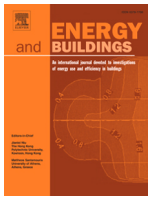
Energy & Buildings 336 (2025) 115632

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enbuild



Transfer learning on transformers for building energy consumption forecasting—A comparative study ⋆

Robert Spencer [a], Surangika Ranathunga [a,*], Mikael Boulic [b], Andries (Hennie) van Heerden [b], Teo Susnjak [a]

[a] School of Mathematical and Computational Sciences, Massey University, Auckland, 0632, New Zealand
[b] School of Built Environment, Massey University, Auckland, 0632, New Zealand

ARTICLE INFO

Keywords:
Building energy consumption forecasting
Transfer learning for time series
Transformer models for time series forecasting
Data-centric transfer learning strategies
PatchTST
Informer
Zero-shot learning
Model fine-tuning
Data scarcity

ABSTRACT

Energy consumption in buildings is steadily increasing, leading to higher carbon emissions. Predicting energy consumption is a key factor in addressing climate change. There has been a significant shift from traditional statistical models to advanced deep learning (DL) techniques for predicting energy use in buildings. However, data scarcity in newly constructed or poorly instrumented buildings limits the effectiveness of standard DL approaches. In this study, we investigate the application of six data-centric Transfer Learning (TL) strategies on three Transformer architectures—vanilla Transformer, Informer, and PatchTST—to enhance building energy consumption forecasting. Transformers, a relatively new DL framework, have demonstrated significant promise in various domains; yet, prior TL research has often focused on either a single data-centric strategy or older models such as Recurrent Neural Networks. Using 16 diverse datasets from the Building Data Genome Project 2, we conduct an extensive empirical analysis under varying feature spaces (e.g., recorded ambient weather) and building characteristics (e.g., dataset volume). Our experiments show that combining multiple source datasets under a zero-shot setup reduces the Mean Absolute Error (MAE) of the vanilla Transformer model by an average of 15.9 % for 24 h forecasts, compared to single-source baselines. Further fine-tuning these multi-source models with target-domain data yields an additional 3–5 % improvement. Notably, PatchTST outperforms the vanilla Transformer and Informer models. Overall, our results underscore the potential of combining Transformer architectures with TL techniques to enhance building energy consumption forecasting accuracy. However, careful selection of the TL strategy and attention to feature space compatibility are needed to maximize forecasting gains.

1. Introduction

1.1. Background

Driven by the need to mitigate the impact of climate change, there is a global focus on reducing carbon emissions in the building sector [1,2]. Previous studies have emphasised the significant role of the construction industry in energy usage, with buildings accounting for around one-third of global greenhouse gas emissions [3]. Starting from 2012, there has been a consistent annual increase of 1.5 % in energy consumption within buildings in nations belonging to the Organisation for Economic Co-operation and Development (OECD) such as Australia, New Zealand, the United Kingdom, and the USA. In addition, nations outside the OECD have experienced a higher increase of 2.1 % in their energy consumption levels [4]. Therefore, balancing economic viability, environmental sustainability, and occupant comfort, health, and safety is crucial for all stakeholders involved in the building sector [5]. Being able to make accurate forecasting of a building's energy consumption is a crucial requirement in this context.

Approaches for building energy consumption forecasting are typically categorised into three primary groups: 1) engineering calculations, 2) numerical simulations, and 3) data-driven modelling [4]. The first two methods rely heavily on physical laws or physics-based simulations to estimate energy usage. Engineering calculations are best suited for quick initial evaluations, offering straightforward estimations based on standard formulas. Numerical simulations, on the other hand, provide a more detailed analysis by modelling the complex interactions within buildings but demand significant computational resources and time, particularly as the complexity of a project increases [4].

⋆ This research was supported by Massey University.
* Corresponding author.
E-mail address: S.Ranathunga@massey.ac.nz (S. Ranathunga).

https://doi.org/10.1016/j.enbuild.2025.115632
Received 18 November 2024; Received in revised form 6 March 2025; Accepted 16 March 2025
Available online 18 March 2025

Nomenclature

ASHRAE	American society of heating, refrigerating and air-conditioning engineers
BDG2	building data genome project 2
BPNN	back-propagation neural network
CNN	convolutional neural network
DANN	domain adversarial neural network
DL	deep learning
ELM	extreme learning machine
FEDformer	frequency enhanced decomposed transform
FF	feed forward
GRU	gated recurrent unit
LSTM	long-short term memory
MAE	mean absolute error
ML	machine learning
MLP	multi-layer perceptron
MSE	mean squared error
PatchTST	patch time series transformer
RNN	recurrent neural networks
TFT	temporal fusion transformer
TL	transfer learning

The evolution of data-driven modelling and its rise in usage for forecasting building energy consumption and assessments is due to a significant shift away from traditional statistical models such as ARIMA and SARIMA to advanced Machine Learning (ML) techniques [6,7]. ML methods, notably Deep Learning (DL) techniques, are better suited for the dynamic and intricate energy usage patterns, offering significant improvements in forecasting accuracy and applicability across different temporal and spatial scales in energy planning models [8].

Recurrent Neural Networks (RNNs) and their advanced variants Long Short-Term Memory (LSTMs) networks and Gated Recurrent Units (GRUs) have been the most commonly used DL architectures for building energy forecasting (See Table 1). Some researchers have also used Convolutional Neural Networks (CNNs) [9]. However, both RNNs and CNNs have their drawbacks with respect to time series forecasting [10–12]. RNNs suffer from vanishing and exploding gradient problems when processing long input sequences. They also struggle in capturing global information from the input sequence and cannot generally benefit from hardware parallelization. CNNs also struggle to capture long-range dependencies.

As an alternative, Transformer models, which are a sophisticated type of neural network architecture, have emerged as exceptionally effective in processing complex data sequences [13]. Renowned for their effectiveness in domains such as Natural Language Processing [14] and Computer Vision [15], Transformers can be used for interpreting the intricate interrelations in time series data that affect building energy usage. While the basic Transformer architecture (which we term the *vanilla Transformer*) has been shown to outperform other DL models such as LSTMs for building energy consumption forecasting [16], several recent studies have demonstrated that advanced Transformer variants such as Informer [17], PatchTST [18] and Temporal Fusion Transformers [19] have already outperformed their older DL counterparts [16,20].

Despite their promise, all DL techniques - CNNs, RNNs and Transformers alike, have one major limitation - they rely on vast amounts of training data to make accurate forecasting. However, such large amounts of energy consumption data may not exist for some buildings, due to practical reasons such as the building being newly constructed, not having the facilities to record energy consumption data in a timely manner, privacy concerns, data collection costs, data ownership, and the sheer diversity of buildings [21,22]. However, there are multiple publicly available datasets that contain energy consumption data from different buildings around the world, such as the Building Data Genome Project 2 (BDG2) [23–25], which can be leveraged.

Transfer Learning (TL) can be broadly defined as 'the ability of a system to recognise and apply knowledge and skills learned in previous tasks to novel tasks' [26]. In other words, TL utilises insights derived from a more documented dataset (the source) to bolster the predictive accuracy of models applied to a new, data-sparse context (the target). TL is an excellent way to make use of the aforementioned existing datasets to build energy consumption models for buildings that have limited or no data of their own.

Our survey of existing literature (See Table 1) revealed that the TL techniques used in the context of building energy consumption forecasting can be categorised based on how the *source* and *target* datasets are being used, as shown in Fig. 1 and outlined below.[1] We term these *data-centric TL strategies*.

- **Strategy 1:** Train a model with one source (S) → test with the target (T) - (1S → T)
- **Strategy 2:** Train a model with multiple sources (MS) → test with the target - (MS → T)
- **Strategy 3:** Train a model with one source → further train with the target (FT)→ test with the target - (1S → FT → T)
- **Strategy 4:** Train a model with multiple sources → further train with the target → test with the target - (MS → FT → T)
- **Strategy 5:** Train a model with one source and target → test with the target - (1S + T → T)
- **Strategy 6:** Train a model with multiple sources and target → test with the target - (MS + T → T)
- **Strategy 7:** Train a model with one source and target → further train with the target → test with the target - (1S + T → FT → T)
- **Strategy 8:** Train a model with multiple sources and target → further train with target → test with the target - (MS + T → FT → T)

1.2. Problem and motivation

While TL has been used for building energy consumption forecasting, most research has experimented with only one data-centric TL strategy. Moreover, only a few studies used the Transformer architecture [27,28]. Laitsos et al. [29] stand out for investigating four data-centric TL strategies, but their experimentation was limited in scale, involving only three buildings, without considering the Transformer architecture. Similarly, Lu et al. [30] experimented with two strategies on the LSTM model. Consequently, the potential of combining various TL techniques with Transformers for building energy consumption forecasting remains largely unexplored.

1.3. Aims of the study

This study aims to comprehensively investigate the effectiveness of various data-centric TL strategies when applied to Transformer architectures for building energy consumption forecasting. Specifically, we seek to answer the following Research Questions (RQs):

- RQ1: What is the best data-centric TL strategy for building energy consumption forecasting under a given data setup?
- RQ2: What specific features of building energy datasets (e.g. ambient weather features, climate zone, data volume) influence the effectiveness of different data-centric TL strategies?
- RQ3: How does the performance of various data-centric TL strategies differ when applied to advanced Transformer architectures specifically designed for time series forecasting, compared to the vanilla Transformer models?

[1] Note that the first two data-centric TL strategies above can be identified as the zero-shot setup, as the model does not see any target data during training time.

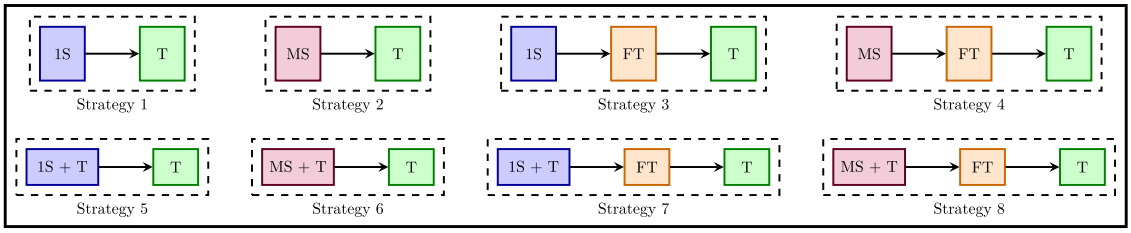


Fig. 1. Data-centric TL strategies.

1.4. Contribution

In order to answer the three RQs listed above, we conducted a large-scale experiment involving 16 datasets from BDG2 [23]. Our study makes several key contributions to the field of building energy forecasting:

1. Comprehensive analysis of TL Strategies: We conducted experiments for six out of the eight data-centric transfer learning strategies (1, 2, 5, 6, 7, and 8), making this one of the most comprehensive studies on the use of TL for building energy forecasting. This broad approach allows for a rich understanding of how different TL techniques perform in various scenarios.
2. Impact analysis of dataset characteristics: We analysed the impact of the following features in the dataset: climate zone, weather features, data volume and temporal range.
3. Large-Scale Modelling with Advanced Transformer Variants: In addition to the vanilla Transformer, we extended our experiments to include two advanced Transformer variants: Informer and PatchTST.

By combining these elements, our study provides a holistic view of TL applications on the Transformer architectures for the task of building energy forecasting models and offers practical insights for researchers and practitioners in the field of building energy consumption forecasting.

2. Related work

2.1. Transfer learning

Transfer Learning (TL) is an ML technique that makes use of the knowledge acquired from a *source* model that has been trained on different, but sufficient datasets to improve the performance of a new *target* model from a specific domain where there is insufficient or no data. The conceptual backbone of TL is structured around two key elements: domains and tasks. A domain is defined by a feature space $\mathcal{X}$ and a corresponding marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$. A task consists of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$. Together, these define the specific regression or classification task at hand.

In the field of building energy consumption forecasting, three types of input data can be identified: ambient environmental data, historical data, and time data [31]. Ambient environmental data is usually obtained from the Meteorological Agency or by measuring it in the field. Examples of environment data types include outdoor temperature, humidity, and solar radiation [31]. Historical data refers to the building energy consumption data that has already been recorded. Time data refers to the time of the year. Huy et al. [32] showed that special days such as New Year and public holidays have an impact on the prediction accuracy of TL techniques. Time data is particularly important in countries where distinct seasons are present, and some researchers refer to such data as seasonal data [33].

These types of data recorded in the context of a building form the feature space $\mathcal{X}$ of the TL problem. Lu et al. [34] argue that every building is personalised, with their thermal performance and occupant behaviour being different. Therefore it is safe to consider each building as a separate domain [35]. If the source and target domains have the same feature space, it is termed *homogeneous*, otherwise *heterogeneous*.

Based on the nature of source and target tasks and domains, TL can be categorised into three primary types—inductive, transductive, and unsupervised [36]. Inductive TL refers to the case where the target task is different from the source task (source and target domains can be similar or not), while transductive TL refers to the case where the source and target domains are different, but the tasks are the same. Transductive TL is also referred to as 'domain adaptation'. If the label information is unknown for both domains, this is considered unsupervised TL. However, for time series forecasting tasks, the use of unsupervised TL is rather uncommon.[2]

Approaches for TL can be categorised into 4 groups: instance-based, feature-based, parameter-based (aka model-based), and relational-based approaches. Table 1 shows a non-exhaustive list of past research on building energy consumption forecasting with TL and DL techniques. Most of this research can be identified as parameter-based techniques, with the exception of Fang et al. [37],Li et al. [38,39] and some experiments of Li et al. [40], which are feature-based techniques.

Table 1 also lists the data-centric TL strategy employed by the previous research. This confirms that the past research, except Laitsos et al. [29] and Lu et al. [30], experimented with only one data-centric TL strategy. With respect to various TL strategies described in Fig. 1, out of the 38 papers we surveyed, 44 % used Strategy 3, followed by Strategy 4, which was used by 23 %. Strategies 6, 5, 1 and 8 were applied by 16 %, 13 %, 8 % and 5 %, respectively. To the best of our knowledge, no studies have investigated strategies 2 and 7.

Further training a model that has already been trained is termed 'fine-tuning'. When fine-tuning a DL model that consists of multiple layers, the decision regarding which layers to fine-tune is an important factor. As shown in Table 1, related research suggests several options. The simplest option is not to fine-tune layers of the pre-trained model. In other words, all the layers in the pre-trained model are frozen, and the model is simply used for inferencing (i.e., test with the target—this refers to zero-shot learning and is useful when the target has no data for model training). Fine-tuning all the layers of the pre-trained model with target data is termed 'full fine-tuning' or 'weight initialisation'. On the other hand, fine-tuning only the last layer of the pre-trained model with target data is termed 'feature extraction'. It is always possible to fine-tune a custom set of layers of the pre-trained model with target data. Which layers to fine-tune is important to consider, as the performance of transfer learning depends on the similarity between the source and target domains. The more layers that are fine-tuned, the closer the model resembles the target domain.

Relatedly, various similarity measurement indexes have been used to identify source domain(s) similar to the target domain. Jung et al. [45] used the 'Pearson Correlation' for this task, while Lu et al. [34] developed their own correlation measurement, which they termed 'Similarity Measurement Index'. Peng et al. [52] and Li et al. [39] used 'Dynamic Time Warping', combined with 'Euclidean distance'. Other techniques

[2] We did not find any research that claims to use unsupervised TL for building energy consumption forecasting.

Table 1
Summary of TL techniques in building energy consumption forecasting.

Authors	DL	Strategy	Fine-tuning
Ribeiro et al. [33]	MLP	Strategy 6	N/A
Voß et al. [41]	CNN	Strategy 6	N/A
Tian et al. [35]	RNN	Strategy 3	Full
Hooshmand & Sharma [42]	CNN	Strategy 4	Last FC layer
Fan et al. [43]	LSTM	Strategy 4	Last layer, full
Gao et al. [44]	CNN & LSTM	Strategy 3	Top dense layers
Jung et al. [45]	FF	Strategy 4	Full
Ma et al. [46]	LSTM	Strategy 3	Bottom layers
Hu et al. [47]	DRN	Strategy 3	Not mentioned
Lee et al. [48]	LSTM	Strategy 4	Full
Li et al. [49]	BPNN	Strategy 3	Full
Jain et al. [50]	FF	Strategy 3	Last two layers
Fang et al. [37]	LSTM-DANN	Strategy 5	N/A
Lu et al. [34]	LSTM	Strategy 3	Full
Park et al. [51]	LSTM, CNN	Strategy 3	Partial
Peng et al. [52]	LSTM	Strategy 3	Last layer
Ahn et al. [31]	LSTM	Strategy 3	Top/bottom layers
Li et al. [40]	LSTM	Strategy 3	Last layer
Yan et al. [53]	bi-LSTM	Strategy 8	Last layer
Kim et al. [54]	LSTM	Strategy 3	No/Full/Partial
Lu et al. [30]	LSTM	Str. 3, 4	Partial layers
Tzortzis et al. [55]	MLP	Strategy 4	Full
Yuan et al. [56]	CNN-LSTM	Strategy 1	N/A
Yang et al. [57]	LSTM	Strategy 4	New layers only
Zhou et al. [58]	TAB-LSTM	Strategy 5	N/A
Fang et al. [59]	LSTM	Strategy 5	Not mentioned
Gokhale et al. [27]	TFT	Strategy 4	Full
Laitsos et al. [60]	CNN, GRU, CNN-GRU	Strategy 1	N/A
Santos et al. [28]	TFT	Strategy 3	Last/Partial/Full
Laitsos et al. [29]	MLP, CNN, ELM	Str. 1,3,4,6	Full
Li et al. [61]	LSTM-DANN	Strategy 6	N/A
Kim et al. [62]	LSTM	Strategy 3	Partial
Xiao et al. [63]	LSTM, GRU	Strategy 3	Last layer
Wei et al. [64]	LSTM	Strategy 8	Not mentioned
Xing et al. [65]	LSTM, vanilla Transformer	Strategy 3	Last layer
Li et al. [38]	LSTM	Strategy 5	N/A
Li et al. [39]	LSTM-DANN	Strategy 6	N/A
Wei et al. [66]	LSTM	Strategy 6	N/A

Abbreviations: 1S: One-source, MS: Multi-source, T: Target, DRN: Deep Residual Network, FF: Feedforward Neural Network, MLP: Multi-Layer Perceptron, BPNN: Back-Propagation Neural Network, TFT: Temporal Fusion Transformer, DANN: Domain Adversarial Neural Network, FC: Fully Connected, ELM: Extreme Learning Machine. N/A fine-tuning is not applicable.

include 'Maximum Mean Dispersion' [58], 'Variational Mode Decomposition' [65] and a combination of 'Wasserstein Distance' and 'Maximal Information Coefficient' [64].

In order to further explain results variations when using different source domains, some studies evaluated the impact of source domain data volume size [38,61], seasonality [33], and climate zone [31]. Li et al. [39] investigated the impact of training data volumes, seasonal information, building type and location, while Park et al. [51] considered seasonality and occupation level.

2.2. Transformers & their application in building energy consumption forecasting

The Transformer model (i.e., the vanilla Transformer), introduced by Vaswani et al. [13] in 2017, represents a significant advancement in neural network design for processing data sequences. Unlike other neural networks that rely on recurrent or convolutional layers, the Transformer uses a self-attention mechanism to compute outputs based on the entirety of input sequences directly, which allows it to efficiently handle long-range dependencies. In the context of time series forecasting, the raw sequential data (e.g., historical measurements) are first converted into vector embeddings, ensuring that numerical features, categorical features, or both are represented appropriately. Positional encodings are then added to these embeddings—commonly computed via trigonometric functions with varying frequencies—to capture the temporal order that self-attention alone does not inherently model. These embedded and positionally encoded inputs are passed into the encoder, a stack of multi-head self-attention and feed-forward layers that learns representations of the entire sequence. The decoder, which also employs self-attention, then takes as input either the ground truth or previously forecast values (embedded and positionally encoded in the same fashion) and attends to the encoders output to generate predictions for future time steps.

