MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Synthesizing Building Operation Data with Generative Models: VAEs, GANs, or Something In Between?

Salatiello, Alessandro; Wang, Ye; Wichern, Gordon; Koike-Akino, Toshiaki; Yoshihiro, Ohta; Kaneko, Yosuke; Laughman, Christopher R.; Chakrabarty, Ankush

TR2023-072 June 21, 2023

Abstract

The generation of time-series profiles of building operation requires expensive and time-consuming data consolidation and modeling efforts that rely on extensive domain knowledge and need frequent revisions due to evolving energy systems, user behavior, and environmental conditions. Generative deep learning may be used to provide an automatic, scalable, data-source-agnostic, and efficient method to synthesize these artificial time-series profiles by learning the distribution of the original data. While a range of generative neural networks have been proposed, generative adversarial networks (GANs) and variational autoencoders (VAEs) are most popular models; GANs typically require considerable customization to stabilize the training procedure, while VAEs are often reported to generate lower-quality samples compared to GANs. In this paper, we propose a network architecture and training procedure that combines the strengths of VAEs and GANs by incorporating Regularized Adversarial Fine-Tuning (RAFT). We imbue the architecture with conditional inputs to reflect ambient/outdoor conditions and operating conditions, and demonstrate its effectiveness by using operational data collected over 585 days from SUSTIE: Mitsubishi Electric's net-zero energy building. Comparing against classical GAN, VAE, Wasserstein-GAN, and VAE-GAN, our pro- posed conditional RAFT-VAE-GAN outperforms its competitors in terms of mean accuracy, training stability, and several metrics that ascertain how close the synthetic distribution is to the measured data distribution.

ACM e-Energy Conference 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Synthesizing Building Operation Data with Generative Models: VAEs, GANs, or Something In Between?

Alessandro Salatiello
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA

Ye Wang
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA

Gordon Wichern
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA

Toshiaki Koike-Akino
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA

Yoshihiro Ohta
Mitsubishi Electric Corporation
Kamakura, Kanagawa, Japan

Yosuke Kaneko
Mitsubishi Electric Corporation
Kamakura, Kanagawa, Japan

Christopher Laughman
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA

Ankush Chakrabarty
Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA, USA
chakrabarty@merl.com

ABSTRACT

The generation of time-series profiles of building operation requires expensive and time-consuming data consolidation and modeling efforts that rely on extensive domain knowledge and need frequent revisions due to evolving energy systems, user behavior, and environmental conditions. Generative deep learning may be used to provide an automatic, scalable, data-source-agnostic, and efficient method to synthesize these artificial time-series profiles by learning the distribution of the original data. While a range of generative neural networks have been proposed, generative adversarial networks (GANs) and variational autoencoders (VAEs) are most popular models; GANs typically require considerable customization to stabilize the training procedure, while VAEs are often reported to generate lower-quality samples compared to GANs.

In this paper, we propose a network architecture and training procedure that combines the strengths of VAEs and GANs by incorporating **R**egularized **A**dversarial **F**ine-**T**uning (RAFT). We imbue the architecture with conditional inputs to reflect ambient/outdoor conditions and operating conditions, and demonstrate its effectiveness by using operational data collected over 585 days from SUSTIE: Mitsubishi Electric's net-zero energy building. Comparing against classical GAN, VAE, Wasserstein-GAN, and VAE-GAN, our proposed conditional RAFT-VAE-GAN outperforms its competitors in terms of mean accuracy, training stability, and several metrics that ascertain how close the synthetic distribution is to the measured data distribution.

CCS CONCEPTS

• **Computing methodologies → Modeling methodologies**; • **Applied computing → Computer-aided design**.

KEYWORDS

Generative models, adversarial learning, variational methods, energy systems, real building data, net-zero energy building.

ACM Reference Format:
Alessandro Salatiello, Ye Wang, Gordon Wichern, Toshiaki Koike-Akino, Yoshihiro Ohta, Yosuke Kaneko, Christopher Laughman, and Ankush Chakrabarty. 2023. Synthesizing Building Operation Data with Generative Models: VAEs, GANs, or Something In Between?. In *Proceedings of International Workshop on Applied Machine Learning for Intelligent Energy Systems (AMLIES '23).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Traditional processes for developing stochastic physics-based models of building operation are typically expensive and time-consuming, as their formulation requires extensive domain knowledge, mining of time-use surveys, and the implementation of computationally intensive data formatting, calibration, and modeling procedures. Moreover, such models require frequent revision due to the introduction of new components or configurations into the system, such as the replacement of appliances with energy-efficient variants or as occupancy profiles evolve over time.

Deep generative models [5] offer a convenient automated alternative to this manual process by generating time-series profiles of interest without extensive domain knowledge in a manner agnostic to the sources of data (e.g. sensors, surveys, etc.) by learning directly from the experimental dataset. The efficiency and scalability of these models facilitates the generation of arbitrarily large artificial datasets, which are very hard to obtain, especially for new or experimental building energy systems.

AMLIES '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

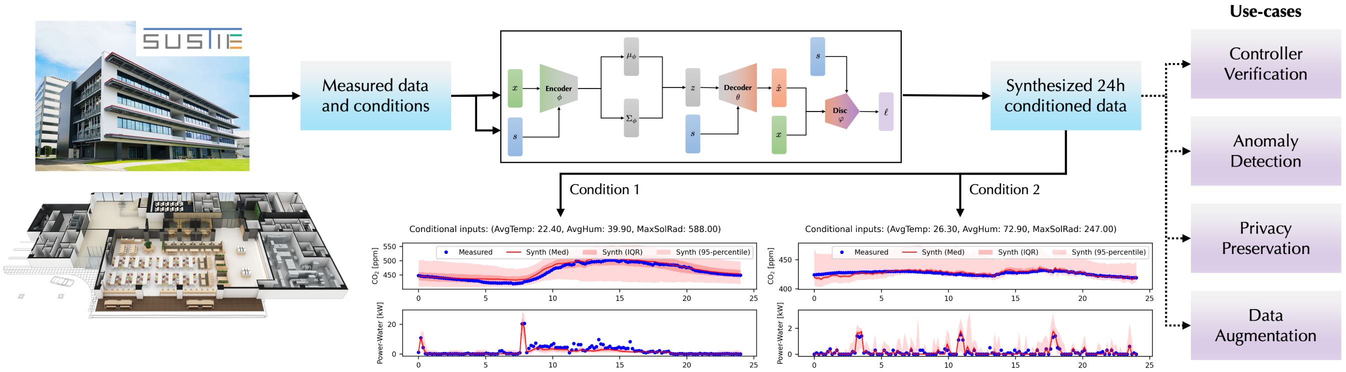

Figure 1: Overview and potential use-cases of the proposed generative modeling approach.

