Physical Data Curation and Augmentation

Presented by : Divij Gupta, Krishna Konda

Agenda

Intro

- Motivation: Data Challenges in AI-IoT
- The Physical Data Curation Problem

Data Curation

- Taxonomy of Data Curation Techniques
- Filtering Methods (OpenMAE)
- Coreset Selection Methods (Ju, SimCore, ELFS)

Data Augmentation

- Distribution Mismatch in IoT Systems
- Data Augmentation for Sensor Signals
- Generative Augmentation (NeurIPS 2024, SudokuSens)

Wrap Up

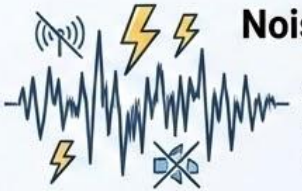
- End-to-End AI-IoT Data Curation Pipeline
- Open Research Challenges

Motivation: Why Data Matters in AI for IoT

AI models need massive, high-quality data. IoT sensors produce vast amounts of complex, hard-to-interpret signals.

Therefore, data curation and augmentation are critical for robust AI-for-IoT pipelines.

Unique Challenges of IoT Data

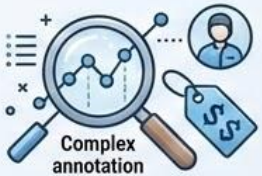


Noisy & Disturbed Signals

Sensor signals are noisy and may contain environmental disturbances.

Sparse & Inactive Periods

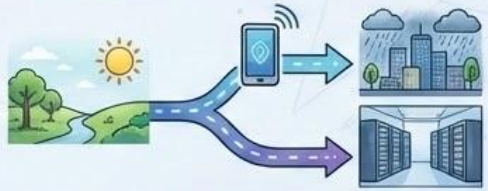
IoT signals often contain long periods of inactivity or background noise.



Expensive Labeling

Labeling sensor data requires domain expertise and expensive experiments.

Dynamic Distribution Shift



Sensor data distribution changes depending on environment (terrain, weather, device placement).

Limited Dataset Size

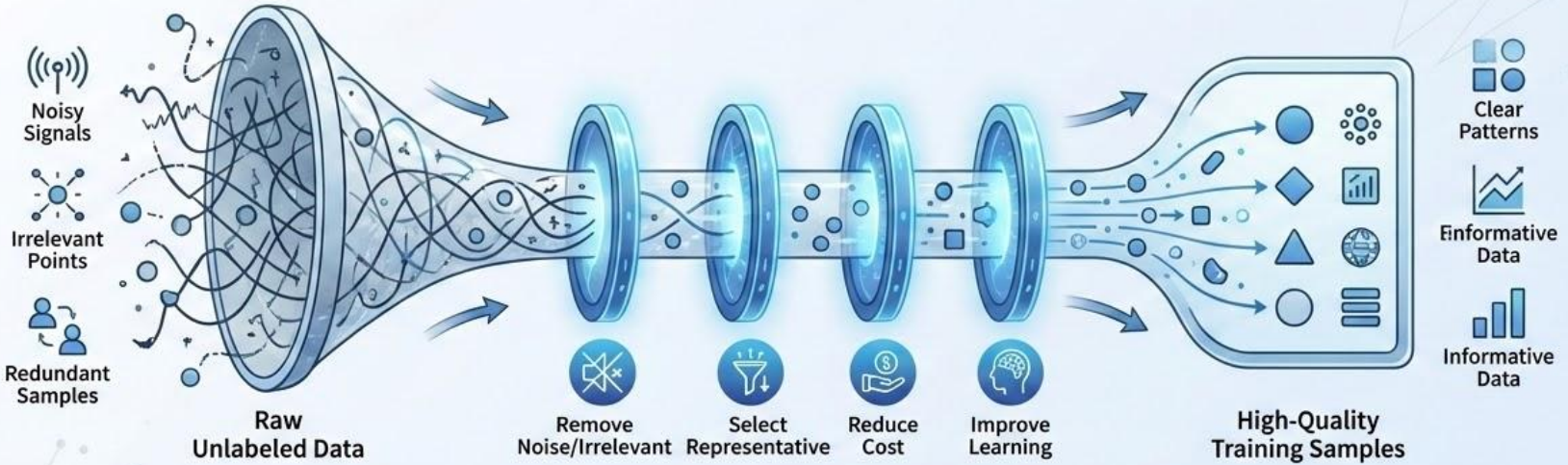
IoT datasets are usually smaller than datasets in computer vision or NLP.

⚠ These challenges make raw IoT data **unsuitable for direct training.**

Aspect	Data Curation	Data Augmentation
Definition	Process of organizing, cleaning, and managing existing data to ensure quality and usability.	Technique of generating new training samples by modifying existing data.
Main Goal	Improve data quality and reliability.	Increase data diversity and quantity.
Dataset Size	Often reduces or refines the dataset by removing errors or duplicates.	Expands the dataset by creating additional samples.
Methods Used	Data cleaning, deduplication, correcting labels, filtering irrelevant data.	Transformations such as rotation, flipping, cropping (images), synonym replacement (text), noise addition (audio).
Impact on Model	Leads to more accurate and reliable training data.	Helps improve generalization and reduce overfitting.