More recently, several advanced Transformer architectures have been introduced for the general problem of time series forecasting. Some of these architectures are Frequency Enhanced Decomposed Transformer (FEDformer), Informer, Patch Time Series Transformer (PatchTST), and Temporal Fusion Transformer (TFT).

FEDformer model combines frequency domain analysis with the Transformer architecture. It decomposes time series data into different frequency components and applies attention mechanisms in both time and frequency domains, potentially capturing both short-term and long-term patterns more effectively. Informer is designed to address the quadratic computational complexity of standard Transformers. It uses a ProbSparse self-attention mechanism. This allows it to handle longer sequences more efficiently, making it particularly suitable for long-term time series forecasting tasks. PatchTST architecture adapts ideas from vision Transformers to time series data. It divides the input time series into patches, similar to how image patches are used in vision Transformers. This approach can capture local patterns within patches while maintaining the ability to model long-range dependencies. TFT is specifically designed for multi-horizon forecasting with multiple related time series. It incorporates specialised components for processing static covariates, known future inputs and observed inputs,

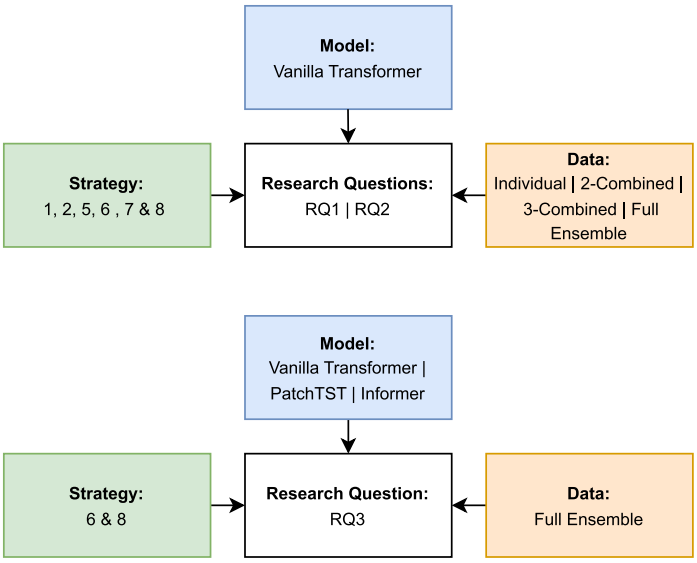


Fig. 2. Conceptual diagram of the research framework.

making it well-suited for complex forecasting scenarios with multiple input types.

Out of these, TFT, which is relatively older has been experimented with by several researchers for the task of energy consumption forecasting [20,32,67–70], and one has used FEDformer [71]. Hertel et al. [72] evaluated multiple time series Transformer architectures: vanilla Transformers, convolutional self-attention Transformer, and Informer, and reported that the latter two are superior. The TL research that used Transformer models has primarily focused on the TFT architecture [27,28]. Gokhale et al. [27] applied TFT with TL for demand forecasting in home energy management systems, while Santos et al. [28] explored various fine-tuning strategies using TFT for short-term load forecasting in data-poor buildings within local energy communities. These studies demonstrate the growing interest in leveraging advanced Transformer architectures.

3. Methodology and experimentation

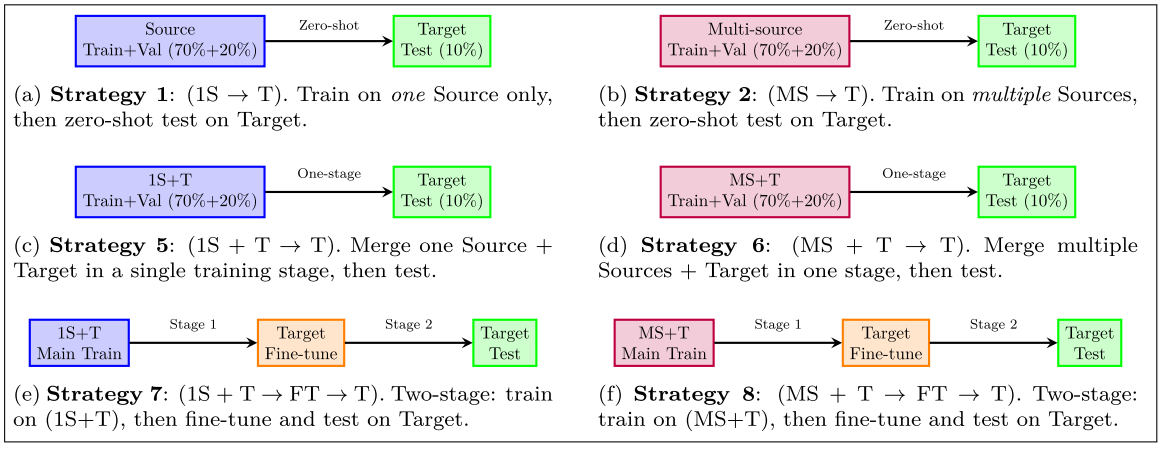
In order to answer the research questions, we experimented with different data-centric TL strategies, using different dataset combinations mentioned above and three different Transformer variants. Fig. 2 depicts the conceptual research framework. In order to answer RQ1, we implemented six of the eight data-centric TL strategies (1–2 and 5–8) on the vanilla Transformer (see Fig. 3). Strategies 3 and 4 were not considered because if target data is available, fine-tuning a model by combining that target data with the source data (strategies 5–8) is always beneficial. As the source domain data, we used the individual datasets, as well as combinations of datasets (2, 3 and all the datasets). In order to answer RQ2, we combined and truncated individual datasets as necessary (Fig. 4). For investigating RQ3, we experimented with PatchTST and Informer, in addition to the vanilla Transformer.

3.1. Performance comparison of different TL strategies with vanilla transformers

For the vanilla Transformer, we implemented six of the eight data-centric TL strategies (1–2 and 5–8), as mentioned above.

Baseline. We trained vanilla Transformer models with the training split of each dataset and tested with the corresponding test split of the same dataset.

Strategy 1 (single source zero-shot). We used the baseline models for zero-shot testing on other datasets. Here, the 'source' data is the dataset



(a) **Strategy 1**: (1S → T). Train on *one* Source only, then zero-shot test on Target.

(b) **Strategy 2**: (MS → T). Train on *multiple* Sources, then zero-shot test on Target.

(c) **Strategy 5**: (1S + T → T). Merge one Source + Target in a single training stage, then test.

(d) **Strategy 6**: (MS + T → T). Merge multiple Sources + Target in one stage, then test.

(e) **Strategy 7**: (1S + T → FT → T). Two-stage: train on (1S+T), then fine-tune and test on Target.

(f) **Strategy 8**: (MS + T → FT → T). Two-stage: train on (MS+T), then fine-tune and test on Target.

Fig. 3. The Six Chosen Data-Centric Transfer Learning Strategies in This Study. Each box indicates a training or fine-tuning stage (the first 70 % of data), followed by validation (20 %) on the same dataset(s), and final testing (10 %) on the target.

- Unmodified individual : Original, unprocessed datasets from BDG2.
- Unmodified combined : Combinations of two or more original datasets without modifications (meaning that these original datasets may vary with respect to climate zone, building count, etc).
- Uniform : Combinations standardised across all features for consistent evaluation (meaning that these original datasets have the same values with respect to climate zone, building count, etc.).
- Climate-variant : Combined datasets vary only by the geographical location.
- Building count-variant : Combined datasets vary only by the number of buildings included in the dataset.
- Weather feature-variant : Combined datasets vary only with respect to the weather features.
- Temporal range-variant : These combinations align datasets with different original time spans.

Fig. 4. Colour coding scheme that describes the variation in feature spaces of the combined datasets.

Table 2
Temporal range-variant and truncated dataset combinations.

Dataset	Composition	Key Modifications
Peacock + Wolftruncated1	36 Buildings each, Climate Zone 5A	**Wolftruncated1** is derived from the original Wolf dataset by reducing its number of weather features from 6 to 3 to match the Peacock dataset.
Eagletruncated1 + Robin	50 Buildings each, Climate Zone 4A	**Eagletruncated1** is a version of the Eagle dataset where the number of buildings is reduced from 87 to 50 to match the Robin dataset.
Bear + Foxtruncated1	73 Buildings each, Bear: Climate Zone 3C, Foxtruncated1: Climate Zone 2B	**Foxtruncated1** is created from the Fox dataset by reducing the number of buildings from 127 to 73 to match the Bear dataset.
Bobcat + Moosetruncated1	7 Buildings each, Bobcat: Climate Zone 5B, Moosetruncated1: Climate Zone 6A	**Moosetruncated1** is derived from the Moose dataset by reducing the number of buildings from 9 to 7 to match the Bobcat dataset.
Lamb + Robintruncated1	41 Buildings each, Climate Zone 4A	**Robintruncated1** is created by reducing the number of weather features in the Robin dataset from 5 to 4 to match the Lamb dataset.
Bulltruncated1 + Gator	Bulltruncated1: 41 Buildings, Gator: 29 Buildings, Climate Zone 2A	**Bulltruncated1** is derived from the Bull dataset by reducing the number of weather features from 3 to 0 to match the Gator dataset, which has no weather features.
Hogtruncated1 + Moosetruncated2	9 Buildings each, Climate Zone 6A	**Hogtruncated1** is derived from the Hog dataset by retaining only 35% of the data for training, excluding 35% by zero-padding in the middle, and using the remaining 30% for validation and testing. **Moosetruncated2** is obtained by excluding the first 35% of data through zero-padding, using the next 35% for training, and the final 30% for validation and testing.
Hogtruncated2 + Moosetruncated2	9 Buildings each, Climate Zone 6A	**Hogtruncated2** is created from the Hog dataset without zero-padding, using a standard 70% training and 30% validation/testing split. **Moosetruncated2** is the same as in the previous combination.

Table 3
Dataset characteristics and experiment categories.

Dataset	B	CZ	AT	DT	SLP	WD	WS	CC
Bear	73	3C	1	1	1	1	1	0
Bobcat	7	5B	1	1	1	1	1	0
Bull	41	2A	1	1	1	0	0	0
Cockatoo	1	6A	1	1	1	1	1	0
Crow	4	6A	1	1	1	1	1	0
Eagle	87	4A	1	1	1	1	1	0
Fox	127	2B	1	1	1	1	1	0
Gator	29	2A	0	0	0	0	0	0
Hog	24	6A	1	1	1	1	1	0
Lamb	41	4A	1	1	0	1	1	0
Moose	9	6A	1	1	1	1	1	0
Mouse	3	4A	1	1	1	1	1	0
Peacock	36	5A	1	1	1	0	0	0
Rat	251	4A	1	1	1	1	1	0
Robin	50	4A	1	1	1	1	1	0
Wolf	36	5A	1	1	1	1	1	1
Bull + Cockatoo + Hog	66	MC	3×1	3×1	3×1	2×1,1×0	2×1,1×0	3×0
Bear + Fox	200	MC	2×1	2×1	2×1	2×1	2×1	2×0
Bobcat + Moose	16	MC	2×1	2×1	2×1	2×1	2×1	2×0
Bull + Gator	70	2A	1×1,1×0	1×1,1×0	1×1,1×0	2×0	2×0	2×0
Crow + Robin	54	MC	2×1	2×1	2×1	2×1	2×1	2×0
Hog + Moose	33	6A	2×1	2×1	2×1	2×1	2×1	2×0
Lamb + Robin	91	4A	2×1	2×1	1×1,1×0	2×1	2×1	2×0
Mouse + Rat	254	4A	2×1	2×1	2×1	2×1	2×1	2×0
Peacock + Wolftruncated1	36×2	5A	2×1	2×1	2×1	2×0	2×0	2×0
Eagletruncated1 + Robin	50×2	4A	2×1	2×1	2×1	2×1	2×1	2×0
Bear + Foxtruncated1	73×2	MC	2×1	2×1	2×1	2×1	2×1	2×0
Bobcat + Moosetruncated1	7x2	MC	2×1	2×1	2×1	2×1	2×1	2×0
Eagle + Robin	87×1,50×1	4A	2×1	2×1	2×1	2×1	2×1	2×0
Bulltruncated1 + Gator	41×1,29×1	2A	2×0	2×0	2×0	2×0	2×0	2×0
Peacock + Wolf	36×2	5A	2×1	2×1	2×1	1×1,1×0	1×1,1×0	1×1,1×0
Lamb + Robintruncated1	41×2	4A	2×1	2×1	1×1,1×0	2×1	2×1	2×0
Hogtruncated1 + Moosetruncated2	9×2	6A	2×1	2×1	2×1	2×1	2×1	2×0
Hogtruncated2 + Moosetruncated2	9×2	6A	2×1	2×1	2×1	2×1	2×1	2×0
Full Ensemble	819	MC	15×1,1×0	15×1,1×0	14×1,2×0	13×1,3×0	13×1,3×0	1×1,15×0

Abbreviations: B: Number of buildings, CZ: Climate Zone, MC: Mixed Climates, AT: Air Temperature, DT: Dew Temperature, SLP: Sea Level Pressure, WD: Wind Direction, WS: Wind Speed, CC: Cloud Coverage, MC: different climate zones.

Colour coding: Unmodified individual, Unmodified combined, Uniform, Climate-variant, Building count-variant, Weather feature-variant, Temporal range-variant, Full Ensemble.

For combined datasets, entries show the count of each feature value (e.g., 2×1,1×0 means two '1' values and one '0' value). Full Ensemble includes all unmodified individual datasets. Truncation (indicated by 'truncated1' or 'truncated2' appended to dataset names) refers to reducing one or more of these: the number of buildings, the number of weather features, or the temporal range of the dataset.

used to train the model, whereas the 'target' data is any external dataset on which zero-shot inference is carried out. No additional training occurs on the target data in this scenario.

Strategy 2 (multi-source zero-shot). We trained models on combinations of datasets and tested them on the test splits of other datasets (I.e. those that were not used in training).The combinations included two-dataset combinations and one three-dataset combination. In this scenario, multiple datasets serve as the source for training, while the unseen dataset is the target.

Strategies 5 and 6. We reused the models trained under Strategy 2 and tested them with the test splits of the datasets that were used to train those models. In addition, we tested with the full ensemble (combining all datasets).

Strategy 7 (fine-tuning two-dataset models). We further fine-tuned the models trained on two-dataset combinations with additional training on each target dataset alone.

Strategy 8 (fine-tuning three-dataset models and full ensemble models). We further fine-tuned the models trained on three-dataset combinations and the full ensemble model on each individual dataset.

During fine-tuning, we adjusted all parameters of the model without freezing any layers. The findings of Santos et al. [28] support this ap-

Table 4
An excerpt from the Wolf dataset in BDG2, illustrating hourly energy consumption and wind data.

datetime_index	Wolf_assembly_Sallie (kWh)	Wolf_education_Cody (kWh)	windDirection (deg)	windSpeed (m/s)
2016-01-01 00:00	154.33	9.89	200	5
2016-01-01 01:00	154.47	12.90	190	3
2016-01-01 02:00	154.17	10.44	170	3
2016-01-01 03:00	152.62	9.87	170	4

Note. Additional weather columns (e.g., `airTemperature`, `cloudCoverage`, `dewTemperature`, `seaLvlPressure`) are present in the full dataset but omitted here for brevity.

proach, which demonstrated the benefits of fine-tuning all parameters on Transformers instead of freezing layers in load forecasting contexts.

3.2. Evaluation of Transformer variants

We extended our experiments to include advanced Transformer variants: Informer and PatchTST, to compare their performance under similar experimental conditions. For both Informer and PatchTST architectures, we conducted the following experiments: baseline experiments (Trained only on each dataset individually), Strategy 6 (full ensemble model) and Strategy 8 (further fine-tuning full ensemble model on individual targets).

3.3. Dataset description

This study employs a comprehensive and diverse collection of datasets from the BDG2. The datasets include hourly energy consumption data for various buildings across multiple geographic locations and climate zones, such as educational institutions, offices, and public buildings. Each dataset exhibits a unique combination of features that influence energy consumption:

- Load Type: All datasets focus on building energy consumption.
- Datetime Range and Number of Rows: The data spans two full years (01/01/2016–31/12/2017), with each dataset containing 17,544 hourly records, 53.6 million in total. This provides a substantial temporal range for analysing seasonal variations and trends in energy use.
- Granularity: Data is recorded hourly, allowing for detailed analysis of daily and seasonal peaks in energy demand.
- Actual Site Name and Location: Researchers collected data from 19 sites across North America and Europe, anonymising some datasets while specifying other locations. This variation allows researchers to explore how regional climate conditions and building practices influence energy consumption.
- Climate Conditions: The datasets are categorised using ASHRAE climate zones [73] (e.g., '3C - Warm and marine'), enabling the study of climate-specific energy usage patterns and the development of tailored forecasting models.
- Building and Weather Features: Detailed information on operational characteristics (e.g., building type and usage) and weather conditions (e.g., temperature, humidity) is included.
- In this study, each dataset is initially split into 70 % training (years 2016–2017 with partial daily coverage), 20 % validation, and 10 % testing segments, chronologically. This ensures no data leakage from the future into the training period.

3.4. Data preprocessing and dataset combinations

The BDG2 datasets underwent initial cleaning by the original dataset authors, though further preprocessing was necessary to refine the data quality for load forecasting. Using statistical thresholds, we identified outlier buildings and weather features, then employed the ProEnFo package, as described by Wang et al. [74], to further clean the datasets. Buildings with 10 % or more missing values were removed to maintain data integrity, and remaining missing data points were imputed using linear interpolation, applied consistently across all datasets. This uniform imputation method ensures that differences in model performance are not confounded by divergent approaches to missing-data handling. After interpolation, we eliminated columns with more than 3000 zero values and converted timestamp data into a date-time index for time series analysis.

Following pre-processing, 16 datasets remained. We label each dataset as a source if used for model training, and as a target if serving for zero-shot or fine-tuned testing. For experiments involving multiple sources, we combined selected datasets to create variations in feature spaces, such as weather variables, building counts, or temporal ranges. To simulate data shortage scenarios, we introduced truncated datasets by reducing available columns, limiting training periods, or removing building data. These truncations took three forms: weather feature removal, temporal range reduction with zero-padding, and building data elimination. The final set of datasets and their characteristics are detailed in Table 3. An excerpt of the dataset is shown in Table 4.

3.5. Experimental setup

Our implementation builds upon the work of Hertel et al. [16] for the Vanilla Transformer, Informer, and PatchTST models. We maintained their hyper-parameters with minor adjustments to the learning rate: 0.0001 for strategies 1, 2, 5, and 6, and 0.00001 for strategies 7 and 8. The models were trained using an NVIDIA GeForce RTX 3090 GPU and implemented with PyTorch running on Python 3.10. We use a default input context window of 168 hourly time steps and generate multi-step forecasts for 24 h and 96 h horizons. Performance evaluation follows previous research, using MAE for primary discussion and MSE results provided in the Appendix.

Our study encompasses 332 unique experiments, each conducted three times with different seeds to ensure robustness, resulting in 996 total models. Each experiment represents a unique combination of dataset configuration, transfer learning strategy, forecast horizon, and model architecture.

The experimental design spans multiple model architectures, with the Vanilla Transformer receiving the most extensive testing (200 experiments), followed by Informer and PatchTST (66 experiments each). This comprehensive approach allows us to evaluate performance across different architectural choices and TL strategies, particularly in scenarios involving data shortage through truncated or weather-feature-limited datasets. The complete distribution of experiments across different strategies and architectures is detailed in Table 5.

