This document contains the Repeatability Evaluation Package (REP) followed by the paper, and the supplementary information at the end.

Repeatability Evaluation Package (REP) Documentation for the Paper

“Learning Spatio-Temporal Aggregations for Large-Scale Capacity Expansion Problems”

Authors: Rahman Khorramfar, Aron Brenner, Saurabh Amin

1 Overview

The methodology presented in the paper consists of two parts. The first part is the autoencoder (AE) procedure explained in Section 4, and the second part is the capacity expansion model (i.e., GTEP) presented in Section 2. Therefore, the REP consists of two main folder: 1)‘*AE-Module’* folder which contains all the data and codes used in the AE; 2) ‘*GTEP-Module’* folder which contains all the data and codes used in the GTEP. All the data used are from publicly available sources, hence our submission contains no proprietary data.

Each successful run involves at most two steps. In the first step the AE module is run to obtain spatio-temporal aggregations for the network size and number of representative days. The second step uses the output from the first step to run the GTEP for the given parameters. Both modules are implemented in Python with transferability and legibility in mind. The details of both steps is given in the Section 3 and 4 of this document.

The codes and data are all available in the GitHub repository at:
https://github.com/RahmanKhorramfar91/ICCPS-2023

2 System and Resource Requirements

All the codes are developed and tested in Python 3.8. The codes do not need installation as they are presented in the raw script format. The following packages are required for a successful run:

```
numpy, pandas, matplotlib, random, geopp, os, gurobipy, time, sys, pytorch, torch_geometric, torch_sparse, torch_scatter, sklearn, scikit-learn-extra, networkx, tqdm, datetime
```

Note that the systems must have Anaconda as wells as Gurobi solver installed. We run the AE module on a personal MacBook laptop, and run the GTEP module on the MIT Supercloud system (Intel Xeon Platinum 8260 processor with up to 48 cores and 192 GB of RAM) with multiple instances running in parallel. However, both modules are successfully tested on Windows machines. All the packages are generic Python packages, so we expect a smooth run on other machines with recent version of Windows, Linux, or Mac.

3 How to Run

This section explains the running procedure for both modules. The first module, which is AE, usually runs under 4 hours depending on the host machine and aggregation parameters. The run time of GTEP greatly varies based on the spatio-temporal aggregation level as well as problem parameters, but they are usually run under 10 hours.

3.1 Autoencoder

As mentioned, all code and data related to the autoencoder are available in the folder '*AE-Module'*. The only code that needs to be run can be found in `Autoencoder.ipynb`, and all data is given in the '*Data'* folder.

To train the spatial and temporal aggregation autoencoders, the Jupyter notebook `Autoencoder.ipynb` can be run cell by cell. The data can be uploaded and processed by running the cells under the section "*Constructing Datasets*." Specifically, the electricity load data is given in "`Data/Power Network Topology-full network (188 nodes)/bus_load_RM_2050.csv`", the NG data is given in "`Data/ng_daily_load2050_RM.csv`", the wind CF data is given in "`Data/wind-CF-188-nodes.csv`", and the solar CF data is given in "`Data/solar-CF-188-nodes.csv`".

To train the spatial aggregation autoencoder, one should run all cells in the section titled '*Spatial Aggregation Autoencoder*'. The parameters/variables to be tuned are as follows:

- Number of spatial clusters: the final spatial resolution (number of nodes) of the GTEP model. This tunes the graph pooling block in the autoencoder.
- Learning rate (*lr*): the learning rate for the Adam optimizer. This should be tuned to minimize the autoencoder validation loss.
- Epochs: the number of iterations of gradient descent during training. This should also be tuned to minimize the autoencoder validation loss.

The code then saves the learned spatial clusters to the file '`spatial_cluster.csv`'. This file has two columns:

- Node: the node label.
- Cluster: the assigned cluster of the corresponding node, or in other words, the node label to which it is absorbed in the spatially aggregated GTEP.

Similarly, to train the temporal aggregation autoencoder, one should run all cells before the Training subsection under '*Spatial Aggregation Autoencoder*' as well as all cells in the section titled '*Temporal Aggregation Autoencoder*'. The parameters/variables to be tuned are as follows:

- Days: the number of representative days to be used in the temporally aggregated GTEP. This is a hyperparameter for the K-medoids algorithm.
- Learning rate (*lr*): the learning rate for the Adam optimizer. This should be tuned to minimize the autoencoder validation loss.
- Epochs: the number of iterations of gradient descent during training. This should also be tuned to minimize the autoencoder validation loss.
- *alpha_G, alpha_W, alpha_S*: the objective weight coefficients corresponding to NG, wind CF, and solar CF reconstruction loss (relative to the electricity reconstruction loss).
- *max_iter*: the number of iterations for the K-medoids algorithm.

The code then saves the learned representative days to the file `temporal_cluster.csv`. This file has three columns:

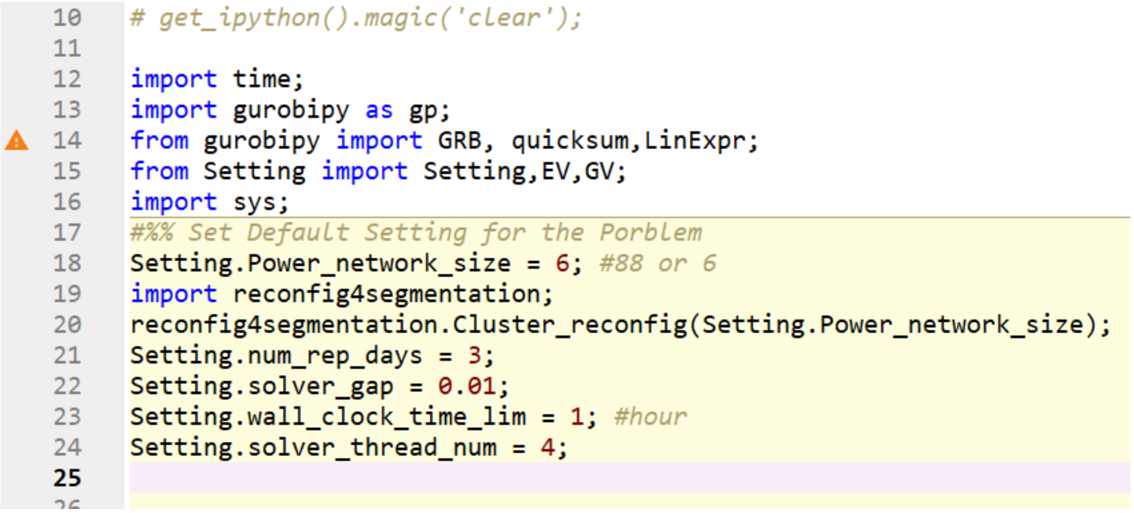
- Day of Year: number corresponding to the day of the year (e.g., 0 corresponds to January 1)
- Date
- Weight: the relative weight of the representative day in the temporally aggregated GTEP. This is given by the number of members of the cluster as learned by the K-medoids algorithm.

3.2 GTEP

Once the Autoencoder codes are run, the output is stored in '*joint_CF_with_extreme_days*' folder insider the 'GTEP-Module' folder. The 'GTEP-Module' folder contains several folders, Python scripts, and CSV file from which we only explain the files that are directly related to running the GTEP model. The GTEP model runs from `UB.py` script. The code can both be run from the Anaconda's command window or an IDE such as Spyder. The scrips require 5 key parameter as follows:

- Network size: the number of nodes in the power system. The parameters should set to one 6, 10, 15, or 20. Default value is 6.
- Number of representative days: the number of planning days considered in the model. The parameters can take any integer number between 2 and 30. Default value is 3.
- Solver Gap: the MIP gap for Gurobi to terminate. The parameters is any number between 0 and 1. Default value is 0.01.
- Solver Time Limit in terms of hours: the solution time limit for the solver. The parameter can take any integer number. Default value is 1.
- Solver Thread: the number of CPU core threads dedicated to the solver. The parameters takes integer numbers and can vary between 1 and the number of thread on the machine. Default is 4.

The following shows these parameters in the `UB.py` opened in Spyder:

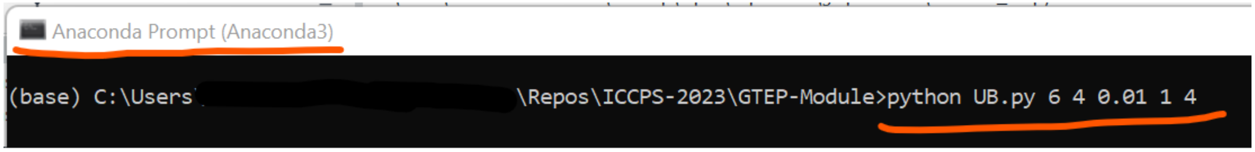


```
# get_ipython().magic('clear');

import time;
import gurobipy as gp;
from gurobipy import GRB, quicksum,LinExpr;
from Setting import Setting,EV,GV;
import sys;
#%% Set Default Setting for the Porblem
Setting.Power_network_size = 6; #88 or 6
import reconfig4segmentation;
reconfig4segmentation.Cluster_reconfig(Setting.Power_network_size);
Setting.num_rep_days = 3;
Setting.solver_gap = 0.01;
Setting.wall_clock_time_lim = 1; #hour
Setting.solver_thread_num = 4;

```

The following is an example of running the script from the anaconda prompt:



```
(base) C:\Users[redacted]\Repos\ICCPS-2023\GTEP-Module>python UB.py 6 4 0.01 1 4
```

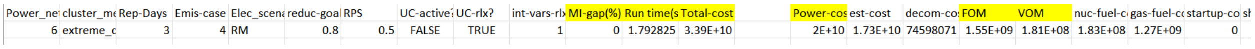
●

The parameters after `UB.py` should be separated by a comma and are in the order they were explained above.

The results of each run is stored in the output file `JPoNG_Results.csv`. The code reports a detailed output in 87 columns and two rows. The first row is the header for each column and the other column is the value for each column. The initial columns show the parameters of the instance. Column K and L show MIP gap and run time of the problem in seconds. Other columns provide details of the solution and are named mnemonically for readability. The following columns used in the paper:

- Column N (Total-cost): total cost of the GTEP model including the total cost of power and NG systems
- Column O (*Power-cost*): total cost of the power system
- Column P (*est-cost*): power plant establishment cost
- Column R (*FOM*): fixed operating and maintenance cost for the power system
- Column S (*VOM*): variable operating and maintenance cost for the power system
- Column AH (*NG-cost*): total cost of the NG system

The following figure shows part of the solution output for an instance:



Power_ne	cluster_me	Rep-Days	Emis-case	Elec_scena	reduc-goal	RPS	UC-active?	UC-rlx?	int-vars-rlx	MI-gap(%)	Run time(s	Total-cost		Power-cos	est-cost	decom-co:	FOM	VOM	nuc-fuel-c	gas-fuel-c	startup-co	sh
6	extreme_c	3	4	RM	0.8	0.5	FALSE	TRUE	1	0	1.792825	3.39E+10		2E+10	1.73E+10	74598071	1.55E+09	1.81E+08	1.83E+08	1.27E+09	0	

All computational experiments reported in the paper can be repeated by running the GTEP model with different parameters. The submitted code can also be run for many other instances with certain configurations.

Learning Spatio-Temporal Aggregations for Large-Scale **Energy** Capacity Expansion Problems

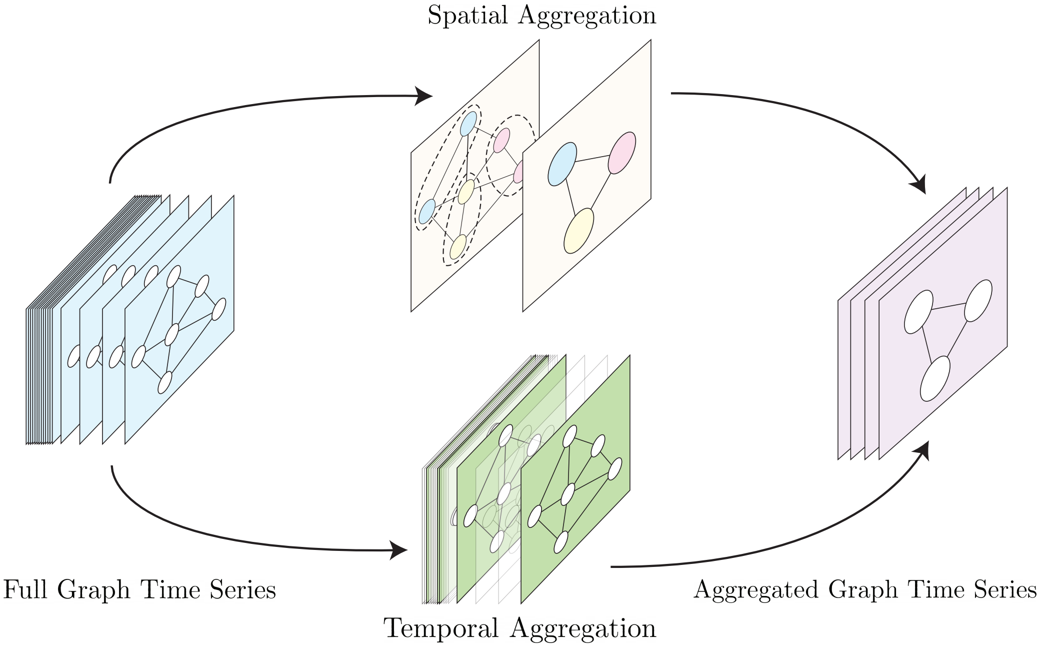


Fig. 1. Spatio-temporal aggregation is often needed to tractably solve capacity expansion problems. Spatial aggregation is applied to reduce the number of decision variables and constraints associated with individual nodes while temporal aggregation is applied to reduce the number of decision variables and constraints associated with operational periods.

