Enabling Time-series Foundation Model for Building Energy Forecasting via Contrastive Curriculum Learning

Rui Liang [1] **Yang Deng** [1] **Donghua Xie** [2] **Dan Wang** [1]

Abstract

Advances in time-series forecasting are driving a shift from conventional machine learning techniques to foundation models (FMs) that are trained with generalized knowledge. However, existing FMs still perform poorly in the energy fields, such as building energy forecasting (BEF). This paper studies the adaptation of FM to BEF tasks. We demonstrate the shortcomings of fine-tuning FM straightforwardly from both the perspectives of FM and the data. To overcome these limitations, we propose a new *contrastive curriculum learning*-based training method. Our method optimizes the ordering of training data in the context of TSFM adaptation. Experiments show that our method can improve the zero/few-shot performance by 14.6% compared to the existing FMs.

1. Introduction

Building energy forecasting (BEF), i.e., energy consumption forecasting for a building, plays a crucial role in many downstream applications, such as equipment control and fault detection. Currently, the majority of BEF schemes are based on machine learning techniques. To achieve acceptable forecasting performance, the common practice is to develop specific models for individual buildings, yet it is hard to generalize at scale and requires significant effort. A growing promising paradigm is *Foundation models* (Liang et al., 2024): large AI model trained on broad data such that it can be applied across a wide range of tasks, such as LLM in the NLP domain. Foundation models are capable of making inferences on a dataset with only a small fraction of training data, or even none at all, which corresponds to few-shot and zero-shot settings, respectively.

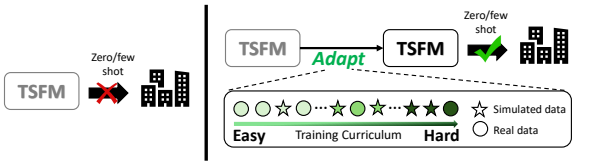

Figure 1. The curriculum enhances TSFM adaptation in building energy forecasting tasks.

There are also some time-series foundation model (denoted as TSFM for simplicity) products available recently, such as IBM Granite (Ekambaram et al., 2024) and Amazon Chronos (Ansari et al., 2024), with a series of TSFMs[1] which are trained on various data source, e.g., weather, energy, medical, financial. From existing benchmarking (Liang et al., 2024), these TSFMs perform well in tasks, e.g., climate forecasting, where large-scale real datasets are adopted for pre-training. However, there are limited real-world data resources in building scenarios and many BEF works rely on the simulated dataset. This is because the building energy is related to occupant privacy or business confidentiality and thus leads to the building managers having a low willingness to share the energy data (Xu & Wang, 2022). From the recent measurement study (Mulayim et al., 2024), the existing TSFMs can not achieve acceptable performance in BEF.

This paper is motivated by an essential question: *Can we adapt the existing TSFMs to support the building energy forecasting tasks via the currently available data resources?* We perform a preliminary evaluation of day-ahead BEF on a product-level TSFM (under zero-shot setting). Our analysis compares the forecasting accuracy of i) the original pre-trained FM and ii) the FM fine-tuned using the BEF dataset in a straightforward manner. This fine-tuning can be conducted using either real-world data alone (R) or a combination of real-world and simulated data (R+S)[2]. And half of

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China [2]Department of Accountancy, Economics and Finance, Hong Kong Baptist University, Hong Kong, China. Correspondence to: Yang Deng <marco.deng@polyu.edu.hk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

[1]TSFMs parameter size of IBM Granite are from 1M to 5M; for Amazon Chronos is range from 8.3M to 709M.

[2]Real-world dataset: BDG (Miller et al., 2020) consists of 1,000+ buildings; Simulated dataset: Building-900K (Emami et al., 2023) consists of 900,000 energy traces simulated by a business software EnergyPlus (Crawley et al., 2001). All the buildings have a energy consumption time-series with a length of two-year.

Table 1. The performance of IBM TSFM. The metric is CV-RMSE (lower the better) and 0.3 is acceptable for engineering purposes.