Natural use-cases for these generated datasets include their use in the training and evaluation of data-driven forecasting models [28, 40], as well as in the direct sequence generation of stochastic occupant behavior over specified time periods. Such tools are being increasingly used in the building energy systems domain in the design and operation of energy-efficient controllers [4, 20, 33]. Such "synthetic" datasets can be used for a wide variety of purposes, such as reducing overfitting in forecasting models [37], to evaluate the robustness of building control systems [39], to enable the dissemination of anonymous data [43], and to identify potential systemic malfunctions [19]. An overview of a typical generative modeling pipeline and some popular use-cases are illustrated in Fig. 1.

Recent work in the building energy systems domain has shown that deep generative models are effective at capturing the distribution of single-output operational building profiles, including energy consumption [44, 48], energy yield [10], cooling load [13], thermal comfort [8], and occupancy profiles [7]. A handful of recent studies have also examined the construction of approximate multi-dimensional distributions of building performance profiles [14, 21, 50]. While most prior work has only assessed the intrinsic quality of the learned distributions or showed their usefulness as a data augmentation tool for forecasting models [3, 12, 41, 45] and controllers [14], some recent work has also showcased interesting applications including fault detection [25, 26, 47], and controller validation [21].

Most applications in the building energy systems domain are based on either of the two most popular deep generative models: Generative Adversarial Networks (GANs) [15] and Variational AutoEncoders (VAEs) [22], which are known to suffer from convergence issues leading to mode collapse [36] and underfitting [46], respectively. Even though GAN-based (e.g., Conditional GAN [21, 31, 48], Auxiliary Classifier GAN [16, 34, 48], TimeGAN [3, 14, 49]), and VAE-based (e.g., Conditional VAE [12, 38, 42]) architectures have been used to better handle the characteristics of the building data of interest, their combination has not been well explored. Such combinations have significant potential, given the proven effectiveness of hybrid models that leverage the strengths of GANs and VAEs [23, 24, 30]. Initial work in exploring these combinations include [35] which captured the distribution of energy consumption and photovoltaic production profiles using a VAE-GAN [23], and [50] which captured the distribution of solar irradiance time series using a combination of an ID-GAN [24] and a DoppelGANger [27].

In this work, we introduce a new framework to generate conditional multivariate time-series of building operation, which builds on the benefits of VAEs and GANs, while addressing some of their inherent drawbacks. The main *contributions* of our work can be summarized as follows: (i) we propose RAFT-VAE-GAN (or RAFT-VG to be succinct): a new deep generative model and training procedure to synthesize arbitrarily large multivariate building operation time-series profiles, which hinges on VAE-based pre-training and regularized adversarial fine-tuning (RAFT); (ii) we show that the proposed framework provides a stable training procedure and allows realistic approximations of the true data distribution; (iii) we compare RAFT-VG with baseline generative models, and show that our model outperforms the competing models in terms of similarity with the ground-truth distribution and training stability; and, (iv) we demonstrate the effectiveness of combining VAE-based pre-training with GAN-based fine-tuning by studying the trade-off between the amount of pre-training and fine-tuning.

The rest of the paper is organized as follows: Section 2 introduces conditional generative modeling with VAEs and GANs, and elaborates on their strengths and weaknesses. Section 3 describes RAFT-VG: the new architecture we propose in this work to overcome such weaknesses. Section 4 describes the dataset and the metrics we used to validate our architecture, and presents the results of our validation experiments. Finally, Section 5 summarizes the main findings of our work to discuss potential applications and future directions.

Notations. We adopt standard notation for $\mathbb{R}^n$ and $\mathbb{Z}^n$ to represent real-valued and integer-valued vectors, respectively. We denote ground-truth data by $x \in X$, conditional variables $s \in S$, and latent/nuisance variables by $z \in Z$; the uppercase notation implies sets. Synthetic data are denoted by $\hat{x} \in X$. Any boldface quantity indicates a dataset or collection of individual variables, for instance $\mathbf{x} := \{x\}_\mathcal{I}$ for some index set $\mathcal{I} \subset \mathbb{Z}$. A multi-dimensional Gaussian density function with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ is denoted by $\mathcal{N}(\mu, \Sigma)$. By $z \sim \mathcal{N}(0, I)$, we mean that the vector z is drawn from the standard, multivariate normal distribution. The expectation operator is given by $\mathsf{E}(\cdot)$ and the Kullback-Leibler divergence (KLD) between two distributions π_1

and π_2 is written as $\mathsf{KLD}(\pi_1 || \pi_2)$. We use $\|\cdot\|_2 \equiv \|\cdot\|$ to denote the 2-norm. The natural logarithm is denoted $\log(\cdot)$ and base-10 logarithm $\log_{10}(\cdot)$.

2 PRELIMINARIES

In this section, we present a brief summary of conditional generative modeling with VAEs and GANs. Throughout the paper, we may omit the qualifier 'conditional' for simplicity, but unless specified, by VAE and GAN we mean conditional VAE and conditional GAN.

Formally, the generative learning problem involves estimating the underlying conditional distribution $p_{x|s}$ from joint data samples of (x, s). Under both approaches, a generative model implicitly captures the distribution, while providing a method to synthesize samples of the learned distribution.

2.1 Conditional VAEs

In conditional VAEs [22, 38], the generative model is specified by the distribution $p_\theta(x|s, z)$, where z is sampled from a latent prior distribution $p(z)$. This implicitly specifies the conditional distribution:

$$p_\theta(x|s) = \int p_\theta(x|s, z) p(z)\, \mathrm{d}z.$$

In principle, the learning objective is to maximize the expected log-likelihood, i.e., $\max_\theta \mathsf{E}[\log p_\theta(x|s)]$. However, this implicit conditional distribution is generally intractable, which motivates the introduction of a variational posterior $q_\phi(z|x, s)$ that approximates the actual posterior: $p_\theta(z|x, s) = p(z)p_\theta(x|s, z)/p_\theta(x|s)$. This q_ϕ is utilized in a variational lower bound of the expected log-likelihood, also known as the evidence lower bound (ELBO):

$$\mathsf{E}[\log p_\theta(x|s)] \geq \mathsf{E}\left[\log p_\theta(x|s, z) + \mathsf{KLD}\big(q_\phi(z|x, s) \| p(z)\big)\right], \quad (1)$$

where the expectation is computed with respect to $z \sim q_\phi(z|x, s)$, and (x, s) are drawn from the data distribution. The parameters (θ, ϕ) of the generative model $p_\theta(x|s, z)$ and variational posterior $q_\phi(z|x, s)$ are jointly optimized to maximize the ELBO (1). Note that the variational posterior is typically parameterized as a conditional Gaussian: $q_\phi(z|x, s) = \mathcal{N}(z; \mu_\phi(x, s), \Sigma_\phi(x, s))$, with the mean vector μ_ϕ and diagonal covariance matrix Σ_ϕ given by parametric functions of (x, s). With the typical assumption of a latent prior distribution being the standard Gaussian distribution, $p(z) = \mathcal{N}(0, I)$, the KLD term in (1) is readily tractable and differentiable [22].

