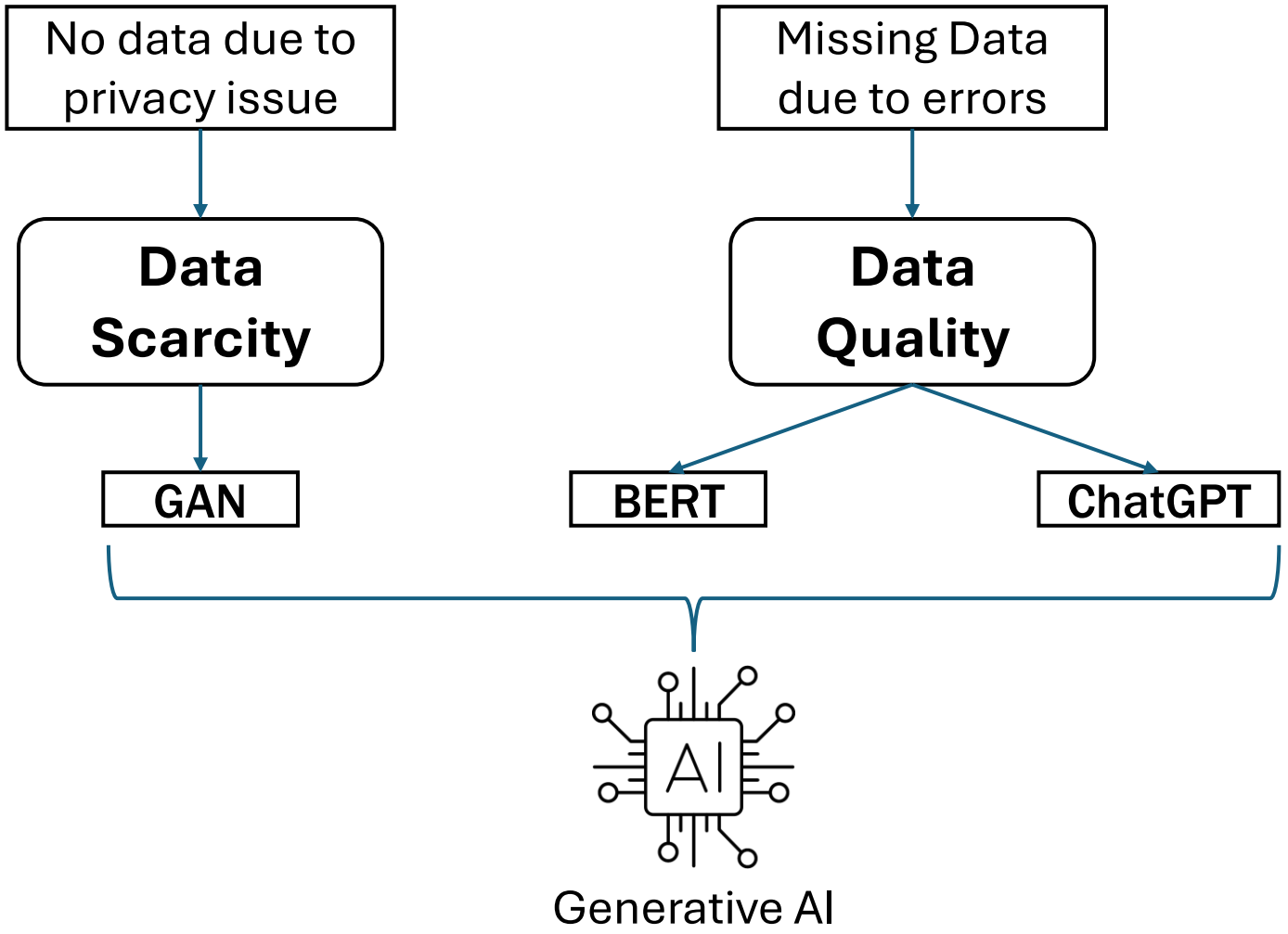
Modern Power System



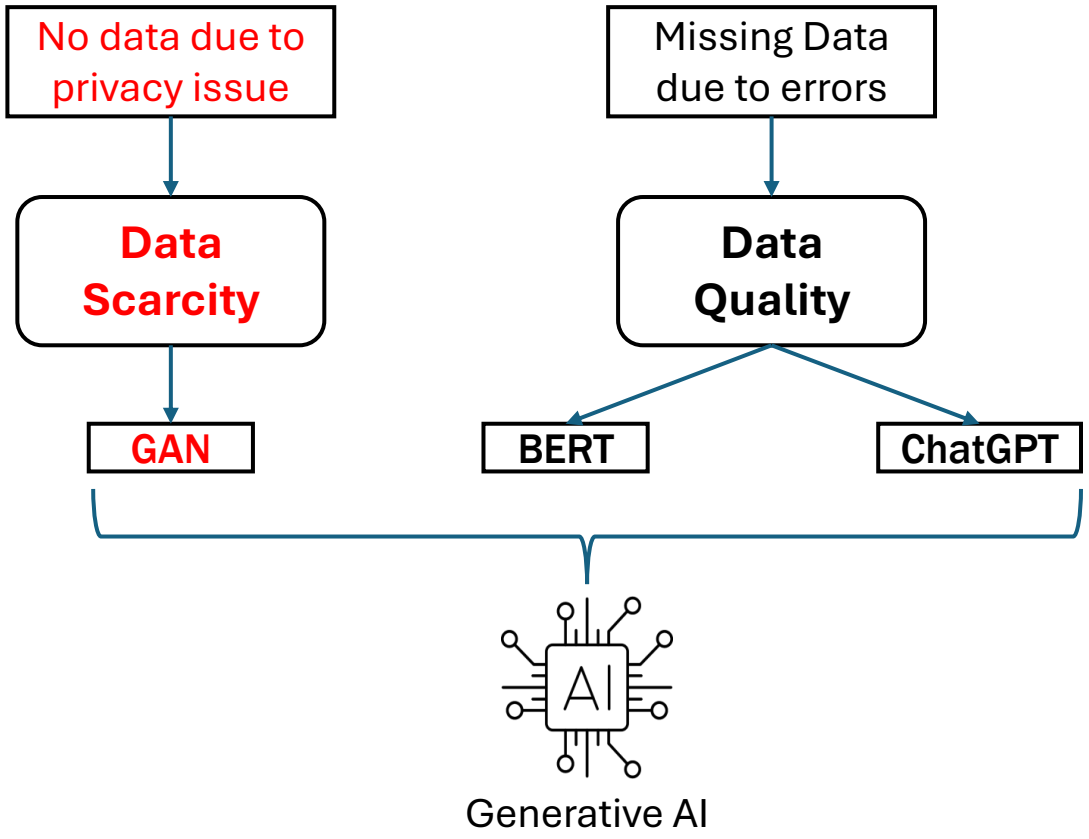


Equipment error or communication error will result in bad data.

Data Challenges in Power Engineering



Synthetic Data Generation



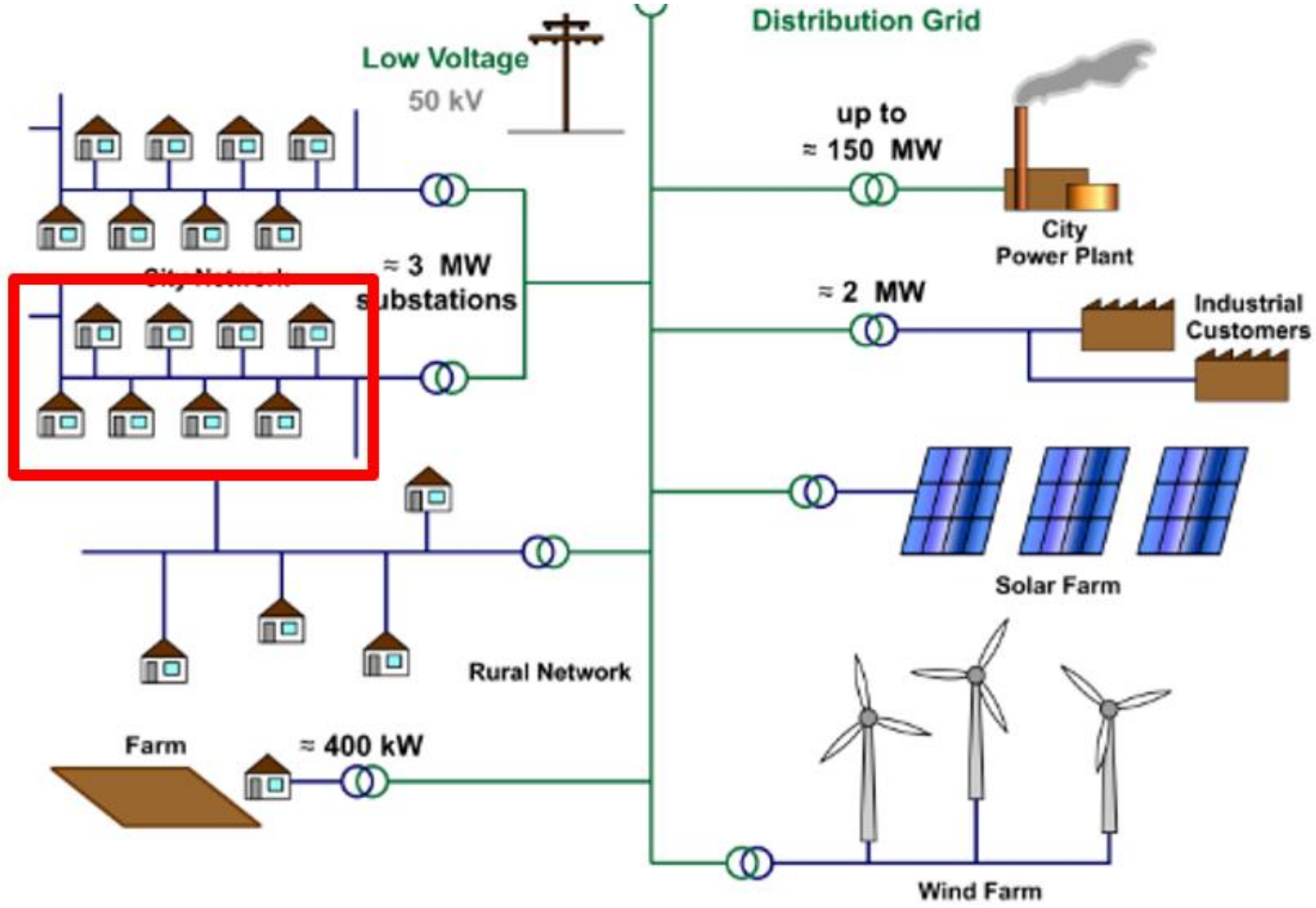
Synthetic Load Profile Generation

Challenge: Data scarcity

Solution: Synthetic data generation

Gap: Can only generate one load profile at a time

Goal: Generate **a group of diversified load profiles** considering **spatial-temporal correlations**

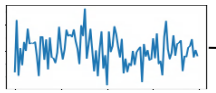


Y. Hu, Y. Li, L. Song, H.P. Lee, PJ. Rehm, M. Makdad, E. Miller, and N. Lu, "MultiLoad-GAN: A GAN-Based Synthetic Load Group Generation Method Considering Spatial-Temporal Correlations," in *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 2309-2320, Mar. 2024, doi: 10.1109/TSG.2023.3302192

Existing Approach

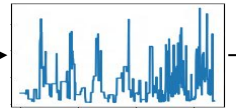
Step 1: Generate one load profile at a time

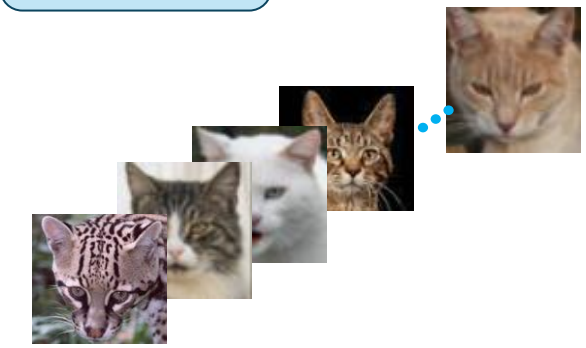
Noise



SingleLoad -GAN

1 load profile

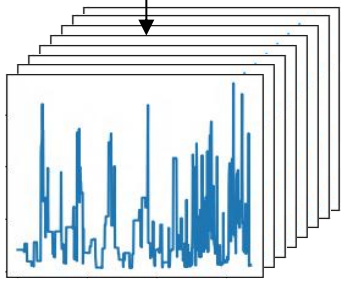




Step 2: Run step 1 iteratively to obtain a group of load profiles

Load profile Database

Step 3: Randomly sample N load profiles to form a group of loads



8 load profiles

Drawbacks:

Cannot account for group-level characteristics

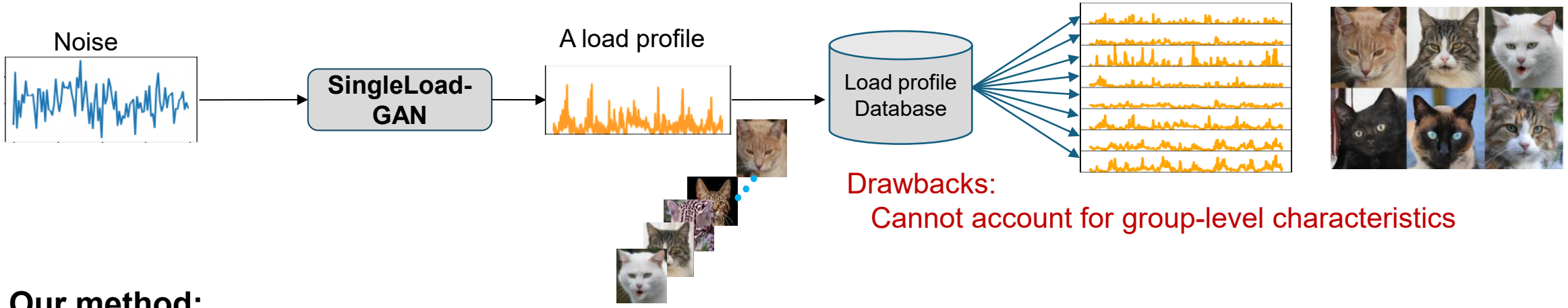
Multiload-GAN Approach → Correlation

Existing method:

Step 1: Generate one load profile at a time

Step 2: Run step 1 for many times to obtain a database of load profiles

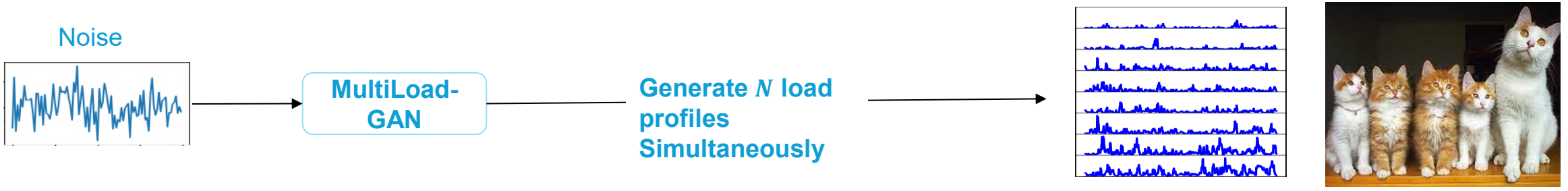
Step 3: Randomly sample N load profiles



Drawbacks:

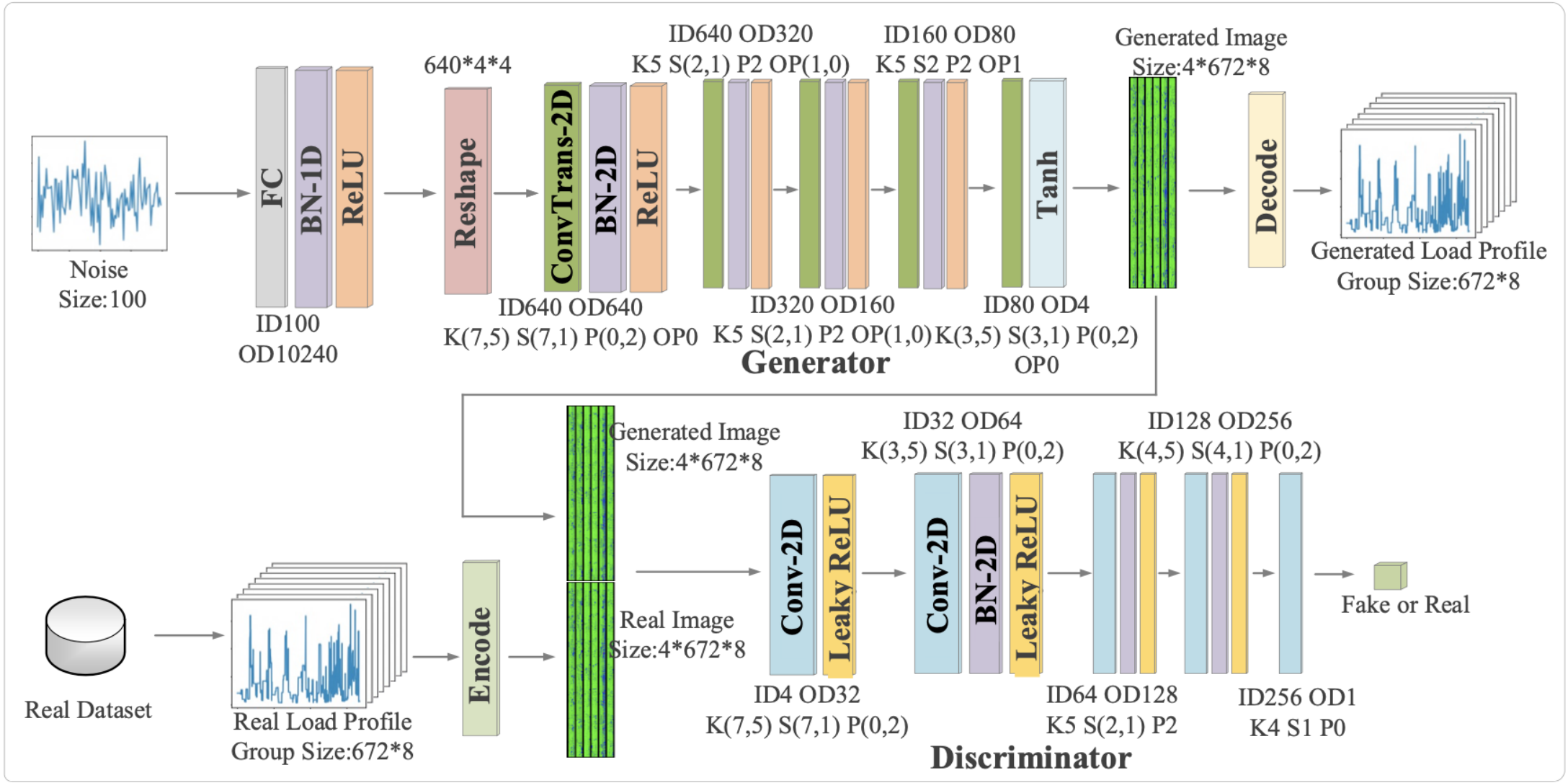
Cannot account for group-level characteristics

Our method:

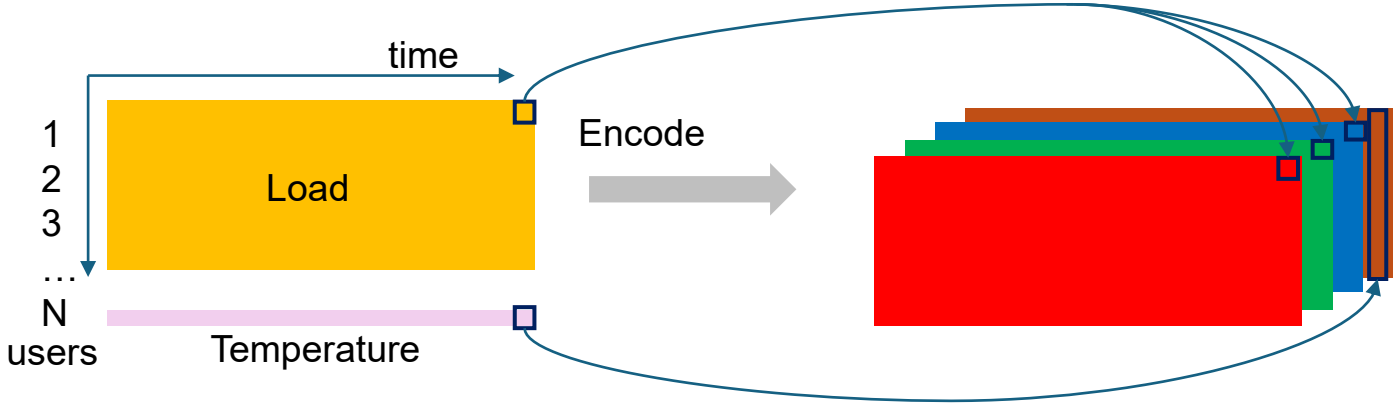


A group of load profiles

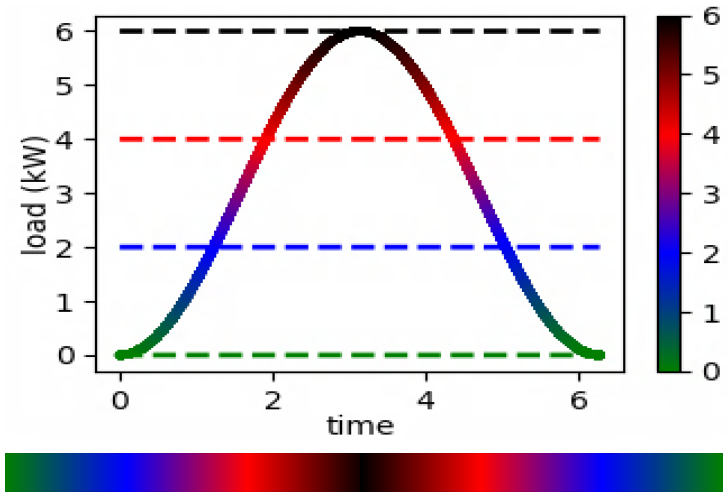
MultiLoad-GAN Configuration



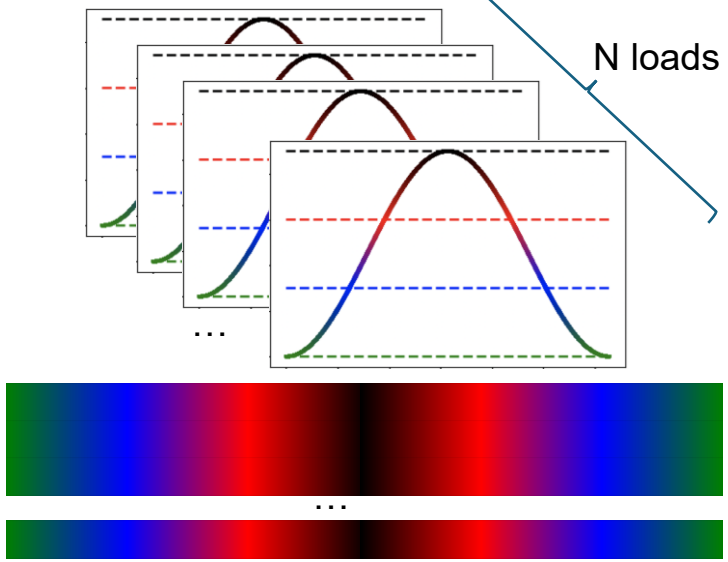
Load-to-Image Encoding



(a) Load group and temperature to 4-channel image



(b) Load profile to image bar

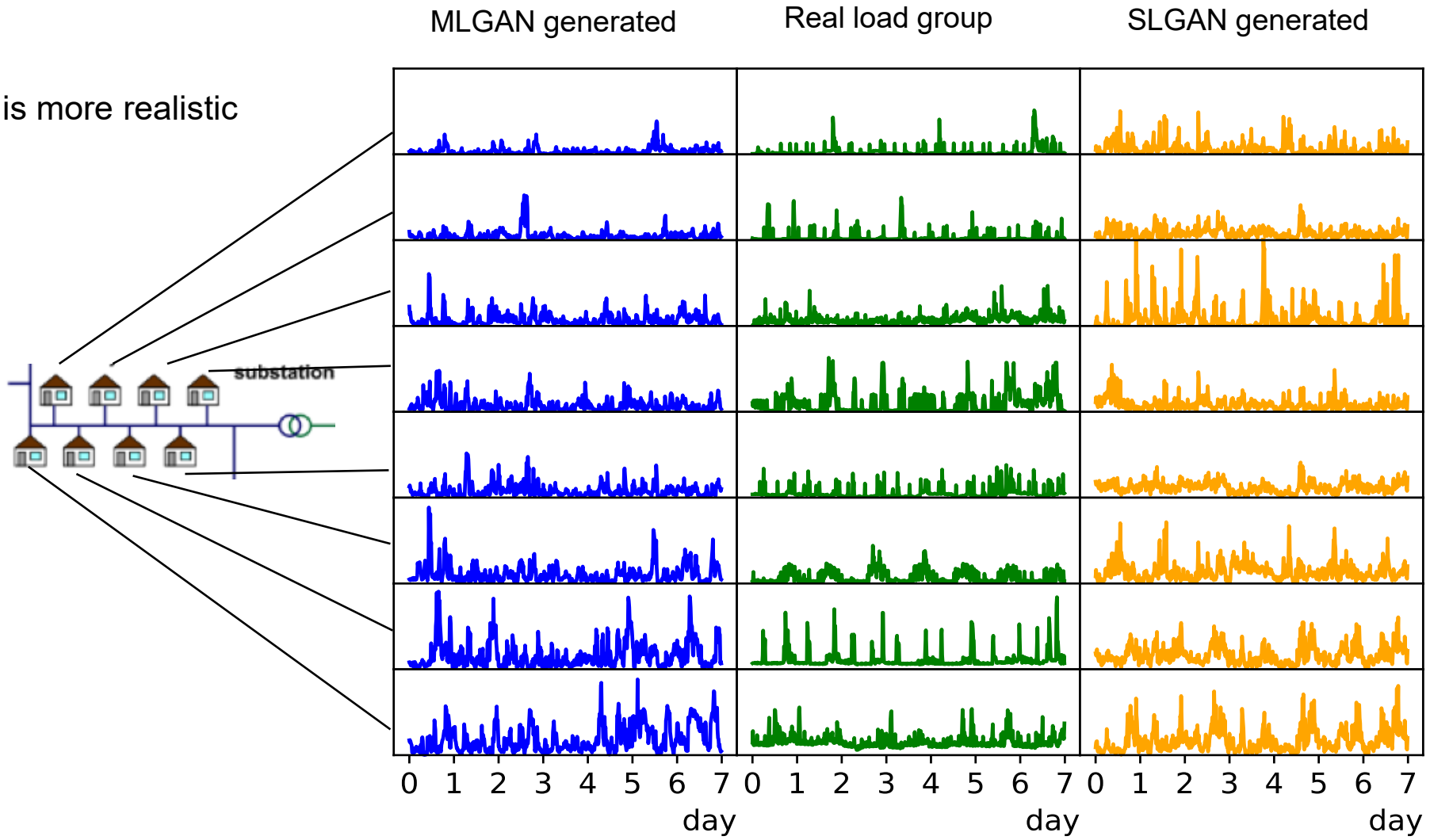


(c) Load group to image (N bars)

How to Evaluate the Generated Data?

Unique Challenge:

It's hard to decide which one is more realistic by visual inspection.

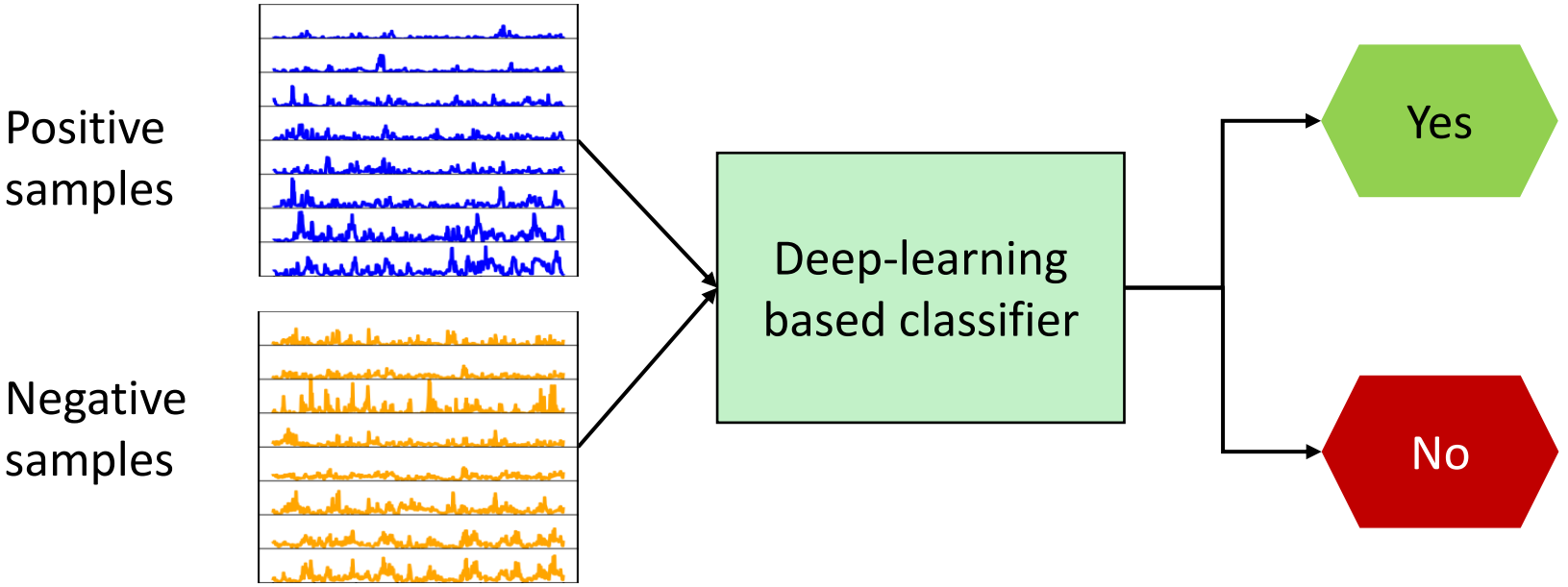


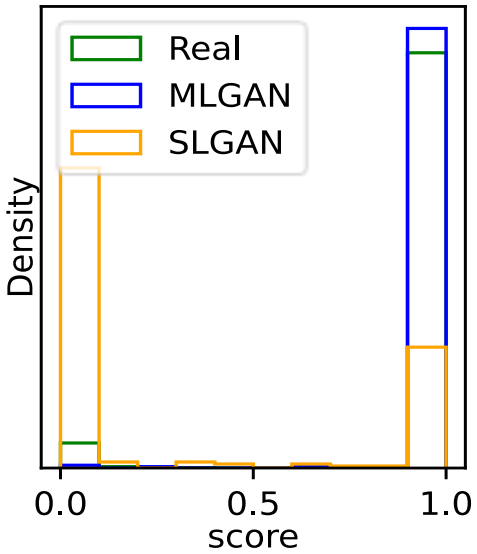
Statistical Evaluation

Level	Indices
Household	Peak load distribution
	Mean power consumption distribution
	Load ramps distribution
	Hourly energy consumption distribution
	Daily energy consumption distribution
Transformer	Peak load distribution
	Mean power consumption distribution
	Load ramps distribution
	Hourly energy consumption distribution
	Daily energy consumption distribution

Machine Learning Evaluation

– Whether or Not Group-Level Hidden Features Are Similar?