Time-series FMs	*zero-shot*	*FT on R*	*FT on R+S*
IBM-Granite-TTM-5M	0.409	0.384	0.378

the real buildings was allocated for the test set. As shown in Table 1, the improvement by fine-tuning is very limited, and the performance can not reach engineering purposes. Specifically, even after incorporating 900,000 simulated buildings into the real dataset, the accuracy improved by only 0.6%. These results provide two key insights: (1) There are no universal energy patterns for buildings, as patterns can vary significantly in complexity due to factors such as diverse occupancy behaviors and meteorological conditions. (2) Straightforward training or fine-tuning does not enhance the FM, even with a sufficiently large training set. This is because the knowledge embedded in the pre-trained FM is not easily quantifiable.

To address this challenging problem, we propose a new *contrastive curriculum learning (CCL)* method to adapt the existing TSFMs to BEF tasks. This curriculum strategy organizes the training samples in order of difficulty, thereby guiding the adaptation process of the TSFMs. Instead of building an FM from scratch, which requires substantial resources and effort and is not feasible at this stage, we leverage the knowledge embedded in existing FMs. Our contributions are as follows:

- We for the first time study the TSFM adaptation for a specific domain, building energy. And we demonstrate that straightforwardly fine-tuning brings limited gain.
- We present a new contrastive curriculum learning method for adapting TSFMs to building energy forecasting tasks.
- We evaluate our method on three public building datasets. Our evaluation indicates 9.9% and 10.4% overall zero-shot and few-shot performance improvement of our method as compared to direct fine-tuning.

2. Preliminaries

Building Energy Forecasting. BEF is a domain-specific task of time-series forecasting: under the rolling forecasting setting with a fixed size window with a length of $L+T$, we have the data sample $u^t = (x^t, y^t)$ at time t, comprising past data $x^t = \{l_1^t, ..., l_L^t\}$ with a look-back window length L and future data $y^t = \{l_{L+1}^t, ..., l_{L+T}^t\}$, where l can be multi-dimension. Considering y in BEF task is single dimension and currently published TSFMs mainly support univariate forecasting (Mulayim et al., 2024), thus $l \in \mathbb{R}$ in this paper.

Curriculum Learning. Motivated by the feature of human education, curriculum learning is a data-centric training strategy in which an ML model is trained on samples of *increasing difficulty* to smooth the training process and get better performance (Wang et al., 2021). The two subtasks are: a *Difficulty Measurer* to measure and rank the difficulty of samples; and a *Training Scheduler* to decide the sequence of samples throughout the training process.

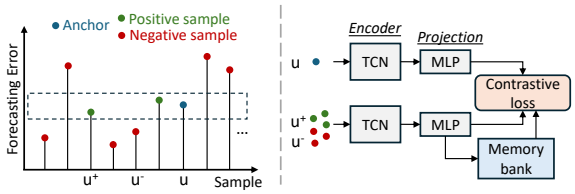

Figure 2. Contrastive Learning model. *Left*: contrastive pairs construction. *Right*: NN model design.

Contrastive Learning. Contrastive learning is to learn an embedding space to represent data samples, in which similar samples are grouped closer while dissimilar samples are pushed apart. The core is to construct positive pair (u, u^+) and negative pair (u, u^-) for an anchor sample u, i.e., to define the *similarity*, based on which to train the NN-based encoder with a contrastive loss function.

3. Methodology

Problem statement: given a pre-trained TSFM $\mathcal{M}$, the existing/available BEF datasets $\mathcal{D} = \{D_{train}, D'_{train}\}$, where $D_{train} = \{u\}$ is the real-world dataset, $D'_{train} = \{u'\}$ is the simulated dataset, and $\frac{|D_{train}|}{|D'_{train}|} \ll 1$. Our objective is to adapt $\mathcal{M}$ to a new $\mathcal{M}'$ using $\mathcal{D}$, to minimize the loss of $\mathcal{M}'$ under zero/few-shot settings.

We propose a new Contrastive-aware Curriculum Learning (CCL) method to schedule the training process of $\mathcal{M}$ on $\mathcal{D}$, with samples ordered as *easy-to-difficult* which is a common paradigm for ML model training, for example, (Wang et al., 2023) schedules the images from blur to clear to train the model. A unique challenge in our scenario is to measure the difficulty of the simulated data. We leverage contrastive representation to cope with this challenge, and the *difficulty measurer* and *training scheduler* are presented as follows.