4. Evaluation

4.1. Results of zero-shot testing

These results correspond to Strategies 1 and 2. Tables 6 and 7 show the MAE results for 24 h and 96 h forecasting, respectively (Corresponding MSE results can be found in Tables A.10 and A.11 in Appendix). The top part of the table contains results for models trained with individual datasets. The bottom part shows the models trained with multiple datasets. In these tables, values highlighted in grey are the baseline results (i.e. train and test set belong to the same dataset). For models

Table 5
Summary of models trained across strategies and architectures.

Strategy	Experiments	Total models
Vanilla Transformer		
Baseline	38	114
Strategy 2 (Multi-source Zero-shot)	44	132
Strategy 7 (Fine-tuning Two-Dataset Models)	80	240
Strategy 8 (Fine-tuning Three-Dataset Models)	6	18
Strategy 8 (Fine-tuning Full Ensemble Models)	32	96
Subtotal	**200**	**600**
Informer		
Baseline	32	96
Strategy 6 (Non-fine-tuned Full Ensemble Models)	2	6
Strategy 8 (Fine-tuning Full Ensemble Models)	32	96
Subtotal	**66**	**198**
PatchTST		
Baseline	32	96
Strategy 6 (Non-fine-tuned Full Ensemble Models)	2	6
Strategy 8 (Fine-tuning Full Ensemble Models)	32	96
Subtotal	**66**	**198**
Grand Total	**332**	**996**

trained with multiple datasets, when a model is tested with a dataset that is included in the model, those columns are highlighted in purple. To reiterate, results highlighted in grey or purple are not zero-shot results—these were included in the tables for comparison purposes.

Baseline results. Even in baseline experiments, a significant variation in the results can be observed. In general, datasets with more building data show better performance than those with a lesser number of buildings. For example, datasets (except Bear and Cockatoo) that have at least about 30 buildings show an MAE less than 0.3 for the 24 h horizon forecasting case. However, it is difficult to say this observation always strictly holds. The correlation between the number of buildings and the MAE for the 24 h horizon baseline results is $R = -0.356$, $p = 0.05$, indicating a modest negative correlation, while for the 96 h baseline results, the correlation is $R = -0.413$, $p = 0.03$. These results suggest that increasing building counts tend to slightly reduce the forecast error, but the relationship is not strictly linear, possibly due to variations in data quality. Yuan et al. [56] also mention that it is not possible to remove all the noise and errors only by pre-processing. To analyze the impact of temporal range truncation (which simulates a low-data scenario), we consider the Moose and Hog datasets and their truncated versions (Moosetruncated2, Hogtruncated1, and Hogtruncated2). For Moose, which has data from only 3 buildings, temporal truncation results in a significant performance drop. For Hog, which has data from 24 buildings, no such drastic reduction is observed. Hogtruncated2 reports a slight increase over Hog, while Hogtruncated1 (zero-padded) shows degraded performance compared to Hog. This indicates that when the number of buildings is low, prediction results are impacted by data scarcity.

Zero-shot on individual dataset models (Strategy 1). Models trained on larger datasets (e.g. Fox with 127 buildings, Rat with 251 buildings) generally performed better in zero-shot scenarios compared to those trained on smaller datasets (e.g. Cockatoo with 1 building, Crow with 4 buildings, Mouse with 3 buildings, and Moose with 9 buildings), with datasets containing more than 70 buildings achieving an average MAE of 0.370 ± 0.075 compared to 0.497 ± 0.098 for datasets with fewer than 10 buildings, representing a 25.5 % improvement. Some models even outperformed their relevant baseline, with the Fox model showing improvements of 1–8.2 % over baseline for 8 different datasets (with the highest improvements for Bobcat at 8.2 % and Crow at 6.1 %), and the Rat model showing improvements of 0.6–6.7 % over baseline for 5 datasets (with the highest improvements for Crow at 6.7 % and Bobcat at 4.8 %) in the 24 h experiments. These results align with observations of prior research, which also observed that higher amounts of source domain data are beneficial for TL [28,40,59].

We also computed the correlation between the *source* dataset size and the *average zero-shot MAE* across all target datasets. For the 24 h experiments, this correlation is $R = -0.442$, $p = 0.02$, implying that source datasets with more buildings generally yield better zero-shot results across multiple targets.

Extreme differences between the weather features of source and target datasets cause a large impact as well. For example, zero-shot results derived from the Gator model (the Gator dataset does not record any weather feature) show consistently worse results across datasets. On the contrary, just because two datasets share the same weather features and climate zone, this does not guarantee optimal performance when transferring models between them. We can see this clearly in our analysis of the Robin, Rat, Mouse, and Eagle datasets. While all these datasets contain identical weather features and come from the same climate zone, models trained on one dataset often do not perform best when tested on another. In the case of Robin for example—when we used it as our test dataset for 24 h forecasts, the best results came from models trained on Fox, Bear, and Peacock datasets, rather than from the seemingly more similar datasets (Rat, Mouse, or Lamb). Mouse was the only exception to this pattern, where the shared features and climate zone actually did lead to the best performance.

Combined dataset models (Strategy 2). Combined dataset models often outperformed individual dataset models in zero-shot scenarios by 15.9 % (average MAE of 0.367 vs. 0.437), suggesting improved generalisation. However, due to the observations discussed above, blindly combining multiple data sources will not provide better results over using a single source dataset. For example, Bull (separately) combined with Cockatoo, Gator and Hog shows dissimilar results - Bull + Cockatoo and Bull + Gator degrade performance by 0.2 % and 1.1 % respectively while Bull + Hog improves by 1.5 % compared to Bull's baseline MAE of 0.475.

The impact of the weather features and the dataset size is evident in these experiments as well. For example, when weather data of Wolf are removed to obtain Wolftruncated1 to match with the weather features of Peacock, the result of Peacock + Wolftruncated1 falls below the result of Peacock + Wolf (0.329 MAE). Similarly, when the building count of Eagle is reduced to obtain Eagletruncated1 to match with Robin, the result of Robin + Eagletruncated1 falls below Robin + Eagle (0.254 MAE).

Our analysis reveals a consistent degradation in forecast accuracy when moving from 24 h to 96 h predictions. Looking at average MAE values across different model configurations: models related to Strategy 2 showed the most resilient performance, with mean MAE increasing

Table 6
Average MAE for 24 h horizon zero-shot forecasting.

Model	Bear	Bobcat	Bull	Cockatoo	Crow	Eagle	Fox	Gator	Hog	Lamb	Moose	Mouse	Peacock	Rat	Robin	Wolf
Bear	0.305	0.393	0.483	0.483	0.346	0.401	0.296	0.432	0.347	0.237	0.293	0.362	0.334	0.320	0.274	0.396
Bobcat	0.428	0.415	0.509	0.471	0.368	0.498	0.393	0.438	0.360	0.335	0.337	0.404	0.414	0.391	0.382	0.464
Bull	0.442	0.478	0.475	0.503	0.389	0.469	0.404	0.480	0.360	0.279	0.417	0.417	0.415	0.388	0.388	0.509
Cockatoo	0.678	0.634	0.582	0.468	0.418	0.731	0.623	0.466	0.377	0.407	0.575	0.426	0.620	0.532	0.589	0.652
Crow	0.491	0.592	0.646	0.503	0.359	0.717	0.488	0.452	0.440	0.440	0.475	0.413	0.516	0.476	0.456	0.526
Eagle	0.435	0.473	0.506	0.464	0.357	**0.331**	0.409	0.414	0.351	0.279	0.512	0.360	0.418	0.388	0.334	0.528
Fox	0.302	0.381	**0.461**	0.473	0.337	0.371	0.268	0.433	0.329	0.207	0.281	0.355	0.322	0.302	0.265	0.402
Gator	0.838	0.698	0.581	0.523	0.492	0.795	0.758	**0.292**	0.391	0.406	0.626	0.490	0.747	0.578	0.724	0.803
Hog	0.499	0.521	0.515	0.470	0.362	0.610	0.468	0.432	0.336	0.298	0.476	0.376	0.476	0.424	0.426	0.550
Lamb	0.383	0.461	0.530	0.497	0.336	0.496	0.371	0.440	0.353	0.208	0.319	0.380	0.405	0.363	0.326	0.416
Moose	0.467	0.535	0.610	0.500	0.364	0.738	0.463	0.460	0.390	0.308	0.294	0.411	0.498	0.436	0.427	0.446
Mouse	0.445	0.535	0.548	0.470	0.364	0.608	0.446	0.422	0.367	0.356	0.461	0.365	0.470	0.441	0.374	0.499
Peacock	0.327	0.437	0.542	0.507	0.335	0.424	0.320	0.439	0.372	0.246	0.301	0.365	**0.316**	0.340	0.280	0.410
Rat	0.321	0.395	0.490	0.472	0.335	0.402	0.293	0.419	0.334	0.216	0.288	0.357	0.339	**0.293**	0.296	0.409
Robin	0.338	0.437	0.510	0.462	0.321	0.419	0.338	0.405	0.336	0.232	0.295	0.347	0.353	0.344	0.254	0.411
Wolf	0.417	0.494	0.610	0.533	0.382	0.536	0.405	0.492	0.424	0.365	0.360	0.419	0.445	0.425	0.370	0.387
Hogtrunc2	0.545	0.533	0.516	0.457	0.377	0.607	0.501	0.420	0.331	0.317	0.514	0.392	0.505	0.445	0.454	0.584
Hogtrunc	0.636	0.581	0.539	0.504	0.434	0.653	0.588	0.448	0.382	0.346	0.546	0.435	0.575	0.484	0.545	0.621
Moosetrunc2	0.629	0.631	0.592	0.533	0.445	0.772	0.613	0.473	0.408	0.360	0.459	0.457	0.618	0.515	0.581	0.634
Bear + Fox	**0.291**	**0.378**	0.463	0.466	0.330	0.376	**0.264**	0.422	0.326	0.212	0.279	0.347	0.320	0.299	0.263	0.395
Bobcat + Moose	0.384	0.399	0.522	**0.450**	0.331	0.503	0.365	0.420	0.337	0.268	**0.278**	0.365	0.392	0.364	0.330	0.419
Bull + Cockatoo	0.451	0.475	0.476	0.468	0.367	0.496	0.415	0.449	0.352	0.278	0.406	0.390	0.417	0.392	0.398	0.492
Bull + Gator	0.493	0.499	0.480	0.491	0.379	0.565	0.457	0.312	0.354	0.288	0.448	0.396	0.450	0.410	0.436	0.536
Bull + Hog	0.433	0.455	0.468	0.468	0.357	0.501	0.396	0.440	0.325	0.271	0.383	0.380	0.404	0.372	0.380	0.483
Cockatoo + Hog	0.488	0.512	0.521	0.451	0.347	0.620	0.458	0.426	0.323	0.301	0.447	0.375	0.473	0.416	0.426	0.533
Crow + Robin	0.337	0.441	0.543	0.460	**0.313**	0.432	0.343	0.400	0.337	0.233	0.301	**0.344**	0.350	0.347	0.255	0.407
Eagle + Robin	0.463	0.488	0.506	0.464	0.374	0.354	0.432	0.423	0.349	0.289	0.521	0.378	0.439	0.402	0.368	0.547
Hog + Moose	0.423	0.467	0.530	0.464	0.343	0.608	0.410	0.408	0.323	0.262	0.297	0.366	0.437	0.384	0.368	0.435
Lamb + Robin	0.322	0.411	0.516	0.474	0.321	0.393	0.311	0.403	0.337	**0.185**	0.284	0.358	0.341	0.322	**0.248**	0.395
Mouse + Rat	0.319	0.395	0.493	0.467	0.330	0.397	0.292	0.412	0.332	0.216	0.286	0.348	0.339	0.294	0.289	0.407
Peacock + Wolf	0.341	0.422	0.537	0.493	0.339	0.424	0.327	0.435	0.365	0.242	0.305	0.361	0.329	0.347	0.286	0.379
Bear + Foxtrunc	0.295	**0.378**	0.462	0.471	0.336	0.380	0.273	0.425	0.331	0.212	0.279	0.352	0.322	0.304	0.263	0.394
Bobcat + Moosetrunc	0.388	0.400	0.518	**0.450**	0.333	0.503	0.368	0.417	0.339	0.282	0.282	0.364	0.397	0.368	0.336	0.420
Bulltrunc + Gator	0.490	0.505	0.483	0.499	0.384	0.576	0.459	0.308	0.357	0.279	0.454	0.393	0.451	0.413	0.433	0.524
Eagletrunc + Robin	0.463	0.488	0.506	0.464	0.374	0.354	0.432	0.423	0.349	0.289	0.521	0.378	0.439	0.402	0.368	0.547
Hogtrunc2 + Moosetrunc2	0.518	0.565	0.550	0.503	0.395	0.688	0.507	0.426	0.371	0.305	0.408	0.401	0.544	0.453	0.464	0.562
Hogtrunc + Moosetrunc2	0.518	0.565	0.550	0.503	0.395	0.688	0.507	0.426	0.371	0.305	0.408	0.401	0.544	0.453	0.464	0.562
Lamb + Robintrunc	0.337	0.411	0.487	0.473	0.333	0.399	0.321	0.404	0.328	0.218	0.300	0.365	0.353	0.331	0.271	0.405
Peacock + Wolftrunc	0.340	0.424	0.522	0.492	0.348	0.417	0.327	0.427	0.368	0.248	0.318	0.370	0.329	0.344	0.288	**0.376**
Bull + Cockatoo + Hog	0.440	0.460	0.469	0.455	0.351	0.526	0.409	0.417	**0.322**	0.276	0.382	0.374	0.414	0.379	0.391	0.487

Table 7
Average MAE for 96 h horizon zero-shot forecasting.

Model	Bear	Bobcat	Bull	Cockatoo	Crow	Eagle	Fox	Gator	Hog	Lamb	Moose	Mouse	Peacock	Rat	Robin	Wolf
Bear	0.375	0.495	0.600	0.608	0.470	0.496	0.365	0.617	0.484	0.403	0.385	0.505	0.419	0.446	0.370	0.480
Bobcat	0.554	0.543	0.620	0.614	0.497	0.617	0.503	0.643	0.476	0.488	0.463	0.525	0.557	0.532	0.502	0.592
Bull	0.592	0.599	0.602	0.640	0.506	0.607	0.530	0.650	0.472	0.382	0.552	0.542	0.553	0.525	0.540	0.631
Cockatoo	0.780	0.719	0.667	0.600	0.545	0.803	0.708	0.661	0.488	0.519	0.662	0.561	0.726	0.638	0.686	0.718
Crow	0.610	0.727	0.814	0.658	0.476	0.851	0.592	0.698	0.648	0.713	0.630	0.562	0.677	0.648	0.596	0.615
Eagle	0.569	0.581	0.653	0.602	0.483	**0.392**	0.518	0.605	0.499	0.394	0.656	0.475	0.546	0.518	0.446	0.644
Fox	0.382	0.480	0.581	0.591	0.435	0.454	**0.325**	0.595	0.448	0.339	**0.362**	**0.463**	0.411	**0.413**	0.360	0.484
Gator	0.843	0.727	0.677	0.638	0.568	0.794	0.753	**0.503**	0.503	0.492	0.703	0.579	0.751	0.631	0.748	0.806
Hog	0.631	0.635	0.619	0.606	0.480	0.703	0.571	0.623	0.470	0.396	0.604	0.520	0.586	0.538	0.551	0.663
Lamb	0.476	0.567	0.674	0.637	0.460	0.613	0.464	0.630	0.496	0.269	0.421	0.499	0.527	0.488	0.422	0.507
Moose	0.578	0.652	0.749	0.648	0.504	0.880	0.561	0.669	0.539	0.453	0.377	0.549	0.646	0.574	0.551	0.524
Mouse	0.543	0.707	0.769	0.665	0.493	0.726	0.541	0.715	0.562	0.718	0.564	0.532	0.646	0.626	0.505	0.599
Peacock	0.420	0.528	0.674	0.639	0.467	0.517	0.400	0.643	0.513	0.373	0.400	0.484	0.411	0.467	0.374	0.492
Rat	0.419	0.487	0.612	0.611	0.458	0.504	0.380	0.621	0.454	0.289	0.377	0.478	0.442	**0.413**	0.402	0.505
Robin	0.427	0.550	0.650	0.611	0.447	0.519	0.419	0.613	0.474	0.321	0.398	0.484	0.454	0.462	0.344	0.488
Wolf	0.517	0.625	0.745	0.688	0.515	0.686	0.518	0.703	0.608	0.668	0.473	0.610	0.579	0.611	0.499	**0.453**
Hogtrunc2	0.743	0.673	0.635	0.606	0.520	0.742	0.660	0.630	0.479	0.450	0.666	0.546	0.654	0.583	0.646	0.725
Hogtrunc	0.819	0.705	0.648	0.639	0.591	0.796	0.746	0.697	0.502	0.462	0.712	0.584	0.702	0.615	0.725	0.749
Moosetrunc2	0.680	0.739	0.748	0.723	0.573	0.880	0.668	0.712	0.528	0.493	0.523	0.586	0.679	0.633	0.634	0.695
Bear + Fox	**0.367**	0.468	**0.576**	**0.587**	0.437	0.449	0.329	0.580	0.447	0.311	0.363	0.467	**0.400**	0.414	0.351	0.483
Bobcat + Moose	0.492	0.519	0.642	0.597	0.470	0.604	0.456	0.623	0.475	0.405	0.364	0.503	0.510	0.501	0.441	0.513
Bull + Cockatoo	0.580	0.592	0.604	0.603	0.496	0.626	0.531	0.657	0.466	0.403	0.542	0.515	0.539	0.528	0.541	0.599
Bull + Gator	0.626	0.616	0.602	0.618	0.506	0.668	0.566	0.525	0.468	0.398	0.625	0.523	0.574	0.540	0.559	0.666
Bull + Hog	0.571	0.577	0.598	0.618	0.487	0.633	0.510	0.637	0.459	0.373	0.519	0.516	0.529	0.504	0.511	0.622
Cockatoo + Hog	0.687	0.661	0.650	0.600	0.477	0.736	0.603	0.647	0.474	0.440	0.632	0.518	0.627	0.559	0.586	0.695
Crow + Robin	0.422	0.530	0.662	0.594	**0.423**	0.517	0.413	0.592	0.459	0.331	0.393	0.465	0.449	0.458	0.339	0.482
Eagle + Robin	0.628	0.610	0.639	0.595	0.498	0.441	0.560	0.620	0.477	0.425	0.679	0.505	0.579	0.538	0.502	0.688
Hog + Moose	0.526	0.566	0.632	0.603	0.465	0.719	0.499	0.626	0.464	0.367	0.402	0.491	0.527	0.497	0.485	0.528
Lamb + Robin	0.407	0.505	0.619	0.600	0.431	0.485	0.386	0.585	0.453	**0.260**	0.377	0.472	0.436	0.431	**0.338**	0.478
Mouse + Rat	0.414	0.485	0.614	0.610	0.448	0.492	0.378	0.608	0.464	0.293	0.378	0.475	0.437	0.414	0.397	0.504
Peacock + Wolf	0.437	0.549	0.669	0.615	0.454	0.516	0.418	0.637	0.507	0.433	0.413	0.492	0.429	0.495	0.381	0.455
Bear + Foxtrunc	0.371	**0.465**	0.579	0.595	0.441	0.464	0.336	0.588	0.457	0.328	0.373	0.468	0.401	0.414	0.360	0.489
Bobcat + Moosetrunc	0.496	0.521	0.639	0.593	0.467	0.612	0.460	0.621	0.468	0.407	0.370	0.500	0.523	0.504	0.448	0.516
Bulltrunc + Gator	0.610	0.609	0.611	0.634	0.514	0.688	0.563	0.524	0.482	0.383	0.629	0.513	0.562	0.532	0.552	0.637
Eagletrunc + Robin	0.628	0.610	0.639	0.595	0.498	0.441	0.560	0.620	0.477	0.425	0.679	0.505	0.579	0.538	0.502	0.688
Hogtrunc2 + Moosetrunc2	0.646	0.654	0.652	0.642	0.521	0.767	0.599	0.682	0.489	0.403	0.523	0.531	0.619	0.555	0.584	0.669
Hogtrunc + Moosetrunc2	0.646	0.654	0.652	0.642	0.521	0.767	0.599	0.682	0.489	0.403	0.523	0.531	0.619	0.555	0.584	0.669
Lamb + Robintrunc	0.431	0.535	0.621	0.597	0.436	0.493	0.402	0.584	**0.435**	0.304	0.384	0.471	0.474	0.452	0.358	0.485
Peacock + Wolftrunc	0.442	0.570	0.688	0.639	0.473	0.526	0.424	0.660	0.536	0.465	0.436	0.514	0.435	0.514	0.393	0.462
Bull + Cockatoo + Hog	0.566	0.569	0.604	0.598	0.461	0.640	0.506	0.618	0.454	0.367	0.513	0.489	0.520	0.495	0.509	0.604

from 0.367 to 0.463 (a 26.0 % degradation); models related to Strategy 1 showed slightly higher degradation, with mean MAE increasing from 0.430 to 0.554 (a 28.6 % increase). This systematic degradation in longer-horizon forecasts aligns with prior observations that shorter-term forecasting is typically more accurate in time series prediction [61].