1. Percentage of Real (higher)

$$POR = \frac{Q_{real}\ for\ P_{true}(Q) > 0.5}{Q} \times 100\%$$

2. Mean Confidence Level

$$MCL = \frac{1}{Q}\sum_{i=1}^{Q} P_{true}(i)$$

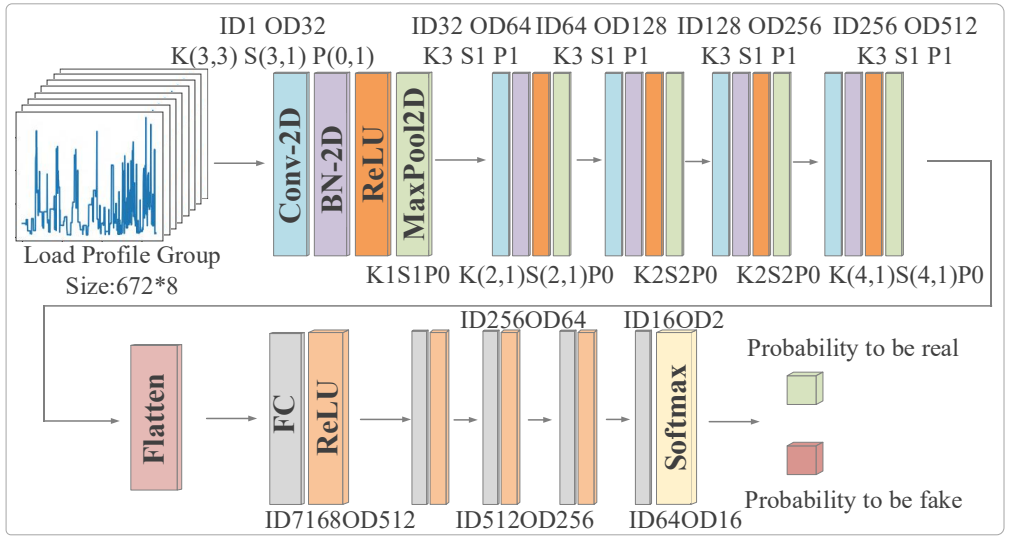
3. Confidence distribution

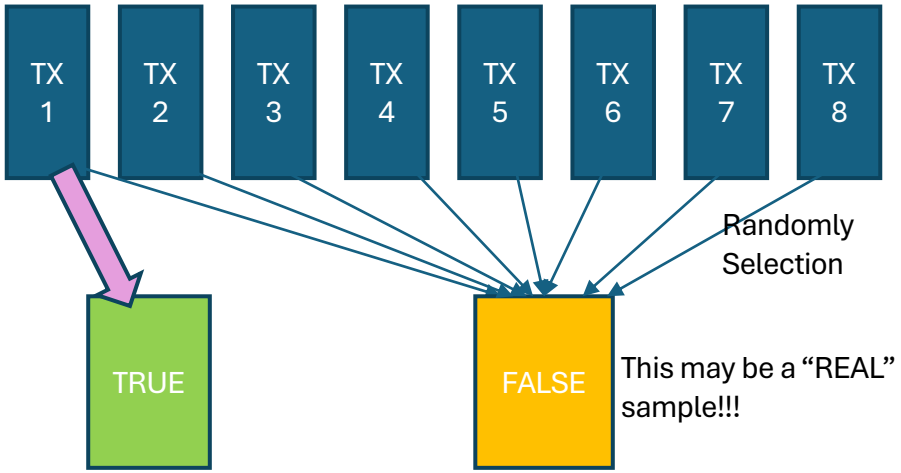
$$\tau\big(C(\boldsymbol{\Omega}_{LG})\big) = \tau([P_{true}(1), P_{true}(2), \dots, P_{true}(Q)])$$

4. Fréchet inception distance (small)

$$Similarity = FID(\tau(\boldsymbol{\Omega}_{\mathbf{LG}}), \tau(\boldsymbol{\Omega}_{\mathbf{LG}}^{\boldsymbol{MLGAN}}))$$

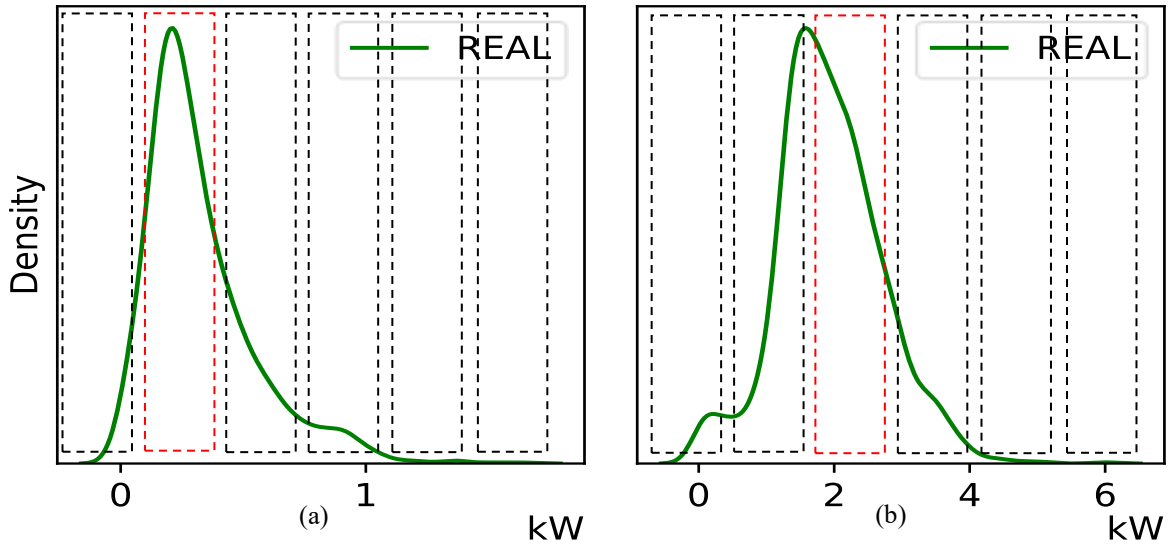
Deep Learning Classifier (DLC)





Negative Sample Generation:

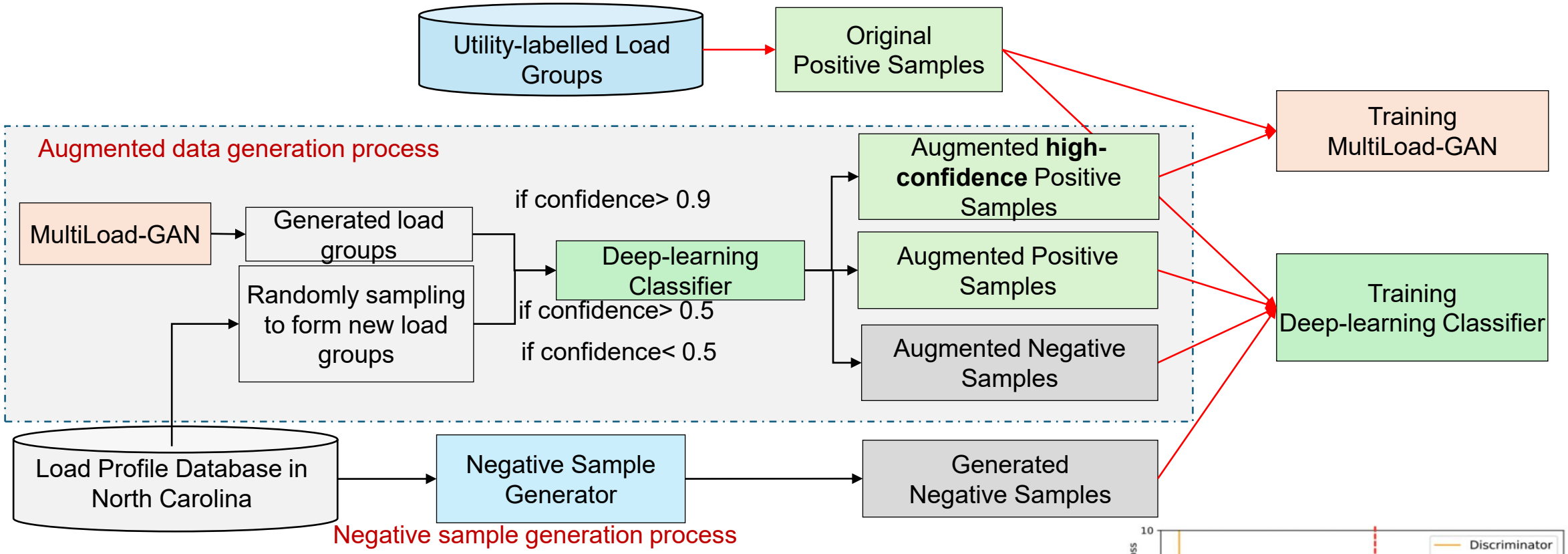
Select negative samples by mean and peak values

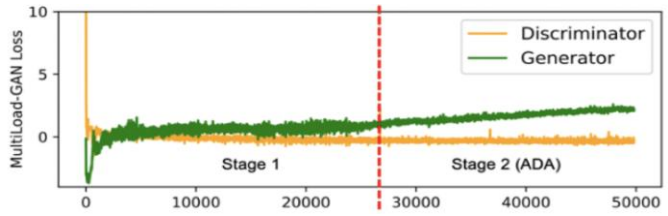


Select k loads from red box, 8-k loads from black.
k is a random integer in [0, 4]

Automatic Data Augmentation (ADA)

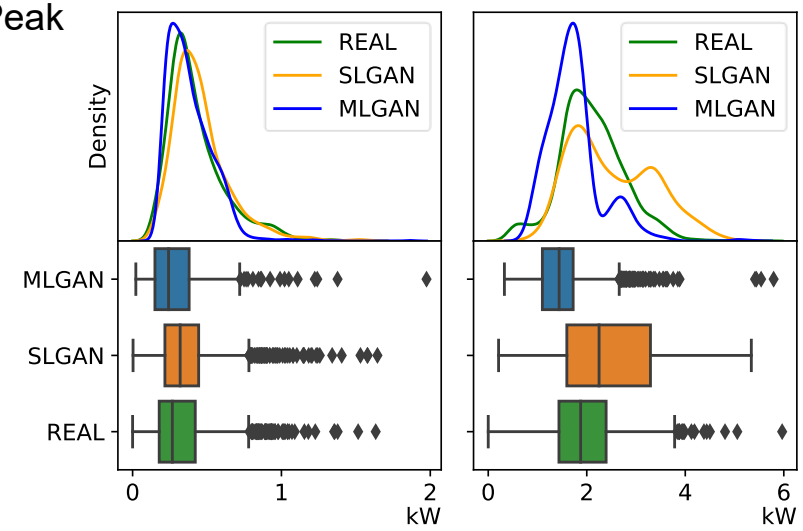
DLC and MultiLoad-GAN **iteratively augment** the training data of each other and **improve** the performance of each other.