Operational performance of future cyber-physical infrastructures crucially relies on the effectiveness of long-term planning decisions related to network capacity expansion. Such Capacity Expansion Problems (CEPs) quickly become intractable due to a large number of operational periods and realistic network sizes. Traditionally, this issue is tackled by selecting some representative periods and/or aggregating the network. This process often entails some rather ad hoc choices, leading to suboptimal or even infeasible solutions to the original CEP.

Here, we propose a data-driven approach to finding spatio-temporal aggregations that are useful in efficiently solving CEPs. Our approach is based on leveraging the (projected) time series of relevant supply-demand variables (e.g., nodal demand, generation capacity factors) to train a graph autoencoder model with a pooling mechanism. By minimizing the expected reconstruction loss over operational periods, we obtain data-driven spatial aggregations (i.e., clusterings of nodes). To obtain the temporal aggregation (i.e., set of representative periods), we cluster the latent representation output from the encoder. The resulting aggregations can be used to obtain high-quality solutions to large-scale CEPs. We demonstrate the efficacy of our approach by solving for generation and transmission expansion planning decisions for an interdependent power-NG model calibrated for the New England region. We also report computational results on the sensitivity of planning decisions to the extent of aggregation. Interestingly, our approach provides non-trivial upper bounds on the optimal objective of the original large-scale CEP.

ACM Reference Format:
. 2022. Learning Spatio-Temporal Aggregations for Large-Scale **Energy** Capacity Expansion Problems. 1, 1 (February 2022), 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Capacity Expansion Problems (CEPs) are a family of optimization formulations that aim to assist in generating high quality planning decisions for a broad range of applications in cyber-physical networks, including communication systems, water resource systems, and regional power networks [Luss 1982]. To minimize total cost of operations over a long-term horizon, CEP formulations integrate long-term investment decisions, such as production capacity expansion, facility location/decomissioning, with hour-to-hour operations, such as generation and transmission of resources. Naturally, the inclusion of these planning and operational considerations across a large network and over a multidecadal horizon makes CEP formulations powerful modeling tools. However, these features also

Author's address:

contribute to the computational intractability of solving such problems.

To this end, it has become common practice for modelers to *aggregate* CEP formulations both spatially by grouping nodes, and also temporally by selecting a set of "representative periods" to roughly capture the distribution of demands, capacity factors, and other parameters of interest across operational periods. Given that the planning decisions depend closely on the network structure and planning horizon, the choice of aggregation can crucially impact the CEP's solution. Machine learning (ML) methods have indeed been applied previously to learn high quality temporal aggregations [Teichgraeber and Brandt 2022]. However, to the best of our knowledge, applications of ML towards learning *spatial* aggregations are yet to be explored. Motivated by the enormous potential of ML models in learning representations as well as recent developments in geometric deep learning, we propose a framework for identifying potential *spatio-temporal* aggregations using a graph convolutional autoencoder approach. We apply our framework to aggregate and solve an otherwise intractable CEP for a regional energy network. Our methodology proposes a new direction in making CEPs tractable by means of spatio-temporal aggregation and can potentially yield high quality planning decisions for sustainable and efficient operations of cyber-physical infrastructure.

Specifically, we apply our framework to address several practical and computational challenges associated with capacity expansion models (CEMs) for decarbonization of interdependent power-NG infrastructures. Classical examples of such models include the generation expansion problem (GEP) and generation and transmission expansion problem (GTEP), both of which are well-studied in the context of power systems [He et al. 2018; Li et al. 2022a]. In this paper, we focus on a GTEP formulation that determines the optimal location and timing of generation units, transmission lines, and pipelines to meet future energy demands under a range of operational and policy constraints such as joint emission constraints. In our work, we extend the model to include two main interdependencies between power and NG systems. The first interdependency captures the increasing role of gas-fired power plants in the generation mix of electricity production [(EIA) 2022; He et al. 2018]. The second interdependency reflects the *joint* emission of CO_2 in both systems.

The key *computational challenge* in solving the GTEP arises from the fact that it links long-term investment decisions (e.g. capacity and network expansion) to short-term operational ones (e.g. unit commitment, power production, and energy storage). The former decisions have a planning horizon of 10-30 years with yearly granularity, while the latter usually require hourly or sub-hourly resolution. CEPs in general, and GTEPs in particular, are usually a large-scale mixed-integer linear program (MILP), but current literature has limited success in tractably solving these problems to an adequate level of spatial and temporal resolution. In our case, the computational difficulty in solving the GTEP increases further because we model both power and NG networks. Thus, taking into account (projected) demand information on a day-to-day basis becomes prohibitively expensive from a computational viewpoint. In the classical GTEP problems for power systems, the computational challenge is addressed by aggregating power system nodes (buses) within a geographical neighborhood (power zone) to a single node [Li et al. 2022a] and by solving the GTEP for a set of representative days [Hoffmann et al. 2020]. Crucially, the new formulation instantiated on this node aggregation and representative day set must proportionately capture demand and supply patterns across the network and throughout the planning horizon.

Our work also addresses the *practical issues* arising from coarse data availability from the NG network. Firstly, we do not have access to the detailed connectivity and transmission information in the NG network while this information is readily available for the power network. Secondly, power systems typically collect demand and generation data at a fine temporal resolution (hourly or less), but this data is usually not publicly accessible for NG systems. These issues thus require us to (a) formulate network constraints based on loosely specified information on power and NG node connectivity and (b) develop an approach to leverage demand and supply data from the power system with demand data of NG system despite their different temporal resolutions.

We address the aforementioned challenges by developing a *graph convolutional autoencoder approach* that captures the physical interdependencies between power and NG networks and also handles the different granularity of data at each network. We consider demand data for both systems, and consider capacity factor (CF) data for solar and wind plants to reflect the supply pattern in the renewable-dominated future grid. We utilize graph convolutions to capture the network interactions both within and across power and NG networks, and adopt an autoencoder architecture with tuneable reconstruction losses for the respective input data. Moreover, we incorporate a graph pooling mechanism to automatically learn spatial aggregations that group together nodes that exhibit similar electricity demand behavior. We demonstrate that the resulting autoencoder models are ideally suited to learning latent embeddings of the spatio-temporal patterns in power and NG demand as well as wind and solar CF data, which can be readily incorporated into high quality spatio-temporal aggregations. Furthermore, our approach to identifying spatio-temporal aggregations can also enable an accurate estimation of the trade-off between costs (both investment and operational) and joint emissions from power and NG systems.[1]

Previous studies for spatio-temporal aggregation focus on selecting sets of representative days using variants of k-means [Barbar and Mallapragada 2022; Li et al. 2022b; Mallapragada et al. 2018; Teichgraeber and Brandt 2019], k-medoids [Scott et al. 2019; Teichgraeber and Brandt 2019], and hierarchical clustering [Liu et al. 2017; Teichgraeber and Brandt 2019]. The distance matrices used in clustering algorithms for most previous works are constructed based on a set of time series inputs such as load data and variable renewable energies (VRE) capacity factors [Hoffmann et al. 2020; Li et al. 2022a]. Notably, these approaches neither account for demand data with multiple time resolutions nor account for network interdependencies. Hence, they cannot be readily extended to address the task of extracting representative days for joint power-NG systems – an aspect that is crucial for realism and tractability in joint planning optimization models for decarbonizing these systems.

[1] We believe this capability can have a significant societal impact by lowering the barriers to investment in renewable energy resources and alleviating reliability concerns in a low-carbon energy system.

Moreover, none of these studies apply a data-driven approach to aggregating buses, which has the potential to more accurately represent heterogeneity of electricity demand behavior across nodes. We believe that our approach addresses these challenges and provides a promising path to better extract spatio-temporal aggregations in interdependent power and NG systems.

2 CAPACITY EXPANSION MODEL

Our formulation of the joint power-NG planning is based on the model proposed in [Brenner et al. 2022]. In this section, we briefly introduce the GTEP formulation for the joint power-NG system and provide its details in the Supplementary Information (SI) [SI 2022]. The problem determines the minimum investment and operational costs of co-optimizing electricity and natural gas (NG) systems for the year 2050 under various investment, operational, and policy constraints. The investment decisions for the power system include establishing new plants, transmission lines, and decommissioning existing plants, and for the NG system it includes establishing new pipelines. Our model consist of major power and NG systems' planning constraints such as minimum stable production, ramping, energy balance, flow, and storage. The interdependency between the two systems is realized through two sets of constraints. The first coupling constraint captures the flow of NG to power system to enable operation of gas-fired power plants. The second constraint imposes an economy-wide emission constraint, limiting the emission of CO_2 from both system to a pre-specified value.

To conceptualize the model, let $\boldsymbol{z}^e = (\boldsymbol{x}^e, \boldsymbol{y}^e, \boldsymbol{p})$ denote the set of variables for the power system. The integer variable $\boldsymbol{x}^e$ is the investment decision for the power system (i.e., establishing plants, decommissioning plants, and establishing new transmission lines). The continuous variable $\boldsymbol{p}$ denotes the power generation in gas-fired plants and $\boldsymbol{y}^e$ is a continuous variable that signifies all the remaining variables including power generation from non-gas-fired plants, electricity flow between nodes, storage charge/discharge, and load shedding variables. We define $\boldsymbol{z}^g = (\boldsymbol{x}^g, \boldsymbol{y}^g, \boldsymbol{f})$ to denote the NG system's variables. The variable $\boldsymbol{x}^g$ is a mixed-integer variable denoting all investment, storage, and load shedding decisions. The continuous variable $\boldsymbol{y}^g$ captures the flow inside the NG system. Finally, $\boldsymbol{f}$ denotes the flow of NG to power systems. We formulate the joint power-NG system as follows:

$$\min\ (\boldsymbol{c}_1^e \boldsymbol{x}^e + \boldsymbol{c}_2^e \boldsymbol{y}^e + \boldsymbol{c}_3^e \boldsymbol{p}) + (\boldsymbol{c}_1^g \boldsymbol{x}^g + \boldsymbol{c}_2^g \boldsymbol{y}^g + \boldsymbol{c}_3^g \boldsymbol{f}) \tag{1a}$$

$$\text{s.t. } \boldsymbol{A}^e \boldsymbol{x}^e + \boldsymbol{B}^e \boldsymbol{y}^e + \boldsymbol{D}^e \boldsymbol{p} \leq \boldsymbol{b}_1^e \tag{1b}$$

$$\boldsymbol{H}^e \boldsymbol{y}^e \geq \boldsymbol{b}_2^e \tag{1c}$$

$$\boldsymbol{A}^g \boldsymbol{x}^g + \boldsymbol{B}^g \boldsymbol{y}^g + \boldsymbol{D}^g \boldsymbol{f} \leq \boldsymbol{b}_1^g \tag{1d}$$

$$\boldsymbol{f} = \boldsymbol{E}_1 \boldsymbol{p} \tag{1e}$$

$$\boldsymbol{G}_2 \boldsymbol{y}^g + \boldsymbol{E}_2 \boldsymbol{p} \leq \eta \tag{1f}$$

$$\boldsymbol{x}^e \in \mathbb{Z}^+, \boldsymbol{y}^e, \boldsymbol{x}^g \in \mathbb{Z}^+ \times \mathbb{R}^+, \boldsymbol{p}, \boldsymbol{y}^g, \boldsymbol{f} \in \mathbb{R}^+ \tag{1g}$$

The objective function (1a) minimizes the investment and operational costs for the power system (first term) and NG system (second term). The constraint (1b) represents all investment, and operational constraints for the power system. The constraint (1c) ensures policy considerations such as the minimum requirement for renewable portfolio standard (RPS). The NG constraints are reflected in constraint (1d), which includes technological and operational constraints such as the supply limit at each node, flow between NG nodes, and storage.

The coupling constraint (1e) ensures that gas-fired plants operate based on the gas they receive from the NG network. The economy-wide decarbonization constraint is defined by constraint (1f) and limits emissions resulting from NG consumption in electricity generation and other sectors such as heating, transportation and industry to η. The coefficient matrices $\boldsymbol{E}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{E}_2$ represent the heat rate, emission factors for NG usage, and emission factor for gas-fired plants, respectively. Notice that the conceptualized model does not consider other major types of fossil-fueled plants such as coal and oil due to their declining role in the generation mix of US system

3 AGGREGATION FORMULATION

This section presents our formulation for a generation capacity expansion problem (CEP), an example of which is the GTEP considered in this paper. For ease of exposition, we only explain our method for the temporal aggregation as it can naturally be extended to spatial aggregation as well. Let $\boldsymbol{x} \in \mathbb{Z}^k$ denote the integer variables of size k associated with investment decisions. The variable $\boldsymbol{y} \in \mathbb{R}_+^n$ denotes the n continuous operational decisions whose size are subject to reduction as a result of temporal aggregation. A generic CEP can be defined as:

$$\begin{aligned} \mathcal{Z} \coloneqq \min_{\boldsymbol{x},\boldsymbol{y}} \quad & \boldsymbol{f}^\top \boldsymbol{x} + \boldsymbol{g}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} \\ \boldsymbol{R} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix} \\ & \boldsymbol{x} \in \mathbb{Z}^k \\ & \boldsymbol{y} \in \mathbb{R}_+^n. \end{aligned}$$

where constraints $\boldsymbol{Px} \leq \boldsymbol{a}$ and $\boldsymbol{Qy} \leq \boldsymbol{b}$ limit $\boldsymbol{x}$ and $\boldsymbol{y}$ variables respectively, and $\boldsymbol{Rx} + \boldsymbol{Sy} \leq \boldsymbol{c}$ is a coupling constraint linking $\boldsymbol{x}$ and $\boldsymbol{y}$ variables. We assume that $\boldsymbol{Q} \in \mathbb{R}^{m_1 \times n}$ and $\boldsymbol{S} \in \mathbb{R}^{m_2 \times n}$ and are matrices of compatible sizes.