The variational posterior $q_\phi(z|x, s)$ can be viewed as an encoder that induces a probabilistic map from x to a latent representation z, conditioned on s. The generative model $p_\theta(x|s, z)$ can be viewed as a decoder that recovers likelihoods for x, conditioned on s, from a sampled latent representation z. This decoder is also commonly parameterized as a conditional Gaussian, $p_\theta(x|s, z) = \mathcal{N}\big(x; \hat{x}_\theta(s, z), I\big)$, where the mean vector $\hat{x}_\theta$ is a parametric function of (s, z) and the covariance is the identity matrix. This simplifies the first term of the ELBO in (1) to be essentially a negative reconstruction loss, i.e., shift-scale of mean-square error (MSE):

$$\mathsf{E}[\log p_\theta(x|s, z)] = -\frac{1}{2}\|x - \hat{x}_\theta(s, z)\|^2 + c,$$

where c is a constant that does not impact the optimization.

Given a trained VAE, the decoder can be used to generate synthetic data by conditionally sampling from the model $p_\theta(x|s, z)$. This is done by drawing a latent vector z from its prior distribution $p(z) = \mathcal{N}(0, I)$, and subsequently, for a given s, employing the generative model to specify the distribution $p_\theta(x|s, z)$ from which the synthetic data should be sampled. For example, with the Gaussian decoder described above, we generate the corresponding sample as simply the mean vector $\hat{x}_\theta(z, s)$. This omits the variation implied by the Gaussian decoder model, with only the sampling of z providing randomness to the sample generation.

2.2 Conditional GANs

In conditional GANs [15, 31], the conditional generator $G_\vartheta : S \times Z \longrightarrow X$ is trained in an adversarial framework that involves a discriminator $D_\varphi : X \times S \longrightarrow [0, 1]$ that aims to distinguish between genuine samples drawn from the actual data distribution versus synthetic samples produced by the generator. The training process is based on a min-max optimization problem given by

$$\min_\vartheta \max_\varphi \mathsf{E}\big[\log(1 - D_\varphi(x, s)) + \log D_\varphi(G_\vartheta(s, z), s)\big], \quad (2)$$

where the latent variable is drawn from its prior $z \sim p(z)$, and (x, s) are drawn from the data distribution.

For a fixed generator G_ϑ, the inner maximization of the GAN objective given in (2) can be viewed as training the discriminator to perform binary classification between sample pairs (x, s) drawn from the true data distribution versus pairs $(\hat{x}, s)$, where x has been replaced by the synthetic samples produced by the generator $\hat{x} = G_\vartheta(s, z)$. Indeed, the objective function for training the discriminator is essentially the binary cross-entropy loss. In principle, for the optimal discriminator on the inner maximization, the objective for training the generator is the Jensen–Shannon divergence between the true data distribution and the distribution of the synthetic pairs $(\hat{x}, s)$ [15].

In practice, however, the min-max optimization of (2) is often tackled with the heuristic method of alternating between stochastic gradient ascent and descent steps for the inner and outer optimizations (or performing a few discriminator updates between each generator update). This heuristic has both the practical benefit of avoiding the computational cost of iterating the inner maximization to convergence and empirically helps to stabilize the GAN training by avoiding vanishing or exploding gradients (see [1] which investigates the fundamental GAN stability issues).

3 SOMETHING IN BETWEEN: RAFT-VG

For many natural data domains, GANs synthesize more realistic samples compared to VAEs, as the discriminator serves as a learned perceptual loss for training the generator. VAEs may be inherently limited by the choice of the generative model, such as isotropic Gaussian statistics (as implied by the MSE loss) being overly simplistic and leading to blurrier samples. Conversely, GANs are notoriously difficult to train, due to the instability of adversarial optimization, while VAEs are relatively straightforward. Thus, ideally, one would wish to combine the perceptual quality of GANs with the stability and ease of training of VAEs.

Our proposed architecture to merge the benefits of VAEs and GANs is presented in Fig. 2. The proposed RAFT-VG comprises a conditioned encoder, a conditioned decoder, and a conditioned discriminator. The encoder takes an entire time-series profile x as

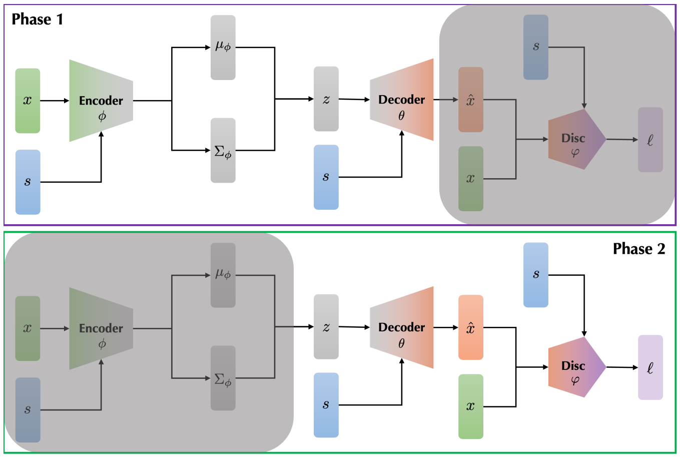

Figure 2: Two-phase training procedure for RAFT-VG. Dark shading indicates frozen weights.

an input, along with a conditioning input s which is appended to x. The encoder then generates mean and variance vectors that define a probabilistic latent vector z, assumed to have been extracted from a Gaussian distribution. The decoder converts this latent vector, along with the appended conditioning input into a reconstructed time-series profile $\hat{x}$. Rather than constructing a separate generator, we utilize the decoder in an overlapping role to also serve directly as the generator. That is, one can randomly generate latent variables z, which can be decoded to produce corresponding $\hat{x}$. By comparing with the true data x and assigning plausibility scores via the discriminator (which is also conditioned on s), we incorporate the GAN philosophy into our proposed method. Rather than incorporating the discriminator in an *ad hoc* manner, a major contribution of this paper is to introduce a regularized adversarial fine-tuning (RAFT) algorithm. The objective of regularizing the weights of the decoder in the RAFT procedure promotes selecting decoder weights such that the encoder-decoder pair remain 'close to' inverse functions of each other, assuming the encoder and decoder are sufficiently smooth; c.f. [6]. The quality of the final decoded output is enhanced by fine-tuning the decoder (i.e., generator) by utilizing the discriminator.

