Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties

Fazel Khayatian [a,*], Zoltán Nagy [b], Andrew Bollinger [a]

[a] Urban Energy Systems Laboratory, Swiss Federal Laboratories for Materials Science and Technology, Empa, Dübendorf, Switzerland
[b] Intelligent Environments Laboratory, Department of Civil, Architectural and Environmental Engineering, The University of Texas at Austin, Austin, TX, USA

ARTICLE INFO

ABSTRACT

This paper describes the process of creating uncertainty-infused synthetic profiles of building performance. The synthetic profiles are utilized as a resource for evaluating the response of trained machine learning models to unseen events. Applications of the introduced method can benefit researchers and practitioners who train data-driven building models on limited historical data and is particularly useful when a physics-based model of the building is unavailable. As an original contribution, we propose a conditional deep convolutional Generative Adversarial Network (GAN) for projecting multi-dimensional time-series profiles of building performance. The proposed GAN reflects climate and operation variations into the synthetic building performance profiles, while preserving the internal consistency within the generated data. To ensure high quality synthetic profiles, this study validates the plausibility of generated data through qualitative (visualization) and quantitative (Pearson correlation, Wasserstein distance) assessments. Synthetic profiles are fed to a trained reinforcement learning model and a rule-based controller to compare their performances in the presence of uncertainty. Results show that with limited training data, a reinforcement learning model's response can be fairly sensitive to uncertainties and disturbances, insofar, some advantages over rule-based controllers may be overestimated. To ensure the reproducibility of the presented results, this study is conducted on open data and models are shared as open source.

1. Introduction

1.1. Background

The emergence of data-driven models has played an important role in improving the performance of building energy systems, be it at planning and design phase, throughout the operation stage, or during retrofit. As data collection from buildings grow in popularity, data-driven models adopt a more essential position in building energy research. Successful implementations of data-driven models are reported for a wide range of applications spanning from energy saving [1] to peak shaving [2] and display a significant impact on building performance. For instance, a recent study showed that data predictive control can save roughly 25% of cooling energy when compared to conventional rule-based controllers [3].

One challenge that impedes widespread application of data-driven models is the inadequacy of historical data for assessing a model's response to uncertainties and disturbances [4]. While most studies of building energy analytics discuss data adequacy for training a model, few raise questions about a model's response to unseen data, particularly beyond the available training, validation, and test datasets. Meanwhile, a model's robustness to shifts in a dataset can determine its practicality for real-world applications [5].

1.2. Knowledge gap

The aforementioned inadequacy of data can be potentially addressed by creating synthetic alternatives to the original dataset. There are two common approaches for producing synthetic building performance profiles: (1) infusing randomness into calibrated physics-based models [6] and (2) generating synthetic data directly from measurements by using back-box models [7]. A combination of both methods have also proven useful. For instance, by combining physics-based and data-driven techniques a recent study demonstrated a promising solution to create synthetic

* Corresponding author.
E-mail addresses: fazel.khayatian@empa.ch (F. Khayatian), nagy@utexas.edu (Z. Nagy), andrew.bollinger@empa.ch (A. Bollinger).

energy profiles for an entire city [8]. Furthermore, generating permuted building energy data has also been explored by combining data-driven models with manual random sampling [9].

Given the complexity and computational burden of generating multidimensional time-series profiles, using end-to-end black-box models for creating synthetic data has been seldom explored. However, with the introduction of implicit density modelling using Generative Stochastic Networks [10] and later Generative Adversarial Networks (GAN) [11], projecting synthetic data has been greatly facilitated. GANs have been studied for non-intrusive load monitoring [12], projecting buildings occupancy patterns [13], generating electricity load patterns [7], and creating cooling load profiles [14].

There is an opportunity to further explore the capabilities of GANs for projecting synthetic data, particularly with the aim to evaluate the robustness of models. However, using GAN projections for model assessment faces a number of challenges, which have not been addressed in the literature. Previous studies use GAN to project one-dimensional synthetic energy data. There is a need to assess GANs' potential for projecting multiple time-series profiles and evaluate their capability to preserve covariations within multidimensional outputs. Furthermore, none of the previous studies seek to control the characteristics of the GAN outputs based on a desired set of features (e.g., climate, or operation). Instead, earlier studies divide the profiles into a number of groups and train a separate GAN on each subset, which is inefficient and prone to overfitting. Finally, an in-depth discussion on the plausibility of the synthetic building energy data is still missing from the literature. This includes GANs' capabilities to project out-of-sample scenarios for expanding training or test datasets, as well as its potential to test the robustness of trained machine learning models.

1.3. Original contributions

With the aim to address the abovementioned challenges, this paper introduces a new end-to-end data generation method for projecting synthetic building performance profiles. The proposed method is particularly useful for data-driven building energy research with limited access to data for rigorous testing. The original contributions of this study is as follows:

1. **A GAN that generates multi-dimensional synthetic time-series profiles with constraints over climate and operation characteristics.** Contrary to the existing methods that project one-dimensional profiles and have no control over the characteristics of synthetic generations, we propose a conditional GAN that creates multidimensional profiles with association to desired occupancy and weather conditions.
2. **Proposing qualitative and quantitative methods to evaluate the plausibility of the synthetic data.** Previous studies sufficed to comparing the statistical characteristics of the original data with synthetic projections. The current study uses statistical and element wise metrics for evaluating the capability of the proposed GAN in preserving internal and external covariations.
3. **Demonstrating the potential of synthetic profiles for evaluating models' response to unseen events**. This study is the first to use GAN-produced building energy profiles as out-of-sample test data. Through synthetic scenarios, we evaluate the robustness of a Reinforcement Learning (RL) agent to potential uncertainties and disturbances.

1.4. Organization of the paper

The remainder of the paper is organized as follows. Section 2 provides a general description of GANs, conditioning generative models, and the application of convolution layers. Section 3 introduces the case study, describes details of the proposed GAN, and explains the techniques that are adopted for improving stabilization and convergence during training. Section 4 provides qualitative evaluations of the synthetic data, followed by quantitative assessments of the GAN projections. This section then provides an example of how synthetic projections can be utilized to evaluate the performance of a trained RL agent. Section 5 opens a discussion on the strengths and weaknesses of the proposed GAN as well as the evaluation methods and metrics. Section 6 draws the conclusions and maps possible future paths of synthetic data generation for evaluating machine learning models.

2. Method

2.1. GAN

Creating synthetic building performance profiles with physics-based models requires setting up a building energy model followed by a calibration process, which makes the procedure cumbersome and difficult for non-experts. An alternative is to generate synthetic data directly from observations without the need for any physics-based modelling. Recent advances in AI enables learning of patterns directly from a dataset and generating samples that are similar - but not identical - to the original data.

The most successful examples of AI-generated synthetic data are demonstrated by GANs. For instance, fake images that are indistinguishable from real photos [15]. GANs are often composed of two multilayer perceptrons (neural networks), i.e., a generator and a discriminator [16]. The objective of the generator is to create synthetic samples that resemble the actual data. The objective of the discriminator is to reject unrealistic samples that are created by the generator. To enable a variety of outputs from the generator, it is vital to incorporate randomness into the model. This is foreseen through a latent noise placeholder for the generator (Fig. 1).

The process of training a conventional GAN is composed of two main steps: (1) training the discriminator, and (2) training the generator (Fig. 2). During the first step, the discriminator is trained on a set of real samples with their corresponding targets set to "True" (i.e. 0), and a set of synthetic samples with their corresponding targets set to "False" (i.e. 1). The first step helps the discriminator to learn manifolds that distinguish real samples from synthetic samples. During the second step, the discriminator is trained on synthetic samples that are produced by the generator, and assigned with a target label of "True" (i.e., 0). In this step, the discriminator's layers are frozen. Namely, the weights and biases of the discriminator will not update during training. Instead, the gradient of the error is backpropagated [17] from the discriminator to the generator, and therefore, updates the generator's weights and biases. These two training steps are repeated sequentially until the error of the discriminator and the error of the generator converge. Once the training is finished, the generator encodes the synthetic samples.