The Data Curation Problem

The **process of selecting high-quality training samples** from large unlabeled datasets.

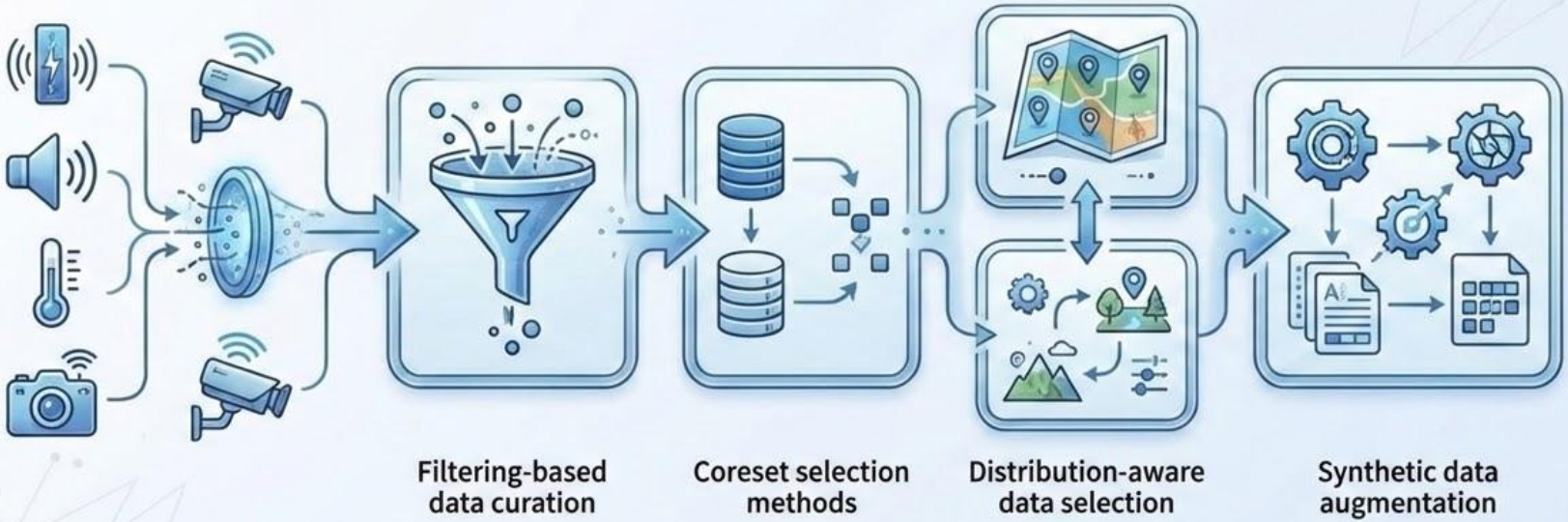


The process of selectiting high-quality training samples from the model.

Without proper curation, adding more data can actually reduce model accuracy.

Taxonomy of Physical Data Curation Techniques

Main categories used in AI-for-IoT:



These techniques attempt to **improve dataset quality** before training AI models.

Filtering-Based Data Curation

Common signals removed:

- Background noise
- Idle sensor periods
- Corrupted sensor measurements
- Irrelevant environmental signals

This ensures the training data contains meaningful events rather than noise.

OpenMAE

Paper: OpenMAE – Efficient Masked Autoencoder for Vibration Sensing

Key Idea & Challenge

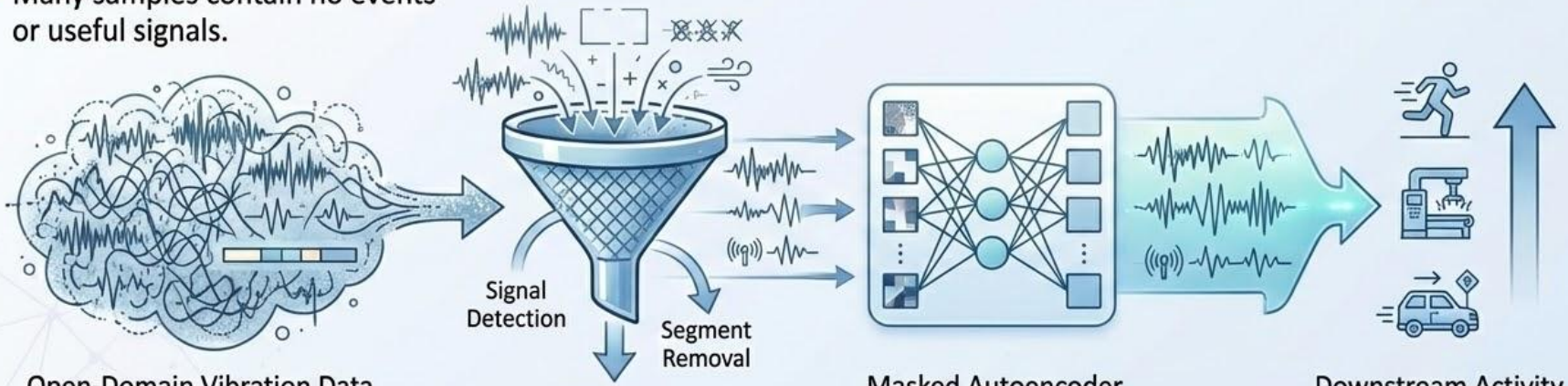
- Use large open-domain vibration datasets for self-supervised learning.
- However, naive use decreases performance.
- Many samples contain no events or useful signals.