To explain further, the training mechanism for the RAFT-VG is composed of two decoupled phases since we have observed that, in practice, simultaneously trying to train the VAE components and the GAN components often requires significant manual hyperparameter selection and design effort. Therefore, the first phase of the training, shown in the upper subplot of Fig. 2, freezes the discriminator weights, and trains a classical conditioned VAE with the building time-series profiles as described in section 2.1. This phase of the training is stable, and at the end of the training, we expect that the encoder has learned a distribution that can effectively project the time-series profiles to a reduced-dimensional latent space, and that the decoder is a suitable inverse transformation that lifts the latent vectors to well-reconstructed time-series profiles. In the second phase of training as shown in the lower subplot of Fig. 2, we freeze the encoder and train the decoder and the discriminator. As explained in the previous paragraph, we use the decoder as a conditional generator, and therefore, the training in this phase follows a similar regime to that discussed in section 2.2. However, we make two major changes to improve GAN training stability while maintaining the encoder-decoder consistency, and we call this the RAFT procedure.

The first change is to incorporate a Wasserstein GAN loss for the discriminator, which is an idea proposed in a widely adopted architecture known as the Wasserstein GAN (W-GAN) [2] for elevating GAN performance and stabilizing the training regime. The W-GAN employs an alternative training objective given by

$$\min_{\theta} \max_{\varphi} \mathsf{E} \left[D_\varphi(x,s) - D_\varphi(G_\theta(z,s),s) \right], \tag{3}$$

where the discriminator $D_\varphi : X \times S \to [0,1]$ should be constrained to be 1-Lipschitz continuous. In principle, with the optimal discriminator in the inner maximization, the objective for the outer generator optimization becomes the Wasserstein distance between the true data distribution $p_{x,s}$ and that of the synthetic pairs $(\hat{x}, s)$. There are several approaches to enforce Lipschitz continuity on the discriminator, such as weight clipping [2] or gradient penalization [17]. We utilize the popular and effective method of spectral normalization [32], which involves normalizing each weight matrix with a running estimate of its largest singular value, in order to approximately enforce 1-Lipschitz continuity.

The second change is motivated by a desire to maintain consistency of the encoder-decoder pair. Since the encoder is frozen, the encoding is fixed, but we allow the decoder weights to vary. To prevent the decoder weights from changing to an extent that they are too far to constitute an inverse function of the encoder, we regularize the decoder weights. That is, we try to restrict θ to be close to $\theta_\star$ such that the encoder-decoder pair remain capable of reconstructing the inputs. In particular, we add an $\mathcal{L}_2$ regularizer to the W-GAN loss (3), which yields our RAFT-VG loss for the second phase:

$$\min_{\theta} \max_{\varphi} \mathsf{E} \left[D_\varphi(x,s) - D_\varphi(\hat{x}_\theta(z,s),s) \right] + \lambda \|\theta - \theta_\star\|_2^2, \tag{4}$$

where $\lambda > 0$ is a scalar, $\theta_\star$ denotes the weights of the decoder obtained after the encoder-decoder training in the first phase, and $\hat{x}_\theta(z,s)$ is the output of the decoder, with the latent z sampled from its prior $p(z) = \mathcal{N}(0, I)$.

3.1 Related Conventional Approaches

Of course, we are not the first to investigate combining the benefits of VAEs and GANs. Herein, we briefly summarize the most relevant works and delineate how RAFT-VG is different and may have some advantages over these previously proposed combinations.

In VAE-GAN [23], the authors introduce a discriminator during VAE training such that it provides a learned perceptual loss to refine the VAE. Unlike our RAFT-VG approach, the VAE-GAN trains all three network components (i.e., the encoder, decoder, and discriminator) simultaneously in an adversarial training framework, which makes it more susceptible to GAN-like training stability issues, as we will report in the next section.

In an alternative adversarial approach shared between [29, 30], the discriminator is applied to the latent representation and serves to approximate the KLD term of the ELBO loss in (1), which allows for more general encoder architectures and latent prior distributions. Unlike RAFT-VG, the discriminator is connected to a latent rather than a synthetic data sample. Another adversarial VAE training approach [9, 11] applies the discriminator to distinguish between pairs of data samples and the representations extracted by the encoder versus synthetic samples and the corresponding latent representations from which they were generated by the decoder. The architecture is a GAN trained similarly to a VAE, not a combination of elements from VAEs and GANs.

TimeGAN [49] involves training an autoencoder, with a recurrent network architecture, that encodes data sequences into a sequential latent space, and a GAN model that aims to generate these sequential latent representations. This structured approach for sequential data aims to enforce statistical consistency across temporal transitions. In contrast, we did not apply recurrent architectures in our approach, since we found our lower dimensionality and sequence lengths to be manageable with less complex methods.

The ID-GAN framework [24] also takes a two-stage approach, where the first stage trains a VAE, before applying adversarial training in the second stage. Their approach differs in that their second stage discards the VAE decoder and instead trains a generator model from scratch (in contrast to the decoder fine-tuning of our method), which introduces significantly more variables for training.

4 CASE STUDY: SYNTHETIC DATA GENERATION FOR SUSTIE BUILDING

4.1 Data Collection

SUSTIE is a next-generation office building with a total floor area of approximately 6456 m^2, designed to research and demonstrate energy savings and workers' health and comfort. The name SUSTIE combines the words "Sustainability" and "Energy". SUSTIE has nine experimental rooms (offices), where around 260 office workers work, as well as an open-feel atrium area, a cafeteria and a gym. SUSTIE is the first building in Japan to achieve the highest level of Japan's net Zero Energy Building (ZEB) certification from the Building-Housing Energy-efficiency Labeling System (BELS), the highest level of CASBEE Wellness Office, a certification system for health and comfort in Japan, and the platinum rank of WELL certification[1].

SUSTIE collects electrical energy, meteorological, indoor environment, occupancy, and equipment operation data to analyze and control energy consumption and comfort during building operations. The electrical energy is measured for each type of equipment (air-conditioning, ventilation, lighting, hot water supply and elevators) and for each room. The meteorological data are measured by weather sensors installed on the roof of the building, including outdoor temperature and humidity, wind speed and direction, rainfall, and solar radiation. The indoor environment is measured by a total of 330 sensors including air quality sensors (temperature, humidity, CO, CO_2, pollen, formaldehyde, etc.) and illumination level sensors. The number of people in each room is counted by the access control system using card readers installed in each area. Equipment operating data collected by the equipment controller includes operating mode, temperature settings, measured temperatures, flow rates and light dimming rates. The data is collected 24 hours a day, with a sampling rate of 1 minute by the building management system.

4.2 Experimental Setup

In this paper, we use a subset of this collected data: namely CO_2, along with power[2] consumed by hot water generation systems, lighting, and ventilation. This dataset is down-sampled to obtain a sampling rate of 10 min per sample by selecting the median value in every 10 min for each signal; thus, we have 144 samples per day for each of the 4 signals under consideration. We find this an effective approach to tackle missing sensor values without imputing or discarding data. The final dataset has a size of $585 \times 144 \times 4$, since we have 585 valid days worth of data. As multiple sensors measure the CO_2 concentration, we average over sensors since they could be placed in the same living space at different locations. We also sum the power signals to compute the total power usage.