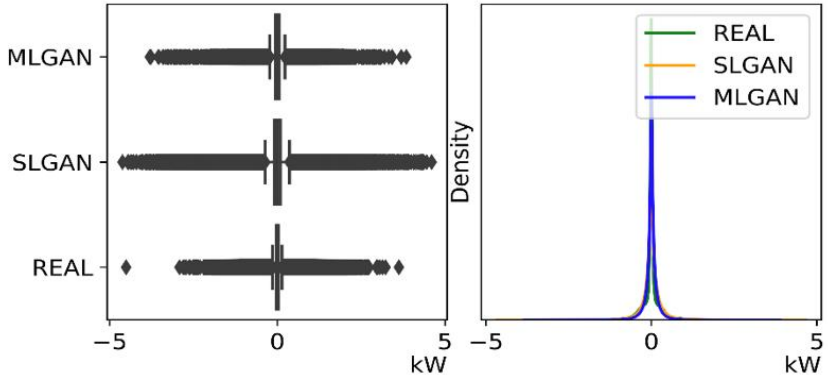


Results – Statistical Household Level

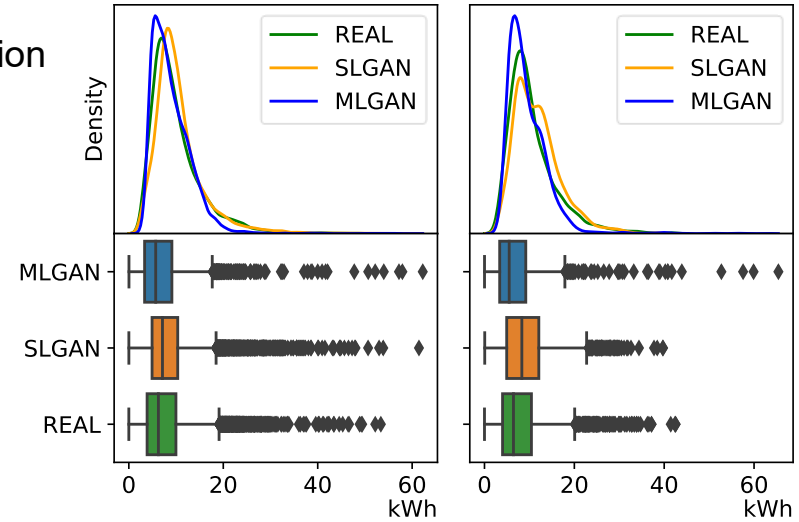
Mean & Peak



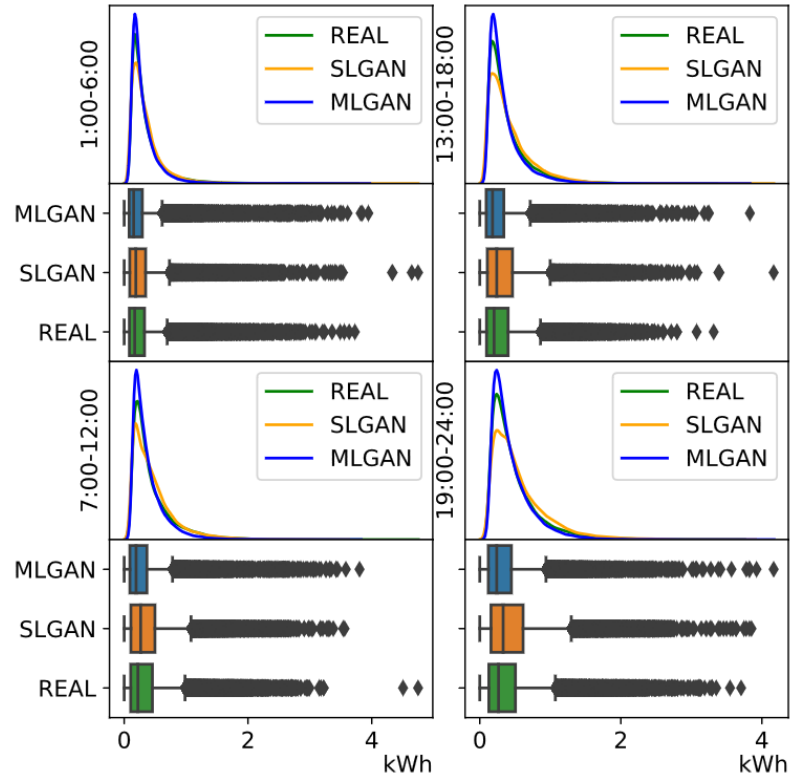
Ramping



Daily consumption

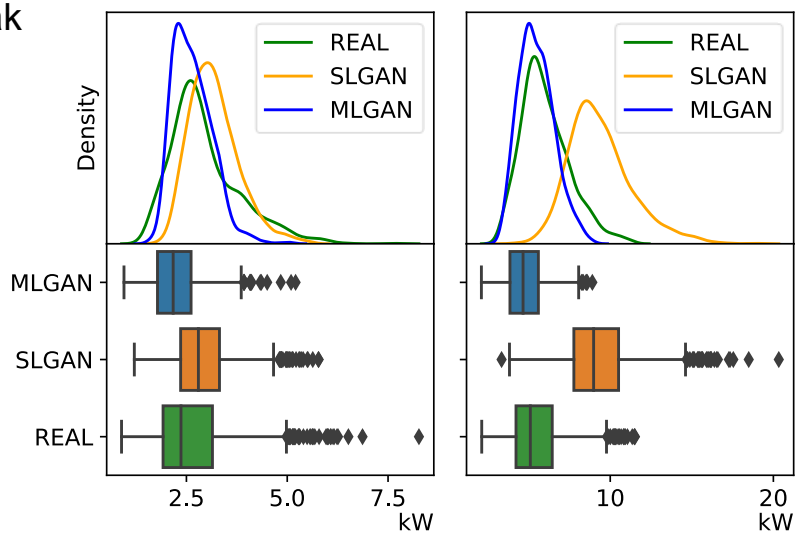


Hourly consumption

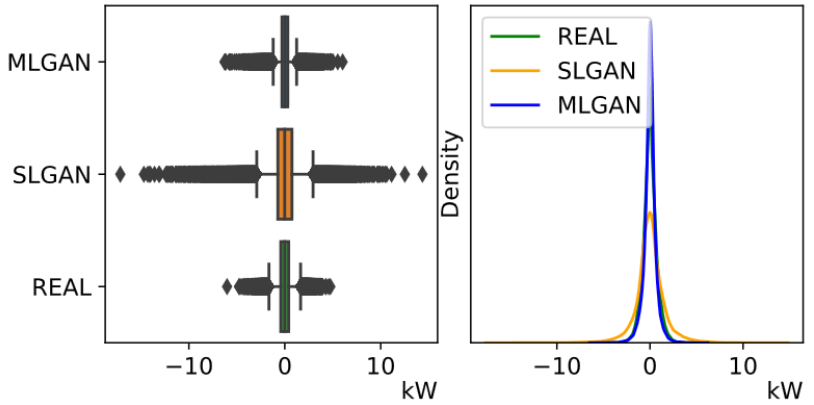


Results – Statistical Group Level

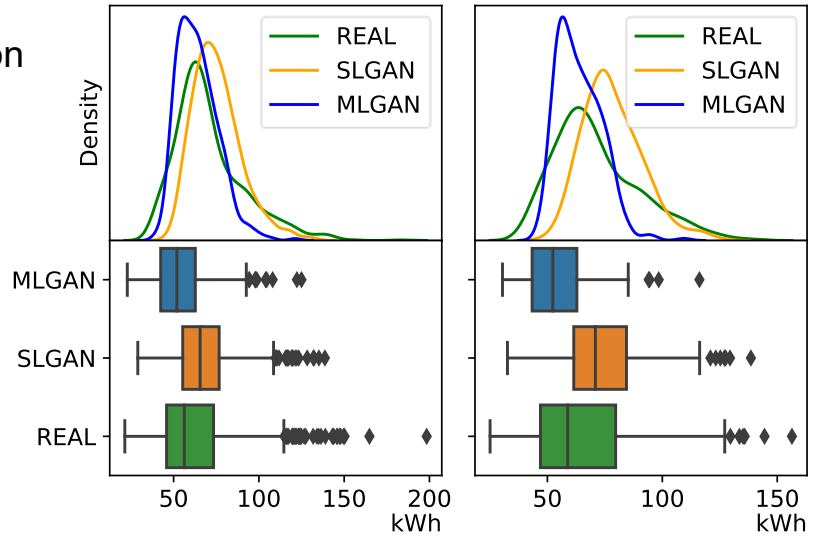
Mean & Peak



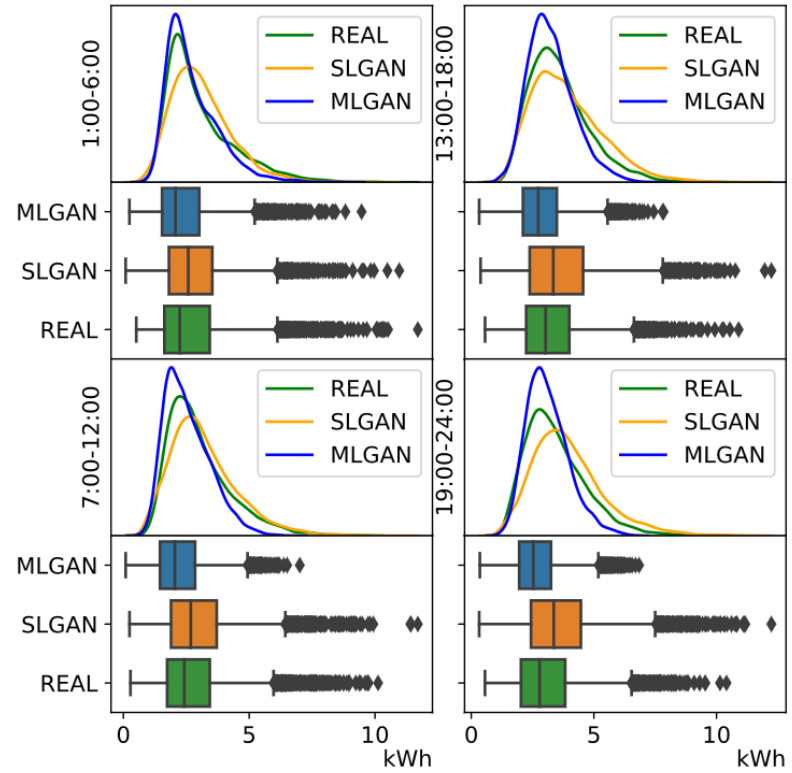
Ramping



Daily consumption



Hourly consumption



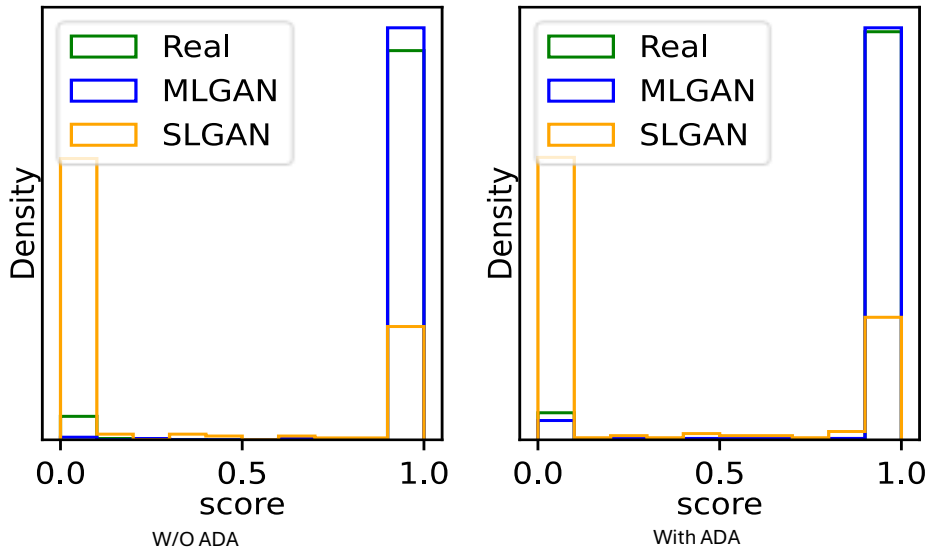
Results – Statistical Evaluation Summary

Evaluation Criteria		MLGAN	SLGAN	$\frac{\text{MLGAN}}{\text{SLGAN}}$ (<1 → MLGAN performs better)
Household Level	**Mean**	5.319e-4	6.473e-4	**0.8217**
	Peak	2.431e-2	4.884e-2	**0.4977**
	Ramping	6.700e-4	3.445e-4	1.945
	Hourly Consumption	(4.125e-4 1.115e-3 4.556e-4 6.726e-4) 6.639e-4	(1.464e-4 5.098e-4 5.225e-4 1.482e-3) 6.652e-4	**0.9980**
	Daily Consumption	(3.245e-1 3.610e-1) 3.428e-1	(2.516e-1 3.411e-1) 2.964e-1	1.157
Transformer Level	**Mean**	3.454e-2	3.491e-2	**0.9894**
	Peak	5.822e-2	2.252	**2.585e-2**
	Ramping	6.893e-4	1.074e-1	**6.418e-3**
	Hourly Consumption	(1.611e-2 3.608e-2 1.728e-2 2.738e-2) 2.421e-2	(5.634e-3 2.821e-2 3.507e-2 7.370e-2) 3.565e-2	**0.6791**
	Daily Consumption	(2.030e1 1.943e1) 1.987e1	(1.649e1 3.230e1) 2.440e1	**0.8143**

MultiLoad-GAN performs similar in household level but much better in transformer level evaluation.

Results

	Real Data	Scenarios			
		#1	#2	#3	#4
SLGAN		√			
MLGAN			√	√	√
ADA				√	√
Encoding			√		√
Percent of Real	94.38%	19.69%	99.06%	72.19%	**94.99%**
Mean Confidence Level	0.9371	0.1913	0.9899	0.7136	**0.9491**
Fréchet Inception Distance with the Real Load Group	N/A	0.5173	0.01106	**0.01120**	**0.000055**

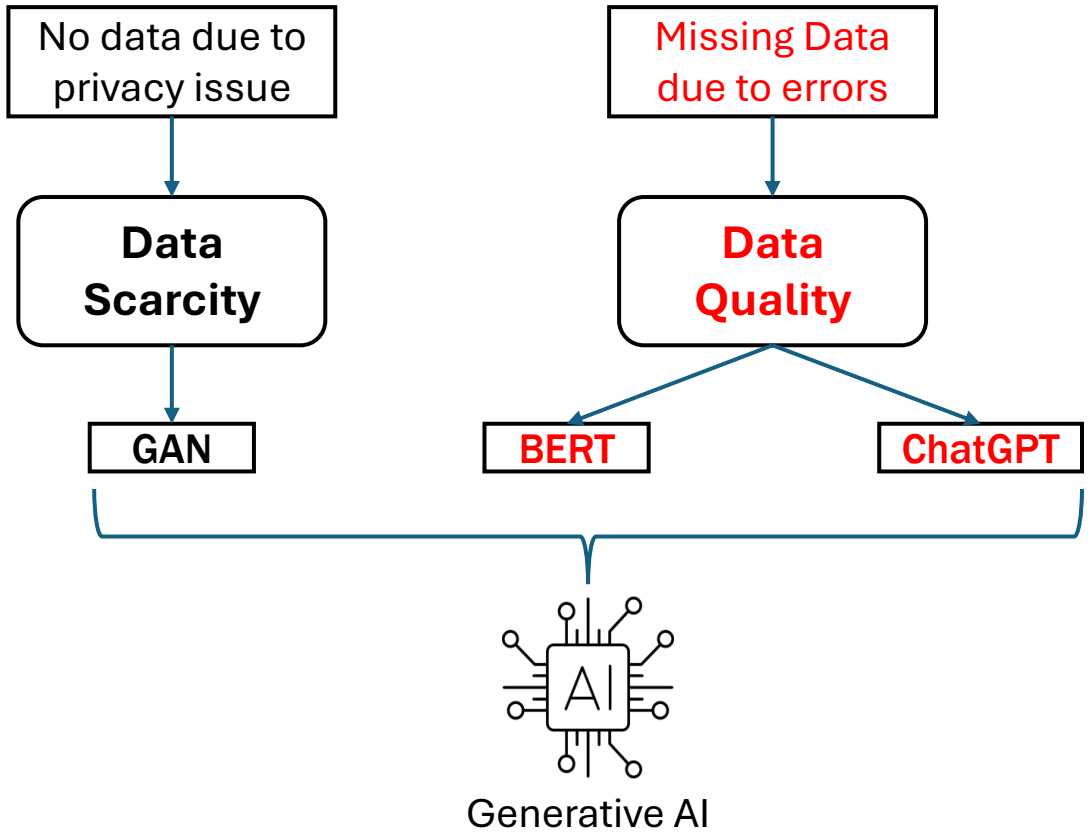


Conclusion – Multiload-GAN

- A SingleLoad-GAN and MultiLoad-GAN models are developed to generate synthetic load profiles;
- With the novel profile-to-image encoding-decoding method and the MultiLoad-GAN architecture design, ***a group of load profiles bearing realistic spatial-temporal correlations*** can be generated simultaneously;
- Two complementary metrics for evaluating realisticness of generated load profiles are developed: ***statistics metrics*** based on domain knowledge and ***a deep-learning classifier*** for comparing high-level features.
- To tackle data scarcity, a novel ***iterative data augmentation*** mechanism is developed to generate training samples for ***enhancing the training*** of both the deep-learning classifier and the MultiLoad-GAN model, which can improve the performance of MultiLoad-GAN by approximately 4.07%.

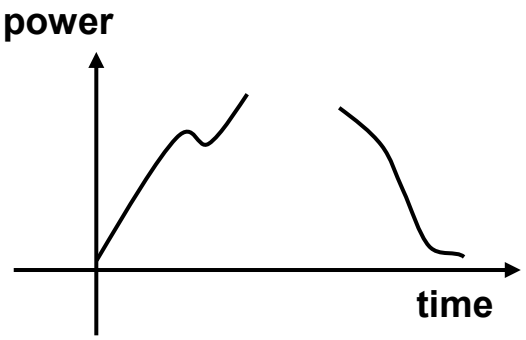
Y. Hu, Y. Li, L. Song, H.P. Lee, PJ. Rehm, M. Makdad, E. Miller, and N. Lu, "MultiLoad-GAN: A GAN-Based Synthetic Load Group Generation Method Considering Spatial-Temporal Correlations," in *IEEE Transactions on Smart Grid*, vol. 15, no. 2, pp. 2309-2320, Mar. 2024, doi: 10.1109/TSG.2023.3302192

Missing Data Restoration



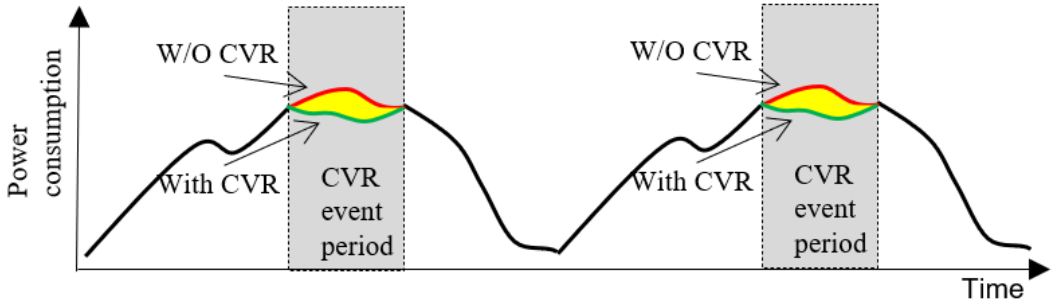
Missing Data Restoration

Improve data quality



Potential for **baseline estimation** for:

- Conservation Voltage Reduction (CVR)
- Demand Response (DR)



Two methods for this task:

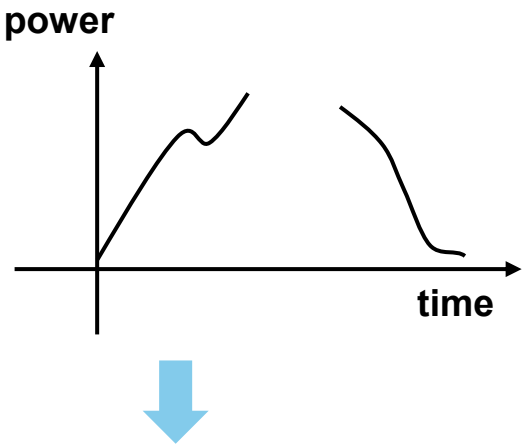
- Method 1: Bidirectional Encoder Representation from Transformer (BERT)
- Method 2: Fine-tuned Large Language Model (LLM)

Y. Hu, K. Ye, H. Kim and N. Lu, "BERT-PIN: A BERT-Based Framework for Recovering Missing Data Segments in Time-Series Load Profiles," in *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 12241-12251, Jul. 2024, doi: 10.1109/TII.2024.3417272.