Solution (The OpenMAE Approach)

- Detect informative signals & remove inactive segments.
- Use filtered data for training masked autoencoders.

Result (Significant Improvement)

- Significant improvement in downstream activity recognition tasks.



OpenMAE Data Curation Pipeline

1. Collect Open-Domain Data

Collect open-domain vibration signals from sensor networks.

2. Identify Informative Samples

Identify informative samples using unsupervised filtering.

3. Remove Inactive/Noisy Samples

Remove inactive or noisy samples.

4. Combine with In-Domain Data

Combine curated open-domain data with in-domain training data.

5. Train Masked Autoencoder (MAE)

Train masked autoencoder (MAE) models for representation learning.

Coreset Selection

Full Dataset
(Large, Noisy, Redundant)

Coreset
(Representative Subset)

Faster Training
Accelerates model training times.

Reduced Storage Requirements
Minimizes data storage needs.

Improved Data Quality
Enhances dataset integrity.

Better Generalization
Improves performance on diverse tasks.

SimCore

Paper: SimCore - Coreset Sampling from Open-Set for Self-Supervised Learning

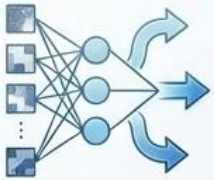
Problem

Open datasets contain samples **unrelated** to the target domain.

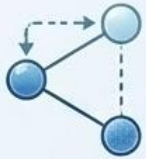
Key Idea

Select samples that are **semantically closest** to the target dataset.

Method



Compute **latent embeddings** of unlabeled samples

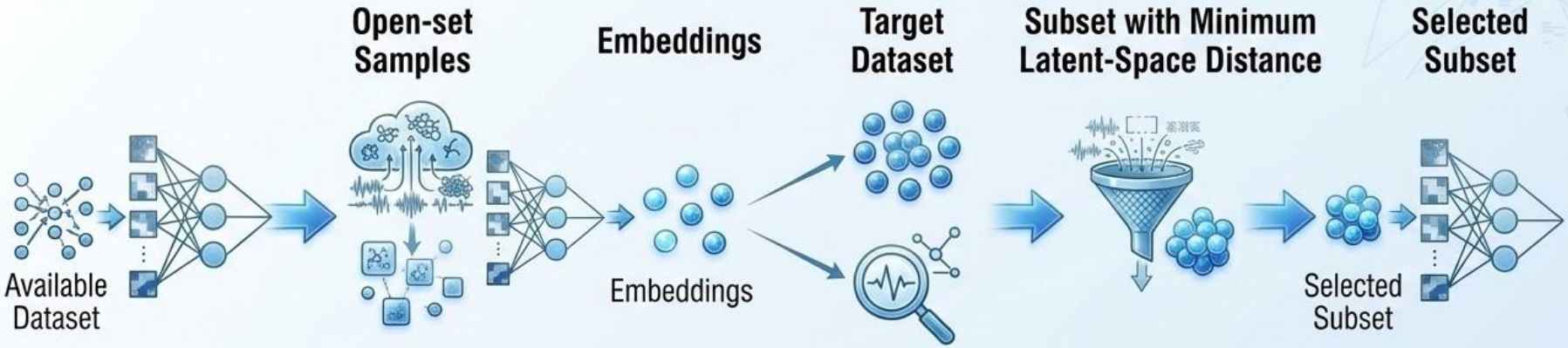


Measure similarity with target dataset

Select **most similar samples** as coreset

This improves representation learning for fine-grained tasks.

How SimCore Works



1. Train representation model on available dataset.

2. Extract embeddings from open-set samples.

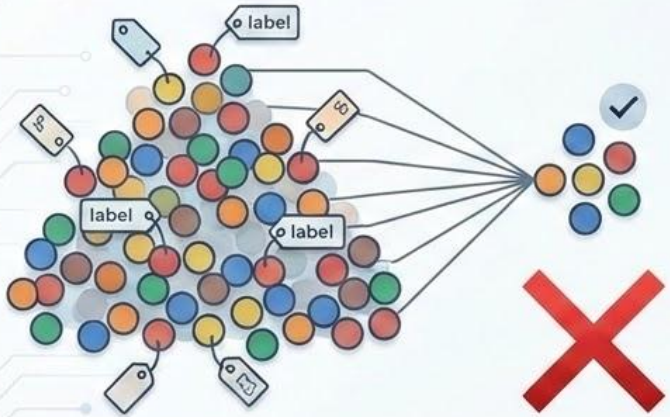
3. Measure similarity between open-set and target dataset.