Inspired by the arguments made in [21], we choose the conditional inputs to be: (i) a binary variable indicating whether a specific day is a workday or a holiday; (ii) the median outdoor dry-bulb temperature over 24 hr; (iii) the median outdoor relative humidity over 24 hr; and, (iv) the maximum solar radiation over 24 hr. All conditional inputs are normalized to lie with the range $[0, 1]$. Vapor compression cycle variables are not selected as conditioning inputs because the conditional inputs provided to the generator must be decoupled from the building and cycle variables that are being controlled. As our primary use case is in using deep generative models that provide realistic scenarios for assessing the building control performance of simulation models that include both building and HVAC equipment dynamics, we restrict the conditioning inputs to only include exogenous inputs to the overall system-of-systems. As a result, we do not select the same conditioning inputs as those of [21], such as room temperatures/humidities, unmet setpoints, and cooling capacities.

In formulating these methods, we restrict ourselves to scalar conditional inputs (e.g., the median temperature over a 24 hr period), as opposed to vector conditional inputs (e.g., a vector containing the temperature every hour) as this simplifies the specification of the conditioning input at inference time due to the fact that only a single value is required. Furthermore, imbalances in the prior distribution of the training data (e.g., there are very few hot days) can be easily compensated by increasing the weight to the loss function for under-represented condition samples, or oversampling the underrepresented conditions during training to ensure they are accurately represented in our model. Conversely, under-represented conditions are more difficult to identify when using vector conditioning inputs, as the data is generally more sparse due to the higher dimension. In this case, identifying under-represented conditions requires the use of dimensionality reduction techniques (e.g., principal component analysis: PCA), which are generally less interpretable than the median. Lastly, the use of scalar conditional inputs greatly simplifies the network architecture and reduces the overall parameter count, as we do not have to concatenate high dimensional conditioning vectors to network layer inputs or add

[1] https://www.mitsubishielectric.com/en/about/rd/sustie/index.html

[2] We employ numerical differentiation to compute power from energy sensors.

additional layers that project the high dimensional inputs to a lower dimension prior to concatenation. For similar reasons, we do not employ any positional encoding.

The encoder of the RAFT-VG has an input dimension of $B \times 580$, where B is the batch-size (we select $B = 128$), and $580 = 144 \times 4$ flattened signals plus 4 conditional input scalars. The conditional encoder has 3 hidden layers with 256 neurons per layer, activated by smooth ELU functions ($\alpha = 0.1$), and two outputs: a mean and a log variance, both of which are 64-dimensional (i.e., the latent dimension of the VAE is 64). The conditional decoder of the RAFT-VG is deeper, with 3 linear hidden layers followed up with 3 convolutional layers of kernel widths 15, 7, and 5, respectively, to promote decoder outputs to be smooth, as we observed that the outputs of the VAE may sometimes contain unwanted high-frequency oscillations and jumps without convolutional filtering. Since our signals are all non-negative, the output layer of the decoder is passed through a ReLU layer, while all other layers in the decoder are activated by Leaky ReLU functions ($\alpha = 0.1$). The conditional discriminator has an input layer and 3 hidden layers, all of which are regularized by a spectral normalization layer, activated by Leaky ReLUs ($\alpha = 0.1$), and passed through a dropout layer with a rate of 0.3. The dropout is a particularly effective tool in phase-2 training for the RAFT-VG because it prevents the discriminator from overfitting. The output dimension of the discriminator is 1. The choice of activation functions and other hyperparameters are informed by performing hyperparameter tuning on a 10% hold-out validation set.

In the first phase of training, we employ an Adam optimizer for the VAE training with a learning rate of 10^{-3}, and fine-tune the decoder with a learning rate of 10^{-5} and set Adam's $\beta_1 = 0.5$ in the second phase. The discriminator is trained with RMSprop with a learning rate of 10^{-5}; these smaller learning rates affirm that the phase-2 training is geared towards fine-tuning. The discriminator is updated 5× for each decoder update and $\lambda = 10$. Both phases are allotted 5000 epochs for training, with each epoch covering the entire dataset via mini-batches. Our data loader is designed to prevent the imbalance arising from fewer weekends compared to weekdays, and therefore we over-sample weekend data based on the binary conditioning input in each mini-batch to promote balanced learning.

4.3 Performance Evaluations

4.3.1 Generation of Synthetic Sequences. We begin by showing building time-series profiles generated over 24 hr for two conditioning inputs in Fig. 3. In the figure, the left subplot is conditioned on a holiday and the sky is overcast: therefore, the conditioning inputs have a low average temperature, low humidity, and low maximum solar radiation over the entire day. Conversely, the right subplot depicts a workday with considerably higher average temperature, higher humidity, and much higher maximum solar radiation, indicating a sunny day. The two measured data streams are shown using blue dots, while the light pink lines are individual synthetic data traces, and the pink band is a 95% confidence interval (CI) calculated from the statistics of 1000 synthetic samples generated by RAFT-VG.

The CIs clearly superimpose most of the measured data, indicating that for both conditioning inputs, the synthetic distribution learned by the RAFT-VG covers the measured data, indicating that the distribution has been learned reasonably well. In addition, the difference in the shapes of the CIs indicate that the conditioning inputs influence the inferred distributions. The rise in the CIs for carbon-dioxide levels during work hours in the right plot, in contrast with the relatively uniform CI on the holiday, indicates a good correlation with learned distributions and measured data, as well as with domain knowledge-based expectations. The single sharp spike of hot water around 8AM on the workday (which does not exist on holidays) is another example of the conditioning effects having been learned well: the reason for this spike is because the hot water used in the cafeteria for the day is boiled in the morning and stored in a tank for use. We also note that the CIs for hot water on the holiday (left subplot) exhibit many kinks, and do not vary smoothly like the other CIs. As is evident from the corresponding blue data dots, these kinks are caused by the fact that the use of hot water over weekends and national holidays is highly irregular and depends on individual behaviors, rather than synchronized behaviors across the company. The learned distribution, in trying to cover all the vagaries of individual behaviors, therefore deduces that it is best to forego reconstruction accuracy (e.g. mean behavior) and instead resorts to learning wide uncertainty bands around the mean behavior.