GANs are particularly suitable for creating synthetic data because the generator does not average the outputs, but rather selects one or another. This prevents the generator to project smoothed (averaged) outputs. Furthermore, the only way that a generator can improve the quality of its synthetic creations is through instructions from the discriminator. These instructions (i.e., the gradient of the error) are conveyed through the backpropagation of information from the discriminator to the generator. Backpropagating the error highlights which parts of a synthetic projection are inconsistent with the characteristics of real samples.
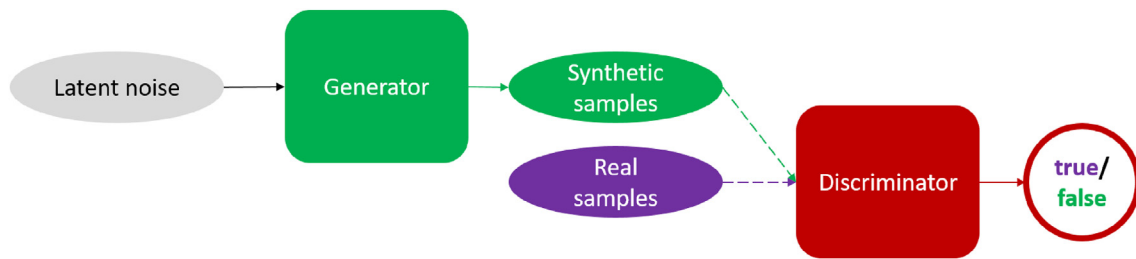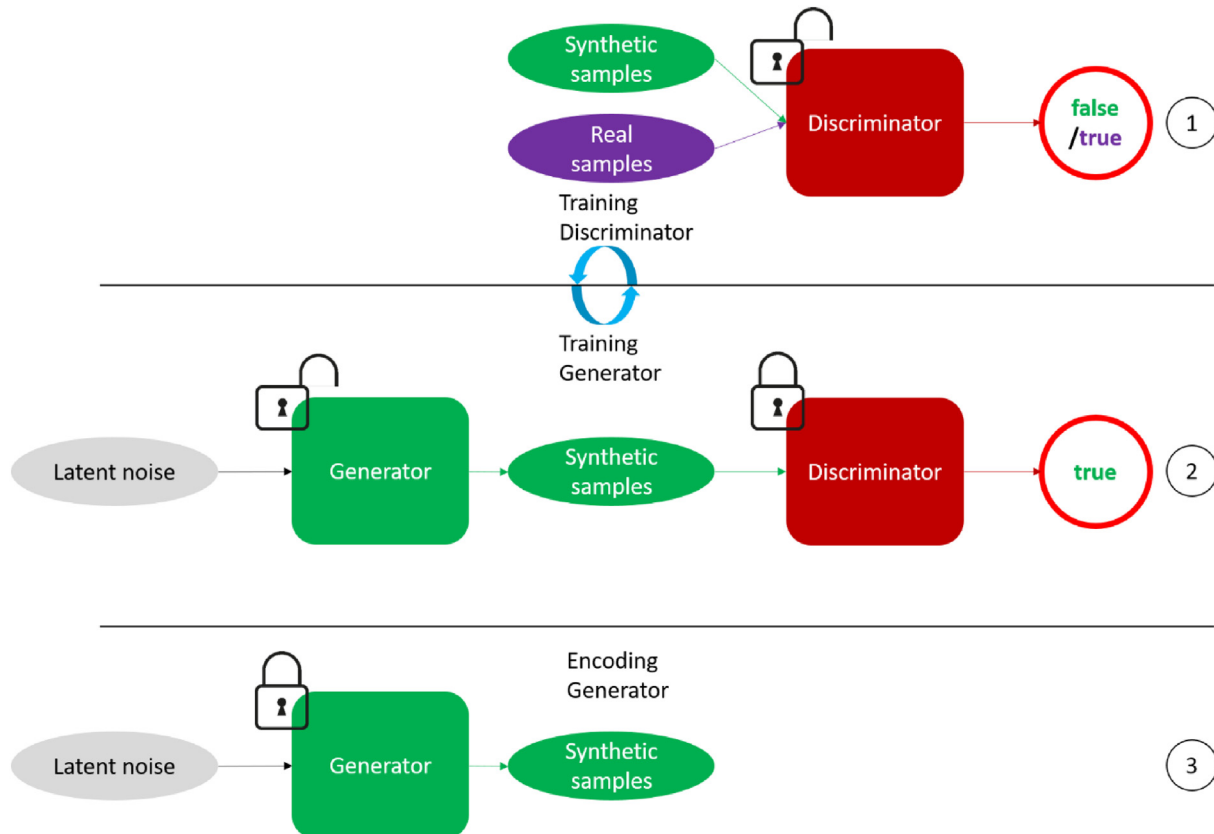
Fig. 1. The architecture of a GAN.

Fig. 2. The process of training a GAN and encoding synthetic data: (1) training the discriminator, (2) training the generator, (3) encoding synthetic data from the trained generator. Steps (1) and (2) are iteratively repeated until the model converges. An open padlock means the weights and biases of the model are updated during training/encoding.

2.2. Conditional GAN

In this study we propose a conditional GAN as initially introduced by Mirza and Osindero [18]. Conditional GANs are a subcategory of generative models, in which the generator and the discriminator are both conditioned (labelled) based on a set of exogenous features. This capability of conditional GANs makes them particularly suitable for controlling the characteristics of synthetic outputs [19]. Hence, an important application of conditional GANs is to address shortcomings in the training dataset [20]. Fig. 3 shows the scheme of a GAN that is conditioned on exogenous features. Similar to the original notion of GANs, the proposed model is composed of a discriminator and a generator, both of which are multilayer perceptrons. The process of training a conditional GAN is also similar to the description in Section 2.1.The GAN proposed in this study is conditioned based on both climate and operation characteristics, as previous studies have proven their importance in training neural networks [21]. Imposing such constraints allows

us to control the synthetic outputs of the generator, based on the outdoor climate and building's operation.

(1) Conditioning the model on climate features will ensure that the synthetic profiles co-vary with the weather condition. This conditioning is necessary when generating data for a district as the climate is identical for all member buildings.
(2) Conditioning the model on operation features will guarantee that the synthetic profiles also co-vary with the occupancy pattern. Imposing this constraint enables us to combine different occupancy scenarios with various climate conditions.

2.3. Deep convolutional GAN

GANs are mostly composed of two perceptrons, i.e. discriminator and generator. Each perceptron can be either a shallow (vanilla) neural network, or a deep network composed of convolutional layers [22]. Convolutional layers are preprocessing kernels that apply
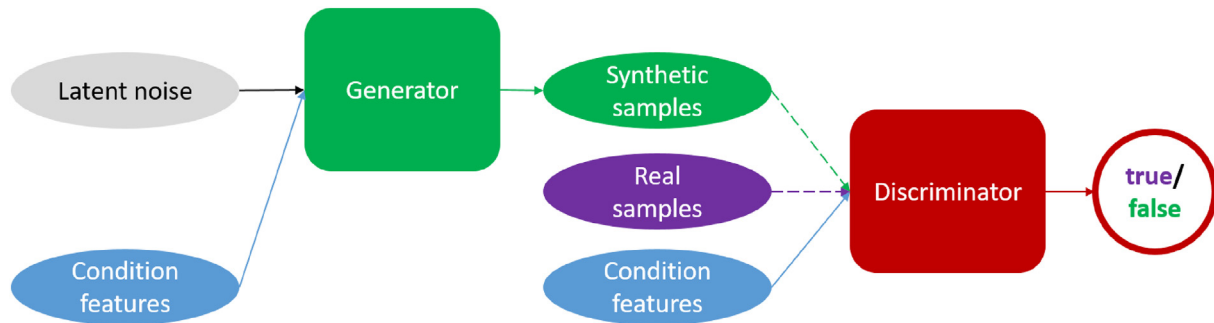
Fig. 3. The architecture of a conditional GAN.

filters to multi-dimensional data (e.g. images), with the aim of extracting important features from the inputs [23]. The filters often scan 2D arrays or a 3D tensors by sliding from one corner to the opposite. This scanning process can be repeated for maximizing the compression of features with minimal information loss (i.e. deep convolutional network). Previous studies have shown that GANs with sequences of convolution layers (i.e. deep convolutional GANs) noticeably outperform shallow variants [24].

2.4. Conditional deep convolutional GAN

The model proposed in this study follows the same procedures of sections 2.1, 2.2 and 2.3, forming a conditional deep convolutional GAN [25]. In this setup, both generator and discriminator are convolutional neural networks, which are conditioned on identical exogenous features. In the generator, concatenated inputs and conditions are first passed through dense layers and then fed to convolution layers. In the discriminator, inputs are first fed to convolutional layers, then reshaped into a 1-dimensional vector (also known as flattening), and after concatenation with condition features are forwarded to the dense layers. Further details of the conditional deep convolutional GAN adopted in this study (e.g., layers, filters, hidden units, etc.) as well as the training process are provided in the case study section.

3. Case study

3.1. Data and modelling environment

Given the importance of reproducibility – particularly for data driven and machine learning models – this study resorts to open data and codes for model development and evaluation. We demonstrate the potential of GANs for generating synthetic data by using the CityLearn OpenAI Gym environment [26]. CityLearn is a python-based open-source environment for training RL models, and benchmarking performance against Rule-Based Controllers (RBC) [27]. The open-source characteristics of CityLearn facilitates reproducing the results of this study [28]. Furthermore, all scripts and codes that are necessary for setting up and training the GAN are hosted at an open GitHub repository and shared though a permanent link [29] for future benchmarking.

The CityLearn environment hosts a virtual district of nine buildings with dissimilar occupancy, geometry, and construction characteristics. The objective is to control the Domestic Hot Water (DHW) and chilled water storages to reduce net electricity consumption. The model already contains pre-computed values of building demand profiles, as well as indoor air temperature and relative humidity. In this study we focus on Climate Zone 1 of the CityLearn datasets which corresponds to a hot – humid weather (ASHARE climate zone 2A).