4. Select subset with minimum latent-space distance.

5. Use selected samples for self-supervised training.

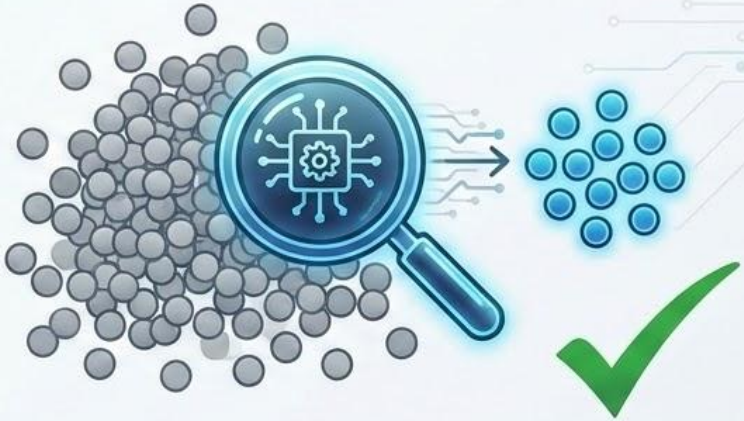
Label-Free Coreset Selection

Traditional Methods (Labeled Data)



Require labeled data for coreset selection.

New Research (Unlabeled Data)



Focus on methods working with unlabeled datasets.

ELFS

Paper: ELFS – Effective Label-Free Coreset Selection

Goal: Select representative training data without labels.

Generate pseudo-labels

Estimate data difficulty

Remove easy and noisy samples

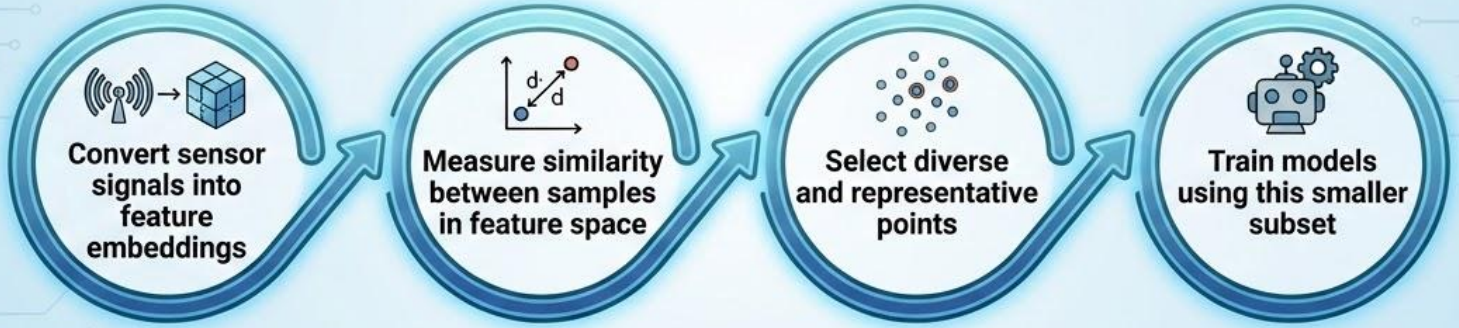
Result: Improved performance across multiple datasets.

Unsupervised Coreset Selection

Goal & Key Idea

- **Goal:** Select a small representative subset of large unlabeled IoT datasets.
- **Key Idea:** Choose samples that preserve the diversity and structure of the dataset without using labels.

How It Works



Impact

- Reduces dataset size and training cost
- Maintains important signal patterns
- Enables scalable learning for large IoT systems

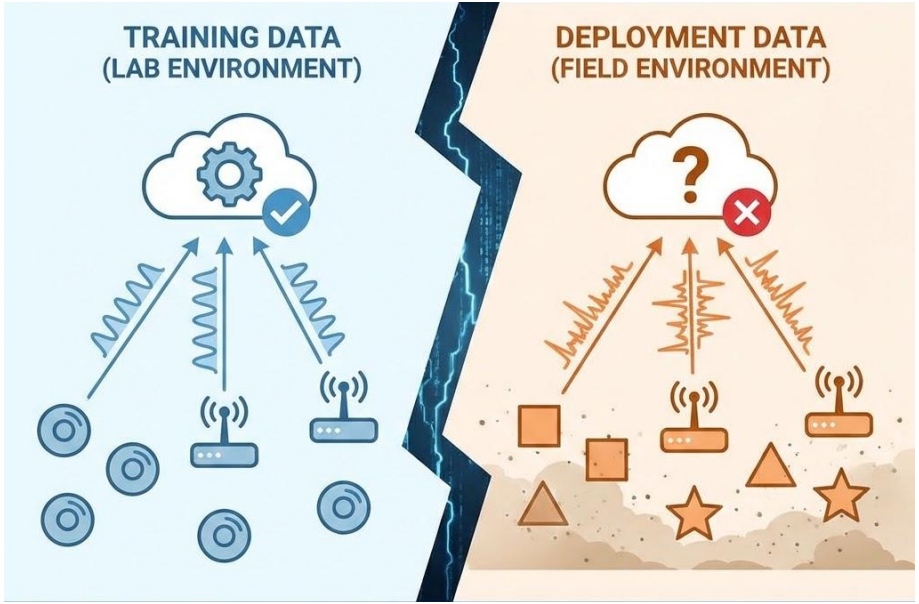
Distribution Mismatch in IoT

A major challenge in IoT learning: Training environment differs from deployment environment.