3.1. Contrastive-aware Difficulty Measurer

The design of the difficulty measurer is usually based on model performance or data pattern analysis. Considering that the curriculum is for adapting an existing TSFM $\mathcal{M}$, which has been pre-trained with various knowledge and patterns, we can directly make inference with the TSFM and use the performance as the *difficulty score* of samples in the real-world dataset D_{train} (Eq. 3.1). Here, $\mathcal{L}(\cdot, \cdot)$ denotes the prediction error.

$$D_{\mathcal{M}}(u) = \mathcal{L}(\mathcal{M}(x), y),$$

For the simulated dataset D'_{train}, the key challenge is that the difficulty measurer for u is not suitable for u' because

$\mathcal{L}(\mathcal{M}(x'), y')$ introduces bias. We then leverage contrastive learning to predict the TSFM comprehension on the representation of u' and hence to determine the difficulty. We first define *TSFM comprehension* and *contrastive pairs*, based on which we introduce the contrastive model and how to estimate the difficulty of u'.

Definition 3.1. *TSFM comprehension on samples*. Let $C_{\mathcal{M}}(u_1, u_2) = |D_{\mathcal{M}}(u_1) - D_{\mathcal{M}}(u_2)| \in \mathbb{R}$ be the comprehension of a pre-trained TSFM $\mathcal{M}$ on two real sample u_1 and u_2. Less value denotes a similar comprehension of $\mathcal{M}$ on u_1 and u_2, and vice versa.

Definition 3.2. *Contrastive pairs*. We define positive and negative contrastive pairs for real u based on the value of $C_{\mathcal{M}}(\cdot, \cdot)$. Given an anchor u, the pair (u, u^+) is positive if $C_{\mathcal{M}}(u, u^+) < \delta$ since $\mathcal{M}$ shows similar comprehension on the two samples; otherwise, the pair is negative, as (u, u^-).

Note that, contrastive pairs construction relies solely on real $\{u\}$ and we set δ to 0.01 based on our experimental analysis. It is a typical phenomenon that two dissimilar samples, which appear dissimilar in terms of time-series patterns (e.g., as measured by DTW similarity), may correspond to a similar TSFM comprehension, indicated by a low value of $C_{\mathcal{M}}(\cdot, \cdot)$. This is related to the uncertain knowledge encapsulated by the TSFM.

The right part of Figure 2 shows the design of our contrastive learning model f. Considering the huge amount of negative pairs, we leverage the classical memory bank structure and adopt temporal convolutional network (TCN) as the encoder since it can be trained efficiently and shows superior performance in capturing daily and weekly seasonality, which are major temporal patterns in building energy time-series. As shown in Eq. 3.1, we apply InfoNCE loss (Oord et al., 2018) and introduce the value of $C_{\mathcal{M}}(u, u_k)$ as the weight ω_k of negative pairs. Here, u_j, u_k are positive and negative samples of u, $sim(\cdot, \cdot)$ calculates the cosine similarity between each pair of data embeddings, and τ is a scaling parameter. After training f, we obtain the difficulty of a simulated sample u' through Eq. 3.1.

$$\mathcal{L}_{cl} = -\log \frac{\sum_{j=1}^{J} \exp(sim(f(u), f(u_j))/\tau)}{\sum_{k=1}^{K} \omega_k \cdot \exp(sim(f(u), f(u_k))/\tau)}$$

$$D_{\mathcal{M}}(u') = D_{\mathcal{M}}(\arg \min_{u \in D_{train}} (sim(f(u), f(u'))))$$

3.2. Training Scheduler

After measuring the difficulty of samples, we leverage a linear continuous scheduler to select training samples at each epoch. Specifically, samples are first sorted by their difficulty. Then, a function $\lambda(t) = \min(1, \lambda_0 + (1 - \lambda_0) \cdot t / T_{\text{grow}})$ decides the percentage of the easiest samples to be used at the t-th epoch, where λ_0 denotes the initial percentage of the easiest samples for training and T_{grow} is the epoch when $\lambda(t)$ grows to 1. Then, let $\mathcal{D} = \{v_i\}_{i=1}^{n}$, the training set at the t-th epoch is given by $\mathcal{D}_t = \{v_i\}_{i=1}^{n \cdot \lambda(t)}$.