3.2. Model architecture

Since the objective of this study is evaluate the robustness of RL agents against unseen data, we aim to distort the RL inputs and observe the model's response. Specifically, we focus on perturbing building energy performance profiles, as they play a major role in the RL's performance. The CityLearn environment requires six building performance parameters to train an RL agent (Table 1, Feature category: Building performance). These six parameters form the target of the generator and the input of the discriminator in our GAN setup. As mentioned before, conditioning GANs on a set of constrains can allow us to control the characteristics of projections. Given the availability of information, we assume that building performance values are a function of variation in the climate and building's operation. Climate characteristics (Table 1, Feature category: Climate constraints) can affect a variety of building performance features i.e., indoor air temperature, unmet set-point difference, indoor relative humidity, and cooling load. Operation features (Table 1, Feature category: Operation constraints) can

Table 1
Description of GAN input features, divided by category.

Feature category	GAN Features	Size and unit (type)
Building	performance 24 values per day in [C]	Indoor air temperature
	Unmet set-point difference	24 values per day in [C]
	Indoor relative humidity	24 values per day in [%]
	Equipment electric power	24 values per day in [kWh]
	DHW heating load	24 values per day in [kWh]
Cooling load	24 values per day in [kWh]	
Operation	constraints 1 categorical label per day type: [Weekday, Saturday, Sunday/ holiday]	Day Type
Daylight saving status	1 categorical label per day: [on, off]	
Climate	constraints 24 values per day in [C]	Outdoor Dry-bulb Temperature
	Outdoor Relative Humidity	24 values per day in [%]
	Diffuse Solar Radiation	24 values per day in [W/m2]
Direct Solar Radiation	24 values per day in [W/m2]	

affect all of the six building performance features. Therefore, we condition the proposed GAN on climate and operation constraints to control the shape and magnitude of synthetic building performance profiles. We did not condition the GAN on building's characteristics (e.g. thermal conductivity), as these values are assumed to be fixed or have very little variation throughout the year. Indeed, some variables such as the efficiency of the systems and the degradation of materials' reflectance and conductance over multiple years can affect long-term building performance profiles. However, these information are seldom readily available, and therefore, are excluded from GAN conditions. Given that the RL requires hourly inputs of building performance for training the agent, the GAN is set to project synthetic outputs at hourly temporal resolution. On the other hand, operation constraints are assigned at daily intervals, which limits hourly GAN projections a length of 24.

Given that the proposed model should generate 24-hour time-series profiles of building performance, the annual training dataset is reshaped into daily arrays. Each array is a vector of six building performance variables consisting of 24 values. Similarly, the annual climate data is reshaped into daily profiles, in which temperature, relative humidity, as well as direct and diffuse solar radiation each have 24 values. Once converted into daily profiles, the annual dataset of 365 days is divided into 252 weekdays, 51 Saturdays and 62 Sundays/holidays. Such disproportionate division of labels can cause an imbalance in the distribution of the input data and potentially bias the GAN toward favoring weekday profiles. To overcome this issue, samples with Saturday and Sunday/holiday labels are repeated in the dataset until all three categories (Weekday, Saturday, and Sunday/holiday) have equal shares in the input dataset. The post-processed datasets that are reshaped into 24-hour time-series profiles are available from a permanent link to an open repository [30].

As mentioned in section 2.4, the GAN proposed in this study is composed of a generator and a discriminator, both of which are deep convolutional neural networks. The architecture of the conditional GAN is borrowed from [31] and then iteratively optimized. In the following, the generator and discriminator of the optimized GAN are individually discussed and the training and tuning process is explained in detail.

3.2.1. Discriminator

The discriminator is a deep convolutional neural network (Fig. 4). The inputs, targets and the network's architecture is provided in Table 2.

- The building performance feature is a 24 by 6 array, in which columns represent the hours of the day. Array rows correspond to the building performance features as described in Table 1.
- The climate constraint feature is a one-dimensional array and composed of four climate parameters as described in Table 1. Each climate parameter has one value for every hour of the day, forming an array with size of 24. We concatenate all four climate parameters into a single array with a shape of 96.
- The operation constrains are defined based on the type of day, and therefore, the labels are assigned to days rather than hours. Since the operation features are discrete categorical labels, they are one-hot-encoded onto a logic array. The type of day feature has three categories, and the daylight saving status has two modes (Table 1). Therefore, the operation constraints form a single array with a size of five.

3.2.2. Generator

The Generator is also a deep convolutional neural network. Inputs are fed into the generator in two separate steps. First, operation and climate constrains are concatenated with the latent noise. After a dense layer of activations, the array is reshaped to form a tensor. A schematic representation of the generator's architecture is provided in Fig. 5.

3.3. Training and encoding

Inputs are scaled between -1 and 1, while the targets are set to 0 ("True") and 1 ("False"). Both the generator and discriminator are trained using the Adam optimizer [32] with a learning rate of 0.0002, a momentum decay rate of 0.5, and binary cross-entropy as the loss [33]. The model is trained for 5′000 epochs, where each epoch is a full pass through all samples. During each pass, the discriminator and the generator are updated twice on different combinations of data.
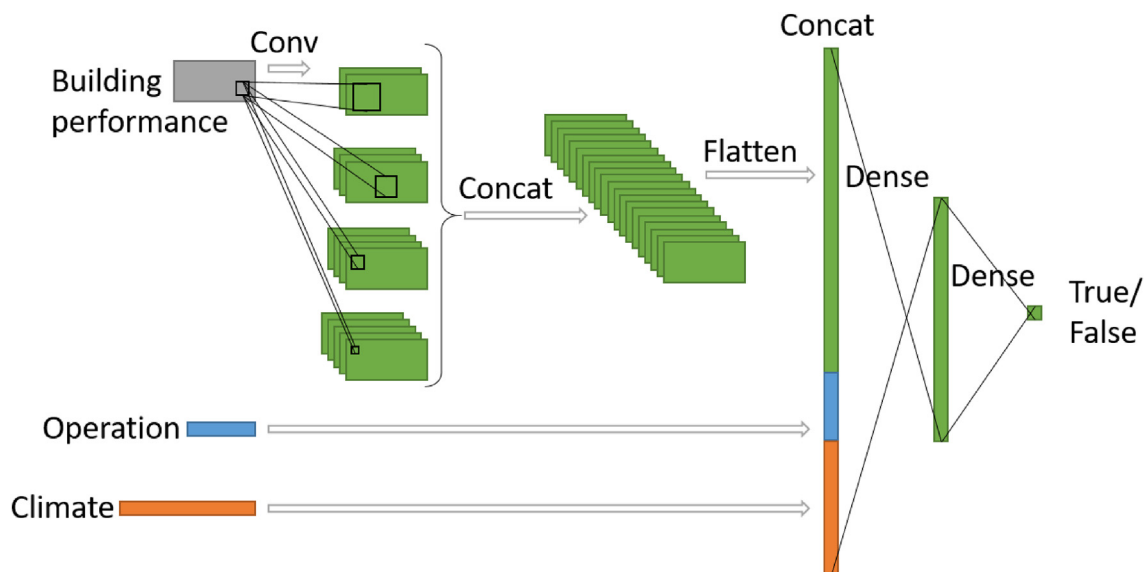

Fig. 4. Scheme of the proposed discriminator, conditioned on operation and climate features (for more details see Table 2).

Table 2
Detailed description of the proposed discriminator (for a schematic representation see Fig. 4).

Layer type (name)	Input size (from layer)	Function (details)	Output size
Input Layer (input1)	–	Input (building performance)	[24,6,1]
Input Layer (input2)	–	Input (operation constraints)	[5]
Input Layer (input3)	–	Input (weather constraints)	[96]
Convolution 2D (Conv1)	[24,6,1] (input1)	ELU, depth = 4, filter = [48,12], stride=[1,1]	[24,6,4]
Convolution 2D (Conv2)	[24,6,1] (input1)	ELU, depth = 8, filter=[12,6], stride=[1,1]	[24,6,8]
Convolution 2D (Conv3)	[24,6,1] (input1)	ELU, depth = 12, filter=[6,3], stride=[1,1]	[24,6,12]
Convolution 2D (Conv4)	[24,6,1] (input1)	ELU, depth = 16, filter=[3,3], stride=[1,1]	[24,6,16]
Concatenate (Cat1)	[24,6,4] (Conv1) [24,6,2] (Conv2) [24,6,12] (Conv3) [24,6,16] (Conv4)	Concatenate (3rd axis)	[24,6,40]
Flatten	[24,6,40] (Cat1)	Flatten	[5760]
Concatenate (Cat2)	[5760] (Flatten) [5] (input2) [96] (input3)	Concatenate (1st axis)	[5861]
Fully Connected (Dense1)	[5861] (Cat2)	ELU, n = 1024, dropout = 0.4	[1024]
Fully Connected (Dense2)	[1024] (Dense1)	Sigmoid, n = 1,	[1]

Table 3
Detailed description of the proposed generator (for a schematic representation see Fig. 5).