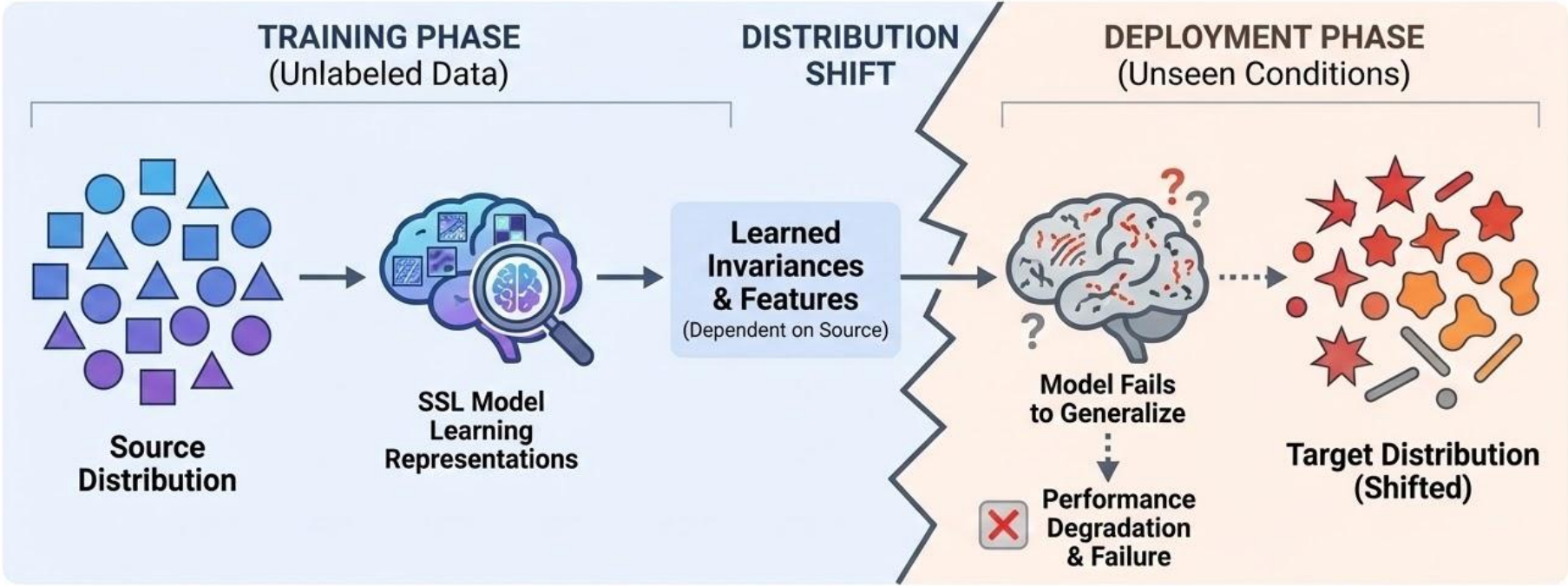
Examples:

- Sensors trained indoors but deployed outdoors
- Data collected on asphalt but deployed on gravel
- Device location changes signal patterns

Distribution mismatch reduces AI performance.



Why Distribution Shift Hurts SSL



Data Augmentation - Sensor Signals

- Data augmentation artificially increases dataset diversity.
- Goal: Expose AI models to many environmental conditions.

- Benefits:
  - Simulates environmental and operational variations
  - Improved robustness and better generalization
  - Reduced need for expensive data collection.

Traditional IoT Data Augmentation

Typical signal transformations include:

Time-domain transformations

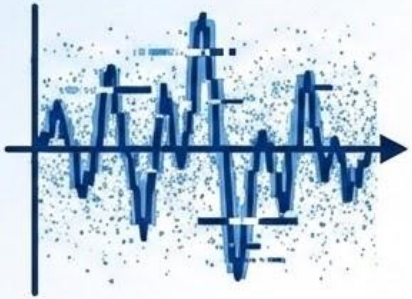
- Jittering 
- Scaling 
- Cropping 
- Rotation 

Frequency-domain transformations

- Spectral masking 
- Frequency warping 
- Noise injection 

Limitations of Traditional Augmentation

Problems with simple transformations:



May produce unrealistic sensor signals

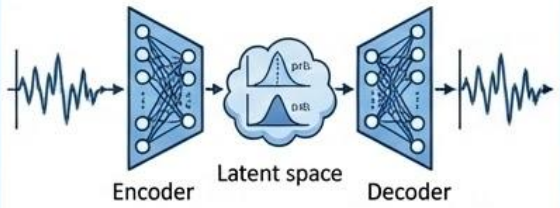
Struggles with complex environmental variations

Limited improvement in downstream performance

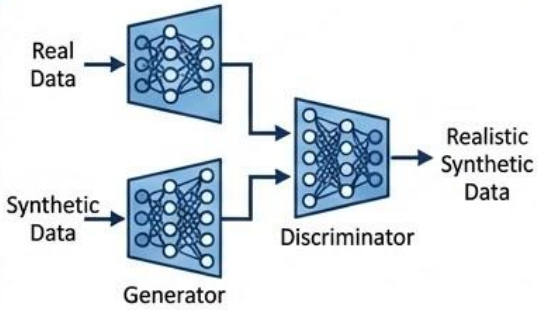
This motivates generative data augmentation methods.