4. Evaluation

4.1. Methodology

TSFMs. We adopt two product-level TSFMs, which are Tiny Time Mixer (TTM) (Ekambaram et al., 2024) from IBM and Chronos (Ansari et al., 2024) from Amazon, for adapting to BEF tasks[3].

Datasets. Simulated and real-world public building energy datasets are used for experiments: (1) Buildings-900K (Emami et al., 2023). This dataset contains hourly energy consumption time-series from 900k simulated buildings over two years. (2) Building Data Genome Project (BDG) (Miller et al., 2020). This project aggregates 19 real-world building energy datasets from different locations around the world (totaling 1,636 buildings), where hourly energy meter data over a two-year period are collected for each building. (3) UCI Electricity (Trindade, 2015). This dataset collects electricity consumption data from 370 houses for four years, sampled at 15-minute interval.

Baselines & Metrics. We compare the TSFMs fine-tuned with our method (denoted as TSFM+CCL-FT) against: (1) the original pre-trained TSFMs (denoted as TSFM); and (2) the TSFMs directly fine-tuned without our method (denoted as TSFM+FT). Besides, we adopt three state-of-the-art time-series forecasting models adopted in BEF field for comparisons: LSTM (Chitalia et al., 2020) (a classical RNN architecture for handling sequential data), Autoformer (Jiang et al., 2022) and Temporal Fusion Transformer (TFT) (Giacomazzi et al., 2023) (two transformer-based models tailored for time-series forecasting). We use CV-RMSE for performance evaluation, a standard metric in BEF tasks.

Setup. We select five datasets from BDG together with the UCI dataset as the evaluation set since they cover most building types and climate conditions. In zero-shot setting, all data from the target building are used for testing. In few-shot setting, 10% of data are used for training and the remaining 90% of data are used for testing. The other 15 datasets from BDG and the simulated dataset Buildings-900K are used for TSFMs fine-tuning. The number of fine-tuning steps is set to 1000. The look-back window length and forecast horizon is set by default values of TSFMs during fine-tuning.[4] For evaluation, we set three forecast horizons, i.e., 24, 96, 192, as TSFMs can adapt to different horizons. The experiments are conducted on a Linux server with two NVIDIA GeForce RTX 4090 24GB GPUs.

[3] TTM-5M and Chronos-710M.

[4] 512-96 for TTM and 512-64 for Chronos.

Table 2. Performance comparison (CV-RMSE, lower is better) under zero-shot and few-shot forecasting settings. Improvement ratio of fine-tuned TSFMs as compared to original pre-trained TSFMs is shown at the last row.