A temporal aggregation involves representing the entire planning horizon with a set of representative periods (i.e, representative days in energy systems). This operations results in a problem $\mathcal{Z}$ with either fewer constraints ($m' < m$) or operational variables ($n' < n$). The resulting problem can also accommodate a coefficient change for the operational variables. For example, temporal aggregation in energy problems is usually carried out by clustering and the selected operational variables are appropriately weighted to reflect the number of periods each of them represent [Teichgraeber and Brandt 2022]. From optimization perspective, temporal aggregation is the operation of removing a set of constraints and projecting the full problem to a lower dimensional space. Let a parameterized mapping $\boldsymbol{\theta} = (\Lambda, \Sigma, \Gamma)$ be a function that carries out the temporal aggregation on the full problem where $\Gamma \in \mathbb{R}^{m' \times m}$ is a binary matrix that specifies which constraint to retain and which one to remove. The matrix $\Sigma \in \mathbb{R}^{n' \times n}$ encodes the variable aggregation

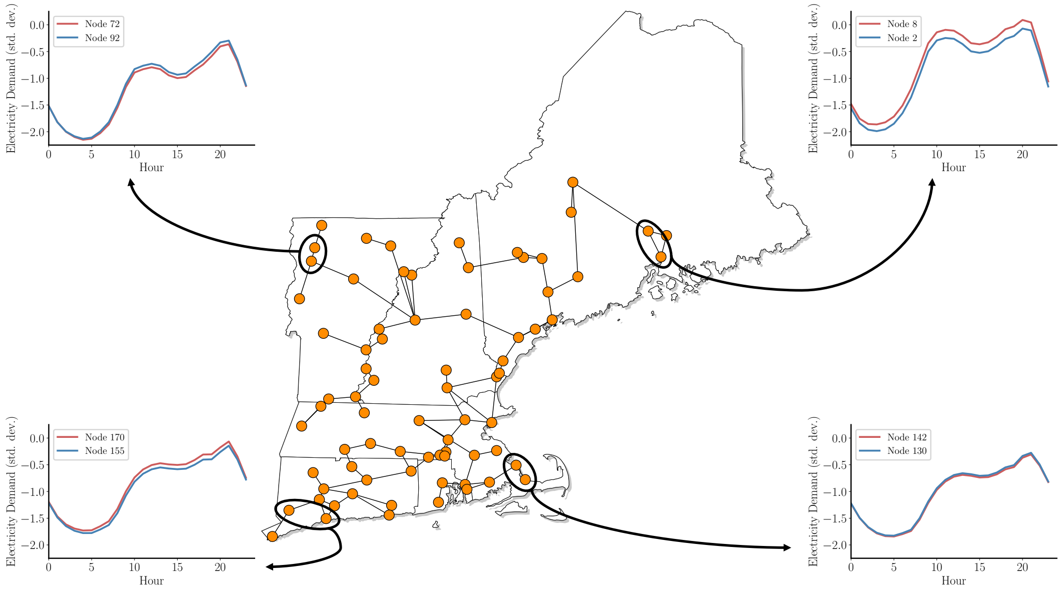


Fig. 2. Adjacent nodes in the power network demonstrate similar variations in demand over the course of the day. These spatial dependencies are modeled explicitly by graph convolutional layers in the proposed autoencoder architectures.

assignments, and $\Lambda \in \mathbb{R}^{n' \times n'}$ is a matrix that specifies the new coefficient of variables. A variable is represented by another variable if its corresponding entry in the Λ is zero. Specifically, $\Gamma_{ij} = 1$ if constraint j is kept as the i-th constraint in the aggregation; otherwise, $\Gamma_{ij} = 0$. We define $\Sigma_{ij} = \mathbb{1}_{ij}$ where $\mathbb{1}_{ij}$ indicates whether the j-th decision variable will be kept in the aggregation as the i-th decision variable. Λ is a diagonal matrix with Λ_{ii} encoding the "weight" of the i-th variable in the aggregation. Applying θ, the full model becomes:

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Q} \\ \boldsymbol{R} & \boldsymbol{S} \end{bmatrix} \mapsto \begin{bmatrix} \boldsymbol{P} & \boldsymbol{0} \\ \boldsymbol{0} & \Gamma \boldsymbol{Q} \Sigma^\top \\ \Gamma \boldsymbol{R} & \Gamma \boldsymbol{S} \Sigma^\top \end{bmatrix}, \quad \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \end{bmatrix} \mapsto \begin{bmatrix} \boldsymbol{a} \\ \Gamma \boldsymbol{b} \\ \Gamma \boldsymbol{c} \end{bmatrix}, \quad \boldsymbol{g} \mapsto \Lambda \Sigma \boldsymbol{g}$$

Note that be definition x variables are not subject to mapping. It is also worth noting that in practice, decision variables can be defined on a sub-period resolution. For example, while most energy system planning is expressed in term of days, the operational variables are defined on hourly basis. The generality of our exposition remain valid in such cases as one can define each operational variable of a period as a set of more granular variables. For illustrative purposes, consider a simple CEP problem with only three planning periods:

$$\begin{aligned} \mathcal{Z} := \min \quad & fx + g_1|b_1 - y_1| + g_2|b_2 - y_2| + g_3|b_3 - y_3| \\ \text{s.t.} \quad & y_1 \leq x, \quad y_2 \leq x, \quad y_3 \leq x \\ & x \in \mathbb{Z}_+ \\ & y_1, y_2, y_3 \in \mathbb{R}_+. \end{aligned}$$

where the constraints specify that production (an operational variables) of the capacity is limited by the established capacity (an investment decision). Letting b_i to be the demand in period i, the objective function wants to minimize the total cost of investment and the cost of deviations from demand at each period. Assume that period 1 is selected to represent periods 1 and 2. Then the linearized form of $\mathcal{Z}$ can be aggregated to $\mathcal{Z}_\theta$ as follows:

$$\begin{array}{ll} \min & fx + g_1 z_1 + g_2 z_2 + g_3 z_3 \\ \text{s.t.} & z_1 + y_1 \geq b_1 \\ & z_1 - y_1 \geq -b_1 \\ & z_2 + y_2 \geq b_2 \\ & z_2 - y_2 \geq -b_2 \\ & z_3 + y_3 \geq b_3 \\ & z_3 - y_3 \geq -b_3 \\ & -x + y_1 \leq 0 \\ & -x + y_2 \leq 0 \\ & -x + y_3 \leq 0 \\ & x \in \mathbb{Z}^+ \\ & y_1, y_2, y_3 \in \mathbb{R}_+ \\ & z_1, z_2, y_3 \in \mathbb{R}_+ \end{array} \quad \underset{\theta}{\mapsto} \quad \begin{array}{ll} \min & fx + 2g_1 z_1 + g_3 z_3 \\ \text{s.t.} & z_1 + y_1 \geq b_1 \\ & z_1 - y_1 \geq -b_1 \\ & z_3 + y_3 \geq b_3 \\ & z_3 - y_3 \geq -b_3 \\ & -x + y_1 \leq 0 \\ & -x + y_3 \leq 0 \\ & x \in \mathbb{Z}^+ \\ & y_1, y_3 \in \mathbb{R}_+ \\ & z_1, z_3 \in \mathbb{R}_+. \end{array}$$

Then the components of the mapping θ become:

$$\Lambda = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

3.1 Learning Aggregations

Let $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ and $(\boldsymbol{x}^*_{\boldsymbol{\theta}}, \boldsymbol{y}^*_{\boldsymbol{\theta}})$ denote the optimal solutions to $\mathcal{Z}$ and $\mathcal{Z}_{\boldsymbol{\theta}}$ respectively. Supposing that our objective for aggregation is to find a solution $\boldsymbol{x}^*_{\boldsymbol{\theta}}$ close to $\boldsymbol{x}^*$, we can formulate the aggregation objective as

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{x}^* - \boldsymbol{x}^*_{\boldsymbol{\theta}}\|. \tag{3}$$

Direct optimization of this objective is impossible without knowledge of $\boldsymbol{x}^*$. Moreover, even if one somehow knew $\boldsymbol{x}^*$ a priori, aggregating $\mathcal{Z}$ would still be a combinatorially hard problem. As such, we select the parameters $\boldsymbol{\theta}$ using surrogate objectives that are easier to optimize. A natural choice for such an objective would be the clustering objective associated with operational periods (in the case of temporal aggregation) and nodal features (in the case of spatial aggregation). By grouping nodes as well as representative periods in such a way as to faithfully capture the heterogeneity of nodal demands over the planning horizon, one can hope to (indirectly) minimize (3) and parametrize an aggregation $\boldsymbol{\theta}$ that in turn yields high quality planning decisions with computational tractability.

3.1.1 Temporal Aggregation. To select a temporal aggregation, we can apply clustering to the time series data corresponding to constraints (e.g., demand data, capacity factor data). Given the number of clusters or representative periods n', we apply a clustering algorithm such as k-medoids clustering to minimize the objective function

$$\min \sum_{j \in \mathcal{S}} \sum_{i \in C_j} \|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2^2 \tag{4}$$

[Hastie et al. 2001], where $\boldsymbol{b}_k$ denotes the vector of constraint features in operational period k. The set of representative days is denoted by $\mathcal{S}$, and the set of days clustered around (i.e., represented by) any day $k \in \mathcal{S}$ is denoted by C_k. Semantically, minimizing (4) can be understood as aiming to ensure that the set of representative days proportionately partitions the full set of days in the dataset by minimizing squared Euclidean distances between the time series of constraints. One might hope that in minimizing the clustering objective, the resulting set of cluster medians (i.e., representative days) and their corresponding weights proportionately capture the distribution of constraints, which will in turn yield an aggregation that is "closer" to the original formulation.

3.1.2 Spatial Aggregation. In contrast to a temporal aggregation, for which we aim to identify groups of *operational periods* with similar features (e.g., electricity demands, capacity factors), an ideal spatial aggregation would group *nodes* that demonstrate similarities in their realization of these time-varying features. With this in mind, we might hope to cluster the dataset along the spatial dimension.

3.1.3 Curse of Dimensionality. While standard clustering algorithms such as k-medoids clustering can be applied to the raw data, one might expect that such an approach would be prone to the "curse of dimensionality" due to the high dimensionality of the time series data and the large number of nodes. This is particularly true with spatial clustering, for which there are far more time series observations over the course of the planning period than there are nodes.

To resolve this challenge, we propose an autoencoder-based approach for spatio-temporal aggregation. Specifically, we learn temporal aggregations by training a graph convolutional autoencoder to reconstruct the multivariate time series corresponding to each operational period. In doing so, we are able to extract low-dimensional and denoised representations of the data, on which we can apply a clustering algorithm [Parsons et al. 2004]. Regarding spatial aggregation, we train an autoencoder with a graph pooling mechanism to reconstruct electricity demand time series data. By incorporating a pooling block, we constrain the autoencoder to identify groups of nodes that demonstrate similar electricity demands. Then, we are able to extract the learned pooling assignments as our spatial aggregations. In the following section, we elaborate on our approach for spatio-temporal aggregation.

4 GRAPH CONVOLUTIONAL AUTOENCODER APPROACH

Motivated by the ability of deep unsupervised learning methods to extract latent features from high-dimensional data, we introduce the autoencoder modeling paradigm as follows. Given a high-dimensional input such as a multivariate time series, $\boldsymbol{X} \in \mathbb{R}^p$, an autencoder can be trained to jointly learn an encoder, $g : \mathbb{R}^p \to \mathbb{R}^k$, and a decoder, $f : \mathbb{R}^k \to \mathbb{R}^p$ that minimize the reconstruction loss function $\|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_2^2$, where $\hat{\boldsymbol{X}} = f(g(\boldsymbol{X}))$ is the reconstructed input. Here, $k \ll p$ denotes the dimension of the learned latent space.

Recent work on graph representation learning has facilitated the extension of deep unsupervised learning to the graph time series domain. In Sec. 4.1, we introduce relevant methods from modeling with graph neural networks. Then, in Sec. 4.2 and 4.3, we describe our exact autoencoder-based approach for spatio-temporal aggregation.

4.1 Graph Representation Learning

4.1.1 Preliminaries. We encode a graph topology with the binary adjacency[2] matrix $\boldsymbol{A}$, which we construct such that

$$\boldsymbol{A}_{ij} = \begin{cases} 0 & (i,j) \notin \mathcal{E} \\ 1 & (i,j) \in \mathcal{E}. \end{cases}$$

We also construct the diagonal degree matrix $\boldsymbol{D}$ such that $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$.