Generative Data Augmentation

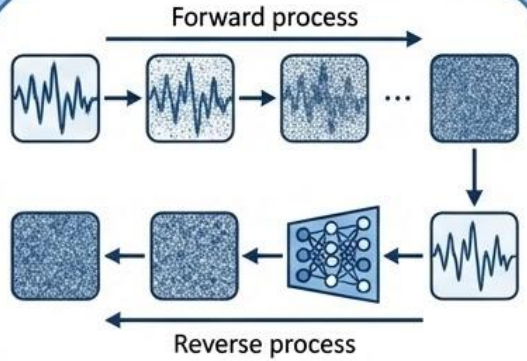
Modern AI approaches generate synthetic sensor data.



Variational Autoencoders (VAE)



Generative Adversarial Networks (GAN)

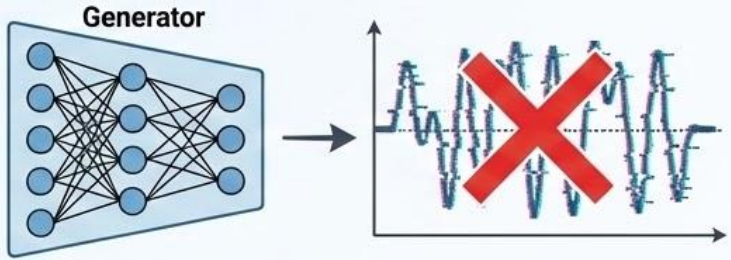


Diffusion models

These models learn the distribution of sensor signals and generate new realistic samples.

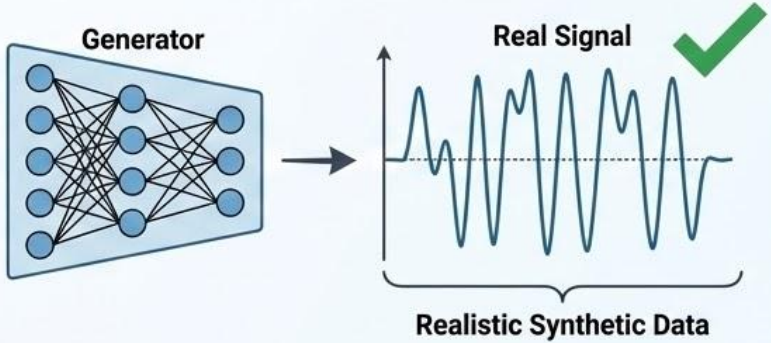
Example Paper: Fine-Grained Generative Augmentation

Problem: Generative models may produce unrealistic sensor signals



- ⚠ Generated signals may violate physical sensor properties
- ⚠ Models may learn incorrect signal patterns
- ⚠ Synthetic data may harm training

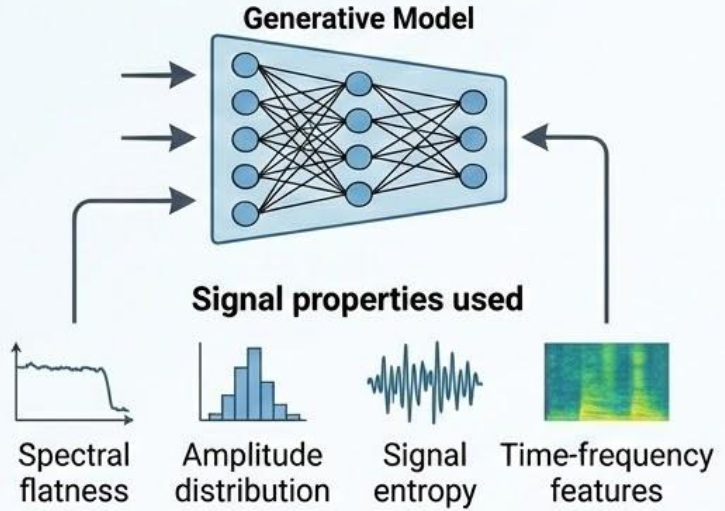
Goal: Generate synthetic sensor data that preserves real signal characteristics



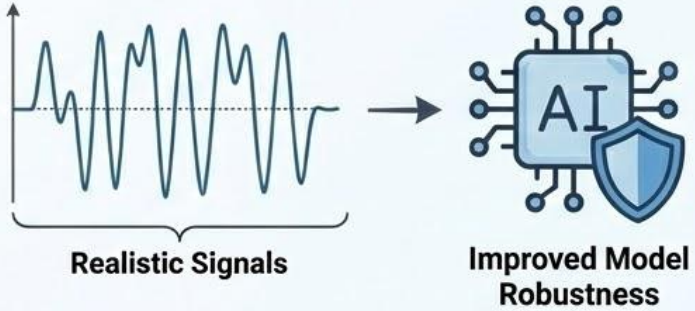
The goal is to ensure generated data respects physical constraints and preserves the characteristics of real sensor signals for effective training.