Dataset	Forecast horizon	Zero-shot setting						Few-shot setting					
		TSFM: TTM-5M			TSFM: Chronos-710M			TSFM: TTM-5M			TSFM: Chronos-710M		
		Original	+FT	**+CCL-FT**	Original	+FT	**+CCL-FT**	Original	+FT	**+CCL-FT**	Original	+FT	**+CCL-FT**
BDG-Fox	24	0.2175	0.2234	**0.1952**	0.1852	0.1735	**0.1636**	0.2183	0.2101	**0.1947**	0.1717	**0.1566**	0.1588
	96	0.2942	0.2551	**0.2367**	0.2201	0.2174	**0.1918**	0.2616	0.2536	**0.2244**	0.2131	0.2147	**0.1841**
	192	0.3306	0.2725	**0.2565**	0.2570	0.2446	**0.2291**	0.2773	0.2673	**0.2582**	0.2554	0.2477	**0.2224**
BDG-Rat	24	0.3094	0.3023	**0.2714**	0.2095	0.2078	**0.1936**	0.3097	0.2942	**0.2628**	0.2018	0.2024	**0.1843**
	96	0.4348	0.3687	**0.3388**	0.2468	0.2393	**0.2033**	0.3919	0.3624	**0.3164**	0.2294	0.2346	**0.1996**
	192	0.4582	0.3949	**0.3676**	0.2772	0.2516	**0.2486**	0.4204	0.4007	**0.3555**	0.2549	0.2324	**0.2283**
BDG-Bear	24	0.2514	0.2446	**0.2103**	0.1635	0.1611	**0.1525**	0.2546	0.2382	**0.2080**	0.1572	0.1583	**0.1445**
	96	0.3197	0.3009	**0.2586**	0.1919	**0.1843**	0.1860	0.3157	0.2996	**0.2528**	0.1739	0.1791	**0.1718**
	192	0.3162	0.3291	**0.2827**	0.2010	0.2125	**0.1847**	0.3083	0.3198	**0.2779**	0.1918	0.1883	**0.1709**
BDG-Panther	24	0.2736	0.2705	**0.2482**	0.1781	0.1657	**0.1554**	0.2708	0.2683	**0.2423**	0.1586	0.1487	**0.1420**
	96	0.3163	0.3068	**0.2876**	0.2154	0.1981	**0.1612**	0.3041	0.2852	**0.2655**	0.2039	0.1920	**0.1464**
	192	0.3259	0.3175	**0.2983**	0.2227	0.2168	**0.2037**	0.3142	0.3146	**0.2914**	0.2056	0.1923	**0.1848**
UCI	24	0.3848	0.3322	**0.2761**	0.2262	0.2035	**0.1713**	0.2981	0.3173	**0.2666**	0.2215	0.1966	**0.1683**
	96	0.3831	0.3415	**0.2794**	0.2564	0.2358	**0.2106**	0.3620	0.3376	**0.2748**	0.2425	0.2119	**0.1937**
	192	0.4074	0.3684	**0.2955**	0.2759	0.2664	**0.2488**	0.3743	0.3506	**0.2906**	0.2587	0.2544	**0.2402**
Improvement ratio		-	7.8% ↑	**18.3%** ↑	-	4.4% ↑	**12.6%** ↑	-	3.4% ↑	**14.9%** ↑	-	4.1% ↑	**12.7%** ↑

Table 3. Performance comparison of TSFM+CCL-FT and SOTA forecasting models in BEF field.

Dataset	LSTM	Autoformer	TFT	TTM+**CCL-FT**	Chronos+**CCL-FT**
BDG-Fox	0.2797	0.3321	0.2259	0.2257	0.1884
BDG-Rat	0.4132	0.3407	0.2176	0.3115	0.2040
BDG-Bear	0.4308	0.4116	0.2093	0.2462	0.1624
BDG-Panther	0.2231	0.2506	0.3416	0.2664	0.1577
UCI	0.3592	0.1942	0.2125	0.2773	0.2007

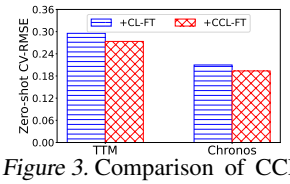

Figure 3. Comparison of CCL method and the variant.

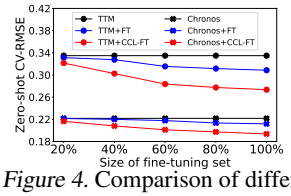

Figure 4. Comparison of different sizes of fine-tuning set.

4.2. Performance Result

Overall performance. We evaluate our method and baselines in zero-shot and few-shot settings under three forecast horizons in Table 2. Overall, TSFM+CCL-FT consistently outperforms TSFM and TSFM+FT in zero-shot setting, with average improvements in CV-RMSE at 18.3%, 11.3% for TTM and 12.6%, 8.5% for Chronos. Similar results are observed in few-shot setting where our method surpasses baselines by 14.9%, 11.9% for TTM and 12.7%, 8.9% for Chronos. Besides, as CV-RMSE <0.3 is an industrial requirement defined by ASHRAE (ASHRAE, 2002) for deployable forecasting models, we observe that on each dataset, there are cases where our method successfully reduces the error of pre-trained TSFMs to less than 0.3.