4.1.2 Graph Convolutions. Our graph autoencoder approach follows [Kipf and Welling 2017] in utilizing *Chebyshev convolutional filters*, which approximate spectral convolutions to learn node embeddings as weighted local averages of embeddings of adjacent nodes. This is ideal for learning low-dimensional embeddings of energy networks as neighborhoods of nodes typically exhibit similar energy demands patterns and can thus be represented jointly (see Fig. 2). Chebyshev filters operate on the "renormalized" graph Laplacian $\tilde{\boldsymbol{L}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}}$, where $\tilde{\boldsymbol{D}} = \boldsymbol{I} + \boldsymbol{D}$ and $\tilde{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{A}$, and

[2] Ideally, one should construct an affinity matrix $\boldsymbol{A}$ with a Gaussian kernel such that $\boldsymbol{A}_{ij} = \exp\left(-\frac{\mathrm{dist}(i,j)^2}{\sigma^2}\right)$ for all edges (i,j), where $\mathrm{dist}(i,j)$ denotes the distance of edge (i,j) and σ denotes the standard deviation of distances in the network [Shuman et al. 2012]. Since we do not have access to edge distance data in our case study, we proceed with the binary adjacency matrix.

perform a form of Laplacian smoothing [Li et al. 2018; Taubin 1995]. We initialize $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ and apply convolutional filters to learn subsequent node embeddings as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma(\tilde{\boldsymbol{L}}\boldsymbol{H}^{(l)}\Theta^{(l)}),$$

where $\Theta^{(l)}$ is a trainable weight matrix and $\boldsymbol{H}^{(l)}$ is a matrix of node embeddings in layer l. $\sigma(\cdot)$ is typically a nonlinear activation function, such as ReLU or tanh.

In each layer, GCNs aggregate features from the immediate neighborhood of each node. Deep GCNs stack multiple layers with nonlinear activations to learn node embeddings as nonlinear functions of both local and global node features. In contrast, [Salha et al. 2019] propose a simpler graph autoencoder model, which they demonstrate to have competitive performances with multilayer GCNs on standard benchmark datasets despite being limited to linear first-order interactions. Shallow neural architectures are also better suited for settings where data is scarce. This is particularly significant in modeling energy systems whose data may only be available for a few historical years. Indeed, we find this shallower GCN approach to perform well for our case study.

4.1.3 Graph Pooling. Motivated by the need for dimensionality reduction and coarsening in graph-level ML tasks, multiple approaches for hierarchical graph pooling have been developed in recent years [Bianchi et al. 2020; Tsitsulin et al. 2020; Ying et al. 2018]. A typical approach to graph pooling is to train a GCN block with a softmax output activation for node cluster assignment jointly with the full GCN. Given k nodes and $k' < k$ desired node clusters, the output of this cluster assignment block is $\boldsymbol{S} \in \mathbb{R}^{k\times k'}$, a matrix of soft assignments in which $\boldsymbol{S}_{ij} \in (0, 1)$ encodes the degree to which node i is assigned to cluster j. To pool the node features after the l-th layer, $\boldsymbol{H}^{(l)}$, we simply compute $\tilde{\boldsymbol{H}}^{(l)} = \boldsymbol{S}^\top \boldsymbol{H}^{(l)}$. Conversely, we can disaggregate the pooled feature matrix by computing $\boldsymbol{S}\tilde{\boldsymbol{H}}^{(l)}$.

Intuitively, one can imagine this block as taking in a time series of electricity demands for each node and returning a "predicted" membership distribution for each node over the k' clusters. Although there is no observed membership data with which to verify node cluster assignments, the quality of the assignments is evaluated implicitly by the downstream reconstruction error, and consequently the reconstruction loss is backpropagated to the pooling block during training of the autoencoder. To minimize reconstruction loss, the pooling mechanism will learn to pool nodes that exhibit similar electricity demands in each day.

For our application, we utilize the MinCutPool operator proposed by [Bianchi et al. 2020], which corresponds to incorporating the objective

$$\min_{\boldsymbol{S}} \mathcal{L}_u = \min_{\boldsymbol{S}} \underbrace{-\frac{\mathrm{Tr}(\boldsymbol{S}^\top \tilde{\boldsymbol{A}}\boldsymbol{S})}{\mathrm{Tr}(\boldsymbol{S}^\top \tilde{\boldsymbol{D}}\boldsymbol{S})}}_{\mathcal{L}_c} + \underbrace{\left\| \frac{\boldsymbol{S}^\top \boldsymbol{S}}{\|\boldsymbol{S}^\top \boldsymbol{S}\|_F} - \frac{\boldsymbol{I}_{k'}}{\sqrt{k'}} \right\|_F}_{\mathcal{L}_o}.$$

into the autoencoder loss function. This MinCutPool objective $\mathcal{L}_u$ is the sum of the *cut loss*, $\mathcal{L}_c$, and the *orthogonality loss*, $\mathcal{L}_o$. Minimizing the cut loss yields a cluster assignment that groups strongly connected nodes, while minimizing the orthogonality loss yields cluster assignments that are orthogonal and of similar sizes, i.e., each node is fully assigned to one cluster. The orthogonality loss term is included to discourage convergence to degenerate minima of the cut loss such as the uniform assignment of all nodes to all clusters. Consequently, to train an autoencoder with a pooling mechanism, we can minimize a weighted sum of the reconstruction loss $\mathcal{L}_r$ and the MinCutPool loss $\mathcal{L}_u$.

4.2 Temporal Aggregation Architecture

We will first present the proposed GCN architecture for temporal aggregation, which is illustrated in Fig. 3.

Variable	Interpretation	Granularity	Nodes
$\boldsymbol{X}_E$	Electricity	Hourly	88
$\boldsymbol{X}_W$	Wind	Hourly	88
$\boldsymbol{X}_S$	Solar	Hourly	88
$\boldsymbol{X}_G$	Natural Gas	Daily	18

Table 1. Notation for input variables.

We denote by $\boldsymbol{X}_E \in \mathbb{R}^{d\times n_E\times t_E}$ the data tensor of electricity demands over all days d, nodes n_E, and times t_E. Similarly, we denote the natural gas data tensor by $\boldsymbol{X}_G \in \mathbb{R}^{d\times n_G\times t_G}$, the wind capacity factor tensor by $\boldsymbol{X}_W \in \mathbb{R}^{d\times n_W\times t_W}$, and the solar capacity factor data tensor by $\boldsymbol{X}_S \in \mathbb{R}^{d\times n_S\times t_S}$ (see Table 1). Because the GTEP considers different associated costs for investment and operational decisions related to power, NG, wind, and solar, we introduce hyperparameters $\alpha_G, \alpha_W, \alpha_S$ in the autoencoder objective function to tune the trade-off between the multiple reconstruction losses. This parameter reflects the contribution of each system towards the total cost. For example, if the NG system cost is twice the power system cost, then higher values of α_G ensure that the reconstruction cost is penalized more when deviating from the data of the NG system. This gives us the following reconstruction loss function:

$$\mathcal{L}_r = \sum_{i=1}^{d} \Big(\frac{1}{dn_E t_E} \|\boldsymbol{X}_E^{(i)} - \hat{\boldsymbol{X}}_E^{(i)}\|_F^2 + \frac{\alpha_G}{dn_G t_G} \|\boldsymbol{X}_G^{(i)} - \hat{\boldsymbol{X}}_G^{(i)}\|_F^2 + \frac{\alpha_W}{dn_W t_W} \|\boldsymbol{X}_W^{(i)} - \hat{\boldsymbol{X}}_W^{(i)}\|_F^2 + \frac{\alpha_S}{dn_S t_S} \|\boldsymbol{X}_S^{(i)} - \hat{\boldsymbol{X}}_S^{(i)}\|_F^2 \Big),$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

In our case study, we set $\alpha_G = 2, \alpha_S = 0.5, \alpha_W = 0.5$. However, we note that it is possible to choose the hyperparameters by evaluating the downstream GTEP objective for different values. Specifically, this can be performed using a grid search in which the quality of a combination of hyperparameters $\{\alpha_G, \alpha_W, \alpha_S\}$ is measured by GTEP objective costs given by solving the optimization model rather than the autoencoder validation loss directly.

4.2.1 Encoder. We begin by constructing the data matrix $\boldsymbol{X}^{(i)}$ corresponding to day (i) as

$$\boldsymbol{X}^{(i)} = \begin{pmatrix} \boldsymbol{X}_E^{(i)} & \boldsymbol{X}_W^{(i)} & \boldsymbol{X}_S^{(i)} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{X}_G^{(i)} \end{pmatrix}.$$

Note that $\boldsymbol{X}^{(i)} \in \mathbb{R}^{n\times t}$, where $n := n_E + n_G$ and $t := t_E + t_W + t_S + t_G$. This is because capacity factor data exists for all nodes in the power network and utilizes the same network topology. $\boldsymbol{X}^{(i)}$ is then passed

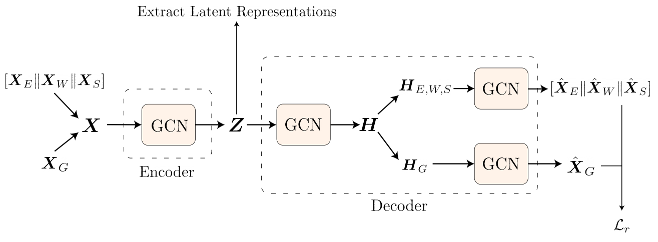


Fig. 3. The temporal aggregation autoencoder aims to learn latent embeddings as a preprocessing step for clustering the combined electric power, wind CF, solar CF, and NG time series corresponding to different days.

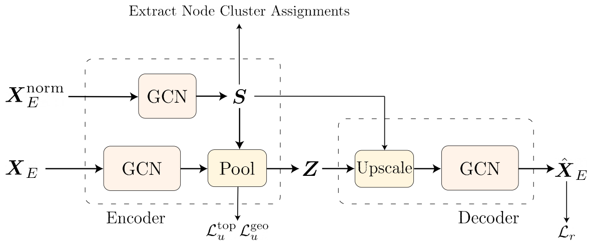


Fig. 4. The spatial aggregation autoencoder utilizes a pooling mechanism to cluster nodes based on their electricity demands.

through a GCN block to produce the low-dimensional embedding $\boldsymbol{Z}^{(i)} \in \mathbb{R}^{n\times k}$. The hyperparameter k defines the bottleneck of the autoencoder architecture (i.e. the dimension of each node embedding) and consequently the tradeoff between compression and reconstruction loss.

4.2.2 *Decoder.* $\boldsymbol{Z}^{(i)}$ is passed through a GCN block to produce the embedding $\boldsymbol{H}^{(i)} \in \mathbb{R}^{n\times t}$. This reconstructed matrix is then split along the second dimension into two tensors: $\boldsymbol{H}^{(i)}_{E,W,S} \in \mathbb{R}^{(n_E+n_W+n_S)\times t}$ and $\boldsymbol{H}^{(i)}_G \in \mathbb{R}^{n_G\times t}$. Each tensor is then passed to a separate GCN block that maps the node embeddings in $\boldsymbol{H}^{(i)}_{E,W,S}$ and $\boldsymbol{H}^{(i)}_G$ respectively to the reconstructions $\hat{\boldsymbol{X}}^{(i)}_{E,W,S}$ and $\hat{\boldsymbol{X}}^{(i)}_G$. Finally, the tensor $\hat{\boldsymbol{X}}^{(i)}_{E,W,S}$ is split into the reconstructions $\hat{\boldsymbol{X}}^{(i)}_S, \hat{\boldsymbol{X}}^{(i)}_W, \hat{\boldsymbol{X}}^{(i)}_S$.

In our case study, we find utilizing shallow GCN blocks (one or two convolutional layers followed by a tanh activation) and setting $k = 3$ to yield a sufficient performance for our application of temporal aggregation.

4.3 Spatial Aggregation Architecture

The spatial aggregation approach differs from the temporal aggregation approach in four ways: (1) a GCN block with a pooling operator such as MinCutPool is trained jointly as part of the autoencoder to aggregate nodes; (2) a clustering algorithm is not applied to the latent embeddings; rather, the spatial aggregations are directly output by the graph pooling block in the encoder; (3) the loss that is minimized is a weighted sum of the reconstruction loss with the MinCutPool objective; (4) the autoencoder is trained using only the electricity demand data tensor $\boldsymbol{X}_E$ since the goal of the spatial aggregation is to identify groups of nodes in the power network with similar demand behaviors (and consequently, similar constraints).

4.3.1 *Encoder.* The encoder consists of two GCN blocks in parallel that feed into a pooling mechanism. In constructing the autoencoder architecture, the hyperparameter $k' < k$ is specified as the desired number of node clusters (i.e., nodes in the spatially aggregated formulation). Then for a given day i, the first block extracts low-dimensional features from the electricity time series $\boldsymbol{X}^{(i)}_E$ while the second block outputs the node-cluster assignments $\boldsymbol{S}^{(i)} \in (0,1)^{k\times k'}$ given the normalized electricity time series. This normalization is performed to prevent the autoencoder from greedily assigning nodes to clusters based only on the magnitudes of the electricity demands; otherwise, the pooling block might assign all nodes to one cluster corresponding to days of uniformly low demand (e.g., days in winter) or another cluster corresponding to days of uniformly high demand etc. One can also include node-specific one-hot features in the normalized time series matrix, which we find to greatly increase the performance of the pooling mechanism.