Layer type (name)	Input size (from layer)	Function (details)	Output size
Input Layer (input1)	–	Input (operation constraints)	[5]
Input Layer (input2)	–	Input (weather constraints)	[96]
Input Layer (input3)	–	Input (latent Gaussian noise)	[101]
Concatenate (Cat1)	[5] (input1) [96] (input2) [101] (input3)	Concatenate (1st axis)	[202]
Fully Connected (Dense1)	[202] (Cat1)	ELU, n = 2304, dropout = 0.4	[2304]
Reshape (Prep1)	[2304] (Dense1)	Reshape (24,6,16)	[24,6,16]
Convolution 2D (Conv1)	[24,6,16] (Prep1)	ELU, depth = 16, filter = [48,12], stride=[1,1]	[24,6,16]
Convolution 2D (Conv2)	[24,6,16] (Conv1)	ELU, depth = 8, filter = [48,12], stride=[1,1]	[24,6,8]
Convolution 2D (Conv3)	[24,6,8] (Conv2)	ELU, depth = 4, filter = [48,12], stride=[1,1]	[24,6,4]
Convolution 2D (Conv4)	[24,6,4] (Conv3)	tanh, depth = 1, filter=[1,1], stride=[1,1]	[24,6,1]

Following the recommendations in the literature [24], we refrain from modifying the properties of the Adam optimizer and instead focus on tuning the hyper-parameters of the GAN. Characteristics of the convolution layers (e.g., filter size, channel depth, stride) and the properties of the dense layers (e.g., number of hidden units) are fine-tuned through a grid search. In the proposed setup, we noticed that under-parametrizing the discriminator with fewer channels or smaller filters forces the model to converge after a few hundred epochs, namely, before the generator learns to produce satisfactory results. On the other hand, over-parameterizing the discriminator heavily penalizes the generator and obstructs the convergence of the GAN, even after 10′000 epochs of training. In the proposed setup, using dissimilar architectures for the generator and the discriminator improves perturbations in the synthetic data. Therefore, the generator is devised with a deep architecture (i.e. five consecutive convolution layers) and the discriminator with a wide architecture (i.e., four parallel convolution layers). The discriminator shows high sensitivity to the size and depth of the convolution filters and channels, insofar that small modifications impedes convergence. On the contrary, the performance of the generator shows high sensitivity to the number of hidden units in the dense layer.

To improve the chances of convergence, provide better stability, and encourage variability in synthetic projections, the following techniques are adopted:

- *Noisy labels*: A small noise is added to the discriminator's target labels. The noise is deployed by flipping five percent of the target labels from true to false, and vice versa. This percentage decays with each epoch and reaches zero before the training is finalized. Since we want the discriminator and the generator to incrementally improve their performances together, flipping the labels will prevent the discriminator to confidently reject the generator's projections early in the training process [34].
- *Soft labels*: The true labels are uniformly scattered between 0.9 and 1.0. Similarly, false labels are uniformly distributed between 0 and 0.1. This strategy prevents the discriminator from assigning excessive weight to a small set of features [35].
- *Noisy samples*: Building performance profiles (i.e., energy consumption and indoor air quality), are associated with 5% of uniformly distributed noise. Spreading the samples between 0.95 and 1.0 adds a small tolerance to the discriminator's acceptance of viable samples and improves the chances of convergence [36].
- *Experience replay*: A random sample (also called experience) is saved from the mini-batch in every epoch of training. Once the number of experiences match the mini-batch size, the discriminator is trained on a mini-batch of experiences. This practice prevents the generator from projecting the same output from different samples of the latent space (i.e. mode collapse) [35].
- *Felix culpa*: To boost variability in GAN's outputs, we add an extra step to the training process, in which (1) the generator and discriminator are fed with dissimilar occupancy constrains, and (2) generator's projections are neither rewarded, nor penalized. The objective of this step is to encourage the model to explore plausible combinations of climate and operation, particularly ones that are not included in the training dataset. When using dissimilar occupancy constraints, the discriminator conveys inconsistant feedbacks to the discriminator. As a result, the generator occasionally swaps the occupancy profiles of weekday, Saturday, and Sunday (holiday). On the other hand, using different occupancy constrains makes the training process unstable. To prevent divergence and mode collapse, we refrain from using the conventional reward strategy (i.e. "True" or "False") at this particular training step. Conventional reward would require the discriminator to either fully embrace the unseen outputs (i.e. "True") or completely reject them (i.e., "False"). However, each of these responses for unseen data would adversely affect the generator. If rewarded with "True", the generator will converge too quickly and never learn the covariation between climate features and building performance variables. If penalized with "False", the generator never learns to explore out-of-sample combinations. Therefore, we set dis-
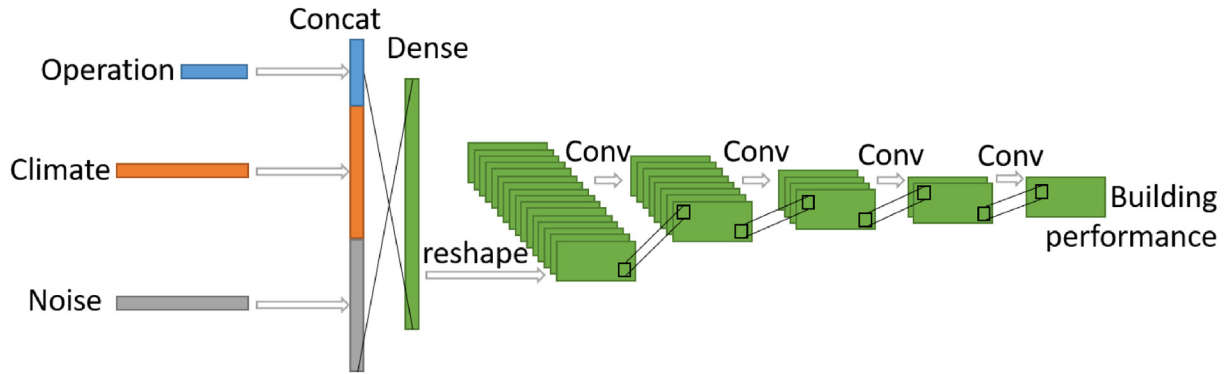
Fig. 5. Scheme of the proposed generator, conditioned on operation and climate features (for more details see Table 3).

criminator's target to indifference, i.e. neither fully "True", nor completely "False", (Gaussian noise with $\mu = 0.5$ and $\sigma = 0.1$). Our conditioning and rewarding strategy here dubbed *'felix culpa*' allows the generator to gain some degree of confidence over its perturbed projections, which enables exploring plausible scenarios beyond the training domain.

The process of training the GAN consists of four steps:

1) The discriminator is fed with real building performance samples, the corresponding climate data, and operation labels. The target of discriminator is set to 0 ("True") with a small Gaussian noise. During training, the weights and biases of the discriminator are updated.
2) The generator is fed with random noise, climate data, and random operation conditions to project synthetic building performance samples. The synthetic building performance samples, climate data, and a random set of operation conditions are fed to the discriminator. The target of the discriminator is set to 0.5 ("indifference") with a small Gaussian noise. During training, the weights and biases of discriminator are frozen and do not update. However, the weights and biases of generator are updated through the backpropagation of the gradient of the error from the discriminator to the generator.
3) The generator is fed with random noise, climate data, and operation conditions. The generator then projects synthetic building performance samples. Synthetic building performance samples, climate data, and operation conditions are fed to the discriminator. The target of discriminator is set to 1 ("False") with a small Gaussian noise. During training, the weights and biases of the discriminator are updated.
4) The generator is fed with random noise, climate data, and random operation conditions and projects synthetic building performance samples. The synthetic building performance samples, climate data, and a random set of operation conditions are fed to the discriminator. The target of discriminator is set to 0 ("True") with a small Gaussian noise. During training, the weights and biases of the discriminator are frozen and do not update. However, the weights and biases of the generator are updated through the backpropagation of the gradient of the error from the discriminator to the generator.

The model is separately trained on each building's data of the CityLearn environment, resulting in nine different GANs. The architecture of the GAN and the hyperparameters of training are identical for all buildings. The training often converges after ca. 2000 epochs, an example of which is shown in Fig. 6. Rarely, the training process may need re-initialization, particularly if the losses of the
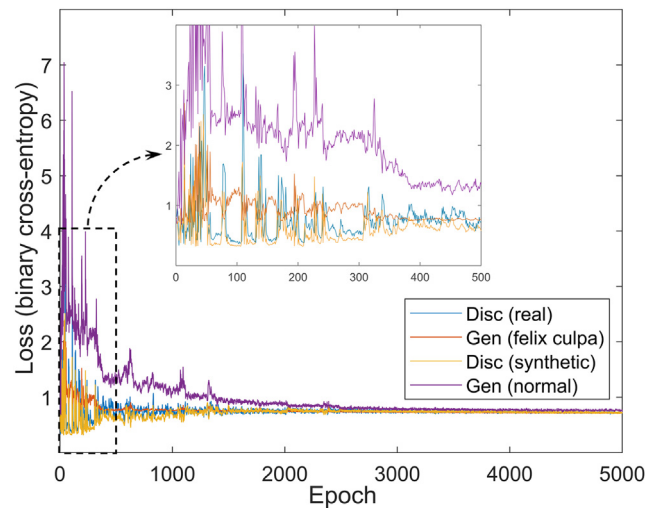

Fig. 6. The Progress of training the GAN. Each epoch consists of four steps: (1) training the discriminator on real samples, (2) training the generator using felix culpa, (3) training the discriminator on synthetic samples, and (4) training the generator using conventional rewards.

generator and the discriminator do not converge. Training is executed on an NVIDIA Quadro RTX 6000 with 24 GB of GDDR6 memory, which lasts ca. 1500 seconds. Once the training is finished, GAN models are fed with climate conditions, occupancy conditions, as well as latent noise for every day of the year. During this process, the GAN encodes 365 samples, each consisting of 24-hour profiles. With this procedure, the GAN reconstructs the profiles of an entire year (i.e., 8760 hours) for every building. The encoding process is repeated to create 50 synthetic annual profiles (i.e., building performance scenarios) for each building.