Next, we compare TSFM+CCL-FT with three state-of-the-art baseline forecasting models. Here, TSFM+CCL-FT is evaluated under few-shot setting while the baselines are first trained using the first 50% of data from each test building and then tested on the remaining 50% of data. As shown in Table 3, we observe that Chronos fine-tuned with the CCL strategy outperforms the best of baselines on almost all datasets, with an improvement of 14.6% on average. For TTM, although its performance is improved with our method, it still lags behind the best baseline, particularly on the BDG-Rat and UCI datasets.

Ablation Study. To take a closer look at the contribution of the designed contrastive-aware difficulty measurer in our method, we implement a variant of our method named TSFM+CL-FT, which simply uses the performance of TSFMs as difficulty for both real and simulated samples. The zero-shot performance of our method and this variant is compared in Figure 3. We observe that TSFM+CCL-FT outperforms TSFM+CL-FT by 7.4% and 7.7% for TTM and Chronos, respectively. With further analysis, we find that TSFM+CL-FT is still better than TSFM-FT under the same experiment setting, which validates the effectiveness of curriculum learning in enhancing TSFMs adaptation.

Next, we study the effect of the size of fine-tuning set on the performance of TSFMs. In Figure 4, we evaluate the corresponding versions of fine-tuned TSFMs under varying proportion of fine-tuning set. The performance of the original pre-trained TSFMs is included for reference. The results indicate the superior performance of our method on each setting. Specifically, we observe that the improvement over baseline increases along with the size of fine-tuning set from 2.9% to 13.2%. This implies the capability of our method in handling a larger and more complicated dataset.

5. Conclusion

This work identifies a discrepancy between existing TSFMs and the in-use performance of TSFMs in a specific domain: building energy forecasting. To bridge this gap, we have introduced a new curriculum learning-based training method in this context, to identify the difficulty of both real-world and simulated building data, and based on this to manage

the order of train set during the process of TSFM adaptation. The experiments show that the proposed curriculum design can greatly improve the zero/few-shot performance of company-level TSFM products.

Acknowledgements

Dan Wang's work is supported in part by RGC GRF 15200321, 15201322, 15230624, 15239925, ITC ITF-ITS/056/22MX, ITS/052/23MX, and PolyU 1-CDKK, G-SAC8, K-ZYAP.

References

Ansari, A., Stella, L., et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

ASHRAE, A. Ashrae guideline 14: measurement of energy and demand savings. *American Society of Heating, Refrigerating and Air-Conditioning Engineers*, 35:41–63, 2002.

Chitalia, G., Pipattanasomporn, M., et al. Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. *Applied Energy*, 278:115410, 2020.

Crawley, D., Lawrie, L., et al. Energyplus: creating a new-generation building energy simulation program. *Energy and buildings*, 33(4):319–331, 2001.

Ekambaram, V., Jati, A., et al. Tiny time mixers (ttms): Fast pre-trained models for enhanced zero/few-shot forecasting of multivariate time series. *CoRR*, 2024.

Emami, P., Sahu, A., et al. Buildingsbench: A large-scale dataset of 900k buildings and benchmark for short-term load forecasting. *Advances in Neural Information Processing Systems*, 36:19823–19857, 2023.

Giacomazzi, E., Haag, F., et al. 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*, pp. 353–360, 2023.

Jiang, Y., Gao, T., et al. Very short-term residential load forecasting based on deep-autoformer. *Applied Energy*, 328:120120, 2022.

Liang, Y., Wen, H., et al. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6555–6565, 2024.

Miller, C., Kathirgamanathan, A., et al. The building data genome project 2, energy meter data from the ashrae great energy predictor iii competition. *Scientific data*, 7(1):368, 2020.

Mulayim, O., Quan, P., et al. Are time series foundation models ready to revolutionize predictive building analytics? In *Proceedings of the 11th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pp. 169–173, 2024.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Trindade, A. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

Wang, X., Chen, Y., et al. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

Wang, Y., Yue, Y., et al. Efficienttrain: Exploring generalized curriculum learning for training visual backbones. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5852–5864, 2023.

Xu, Y. and Wang, D. Understanding the willingness to share building data by a social study based on privacy calculus theory. In *ACM BuildSys*, pp. 59–68, 2022.