Given the output of the former GCN block, $\boldsymbol{H}^{(i)} \in \mathbb{R}^{k\times\eta}$, and the cluster assignment matrix $\boldsymbol{S}^{(i)}$ output from the latter GCN block, the pooled feature matrix $\boldsymbol{Z}^{(i)} = \boldsymbol{S}^{(i)^\top}\boldsymbol{H}^{(i)}$ is passed to the decoder along with $\boldsymbol{S}^{(i)}$.

4.3.2 *Decoder.* To reconstruct the input, the decoder upscales the pooled feature matrix by computing $\boldsymbol{S}^{(i)}\boldsymbol{Z}^{(i)} \in \mathbb{R}^{k\times\eta}$. This matrix is then passed to a final GCN block, which outputs the reconstructed electricity demand time series $\hat{\boldsymbol{X}}^{(i)}_E$.

4.3.3 *Geospatial and Graph Topological Clustering.* It should be noted that pooling assignments that minimize the MinCutPool loss for the graph adjacency matrix occasionally assign geographically distant nodes to the same cluster. This property may be undesirable for CEP applications as geographic distance is usually regarded in conventional approaches to spatial aggregation. To resolve this, we introduce an additional objective to the loss function to penalize cluster assignments that span large geographic distances. Specifically, we construct an additional affinity matrix $\boldsymbol{A}^{\text{geo}}$ with a Gaussian kernel such that $\boldsymbol{A}^{\text{geo}}_{ij}$ encodes the geographic distance between node i and node j. Given $\boldsymbol{A}^{\text{geo}}$ and its corresponding degree matrix, we can pass a learned assignment matrix $\boldsymbol{S}$ through a MinCutPool operator to yield $\mathcal{L}^{\text{geo}}_u$, or the MinCutPool loss with respect to geographic affinities. We distinguish this loss from $\mathcal{L}^{\text{top}}_u$, or the MinCutPool loss with respect to graph topological affinities, and train the autoencoder to minimize the objective

$$\min \mathcal{L}_r + \frac{1}{2}(\mathcal{L}^{\text{top}}_u + \mathcal{L}^{\text{geo}}_u),$$

where

$$\mathcal{L}_r = \sum_{i=1}^{d} \frac{1}{dn_E t_E} \|\boldsymbol{X}^{(i)}_E - \hat{\boldsymbol{X}}^{(i)}_S\|^2_F.$$

4.3.4 *Retrieving Spatial Aggregations.* Once the autoencoder has been trained, d spatial aggregations can be retrieved by passing each of the d days through the pooling GCN block. From here, a single spatial aggregation can be chosen from the d candidates (many of which may be identical). One approach, which we chose for our case study, is to assign each node's respective cluster by taking a "vote" of cluster memberships over the d days. Another valid approach would be to weight the vote based on the learned temporal aggregation.

5 INPUT DATA

We used the input data provided in [Brenner et al. 2022]. The authors construct the New England power and NG networks by publicly available data. The power network consists of 88 nodes, located in 88 distinct locations with 338 existing and candidate transmission lines. The NG network consists of 18 NG nodes and 7 storage nodes. Each NG node is connected to two other storage nodes. Also, each power node is connected to three of its closest NG nodes. The input data consist of 12 power plants types, 5 of which existing types (e.g., "ng" which is existing gas-fired plants) and the 7 remaining are new types (e.g., CC, wind-offshore, and solar-UPV). The SI [SI 2022] provides the details of the input data for the joint power-NG planning model [SI 2022].

6 UPPER BOUND SOLUTION

To evaluate each instance of the spatio-temporal aggregation, we propose a heuristic algorithm that generates a feasible solution for the full problem (88 nodes and the entire year). The algorithm consist of three steps. The first step solves the aggregated problem and retrieves investment decisions for each spatial cluster. The second step considers the full network for only 2 days and adds sets of constraints to ensure that the sum of investment decisions over all nodes represented by a cluster equals the number of investment decisions for the cluster in the first step. For example, suppose that 5 solar plants are established in the first step for node 7, and assume that nodes 3, 6, and 44 are represented by cluster 7. Then in the second step, we set the sum of established solar plants over nodes 3, 6, and 44 to 5. The result of the second step provides investment decisions for each node. We consider the full network over the whole year in the third step and use these values instead of investment decisions. This renders the problem as a linear program with an optimal solution because the problem is feasible for any investment decisions. The solution to the third step is a feasible solution to the original problem, and consequently is an upper bound (UB).

7 RESULTS AND DISCUSSION

We consider network sizes of 6, 10, 15, and 20 for our computational study and experiment with all even representative days between 4 and 30. All instances are implemented in Python using Gurobi 9.5 and run on the MIT Supercloud system, which uses an Intel Xeon Platinum 8260 processor with up to 48 cores and 192 GB of RAM [Reuther et al. 2018]. We limit the CPU time to 3 hours for all instances, by which point all models were solved to optimality with a mixed integer gap of 1% or lower.

Fig. 6 illustrates the impact of different spatio-temporal aggregations on the total cost and various cost components. As stipulated in Section 2, the total cost consist of power and NG system's cost. The total cost of each system is comprised of different cost components including investment, fixed operation and maintenance (FOM), variable operation and maintenance (VOM) costs. Fig. 6 illustrates the change in cost components with respect to different aggregations as heatmaps (brighter colors correspond to higher costs). The horizontal axis represents the network size (spatial aggregation) while the vertical axis conveys the number of representative days (temporal aggregation). We observe that, for a given number of representative days, the total system cost increases monotonically with the network size. However, there is no obvious trend in total cost with regard to the number of representative days. The increasing total cost for larger network sizes can be explained by increased congestion in the network. Larger networks necessitate more transmission as establishing new plants, especially gas-fired plants, provides a higher capacity than the node demands. However, a large transmission of power from a node to another may not be possible due to network congestion.

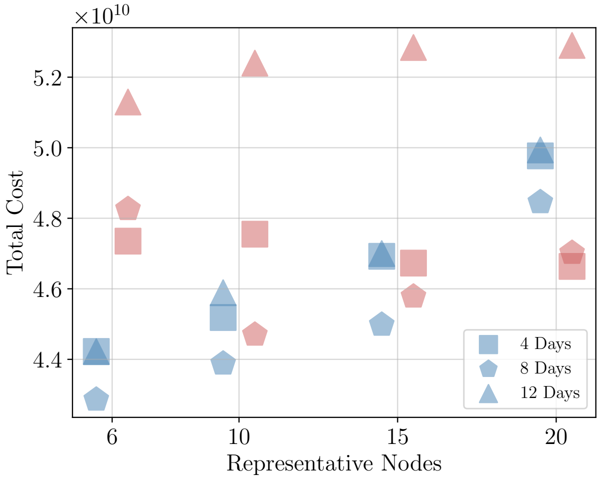


Fig. 5. The UB and objective values associated with each aggregation instance. For each value in the horizontal axis, the left hand side markers show the objective value for the aggregation instance while the markers on the right hand side represent the corresponding UB.

As expected, the NG system's cost is almost agnostic towards the power network size. However, its variability is significant with respect to the number of representative days. For example, the NG cost for 8 days is almost twice the NG cost for 10 days. The total power system cost demonstrates a pattern similar to that of the total cost, indicating that the behavior of the total cost is largely driven by the power system cost. We observe that establishment cost increases with the network size for all representative days. A similar pattern can be observed for FOM. The VOM cost, however, sharply decreases as the network size exceeds 10. The increased establishment cost and reduction of VOM indicate that larger power network sizes tend to dispatch lower firm energies and greater renewable energies. Firm energies refer to plant types whose outputs can be changed to compensate for the load variability. Gas-fired and nuclear plants are examples of firm energies and incur variable costs, but their establishment cost is relatively low. Renewable energies such as solar and wind, in contrast, do not incur variable costs but require higher investments to provide sufficient generation.

We evaluate the UB heuristic on 4, 8, and 12 days over various network sizes and illustrate the results in Fig. 5. For each network

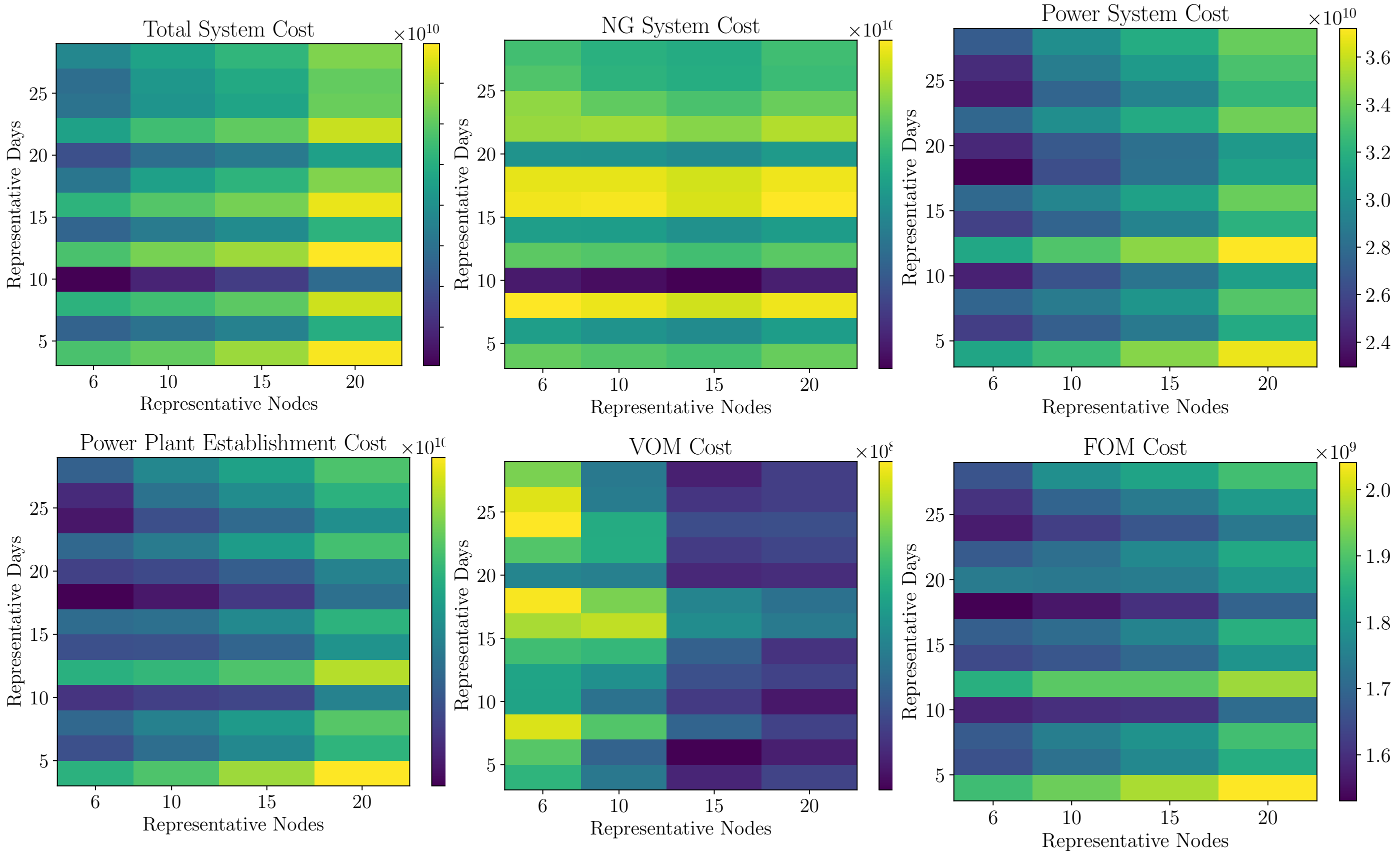


Fig. 6. Some system costs, such as VOM cost, demonstrate an obvious relationship with the resolution of spatio-temporal aggregations, while others, such as the NG system cost, do not exhibit such a clear relationship.

size, the blue markers on the left hand side correspond to the objective values for the aggregated problems while the red markers on the right hand side correspond to the UB values. Depending on the aggregation, the objective value for each aggregation can be lower or higher than the optimal objective for the original problem. For most instances, the UB yields a higher objective than that of the aggregation. However, for a network size of 15 nodes and using 4 representative days, the UB is slightly lower. This may be associated with the specific choice of representative days. If the selected days favors extreme days (i.e., days exhibiting significantly higher load than average), then the objective value of the aggregated problem may exceed the UB.

Interestingly, the analysis of the spatio-temporal aggregation results demonstrates that the *quality* of the solution does not necessarily increase with the resolution of the spatial or temporal aggregation. This is an important observation as it suggest that CEP models could be solved for highly aggregated instances without significantly compromising the solution fidelity. For instance, the best UB value is obtained in 10-node network for 8 representative days as it is shown in Fig. 5. It is also worth mentioning that, although some cost components demonstrate a clear relationship with spatio-temporal aggregations, many appear to have no discernible relationship to system costs, e.g., the seemingly random and volatile response of total system cost with regard to the number of representative days (see Fig. 6). To this end, additional investigation would be needed to understand to what degree these results are specific to our formulation and choice of aggregations as opposed to the GTEP in general.