4.3.2 Performance Comparison with Popular Generative Models. For the purposes of comparison, we use three metrics. The first is the normalized root-mean-squared error (NRMSE):

$$\mathsf{NRMSE}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \frac{1}{T}\|\mu_{\mathbf{x}} - \mu_{\hat{\mathbf{x}}}\|,$$

which attempts to quantify how well the synthetic data samples agree with the measured data on average. Our second performance metric is tailored to understanding how close the learned distribution, from which synthetic data is sampled, is to the measured data distribution. To this end, we employ the KLD:

$$\mathsf{KLD}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \log \frac{\sigma_{\mathbf{x}}}{\sigma_{\hat{\mathbf{x}}}} + \frac{\sigma_{\hat{\mathbf{x}}}^2 + \left(\mu_{\mathbf{x}} - \mu_{\hat{\mathbf{x}}}\right)^2}{2\sigma_{\mathbf{x}}^2} - \frac{1}{2}.$$

Finally, we investigate mode-collapse: a phenomenon often seen in poorly trained GANs, where the generator samples from a distribution with a very restricted variance (hence, collapsed) and therefore is only capable of generating extremely similar data sequences. While it is not simple to quantify mode-collapse, one popular metric is the Fréchet inception distance (FID) score [18]. Unfortunately, calculating the FID score requires pre-trained Inception-V3 models that are suited to image generation, not building time-series profiles. This is why we use its precursor, the Fréchet distance (FD)

$$\mathsf{FD}(\mathbf{x}, \hat{\mathbf{x}}) \triangleq \|\mu_{\mathbf{x}} - \mu_{\hat{\mathbf{x}}}\|^2 + \|\sigma_{\mathbf{x}} - \sigma_{\hat{\mathbf{x}}}\|^2$$

as our metric to ascertain mode-collapse, since it can be generalized to wider types of data than just images. For all the above, $\mu_{\mathbf{x}}$ denotes the expected value of the true distribution $p(x, s)$ and $\Sigma_{\mathbf{x}}$ is its corresponding covariance matrix. The notation is consistent for the synthetic distribution $p(\hat{x}, s)$ and its moments.

To compute these metrics, we evaluate a Monte–Carlo estimate of the metric with the effects of the conditionals marginalized out. For example, we first generate 1000 synthetic samples for each $(x, s) \in \mathbf{x} \times \mathbf{s}$ to compute the FD score for the entire dataset $\mathbf{x}$,

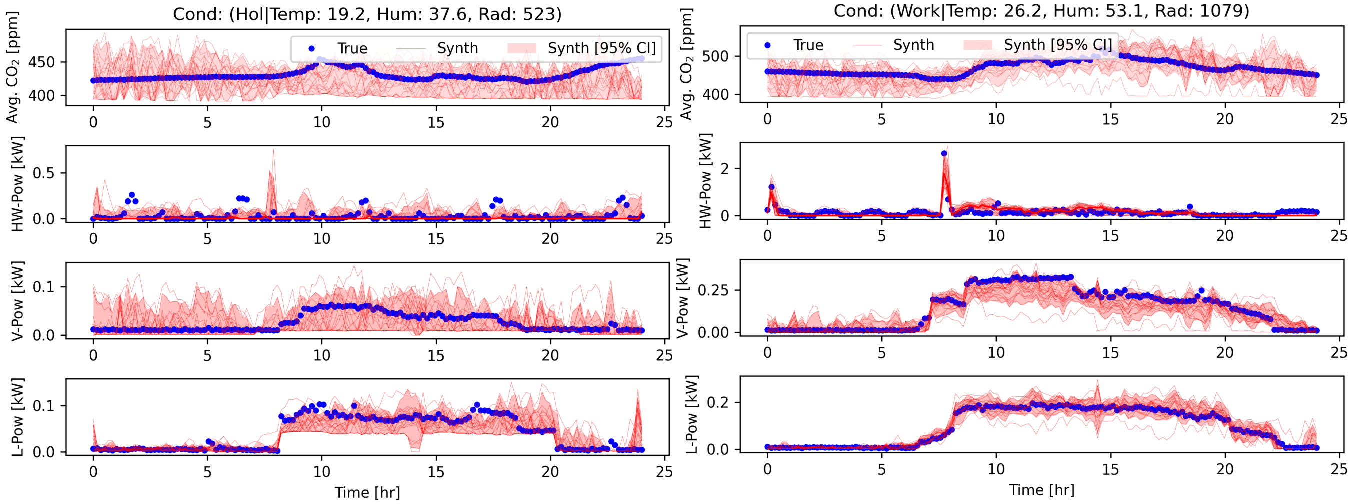

Figure 3: Measured and synthetic data with 95% confidence intervals for two sets of conditioning inputs. Time axis starts at 0: midnight. HW: hot water. V: ventilation. L: lighting. Temperature is in degree-C, humidity in %, solar radiation in W/m^2.

and then concatenate all the synthetic data samples to form the synthetic dataset $\hat{\mathbf{x}}$ and compute the FD score for the pair $\mathbf{x}$ and $\hat{\mathbf{x}}$. We then sum over every channel of x to scalarize.

We compare the performance of our proposed RAFT-VG with the performance of popular generative models like vanilla GAN, Wasserstein-GAN, and VAE-GAN [23]. Note that we do not compare against recurrent VAE or GAN architectures such as TimeGAN [49] or ID-GAN [24] as we argue that we do not need to add the computational overhead of recurrent layers for our specific problem because of the following reasons: (i) we generate 24h building load profiles at fixed sampling rate, that is, we have fixed-length sequences, so the power of variable-length sequence by recurrence is unwarranted; (ii) we do not need sample-by-sample generation; and, (iii) we do not generate long sequences and therefore, do not need to share parameters across networks.

For fairness, the network size and depth are kept identical across all paradigms, and randomized portions of the code are seeded from the same initial seed, with identical latent dimensions as well. We attempted to use identical optimizers with identical learning rates, but this led to poor performance in VAE-GAN, GAN, and W-GAN, so we changed those optimizers from Adam to RMSprop and manually tuned learning rates until performance improved. Since an objective of this paper is to demonstrate the difficulty of training GANs and architectures that involve GANs, we did not heavily customize each implementation. Instead, we attempted to allot equal amounts of manual tuning effort to each competitor; VAE and W-GAN notably required the least tuning effort.

The results of the comparison are summarized in Table 1. The RAFT procedure clearly enhances the VAE-GAN performance, with metrics reflecting both the accuracy of average fit (NRMSE) and distributional fit (FD, KLD) being an order of magnitude better compared to the best amongst its competitor algorithms. In particular, the classical GAN shows the weakest performance, and the high FD indicates mode-collapse, which we will investigate in more detail in Section 4.3.3. The VAE-GAN demonstrates slightly better performance, but despite careful selection of its hyperparameters, it is still often susceptible to mode-collapse since the discriminator does not employ a Wasserstein loss. We have observed empirically that training the encoder, decoder, and discriminator jointly often incurs the same instability during training that we have come to expect from the classical GAN. This is evident from the high FD score and higher KLD score compared to W-GAN, classic VAE, and our RAFT-VG. The VAE exhibits the next best performance, and is comparable to the W-GAN, supporting why combining the strengths of these algorithms is expected to exhibit good performance, which motivated the RAFT-VG. Wall-clock training times of all the models considered in this work were similar.