Average gains over baselines. We computed the average gains over the corresponding baselines for the 16 BDG2 datasets under the best-performing TL strategies. For 24 h forecasts, smaller datasets (fewer than 10 buildings) saw an average improvement of 12.8 %, medium-sized datasets (10–70 buildings) improved by 8.1 %, and large datasets (over 70 buildings) improved by 5.5 %. This indicates that data-scarce scenarios benefit more from TL, whereas large datasets can still gain but to a lesser degree.

4.2. Results comparison of strategies 5–8

Table 8 shows the results for models trained by combining 2 or 3 datasets. Table 9 reports results for Strategies 6 and 8 considering the ensemble model trained with all the datasets. Note that in both these tables, the first six rows refer to strategies 6 and 8. All the other results refer to strategies 5 and 7. In both the tables, the first 'Imp' column shows the improvement of strategy 5/6 over baseline. The second 'Imp' column shows the improvement of strategy 7/8 over the baseline.

4.2.1. Strategies 5 and 6

We make the following observations from our results for Strategy 5:

- **Temporal alignment across the combined datasets:** Combining two datasets representing different time periods hurt both datasets (e.g. Hogtruncated2 + Moosetruncated2). However, zero-padding the datasets to align the temporal ranges of data results in positive gains for both datasets (e.g. Hogtruncated1 + Moosetruncated2).
- **Weather feature consistency:** When the recorded weather features have a large deviation, results can degrade. For example, Bull (with AT, DT, SLP) combined with Gator (no weather features) shows a degradation in results for both datasets. Similarly, despite having the same climate zone, the differing weather features of Peacock (with AT, DT, SLP) and Wolf (with AT, DT, SLP, WD, WS, CC) lead to a results degradation in the Peacock + Wolf dataset for Peacock. Artificially reducing the weather features to align the feature spaces of the two datasets does not help either of the datasets and results are usually below the baseline (e.g. Lamb + Robintruncated, Bulltruncated + Gator, Peacock + Wolftruncated). Xing et al. [65] also reported that weather feature mismatch can lead to results degradation.
- **Building count differences:** When combining datasets of drastically different sizes, the combined model underperforms or shows minimal gains against the baseline for the larger dataset. This observation is similar to the *negative interference* experienced by high-resource languages when trained with low-resource languages in a multilingual setup [75]. However, the smaller dataset gets a positive benefit (e.g. Mouse + Rat, Crow + Robin, Bear + Fox).
- **Climate zone considerations:** Combining datasets from different climate zones does not necessarily harm performance if other factors (i.e. weather features and building counts) align. For example, despite Bobcat and Moose being from different climate zones (5B and 6A respectively), Bobcat + Moose show improvements over the baseline for both datasets. Similar observations hold for Bobcat + Moosetruncated1 and Lamb + Robin. We believe this is an important finding, because the previous research that claimed the high impact of climate zone has not considered the variation of other features [31,44,59].

As for Strategy 6, combining three datasets provides mixed results compared to the baseline. For example, the Bull + Cockatoo + Hog model merely matches the Bull 96 h baseline (both 0.602). However, scaling up to include more datasets shows promise. Training an ensemble model with all available datasets outperforms baseline for 87.5 % of datasets in 24 h forecasting, though this drops to just 43.8 % for 96 h forecasting. When compared to smaller combinations, the full ensemble performs better in 75.0 % of cases for 24 h and 50.0 % for 96 h forecasts. This suggests that larger ensembles are generally better, as evidenced by previous research as well [30]. However, there are exceptions - Bear + Fox is an example where the full ensemble model fails to derive the best result - Bear + Fox performs equally well as the full ensemble model for Bear in 24 h forecasts (both 0.291) and better in 96 h forecasts (0.367 vs. 0.393). Moreover, for datasets with very different weather feature information like Gator (no weather features), the full ensemble model underperforms compared to the baseline at both horizons (0.323 vs. 0.292 for 24 h, 0.529 vs. 0.503 for 96 h).

4.2.2. Strategy 7 and 8

Under Strategy 7, after training a target dataset with a source dataset, further fine-tuning that model with the target data is beneficial in 85.2 % of 24 h cases and 77.8 % of 96 h cases. However, when there are discrepancies among the target and source datasets, this gain can be less. In particular, when the feature spaces differ with respect to weather features between two datasets, even further fine-tuning is not enough to beat the baseline (e.g. Bulltruncated + Gator case). On the other hand, further fine-tuning the ensemble model trained with all the datasets with individual datasets (Strategy 8) always beats the baseline. However, in 18.8 % of experiments (for both 24 h and 96 h horizons), further fine-tuning results fall below the result of Strategy 6, or just on-par (e.g. Bull 96 h: 0.571 vs. 0.565, Cockatoo 24 h: both 0.432). Such results question the utility of Strategy 8, given the fine-tuning overhead.

4.3. Comparison of different transformer models

The result comparison of models trained using all the datasets on vanilla Transformer, Informer and PatchTST are shown in Fig. 5 for the following datasets: Bear, Bobcat, Bull, Cockatoo, Crow and Eagle. Graphical results for other datasets, as well as the tabular version of the results, are in the Appendix. Results of Informer and PatchTST confirm the general trend we observed in the context of vanilla Transformers - results of Strategy 8 are the best, followed by those of Strategy 6. We also note that PatchTST overall shows the best performance. On the other hand, Informer results are on par with that of vanilla Transformer, and sometimes even worse than that of vanilla Transformer.

4.4. Comparison of inference speed

To further assess real-time feasibility, we evaluated the inference speed of each architecture on an NVIDIA GeForce RTX 3090, using a batch size of 128 over 100 runs and averaging the forward pass latency. In our tests, the vanilla Transformer and FEDformer achieved inference times of approximately 15–25 ms per batch, while PatchTST hovered around 170–185 ms. Fine-tuning on multiple sources imposed only a marginal increase in latency (less than 5 % overhead), indicating that multi-source or two-stage training does not significantly affect inference speed.

5. Discussion

5.1. Answering the research questions

With the insights derived from the results reported in Section 4, we now revisit the three research questions.

RQ1: What is the best data-centric TL strategy for building energy consumption forecasting under a given data setup?

If the target domain has no data, zero-shot TL can be employed. If multiple source datasets are available, training a single source model with

Table 8
Performance summary (reported in MAE) for two- and three-dataset combinations *(Vanilla Transformer)*.

Combo	Test Set	Hor	Baseline MAE	Strategy 5/6 MAE	Imp (%)	Strategy 7/8 MAE	Imp (%)
Bull + Cockatoo + Hog	Bull	24 h	0.475	0.469	1.3 %	0.466	1.9 %
Bull + Cockatoo + Hog	Bull	96 h	0.602	0.604	-0.3 %	0.602	0.0 %
Bull + Cockatoo + Hog	Cockatoo	24 h	0.468	0.455	2.8 %	0.445	4.9 %
Bull + Cockatoo + Hog	Cockatoo	96 h	0.600	0.598	0.3 %	0.587	2.2 %
Bull + Cockatoo + Hog	Hog	24 h	0.336	0.322	4.2 %	0.318	5.4 %
Bull + Cockatoo + Hog	Hog	96 h	0.470	0.454	3.4 %	0.458	2.6 %
Bear + Fox	Bear	24 h	0.305	0.291	4.6 %	0.286	6.2 %
Bear + Fox	Bear	96 h	0.375	0.367	2.1 %	0.363	3.2 %
Bear + Fox	Fox	24 h	0.268	0.264	1.5 %	0.257	4.1 %
Bear + Fox	Fox	96 h	0.325	0.329	-1.2 %	0.320	1.5 %
Bobcat + Moose	Bobcat	24 h	0.415	0.399	3.9 %	0.393	5.3 %
Bobcat + Moose	Bobcat	96 h	0.543	0.519	4.4 %	0.507	6.6 %
Bobcat + Moose	Moose	24 h	0.294	0.278	5.4 %	0.271	7.8 %
Bobcat + Moose	Moose	96 h	0.377	0.364	3.4 %	0.355	5.8 %
Bull + Gator	Bull	24 h	0.475	0.480	-1.1 %	0.467	1.7 %
Bull + Gator	Bull	96 h	0.602	0.602	0.0 %	0.596	1.0 %
Bull + Gator	Gator	24 h	0.292	0.312	-6.8 %	0.289	1.0 %
Bull + Gator	Gator	96 h	0.503	0.525	-4.4 %	0.512	-1.8 %
Crow + Robin	Crow	24 h	0.359	0.313	12.8 %	0.311	13.4 %
Crow + Robin	Crow	96 h	0.476	0.423	11.1 %	0.424	10.9 %
Crow + Robin	Robin	24 h	0.254	0.255	-0.4 %	0.253	0.4 %
Crow + Robin	Robin	96 h	0.344	0.339	1.5 %	0.335	2.6 %
Hog + Moose	Hog	24 h	0.336	0.323	3.9 %	0.322	4.2 %
Hog + Moose	Hog	96 h	0.470	0.464	1.3 %	0.463	1.5 %
Hog + Moose	Moose	24 h	0.294	0.297	-1.0 %	0.280	4.8 %
Hog + Moose	Moose	96 h	0.377	0.402	-6.6 %	0.367	2.7 %
Lamb + Robin	Lamb	24 h	0.208	0.185	11.1 %	0.191	8.2 %
Lamb + Robin	Lamb	96 h	0.269	0.260	3.3 %	0.251	6.7 %
Lamb + Robin	Robin	24 h	0.254	0.248	2.4 %	0.247	2.8 %
Lamb + Robin	Robin	96 h	0.344	0.338	1.7 %	0.337	2.0 %
Mouse + Rat	Mouse	24 h	0.365	0.348	4.7 %	0.338	7.4 %
Mouse + Rat	Mouse	96 h	0.532	0.475	10.7 %	0.462	13.2 %
Mouse + Rat	Rat	24 h	0.293	0.294	-0.3 %	0.294	-0.3 %
Mouse + Rat	Rat	96 h	0.413	0.414	-0.2 %	0.413	0.0 %
Peacock + Wolftruncated1	Peacock	24 h	0.316	0.329	-4.1 %	0.320	-1.3 %
Peacock + Wolftruncated1	Peacock	96 h	0.411	0.435	-5.8 %	0.414	-0.7 %
Peacock + Wolftruncated1	Wolf	24 h	0.387	0.376	2.8 %	0.374	3.4 %
Peacock + Wolftruncated1	Wolf	96 h	0.453	0.462	-2.0 %	0.456	-0.7 %
Eagletruncated1 + Robin	Eagle	24 h	0.384	0.368	4.2 %	0.363	5.5 %
Eagletruncated1 + Robin	Eagle	96 h	0.481	0.428	11.0 %	0.427	11.2 %
Eagletruncated1 + Robin	Robin	24 h	0.254	0.368	-44.9 %	0.254	0.0 %
Eagletruncated1 + Robin	Robin	96 h	0.344	0.502	-45.9 %	0.336	2.3 %
Bear + Foxtruncated1	Bear	24 h	0.305	0.295	3.3 %	0.289	5.2 %
Bear + Foxtruncated1	Bear	96 h	0.375	0.371	1.1 %	0.363	3.2 %
Bear + Foxtruncated1	Foxtruncated1	24 h	0.268	0.273	-1.9 %	0.259	3.4 %
Bear + Foxtruncated1	Foxtruncated1	96 h	0.325	0.336	-3.4 %	0.320	1.5 %
Bobcat + Moosetruncated1	Bobcat	24 h	0.415	0.400	3.6 %	0.398	4.1 %
Bobcat + Moosetruncated1	Bobcat	96 h	0.543	0.521	4.1 %	0.512	5.7 %
Bobcat + Moosetruncated1	Moose	24 h	0.294	0.282	4.1 %	0.274	6.8 %
Bobcat + Moosetruncated1	Moose	96 h	0.377	0.370	1.9 %	0.356	5.6 %
Eagle + Robin	Eagle	24 h	0.384	0.368	4.2 %	0.363	5.5 %
Eagle + Robin	Eagle	96 h	0.481	0.428	11.0 %	0.427	11.2 %
Eagle + Robin	Robin	24 h	0.254	0.368	-44.9 %	0.254	0.0 %
Eagle + Robin	Robin	96 h	0.344	0.502	-45.9 %	0.336	2.3 %
Bulltruncated1 + Gator	Bull	24 h	0.475	0.483	-1.7 %	0.466	1.9 %
Bulltruncated1 + Gator	Bull	96 h	0.602	0.611	-1.5 %	0.591	1.8 %
Bulltruncated1 + Gator	Gator	24 h	0.292	0.308	-5.5 %	0.287	1.7 %
Bulltruncated1 + Gator	Gator	96 h	0.503	0.524	-4.2 %	0.512	-1.8 %
Peacock + Wolf	Peacock	24 h	0.316	0.329	-4.1 %	0.320	-1.3 %
Peacock + Wolf	Peacock	96 h	0.411	0.429	-4.4 %	0.412	-0.2 %
Peacock + Wolf	Wolf	24 h	0.387	0.379	2.1 %	0.381	1.6 %
Peacock + Wolf	Wolf	96 h	0.453	0.455	-0.4 %	0.458	-1.1 %
Lamb + Robintruncated1	Lamb	24 h	0.208	0.218	-4.8 %	0.197	5.3 %
Lamb + Robintruncated1	Lamb	96 h	0.269	0.304	-13.0 %	0.273	-1.5 %
Lamb + Robintruncated1	Robintruncated1	24 h	0.254	0.271	-6.7 %	0.252	0.8 %
Lamb + Robintruncated1	Robintruncated1	96 h	0.344	0.358	-4.1 %	0.338	1.7 %
Hogtruncated1 + Moosetruncated2	Hogtruncated1	24 h	0.417	0.364	12.7 %	0.367	12.0 %
Hogtruncated1 + Moosetruncated2	Hogtruncated1	96 h	0.606	0.485	20.0 %	0.492	18.8 %
Hogtruncated1 + Moosetruncated2	Moosetruncated2	24 h	0.254	0.242	4.7 %	0.239	5.9 %
Hogtruncated1 + Moosetruncated2	Moosetruncated2	96 h	0.329	0.331	-0.6 %	0.327	0.6 %
Hogtruncated2 + Moosetruncated2	Hogtruncated2	24 h	0.387	0.429	-10.9 %	0.390	-0.8 %
Hogtruncated2 + Moosetruncated2	Hogtruncated2	96 h	0.539	0.562	-4.3 %	0.520	3.5 %
Hogtruncated2 + Moosetruncated2	Moosetruncated2	24 h	0.254	0.242	4.7 %	0.239	5.9 %
Hogtruncated2 + Moosetruncated2	Moosetruncated2	96 h	0.329	0.331	-0.6 %	0.327	0.6 %

Table 9
Performance summary (reported in MAE) for the model trained with all the datasets *(Vanilla Transformer)*.

Dataset	Horizon	Baseline MAE	Strategy 6 MAE	Imp (%)	Strategy 8 MAE	Imp (%)
Bear	24 h	0.305	0.291	+4.6 %	0.283	+7.2 %
Bear	96 h	0.375	0.393	−4.8 %	0.367	+2.1 %
Bobcat	24 h	0.415	0.366	+11.8 %	0.361	+13.0 %
Bobcat	96 h	0.543	0.463	+14.7 %	0.458	+15.7 %
Bull	24 h	0.475	0.450	+5.3 %	0.447	+5.9 %
Bull	96 h	0.602	0.565	+6.1 %	0.571	+5.1 %
Cockatoo	24 h	0.468	0.432	+7.7 %	0.432	+7.7 %
Cockatoo	96 h	0.600	0.573	+4.5 %	0.567	+5.5 %
Crow	24 h	0.359	0.304	+15.3 %	0.303	+15.6 %
Crow	96 h	0.476	0.430	+9.7 %	0.424	+10.9 %
Eagle	24 h	0.384	0.347	+9.6 %	0.348	+9.4 %
Eagle	96 h	0.481	0.415	+13.7 %	0.420	+12.7 %
Fox	24 h	0.268	0.263	+1.9 %	0.254	+5.2 %
Fox	96 h	0.325	0.351	−8.0 %	0.321	+1.2 %
Gator	24 h	0.292	0.323	−10.6 %	0.288	+1.4 %
Gator	96 h	0.503	0.529	−5.2 %	0.491	+2.4 %
Hog	24 h	0.336	0.296	+11.9 %	0.294	+12.5 %
Hog	96 h	0.470	0.414	+11.9 %	0.427	+9.1 %
Lamb	24 h	0.208	0.182	+12.5 %	0.181	+13.0 %
Lamb	96 h	0.269	0.278	−3.3 %	0.251	+6.7 %
Moose	24 h	0.294	0.265	+9.9 %	0.255	+13.3 %
Moose	96 h	0.377	0.378	−0.3 %	0.342	+9.3 %
Mouse	24 h	0.365	0.329	+9.9 %	0.325	+11.0 %
Mouse	96 h	0.532	0.447	+16.0 %	0.441	+17.1 %
Peacock	24 h	0.316	0.312	+1.3 %	0.303	+4.1 %
Peacock	96 h	0.411	0.412	−0.2 %	0.388	+5.6 %
Rat	24 h	0.293	0.280	+4.4 %	0.282	+3.8 %
Rat	96 h	0.413	0.401	+2.9 %	0.394	+4.6 %
Robin	24 h	0.254	0.247	+2.8 %	0.238	+6.3 %
Robin	96 h	0.344	0.359	−4.4 %	0.324	+5.8 %
Wolf	24 h	0.387	0.379	+2.1 %	0.363	+6.2 %
Wolf	96 h	0.453	0.472	−4.2 %	0.448	+1.1 %

all these data is generally more beneficial than using a single source. In other words, Strategy 2 should be favoured over Strategy 1. We believe this is an important finding, given that none of the prior work we referred to experimented with Strategy 2.

Similarly, if there is some target data available, training a model using that data along with the available source dataset(s), and then further fine-tuning with the target dataset produces the optimal result most of the time. In other words, strategies 6 and 8 are more beneficial than strategies 5 and 7 (respectively). Similar to the observations in zero-shot TL, strategy 7 is generally better than strategy 5, and strategy 8 is generally better than strategy 6. In other words, using all the available source datasets to train a model is beneficial over relying on individual datasets. We believe this is an important finding, given that no related research used Strategy 7, and Strategy 8 was used less frequently than Strategy 6 (see Table 1).

