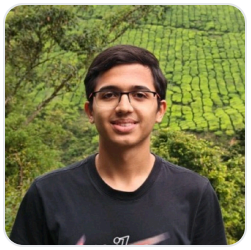
SUSTAINABILITY LAB

Project page

NILMBench2026

A deployment-aware benchmark for energy disaggregation

BEST PAPER CANDIDATE



Aayush Kuloor*
Undergraduate Sophomore
IIT Gandhinagar
aayush.kuloor@iitgn.ac.in

Anurag Singh*
Undergraduate Sophomore
IIT Gandhinagar
anurag.s@iitgn.ac.in

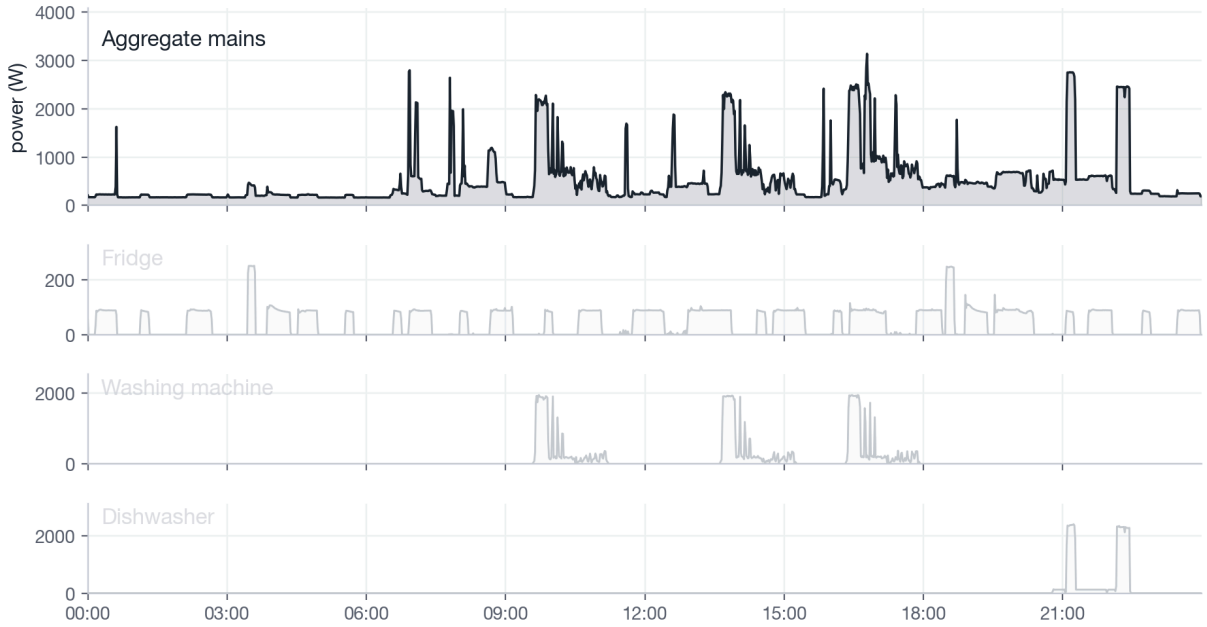
Harsh Dhru*
Undergraduate Sophomore
IIT Gandhinagar
harsh.dhru@iitgn.ac.in

Nipun Batra
Faculty advisor
IIT Gandhinagar
nipun.batra@iitgn.ac.in

ACM BuildSys 2026 · Banff, Canada

MOTIVATION

NILM turns one meter into appliance-level estimates

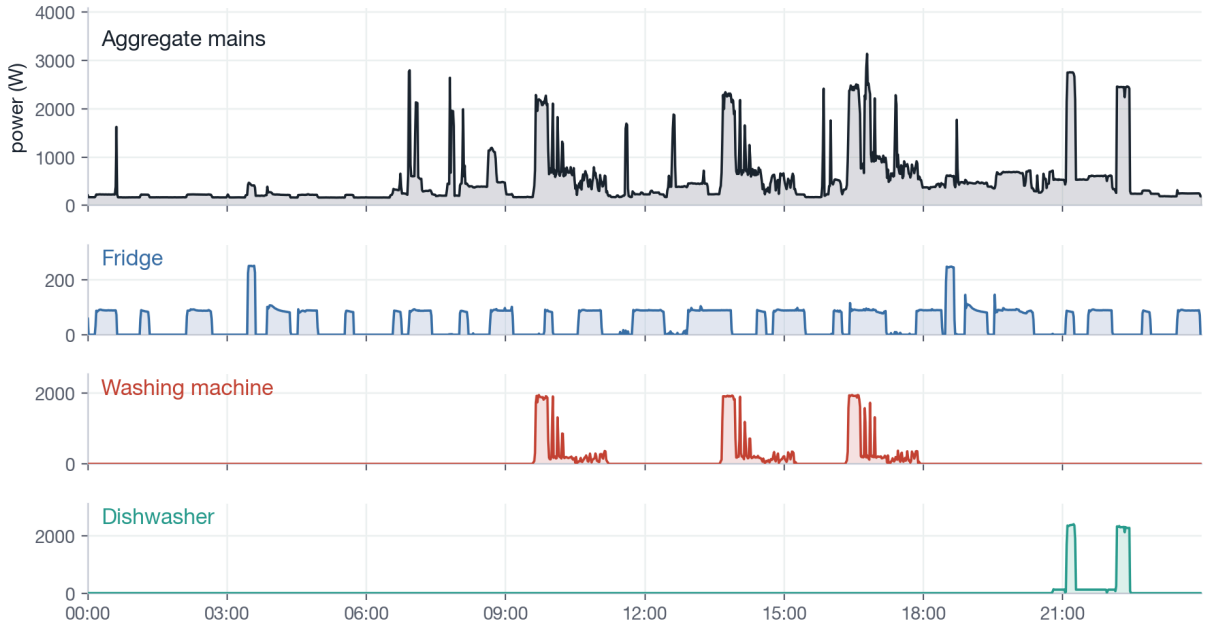


- **One** aggregate smart-meter signal — all that is measured at deployment
- Disaggregates into appliance-level power — no per-appliance sensors
- Feedback can cut household use by up to 15% (Froehlich & Patel 2011; Darby 2006)
- Signatures vary across homes and devices