Table 1: Performance Comparison

Network	NRMSE (↓)	FD (↓)	KLD (↓)
VAE	2.77e-01	1.47e+03	2.29e+01
GAN	9.88e+00	1.01e+06	2.33e+04
W-GAN	1.97e-01	2.22e+03	2.15e+01
VAE-GAN	8.10e-01	1.19e+05	5.58e+01
RAFT-VG (ours)	**7.07e-02**	**1.83e+02**	**2.06e+00**

4.3.3 *Robustness to Mode-Collapse.*

While training the GAN and VAE-GAN, we noticed that varying the network size or latent dimensions had a strong impact on the performance of the method, with many configurations exhibiting mode-collapse. To answer whether RAFT-VG exhibits mode-collapse with varying network size, we vary the number of hidden nodes and the latent dimension of the VAE component (equivalently, the input dimension of the W-GAN component) over (64, 128, 256) and (16, 32, 64), respectively, and compute FD scores of these variants' performance with 1000 synthetic samples. We also compare these to the corresponding FD scores produced by VAE-GAN, W-GAN, and vanilla GAN with the

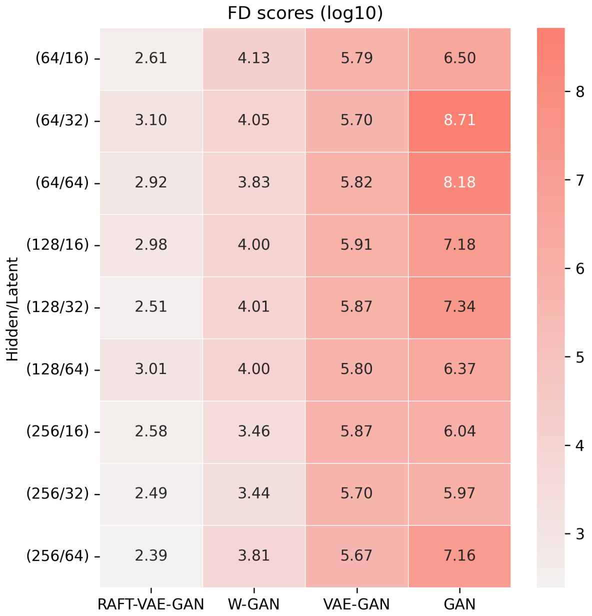

Figure 4: Robustness to mode collapse with varying hidden nodes and latent dimension. Lower values indicate better performance.

same network size alterations. Unlike in Section 4.3.2, we do not customize hyperparameters of the VAE-GAN to result in good performance; this test is conducted fairly by maintaining parity across all networks. Recall that the larger the FD score, the more likely the mode-collapse. The results of this investigation are illustrated via a heatmap in Fig. 4, where larger FD scores are darker red and are an indicator of mode-collapse. As we can see from this heatmap, the RAFT-VG is most robust to size variation, and consistently generates low FD scores. W-GAN is a close second for every tuple of hidden node size and latent dimension, and both VAE-GAN and GAN generate FD scores that are at least an order of magnitude larger, indicating the fragility of their training regimes unless a specific set of (hard to determine) hyperparameters are selected: a task which requires significant manual tuning that our RAFT-VG does not.

4.3.4 Trading-off VAE Pre-training vs. W-GAN Fine-tuning. To understand the importance of the VAE component versus the discriminator component in terms of generative modeling performance, we run a series of experiments where we vary the ratio of pre-training and fine-tuning iterations on a testing set of data that was hidden during training. With 10000 total training iterations allowed to the RAFT-VG, we allot 1000, 2500, 5000, 7500, and 9000 iterations to the VAE; see NRMSE and KLD performance in Fig. 5. We observe that there is a clear benefit to letting the VAE component train well, as can be seen by the consistently low KLD values for RAFT-50/50, RAFT-75/25, and RAFT-90/10, compared to when the VAE is trained for less than 50% of the total iterations. However, the RAFT procedure certainly helps improve performance, as can be seen by the 'VAE' bar, which is effectively RAFT-100/0: that is, entirely training the VAE with no GAN component results in significantly worse generalization performance. In fact, the best performance in terms of KLD is exhibited by RAFT-75/25, where the discriminator was only trained for 2500 iterations. Considering both NRMSE and KLD, RAFT-50/50 exhibits the best performance, demonstrating once again the effectiveness of merging VAEs and GANs with the RAFT procedure.

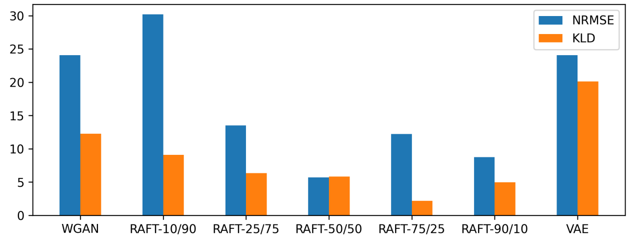

Figure 5: Effect of VAE pre-training vs. WGAN fine-tuning on NRMSE and KLD. 'WGAN': RAFT-0/100; 'VAE': RAFT-100/0. Lower values indicate better performance.

5 CONCLUSIONS

In this paper, we presented a regularized adversarial fine-tuning (RAFT) procedure to combine the advantages of two widely used generative modeling approaches: VAEs and GANs. We demonstrated, on data collected from a commercial sustainable building over 18 months, that our proposed RAFT-VG architecture provides realistic synthetic data conditioned on environmental and operational conditions, has a simple training procedure that (empirically) trains well without significant manual effort expended for selecting hyperparameters, and is more robust to mode-collapse compared against other state-of-the-art algorithms.

We showed our framework to be able to generate daily building operation profiles with a resolution of 1 sample every 10 minutes; however, if required by specific downstream applications, we could generate time series with a finer resolution after retraining the model on higher frequency sensor data. A potential limitation of our framework is that it is not based on recurrent layers. This might become an issue for real-time applications — where one often needs to generate realistic time-series profiles on a sample-by-sample basis — and for the generation of very long time series — where the lack of recurrent layers might need to be compensated by an increase in model size. Our future work will involve validating stochastic energy-optimal control policies by leveraging these synthetic sequences.

REFERENCES

[1] Martin Arjovsky and Leon Bottou. 2017. Towards Principled Methods for Training Generative Adversarial Networks. In *International Conference on Learning Representations*.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[3] Gaby Baasch, Guillaume Rousseau, and Ralph Evins. 2021. A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data. *Energy and AI* 5 (2021), 100087.

[4] Gianni Bianchini, Marco Casini, Daniele Pepe, Antonio Vicino, and Giovanni Gino Zanvettor. 2019. An integrated model predictive control approach for optimal

HVAC and energy storage operation in large-scale buildings. *Applied Energy* 240 (2019), 327–340.

[5] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. 2021. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence* (2021).