Y. Hu, H. Kim, K. Ye, and N. Lu, "Applying Fine-Tuned LLMs for Reducing Data Needs in Load Profile Analysis," in *Applied Energy*, vol. 377, pp. 124666, Jan. 2025.

1. BERT-Based Missing Data Restoration

Power System Data Restoration task



Natural Language Processing task

Example: fill-in-the-blank

I have **lived** in _____ for five years,
so I can **speak fluent French**.

Challenge:

- Low accuracy
- Only generate one result

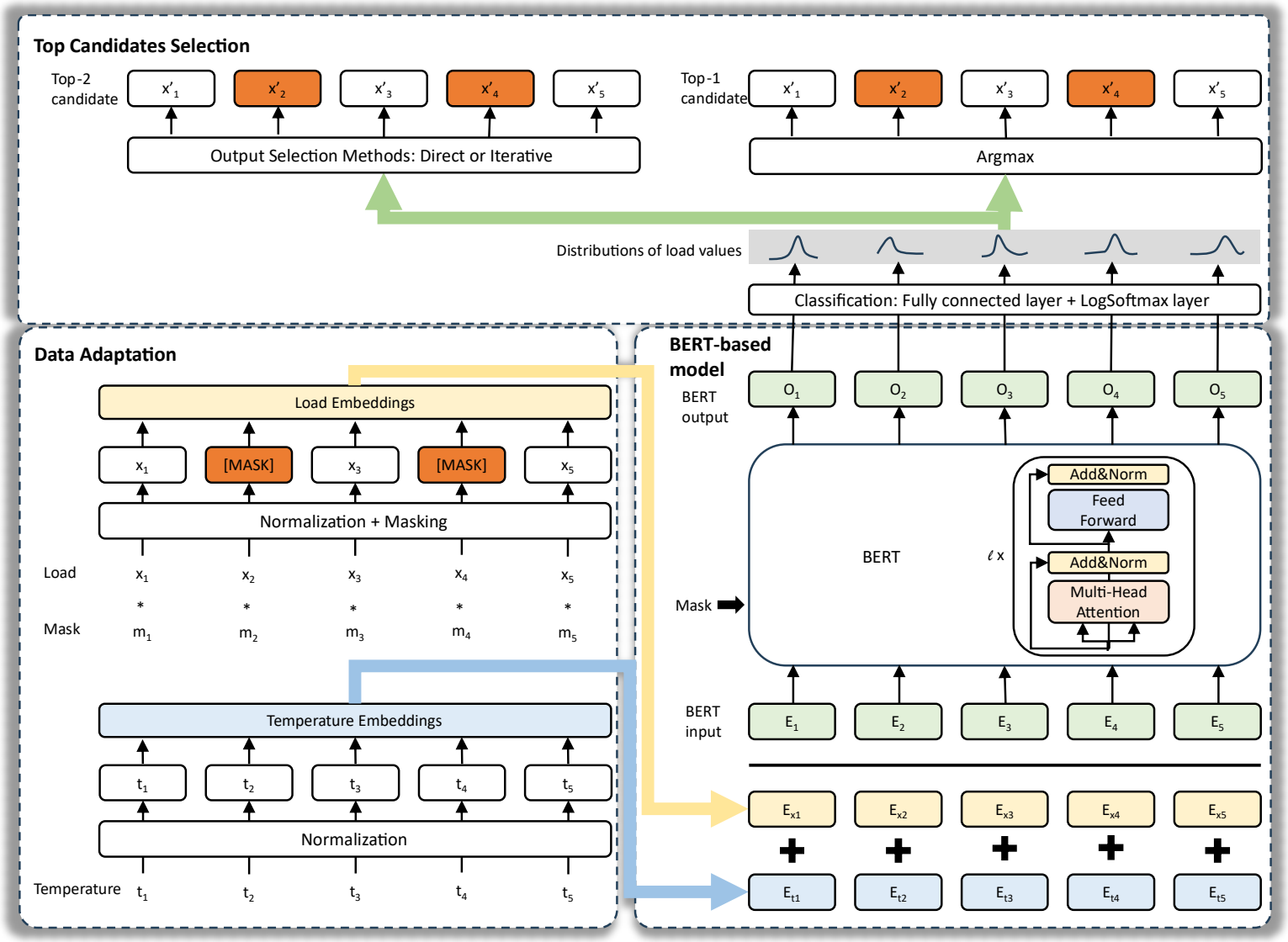
Advantages of BERT-PIN:

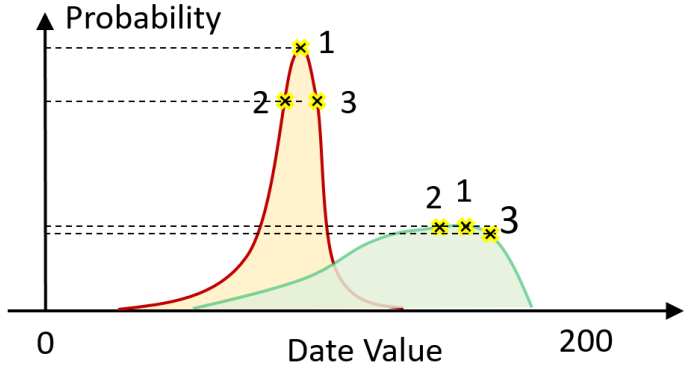
- Higher accuracy
- Multiple candidates to enhance the decision making

I have lived in France for five years, so I can speak fluent French.	**90%**
I have lived in Quebec for five years, so I can speak fluent French.	8%
I have lived in England for five years, so I can speak fluent French.	1%
⋮	
I have lived in apple for five years, so I can speak fluent French.	0.01%

Y. Hu, K. Ye, H. Kim and N. Lu, "BERT-PIN: A BERT-Based Framework for Recovering Missing Data Segments in Time-Series Load Profiles," in *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 12241-12251, Jul. 2024, doi: 10.1109/TII.2024.3417272.

BERT-PIN Configuration



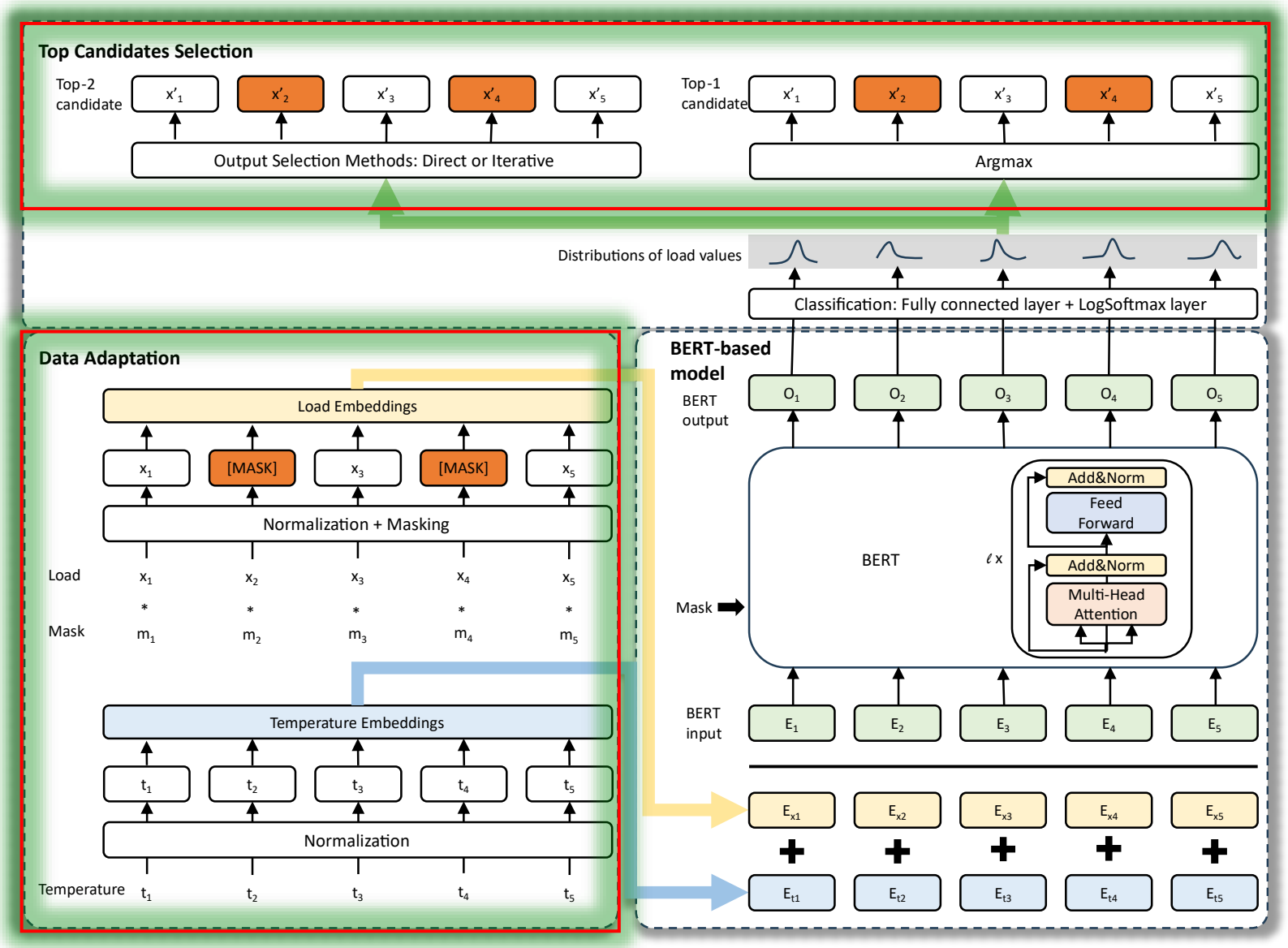


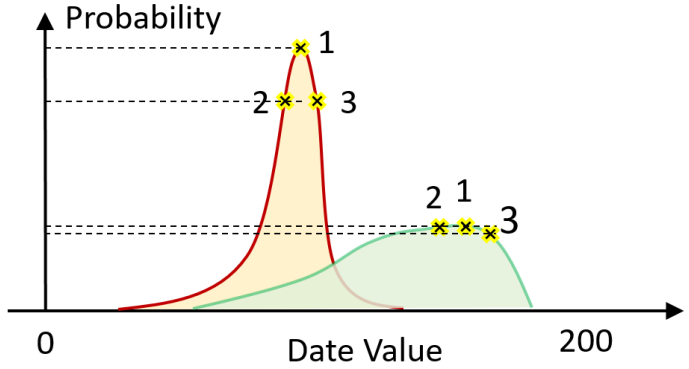
$$X = \frac{1}{P_{MAX}} \sum_{i=1}^{N_{Agg}} P_i \quad (t_{start}: t_{end})$$

It is a classification problem.

$$Loss = (1-\lambda) * CrossEntropy(\boldsymbol{X}, \widehat{\boldsymbol{X}}^{\boldsymbol{1}})$$
$$+\lambda * CrossEntropy(\boldsymbol{X}_m, \widehat{\boldsymbol{X}}_{\boldsymbol{m}}^{\boldsymbol{1}})$$

BERT-PIN Configuration





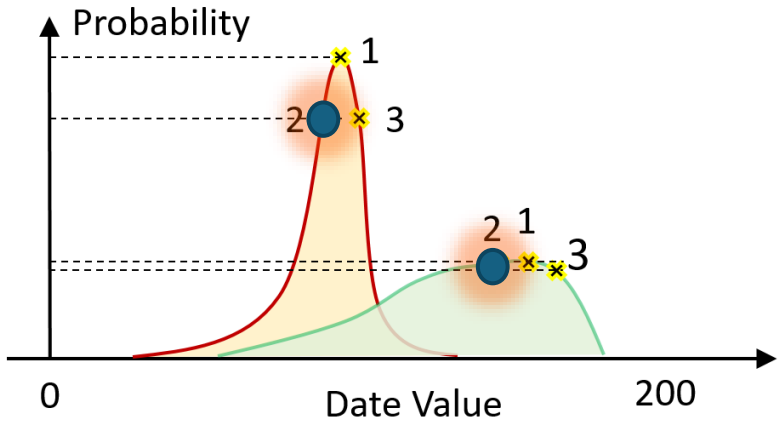
$$X = \frac{1}{P_{MAX}} \sum_{i=1}^{N_{Agg}} P_i \quad (t_{start}: t_{end})$$

It is a classification problem.

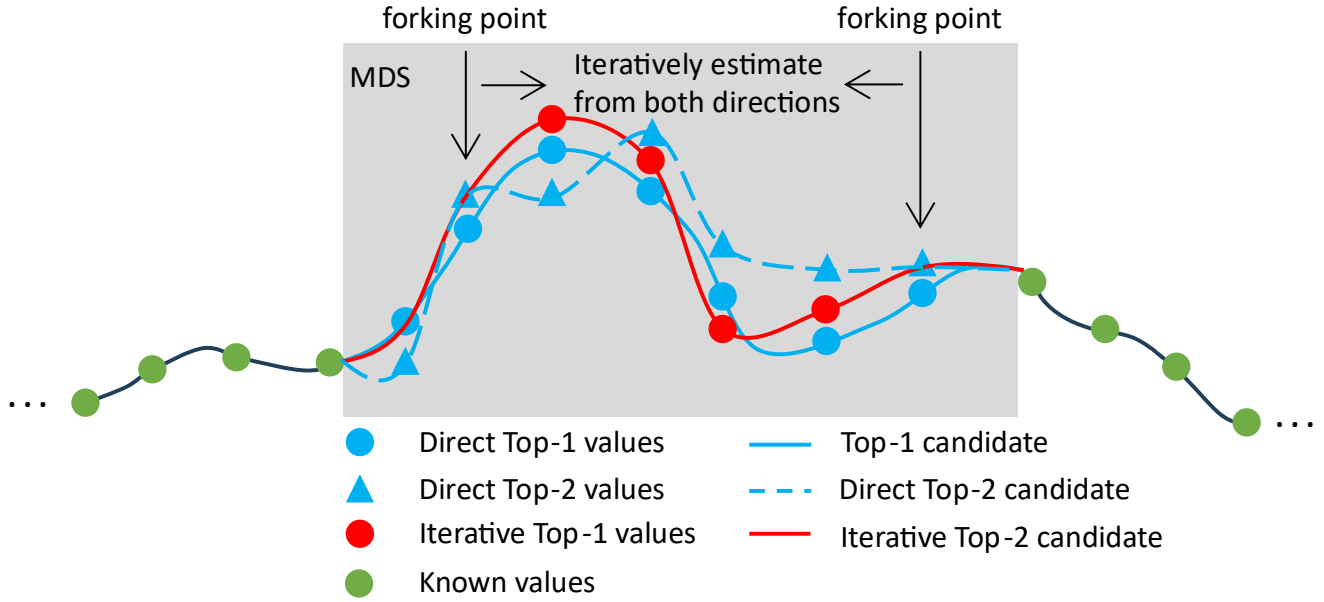
$$Loss = (1-\lambda) * CrossEntropy(\boldsymbol{X}, \widehat{\boldsymbol{X}}^{\boldsymbol{1}}) + \lambda * CrossEntropy(\boldsymbol{X}_m, \widehat{\boldsymbol{X}}_{\boldsymbol{m}}^{\boldsymbol{1}})$$