Real data · UK-DALE house 1
time (x) vs power in W (y)

MOTIVATION

NILM turns one meter into appliance-level estimates

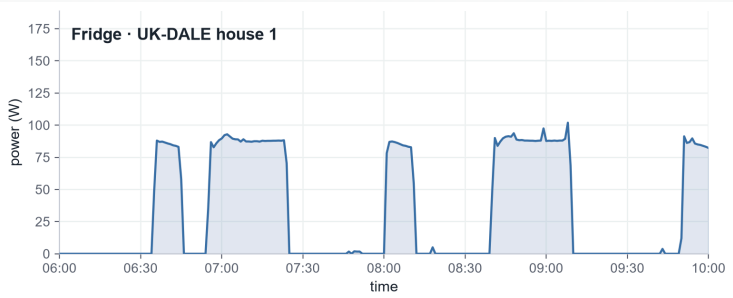


- **One** aggregate smart-meter signal — all that is measured at deployment
- Disaggregates into appliance-level power — no per-appliance sensors
- Feedback can cut household use by up to 15% (Froehlich & Patel 2011; Darby 2006)
- Signatures vary across homes and devices

Real data · UK-DALE house 1
time (x) vs power in W (y)

MOTIVATION

Appliance signatures differ by load type



Fridge · periodic

- Compressor cycles ~80–120 W
- **Defrost spikes** on longer horizon
- Always-on, easy to detect

Washing machine · multi-stage

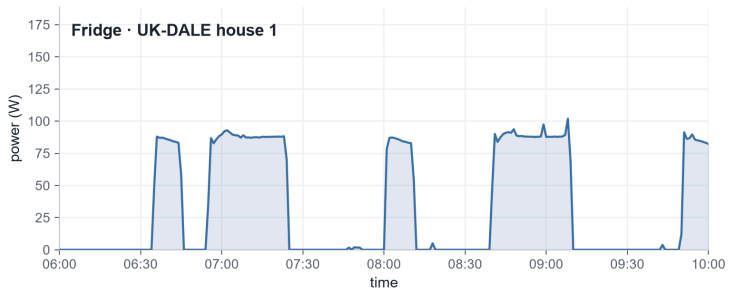
- **Heat → agitate → spin**
- Long, variable duration
- Many sub-states

Dishwasher · sparse / bursty

- High-power heating bursts
- Long idle gaps
- **MAE-deceptive** when mostly off

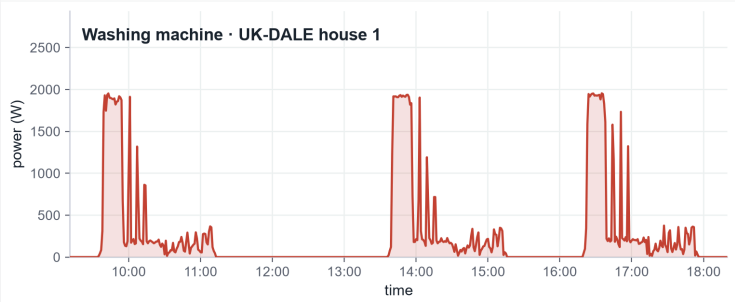
MOTIVATION

Appliance signatures differ by load type



Fridge · periodic

- Compressor cycles ~80–120 W
- **Defrost spikes** on longer horizon
- Always-on, easy to detect



Washing machine · multi-stage

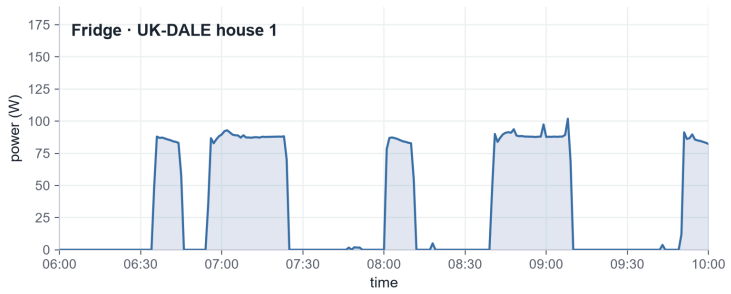
- **Heat → agitate → spin**
- Long, variable duration
- Many sub-states

Dishwasher · sparse / bursty

- High-power heating bursts
- Long idle gaps
- **MAE-deceptive** when mostly off

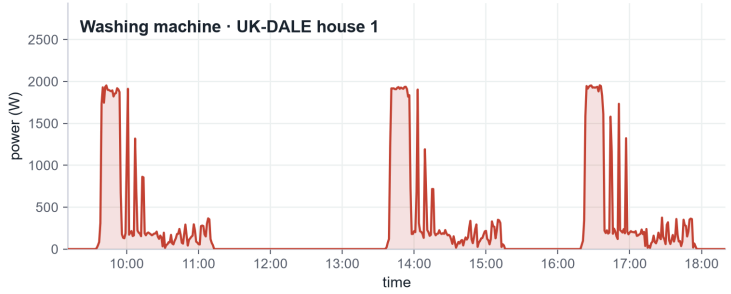
MOTIVATION

Appliance signatures differ by load type



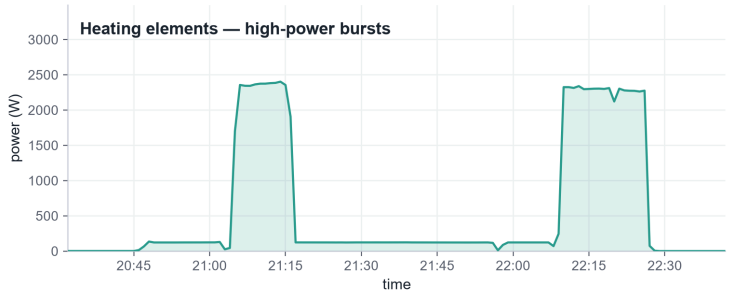
Fridge · periodic

- Compressor cycles ~80–120 W
- **Defrost spikes** on longer horizon
- Always-on, easy to detect