Controlled Generative Sensor Data

Key Idea: Guide generative models using signal statistics

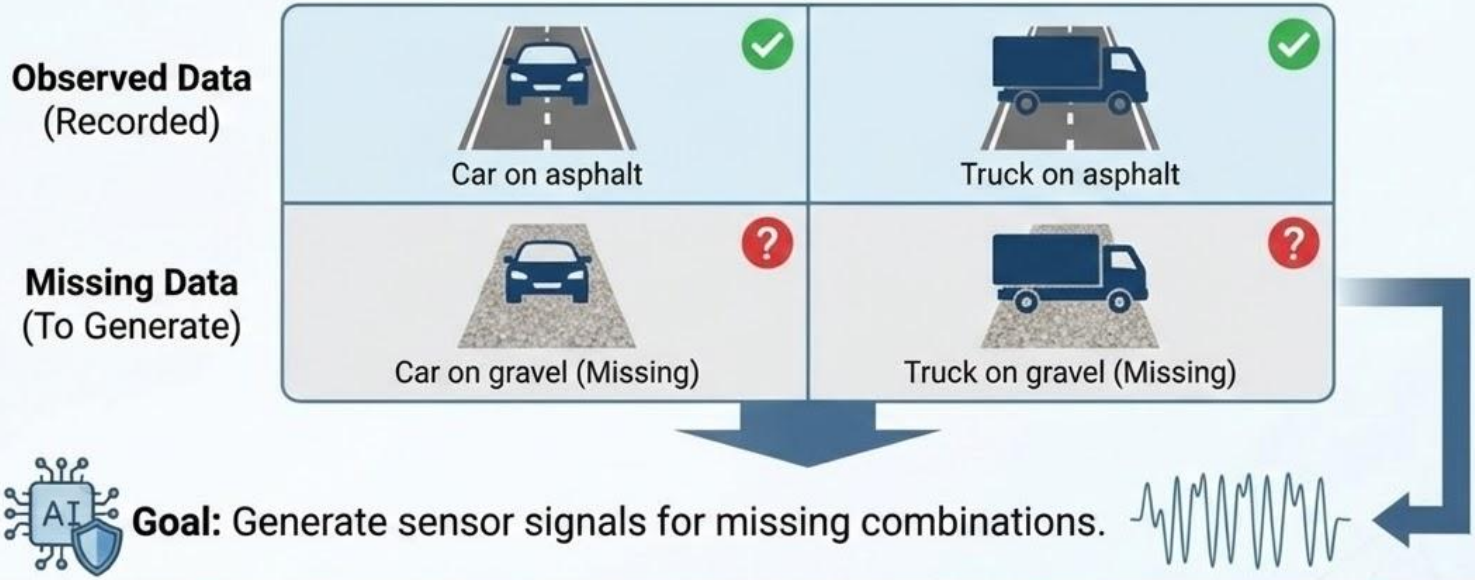


Result: Generated signals remain physically realistic and improve model robustness



Example Paper: SudokuSens

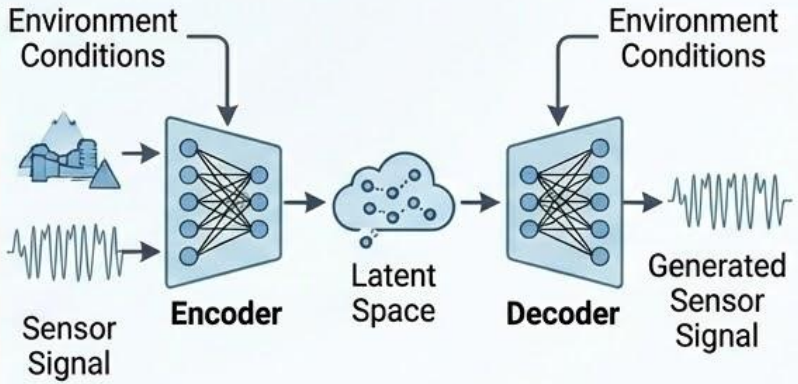
Problem & Goal: Addressing Missing IoT Data Combinations



Filling the gaps in environmental conditions.