Top Candidate Selection

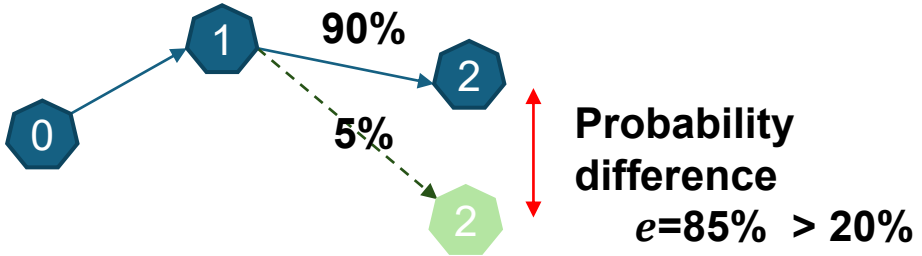
Method 1: Directly output the most likely results



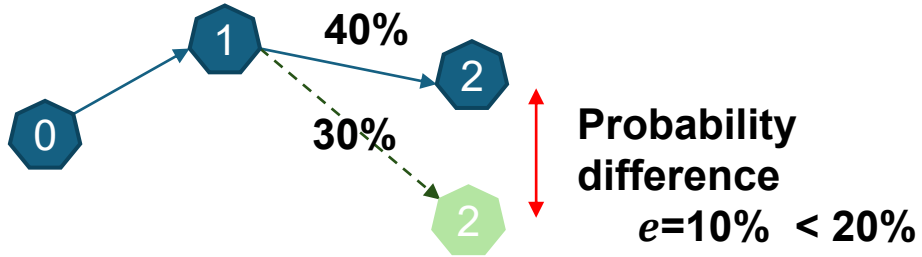
Method 2: Iteratively output top-2 results



Not a Forking Point

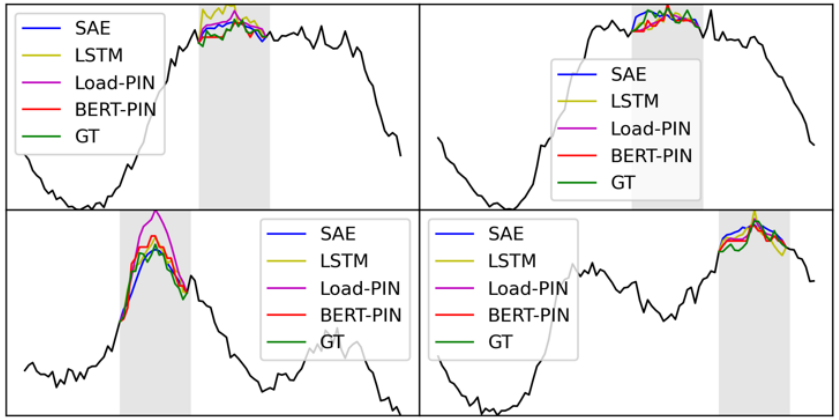


A Forking Point



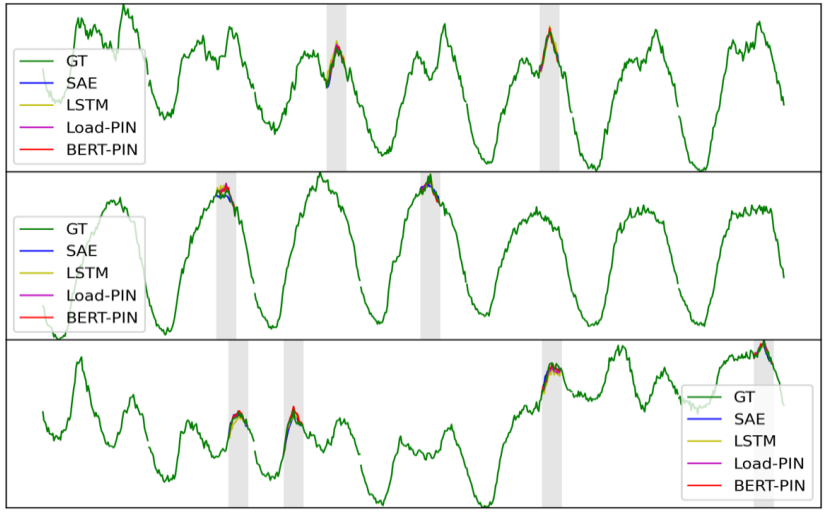
Top-1 Candidate Results

Restoration on Daily Profile



	SAE (%)	LSTM (%)	Load-PIN (%)	BERT-PIN (%)	Improvement
MPE	2.231	2.414	1.670	**1.523**	8.80%
RMSE	1.035	1.112	0.7951	**0.7404**	6.88%
PKE	1.065	0.8491	0.6183	**0.5130**	17.10%
VLE	0.8687	0.8991	0.6185	**0.5870**	5.09%
EGYE	1.525	1.762	1.165	**0.8410**	27.81%
FCE	2.138	2.046	1.615	**1.509**	6.56%

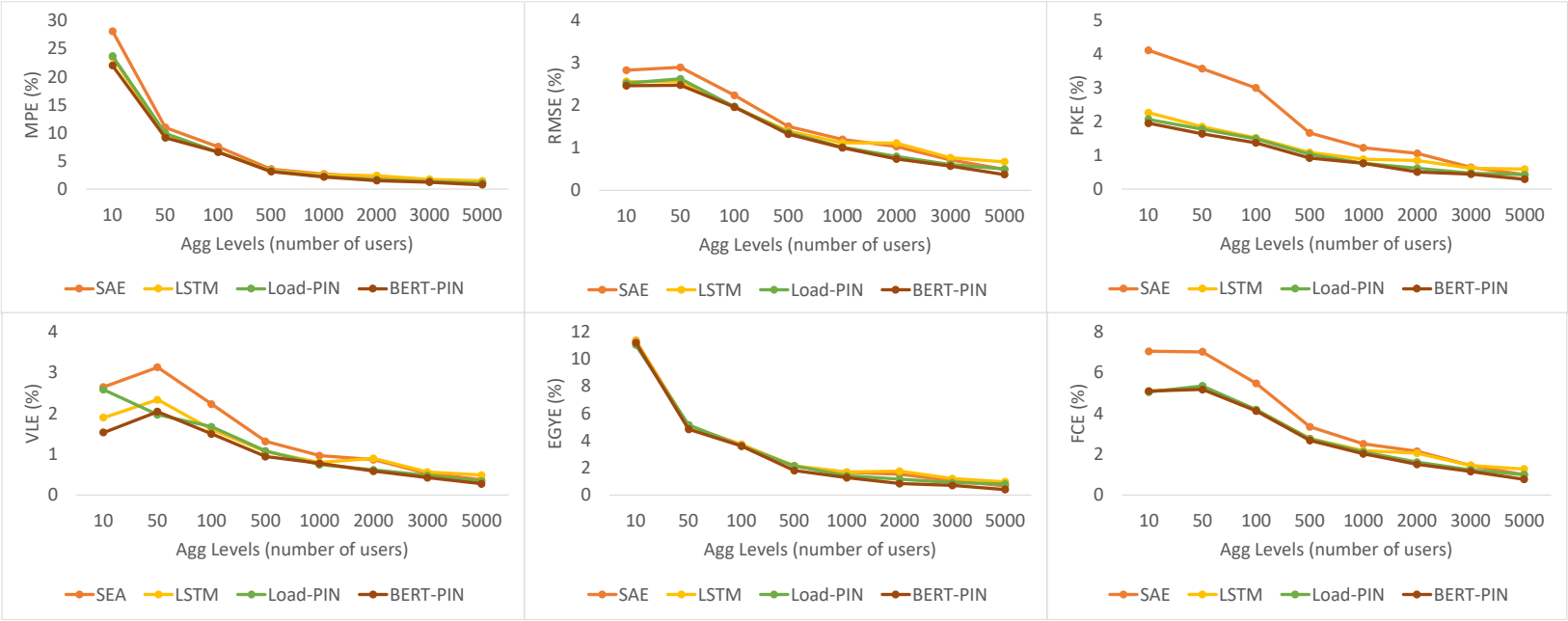
Restoration on Weekly Profile



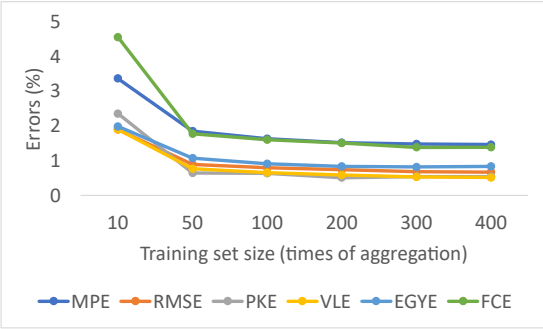
	SAE (%)	LSTM (%)	Load-PIN (%)	BERT-PIN (%)	Improvement
MPE	7.615	6.389	5.168	**4.837**	6.40%
RMSE	3.310	2.848	2.301	**2.221**	3.48%
PKE	3.362	2.193	1.769	**1.746**	1.30%
VLE	2.697	2.407	1.691	**1.627**	3.78%
EGYE	4.874	3.822	3.257	**2.699**	17.13%
FCE	6.828	5.601	4.808	**4.501**	6.39%

Top-1 Candidate Results – More Cases

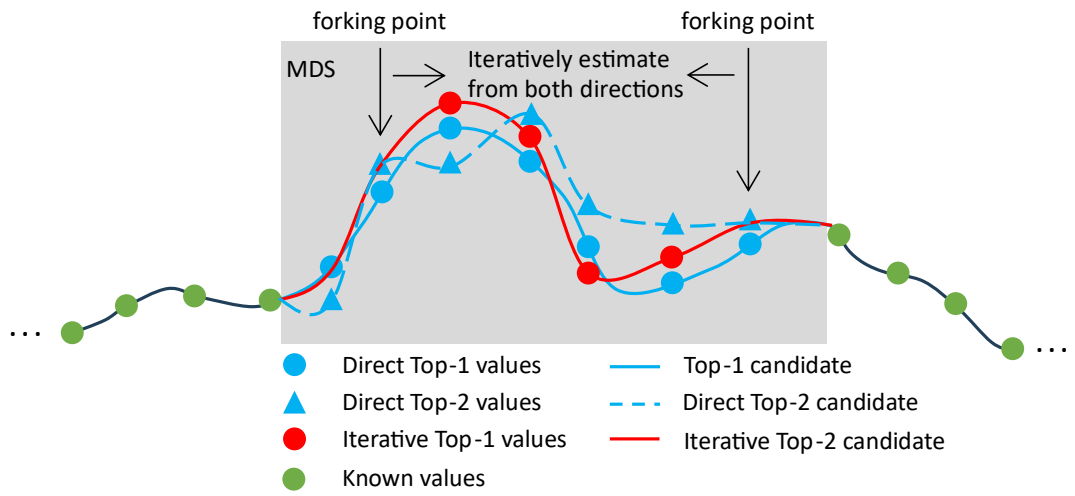
Aggregation Levels vs. Accuracy

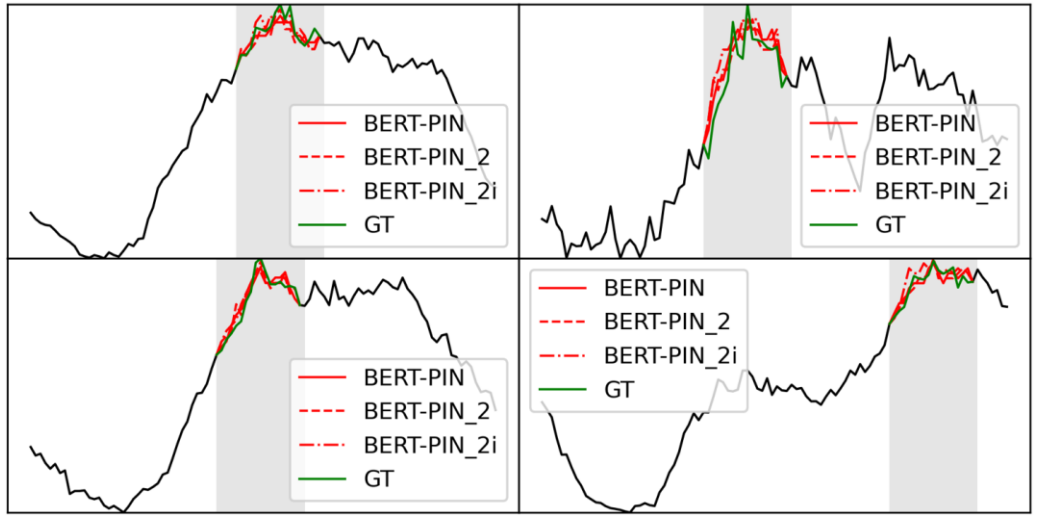


Data Size vs. Accuracy



Top-2 Candidate Results





	Top-1	Top2-direct	Top-2 iterative						Combine
			e=0.8	e=0.5	e=0.3	e=0.1	e=0.05	e=0.02	
MPE	1.523	1.744	2.556	2.433	2.407	2.138	1.87	1.761	**1.211**
RMSE	0.7404	0.9144	1.317	1.173	1.211	1.071	0.896	0.899	**0.577**
PKE	0.5130	0.5917	1.044	0.939	0.927	0.871	0.665	0.663	**0.426**
VLE	0.5870	0.9260	0.827	0.617	0.817	0.573	0.543	0.669	**0.407**
EGYE	0.8410	0.9582	1.618	1.441	1.412	1.301	1.043	0.986	**0.633**
FCE	1.509	1.942	2.447	2.18	2.273	2.007	1.727	1.78	**1.209**
PoCP	-	45.12%	23.88%	24.40%	22.94%	17.29%	12.25%	6.29%	-

Conclusion – BERT-PIN

- BERT-PIN (Bidirectional Encoder Representations from Transformers-based Profile Inpainting Network) aims to **restore multiple missing data segments** within load profiles.

- Contributions
  - Offers multiple restoration options within a given confidence level
  - Capable of restoring extensive missing data segments across long time periods

- Results:
  - Tested on different datasets for different aggregation levels and load sizes.
  - BERT-PIN shows 5%-30% greater accuracy than other methods for missing data recovery tasks.
  - Combining top-1 and top-2 predictions enables more comprehensive missing data predictions.
  - BERT-PIN excels in specific tasks like weekly data gap filling and demand response baseline estimation.

Y. Hu, K. Ye, H. Kim and N. Lu, "BERT-PIN: A BERT-Based Framework for Recovering Missing Data Segments in Time-Series Load Profiles," in *IEEE Transactions on Industrial Informatics*, vol. 20, no. 10, pp. 12241-12251, Jul. 2024, doi: 10.1109/TII.2024.3417272.