Washing machine · multi-stage

- **Heat → agitate → spin**
- Long, variable duration
- Many sub-states



Dishwasher · sparse / bursty

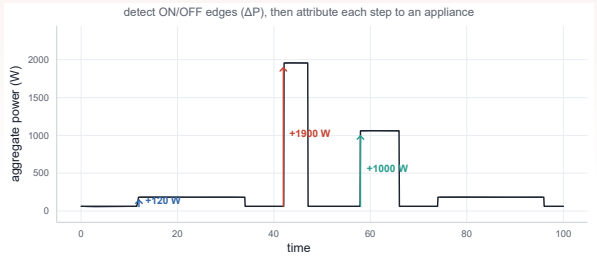
- High-power heating bursts
- Long idle gaps
- **MAE-deceptive** when mostly off

BACKGROUND

NILM methods evolved, but evaluation did not

1980s–90s

Combinatorial



- Edge matching
- Breaks on variable loads

2000s

Probabilistic

- Hidden appliance states
- Poor scaling

2015+

Deep learning

- Seq2Point CNNs
- Strong same-home fit

2020+

Transformers

- Long-range context
- Higher compute

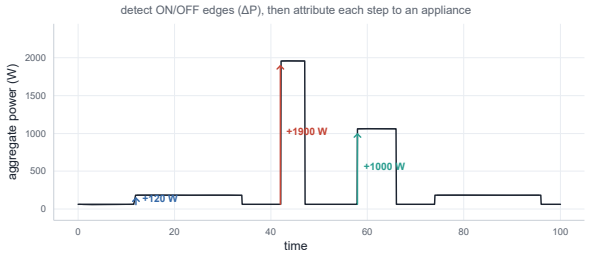
Models changed — evaluation did not. Same-home accuracy became the default; cross-building and cross-dataset tests lagged behind.

BACKGROUND

NILM methods evolved, but evaluation did not

1980s–90s

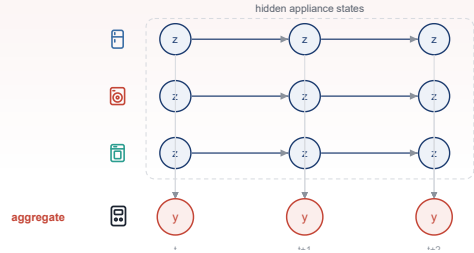
Combinatorial



- Edge matching
- Breaks on variable loads

2000s

Probabilistic



- Hidden appliance states
- Poor scaling

2015+

Deep learning

- Seq2Point CNNs
- Strong same-home fit

2020+

Transformers

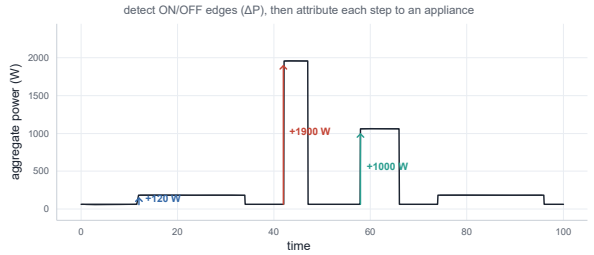
- Long-range context
- Higher compute

Models changed — evaluation did not. Same-home accuracy became the default; cross-building and cross-dataset tests lagged behind.

NILM methods evolved, but evaluation did not

1980s–90s

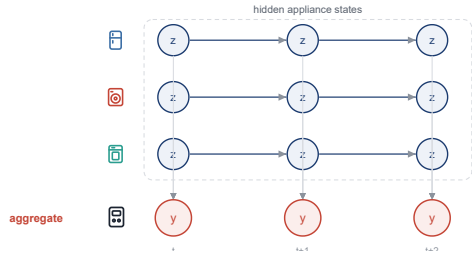
Combinatorial



- Edge matching
- Breaks on variable loads

2000s

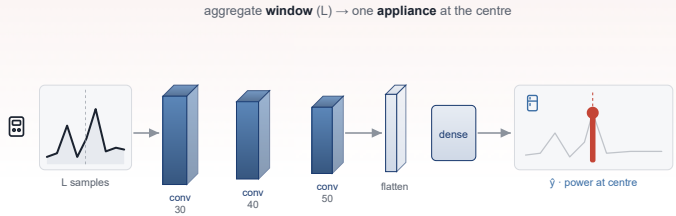
Probabilistic



- Hidden appliance states
- Poor scaling

2015+

Deep learning



- Seq2Point CNNs
- Strong same-home fit

2020+

Transformers

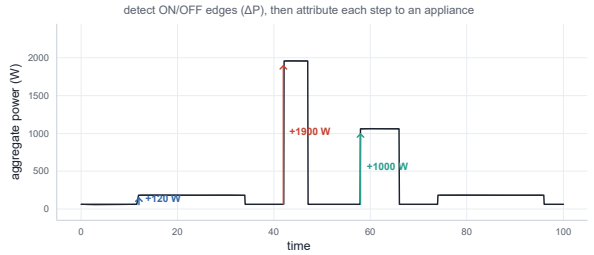
- Long-range context
- Higher compute

Models changed — evaluation did not. Same-home accuracy became the default; cross-building and cross-dataset tests lagged behind.