Another decision to make is how many source datasets to use for training the initial model. While more datasets would be better, this causes a computational overhead. Wei et al. [66] concluded that three source domains is the best. However, blindly combining the available source datasets can harm the target, and the exact benefit of these strategies depends on the feature spaces of source and target domains, as discussed under RQ2.

RQ2: What specific features of building energy datasets (e.g. ambient weather features, climate zone, data volume) influence the effectiveness of different data-centric TL strategies?

Our results indicate that the nature of the ambient weather features has the greatest impact on TL. In other words, using source dataset(s) that have very distinct ambient weather features from the target does not help any of the TL strategies, except strategy 8 when all the datasets are used. The other factor is the number of building records in a dataset. For datasets with a large number of building records, TL using much smaller datasets is not productive. However, it is always beneficial to use TL with very large datasets to boost performance for smaller datasets. The climate zone does show some impact, however if weather features and the number of buildings are similar, the impact of climate zone is negligible. The difference in temporal change of source and target datasets also shows an adverse impact, which can be mitigated to a certain extent by zero-padding data augmentation. It is always possible to experiment with more advanced data interpolation techniques here.

RQ3: How does the performance of various data-centric TL strategies differ when applied to advanced Transformer architectures specifically designed for time series forecasting, compared to the vanilla Transformer?

Our experiments with the vanilla Transformer, PatchTST, and FEDformer indicate that TL strategies behave similarly across the three models. Out of these, PatchTST reported the best forecasting accuracy on most datasets.

Though PatchTST exhibits longer absolute inference times, it is still viable for real-time prediction, especially when employing moderately sized batches or lowering the batch size for single-sample inference. Overall, despite its higher latency, PatchTST offers the strongest predictive performance among our tested models and remains a competitive option for real-time building energy forecasting.

5.2. Recommendation

To gain the maximum benefit of TL, we recommend researchers follow a step-wise approach. For a given target domain, the most suitable subset of source datasets can be selected by inspecting the weather features, number of building data, etc., as mentioned above. To further validate the efficacy of the selected datasets, Strategy 1 (if no target data is available for model training) or Strategy 5 (if there is target data available for model training) can be used on individual source datasets. This way, the best set of source datasets can be selected for model training. If there is no target data for model training, a model trained with

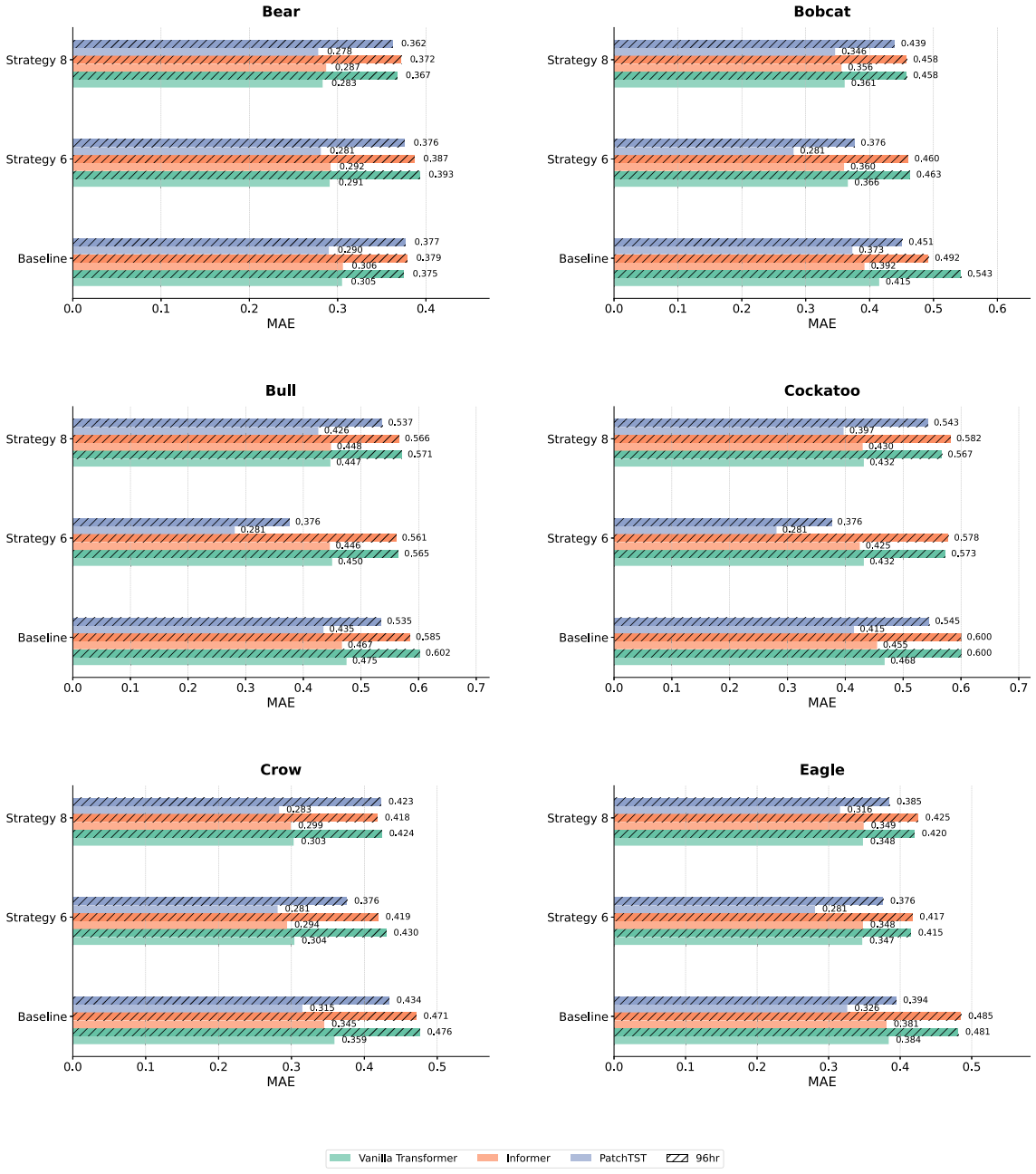


Fig. 5. MAE Comparison for the Ensemble Model Across Different Transformer Architectures for the Following Datasets: Bear, Bobcat, Bull, Cockatoo, Crow and Eagle.

the selected source datasets can be used in zero-shot TL. If there is target data for model training, this data can be combined with the source data sets to train a model, which can be further fine-tuned with target data. However, we recommend testing the combined model both before and after fine-tuning, because we observed a results degradation after fine-tuning for a small number of cases. Finally, we can recommend the use of PatchTST over FEDformer and vanilla Transformer.

5.3. Limitations and future work

Although we experimented with 16 datasets, all these datasets belong to North American and European countries. Our observations might not hold for datasets obtained from other parts of the world. In the future, we expect to find data from these under-represented regions and re-validate our observations.

To keep our experiments at a manageable level, we used only three combinations of datasets: 2, 3, and 16. However, for a given target dataset, the optimal dataset combination may involve a different number of datasets. As discussed in the recommendation, we invite researchers to experiment with these different combinations.

To keep the experiment space manageable, we only considered 24 h and 96 h forecast horizons. We plan to experiment with different forecast horizons in the future.

We followed a pragmatic approach to identify the impact of datasets - we prepared datasets with different feature combinations and compared them based on their performance. Moreover, our analysis was done at a higher granularity. In other words, we considered individual datasets as a single source domain, although a dataset may include data from different building types. It is always possible to do more fine-grained source selection, by considering building type. Building energy consumption

profiles may vary even across buildings of the same type. There also can be other nuances such as the stability of energy consumption [39]. Therefore, in the future, we plan to incorporate similarity measurement indexes discussed in Section 2 to select the most suitable source datasets.

We considered only three Transformer architectures. However, there are other Transformer architectures designed for time series prediction. With the advancements of Foundation models such as TimeGPT [76] and Lag-llama [77], it is now possible to try TL on these already pre-trained models. However, fine-tuning these models requires significant computational resources, which we currently do not have. Thus, these experiments are kept for the future.

6. Conclusion

While TL has been explored for the task of building energy forecasting, a comparative study on different data-centric TL was missing. Many past studies have focused on experimenting novel DL architectures for the task of building energy comsumption prediction, and consequently, less focus was given to determining the best use of available datasets. In response to this research gap, we carried out a large-scale empirical study on the effectiveness of different TL strategies on Transformer architectures. Our results show that combining multiple source datasets under a zero-shot setup reduces the Mean Absolute Error (MAE) by an average of 15.9 % for 24 h forecasts with the vanilla Transformer, compared to single-source baselines. Further fine-tuning these multi-source models with target-domain data yields an additional 3–5 % improvement. We also note that PatchTST performs better than the vanilla Transformer and Informer. However, while TL is generally beneficial, a clear understanding of the datasets is needed to determine which exact data-centric TL strategy to use. Based on our observations, we made several recommendations for researchers who intend to employ TL for building energy consumption forecasting. These recommendations contribute toward laying the foundation for a better understanding of TL for building energy consumption forecasting. This empirical study can be extended further by considering different forecasting horizons, building types, newer Transformer architectures, etc. We plan to focus on these extensions in the future and invite other researchers to contribute to enhance our understanding of the exact impact of TL for building energy forecasting.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Grammarly to identify spelling and grammar issues in the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Data availability

We used a publicly available dataset.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Robert Spencer: Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization; **Surangika Ranathunga:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Mikael Boulic:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization; **Andries (Hennie) van Heerden:** Writing – review & editing, Supervision, Conceptualization; **Teo Susnjak:** Writing – review & editing, Supervision

Acknowledgements

This research was funded by a research grant from the College of Sciences, Massey University, New Zealand.

We sincerely thank Matthias Hertel for providing their codebase to initiate this research, and Clayton Miller for the productive discussions at the beginning of this project.

Appendix A. MSE results

Tables A.10 and A.11 report the zero-shot MSE results, respectively. These correspond to Tables 6 and 7 in Section 4.1. Table A.12 shows MSE results for two- and three-dataset combinations and Table A.13 shows MSE results for the model trained with all the datasets (for vanilla Transformer). These correspond to Tables 8 and 9 (respectively). MAE and MSE results for Informer and PatchTST results are shown in Tables A.14, A.15, A.16, A.17.

Table A.10
Average MSE for 24 h horizon zero-shot forecasting.

Model	Bear	Bobcat	Bull	Cockatoo	Crow	Eagle	Fox	Gator	Hog	Lamb	Moose	Mouse	Peacock	Rat	Robin	Wolf
Bear	0.219	0.341	0.478	0.454	0.304	0.310	0.209	0.460	0.267	0.178	0.228	0.302	0.221	0.238	0.170	0.425
Bobcat	0.355	0.353	0.504	0.435	0.309	0.440	0.312	0.452	0.273	0.253	0.267	0.345	0.310	0.322	0.274	0.485
Bull	0.377	0.445	0.479	0.492	0.349	0.417	0.330	0.540	0.295	0.236	0.365	0.393	0.312	0.321	0.299	0.525
Cockatoo	0.761	0.711	0.647	0.426	0.376	0.890	0.665	0.496	0.297	0.369	0.596	0.397	0.630	0.544	0.603	0.715
Crow	0.434	0.627	0.744	0.479	0.293	0.847	0.451	0.457	0.393	0.354	0.420	0.365	0.476	0.449	0.374	0.501
Eagle	0.354	0.426	0.505	0.418	0.289	**0.234**	0.327	0.420	0.265	0.215	0.444	0.299	0.301	0.316	0.222	0.496
Fox	0.221	0.331	0.452	0.448	0.301	0.280	0.181	0.460	0.254	0.167	0.222	0.309	0.211	0.221	0.166	0.432
Gator	1.249	0.920	0.680	0.559	0.560	1.152	1.045	**0.270**	0.336	0.467	0.836	0.585	0.966	0.699	0.995	1.168
Hog	0.450	0.503	0.519	0.432	0.297	0.621	0.408	0.455	0.247	0.234	0.425	0.319	0.387	0.373	0.336	0.546
Lamb	0.345	0.445	0.578	0.474	0.290	0.474	0.324	0.474	0.278	0.166	0.252	0.349	0.324	0.312	0.244	0.423
Moose	0.457	0.557	0.715	0.481	0.313	0.898	0.443	0.494	0.334	0.269	0.227	0.394	0.480	0.419	0.381	0.459
Mouse	0.397	0.547	0.563	0.437	0.300	0.656	0.398	0.436	0.280	0.272	0.453	0.309	0.400	0.402	0.274	0.509
Peacock	0.248	0.418	0.594	0.508	0.297	0.356	0.240	0.478	0.303	0.201	0.232	0.326	**0.198**	0.270	0.178	0.423
Rat	0.243	0.347	0.501	0.443	0.287	0.330	0.206	0.439	0.260	0.175	0.217	0.311	0.230	**0.207**	0.203	0.423
Robin	0.265	0.407	0.522	0.423	0.275	0.348	0.267	0.434	0.258	0.188	0.221	0.295	0.248	0.281	0.155	0.432
Wolf	0.399	0.500	0.681	0.535	0.336	0.521	0.362	0.530	0.351	0.311	0.315	0.376	0.391	0.394	0.296	0.399
Hogtrunc2	0.510	0.515	0.510	0.411	0.309	0.613	0.448	0.426	0.239	0.245	0.472	0.335	0.420	0.400	0.365	0.576
Hogtrunc	0.749	0.619	0.561	0.487	0.415	0.732	0.635	0.466	0.309	0.324	0.584	0.427	0.565	0.486	0.590	0.737
Moosetrunc2	0.696	0.694	0.664	0.554	0.427	0.986	0.666	0.492	0.340	0.325	0.395	0.466	0.665	0.515	0.623	0.746
Bear + Fox	**0.207**	0.327	0.459	0.435	0.293	0.289	**0.177**	0.450	0.250	0.167	0.220	0.297	0.209	0.218	0.166	0.424
Bobcat + Moose	0.307	0.333	0.525	**0.396**	0.264	0.453	0.282	0.426	0.250	0.203	**0.197**	0.305	0.285	0.291	0.222	0.415
Bull + Cockatoo	0.387	0.437	0.468	0.428	0.308	0.453	0.339	0.486	0.276	0.225	0.335	0.343	0.311	0.324	0.307	0.476
Bull + Gator	0.454	0.474	0.480	0.482	0.335	0.567	0.401	0.289	0.282	0.233	0.395	0.358	0.354	0.354	0.356	0.545
Bull + Hog	0.358	0.405	0.453	0.431	0.297	0.448	0.312	0.469	0.242	0.209	0.302	0.332	0.292	0.294	0.284	0.467
Cockatoo + Hog	0.428	0.486	0.537	0.403	0.279	0.647	0.392	0.445	0.234	0.236	0.377	0.317	0.382	0.359	0.334	0.515
Crow + Robin	0.265	0.410	0.569	0.417	**0.257**	0.365	0.275	0.417	0.255	0.186	0.218	**0.288**	0.248	0.286	0.155	0.413
Eagle + Robin	0.401	0.452	0.509	0.422	0.316	0.258	0.361	0.444	0.263	0.229	0.474	0.322	0.335	0.337	0.264	0.548
Hog + Moose	0.368	0.436	0.559	0.426	0.283	0.643	0.348	0.421	0.235	0.211	0.215	0.319	0.357	0.329	0.274	0.444
Lamb + Robin	0.251	0.374	0.543	0.438	0.272	0.312	0.237	0.421	0.258	**0.148**	0.216	0.315	0.236	0.255	**0.152**	0.410
Mouse + Rat	0.239	0.347	0.504	0.432	0.281	0.319	0.204	0.432	0.254	0.175	0.214	0.300	0.230	0.208	0.192	0.421
Peacock + Wolf	0.275	0.386	0.566	0.476	0.292	0.352	0.252	0.454	0.287	0.199	0.236	0.319	0.216	0.279	0.186	0.392
Bear + Foxtrunc	0.209	**0.324**	0.452	0.437	0.294	0.291	0.184	0.455	0.253	0.166	0.216	0.297	0.209	0.222	0.162	0.418
Bobcat + Moosetrunc	0.313	0.335	0.522	0.397	0.265	0.456	0.287	0.422	0.255	0.218	0.200	0.307	0.292	0.296	0.230	0.422
Bulltrunc + Gator	0.460	0.484	0.484	0.490	0.340	0.593	0.409	0.290	0.285	0.230	0.415	0.357	0.359	0.362	0.361	0.538
Eagletrunc + Robin	0.401	0.452	0.509	0.422	0.316	0.258	0.361	0.444	0.263	0.229	0.474	0.322	0.335	0.337	0.264	0.548
Hogtrunc2 + Moosetrunc2	0.504	0.593	0.588	0.513	0.369	0.809	0.485	0.451	0.297	0.261	0.337	0.372	0.538	0.422	0.419	0.627
Hogtrunc + Moosetrunc2	0.504	0.593	0.588	0.513	0.369	0.809	0.485	0.451	0.297	0.261	0.337	0.372	0.538	0.422	0.419	0.627
Lamb + Robintrunc	0.266	0.372	0.499	0.438	0.286	0.321	0.247	0.416	0.247	0.163	0.231	0.312	0.249	0.261	0.170	0.423
Peacock + Wolftrunc	0.272	0.395	0.543	0.483	0.308	0.346	0.249	0.455	0.291	0.202	0.251	0.326	0.215	0.276	0.188	**0.382**
Bull + Cockatoo + Hog	0.357	0.409	**0.450**	0.410	0.285	0.477	0.322	0.431	**0.233**	0.205	0.295	0.315	0.299	0.300	0.284	0.453

Table A.11
Average MSE for 96 h horizon zero-shot forecasting.