2. Fine-Tuned LLM-Based Missing Data Restoration

BERT-Based Method:

- Trained from scratch
- Long training time
- Large data need

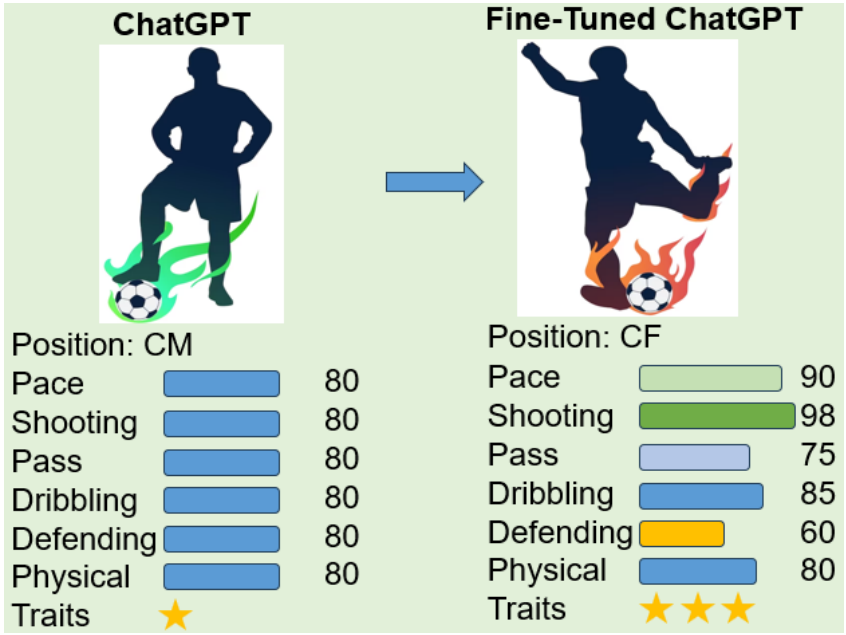
VS

Fine-Tuned LLM-Based Method:

- Fine-tuned from pretrained LLM
- Reduce **computational time and cost**
- Requiring only a **small dataset**

Contributions:

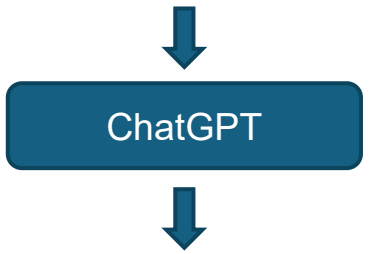
- **Pioneering Application:** the first application of a fine-tuned LLM to a specific power system data analytics task.
- **Innovative Fine-Tuning Strategy:** two-stage fine-tuning strategy using few-shot transfer learning.
- **Empirical Evaluation:** offer valuable insights for other load profile analysis tasks.



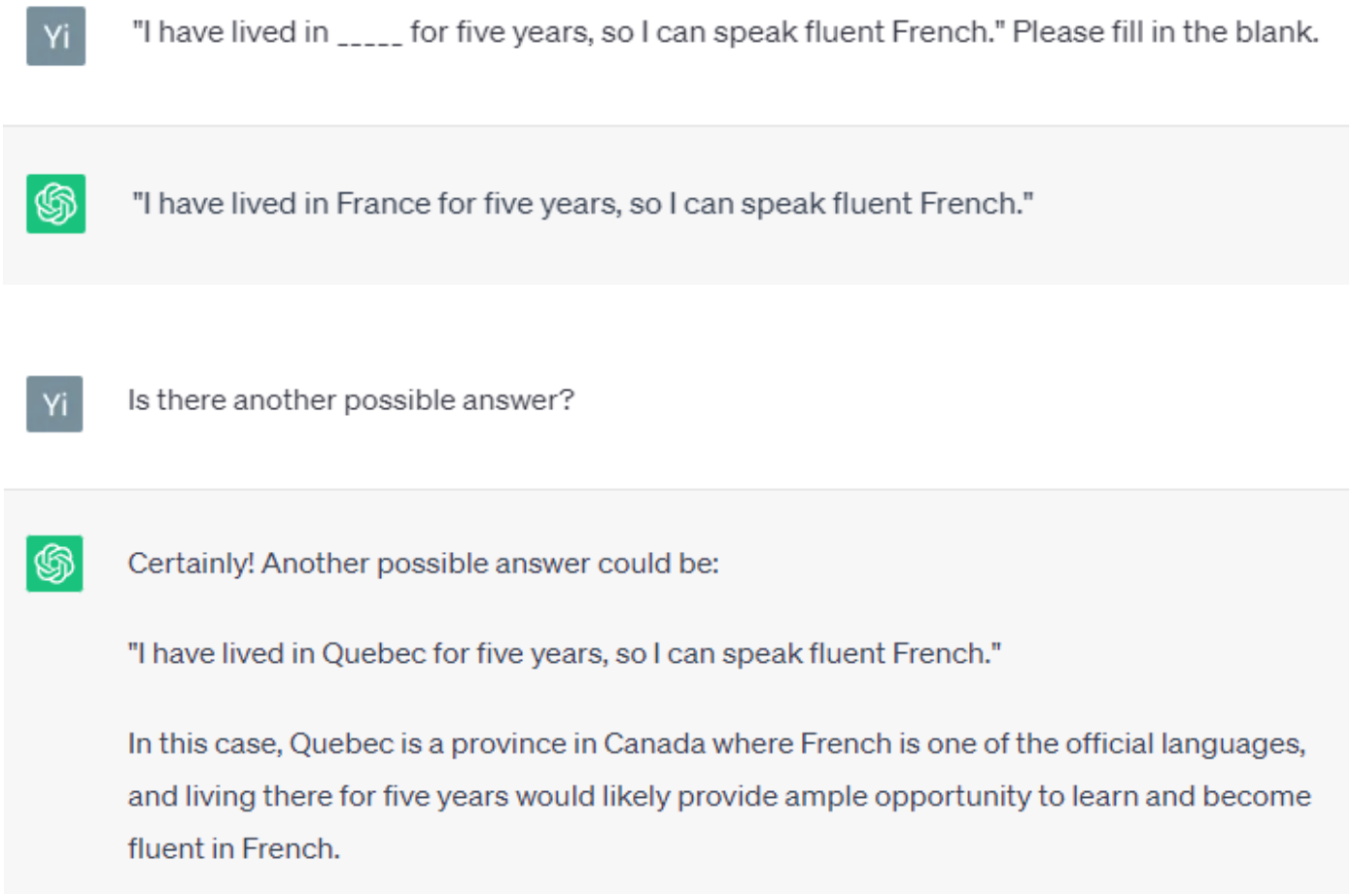
Y. Hu, H. Kim, K. Ye, and N. Lu, "Applying Fine-Tuned LLMs for Reducing Data Needs in Load Profile Analysis," in *Applied Energy*, vol. 377, pp. 124666, Jan. 2025.

Game with ChatGPT

I have lived in ____ for five years, so I can speak fluent French.

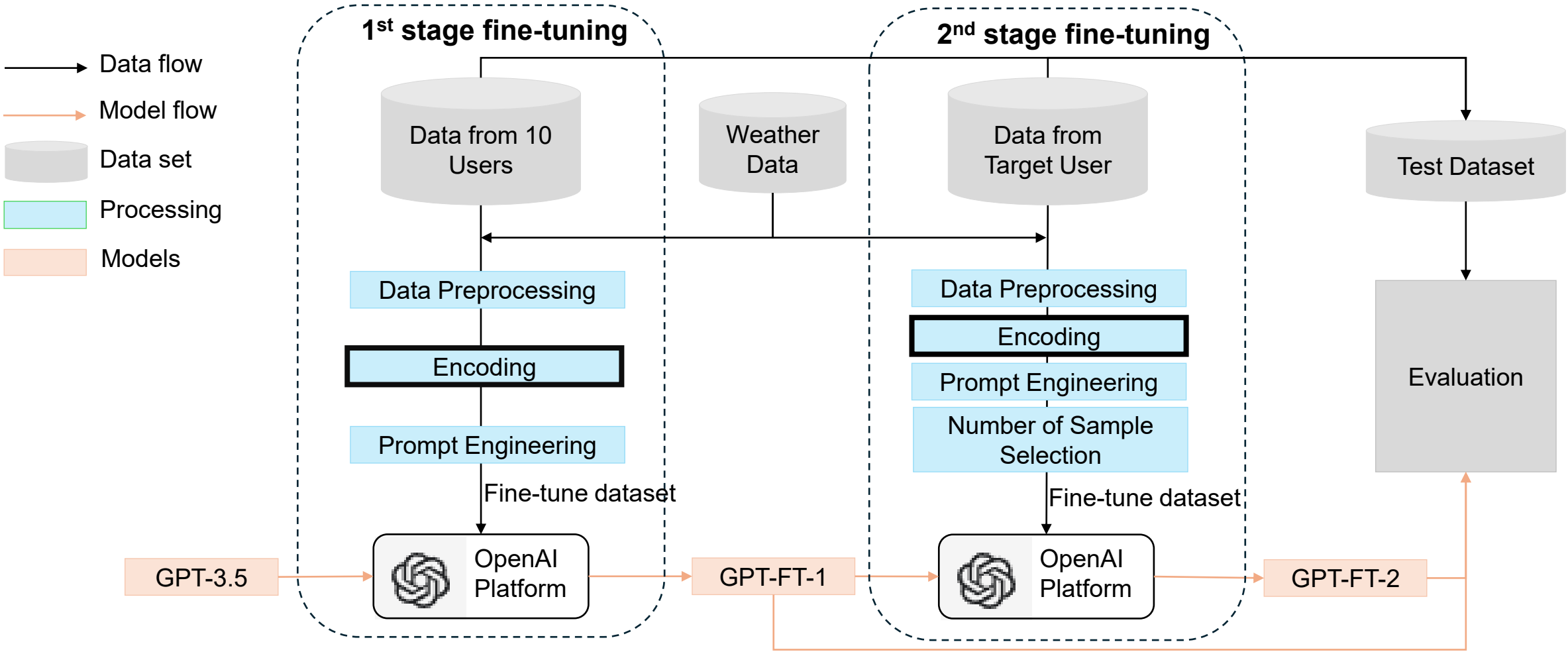


I have lived in France for five years, so I can speak fluent French.	90%
I have lived in Quebec for five years, so I can speak fluent French.	8%
I have lived in England for five years, so I can speak fluent French.	1%
⋮	
I have lived in apple for five years, so I can speak fluent French.	0.01%



But it does not work well on domain specific data!

Two-Stage Fine-Tuning Strategy



General domain knowledge → Specific task knowledge

Specific task knowledge → Specific user knowledge

Data Preprocessing

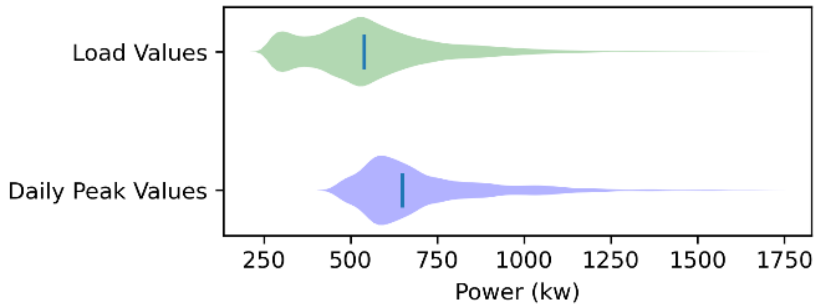
Given raw load profile $\boldsymbol{X}$, the masked load profile $\boldsymbol{X}^m$ can be represented as

$$\boldsymbol{M} = [m_i \; for \; i = 1{:}\,N], m_i = \begin{cases} 0, & missing\ data \\ 1, otherwise \end{cases}$$

$$\boldsymbol{X}^m = \boldsymbol{X} \cdot \boldsymbol{M}$$

Then, normalize masked load profiles to $[0, 200]$

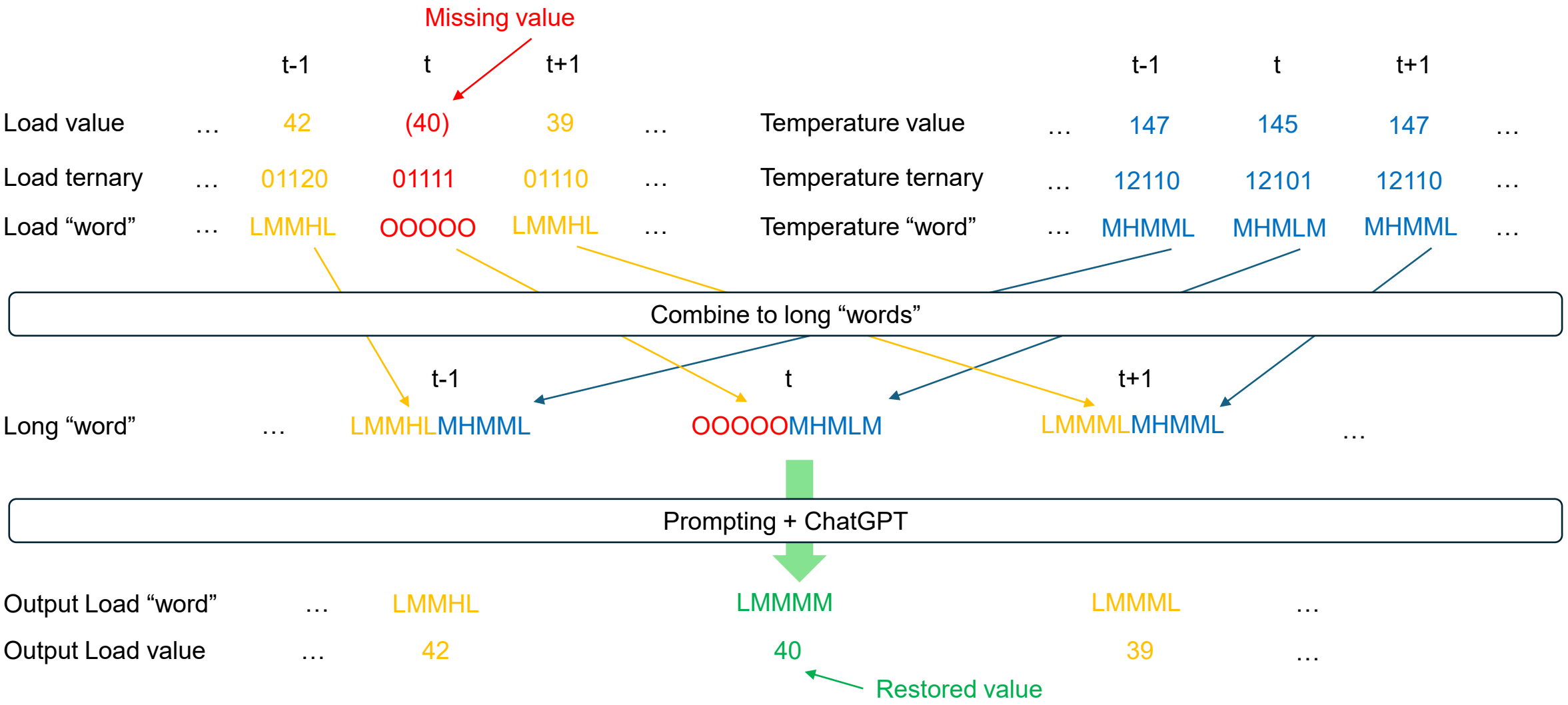
$$\boldsymbol{X^{norm}} = \frac{\boldsymbol{X^m} - min(\boldsymbol{X})}{max(\boldsymbol{X}) - min(\boldsymbol{X})} * 200$$



Temperature data $\boldsymbol{T}$ is normalized in the same way.

Encoding

– From Time-Series to "Words"



Prompt Engineering

- Prompt engineering is a methodology employed in the field of natural language processing (NLP) to design effective prompts for language models like GPT-3.5.
- It involves crafting specific instructions or queries that guide the model to generate desired outputs or responses.