[6] Taylan Cemgil, Sumedh Ghaisas, Krishnamurthy Dj Dvijotham, and Pushmeet Kohli. 2020. Adversarially robust representations with smooth encoders. In *International Conference on Learning Representations*.

[7] Zhenghua Chen and Chaoyang Jiang. 2018. Building occupancy modeling using generative adversarial network. *Energy and Buildings* 174 (2018), 372–379.

[8] Hari Prasanna Das, Ryan Tran, Japjot Singh, Xiangyu Yue, Geoffrey Tison, Alberto Sangiovanni-Vincentelli, and Costas J Spanos. 2022. Conditional synthetic data generation for robust machine learning applications with limited pandemic data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11792–11800.

[9] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2017. Adversarial Feature Learning. In *International Conference on Learning Representations*.

[10] Wei Dong, Xianqing Chen, and Qiang Yang. 2022. Data-driven scenario generation of renewable energy production based on controllable generative adversarial networks with interpretability. *Applied Energy* 308 (2022), 118387.

[11] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. 2017. Adversarially Learned Inference. In *International Conference on Learning Representations*.

[12] Cheng Fan, Meiling Chen, Rui Tang, and Jiayuan Wang. 2022. A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions. In *Building Simulation*, Vol. 15. Tsinghua University Press, 197–211.

[13] Cheng Fan, Yongjun Sun, Yang Zhao, Mengjie Song, and Jiayuan Wang. 2019. Deep learning-based feature engineering methods for improved building energy prediction. *Applied energy* 240 (2019), 35–45.

[14] Marta Fochesato, Fazel Khayatian, Doris Fonseca Lima, and Zoltan Nagy. 2022. On the use of conditional TimeGAN to enhance the robustness of a reinforcement learning agent in the building domain. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 208–217.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[16] Yuxuan Gu, Qixin Chen, Kai Liu, Le Xie, and Chongqing Kang. 2019. GAN-based model for residential load generation considering typical consumption patterns. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 1–5.

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of Wasserstein GANs. *Advances in neural information processing systems* 30 (2017).

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[19] Yassine Himeur, Khalida Ghanem, Abdullah Alsalemi, Faycal Bensaali, and Abbes Amira. 2021. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. *Applied Energy* 287 (2021), 116601.

[20] Achin Jain, Francesco Smarra, Madhur Behl, and Rahul Mangharam. 2018. Data-driven model predictive control with regression trees—an application to building energy management. *ACM Transactions on Cyber-Physical Systems* 2, 1 (2018), 1–21.

[21] Fazel Khayatian, Zoltán Nagy, and Andrew Bollinger. 2021. Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties. *Energy and Buildings* 251 (2021), 111334.

[22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. arXiv:http://arxiv.org/abs/1312.6114v10 [stat.ML]

[23] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. 2016. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*. PMLR, 1558–1566.

[24] Wonkwang Lee, Donggyun Kim, Seunghoon Hong, and Honglak Lee. 2020. High-fidelity synthesis with disentangled representation. In *Proc. of the 16th European Conference on Computer Vision*. Springer, 157–174.

[25] Bingxu Li, Fanyong Cheng, Hui Cai, Xin Zhang, and Wenjian Cai. 2021. A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network. *Energy and Buildings* 246 (2021), 111044.

[26] Bingxu Li, Fanyong Cheng, Xin Zhang, Can Cui, and Wenjian Cai. 2021. A novel semi-supervised data-driven method for chiller fault diagnosis with unlabeled data. *Applied Energy* 285 (2021), 116459.

[27] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for sharing networked time series data: Challenges, initial promise, and open questions. In *Proceedings of the ACM Internet Measurement Conference*. 464–483.

[28] Chujie Lu, Sihui Li, and Zhengjun Lu. 2022. Building energy prediction using artificial neural networks: A literature survey. *Energy and Buildings* 262 (2022), 111718.

[29] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2016. Adversarial autoencoders. In *International Conference on Learning Representations*.

[30] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. 2017. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*. 2391–2400.

[31] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv preprint arXiv:1411.1784*. http://arxiv.org/abs/1411.1784

[32] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

[33] Zoltán Nagy, Fah Yik Yong, Mario Frei, and Arno Schlueter. 2015. Occupant centered lighting control for comfort and energy efficient building operation. *Energy and Buildings* 94 (2015), 100–108.

[34] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier GANs. In *International conference on machine learning*. PMLR, 2642–2651.

[35] Mina Razghandi, Hao Zhou, Melike Erol-Kantarci, and Damla Turgut. 2022. Variational autoencoder generative adversarial network for Synthetic Data Generation in smart home. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 4781–4786.

[36] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training GANs. *Advances in neural information processing systems* 29 (2016).

[37] Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data* 6, 1 (2019), 1–48.

[38] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).

[39] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. 2021. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2611–2619.

[40] Ying Sun, Fariborz Haghighat, and Benjamin CM Fung. 2020. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy and Buildings* 221 (2020), 110022.

[41] Chenlu Tian, Chengdong Li, Guiqing Zhang, and Yisheng Lv. 2019. Data driven parallel prediction of building energy consumption using generative adversarial nets. *Energy and Buildings* 186 (2019), 230–243.

[42] Chenguang Wang, Ensieh Sharifnia, Zhi Gao, Simon H Tindemans, and Peter Palensky. 2022. Generating multivariate load states using a conditional variational autoencoder. *Electric Power Systems Research* 213 (2022), 108603.

[43] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. 2018. Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid* 10, 3 (2018), 3125–3148.

[44] Zhe Wang and Tianzhen Hong. 2020. Generating realistic building electrical load profiles through the Generative Adversarial Network. *Energy & Buildings* 224 (2020), 110299.

[45] Danlan Wu, Kyeon Hur, and Zhifeng Xiao. 2021. A GAN-enhanced ensemble model for energy consumption forecasting in large commercial buildings. *IEEE Access* 9 (2021), 158820–158830.

[46] Yaniv Yacoby, Weiwei Pan, and Finale Doshi-Velez. 2020. Failure modes of variational autoencoders and their effects on downstream tasks. *arXiv preprint arXiv:2007.07124*.

[47] Ke Yan, Jing Huang, Wen Shen, and Zhiwei Ji. 2020. Unsupervised learning for fault detection and diagnosis of air handling units. *Energy and Buildings* 210 (2020), 109689.

[48] Yunyang Ye, Matthew Strong, Yingli Lou, Cary A Faulkner, Wangda Zuo, and Satish Upadhyaya. 2022. Evaluating performance of different generative adversarial networks for large-scale building power demand prediction. *Energy and Buildings* 269 (2022), 112247.

[49] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. 2019. Time-series Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc.

[50] Yufei Zhang, Arno Schlueter, and Christoph Waibel. 2023. SolarGAN: Synthetic annual solar irradiance time series on urban building facades via Deep Generative Networks. *Energy and AI* 12 (2023), 100223.