4. Results

By encoding the trained generator, every building is associated with a set of 50 synthetic performance scenarios. Given the nature of GANs, all synthetic scenarios are assumed to be equally plausible. For brevity, we will only discuss the GAN projections for one building of the dataset (i.e., Building 1). Other buildings display similar performances and may be studied further in detail by exploring the open-source model [29] and data [30] that are shared through permanent links. In this section, we analyze the suitability of GAN's projection though both quantitative and qualitative assessments.
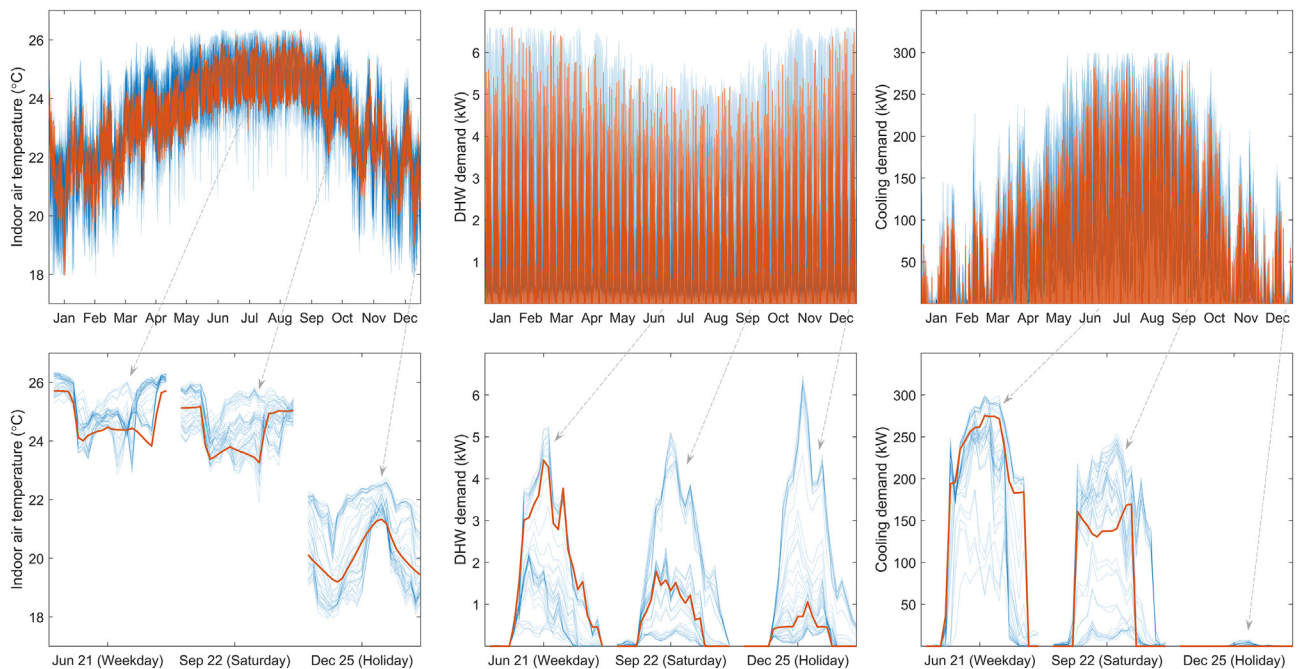
Fig. 7. Contrasting synthetic projections against the original data.

4.1. Qualitative assessment

To evaluate the quality of the synthetic data, we visually compare GAN's projections with the original dataset [37]. Fig. 7 contrasts annual and daily profiles of synthetic projections, against those of the original training data. The figure only compares three variables for brevity, focusing on parameters which are affected by both climate and operation while being easy to interpret. We observe that the synthetic data projected by GAN follow the annual trend of variation for all three variables (Fig. 7 – top row). A deeper dive into daily snapshots of the profiles (Fig. 7 – bottom row), show that the synthetic data also suitably capture hourly variations within a day.

Fig. 7 shows that the shape and magnitude of some synthetic profiles greatly differ from those of the original data. This observation is directly related to the *felix culpa* reward strategy as described in section 3.3. Given that the trained GAN associates uncertainty with the operation constrains (Table 1- Day type), each synthetic profile can be randomly conditioned based on any of the three types of day, i.e. weekday, Saturday, or Sunday/holiday. *Felix culpa* enables the GAN to hallucinate low-probable conditions, in which the actual operation of the building does not follow the presumed type of day. A good example is when an office is open during a weekend to host a special event, or closed during a weekday due to an emergency. The proposed *felix culpa* effect can be simply switched off as shown in Fig. 8. However, high reliance on the type of day can limit the range of synthetic scenarios, and therefore is not recommended. The 2020 pandemic has shown that such unexpected changes in buildings' occcupancy can influence building operation in unforeseen ways.

4.2. Quantitative assessment

For quantitative assessment, we evaluate two sets of characteristics in the GAN's outputs: (1) the synthetic projections' alignment with climate constraints and (2) the internal consistency within synthetic projections.
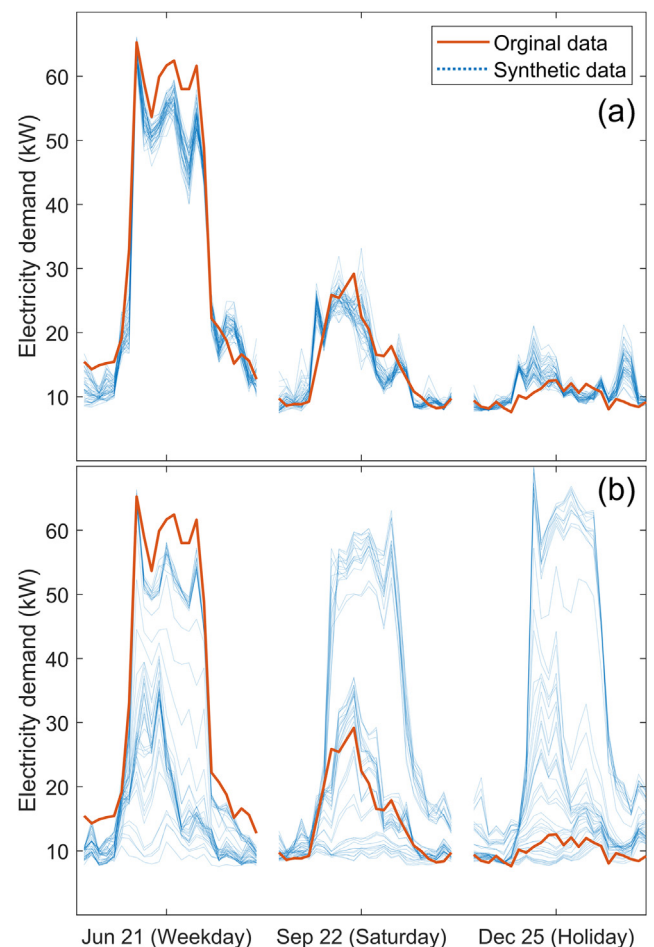

Fig. 8. Comparing synthetic profiles with felix culpa disabled (a) and enabled (b) during training.

- The first assessment reveals to what extent the GAN preserves covariations between climate constraints and synthetic projections. This is to verify that changes in the weather are properly reflected into the synthetic data. For instance, if the original data shows high covariation between cooling loads and direct solar radiation, the same patterns of covariation should be observed in the synthetic data.
- The second assessment shows how well the GAN has learned co-occurrences of different variables within the synthetic profiles. This helps us to understand if synthetic projections are plausible scenarios. For instance, electricity demand and DHW demand are both heavily driven by the type of day, and therefore, any covariation between these two features in the original training data should be also preserved in the synthetic projections.

We use two metrics to quantify the similarity between profiles, i.e., (1) the Pearson correlation coefficient [38] for assessing linear covariation between two variables and (2) the Wasserstein distance [39] for evaluating the distance between the distributions of two variables. For brevity, we discuss the quantitative assessment for three pairs of variables. However, the quality of synthetic projections is consistent for all variables.

4.2.1. External covariations

In this section we demonstrate GANs capability to preserve correlations between climate constraints and synthetic projections. Table 4 shows the Pearson correlation coefficient between the climate characteristics and the building performance within the original training data. In this table, values closer to 1 correspond to greater positive correlation, and values closer to −1 indicate higher negative correlation. Values close to zero indicate very small or no correlation between variables. The p-values for all variable-pairs are significantly smaller than 0.05, hinting that the null hypothesis is extremely unlikely for the reported correlation values.