NILM methods evolved, but evaluation did not

1980s–90s

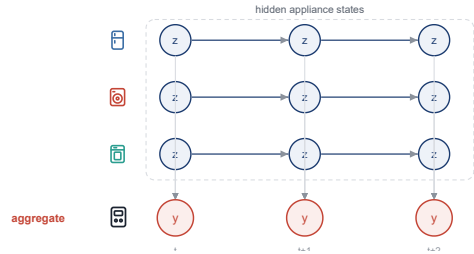
Combinatorial



- Edge matching
- Breaks on variable loads

2000s

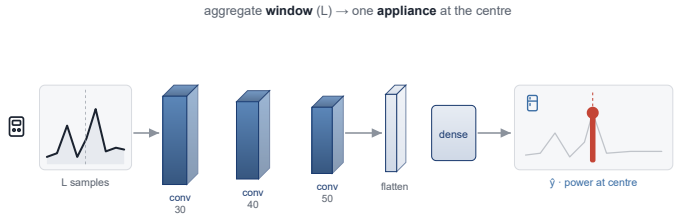
Probabilistic



- Hidden appliance states
- Poor scaling

2015+

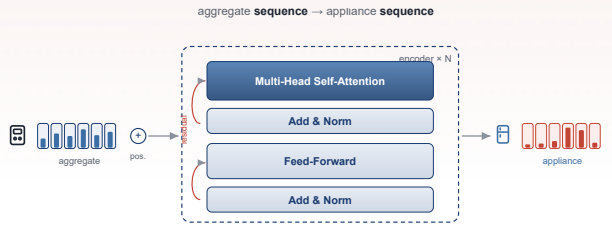
Deep learning



- Seq2Point CNNs
- Strong same-home fit

2020+

Transformers



- Long-range context
- Higher compute

Models changed — evaluation did not. Same-home accuracy became the default; cross-building and cross-dataset tests lagged behind.

WHY A NEW BENCHMARK

Existing benchmarks missed deployment readiness

CAPABILITY	NILMTK E-ENERGY 2014	NILMTK-CONTRIB BUILDSYS 2019	NILMBENCH2026 BUILDSYS 2026
Models	2	9	**16**
Resolutions	variable	1-min	**1-min & 15-min**
Efficiency (FLOPs / time)	—	—	**yes**
Cross-building	—	yes	yes
Cross-dataset	—	—	**yes**
Stack	Python 2.7	TF 1.x	**PyTorch + Docker + uv**

First benchmark to jointly score **efficiency**, **multi-resolution** utility settings, and **cross-domain transfer**.

Datasets cover countries, buildings, and appliance types

DATASET	COUNTRY	BUILDINGS	DURATION	APPLIANCES
REDD	USA — 110 V	6	3–19 days	10–20
UK-DALE	UK — 230 V	5	655 days	5–54
REFIT	UK — 230 V	20	2 years	9–21

Periodic
Fridge

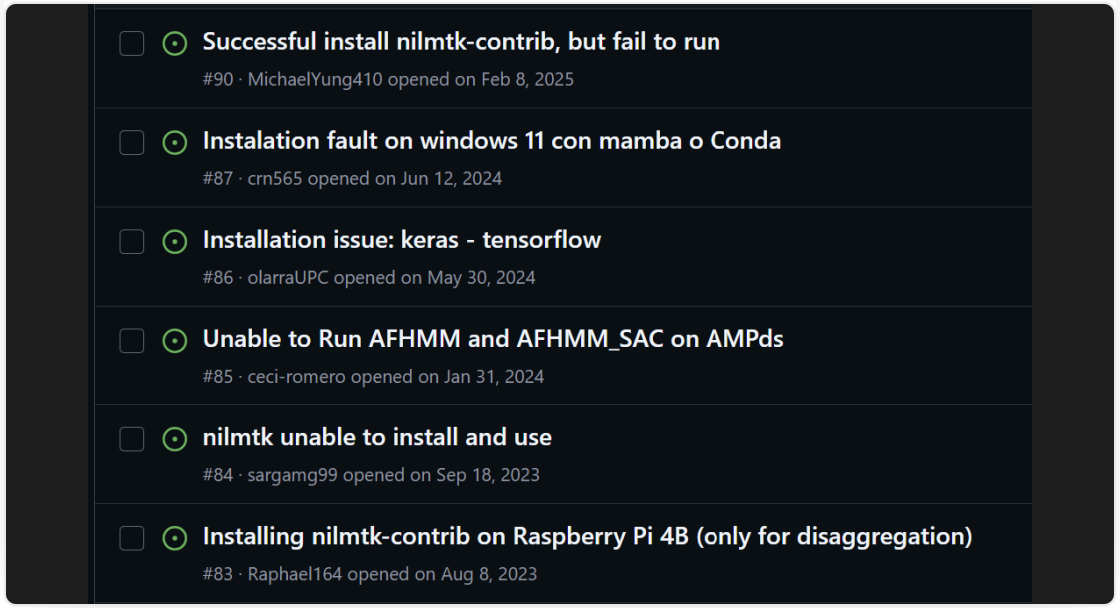
Sparse / bursty
Microwave, kettle, dishwasher

Multi-stage
Washing machine, dishwasher

Design goal 1 — Reproducible ML tooling

MOTIVATION

- Legacy NILMTK stacks were brittle — Python 2.7, TensorFlow 1.x, broken installs
- Hard to reproduce published numbers or compare new models fairly