Model	Bear	Bobcat	Bull	Cockatoo	Crow	Eagle	Fox	Gator	Hog	Lamb	Moose	Mouse	Peacock	Rat	Robin	Wolf
Bear	0.323	0.489	0.683	0.653	0.466	0.443	0.299	0.733	0.442	0.352	0.351	0.514	0.338	0.412	0.290	0.555
Bobcat	0.578	0.541	0.697	0.649	0.473	0.643	0.480	0.751	0.431	0.477	0.414	0.553	0.535	0.547	0.457	0.718
Bull	0.634	0.643	0.717	0.730	0.498	0.641	0.523	0.824	0.451	0.379	0.572	0.611	0.538	0.530	0.537	0.735
Cockatoo	0.976	0.849	0.796	0.611	0.541	1.028	0.823	0.801	0.452	0.544	0.696	0.589	0.829	0.717	0.793	0.821
Crow	0.653	0.921	1.088	0.725	0.446	1.128	0.624	0.864	0.764	0.758	0.664	0.559	0.799	0.745	0.608	0.632
Eagle	0.572	0.587	0.775	0.617	0.450	**0.315**	0.490	0.714	0.475	0.364	0.654	0.491	0.491	0.512	0.377	0.683
Fox	0.336	0.472	0.662	0.616	0.427	0.396	**0.252**	0.707	0.411	0.304	0.312	0.488	0.330	0.375	0.285	0.558
Gator	1.141	0.884	0.828	0.704	0.630	1.043	0.943	**0.544**	0.486	0.518	0.819	0.672	0.884	0.732	0.942	1.051
Hog	0.669	0.686	0.710	0.643	0.451	0.795	0.568	0.762	0.426	0.367	0.605	0.518	0.557	0.556	0.526	0.719
Lamb	0.479	0.608	0.876	0.707	0.454	0.664	0.445	0.799	0.516	0.249	0.375	0.538	0.525	0.493	0.363	0.564
Moose	0.683	0.808	1.025	0.708	0.490	1.230	0.613	0.837	0.556	0.467	0.334	0.592	0.806	0.643	0.584	0.601
Mouse	0.575	0.903	0.938	0.738	0.485	0.887	0.557	0.907	0.549	0.860	0.595	0.575	0.774	0.731	0.482	0.696
Peacock	0.387	0.534	0.831	0.714	0.467	0.489	0.342	0.806	0.491	0.328	0.358	0.499	0.327	0.448	0.292	0.560
Rat	0.384	0.475	0.731	0.660	0.450	0.481	0.312	0.773	0.420	0.268	0.327	0.505	0.364	0.371	0.335	0.573
Robin	0.400	0.592	0.766	0.646	0.428	0.493	0.378	0.743	0.453	0.299	0.347	0.515	0.405	0.450	0.268	0.548
Wolf	0.553	0.735	0.927	0.782	0.512	0.791	0.528	0.883	0.640	0.837	0.454	0.676	0.630	0.701	0.489	**0.477**
Hogtrunc2	0.919	0.753	0.739	0.630	0.521	0.891	0.746	0.759	0.442	0.459	0.724	0.590	0.678	0.653	0.725	0.866
Hogtrunc	1.186	0.854	0.775	0.685	0.690	1.061	0.989	0.907	0.483	0.521	0.825	0.710	0.802	0.746	0.967	0.998
Moosetrunc2	0.786	0.901	1.004	0.919	0.662	1.184	0.763	0.923	0.548	0.540	0.525	0.706	0.778	0.735	0.707	0.860
Bear + Fox	**0.313**	0.449	**0.648**	0.609	0.423	0.384	0.255	0.687	0.406	0.277	0.317	0.479	0.311	0.372	0.271	0.562
Bobcat + Moose	0.488	0.508	0.741	0.608	0.425	0.623	0.419	0.723	0.437	0.387	**0.302**	0.500	0.473	0.500	0.376	0.588
Bull + Cockatoo	0.610	0.624	0.714	0.649	0.485	0.691	0.522	0.837	0.436	0.386	0.531	0.545	0.507	0.536	0.537	0.651
Bull + Gator	0.650	0.654	0.686	0.676	0.496	0.737	0.559	0.580	0.431	0.374	0.616	0.549	0.540	0.553	0.530	0.706
Bull + Hog	0.593	0.602	0.701	0.683	0.473	0.693	0.490	0.807	0.424	0.354	0.490	0.543	0.484	0.506	0.483	0.698
Cockatoo + Hog	0.793	0.741	0.777	0.623	0.455	0.882	0.629	0.806	0.437	0.415	0.664	0.539	0.641	0.589	0.598	0.791
Crow + Robin	0.393	0.542	0.776	0.603	**0.390**	0.488	0.366	0.700	0.427	0.297	0.328	**0.469**	0.399	0.440	**0.258**	0.521
Eagle + Robin	0.657	0.632	0.740	0.608	0.483	0.390	0.546	0.762	0.443	0.391	0.687	0.510	0.539	0.540	0.453	0.729
Hog + Moose	0.540	0.590	0.744	0.637	0.437	0.841	0.480	0.768	0.425	0.340	0.332	0.502	0.492	0.495	0.445	0.599
Lamb + Robin	0.376	0.501	0.732	0.625	0.409	0.444	0.333	0.703	0.417	**0.235**	0.320	0.488	0.377	0.401	0.259	0.533
Mouse + Rat	0.372	0.472	0.730	0.648	0.428	0.458	0.307	0.747	0.434	0.271	0.319	0.498	0.356	**0.369**	0.325	0.564
Peacock + Wolf	0.418	0.576	0.805	0.655	0.431	0.487	0.373	0.768	0.486	0.395	0.360	0.495	0.356	0.489	0.301	0.486
Bear + Foxtrunc	0.314	**0.439**	0.653	0.624	0.426	0.406	0.261	0.695	0.420	0.301	0.331	0.481	**0.309**	0.377	0.284	0.568
Bobcat + Moosetrunc	0.494	0.515	0.732	**0.600**	0.426	0.641	0.424	0.718	0.427	0.386	0.304	0.501	0.494	0.502	0.385	0.597
Bulltrunc + Gator	0.637	0.642	0.708	0.698	0.508	0.778	0.562	0.574	0.456	0.359	0.620	0.535	0.523	0.545	0.526	0.672
Eagletrunc + Robin	0.657	0.632	0.740	0.608	0.483	0.390	0.546	0.762	0.443	0.391	0.687	0.510	0.539	0.540	0.453	0.729
Hogtrunc2 + Moosetrunc2	0.722	0.729	0.776	0.727	0.565	0.921	0.623	0.911	0.461	0.388	0.510	0.608	0.649	0.579	0.606	0.795
Hogtrunc + Moosetrunc2	0.722	0.729	0.776	0.727	0.565	0.921	0.623	0.911	0.461	0.388	0.510	0.608	0.649	0.579	0.606	0.795
Lamb + Robintrunc	0.411	0.570	0.737	0.629	0.420	0.465	0.359	0.696	**0.400**	0.294	0.334	0.501	0.446	0.435	0.286	0.548
Peacock + Wolftrunc	0.428	0.625	0.856	0.711	0.465	0.501	0.380	0.807	0.545	0.434	0.397	0.527	0.365	0.518	0.318	0.502
Bull + Cockatoo + Hog	0.577	0.580	0.706	0.632	0.430	0.689	0.475	0.757	0.405	0.345	0.474	0.496	0.461	0.484	0.471	0.646

Table A.12
MSE results for two- and three-dataset combinations *(Vanilla Transformer)*.

Combo	Test	Hor	Baseline	Strategy 5/6	Imp	Strategy 7/8	Imp
Bull + Cockatoo + Hog	Bull	24 h	0.479	0.450	6.1 %	0.455	5.0 %
Bull + Cockatoo + Hog	Bull	96 h	0.717	0.706	1.5 %	0.716	0.1 %
Bull + Cockatoo + Hog	Cockatoo	24 h	0.426	0.403	5.4 %	0.410	3.8 %
Bull + Cockatoo + Hog	Cockatoo	96 h	0.611	0.623	−2.0 %	0.602	1.5 %
Bull + Cockatoo + Hog	Hog	24 h	0.247	0.233	5.7 %	0.229	7.3 %
Bull + Cockatoo + Hog	Hog	96 h	0.426	0.405	4.9 %	0.412	3.3 %
Bear + Fox	Bear	24 h	0.219	0.207	5.5 %	0.200	8.7 %
Bear + Fox	Bear	96 h	0.323	0.313	3.1 %	0.307	5.0 %
Bear + Fox	Fox	24 h	0.181	0.177	2.2 %	0.170	6.1 %
Bear + Fox	Fox	96 h	0.252	0.255	−1.2 %	0.246	2.4 %
Bobcat + Moose	Bobcat	24 h	0.353	0.333	5.7 %	0.328	7.1 %
Bobcat + Moose	Bobcat	96 h	0.541	0.508	6.1 %	0.491	9.2 %
Bobcat + Moose	Moose	24 h	0.227	0.197	13.2 %	0.192	15.4 %
Bobcat + Moose	Moose	96 h	0.334	0.302	9.6 %	0.298	10.8 %
Bull + Gator	Bull	24 h	0.479	0.480	−0.2 %	0.459	4.2 %
Bull + Gator	Bull	96 h	0.717	0.686	4.3 %	0.691	3.6 %
Bull + Gator	Gator	24 h	0.270	0.289	−7.0 %	0.275	−1.9 %
Bull + Gator	Gator	96 h	0.544	0.580	−6.6 %	0.563	−3.5 %
Crow + Robin	Crow	24 h	0.293	0.257	12.3 %	0.250	14.7 %
Crow + Robin	Crow	96 h	0.446	0.390	12.6 %	0.387	13.2 %
Crow + Robin	Robin	24 h	0.155	0.155	0.0 %	0.154	0.6 %
Crow + Robin	Robin	96 h	0.268	0.258	3.7 %	0.256	4.5 %
Hog + Moose	Hog	24 h	0.247	0.235	4.9 %	0.237	4.0 %
Hog + Moose	Hog	96 h	0.426	0.425	0.2 %	0.428	−0.5 %
Hog + Moose	Moose	24 h	0.227	0.215	5.3 %	0.204	10.1 %
Hog + Moose	Moose	96 h	0.334	0.332	0.6 %	0.313	6.3 %
Lamb + Robin	Lamb	24 h	0.166	0.148	10.8 %	0.150	9.6 %
Lamb + Robin	Lamb	96 h	0.249	0.235	5.6 %	0.232	6.8 %
Lamb + Robin	Robin	24 h	0.155	0.152	1.9 %	0.154	0.6 %
Lamb + Robin	Robin	96 h	0.268	0.259	3.4 %	0.261	2.6 %
Mouse + Rat	Mouse	24 h	0.309	0.300	2.9 %	0.284	8.1 %
Mouse + Rat	Mouse	96 h	0.575	0.498	13.4 %	0.476	17.2 %
Mouse + Rat	Rat	24 h	0.207	0.208	−0.5 %	0.210	−1.4 %
Mouse + Rat	Rat	96 h	0.371	0.369	0.5 %	0.373	−0.5 %
Peacock + Wolftruncated1	Peacock	24 h	0.198	0.215	−8.6 %	0.204	−3.0 %
Peacock + Wolftruncated1	Peacock	96 h	0.327	0.365	−11.6 %	0.336	−2.8 %
Peacock + Wolftruncated1	Wolf	24 h	0.399	0.382	4.3 %	0.383	4.0 %
Peacock + Wolftruncated1	Wolf	96 h	0.477	0.502	−5.2 %	0.492	−3.1 %
Eagletruncated1 + Robin	Eagle	24 h	0.311	0.276	11.3 %	0.271	12.9 %
Eagletruncated1 + Robin	Eagle	96 h	0.461	0.357	22.6 %	0.357	22.6 %
Eagletruncated1 + Robin	Robin	24 h	0.155	0.264	−70.3 %	0.154	0.6 %
Eagletruncated1 + Robin	Robin	96 h	0.268	0.453	−69.0 %	0.252	6.0 %
Bear + Foxtruncated1	Bear	24 h	0.219	0.209	4.6 %	0.203	7.3 %
Bear + Foxtruncated1	Bear	96 h	0.323	0.314	2.8 %	0.306	5.3 %
Bear + Foxtruncated1	Foxtruncated1	24 h	0.181	0.184	−1.7 %	0.172	5.0 %
Bear + Foxtruncated1	Foxtruncated1	96 h	0.252	0.261	−3.6 %	0.244	3.2 %
Bobcat + Moosetruncated1	Bobcat	24 h	0.353	0.335	5.1 %	0.333	5.7 %
Bobcat + Moosetruncated1	Bobcat	96 h	0.541	0.515	4.8 %	0.498	7.9 %
Bobcat + Moosetruncated1	Moose	24 h	0.227	0.200	11.9 %	0.196	13.7 %
Bobcat + Moosetruncated1	Moose	96 h	0.334	0.304	9.0 %	0.299	10.5 %
Eagle + Robin	Eagle	24 h	0.311	0.276	11.3 %	0.271	12.9 %
Eagle + Robin	Eagle	96 h	0.461	0.357	22.6 %	0.357	22.6 %
Eagle + Robin	Robin	24 h	0.155	0.264	−70.3 %	0.154	0.6 %
Eagle + Robin	Robin	96 h	0.268	0.453	−69.0 %	0.252	6.0 %
Bulltruncated1 + Gator	Bull	24 h	0.479	0.484	−1.0 %	0.457	4.6 %
Bulltruncated1 + Gator	Bull	96 h	0.717	0.708	1.3 %	0.681	5.0 %
Bulltruncated1 + Gator	Gator	24 h	0.270	0.290	−7.4 %	0.272	−0.7 %
Bulltruncated1 + Gator	Gator	96 h	0.544	0.574	−5.5 %	0.564	−3.7 %
Peacock + Wolf	Peacock	24 h	0.198	0.216	−9.1 %	0.204	−3.0 %
Peacock + Wolf	Peacock	96 h	0.327	0.356	−8.9 %	0.333	−1.8 %
Peacock + Wolf	Wolf	24 h	0.399	0.392	1.8 %	0.398	0.3 %
Peacock + Wolf	Wolf	96 h	0.477	0.486	−1.9 %	0.493	−3.4 %
Lamb + Robintruncated1	Lamb	24 h	0.166	0.163	1.8 %	0.150	9.6 %
Lamb + Robintruncated1	Lamb	96 h	0.249	0.294	−18.1 %	0.252	−1.2 %
Lamb + Robintruncated1	Robintruncated1	24 h	0.155	0.170	−9.7 %	0.153	1.3 %
Lamb + Robintruncated1	Robintruncated1	96 h	0.268	0.286	−6.7 %	0.262	2.2 %
Hogtruncated1 + Moosetruncated2	Hogtruncated1	24 h	0.359	0.290	19.2 %	0.298	17.0 %
Hogtruncated1 + Moosetruncated2	Hogtruncated1	96 h	0.666	0.457	31.4 %	0.471	29.3 %
Hogtruncated1 + Moosetruncated2	Moosetruncated2	24 h	0.175	0.163	6.9 %	0.163	6.9 %
Hogtruncated1 + Moosetruncated2	Moosetruncated2	96 h	0.313	0.271	13.4 %	0.286	8.6 %
Hogtruncated2 + Moosetruncated2	Hogtruncated2	24 h	0.320	0.397	−24.1 %	0.330	−3.1 %
Hogtruncated2 + Moosetruncated2	Hogtruncated2	96 h	0.540	0.589	−9.1 %	0.505	6.5 %
Hogtruncated2 + Moosetruncated2	Moosetruncated2	24 h	0.175	0.163	6.9 %	0.163	6.9 %
Hogtruncated2 + Moosetruncated2	Moosetruncated2	96 h	0.313	0.271	13.4 %	0.286	8.6 %

Table A.13
MSE performance summary for large-scale modelling on individual datasets *(Vanilla Transformer)*.

Dataset	Horizon	Baseline MSE	Strategy 6 MSE	Imp (%)	Strategy 8 MSE	Imp (%)
Bear	24 h	0.219	0.211	+3.7 %	0.201	+8.2 %
Bear	96 h	0.323	0.347	−7.4 %	0.314	+2.8 %
Bobcat	24 h	0.353	0.311	+11.9 %	0.301	+14.7 %
Bobcat	96 h	0.541	0.438	+19.0 %	0.426	+21.3 %
Bull	24 h	0.479	0.436	+9.0 %	0.432	+9.8 %
Bull	96 h	0.717	0.635	+11.4 %	0.655	+8.6 %
Cockatoo	24 h	0.426	0.384	+9.9 %	0.379	+11.0 %
Cockatoo	96 h	0.611	0.583	+4.6 %	0.566	+7.4 %
Crow	24 h	0.293	0.254	+13.3 %	0.246	+16.0 %
Crow	96 h	0.446	0.404	+9.4 %	0.389	+12.8 %
Eagle	24 h	0.311	0.256	+17.7 %	0.261	+16.1 %
Eagle	96 h	0.461	0.345	+25.2 %	0.351	+23.9 %
Fox	24 h	0.181	0.178	+1.7 %	0.166	+8.3 %
Fox	96 h	0.252	0.281	−11.5 %	0.248	+1.6 %
Gator	24 h	0.270	0.310	−14.8 %	0.270	+0.0 %
Gator	96 h	0.544	0.588	−8.1 %	0.536	+1.5 %
Hog	24 h	0.247	0.213	+13.8 %	0.209	+15.4 %
Hog	96 h	0.426	0.356	+16.4 %	0.372	+12.7 %
Lamb	24 h	0.166	0.150	+9.6 %	0.141	+15.1 %
Lamb	96 h	0.249	0.257	−3.2 %	0.223	+10.4 %
Moose	24 h	0.227	0.198	+12.8 %	0.185	+18.5 %
Moose	96 h	0.334	0.323	+3.3 %	0.285	+14.7 %
Mouse	24 h	0.309	0.277	+10.4 %	0.270	+12.6 %
Mouse	96 h	0.575	0.452	+21.4 %	0.433	+24.7 %
Peacock	24 h	0.198	0.201	−1.5 %	0.189	+4.5 %
Peacock	96 h	0.327	0.326	+0.3 %	0.296	+9.5 %
Rat	24 h	0.207	0.194	+6.3 %	0.194	+6.3 %
Rat	96 h	0.371	0.352	+5.1 %	0.343	+7.5 %
Robin	24 h	0.155	0.150	+3.2 %	0.142	+8.4 %
Robin	96 h	0.268	0.280	−4.5 %	0.240	+10.4 %
Wolf	24 h	0.399	0.394	+1.3 %	0.362	+9.3 %
Wolf	96 h	0.477	0.529	−10.9 %	0.468	+1.9 %

Table A.14
MAE performance summary for large-scale modelling on individual datasets *(Informer)*.

Dataset	Horizon	Baseline MAE	Strategy 6 MAE	Imp (%)	Strategy 8 MAE	Imp (%)
Bear	24 h	0.306	0.292	+4.6 %	0.287	+6.2 %
Bear	96 h	0.379	0.387	−2.1 %	0.372	+1.8 %
Bobcat	24 h	0.392	0.360	+8.2 %	0.356	+9.2 %
Bobcat	96 h	0.492	0.460	+6.5 %	0.458	+6.9 %
Bull	24 h	0.467	0.446	+4.5 %	0.448	+4.1 %
Bull	96 h	0.585	0.561	+4.1 %	0.566	+3.2 %
Cockatoo	24 h	0.455	0.425	+6.6 %	0.430	+5.5 %
Cockatoo	96 h	0.600	0.578	+3.7 %	0.582	+3.0 %
Crow	24 h	0.345	0.294	+14.8 %	0.299	+13.3 %
Crow	96 h	0.471	0.419	+11.0 %	0.418	+11.3 %
Eagle	24 h	0.381	0.348	+8.7 %	0.349	+8.4 %
Eagle	96 h	0.485	0.417	+14.0 %	0.425	+12.4 %
Fox	24 h	0.262	0.266	−1.5 %	0.257	+1.9 %
Fox	96 h	0.330	0.347	−5.2 %	0.324	+1.8 %
Gator	24 h	0.300	0.291	+3.0 %	0.285	+5.0 %
Gator	96 h	0.506	0.508	−0.4 %	0.494	+2.4 %
Hog	24 h	0.320	0.293	+8.4 %	0.296	+7.5 %
Hog	96 h	0.468	0.417	+10.9 %	0.424	+9.4 %
Lamb	24 h	0.192	0.186	+3.1 %	0.179	+6.8 %
Lamb	96 h	0.273	0.279	−2.2 %	0.261	+4.4 %
Moose	24 h	0.285	0.258	+9.5 %	0.255	+10.5 %
Moose	96 h	0.368	0.369	−0.3 %	0.344	+6.5 %
Mouse	24 h	0.351	0.320	+8.8 %	0.319	+9.1 %
Mouse	96 h	0.497	0.444	+10.7 %	0.444	+10.7 %
Peacock	24 h	0.315	0.310	+1.6 %	0.303	+3.8 %
Peacock	96 h	0.395	0.403	−2.0 %	0.392	+0.8 %
Rat	24 h	0.291	0.282	+3.1 %	0.283	+2.7 %
Rat	96 h	0.410	0.399	+2.7 %	0.397	+3.2 %
Robin	24 h	0.259	0.247	+4.6 %	0.238	+8.1 %
Robin	96 h	0.343	0.353	−2.9 %	0.326	+5.0 %
Wolf	24 h	0.379	0.369	+2.6 %	0.362	+4.5 %
Wolf	96 h	0.451	0.463	−2.7 %	0.455	−0.9 %

Table A.15
MSE performance summary for large-scale modelling on individual datasets *(Informer)*.