SudokuSens Generative Framework

Method: Conditional Variational Autoencoder (CVAE)



Model learns:
P(sensor signal | environment conditions)

Process

Learn relationships between sensor signals and environmental factors

Generate signals for unseen combinations

Impact

Expands dataset coverage

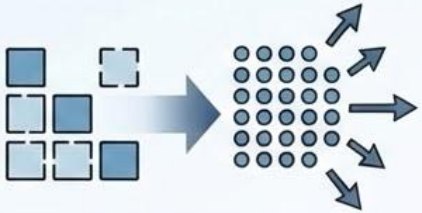
Reduces data collection cost

Improves model robustness

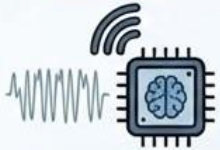
Benefits of Synthetic Data

Using generated sensor data provides:

Improved **dataset coverage**

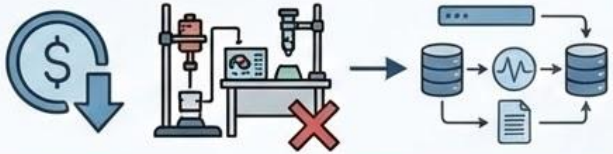


Increased **diversity** of training conditions



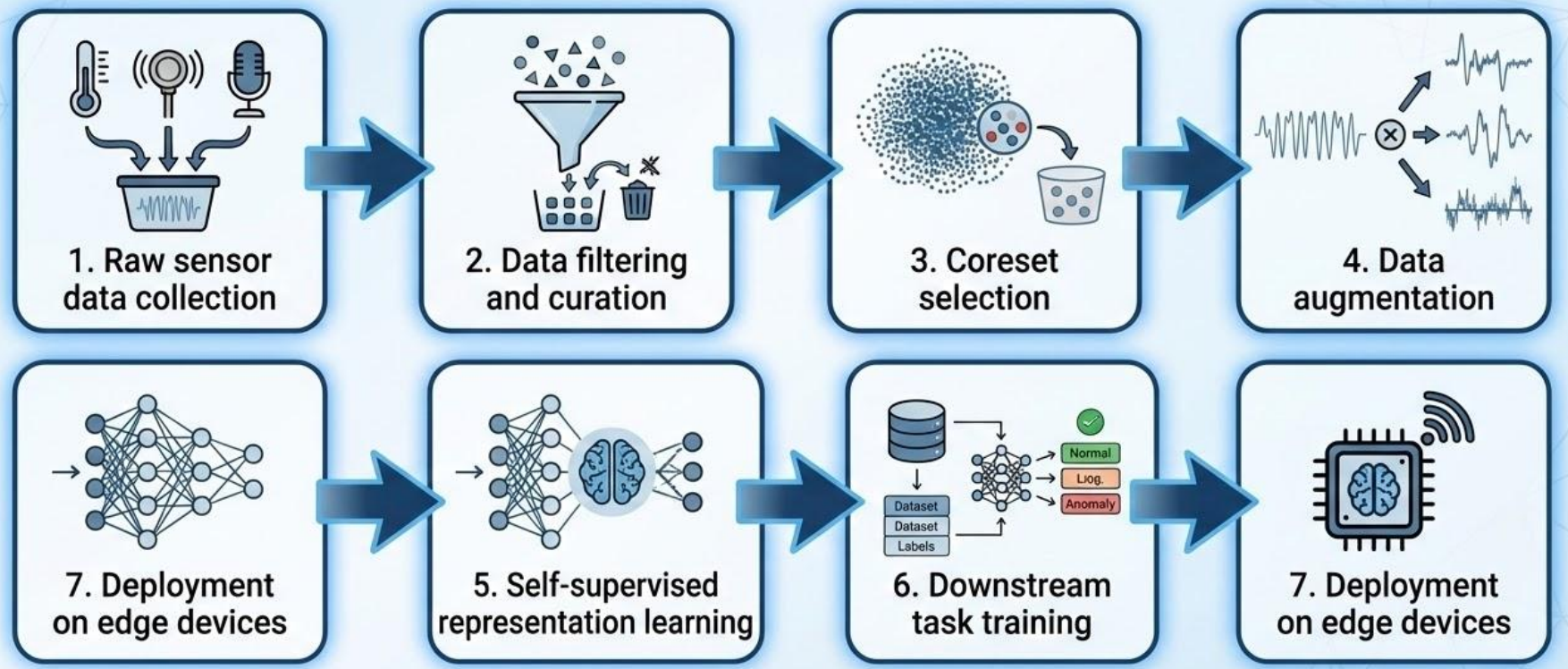
Generated Sensor Data

Reduced need for expensive physical experiments



Improved model robustness

Potential Pipeline for AI-IoT Data



Research Challenges and Future Research Directions

Research Challenges

- Automatic detection of useful sensor signals
- Scalable curation for large IoT networks
- Reliable synthetic data generation
- Cross–sensor domain adaptation
- Real–time learning on edge devices

Future Research Directions

- Foundation models for sensor signals
- Adaptive data filtering during training
- Multimodal sensor fusion
- Simulation environments for IoT training
- Automated dataset quality evaluation

Conclusion

- Key Takeaways:
  - Data quality is critical for AI in IoT.
  - Raw sensor data often contains noise and irrelevant signals.
  - Data curation techniques remove low-quality samples.
  - Coreset selection identifies representative subsets.
  - Data augmentation improves robustness.
  - Combining these approaches enables reliable AI-IoT systems.

References

Key papers discussed:

- SimCore – Coreset Sampling for Self-Supervised Learning
- ELFS – Label-Free Coreset Selection
- OpenMAE – Data Enrichment for Vibration Sensing
- SudokuSens – Generative Data Augmentation for IoT
- Fine-Grained Generative Data Augmentation for IoT sensing signals.
- Unsupervised Coreset Selection