8 CONCLUSION

In this paper, we present a graph convolutional autoencoder approach to spatio-temporal aggregation for capacity expansion problems and apply our approach to a generation transmission and expansion problem for the New England joint power-NG network. Specifically, we present two autoencoder architectures: an architecture for temporal aggregation that utilizes electricity demand, NG demand, wind CF, and solar CF data, and an architecture for spatial aggregation that uses electricity demand data to learn a node cluster assignment via graph pooling. Finally, we interpret the sensitivity of various cost components with respect to spatial and temporal aggregation resolution given a set of aggregations learned by our approach. Our analysis highlights the importance of data-driven aggregation methods and show that some highly aggregated model could provide quality solutions for capacity expansion problems.

Future studies can extend the results, methodology and observations made in this paper in several directions. For one, additional experiments can be performed with different CEP formulations to more comprehensively understand the relationship connecting aggregation resolution to both solution quality and sensitivity of the learned planning decisions. Moreover, the sensitivity of aggregations, and consequently planning decisions, with respect to changes in the dataset should be investigated. The ML technique can be tailored to reflect the characteristics of each CEP model considered.

REFERENCES

Marc Barbar and Dharik S Mallapragada. 2022. Representative period selection for power system planning using autoencoder-based dimensionality reduction. *arXiv preprint arXiv:2204.13608* (2022).

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. 2020. Spectral Clustering with Graph Neural Networks for Graph Pooling. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 82, 10 pages.

Aron Brenner, Rahman Khorramfar, Dharik Mallapragada, and Saurabh Amin. 2022. Graph Representation Learning for Energy Demand Data: Application to Joint Energy System Planning under Emissions Constraints. *arXiv preprint arXiv:2209.12035* (2022).

Energy Information Administration (EIA). 2022. EIA Website. Website. https://www.eia.gov Accessed: 2022-2-18.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2001. *The Elements of Statistical Learning*. Springer New York Inc., New York, NY, USA.

Chuan He, Xiaping Zhang, Tianqi Liu, Lei Wu, and Mohammad Shahidehpour. 2018. Coordination of interdependent electricity grid and natural gas network—a review. *Current Sustainable/Renewable Energy Reports* 5, 1 (2018), 23–36.

Maximilian Hoffmann, Leander Kotzur, Detlef Stolten, and Martin Robinius. 2020. A review on time series aggregation methods for energy system models. *Energies* 13, 3 (2020), 641.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SJU4ayYgl

Can Li, Antonio J Conejo, Peng Liu, Benjamin P Omell, John D Siirola, and Ignacio E Grossmann. 2022a. Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems. *European Journal of Operational Research* 297, 3 (2022), 1071–1082.

Can Li, Antonio J Conejo, John D Siirola, and Ignacio E Grossmann. 2022b. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. *International Journal of Electrical Power & Energy Systems* 137 (2022), 107697.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (New Orleans, Louisiana, USA) *(AAAI'18/IAAI'18/EAAI'18)*. AAAI Press, Article 433, 8 pages.

Yixian Liu, Ramteen Sioshansi, and Antonio J Conejo. 2017. Hierarchical clustering to find representative operating periods for capacity-expansion modeling. *IEEE Transactions on Power Systems* 33, 3 (2017), 3029–3039.

Hanan Luss. 1982. Operations Research and Capacity Expansion Problems: A Survey. *Oper. Res.* 30 (1982), 907–947.

Dharik S Mallapragada, Dimitri J Papageorgiou, Aranya Venkatesh, Cristiana L Lara, and Ignacio E Grossmann. 2018. Impact of model resolution on scenario outcomes for electricity sector system expansion. *Energy* 163 (2018), 1231–1244.

Lance Parsons, Ehtesham Haque, and Huan Liu. 2004. Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explor. Newsl.* 6, 1 (jun 2004), 90–105. https://doi.org/10.1145/1007730.1007731

Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, et al. 2018. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE, 1–6.

Guillaume Salha, Romain Hennequin, and Michalis Vazirgiannis. 2019. Keep it simple: Graph autoencoders without graph convolutional networks. *arXiv preprint arXiv:1910.00942* (2019).

Ian J Scott, Pedro MS Carvalho, Audun Botterud, and Carlos A Silva. 2019. Clustering representative days for power systems generation expansion planning: Capturing the effects of variable renewables and energy storage. *Applied Energy* 253 (2019), 113603.

David Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2012. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Processing Magazine* 30 (10 2012).

SI. 2022. Supplementary material available at: shorturl.at/nK047. shorturl.at/nK047

Gabriel Taubin. 1995. A Signal Processing Approach to Fair Surface Design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. Association for Computing Machinery, New York, NY, USA, 351–358. https://doi.org/10.1145/218380.218473

Holger Teichgraeber and Adam R Brandt. 2019. Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison. *Applied energy* 239 (2019), 1283–1293.

Holger Teichgraeber and Adam R Brandt. 2022. Time-series aggregation for the optimization of energy systems: Goals, challenges, approaches, and opportunities. *Renewable and Sustainable Energy Reviews* 157 (2022), 111984.

Anton Tsitsulin, John Palowitch, and Bryan Perozzi. 2020. Graph Clustering with Graph Neural Networks. https://arxiv.org/abs/2006.16904

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 4805–4815.

Supplementary information:

Learning Spatio-Temporal Aggregations for Large-Scale Capacity Expansion Problems

Aron Brenner[1], Rahman Khorramfar[2], Saurabh Amin[3]

[1],[3] Civil and Environmental Engineering (CEE) and Laboratory for Information & Decision Systems (LIDS)
[2] MIT Energy Initiative (MITEI) and Laboratory for Information & Decision Systems (LIDS)
{khorram, abrenner, amins, dharik}@mit.edu

1 Model Formulation

Our formulation is based on the formulation proposed in [1], but we applied a set of assumption to simplify the model and speed up the running time. The model determines minimum cost investment and operational decision for power and NG system. across a set of representative periods. The formulation allows different temporal resolutions for the operation of both systems as data availability or planning requirements can be different for power and NG systems. The operations of both systems are coupled through two sets of constraints. The first set ensure NG flow to the power system. The second coupling constraints limit the CO_2 emission incurred by consuming NG in both power and NG systems.

The network consists of three sets of nodes as depicted in Figure 1. The first set represents power system nodes and is characterized by different generation technologies (plant types), demand, storage, and the set of adjacent nodes. The second set of nodes are NG nodes each of which associated with injection amount, demand, and its adjacent nodes. Storage tanks, vaporization and liquefaction facilities, which are commonly used in the non-reservoir storage of NG, collectively form the third set of nodes namely SVL nodes. The model also consider the *renewable natural gas* (RNG) which is type of net-zero biofuel fully interchangeable with NG and hence can be imported and transported by the NG pipelines [2]. Details regarding input data including generation plant and

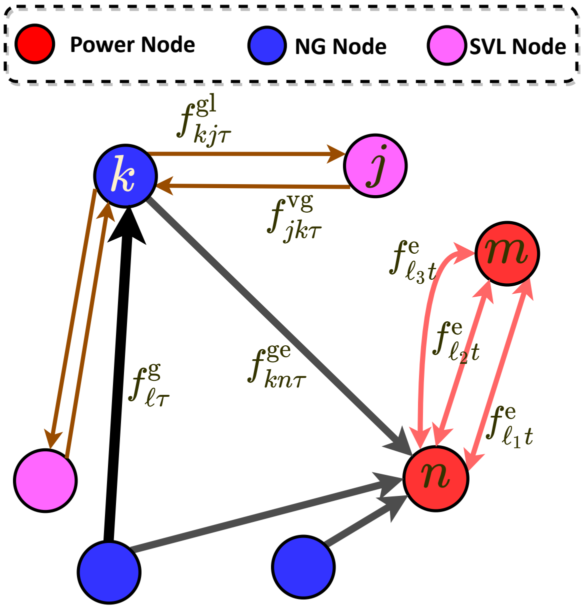

Power Node
NG Node
SVL Node
$f^{\mathrm{gl}}_{kj\tau}$
k
j
$f^{\mathrm{vg}}_{jk\tau}$
$f^{\mathrm{e}}_{\ell_3 t}$
m
$f^{\mathrm{e}}_{\ell_2 t}$
$f^{\mathrm{ge}}_{kn\tau}$
$f^{\mathrm{g}}_{\ell\tau}$
$f^{\mathrm{e}}_{\ell_1 t}$
n

Figure 1: Power nodes can be connected by multiple bi-directional transmission lines denoted by $f^{\mathrm{e}}_{\ell t}$. Each power operate local NG-fired plants by drawing gas from nodes that are connected to it. The variable $f^{\mathrm{ge}}_{\ell\tau}$ captures this flow. Each NG node is connected to its adjacent SVL nodes through two unidirectional pipelines where one is from NG to SVL's liquefaction facilities denoted by $f^{\mathrm{gl}}_{kj\tau}$; and the other one from SVL's vaporization facility to NG node denoted by $f^{\mathrm{vg}}_{jk\tau}$. The variable $f^{\mathrm{g}}_{\ell\tau}$ denotes the flow between NG nodes. Similar to the power system, NG nodes can be connected by one or more uni-directional pipelines, but only one connection is depicted here. Candidate transmission lines and pipelines are not shown in this figure.

storage types, demand, and cost assumptions are provided in the Appendix of Supplementary Information of [1].

Indices	
n, m	Power system node
k	NG system node
j	SVL facility node
i	Power generation plant type
r	Storage type for power network
ℓ	Transmission line or pipeline
t	Time step for power system planning
τ	Time step for NG system planning

Sets	
$\mathcal{N}^{\mathrm{e}}$	Power system nodes
$\mathcal{P}$	Power plant types
$\mathcal{R} \subset \mathcal{P}$	VRE power plant types
$\mathcal{G} \subset \mathcal{P}$	gas-fired plant types
$\mathcal{H} \subset \mathcal{P}$	Thermal plant types
$\mathcal{T}^{\mathrm{e}}$	Representative hours for power system
$\mathfrak{R}$	Representative days
$\mathfrak{T}^{\mathrm{e}}_{\tau}$	Hours in the representative day τ
$\mathcal{L}^{\mathrm{e}}$	Existing and candidate transmission lines
$\mathcal{S}^{\mathrm{e}}_{n}$	Storage facility types
$\mathcal{A}^{\mathrm{g}}_{n}$	Adjacent NG nodes for node n
$\mathcal{L}^{\mathrm{e}}_{nm}$	Existing and candidate transmission lines between node n and m
$\mathcal{N}^{\mathrm{g}}, \mathcal{N}^{s}$	NG and SVL nodes
$\mathcal{T}^{\mathrm{g}}$	Days of the planning year for NG system
$\mathcal{A}^{s}_{k}$	Adjacent SVL facilities of node k
$\mathcal{L}^{\mathrm{g}}$	Existing and candidate pipelines
$\mathcal{L}^{\mathrm{gExp}}_{k}$	Existing and candidate pipelines starting from node k
$\mathcal{L}^{\mathrm{gImp}}_{k}$	Existing and candidate pipelines ending at node k

Annualized Cost Parameters	
C^{inv}_{i}	CAPEX of plants, [\$/plant]
C^{dec}_{i}	Plant decommissioning cost, [\$/plant]
$C^{\mathrm{trans}}_{\ell}$	Transmission line establishment cost, [\$/line]
C^{EnInv}_{r}	Storage establishment energy-related cost, [\$/MWh]
C^{pInv}_{r}	Storage establishment power-related cost, [\$/MW]
C^{pipe}_{ℓ}	Pipelines establishment cost, [\$/line]
C^{strInv}_{j}	CAPEX of storage tanks at SVLs, [\$/MMBtu]
C^{vprInv}_{j}	CAPEX of vapor. plants at SVLs, [\$/MMBtu]

Annual Costs

C_i^{fix}	FOM for plants, [\$]
C_r^{EnFix}	Energy-related FOM for storage, [\$/MWh]
C_r^{pFix}	Power-related FOM for storage, [\$/MW]
C_j^{strFix}	FOM for storage tanks, [\$/MMBtu]
C_j^{vprFix}	FOM for vaporization plants, [\$/MMBtu]

Other Cost Parameters

C_i^{var}	VOM for plants, [\$/MWh]
C^{eShed}	Unsatisfied power demand cost, [\$/MWh]
C_i^{fuel}	Fuel price for plants, [\$/MMBtu]
C^{ng}	Fuel price for NG, [\$/MMBtu]
C^{rng}	Price of RNG, [\$/MMBtu]
C^{gShed}	Unsatisfied NG demand cost [\$/MMBtu]

Other Parameters for the Power System

ρ_{nti}	Capacity factor
D_{nt}^{e}	Power demand, [MWh]
h_i	Heat rate, [MMBtu/MWh]
η_i	Carbon capture rate, [%]
U_i^{prod}	Nameplate capacity, [MW]
L_i^{prod}	Minimum stable output, [%]
U_i^{ramp}	Ramping limit, [%]
$\gamma_r^{\text{eCh}}, \gamma_r^{\text{eDis}}$	Charge/discharge rate for storage
I_ℓ^{trans}	Initial capacity for transmission line ℓ, [MW]
U_ℓ^{trans}	Upper bound for transmission line ℓ, [MW]
$\mathcal{I}_\ell^{\text{trans}}$	1, if trans. line ℓ exists; 0, otherwise
I_{ni}^{num}	Initial number of plants
$U_{\text{emis}}^{\text{e}}$	Baseline emission of CO_2 in 1990 from generation consumption, [ton]
L^{RPS}	Renewable Portfolio Standard (RPS) value
$\mathfrak{D}_n$	Distance between node n and CO_2 storage site
E^{pipe}	Electric requirement for CO_2 pipeline operations [MWh/mile/ton/hour]
E^{pump}	Electric requirement for compression of CO_2
E^{cprs}	Number of compressors required in the pipeline from node n to the storage site pipelines [MWh/ton/hour]
$U_{\mathcal{Q}}^{\text{prod}}$	Production capacity for set of plants $\mathcal{Q} \subset \mathcal{P}$, [MW]
ζ	Emission reduction goal
w_t	Number of periods represented by period t
ϕ_t^{e}	Mapping of representative period t to its original period in the time series