Fig. 9 compares the Pearson correlation coefficients of the original dataset (Table 4), with those of the synthetic projections. We report the Pearson values for three pairs of variables, which are specifically chosen to cover a range of correlation from positive to negative:

- Outdoor air temperature displays high positive correlation with indoor air temperature (Fig. 9, T_out vs T_in),
- Direct solar radiation does not show a significant correlation with indoor relative humidity (Fig. 9, R_dir vs RH_in),
- Outdoor relative humidity has a negative correlation with cooling demand (Fig. 9, RH_out vs Cool).

Fig. 9 shows that the GAN is able to capture linear correlations between climate and building performance features and properly reflect them in its synthetic projections.

We also evaluate the similarity between the probability distributions of climate and building performance variables. This assess-
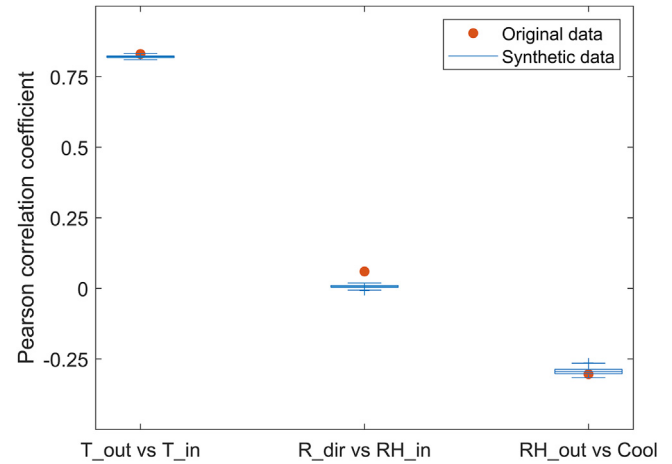

Fig. 9. Contrasting synthetic projections against the original training data using the Pearson correlation coefficient.

ment would provide a statistical overview of the synthetic projections, and specify how well the GAN captures the frequency of occurrence of different events. We use the Wasserstein distance to evaluate the minimum cost of converting one distribution to the other. Smaller Wasserstein values correspond to greater similarity between the probability distributions, while a value of zero indicates that the two distribution are identical. For consistency, we use the same variables of Fig. 9 for evaluating the Wasserstein distance. Interestingly, the Wasserstein metric does not reveal statistical similarities between outdoor relative humidity and cooling demands (Fig. 10, R_out vs Cool), although they showed a negative linear correlation in Fig. 9. The same applies to the similarity between direct solar radiation and indoor relative humidity (Fig. 10, R_dir vs RH_in). However, the GAN properly captures the strong statistical similarity between outdoor and indoor air temperatures (Fig. 10, T_out vs T_in).

4.2.2. Internal consistency

Given that GAN's output is a set of six time-series profiles, any covariations among synthetic projections should be carefully studied. Therefore, we first quantify correlations between building performance variables to understand the strength and direction of linear relationship within the training dataset (Table 5). Afterwards we assess whether the proposed GAN preserves the correlations across its synthetic multi-dimensional projections.

Given that there are no significant negative correlations within building performance variables, we use three pairs of variables that cover the range from strong positive to negligible correlation:

- DHW demand shows a strong positive correlation with electricity demand (Fig. 11, DHW vs Elec),
- Indoor air temperature displays small correlation with cooling demand (Fig. 11, T_in vs Cool),

Table 4
Pearson correlation between climate and building performance variables.

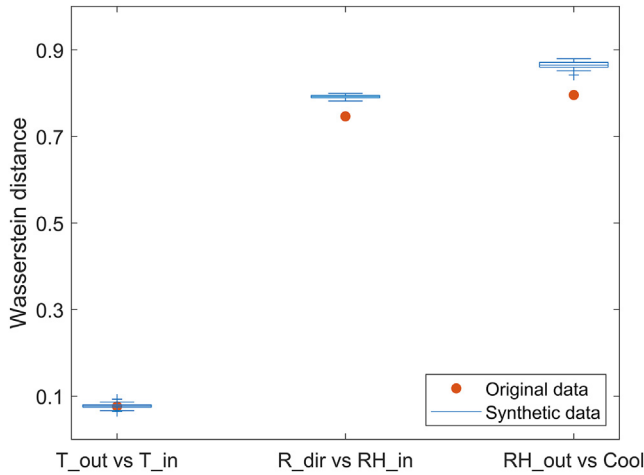|		Climate variables			
		Outdoor temperature	Outdoor relative humidity	Diffuse solar radiation	Direct solar radiation
Building performance variables	Indoor temperature	0.83	−0.09	0.15	0.09
	Unmet set–point	0.27	−0.18	0.15	0.27
	Indoor relative humidity	0.50	0.41	0.06	−0.1
	Electricity demand	0.18	−0.33	0.63	0.44
	DHW demand	0.16	−0.43	0.65	0.45
	Cooling demand	0.66	−0.30	0.59	0.39

Fig. 10. Contrasting the synthetic projections against the original training data using the Wasserstein metric.
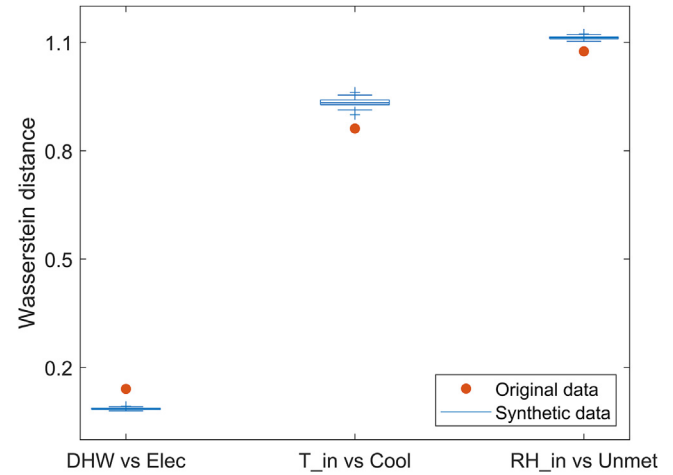

Fig. 12. Contrasting synthetic projections against the original training data using the Wasserstein metric.

Table 5
Pearson correlation coefficient among building performance variables.

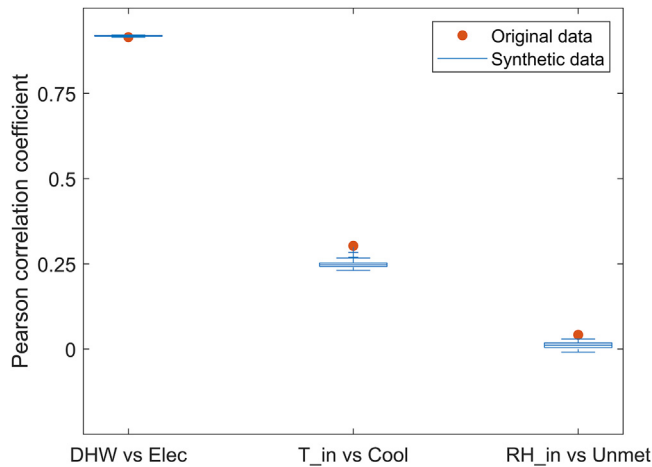|		Building performance variables				
		Indoor temperature	Unmet set-point	Indoor relative humidity	Electricity demand	DHW demand
Building performance variables	Unmet set–point	0.19				
	Indoor relative humidity	0.49	0.04			
	Electricity demand	0.03	0.1	−0.02		
	DHW demand	−0.02	0.09	−0.08	0.91	
	Cooling demand	0.3	0.25	0.25	0.56	0.53

Fig. 11. Contrasting synthetic projections against the original training data using the Pearson correlation coefficient.

- The correlation between indoor relative humidity and the average unmet cooling set-point difference is negligible (Fig. 11, RH_in vs Unmet).

The GAN suitably preserves the internal consistency among building performance variables. However, slight underestimations of weak correlations is also observed.

Aside from linear correlation assessments, we measure the similarity of probability distributions among building performance variables (Fig. 12). Once again, the GAN suitably preserves the internal consistency within multi-dimensional outputs, yet slightly overestimates strong similarities between variables (Fig. 12, DHW vs Elec). On the other hand, the large difference between distribu-

tions of indoor air temperature and cooling demands (Fig. 12, T_in vs Cool) is overestimated by the GAN. A similar pattern is also observed for the probability distributions of indoor relative humidity and the average unmet cooling set-point difference (Fig. 12, RH_in vs Unmet).

It is interesting that the GAN generally underestimates large Wasserstein distances (i.e. low similarities), yet captures small Wasserstein distances (i.e. high similarities) very well (Figs. 10 and 11). In fact, a comparable behavior is observed when analyzing the Pearson correlation coefficients (Figs. 9 and 11), as the GAN suitably captures strong covariations, but underestimates covariation values close to zero. Such behavior may hint on minor overfitting tendencies in the trained GAN and require further in-depth analysis in future research.

4.3. Synthetic data for evaluation purposes

Synthetic data can be employed to evaluate the response of a model to out-of-sample occurrences. This notion can be useful for machine learning models, particularly when the training data partially represents the whole environment. For instance, one year of measurements is insufficient to fully capture the fluctuation range of climate variables as well as the randomness of occupants' behavior. Since we established that the data generated by the GAN can be considered as plausible uncertain scenarios of building's performance, the synthetic data are utilized to evaluate the response of a machine learning model to uncertainties.