Define the task

Describe the input data

Limit the output format

Indicate roles in the chat

```
{"messages":
    [{"role": "user",
      "content":"Given a load profile with missing segments and a complete"
                "daily temperature profile, estimate the missing portions of"
                "the load profile. The load and temperature data are provided"
                "separately and exhibit a correlation. Each value for load or"
                "temperature is encoded as a five-digit word in ternary format."
                "Missing load values are represented by OOOOO. Please provide"
                "the estimated load profile in the same format and length."},
     {"role": "assistant",
      "content": "What's the encoded load profile?"},
     {"role": "user",
      "content": load_encodings},
     {"role": "assistant",
      "content": "What's the encoded temperature?"},
     {"role": "user",
      "content": temperature_encodings},
     {"role": "assistant",
      "content": completion}
    ]
}
```

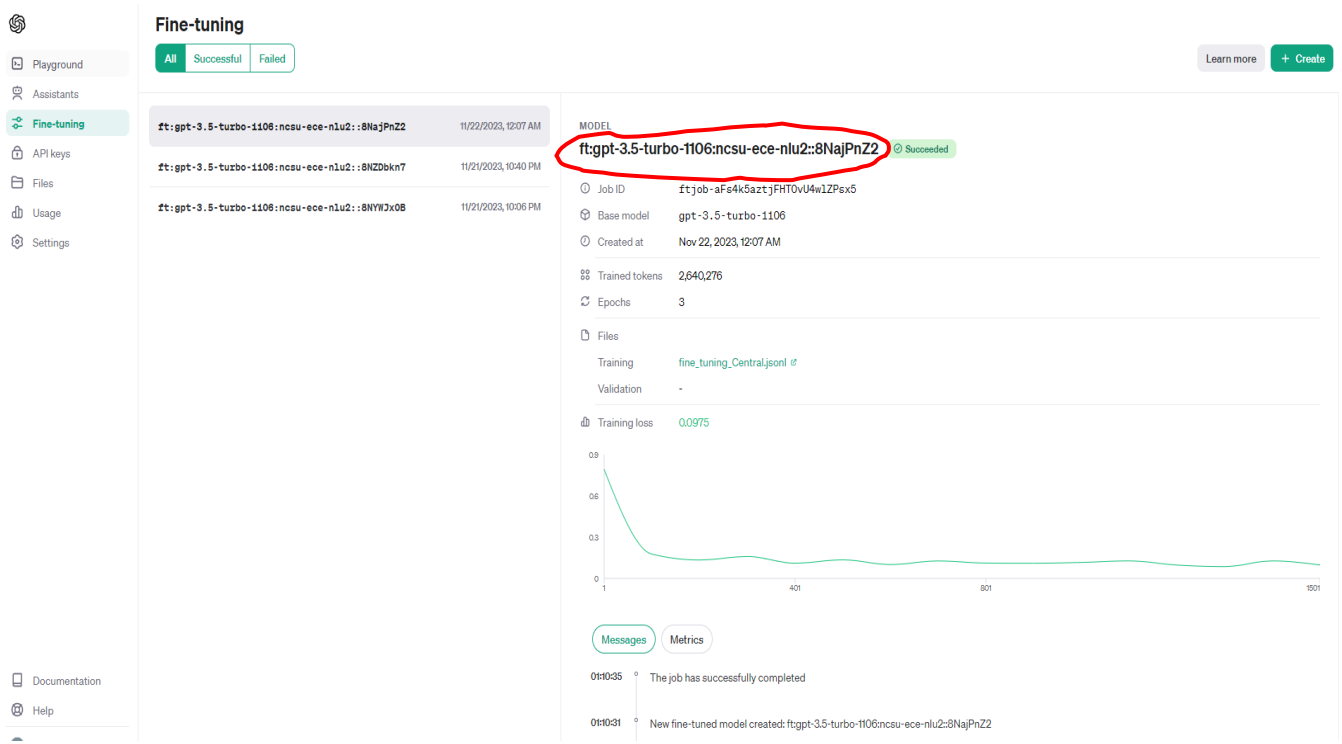
Provide load data

Provide temperature data

Get restored load data

OpenAI Fine-tuning

- Fine Tune website https://platform.openai.com/finetune
- Create a fine-tuning task and upload the data.
- Once the fine-tuning is finished, we can use the model with the name in red circle.

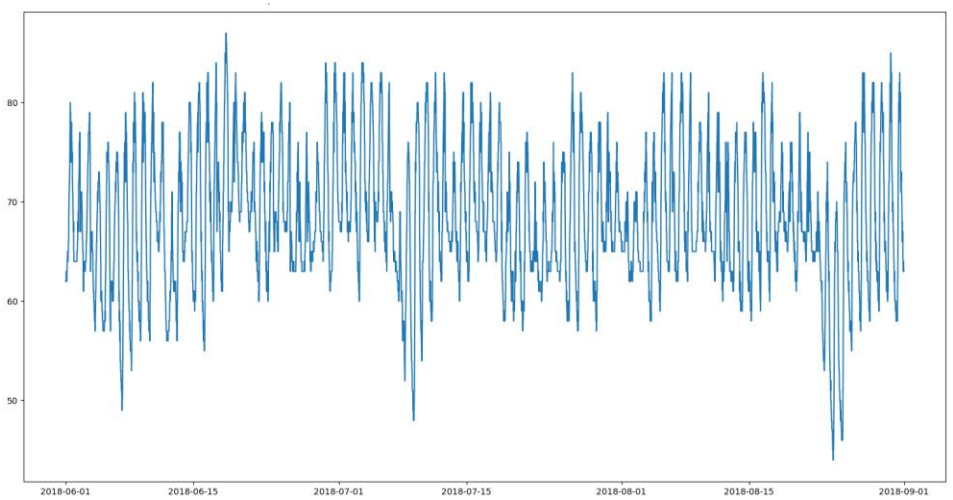


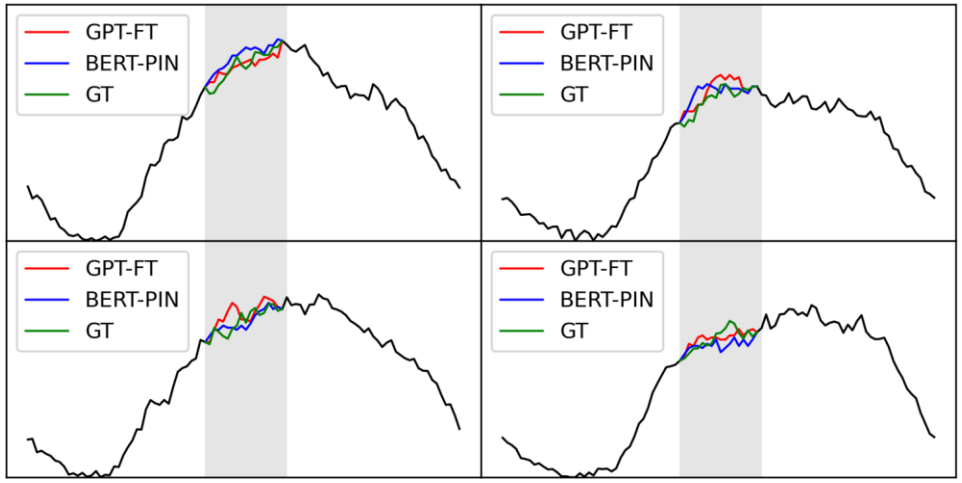
- An example to use fine-tuned model is shown bellow.

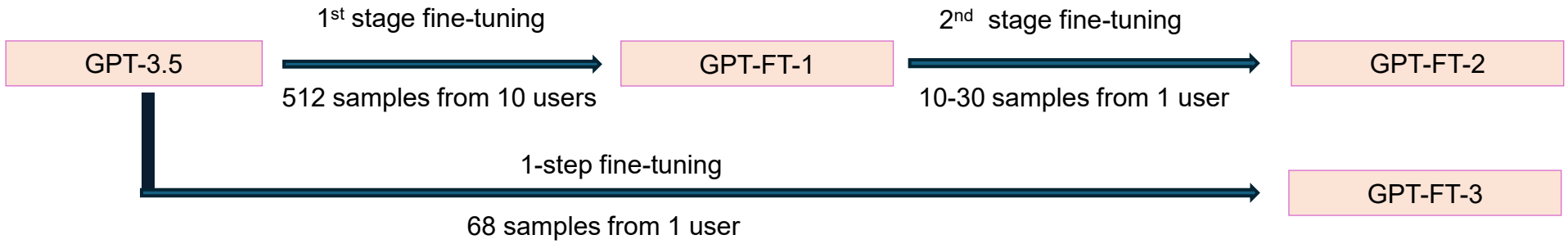
```
def openai_try():
    openai.api_key = "your_api_key"

    response = openai.ChatCompletion.create(
      model="gpt-3.5-turbo",
      messages=[
        {"role": "system", "content": "You will be provided with a sentence in English, and your task is to translate it into French."},
        {"role": "user", "content": "My name is Jane. What is yours?"}
      ]
    )

    print(response.choices[0].message)
```

Data Availability

Data collected from 20 users in 86 summer days, 68 days for training, and 18 days for testing.







1st Stage Results

Train and test on user #0-9

Models		# of samples	Advanced prompt	Separate Load & temperature	Discard Encoding	Remove abnormal days	Errors (%)		
							MPE	RMSE	EGYE
GPT-FT	Scenario 1	128	N	N	N	N	5.609	4.513	4.303
	Scenario 2	256	N	N	N	N	4.609	3.816	3.652
	Scenario 3	512	N	N	N	N	4.59	3.795	3.547
	Scenario 4	512	Y	N	N	N	3.806	3.027	2.987
	Scenario 5	512	Y	Y	N	N	3.266	2.656	2.469
	Scenario 6	512	Y	Y	Y	N	2.48	2.029	1.639
	Scenario 7	512	Y	Y	Y	Y	**2.221**	**1.977**	**1.443**
BERT-PIN		~220K	-	-	-	-	1.612	0.699	0.887

- Use **more samples in the first stage fine-tuning** leads to higher accuracy.
- Detailed information about the **task description**, **encoding strategy**, and **output constraints** provided in the advanced prompt improves the model's performance.
- Present load encoding and temperature encoding in a two-turn chat helps.
- ChatGPT can **comprehend direct numbers better** than encoded "words“.
- The **removal of abnormal days leads** to a further increase in accuracy.

2nd Stage Results

Train and test on user #10-19

Target Users	Errors (%)	GPT-FT-2 fine-tuned with different numbers of samples					GPT-FT-1 (W/O 2nd stage)		GPT-FT-3 (W/O 1st stage)
		10	20	30	40	50	Test on target user	Test on user0-9	
User10	MPE	2.245	2.277	2.428	2.395	2.361	2.497	2.221	2.983
	RMSE	1.988	1.969	2.014	2.094	2.140	2.182	1.977	2.772
	EGYE	1.501	1.513	1.599	1.604	1.637	1.843	1.443	2.461
User11	MPE	2.206	2.168	2.416	2.352	2.549	2.77	2.221	3.216
	RMSE	1.911	1.887	1.957	2.008	2.164	2.438	1.977	2.837
	EGYE	1.499	1.453	1.534	1.649	1.692	2.103	1.443	2.524
User12	MPE	2.276	2.453	2.483	2.533	2.676	2.441	2.221	3.068
	RMSE	1.962	1.997	2.016	2.147	2.163	2.206	1.977	2.649
	EGYE	1.524	1.517	1.567	1.599	1.643	1.784	1.443	2.581
User13	MPE	2.382	2.448	2.565	2.604	2.412	2.503	2.221	2.979
	RMSE	2.009	2.018	2.097	2.105	2.166	2.263	1.977	2.729
	EGYE	1.587	1.589	1.535	1.673	1.749	1.958	1.443	2.441
User14	MPE	2.318	2.248	2.558	2.48	2.336	2.673	2.221	3.110
	RMSE	1.997	1.948	2.021	2.134	2.196	2.273	1.977	2.811
	EGYE	1.472	1.499	1.514	1.731	1.887	2.094	1.443	2.687

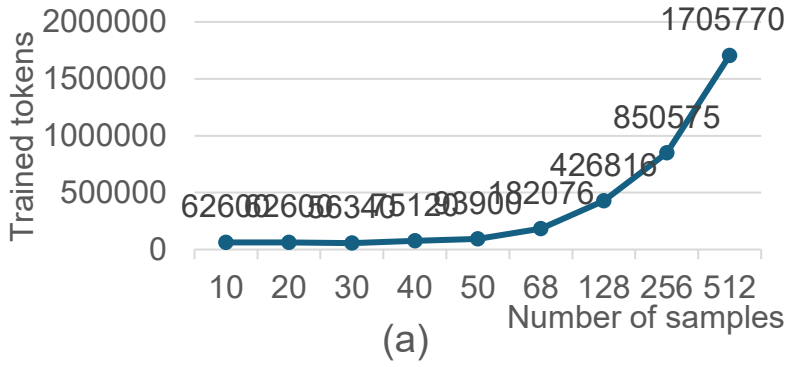
- The performance of the second stage fine-tuned model does not improve with an increase in training data size.
- Only a small number of samples are required for effective transfer of knowledge from general users to a specific user.
- The second stage fine-tuned model GPT-FT-2 exhibits improved accuracy for all target users.
- Direct fine-tuning of the original GPT-3.5 model with 68 samples from the target user results in inferior performance.

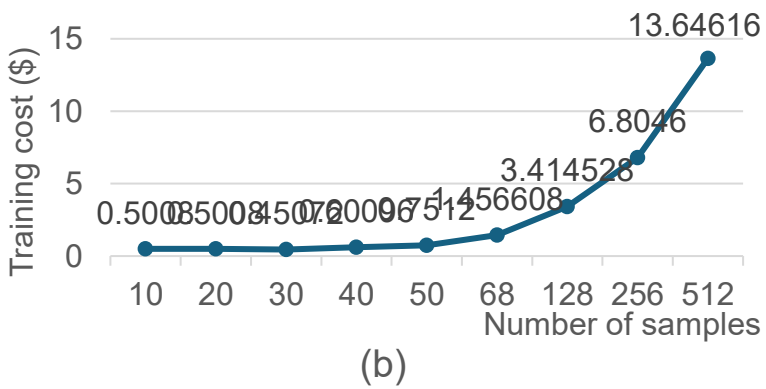
BERT-PIN vs Fine-tuned LLM

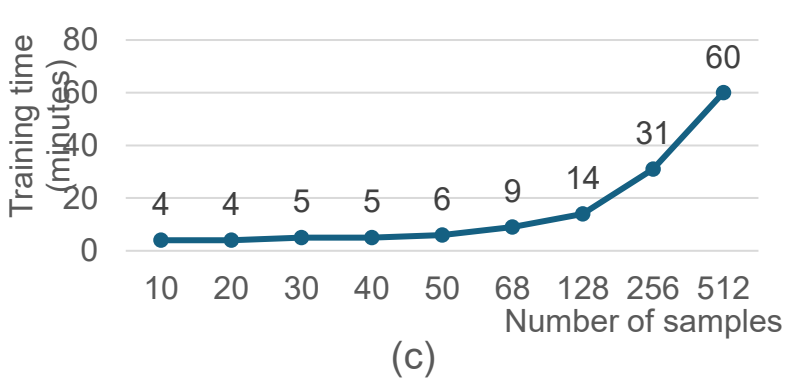
Compared With BERT-PIN

	Errors (%)			Data Needs	Training Time	Training Cost ($/Trail)
	MPE	RMSE	EGYE			
BERT-PIN (Specific model trained from scratch)	**1.612**	**0.699**	**0.887**	220K	6-8 hours	**0**
Fine-tuned ChatGPT (General model fine-tuned from pretrain)	2.221	1.977	1.443	**512**	**1 hour**	13.6

Cost Analysis







Conclusion – Fine-Tuned LLM

- Fine-tuned model restores missing data with accuracy comparable to advanced models like BERT-PIN.
- Key insights include the effectiveness of **using a small number of fine-tuning samples**, which underscores the efficiency of few-shot learning.
- Additionally, advanced **prompt engineering** and **separate encoding of load and temperature data** significantly enhance model performance.
- The fine-tuning strategy proved to be **cost-effective** and **time-efficient**, presenting a viable alternative to training models from scratch, especially in data-limited and resource-scarce settings.

Y. Hu, H. Kim, K. Ye, and N. Lu, "Applying Fine-Tuned LLMs for Reducing Data Needs in Load Profile Analysis," in *Applied Energy*, vol. 377, pp. 124666, Jan. 2025.

Open-Source Code Release

- **Code Repository:**

https://github.com/SyntheticDataGenerationAndSharing/SDG_Algorithms-Data

Thank You!