Dataset	Horizon	Baseline MSE	Strategy 6 MSE	Imp (%)	Strategy 8 MSE	Imp (%)
Bear	24 h	0.218	0.211	+3.2 %	0.202	+7.3 %
Bear	96 h	0.323	0.340	−5.3 %	0.320	+0.9 %
Bobcat	24 h	0.326	0.300	+8.0 %	0.293	+10.1 %
Bobcat	96 h	0.452	0.430	+4.9 %	0.423	+6.4 %
Bull	24 h	0.461	0.425	+7.8 %	0.431	+6.5 %
Bull	96 h	0.678	0.625	+7.8 %	0.635	+6.3 %
Cockatoo	24 h	0.408	0.376	+7.8 %	0.382	+6.4 %
Cockatoo	96 h	0.612	0.593	+3.1 %	0.597	+2.5 %
Crow	24 h	0.273	0.242	+11.4 %	0.240	+12.1 %
Crow	96 h	0.430	0.392	+8.8 %	0.384	+10.7 %
Eagle	24 h	0.305	0.258	+15.4 %	0.262	+14.1 %
Eagle	96 h	0.462	0.344	+25.5 %	0.358	+22.5 %
Fox	24 h	0.173	0.178	−2.9 %	0.168	+2.9 %
Fox	96 h	0.255	0.278	−9.0 %	0.251	+1.6 %
Gator	24 h	0.273	0.279	−2.2 %	0.271	+0.7 %
Gator	96 h	0.541	0.561	−3.7 %	0.540	+0.2 %
Hog	24 h	0.227	0.209	+7.9 %	0.213	+6.2 %
Hog	96 h	0.431	0.364	+15.5 %	0.376	+12.8 %
Lamb	24 h	0.154	0.149	+3.2 %	0.140	+9.1 %
Lamb	96 h	0.243	0.253	−4.1 %	0.225	+7.4 %
Moose	24 h	0.208	0.189	+9.1 %	0.186	+10.6 %
Moose	96 h	0.321	0.310	+3.4 %	0.285	+11.2 %
Mouse	24 h	0.293	0.267	+8.9 %	0.262	+10.6 %
Mouse	96 h	0.496	0.452	+8.9 %	0.438	+11.7 %
Peacock	24 h	0.197	0.197	+0.0 %	0.187	+5.1 %
Peacock	96 h	0.300	0.311	−3.7 %	0.298	+0.7 %
Rat	24 h	0.206	0.198	+3.9 %	0.196	+4.9 %
Rat	96 h	0.365	0.354	+3.0 %	0.349	+4.4 %
Robin	24 h	0.159	0.150	+5.7 %	0.141	+11.3 %
Robin	96 h	0.263	0.277	−5.3 %	0.243	+7.6 %
Wolf	24 h	0.362	0.379	−4.7 %	0.356	+1.7 %
Wolf	96 h	0.470	0.511	−8.7 %	0.476	−1.3 %

Table A.16
MAE performance summary for large-scale modelling on individual datasets *(PatchTST)*.

Dataset	Horizon	Baseline MAE	Strategy 6 MAE	Imp (%)	Strategy 8 MAE	Imp (%)
Bear	24 h	0.290	0.281	+3.0 %	0.278	+4.0 %
Bear	96 h	0.377	0.376	+0.3 %	0.362	+3.8 %
Bobcat	24 h	0.373	0.281	+24.6 %	0.346	+7.1 %
Bobcat	96 h	0.451	0.376	+16.8 %	0.439	+2.8 %
Bull	24 h	0.435	0.281	+35.4 %	0.426	+2.1 %
Bull	96 h	0.535	0.376	+29.8 %	0.537	−0.4 %
Cockatoo	24 h	0.415	0.281	+32.3 %	0.397	+4.3 %
Cockatoo	96 h	0.545	0.376	+31.1 %	0.543	+0.3 %
Crow	24 h	0.315	0.281	+10.6 %	0.283	+9.9 %
Crow	96 h	0.434	0.376	+13.5 %	0.423	+2.6 %
Eagle	24 h	0.326	0.281	+13.7 %	0.316	+3.0 %
Eagle	96 h	0.394	0.376	+4.6 %	0.385	+2.3 %
Fox	24 h	0.259	0.281	−8.6 %	0.250	+3.3 %
Fox	96 h	0.324	0.376	−15.9 %	0.317	+2.1 %
Gator	24 h	0.286	0.281	+1.5 %	0.282	+1.3 %
Gator	96 h	0.488	0.376	+23.0 %	0.486	+0.4 %
Hog	24 h	0.291	0.281	+3.5 %	0.273	+6.3 %
Hog	96 h	0.408	0.376	+7.9 %	0.390	+4.2 %
Lamb	24 h	0.173	0.281	−62.8 %	0.164	+5.3 %
Lamb	96 h	0.222	0.376	−69.4 %	0.216	+2.6 %
Moose	24 h	0.267	0.281	−5.4 %	0.234	+12.3 %
Moose	96 h	0.350	0.376	−7.4 %	0.332	+5.1 %
Mouse	24 h	0.331	0.281	+15.0 %	0.312	+5.6 %
Mouse	96 h	0.435	0.376	+13.6 %	0.439	−1.0 %
Peacock	24 h	0.302	0.281	+7.0 %	0.290	+3.9 %
Peacock	96 h	0.375	0.376	−0.2 %	0.366	+2.4 %
Rat	24 h	0.275	0.281	−2.2 %	0.266	+3.5 %
Rat	96 h	0.378	0.376	+0.8 %	0.371	+2.0 %
Robin	24 h	0.247	0.281	−13.9 %	0.235	+4.6 %
Robin	96 h	0.333	0.376	−12.9 %	0.320	+3.8 %
Wolf	24 h	0.396	0.281	+29.0 %	0.371	+6.3 %
Wolf	96 h	0.495	0.376	+24.1 %	0.489	+1.1 %

Table A.17
MSE performance summary for large-scale modelling on individual datasets *(PatchTST)*.

Dataset	Horizon	Baseline MSE	Strategy 6 MSE	Imp (%)	Strategy 8 MSE	Imp (%)
Bear	24 h	0.201	0.203	−1.4 %	0.191	+4.7 %
Bear	96 h	0.321	0.327	−2.0 %	0.307	+4.4 %
Bobcat	24 h	0.305	0.203	+33.2 %	0.284	+6.9 %
Bobcat	96 h	0.406	0.327	+19.3 %	0.402	+1.1 %
Bull	24 h	0.407	0.203	+49.9 %	0.396	+2.5 %
Bull	96 h	0.582	0.327	+43.7 %	0.589	−1.3 %
Cockatoo	24 h	0.351	0.203	+42.0 %	0.335	+4.6 %
Cockatoo	96 h	0.536	0.327	+38.9 %	0.539	−0.7 %
Crow	24 h	0.249	0.203	+18.1 %	0.226	+9.2 %
Crow	96 h	0.404	0.327	+19.0 %	0.413	−2.1 %
Eagle	24 h	0.226	0.203	+9.9 %	0.219	+3.1 %
Eagle	96 h	0.322	0.327	−1.6 %	0.314	+2.5 %
Fox	24 h	0.169	0.203	−20.1 %	0.164	+3.4 %
Fox	96 h	0.249	0.327	−31.4 %	0.243	+2.5 %
Gator	24 h	0.274	0.203	+25.8 %	0.275	−0.3 %
Gator	96 h	0.547	0.327	+40.1 %	0.543	+0.7 %
Hog	24 h	0.205	0.203	+0.7 %	0.190	+7.0 %
Hog	96 h	0.350	0.327	+6.6 %	0.336	+4.1 %
Lamb	24 h	0.143	0.203	−41.9 %	0.137	+4.5 %
Lamb	96 h	0.219	0.327	−49.8 %	0.214	+2.3 %
Moose	24 h	0.188	0.203	−8.3 %	0.167	+11.3 %
Moose	96 h	0.293	0.327	−11.7 %	0.289	+1.5 %
Mouse	24 h	0.272	0.203	+25.2 %	0.257	+5.3 %
Mouse	96 h	0.427	0.327	+23.4 %	0.457	−7.0 %
Peacock	24 h	0.189	0.203	−7.8 %	0.179	+5.3 %
Peacock	96 h	0.283	0.327	−15.9 %	0.273	+3.3 %
Rat	24 h	0.183	0.203	−11.3 %	0.176	+4.0 %
Rat	96 h	0.316	0.327	−3.5 %	0.308	+2.6 %
Robin	24 h	0.150	0.203	−36.0 %	0.140	+6.4 %
Robin	96 h	0.251	0.327	−30.4 %	0.240	+4.6 %
Wolf	24 h	0.342	0.203	+40.6 %	0.326	+4.8 %
Wolf	96 h	0.494	0.327	+33.7 %	0.492	+0.4 %

Appendix B. Graphical representation of MAE results for strategies 6 and 8

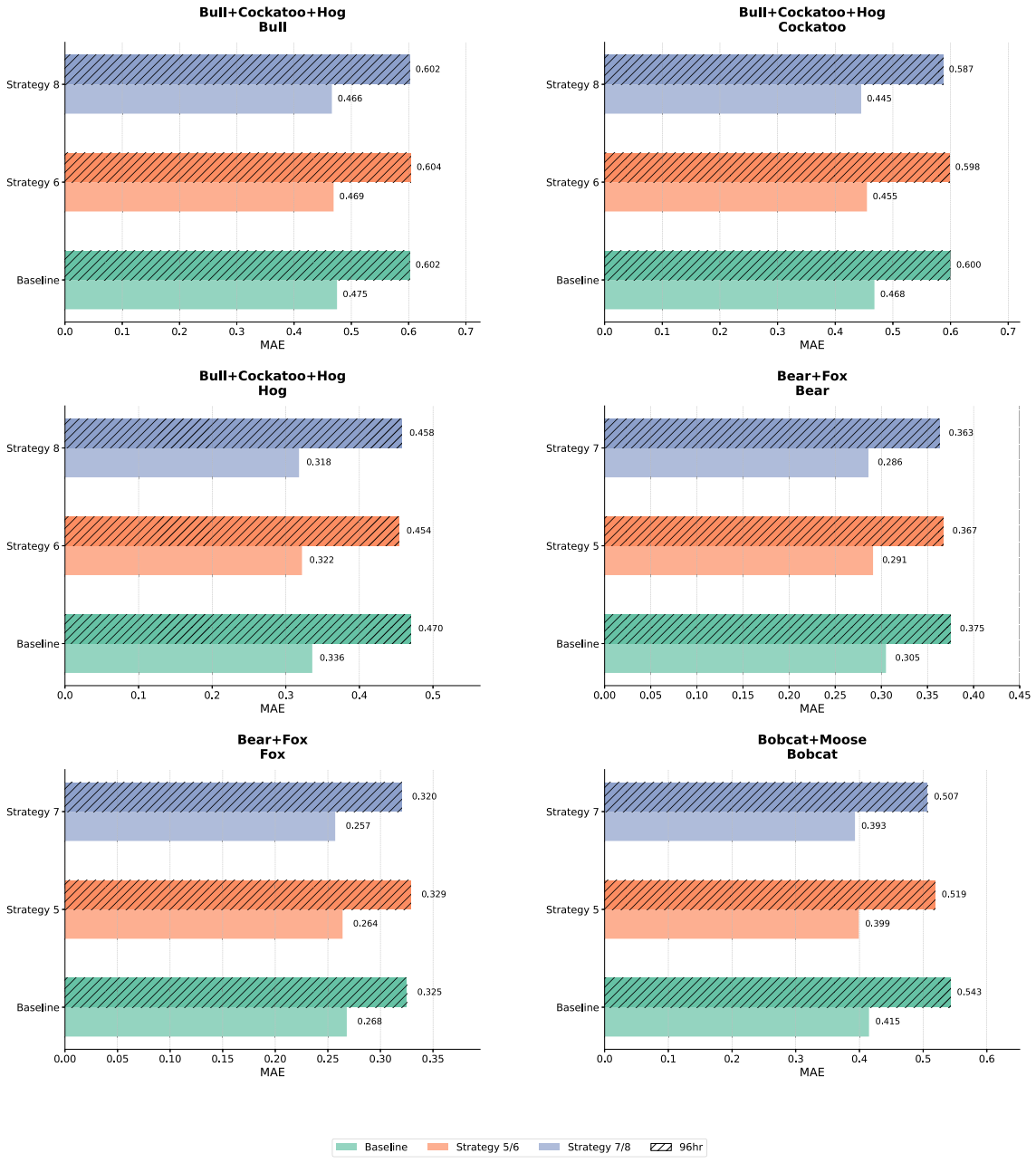


Fig. B.6. Mean absolute error (MAE) comparison for small-scale transfer learning scenarios (Part 1). This figure shows the performance of base, combined, and fine-tuned models for the first 20 dataset combinations and test scenarios.

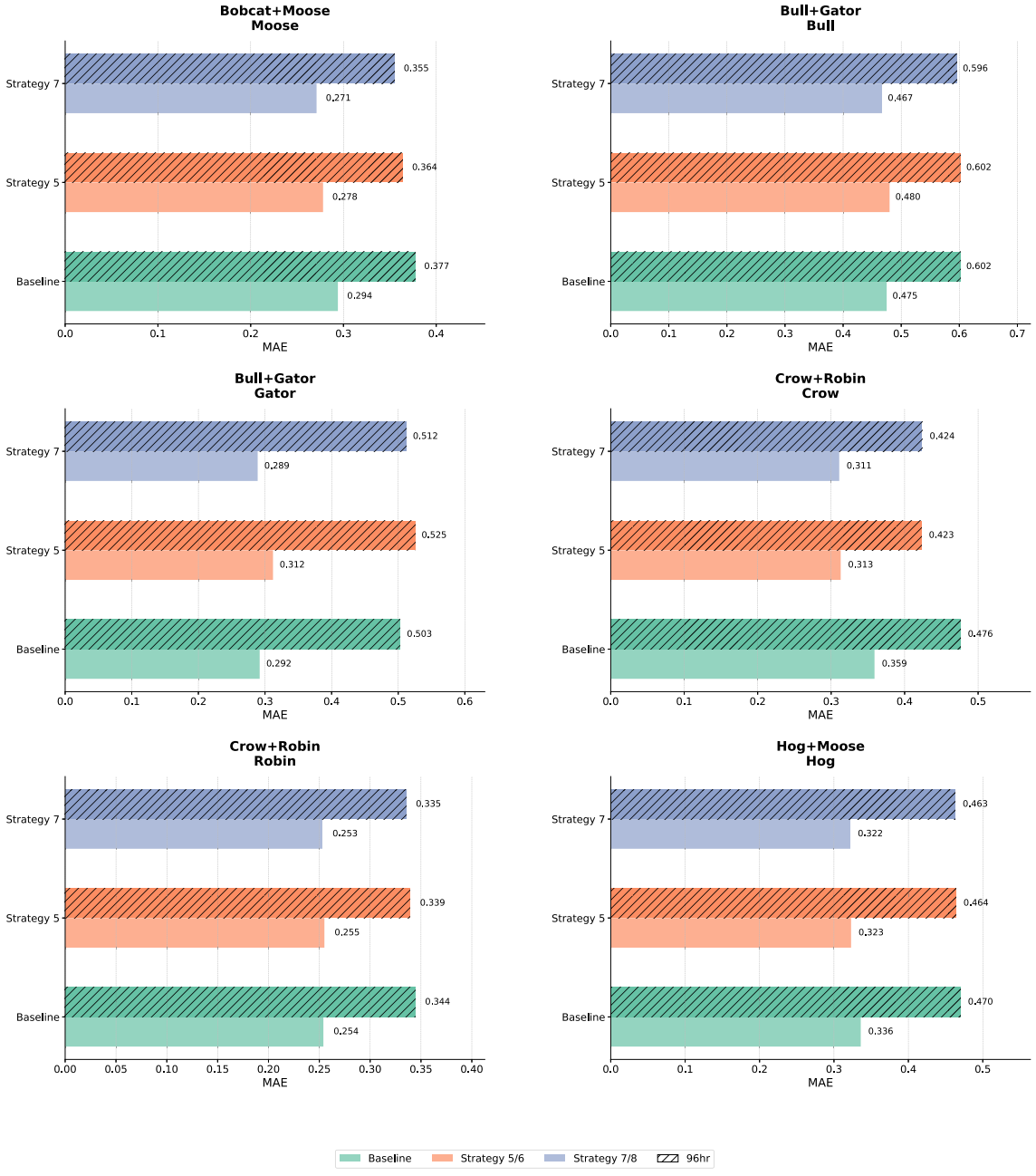


Fig. B.7. Mean absolute error (MAE) comparison for small-scale transfer learning scenarios (Part 2). This figure presents the performance of base, combined, and fine-tuned models for the remaining 14 dataset combinations and test scenarios.

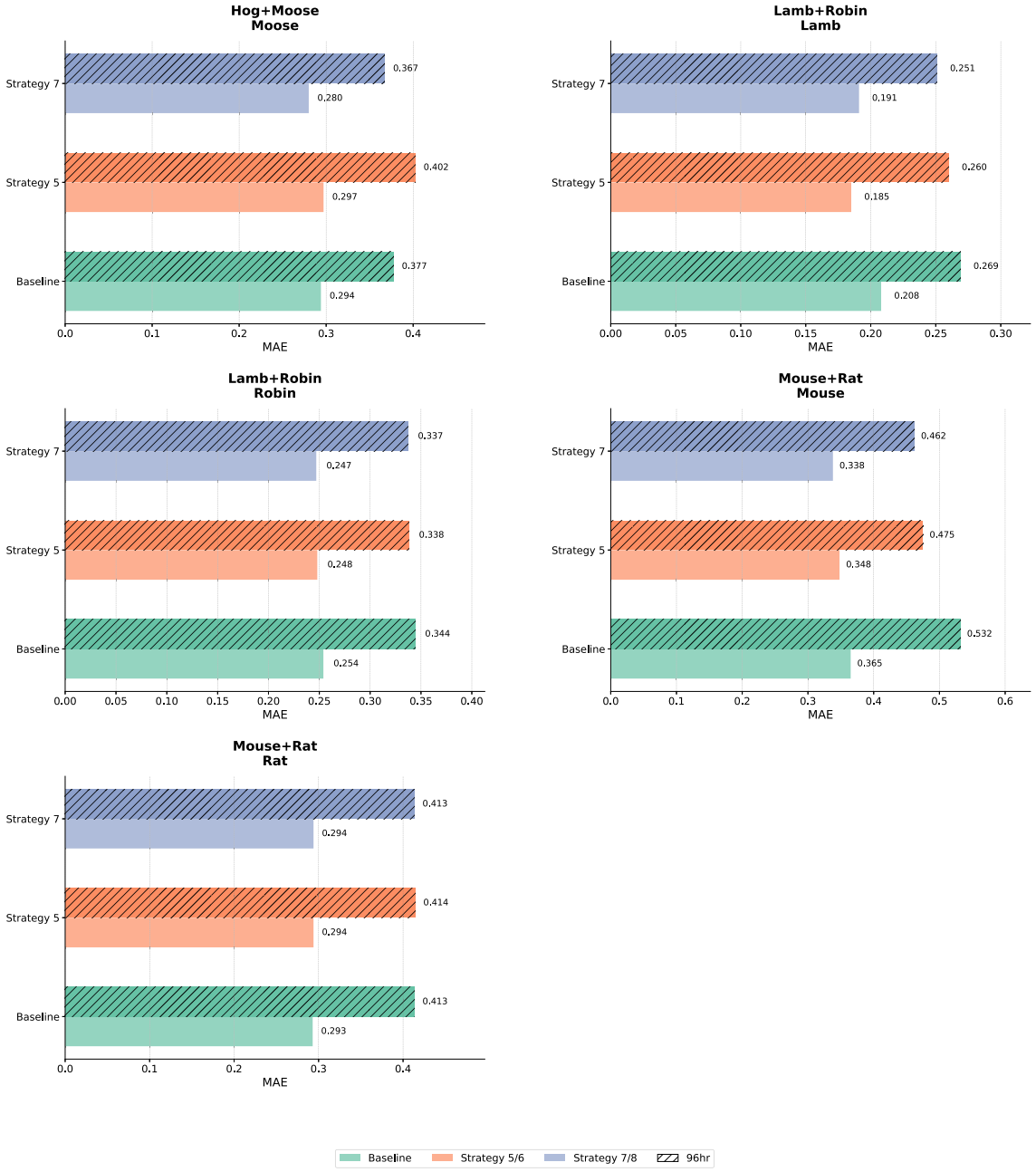


Fig. B.8. Mean absolute error (MAE) comparison for small-scale transfer learning scenarios (Part 3). This figure presents the performance of base, combined, and fine-tuned models for the remaining 14 dataset combinations and test scenarios.