WHAT WE DO

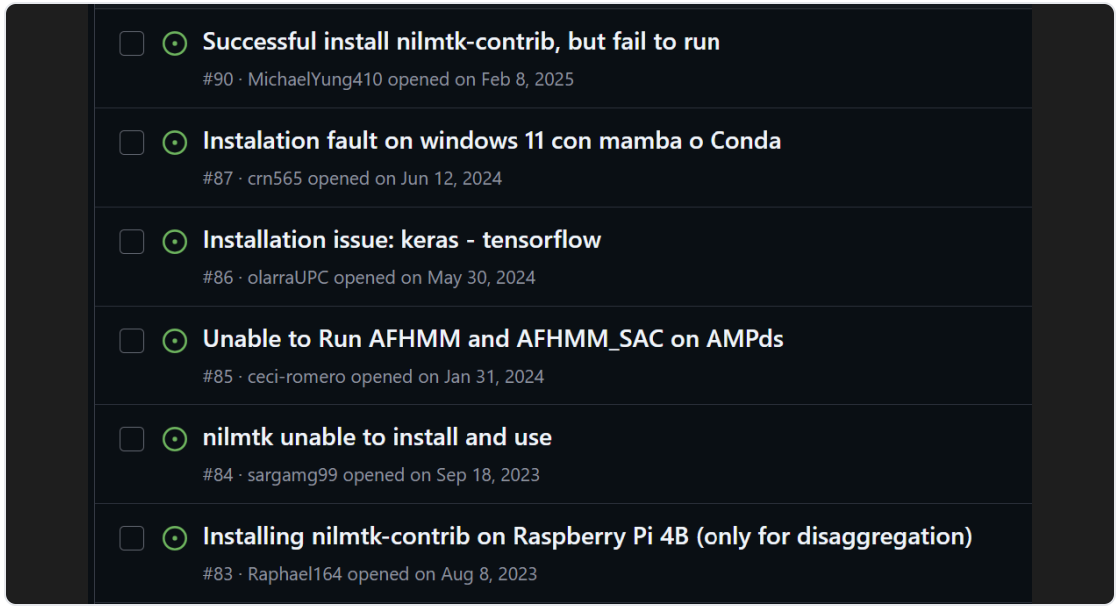
- Re-implement legacy models in **PyTorch** under a unified API
- Ship sealed runs with **Docker** and **uv**
- Extend the NILMTK Experiment API so every model follows the same train / test protocol

Goal: a new architecture should take **minutes to benchmark**, not days to debug the install.

Design goal 1 — Reproducible ML tooling

MOTIVATION

- Legacy NILMTK stacks were brittle — Python 2.7, TensorFlow 1.x, broken installs
- Hard to reproduce published numbers or compare new models fairly



WHAT WE DO

- Re-implement legacy models in **PyTorch** under a unified API
- Ship sealed runs with **Docker** and **uv**
- Extend the NILMTK Experiment API so every model follows the same train / test protocol

Goal: a new architecture should take **minutes to benchmark**, not days to debug the install.

Design goals 2 & 3 — What we evaluate

2 · Real-world relevance

MOTIVATION

- NILM at **user-feedback** and **utility-scale** timescales
- Deployment = new homes, regions, grid conditions

WHAT WE DO

- **1-min** & **15-min** on REDD, UK-DALE, REFIT
- Sealed **cross-building** & **cross-dataset** splits
- Regression **MAE** + event **F1** for sparse loads

3 · Comprehensive coverage

MOTIVATION

- Prior work: few models, **accuracy only**
- Deployment needs efficiency & event detection too

WHAT WE DO

- **16 models** · recurrent, conv, attention, hybrid
- **MAE**, **F1**, **parameters**, **FLOPs** — one protocol
- First to combine multi-resolution, efficiency & transfer

Design goals 2 & 3 — What we evaluate

2 · Real-world relevance

MOTIVATION

- NILM at **user-feedback** and **utility-scale** timescales
- Deployment = new homes, regions, grid conditions

WHAT WE DO

- **1-min** & **15-min** on REDD, UK-DALE, REFIT
- Sealed **cross-building** & **cross-dataset** splits
- Regression **MAE** + event **F1** for sparse loads

3 · Comprehensive coverage

MOTIVATION

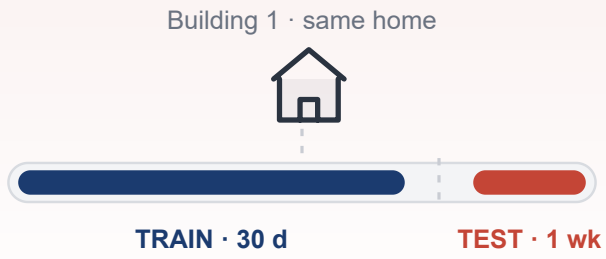
- Prior work: few models, **accuracy only**
- Deployment needs efficiency & event detection too

WHAT WE DO

- **16 models** · recurrent, conv, attention, hybrid
- **MAE**, **F1**, **parameters**, **FLOPs** — one protocol
- First to combine multi-resolution, efficiency & transfer

THE BENCHMARK

Three tasks separate fitting from deployment



T1 · Same building

SETUP Disjoint time windows, one home

SPLIT Train 30 day (B1) → test 1 week (B1)

ROLE Upper-bound sanity check

T2 · New building

SETUP Train on homes, test on unseen home

SPLIT UK-DALE B1,B2 → B4 · REDD B1–B3 → B6

ROLE Cross-building generalization

T3 · New dataset

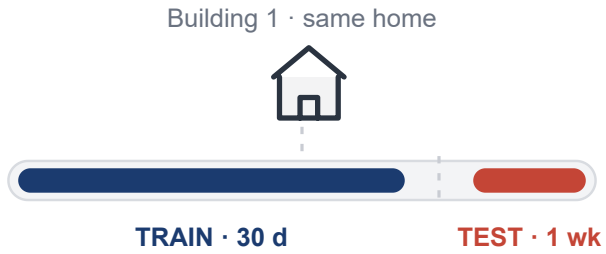
SETUP Train in one country, test in another

SPLIT REDD (USA, 110 V) ⇄ REFIT (UK, 230 V)

ROLE Zero-shot domain shift

THE BENCHMARK

Three tasks separate fitting from deployment

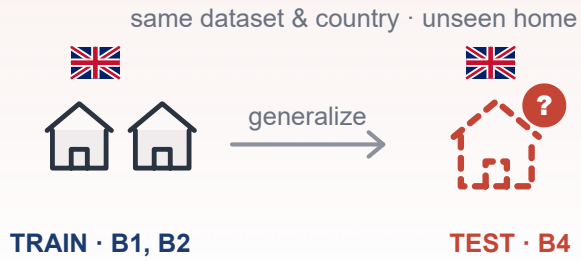


T1 · Same building

SETUP Disjoint time windows, one home

SPLIT Train 30 day (B1) → test 1 week (B1)