Other Parameters for the NG System

$D^{\mathrm{g}}_{k\tau}$	NG demand, [MMBtu]
η^{g}	Emission factor for NG [ton $\mathrm{CO_2}$/MMBtu]
U^{inj}_{k}	Upper bound for NG supply, [MMBtu]
$\gamma^{\mathrm{liqCh}}_{j}$	Charge efficiency of liquefaction plant
$\gamma^{\mathrm{vprDis}}_{j}$	Discharge efficiency of vaporization plant
β	Boil-off gas coefficient
I^{pipe}_{ℓ}	Initial capacity for pipeline ℓ, [MMBtu]
U^{pipe}_{ℓ}	Upper bound capacity for pipeline ℓ, [MMBtu]
$\mathcal{I}^{\mathrm{pipe}}_{\ell}$	1, if the pipeline ℓ exists; 0, otherwise
I^{gStr}_{j}	Initial storage capacity, [MMBtu]
I^{vpr}_{j}	Initial vaporization capacity, [MMBtu/d]
I^{liq}_{j}	Initial liquefaction capacity, [MMBtu/d]
I^{store}_{kj}	Initial capacity of storage facility
$U^{\mathrm{g}}_{\mathrm{emis}}$	Baseline emission of $\mathrm{CO_2}$ in 1990 from non-generation consumption, [ton]
Ω_{τ}	Set of days represented by day $\tau \in \mathfrak{R}$

Investment Decision Variables

$x^{\mathrm{op}}_{ni} \in \mathbb{Z}^{+}$	Number of available plants
$x^{\mathrm{est}}_{ni} \in \mathbb{Z}^{+}$	Number of new plants established
$x^{\mathrm{dec}}_{ni} \in \mathbb{Z}^{+}$	Number decommissioned plants
$y^{\mathrm{eCD}}_{nr} \in \mathbb{R}^{+}$	Charge/discharge capacity of storage battery
$y^{\mathrm{eLev}}_{nr} \in \mathbb{R}^{+}$	Battery storage level
$z^{\mathrm{e}}_{\ell} \in \mathbb{B}$	1, if transmission line ℓ is built; 0, otherwise
$z^{\mathrm{g}}_{\ell} \in \mathbb{B}$	1, if pipeline ℓ is built; 0, otherwise

1.1 Power System Model

Objective Function:

$$\min \sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{i\in\mathcal{P}}(C_i^{\mathrm{inv}}x_{ni}^{\mathrm{est}}+C_i^{\mathrm{fix}}x_{ni}^{\mathrm{op}}+\sum_{r\in\mathcal{S}_n^{\mathrm{e}}}(C_r^{\mathrm{pInv}}+C^{\mathrm{pFix}})y_{nr}^{\mathrm{eCD}})+$$

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{r\in\mathcal{S}_n^{\mathrm{e}}}(C_r^{\mathrm{EnInv}}+C^{\mathrm{EnFix}})y_{nr}^{\mathrm{eLev}}+ \tag{1a}$$

$$\sum_{n\mathcal{N}^{\mathrm{e}}}\sum_{i\in\mathcal{P}}C_i^{\mathrm{dec}}x_{ni}^{\mathrm{dec}} \tag{1b}$$

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{i\in\mathcal{P}}\sum_{t\in\mathcal{T}^{\mathrm{e}}}w_t p_{nti}C_i^{\mathrm{var}}+ \tag{1c}$$

$$\sum_{l\in\mathcal{L}^{\mathrm{e}}}C_\ell^{\mathrm{trans}}z_\ell^{\mathrm{e}}+ \tag{1d}$$

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}}\mathfrak{D}_n C_{\mathrm{CO_2}}^{\mathrm{inv}}\kappa_n^{\mathrm{pipe}}+C_{\mathrm{CO_2}}^{\mathrm{str}}\sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{t\in\mathcal{T}^{\mathrm{e}}}w_t\kappa_{nt}^{\mathrm{capt}}+ \tag{1e}$$

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{i\in\mathcal{P}}\sum_{t\in\mathcal{T}^{\mathrm{e}}}w_t p_{nti}(C_i^{\mathrm{fuel}}h_i)+ \tag{1f}$$

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}}\sum_{t\in\mathcal{T}^{\mathrm{e}}}w_t C_n^{\mathrm{eShed}}a_{nt}^{\mathrm{e}}+ \tag{1g}$$

Other Decision Variables for Power System

$p_{nti} \in \mathbb{R}^+$	Generation rate, [MW]
$f^{\mathrm{e}}_{\ell t} \in \mathbb{R}$	Flow rates, [MW]
$s^{\mathrm{eCh}}_{ntr}, s^{\mathrm{eDis}}_{ntr} \in \mathbb{R}^+$	Storage charged/discharged, [MW]
$s^{\mathrm{eLev}}_{ntr} \in \mathbb{R}^+$	Storage level, [MWh]
$\kappa^{\mathrm{capt}}_{nt} \in \mathbb{R}^+$	Captured $\mathrm{CO_2}$ [ton/h]
$\kappa^{\mathrm{pipe}}_{n} \in \mathbb{R}^+$	$\mathrm{CO_2}$ pipeline capacity [ton/h]
$a^{\mathrm{e}}_{nt} \in \mathbb{R}^+$	Amount of load shedding, [MWh]
$\mathcal{E}^{\mathrm{e}}$	Total emission from power system

Other Decision Variables for NG System (all in MMBtu)

$x^{\mathrm{gStr}}_{j} \in \mathbb{R}^+$	Installed additional storage capacities
$x^{\mathrm{vpr}}_{j} \in \mathbb{R}^+$	Installed additional vaporization capacities
$f^{\mathrm{g}}_{\ell\tau} \in \mathbb{R}^+$	Flow rates
$f^{\mathrm{ge}}_{kn\tau} \in \mathbb{R}^+$	Flow rates from NG nodes to power nodes
$f^{\mathrm{gl}}_{kj\tau} \in \mathbb{R}^+$	Flow rates from node NG nodes to liquefaction plants
$f^{\mathrm{vg}}_{jk\tau} \in \mathbb{R}^+$	Flow rates from vaporization plants to NG nodes
$g_{k\tau} \in \mathbb{R}^+$	NG supply (injection)
$s^{\mathrm{gStr}}_{j\tau} \in \mathbb{R}^+$	Storage capacities
$s^{\mathrm{vpr}}_{j\tau}, s^{\mathrm{liq}}_{j\tau} \in \mathbb{R}^+$	Vaporization and liquefaction amounts
$a^{\mathrm{g}}_{k\tau} \in \mathbb{R}^+$	Amount of load shedding
$a^{\mathrm{rng}}_{k\tau} \in \mathbb{R}^+$	Amount of RNG consumption
$\mathcal{E}^{\mathrm{g}}$	Total emission from NG system

The objective function (1) minimizes the total investment and operating costs incurred in power system. The first term (1a) is the investment and fixed operation and maintenance (FOM) costs for generation and storage. The term (1b) captures the cost of plant retirement or decommissioning. The variable operating and maintenance (VOM) are represented by term (1c). The network expansion costs is included by term (1d). The cost of CO2 transport and storage infrastructure required to accompany CC-CCS power generation is incorporated by term (1e) where it captures the cost associated with establishing $\mathrm{CO_2}$ pipelines and storage. Note that we conservatively assume that each $\mathrm{CO_2}$ pipeline connects a power node to the storage site, which ignores the possibility of meshed network design for CO2 transport. The cost of fuel consumption for non-gas-fired power plants (i.e., nuclear plant) are ensured by term (1f). The term term (1g) penalizes the load shedding in the power system which can occur due to unsatisfied demand.

Investment and Unit Commitment: For every $n \in \mathcal{N}^{\mathrm{e}}, i \in \mathcal{P}$

$$x^{\mathrm{op}}_{ni} = I^{\mathrm{num}}_{ni} - x^{\mathrm{dec}}_{ni} + x^{\mathrm{est}}_{ni} \tag{2a}$$

$$\tag{2b}$$

constraints (2a) specify the number of operating plants. The unit commitment

constraints for each node and plant type are presented in (**??**) which computes the number of plants committed, started up, or shut down during a period. Constraints (**??**) limits the number of committed units to the number of available ones. As per other similar studies [**?** 4, 3**?**], we relaxed the integrality of unit commitment decisions to ease the computational complexity of the problem .
Generation, Ramping, and Load Shedding: For every $n \in \mathcal{N}^{\mathrm{e}}, t \in \mathcal{T}^{\mathrm{e}}$

$$L_i^{\mathrm{prod}} U_i^{\mathrm{prod}} x_{ni}^{\mathrm{op}} \leq p_{nti} \leq U_i^{\mathrm{prod}} x_{ni}^{\mathrm{op}} \qquad i \in \mathcal{H} \tag{3a}$$

$$\begin{aligned} &|p_{nti} - p_{n,(t-1),i}| \leq U_i^{\mathrm{ramp}} U_{ni}^{\mathrm{prod}} x_{ni}^{\mathrm{op}} + \\ &\max(L_i^{\mathrm{prod}}, U_i^{\mathrm{ramp}}) U_i^{\mathrm{prod}} x_{ni}^{\mathrm{op}} \qquad i \in \mathcal{H} \end{aligned} \tag{3b}$$

$$p_{nti} \leq \rho_{nti} U_{ni}^{\mathrm{prod}} x_{ni}^{\mathrm{op}} \qquad i \in \mathcal{R} \tag{3c}$$

$$a_{nt}^{\mathrm{e}} \leq D_{n\phi_t^{\mathrm{e}}}^{\mathrm{e}} \tag{3d}$$

the generation limits are imposed in constraints (3a). Constraints (3b) are the ramping constraints that limit the generation difference of thermal units in any consecutive time periods to a ramping limit in the right-hand-side of the equation. The generation pattern of VREs is determined by their hourly profile in the form of capacity factor; constraints (3c) limit the generation of VRE to hourly capacity factor (i.e. ρ_{nti}) of maximum available capacity (i.e. $U_{ni}^{\mathrm{prod}} x_{ni}^{\mathrm{op}}$). Constraints (3d) state that the load shedding amount can not exceed demand.
Power Balance Constraints: For every $n \in \mathcal{N}^{\mathrm{e}}, t \in \mathcal{T}^{\mathrm{e}}$

$$\begin{aligned} &\sum_{i \in \mathcal{P}} p_{nti} + \sum_{m \in \mathcal{N}^{\mathrm{e}}} \sum_{l \in \mathcal{L}_{nm}^{\mathrm{e}}} sgn(n-m) f_{\ell t}^{\mathrm{e}} + \sum_{r \in \mathcal{S}_n^{\mathrm{e}}} (s_{ntr}^{\mathrm{eDis}} - s_{ntr}^{\mathrm{eCh}}) + \\ &a_{nt}^{\mathrm{e}} = D_{n\phi_t^{\mathrm{e}}}^{\mathrm{e}} + \mathfrak{D}_n E^{\mathrm{pipe}} \kappa_n^{\mathrm{pipe}} + E^{\mathrm{cprs}} E^{\mathrm{pump}} \kappa_{nt}^{\mathrm{capt}} \end{aligned} \tag{4a}$$

constraints (4a) ensures that for each node at each planning period the generation, power curtailment amount, the net flow, and the net storage power is equal to the net demand. The net demand is defined in the right-hand-side where the first term is the baseline demand, the second term is the electricity consumption by CO_2 pipelines and the last term is the electricity used by compressors. The notation $sgn(n-m)$ is the *sign* function that takes value -1 if $n < m$, value 1 if $n > m$, and 0 otherwise. We use this function to ensure that $f_{\ell t}^{e}$ appears with opposite signs in the balance equations of the nodes connected by transmission line ℓ.
Network Constraints: For every $l \in \mathcal{L}^{\mathrm{e}}, t \in \mathcal{T}^{\mathrm{e}}, n, m \in \mathcal{N}_\ell^{\mathrm{e}}$

$$|f_{lt}^{\mathrm{e}}| \leq I_\ell^{\mathrm{trans}} \qquad \text{if } \mathcal{I}_\ell^{\mathrm{trans}} = 1 \tag{5a}$$

$$|f_{lt}^{\mathrm{e}}| \leq U_\ell^{\mathrm{trans}} z_\ell^{\mathrm{e}} \qquad \text{if } \mathcal{I}_\ell^{\mathrm{trans}} = 0 \tag{5b}$$

$$\tag{5c}$$

Flow for the existing transmission lines is limited by constraints (5a). Constraints (5b) limits the flow in candidate transmission lines only if it is already established (i.e., z^{t}_ℓ=1). DC power flow constraints for candidate and existing transmission lines are respectively imposed in constraints (**??**) and (**??**). Constraints (**??**) restricts the phase angles, and constraints (**??**) sets the phase angle for the reference node 0.