References

[1] F. Dinmohammadi, Y. Han, M. Shafiee, Predicting energy consumption in residential buildings using advanced machine learning algorithms, Energies 16 (9) (2023). https://doi.org/10.3390/en16093748

[2] A. Ahmad, T.N. Anderson, S.U. Rehman, Prediction of electricity consumption for residential houses in New Zealand, in: P.H.J. Chong, B.-C. Seet, M. Chai, S.U. Rehman (Eds.), Smart Grid and Innovative Frontiers in Telecommunications, Springer International Publishing, Cham, 2018, pp. 165–172.

[3] S. Christopher, M.P. Vikram, C. Bakli, A.K. Thakur, Y. Ma, Z. Ma, H. Xu, P.M. Cuce, E. Cuce, P. Singh, Renewable energy potential towards attainment of net-zero energy buildings status—A critical review, J. Clean. Prod. 405 (2023) 136942. https://doi.org/10.1016/j.jclepro.2023.136942

[4] D.-K. Bui, T.N. Nguyen, T.D. Ngo, H. Nguyen-Xuan, An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings, Energy 190 (2020) 116370. https://doi.org/10.1016/j.energy.2019.116370

[5] J. Luo, C. Paduraru, O. Voicu, Y. Chervonyi, S. Munns, J. Li, C. Qian, P. Dutta, J.Q. Davis, N. Wu, X. Yang, C.-M. Chang, T. Li, R. Rose, M. Fan, H. Nakhost, T. Liu, B. Kirkman, F. Altamura, L. Cline, P. Tonker, J. Gouker, D. Uden, W.B. Bryan, J. Law, D. Fatiha, N. Satra, J. Rothenberg, M. Waraich, M. Carlin, S. Tallapaka, S. Witherspoon, D. Parish, P. Dolan, C. Zhao, D.J. Mankowitz, Controlling commercial cooling systems using reinforcement learning (2022). 2211.07357 https://doi.org/10.48550/arxiv.2211.07357

[6] C. Deb, F. Zhang, J. Yang, S.E. Lee, K.W. Shah, A review on time series forecasting techniques for building energy consumption, Renew. Sustain. Energy Rev. 74 (2017) 902–924. https://doi.org/10.1016/j.rser.2017.02.085

[7] A.A.A. Gassar, S.H. Cha, Energy prediction techniques for large-scale buildings towards a sustainable built environment: a review, Energy Build. 224 (2020) 110238. https://doi.org/10.1016/j.enbuild.2020.110238

[8] K.B. Debnath, M. Mourshed, Forecasting methods in energy planning models, Renew. Sustain. Energy Rev. 88 (2018) 297–325. https://doi.org/10.1016/j.rser.2018.02.002

[9] R. Chandrasekaran, S.K. Paramasivan, Advances in deep learning techniques for short-term energy load forecasting applications: a review, Arch. Comput. Methods Eng. 32 (2025) 663-692. https://doi.org/10.1007/s11831-024-10155-x

[10] K. Wang, K. Li, L. Zhou, Y. Hu, Z. Cheng, J. Liu, C. Chen, Multiple convolutional neural networks for multivariate time series prediction, Neurocomputing 360 (2019) 107–119.

[11] S. Ahmed, I.E. Nielsen, A. Tripathi, S. Siddiqui, R.P. Ramachandran, G. Rasool, Transformers in time-series analysis: a tutorial, Circuits, Syst., Signal Process. 42 (12) (2023) 7433–7466.

[12] Z. Zeng, R. Kaur, S. Siddagangappa, S. Rahimi, T. Balch, M. Veloso, Financial time series forecasting using cnn and transformer, arXiv preprint arXiv:2304.04912 (2023).

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16×16 words: transformers for image recognition at scale, in: International Conference on Learning Representations, 2020.

[16] M. Hertel, M. Beichter, B. Heidrich, O. Neumann, B. Schäfer, R. Mikut, V. Hagenmeyer, Transformer training strategies for forecasting multiple load time series, Energy Inform. 6 (Suppl 1) (2023) 20. https://doi.org/10.1186/s42162-023-00278-z

[17] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 35, 2021, pp. 11106–11115.

[18] Y. Nie, N.H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: long-term forecasting with transformers, in: The Eleventh International Conference on Learning Representations, 2023.

[19] B. Lim, S.Ö. Arık, N. Loeff, T. Pfister, Temporal fusion transformers for interpretable multi-horizon time series forecasting, Int. J. Forecast. 37 (4) (2021) 1748–1764.

[20] Z. Ni, C. Zhang, M. Karlsson, S. Gong, A study of deep learning-based multi-horizon building energy forecasting, Energy Build. 303 (2024) 113810.

[21] S. Lazarova-Molnar, N. Mohamed, Challenges in the data collection for diagnostics of smart buildings, in: K.J. Kim, N. Joukov (Eds.), Information Science and Applications (ICISA) 2016, Springer Singapore, Singapore, 2016, pp. 941–951.

[22] O. Guerra-Santin, C.A. Tweed, In-use monitoring of buildings: an overview of data collection methods, Energy Build. 93 (2015) 189–207. https://doi.org/10.1016/j.enbuild.2015.02.042

[23] C. Miller, A. Kathirgamanathan, B. Picchetti, P. Arjunan, J.Y. Park, Z. Nagy, P. Raftery, B.W. Hobson, Z. Shi, F. Meggers, The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition, Sci. Data 7 (1) (2020). https://doi.org/10.1038/s41597-020-00712-x

[24] C.M.K. Elizabeth L. Ratnam, Steven R. Weller, A.T. Murray, Residential load and rooftop PV generation: an Australian distribution network dataset, Int. J. Sustain. Energy 36 (8) (2017) 787–806. https://doi.org/10.1080/14786451.2015.1100196

[25] F. Rodrigues, A. Trindade, Load forecasting through functional clustering and ensemble learning, Knowl. Inf. Syst. 57 (1) (2018) 229–244.

[26] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2009) 1345–1359.

[27] G. Gokhale, J. Van Gompel, B. Claessens, C. Develder, Transfer learning in transformer-based demand forecasting for home energy management system, in: Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 458–462. https://doi.org/10.1145/3600100.3626635

[28] M.L. Santos, S.D. García, X. García-Santiago, A. Ogando-Martínez, F.E. Camarero, G.B. Gil, P.C. Ortega, Deep learning and transfer learning techniques applied to short-term load forecasting of data-poor buildings in local energy communities, Energy Build. 292 (2023) 113164. https://doi.org/10.1016/j.enbuild.2023.113164

[29] V. Laitsos, G. Vontzos, A. Tsiovoulos, D. Bargiotas, L.H. Tsoukalas, Enhanced sequence-to-sequence deep transfer learning for day-ahead electricity load forecasting, Electronics 13 (10) (2024). https://doi.org/10.3390/electronics13101996

[30] H. Lu, J. Wu, Y. Ruan, F. Qian, H. Meng, Y. Gao, T. Xu, A multi-source transfer learning model based on LSTM and domain adaptation for building energy prediction, Int. J. Electr. Power Energy Syst. 149 (2023) 109024. https://doi.org/https://doi.org/10.1016/j.ijepes.2023.109024

[31] Y. Ahn, B.S. Kim, Prediction of building power consumption using transfer learning-based reference building and simulation dataset, Energy Build. 258 (2022) 111717. https://doi.org/https://doi.org/10.1016/j.enbuild.2021.111717

[32] P.C. Huy, N.Q. Minh, N.D. Tien, T.T.Q. Anh, Short-term electricity load forecasting based on temporal fusion transformer model, IEEE Access 10 (2022) 106296–106304.

[33] M. Ribeiro, K. Grolinger, H.F. ElYamany, W.A. Higashino, M.A.M. Capretz, Transfer learning with seasonal and trend adjustment for cross-building energy forecasting, Energy Build. 165 (2018) 352–363. https://doi.org/10.1016/j.enbuild.2018.01.034

[34] Y. Lu, Z. Tian, R. Zhou, W. Liu, A general transfer learning-based framework for thermal load prediction in regional energy system, Energy 217 (2021) 119322. https://doi.org/https://doi.org/10.1016/j.energy.2020.119322

[35] Y. Tian, L. Sehovac, K. Grolinger, Similarity-based chained transfer learning for energy forecasting with big data, IEEE Access 7 (2019) 139895–139908. https://doi.org/10.1109/ACCESS.2019.2943752

[36] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[37] X. Fang, G. Gong, G. Li, L. Chun, W. Li, P. Peng, A hybrid deep transfer learning strategy for short term cross-building energy prediction, Energy 215 (2021) 119208. https://doi.org/10.1016/j.energy.2020.119208

[38] G. Li, Z. Wang, J. Gao, C. Xu, Y. Guo, D. Sun, X. Fang, Performance assessment of cross office building energy prediction in the same region using the domain adversarial transfer learning strategy, Appl. Therm. Eng. 241 (2024) 122357. https://doi.org/https://doi.org/10.1016/j.applthermaleng.2024.122357

[39] G. Li, Y. Wu, S. Yoon, X. Fang, Comprehensive transferability assessment of short-term cross-building-energy prediction using deep adversarial network transfer learning, Energy 299 (2024) 131395.

[40] G. Li, Y. Wu, J. Liu, X. Fang, Z. Wang, Performance evaluation of short-term cross-building energy predictions using deep transfer learning strategies, Energy Build. 275 (2022) 112461. https://doi.org/https://doi.org/10.1016/j.enbuild.2022.112461

[41] M. Voß, C. Bender-Saebelkampf, S. Albayrak, Residential short-term load forecasting using convolutional neural networks, in: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2018, pp. 1–6. https://doi.org/10.1109/SmartGridComm.2018.8587494

[42] A. Hooshmand, R. Sharma, Energy predictive models with limited data using transfer learning, in: Proceedings of the Tenth ACM International Conference on Future Energy Systems, e-Energy '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 12–16. https://doi.org/10.1145/3307772.3328284

[43] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, Y.C. Tseng, Statistical investigations of transfer learning-based methodology for short-term building energy predictions, Appl. Energy 262 (2020) 114499. https://doi.org/https://doi.org/10.1016/j.apenergy.2020.114499

[44] Y. Gao, Y. Ruan, C. Fang, S. Yin, Deep learning and transfer learning models of energy consumption forecasting for buildings with poor information data, Energy Build. 223 (2020) 110156. https://doi.org/10.1016/j.enbuild.2020.110156

[45] S.-M. Jung, S. Park, S.-W. Jung, E. Hwang, Monthly electric load forecasting using transfer learning for smart cities, Sustainability 12 (16) (2020). https://doi.org/10.3390/su12166364

[46] J. Ma, J. Cheng, F. Jiang, W. Chen, M. Wang, C. Zhai, A bi-directional missing data imputation scheme based on LSTM and transfer learning for building energy data, Energy Build. 216 (2020) 109941. https://doi.org/10.1016/j.enbuild.2020.109941

[47] H. Hu, M. Tang, C. Bai, DATSING: data augmented time series forecasting with adversarial domain adaptation, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2061–2064. https://doi.org/10.1145/3340531.3412155

[48] E. Lee, W. Rhee, Individualized short-term electric load forecasting with deep neural network based transfer learning and meta learning, IEEE Access 9 (2021) 15413–15425. https://doi.org/10.1109/ACCESS.2021.3053317

[49] A. Li, F. Xiao, C. Fan, M. Hu, Development of an ANN-based building energy model for information-poor buildings using transfer learning, in: Building Simulation, 14, Springer, 2021, pp. 89–101.

[50] M. Jain, K. Gupta, A. Visweswara Sathanur, V. Chandan, M. Halappanavar, Transfer-learnt energy models for predicting electricity consumption in buildings with limited and sparse field data (2021).

[51] H. Park, D.Y. Park, B. Noh, S. Chang, Stacking deep transfer learning for short-term cross building energy prediction with different seasonality and occupant sched-

ule, Build. Environ. 218 (2022) 109060. https://doi.org/https://doi.org/10.1016/j.buildenv.2022.109060
[52] C. Peng, Y. Tao, Z. Chen, Y. Zhang, X. Sun, Multi-source transfer learning guided ensemble LSTM for building multi-load forecasting, Expert Syst. Appl. 202 (2022) 117194. https://doi.org/https://doi.org/10.1016/j.eswa.2022.117194
[53] R. Yan, T. Zhao, Y. Rezgui, S. Kubicki, Y. Li, Transferability and robustness of a data-driven model built on a large number of buildings, J. Build. Eng. 80 (2023) 108127. https://doi.org/10.1016/j.jobe.2023.108127
[54] D. Kim, Y. Lee, K. Chin, P.J. Mago, H. Cho, J. Zhang, Implementation of a long short-term memory transfer learning (LSTM-TL)-based data-driven model for building energy demand forecasting, Sustainability 15 (3) (2023). https://doi.org/10.3390/su15032340
[55] A.M. Tzortzis, S. Pelekis, E. Spiliotis, E. Karakolis, S. Mouzakitis, J. Psarras, D. Askounis, Transfer learning for day-ahead load forecasting: a case study on european national electricity demand time series, Mathematics 12 (1) (2024). https://doi.org/10.3390/math12010019
[56] Y. Yuan, Z. Chen, Z. Wang, Y. Sun, Y. Chen, Attention mechanism-based transfer learning model for day-ahead energy demand forecasting of shopping mall buildings, Energy 270 (2023) 126878. https://doi.org/https://doi.org/10.1016/j.energy.2023.126878
[57] B. Yang, Y. Chen, M. Gül, H. Yu, C. Shields, Ensemble transfer learning strategy in forecasting power consumption for residential buildings, 2023. https://doi.org/10.26868/25222708.2023.1400
[58] M. Zhou, J. Yu, F. Sun, M. Wang, Forecasting of short term electric power consumption for different types buildings using improved transfer learning: a case study of primary school in China, J. Build. Eng. 78 (2023) 107618. https://doi.org/https://doi.org/10.1016/j.jobe.2023.107618
[59] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, X. Shi, Transferability investigation of a Sim2Real deep transfer learning framework for cross-building energy prediction, Energy Build. 287 (2023) 112968. https://doi.org/https://doi.org/10.1016/j.enbuild.2023.112968
[60] V. Laitsos, G. Vontzos, D. Bargiotas, Investigation of transfer learning for electricity load forecasting, in: 2023 14th International Conference on Information, Intelligence, Systems & Applications (IISA), 2023, pp. 1–7. https://doi.org/10.1109/IISA59645.2023.10345954
[61] G. Li, Y. Wu, C. Yan, X. Fang, T. Li, J. Gao, C. Xu, Z. Wang, An improved transfer learning strategy for short-term cross-building energy prediction using data incremental, in: Building Simulation, 17, Springer, 2024, pp. 165–183.
[62] D. Kim, G. Seomun, Y. Lee, H. Cho, K. Chin, M.-H. Kim, Forecasting building energy demand and on-site power generation for residential buildings using long and short-term memory method with transfer learning, Appl. Energy 368 (2024) 123500. https://doi.org/https://doi.org/10.1016/j.apenergy.2024.123500
[63] L. Xiao, Q. Bai, B. Wang, A dynamic multi-model transfer based short-term load forecasting, Appl. Soft Comput. 159 (2024) 111627. https://doi.org/https://doi.org/10.1016/j.asoc.2024.111627
[64] N. Wei, C. Yin, L. Yin, J. Tan, J. Liu, S. Wang, W. Qiao, F. Zeng, Short-term load forecasting based on WM algorithm and transfer learning model, Appl. Energy 353 (2024) 122087. https://doi.org/https://doi.org/10.1016/j.apenergy.2023.122087
[65] Z. Xing, Y. Pan, Y. Yang, X. Yuan, Y. Liang, Z. Huang, Transfer learning integrating similarity analysis for short-term and long-term building energy consumption prediction, Appl. Energy 365 (2024) 123276. https://doi.org/10.1016/j.apenergy.2024.123276
[66] B. Wei, K. Li, S. Zhou, W. Xue, G. Tan, An instance based multi-source transfer learning strategy for buildings short-term electricity loads prediction under sparse data scenarios, J. Build. Eng. 85 (2024) 108713. https://doi.org/https://doi.org/10.1016/j.jobe.2024.108713
[67] E. Giacomazzi, F. Haag, K. Hopf, Short-term electricity load forecasting using the temporal fusion transformer: effect of grid hierarchies and data sources, in: Proceedings of the 14th ACM International Conference on Future Energy Systems, 2023, pp. 353–360.
[68] H. Ye, Q. Zhu, X. Zhang, Short-term load forecasting for residential buildings based on multivariate variational mode decomposition and temporal fusion transformer, Energies 17 (13) (2024) 3061.
[69] E. Saadipour-Hanzaie, M.-A. Pourmoosavi, T. Amraee, Deep learning based electrical load forecasting using temporal fusion transformer and trend-seasonal decomposition, in: 2023 31st International Conference on Electrical Engineering (ICEE), IEEE, 2023, pp. 283–288.
[70] W. Ji, Z. Cao, X. Li, Multi-task learning and temporal-fusion-transformer-based forecasting of building power consumption, Electronics 12 (22) (2023) 4656.
[71] J. Liu, G. Liu, J. Ruan, K. Wen, C. Yang, J. Zhao, Short-term load forecasting with frequency enhanced decomposed transformer, in: 2022 IEEE 6th Conference on Energy Internet and Energy System Integration (EI2), IEEE, 2022, pp. 1766–1771.
[72] M. Hertel, S. Ott, B. Schäfer, R. Mikut, V. Hagenmeyer, O. Neumann, Evaluation of transformer architectures for electrical load time-series forecasting, in: Proceedings 32. Workshop Computational Intelligence, 1, 2022, p. 93.
[73] D.B. Crawley, R.M. Heiden, E. Baert, C.S. Barnaby, R.B. Burkhead, T.E. Cappellin, et al., ANSI/ASHRAE standard 169–2020. Climatic data for building design standards, ASHRAE Standard (2020).
[74] Z. Wang, Q. Wen, C. Zhang, L. Sun, L.V. Krannichfeldt, Y. Wang, Benchmarks and Custom Package for Electrical Load Forecasting, 2023. https://doi.org/10.48550/arxiv.2307.07191
[75] N. Arivazhagan, A. Bapna, O. Firat, D. Lepikhin, M. Johnson, M. Krikun, M.X. Chen, Y. Cao, G. Foster, C. Cherry, et al., Massively multilingual neural machine translation in the wild: findings and challenges, arXiv preprint arXiv:1907.05019 (2019).
[76] A. Garza, C. Challu, M. Mergenthaler-Canseco, TimeGPT-1, arXiv preprint arXiv:2310.03589 (2023).
[77] K. Rasul, A. Ashok, A.R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. Hassen, A. Schneider, et al., Lag-llama: Towards foundation models for time series forecasting, in: R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models, 2023.