Given that the GAN in this study is developed based on the CityLearn OpenAI Gym environment, we use the predefined RL model within CityLearn for evaluation purposes. The chosen model is a single centralized agent to control all nine buildings within the district. The RL agent is trained for 15 epochs on the climate data provided by CityLearn, which is also utilized in this study to train the
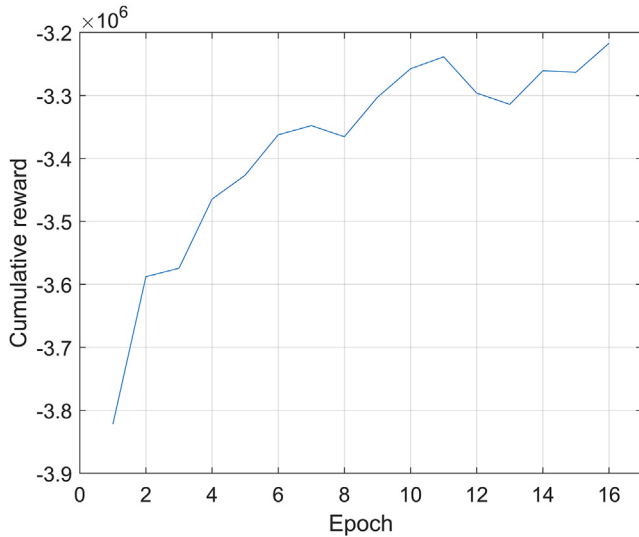
Fig. 13. Performance of the RL model in 15 epochs of training.

Fig. 15. Comparison of RL and RBC models' ramping using default and synthetic data.

GAN and project synthetic building performance profiles. The performance of the RL agent during training is provided in Fig. 13.

Once training of the RL agent is concluded, each building's performance in the Gym environment is replaced with a set of synthetic scenarios, returning 50 uncertain environments for the entire district. The trained RL agent is then deployed on the environment to control electricity and cooling storage. The performance of the RL agent is compared to that of a RBC.

Fig. 14 compares the performances of RL and RBC in shaving the daily electricity peaks. The RL agent outperforms the RBC in both the default training data and the synthetic profiles. However, its performance on the synthetic data (Fig. 14, RL synthetic) is heavily degraded when compared to the default training data (Fig. 14, RL default). On the other hand, comparing RBC's performance on the synthetic data (Fig. 14, RBC synthetic) and the default data (Fig. 14, RBC default) shows that the controller is less sensitive to uncertainties and disturbances.

Comparing the performance of the RL agent with that of the RBC under uncertain scenarios also shows that the superiority of the RL agent can greatly vary depending on the objective. For instance, relying on the default data would indicate that the RL agent
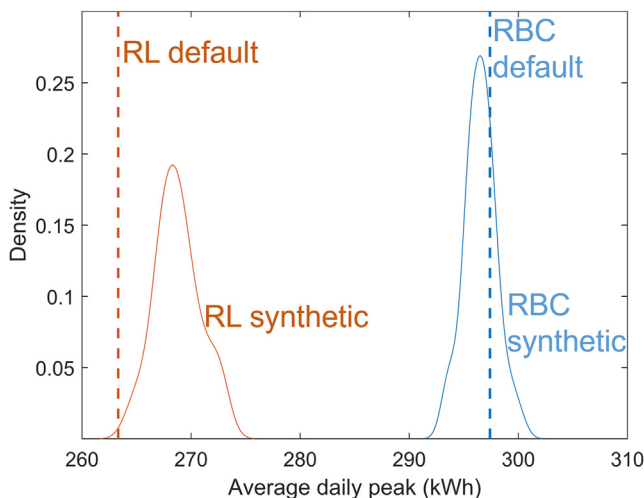
(Fig. 15, RL default) outperforms the RBC (Fig. 15, RBC default) in reducing the ramping effect. Here, the ramping effect refers to the absolute difference of the net non-negative electricity consumption at every time-step. We observe that when encountering unseen data, the RBC (Fig. 15, RBC synthetic) outperforms the RL agent (Fig. 15, RL synthetic) in reducing the ramping effect. Furthermore, we witness that the RBC is much less sensitive to uncertainties and disturbances. This is apparent from the width of RBC's distribution (Fig. 15, RBC synthetic), which is significantly narrower than that of the RL agent (Fig. 15, RL synthetic).

5. Discussion

Element-wise and statistical analysis of the results show that the proposed GAN can potentially replicate a building's dynamic response to climate and operation variations. This is evident from the good agreement between the characteristics of the original training data and the synthetic projections from the GAN. Yet, the synthetic data occasionally display slight divergences from the original dataset, which merits further investigation.

The GAN projections properly capture strong correlations between building performance and climate conditions. For instance, when the Pearson correlation coefficients is large (Fig. 9, T_out vs T_in) (Fig. 11, DHW vs Elec), the synthetic data show very similar performance to the original dataset. On the other hand, our model may slightly underestimate small and insignificant correlations. For instance, when the Pearson correlation coefficient is small (Fig. 9, R_dir vs RH_in) (Fig. 11, RH_in vs Unmet) synthetic projections could display divergence from the original dataset.

We believe that this phenomenon is due to minor overfitting of the discriminator onto the training data. When the Wasserstein distance is large, synthetic projections overestimate the distances (Fig. 10, R_dir vs RH_in) (Fig. 10, RH_out vs Cool) (Fig. 12, T_in vs Cool) (Fig. 12, RH_in vs Unmet). Namely, when the similarity between two variables is already small, our model is likely to further suppress the similarity. On the other hand, our model occasionally overestimates small Wasserstein distances. Namely, when the similarity between two variables is large (Fig. 12 DHW vs Elec) our model may exaggerate the similarity.

This presumed overfitting behavior can be due to memorization of small pieces of the training data as also reported in [40]. The

Fig. 14. Comparison of RL and RBC models' average daily peak using default and synthetic data.

issue may be addressed by modifying the architecture of the GAN as well as the number of training iterations. We believe that minor overfitting on strong correlations would not necessarily return a set of implausible synthetic scenarios, yet it might impede proper exploration of the domain. At the worst case, the synthetic projections would only represent a subset of the 'universe of discourse' of uncertain scenarios.

Furthermore, the temporal dependency between consecutive days was not evaluated in this study. Rather, we focused on the temporal consistency within the each daily profile. It would be interesting to assess how continuous profiles such as indoor air temperature varies from the last hour of one day to the first hour of the next. In fact, expanding the GAN proposed in this study with the recurrent components of the time-series GAN [41] is a potential for future research.

It is important to stress that the proposed GAN architecture has been devised while taking GPU performance into consideration, particularly to ensure reproducibility for peers with access to limited graphical memory. This said, using larger models with more complex architectures such as Variational AutoEncoder GANs (VAE-GAN) [42] could potentially improve the results and alleviate concerns over possible overfittings.

To understand GAN's capability of representing a particular domain, recent studies have proposed out-of-sample testing that estimate the reconstruction error and likelihood of each generated sample [43]. Conditioned that multiple years of building performance data is available, one can quantify the potential of the proposed GAN in reconstructing uncertain scenarios beyond the training set.

Evaluating the performance of a trained RL model on synthetic data showed that a small set of training data can result in overoptimistic expectations of a model's performance. However, it is important to note that the synthetic projections generated in this study do not cover the entire range of uncertainties and disturbances within the environment. Therefore, it would be incorrect to presume that the results provided in this study would favor rule based algorithms over data-driven models. Rather, the information shared in the paper tends to open a discussion on unexpected responses of trained machine learning models to unseen data, as well as the potential of GANs for underlining this vulnerability through a synthetic test set.

The synthetic data generated in this study could also improve the performance of the RL model if used as training data. In fact, studies have shown that adding synthetic data to the training set can improve a model's generalization, particularly for achieving better performances on rare events [44]. However, such assessment would require a separate test set which has not been seen by either the RL model or the GAN. Once again, access to multiple years of building performance data would help validate this hypothesis; namely, whether adding synthetic data to the training set can improve the performance of a model – such as the RL trained in this study – amid unseen disturbances in the test set.

The proposed method for generating synthetic building performance data is particularly useful when white-box modelling is not a viable option. For instance, if information about the building characteristics (geometry, thermal characteristics, HVAC systems, etc.) are unavailable, or when the sheer magnitude of studied buildings makes white-box modeling cumbersome and labor intensive. Previous studies are unable to generate synthetic building performance data for an extended period (e.g. a full year), or generate multiple outputs with internal covariations. The method proposed in this study addresses both shortcomings through conditioning the GAN on external features. However, the setup may suffer from a number of drawbacks and require improvements as described in the following.

1) It is likely that the proposed model is slightly overfitted onto the data, given the limited number of samples and lack of an independent test set.
2) In the proposed model, there are no provisions to account for the continuity of data from one day to another.
3) The behavior of the proposed model when fed with new and unseen climate data for encoding has not been studied.
4) The GAN is conditioned on the type of the day, rather than the hourly occupancy schedule, which implies that changes in building's tenant would require a new training from scratch.
5) Although the synthetic data differ from the actual measurements, they are likely to preserve behavioral patterns and remain susceptible to revealing personal information. The synthetic profiles should not be treated as anonymized data.