Storage Constraints: For every $n \in \mathcal{N}^{\mathrm{e}}, t \in \mathcal{T}^{\mathrm{e}}, r \in \mathcal{S}^{\mathrm{e}}_n$

$$s^{\mathrm{eLev}}_{ntr} = s^{\mathrm{eLev}}_{n,t-1,r} + \gamma^{\mathrm{eCh}}_r s^{\mathrm{eCh}}_{ntr} - \frac{s^{\mathrm{eDis}}_{ntr}}{\gamma^{\mathrm{eDis}}_r} \tag{6a}$$

$$s^{\mathrm{eDis}}_{ntr} \leq y^{\mathrm{eCD}}_{nr}, s^{\mathrm{eCh}}_{ntr} \leq y^{\mathrm{eCD}}_{nr} \tag{6b}$$

$$s^{\mathrm{eLev}}_{ntr} \leq y^{\mathrm{eLev}}_{nr} \tag{6c}$$

$$s^{\mathrm{eLev}}_{n,t_1,r} = s^{\mathrm{eLev}}_{n,t_{24},r} \qquad t_1, t_{24} \in \mathfrak{T}^{\mathrm{e}}_\tau, \tau \in \mathfrak{R} \tag{6d}$$

Constraints (6a) model battery storage dynamics. The charge/discharge limits are imposed in (6b), and constraints (6c) limits the storage level. Note that as for other similar studies [4, 3], we do not account for storage capacity degradation. Representative days are not necessarily consecutive, therefore the formulation should account for the carryover storage level between representative days. Li et al. [3] enforce the beginning and ending storage levels of each representative days to 50% of the maximum storage level. In constraints (6d), we use a similar technique, yet more flexible, as we assume that beginning (i.e., t_1) and ending (i.e., t_{24}) storage levels are the same for any representative day.

Renewable Portfolio Standards (RPS):

$$\sum_{n\in\mathcal{N}^{\mathrm{e}}} \sum_{t\in\mathcal{T}^{\mathrm{e}}} \sum_{i\in\mathcal{R}} p_{nti} \geq L^{\mathrm{RPS}} \sum_{n\in\mathcal{N}^{\mathrm{e}}} \sum_{t\in\mathcal{T}^{\mathrm{e}}} D^{\mathrm{e}}_{n\phi^{\mathrm{e}}_t} \tag{7a}$$

The formulation requires the model to procure a certain share of the total demand from renewable energy sources. The share of renewable energy sources which is known as Renewable Portfolio Share (RPS) is imposed by constraint (7a).

1.2 NG System Model

Objective Function:

$$\min \sum_{l\in\mathcal{L}^{\mathrm{g}}} C^{\mathrm{pipe}}_\ell z^{\mathrm{g}}_\ell + \tag{8a}$$

$$\sum_{k\in\mathcal{N}^{\mathrm{g}}} \sum_{\tau\in\mathcal{T}^{\mathrm{g}}} C^{\mathrm{ng}} g_{k\tau} + \tag{8b}$$

$$\sum_{j\in\mathcal{N}^s} (C^{\mathrm{strInv}}_j x^{\mathrm{gStr}}_j + C^{\mathrm{vprInv}}_j x^{\mathrm{vpr}}_j) + \tag{8c}$$

$$\sum_{j\in\mathcal{N}^s} \Big(C^{\mathrm{strFix}}(I^{\mathrm{gStr}}_j + x^{\mathrm{gStr}}_j) + C^{\mathrm{vprFix}}(I^{\mathrm{vpr}}_j + x^{\mathrm{vpr}}_j) \Big) + \tag{8d}$$

$$\sum_{k\in\mathcal{N}^{\mathrm{g}}} \sum_{t\in\mathcal{T}^{\mathrm{g}}} (C^{\mathrm{rng}} a^{\mathrm{rng}}_{k\tau} + C^{\mathrm{gShed}} a^{\mathrm{ng}}_{k\tau}) \tag{8e}$$

The objective function (8) minimizes the total investment and operating costs incurred in the NG system. The first term (8a) is the investment cost for establishing new pipelines. The second term (8b) is the cost of procuring NG from various sources to the system. For example, New England procures its NG from Canada, and its adjacent states such as New York. The term (8c) and (8d) handle the investment and FOM costs associated with NG storage, respectively. The last term (8e) captures the cost of using RNG and NG load shedding.

NG Balance Constraint: For every $k \in \mathcal{N}^{\mathrm{g}}, \tau \in \mathcal{T}^{\mathrm{g}}$

$$g_{k\tau} - \sum_{l \in \mathcal{L}_k^{\mathrm{gExp}}} f_{\ell\tau}^{\mathrm{g}} + \sum_{l \in \mathcal{L}_k^{\mathrm{gImp}}} f_{\ell\tau}^{\mathrm{g}} - \sum_{n \in \mathcal{A}_k^{\mathrm{e}}} f_{kn\tau}^{\mathrm{ge}} + \sum_{j \in \mathcal{A}_k^{s}} (f_{jk\tau}^{\mathrm{vg}} - f_{kj\tau}^{\mathrm{gl}}) + a_{k\tau}^{\mathrm{rng}} + a_{k\tau}^{\mathrm{g}} = D_{k\tau}^{\mathrm{g}} \tag{9a}$$

constraints (9a) state that for each node and period, the imported NG (i.e., injection), flow to other NG nodes, flow to power nodes, flow from and to storage nodes, satisfied load by RNG should, and unsatisfied NG load add up to demand. Unlike power flow, the flow in pipelines are modeled unidirectional as it is typical for most long-distance transmission pipelines involving booster compressor stations [**?**]. We are ignoring electricity consumption associated with booster compression stations along the NG pipeline network. Note that there is no load shedding in NG system as we assume RNG availability for any quantity.

Representative Days:

$$f_{kn\tau_1}^{\mathrm{ge}} = f_{kn\tau_2}^{\mathrm{ge}} \qquad \tau_1, \tau_2 \in \Omega_\tau, \tau \in \Re \tag{10a}$$

The constraint (10a) captures the impact of representative days on the gas system. It ensures that gas consumption by the power system for all the days in the same cluster is the same.

Gas and RNG Supply Constraints: For every $k \in \mathcal{N}^{\mathrm{g}}, \tau \in \mathcal{T}^{\mathrm{g}}$

$$L_k^{\mathrm{inj}} \leq g_{k\tau} \leq U_k^{\mathrm{inj}} \tag{11a}$$

$$a_{k\tau}^{\mathrm{rng}} + a_{k\tau}^{\mathrm{g}} \leq D_{k\tau}^{\mathrm{g}} \tag{11b}$$

$$a_{k\tau}^{\mathrm{rng}} \leq U_k^{\mathrm{inj}} M \tag{11c}$$

import limits are imposed in constraints (11a). The consumption of RNG plus the load shedding is limit by constraints (11b) to the NG load. The alternative fuel RNG can only be imported from injection points as specified by constraints 11c.

Flow Constraints: For every $\ell \in \mathcal{L}^{\mathrm{g}}, \tau \in \mathcal{T}^{\mathrm{g}}, j \in \mathcal{N}^{s}$

$$\begin{aligned}
& f^{\mathrm{g}}_{\ell\tau} \leq I^{\mathrm{pipe}}_{\ell} && \text{if } \mathcal{I}^{\mathrm{pipe}}_{\ell} = 1 && \text{(12a)}\\
& f^{\mathrm{g}}_{\ell\tau} \leq U^{\mathrm{pipe}}_{\ell} z^{\mathrm{g}}_{\ell} && \text{if } \mathcal{I}^{\mathrm{pipe}}_{\ell} = 0 && \text{(12b)}\\
& \sum_{k \in \mathcal{N}^{\mathrm{g}}: j \in \mathcal{A}^{s}_{k}} f^{\mathrm{gl}}_{kj\tau} = s^{\mathrm{liq}}_{j\tau} && && \text{(12c)}\\
& \sum_{k \in \mathcal{N}^{\mathrm{g}}: j \in \mathcal{A}^{s}_{k}} f^{\mathrm{vg}}_{jk\tau} = s^{\mathrm{vpr}}_{j\tau} && && \text{(12d)}
\end{aligned}$$

constraints (12a) and (12b) limit the flow between NG nodes for existing and candidate pipelines, respectively. The flow to liquefaction facilities is calculated in constraints (12c). Similarly, the flow out of vaporization facilities is modeled via constraints (12d). note that because these SVL nodes are often connected with the NG nodes with truck transport, that is not modeled here, we do not consider capacity constraint related to the flows to and from the SVL nodes.
Storage Constraints: For every $j \in \mathcal{N}^{s}, \tau \in \mathcal{T}^{\mathrm{g}}$

$$\begin{aligned}
& s^{\mathrm{gStr}}_{j\tau} = (1-\beta) s^{\mathrm{gStr}}_{j,\tau-1} + \gamma^{\mathrm{liqCh}}_{j} s^{\mathrm{liq}}_{j\tau} - \frac{s^{\mathrm{vpr}}_{j\tau}}{\gamma^{\mathrm{vprDis}}_{j}} && \text{(13a)}\\
& s^{\mathrm{vpr}}_{j\tau} \leq I^{\mathrm{vpr}}_{j} + x^{\mathrm{vpr}}_{j} && \text{(13b)}\\
& s^{\mathrm{gStr}}_{j\tau} \leq I^{\mathrm{gStr}}_{j} + x^{\mathrm{gStr}}_{j} && \text{(13c)}
\end{aligned}$$

constraints (13a) ensure the storage balance. Constraints (13b) and (13c) limit the capacity of vaporization and storage tanks to their initial capacity plus the extended capacity, respectively.

1.3 Coupling Constraints

The following constraints are coupling constraints that relate decisions of the two systems.

$$\begin{aligned}
& \sum_{k \in \mathcal{A}^{\mathrm{e}}_{n}} f^{\mathrm{ge}}_{kn\tau} = \sum_{t \in \mathfrak{T}^{\mathrm{e}}_{\tau}} \sum_{i \in \mathcal{G}} h_i p_{nti} && n \in \mathcal{N}^{\mathrm{e}}, \tau \in \mathfrak{R} && \text{(14a)}\\
& \mathcal{E}^{\mathrm{e}} = \sum_{n \in \mathcal{N}^{\mathrm{e}}} \sum_{t \in \mathcal{T}^{e}} \sum_{i \in \mathcal{G}} w_t (1-\eta_i) \eta^{\mathrm{g}} h_i p_{nti} && &&\\
& \mathcal{E}^{\mathrm{g}} = \sum_{k \in \mathcal{N}^{g}} \sum_{\tau \in \mathcal{T}^{g}} \eta^{\mathrm{g}} (D^{\mathrm{g}}_{k\tau} - a^{\mathrm{rng}}_{k\tau} - a^{\mathrm{g}}_{k\tau}) && &&\\
& \mathcal{E}^{\mathrm{e}} + \mathcal{E}^{\mathrm{g}} \leq (1-\zeta)(U^{\mathrm{e}}_{\mathrm{emis}} + U^{\mathrm{g}}_{\mathrm{emis}}) && && \text{(14b)}
\end{aligned}$$

The first coupling constraints (14a) captures the flow of NG to the power network for each node and at each time period. The variable $\mathcal{E}^{\mathrm{e}}$ accounts for the emission due to the consumption of NG in the power system. The variable $\mathcal{E}^{\mathrm{g}}$ computes the emission from NG system by subtracting the demand from RNG consumption. The second coupling constraint (14b) ensures that the net CO_2 emissions associated with electric-NG system is below the specified threshold value, which is defined based on reduction relative to some baseline emissions.Since the model cannot track whether RNG is used to meet non-power NG demand or for power generation, the constraint (14b) first computes gross emissions from all NG use presuming it is all fossil and then subtracts emissions benefits from using RNG. Here we treat RNG as a carbon-neutral fuel source, and thus the combustion emissions associated with its end-use are equal to the emissions captured during its production. Recent life cycle analysis studies suggest that depending on the feedstock RNG could have negative to slightly positive life cycle GHG emissions [**?**].

The first term is the emission due to non-generational NG consumption (i.e., NG consumption in the NG system such as space heating, industry use, and transportation) and the second term captures the emission from gas-fired power plants. Alternatively, the emission constraints can only be applied to the power system as in [4] or separately applied to each system as in [**?**].

References

[1] Aron Brenner, Rahman Khorramfar, Dharik Mallapragada, and Saurabh Amin. Graph representation learning for energy demand data: Application to joint energy system planning under emissions constraints. *arXiv preprint arXiv:2209.12035*, 2022.

[2] Wesley J Cole, Danny Greer, Paul Denholm, A Will Frazier, Scott Machen, Trieu Mai, Nina Vincent, and Samuel F Baldwin. Quantifying the challenge of reaching a 100% renewable energy power system for the united states. *Joule*, 5(7):1732–1748, 2021.

[3] Can Li, Antonio J Conejo, John D Siirola, and Ignacio E Grossmann. On representative day selection for capacity expansion planning of power systems under extreme operating conditions. *International Journal of Electrical Power & Energy Systems*, 137:107697, 2022.

[4] Nestor A Sepulveda, Jesse D Jenkins, Aurora Edington, Dharik S Mallapragada, and Richard K Lester. The design space for long-duration energy storage in decarbonized power systems. *Nature Energy*, 6(5):506–516, 2021.