ROLE Upper-bound sanity check



T2 · New building

SETUP Train on homes, test on unseen home

SPLIT UK-DALE B1,B2 → B4 · REDD B1–B3 → B6

ROLE Cross-building generalization

T3 · New dataset

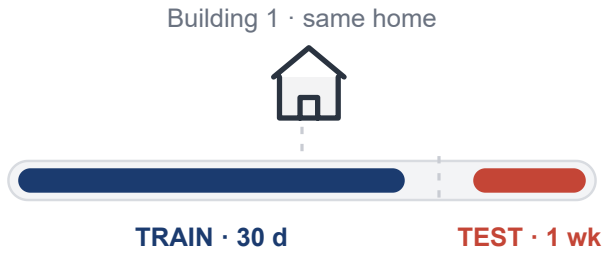
SETUP Train in one country, test in another

SPLIT REDD (USA, 110 V) ⇄ REFIT (UK, 230 V)

ROLE Zero-shot domain shift

THE BENCHMARK

Three tasks separate fitting from deployment

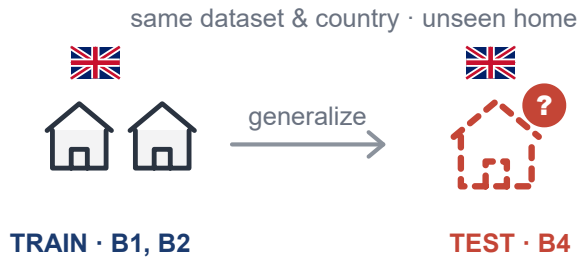


T1 · Same building

SETUP Disjoint time windows, one home

SPLIT Train 30 day (B1) → test 1 week (B1)

ROLE Upper-bound sanity check

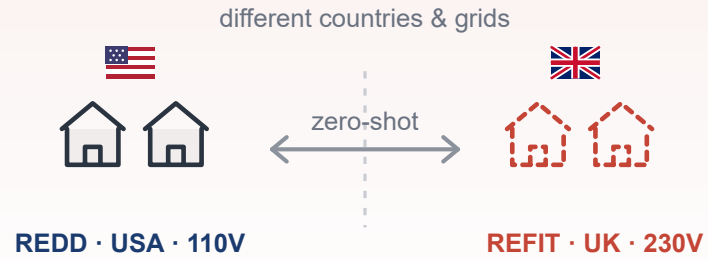


T2 · New building

SETUP Train on homes, test on unseen home

SPLIT UK-DALE B1,B2 → B4 · REDD B1–B3 → B6

ROLE Cross-building generalization



T3 · New dataset

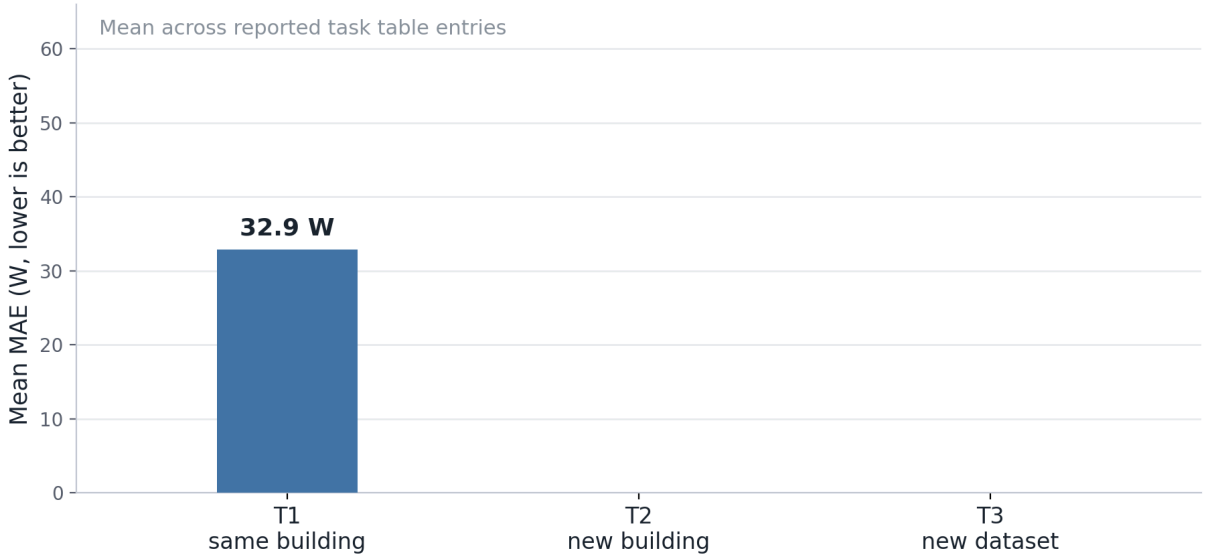
SETUP Train in one country, test in another

SPLIT REDD (USA, 110 V) ⇄ REFIT (UK, 230 V)