6. Conclusion

This paper proposed the application of GANs for creating synthetic building performance data. The model introduced in this study is conditioned based on climate and operative variables with the aim to control weekly and seasonal variations of the outputs. Qualitative and quantitative validation of the synthetic profiles showed that the proposed GAN can properly reflect climate and operation variations into the outputs. Furthermore, the proposed model successfully infused uncertainty into the building performance profiles and generated out-of-sample events. However, the GAN slightly overrepresented some covariations with climate and operation conditions, which can be attributed to small overfitting onto the training data.

The uncertainty-infused synthetic data generated by the GAN was utilized to evaluate the response of an RL model to unseen scenarios. Results showed that the RL model displayed mix performances to out-of-sample data, insofar that the RBC model occasionally outperformed the RL model. Furthermore, given that the RL model is highly reliant on the training data, its performance displayed higher sensitivity to uncertain events.

A particular strength of data-driven models is that their performance can be improved by continuing the training process on new sets of data. Therefore, inclusion of synthetic profiles in the training dataset can potentially alleviate the concerns over the performance of data-driven models as highlighted in this study, and improve their robustness to unseen events. Furthermore, addition of synthetic profiles to the training dataset of an RL model will influence the learned policies, including reliance on operation-related and climate-related profiles. Evaluating such changes in a model's learned policy could potentially pave the way for seamless transfer of pre-trained RL models across different buildings with dissimilar climate and operation characteristics.

CRediT authorship contribution statement

Fazel Khayatian: Data curation, Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing – original draft. **Zoltán Nagy:** Resources, Conceptualization, Formal analysis, Writing - review & editing. **Andrew Bollinger:** Resources, Conceptualization, Writing - review & editing, Supervision, Funding acquisition, Project administration.

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.

Acknowledgement

References

[1] P.M. Ferreira, A.E. Ruano, S. Silva, E.Z.E. Conceição, Neural networks based predictive control for thermal comfort and energy savings in public buildings, Energy Build. 55 (2012) 238–251.

[2] A. Jain, F. Smarra, M. Behl, R. Mangharam, Data-driven model predictive control with regression trees-an application to building energy management, ACM Trans. Cyber-Physical Syst. 2 (1) (2018) 1–21.

[3] F. Bünning, B. Huber, P. Heer, A. Aboudonia, J. Lygeros, Experimental demonstration of data predictive control for energy optimization and thermal comfort in buildings, Energy Build. 211 (2020) 109792, https://doi.org/10.1016/j.enbuild.2020.109792.

[4] Y. Sun, F. Haghighat, B.C.M. Fung, A review of the-state-of-the-art in data-driven approaches for building energy prediction, Energy Build. 221 (2020) 110022.

[5] A. Subbaswamy, R. Adams, S. Saria, Evaluating Model Robustness and Stability to Dataset Shift, in: International Conference on Artificial Intelligence and Statistics, 2021, pp. 2611–2619.

[6] T. Hong, D. Macumber, H. Li, K. Fleming, Z. Wang, Generation and representation of synthetic smart meter data, Build. Simul. 13 (6) (2020) 1205–1220.

[7] Z. Wang, T. Hong, Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN), Energy Build. 224 (2020) 110299.

[8] J. Roth, A. Martin, C. Miller, R.K. Jain, SynCity: using open data to create a synthetic city of hourly building energy estimates by integrating data-driven and physics-based methods, Appl. Energy 280 (September) (2020) 115981.

[9] C. Fan, F. Xiao, C. Yan, C. Liu, Z. Li, J. Wang, A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning, Appl. Energy 235(May 2018) (2019) 1551–1560.

[10] Y. Bengio, É. Thibodeau-Laufer, G. Alain, J. Yosinski, "Deep generative stochastic networks trainable by backprop, in: 31st Int. Conf. Mach. Learn. ICML 2014, vol. 2, 2014, pp. 1470–1485.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.

[12] K. Bao, K. Ibrahimov, M. Wagner, H. Schmeck, Enhancing neural non-intrusive load monitoring with generative adversarial networks, Energy Informat. 1 (Suppl 1) (2018) 295–302.

[13] Z. Chen, C. Jiang, Building occupancy modeling using generative adversarial network, Energy Build. 174 (2018) 372–379.

[14] C. Fan, Y. Sun, Y. Zhao, M. Song, J. Wang, Deep learning-based feature engineering methods for improved building energy prediction, Appl. Energy 240 (February) (2019) 35–45.

[15] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, in: 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc., 2018, pp. 1–26.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.

[17] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.

[18] M. Mirza, S. Osindero, Conditional generative adversarial nets, arXiv Prepr., arXiv:1411.1784, 2014.

[19] H. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network, IEEE Trans. Circuits Syst. Video Technol. 30 (11) (2020) 3943–3956.

[20] G. Douzas, F. Bacao, Effective data generation for imbalanced learning using conditional generative adversarial networks, Expert Syst. Appl. 91 (2018) 464–471.

[21] C. Zhang, J. Li, Y. Zhao, T. Li, Q. Chen, X. Zhang, A hybrid deep learning-based method for short-term building energy load prediction combined with an interpretation process, Energy Build. 225 (2020) 110301.

[22] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, Handb. Brain Theory Neural Netw. 3361 (10) (1995) 1995.

[23] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation Applied to Handwritten Zip Code Recognition, Neural Comput. 1 (4) (1989) 541–551.

[24] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 4th Int Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc. (2016) 1–16.

[25] J. Gauthier, Conditional generative adversarial nets for convolutional face generation, in: Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester 2014(5) (2014) 2.

[26] J.R. Vázquez-Canteli, J. Kämpf, G. Henze, Z. Nagy, CityLearn v1.0: an OpenAI gym environment for demand response with deep reinforcement learning, in: BuildSys 2019 - Proc. 6th ACM Int. Conf. Syst. Energy-Efficient Build. Cities, Transp., pp. 356–357, 2019.

[27] J.R. Vazquez-Canteli, G. Henze, Z. Nagy, MARLISA: Multi-Agent Reinforcement Learning with Iterative Sequential Action Selection for Load Shaping of Grid-Interactive Connected Buildings, in: BuildSys 2020 - Proc. 7th ACM Int. Conf. Syst. Energy-Efficient Build. Cities, Transp., pp. 170–179, 2020.

[28] Intelligent Environments Laboratory (IEL), Intelligent-environments-lab / CityLearn, 2021. [Online]. Available: https://github.com/intelligent-environments-lab/CityLearn. [Accessed: 02-Mar-2021].

[29] F. Khayatian, Seer-Synthetic Energy & Environment Replicator, 2021. [Online]. Available: https://github.com/Khayatian/seer. [Accessed: 02-Mar-2021].

[30] F. Khayatian, Synthetic building performance data," Zenodo, Jul-2021.

[31] Y.R. Cui, Q. Liu, C.Y. Gao, Z. Su, FashionGAN: display your fashion design using conditional generative adversarial nets, Comput. Graph. Forum 37 (7) (2018) 109–119.

[32] D.P. Kingma, J.L. Ba, "Adam: A method for stochastic optimization, 3rd Int Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2015) 1–15.

[33] B. Amos, Image Completion with Deep Learning in TensorFlow. http://bamos.github.io/2016/08/09/deep-completion [Accessed: 02-Mar-2021].

[34] T. Kaneko, Y. Ushiku, T. Harada, Label-noise robust generative adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2467–2476.

[35] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, Adv. Neural Inf. Process. Syst. (2016) 2234–2242.

[36] M. Arjovsky, Ĺ. Bottou, Towards principled methods for training generative adversarial networks, arXiv Prepr., arXiv:1701.04862, 2017.

[37] J. Vazquez Canteli, Z. Nagy, The CityLearn Challenge 2020, Texas Data Repository, https://doi.org/10.18738/T8/ZQKK6E, 2020.

[38] J. Benesty, J. Chen, Y. Huang, I. Cohen, Pearson correlation coefficient, in: Noise reduction in speech processing, Springer, 2009, pp. 1–4.

[39] S.S. Vallender, Calculation of the Wasserstein distance between probability distributions on the line, Theory Probab. Its Appl. 18 (4) (1974) 784–786.

[40] V. Nagarajan, C. Raffel, I.J. Goodfellow, Theoretical Insights into Memorization in GANs, in: Neural Information Processing Systems Workshop (Vol. 1), 2018.

[41] J. Yoon, D. Jarrett, M. van der Schaar, Time-series generative adversarial networks, Adv. Neural Inf. Process. Syst. 32 (NeurIPS) (2019) 1–11.

[42] A.B.L. Larsen, S.K. Sønderby, H. Larochelle, O. Winther, Vae/Gan, Icml 4 (2016) 2341–2349.

[43] P. Sánchez-Martín, P.M. Olmos, F. Perez-Cruz, Out-of-sample testing for GANs, arXiv Prepr., arXiv:1901.09557, 2019.

[44] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, N. Joshi, M. Meister, P. Perona, Synthetic examples improve generalization for rare classes, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2019, pp. 863–873.