ROLE Zero-shot domain shift

RESULTS

Finding 1 — Domain shift drives the largest error jump

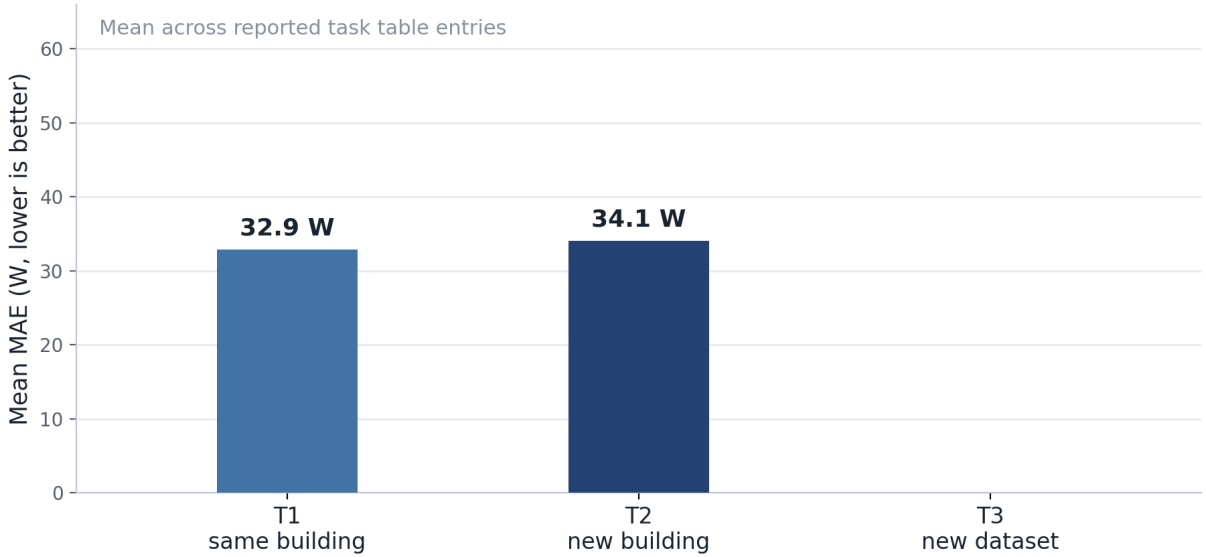


- Mean MAE rises from **T1 → T2 → T3** (UK-DALE / REDD→REFIT splits)
- Same-home accuracy is **not** a deployment metric
- Cross-dataset transfer is the hardest stress test

Use T1 as a sanity check; deployment claims need **T2 and T3**.

RESULTS

Finding 1 — Domain shift drives the largest error jump

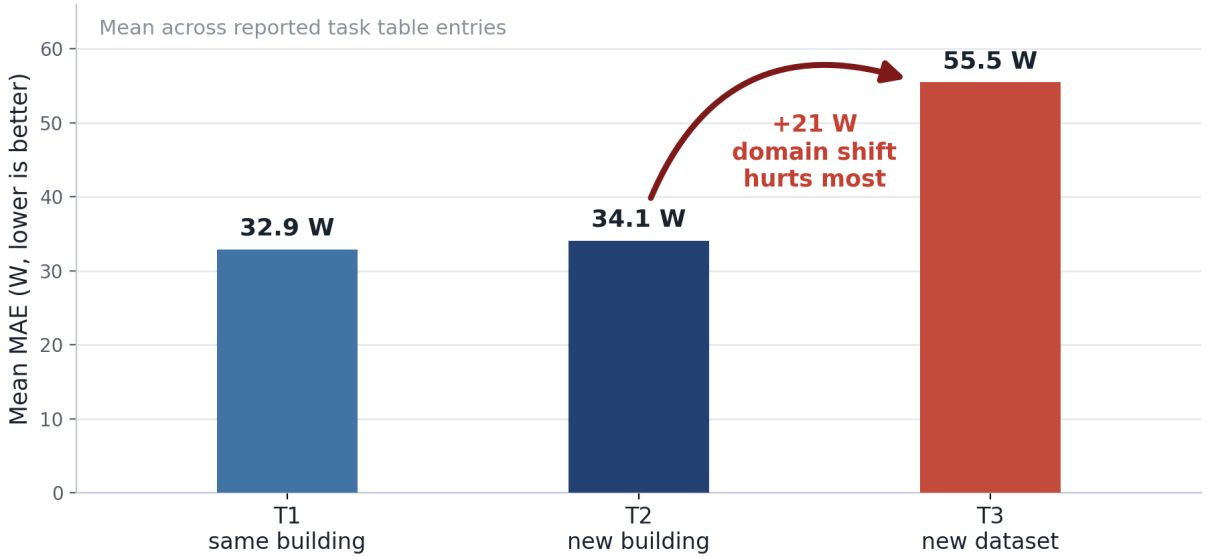


- Mean MAE rises from **T1 → T2 → T3** (UK-DALE / REDD→REFIT splits)
- Same-home accuracy is **not** a deployment metric
- Cross-dataset transfer is the hardest stress test

Use T1 as a sanity check; deployment claims need **T2 and T3**.

RESULTS

Finding 1 — Domain shift drives the largest error jump

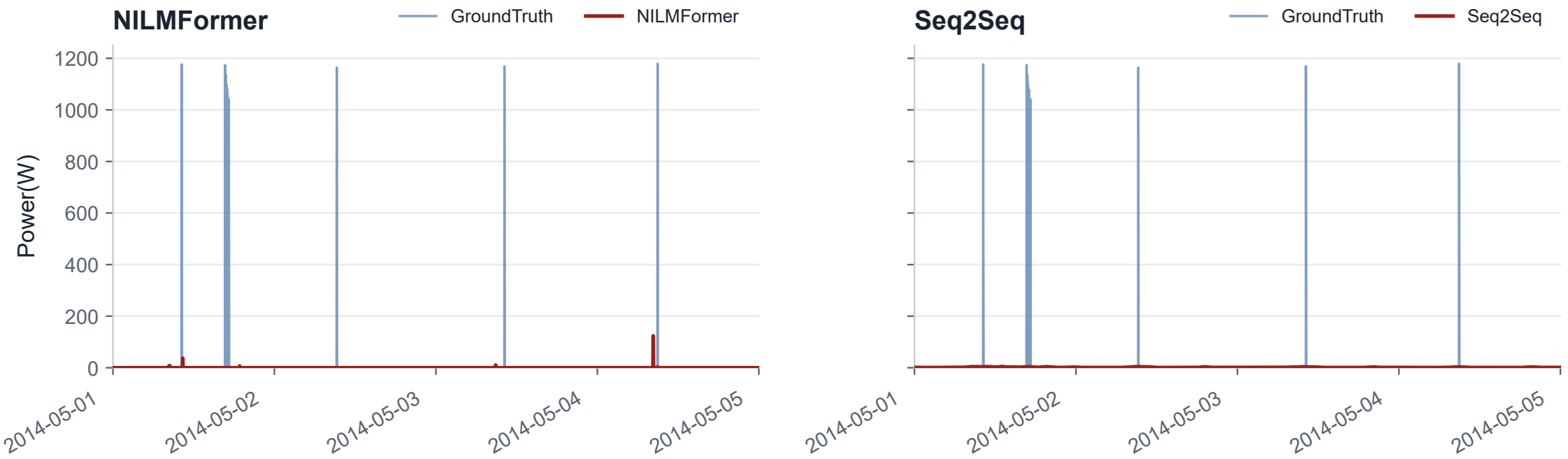


- Mean MAE rises from **T1 → T2 → T3** (UK-DALE / REDD→REFIT splits)
- Same-home accuracy is **not** a deployment metric
- Cross-dataset transfer is the hardest stress test

Use T1 as a sanity check; deployment claims need **T2 and T3**.

RESULTS

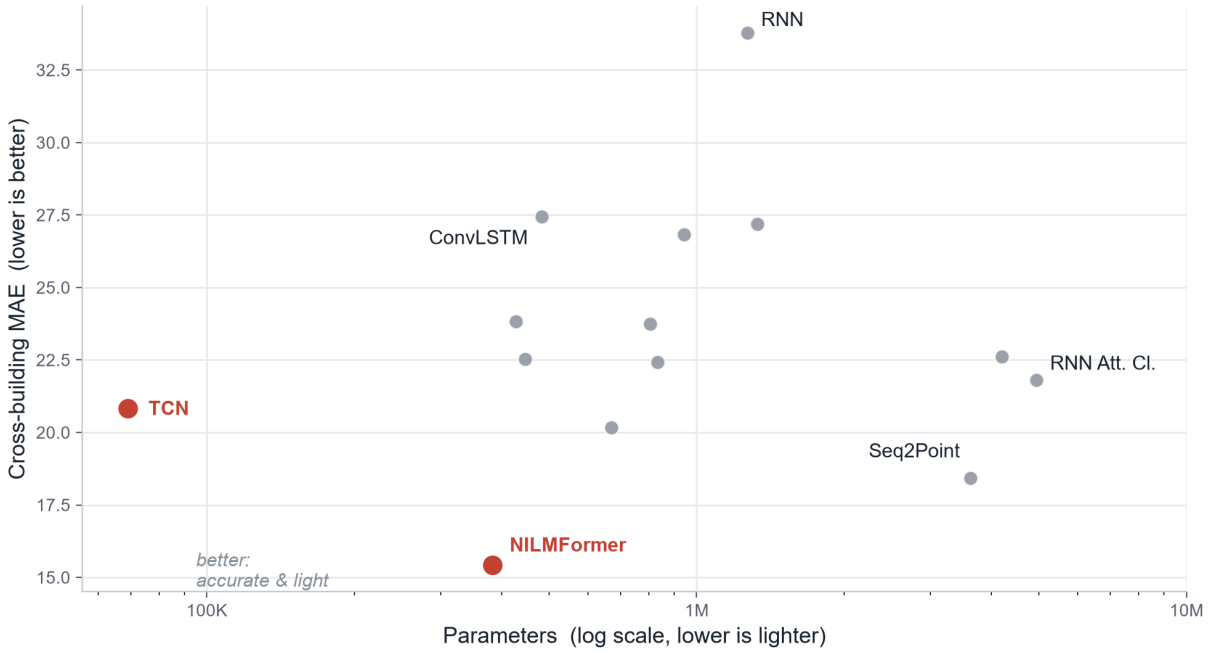
Finding 2 — MAE hides missed events



- Predict ≈ 0 during long idle periods → **deceptively low MAE**
- **NILMFormer & Seq2Seq** miss every microwave spike on REFIT T2
- Sparse, high-power loads need **F1**, not MAE alone

RESULTS

Finding 3 — More compute ≠ better



- Trade-off is **non-monotonic** — size alone does not predict deployment quality
- **TCN** (69K params) matches much heavier models on T2
- **NILMFormer** (383K) leads on hard generalization tasks
- **RNN Att. Cl.** (4.9M) is expensive **and** worse on average

Architectural inductive bias matters more than parameter count.

THE PLATFORM

A new model should take minutes to benchmark

```python
class PatchTSTDisaggregator(Disaggregator):
    def partial_fit(self, train_main, train_appliances):
        self.model.fit(train_main, train_appliances)

    def disaggregate_chunk(self, mains):
        return self.model.predict(mains)
```

- Wrap any PyTorch time-series model in the NILMTK-Contrib API
- Run sealed **T1 / T2 / T3** experiment configs
- Report **MAE**, **F1**, parameters, FLOPs, and runtime together

Code: **github.com/nilmtk/nilmtk-contrib** · Docker + uv for reproducible installs

Demo

Run one sealed benchmark task, inspect MAE / F1 / efficiency metrics, then return to the results.

SUMMARY & OUTLOOK

Summary and future directions

WHAT WE LEARNED

- Same-home accuracy is not enough
- MAE can reward missed sparse events
- More parameters do not guarantee better deployment

PROPOSED FUTURE DIRECTIONS

- Public, OOD-first leaderboard for the community
- **KDD-Cup-style** NILM challenge
- Agent-assisted model onboarding
- Domain adaptation for new homes

NILM OOD-first Leaderboard

Rank	Model	MAE (↓) T1 same home	MAE (↓) T2 new home	MAE (↓) T3 new dataset	F1 Score (↑) T1 same home	F1 Score (↑) T2 new home	F1 Score (↑) T3 new dataset	Params (M)	FLOPs (G)
1	**Model A**	**24.1**	**32.6**	**41.8**	**0.842**	**0.759**	**0.621**	**15.8**	**32.1**
2	Model B	22.7	34.8	48.7	0.830	0.731	0.572	28.3	55.4
3	Model C	23.9	36.2	49.5	0.821	0.722	0.560	9.7	18.9
4	Model D	25.8	38.9	53.6	0.797	0.689	0.512	42.6	81.3
5	Model E	27.1	41.0	56.8	0.781	0.653	0.489	6.3	12.7
6	Model F	26.4	42.3	59.7	0.768	0.628	0.456	18.2	29.6

Ranked by cross-dataset generalization — T1 same home → T2 new home → T3 new dataset

Project page · **sustainability-lab.github.io/nilmbench**