Learning Approaches for Generalization Beyond the IID Assumption

Pankaj Malhotra, Hritik Bansal, Gantavya Bhatt

IIT Delhi

pankajiitdyn@gmail.com, hbansal10n@gmail.com, gbhatt2@uw.edu

October 20, 2020

Overview I

Objective

Challenges in Existing DL Approaches

Biological (Neuroscience) Inspirations

Desiderata

Representation Learning - A perspective
- Learning Disentangled Representations
- Non-Linear Independent Components Analysis
- Structured Networks

World Models

Video Prediction Models

C-SWM and iVAE
- C-SWM Review

Proposal

To Do

Objective

- Long-term objective
  - Build neural networks that
    - generalize to new tasks, i.e. exhibit combinatorial generalization [7]
    - are able to construct new inferences, predictions, and behaviors from known building blocks
    - are able to explain their reasoning
- Short-term objective
  - Approaches to extract the underlying
    - disentangled factors [18]
    - independent mechanisms [78]
    - causal mechanisms [78]

    that result in the observed data that is usually entangled.
  - Very recently, it has been shown that disentangled representations do help with abstract reasoning [85] and life-long learning [1].

In short: *move deep learning from perception to higher-level cognition and knowledge representation* [8].

Challenges in Existing DL Approaches

- Assumed motivation and intuition behind success of DL: higher layers learn abstract concepts and hierarchies in a way similar to how humans do [55].
- Unlike humans, existing DL methods [44, 25, 24]
  - prone to just capturing correlations in data
  - tend to learn non-robust features that are discriminative on the given task
  - fail to align with human perception
  - do not capture the higher-level abstract concepts
  - are not easily transferable to new tasks
  - are extremely sensitive to adversarial perturbations

  In short: *Existing DL methods perform poorly under task-distribution shifts that demand generalization to scenarios violating the standard assumption [63] of i.i.d. samples or tasks at test time.*

Biological (Neuroscience) Inspirations

- Abstraction: brain maximizes predictive information [74, 10] at an abstract level
- Modularity [16, 83]
- Self-organizing and routing in neural networks [32]
- Competitive interactions between neurons and neural circuits [31]
- Bottom-up and top-down attention mechanisms [17]

However, most current deep neural networks are based on stimulus-response, i.e. given an input use statistical correlations to get an output.

Desiderata I

What is needed to achieve human-level cognition?

How many blocks are on the right of the three-level tower?

Will the block tower fall if the top block is removed?

What is the shape of the object closest to the large cylinder?

Are there more trees than animals?

Figure 1: Human reasoning is interpretable and disentangled: we first draw abstract knowledge of the scene via visual perception and then perform logic reasoning on it. This enables compositional, accurate, and generalizable reasoning in rich visual contexts.

Understanding and modeling of **compositional** environments involving multiple interacting objects that can be manipulated independently by an agent.

Figure adapted from [98].

Desiderata II

Given a dynamic "compositional"[1] environment:

- Identify **objects**
- Identify **relations/interactions** between objects
- Identify the **properties** of the objects and their relations
- Reason on top of above points to **explain** decisions / achieve **interpretability**.
- All this needs to be enabled in a **life-long learning** setting.

[1] in the sense that its features can be described by compositions of small sets of primitive mechanisms [71]

Role of Inductive Biases

- Unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the data [60].
- Identifiability [47]
- Symmetry-Based Disentangled Representation Learning requires Interaction with Environments [12]
- All research directions from previous slide can be seen as introducing novel inductive biases for learning.
- Structure of the neural network is important [93].

Structured Networks

- Modularized Networks, e.g. Neural Module Networks [3]
- Routing Mechanisms [75]
- Randomly Wired Networks [93]
- Evolving Deep Neural Networks [64]

Learning Structured Networks I

Visual Question Answering can provide a good testbed for testing models with cognitive ability.

Neural Module Networks (NMN) [3]:

- enable **compositionality**: answer questions with arbitrarily complex structure.
- for a given question, **dynamically lay out a network** composed of reusable modules.

Learning Structured Networks II

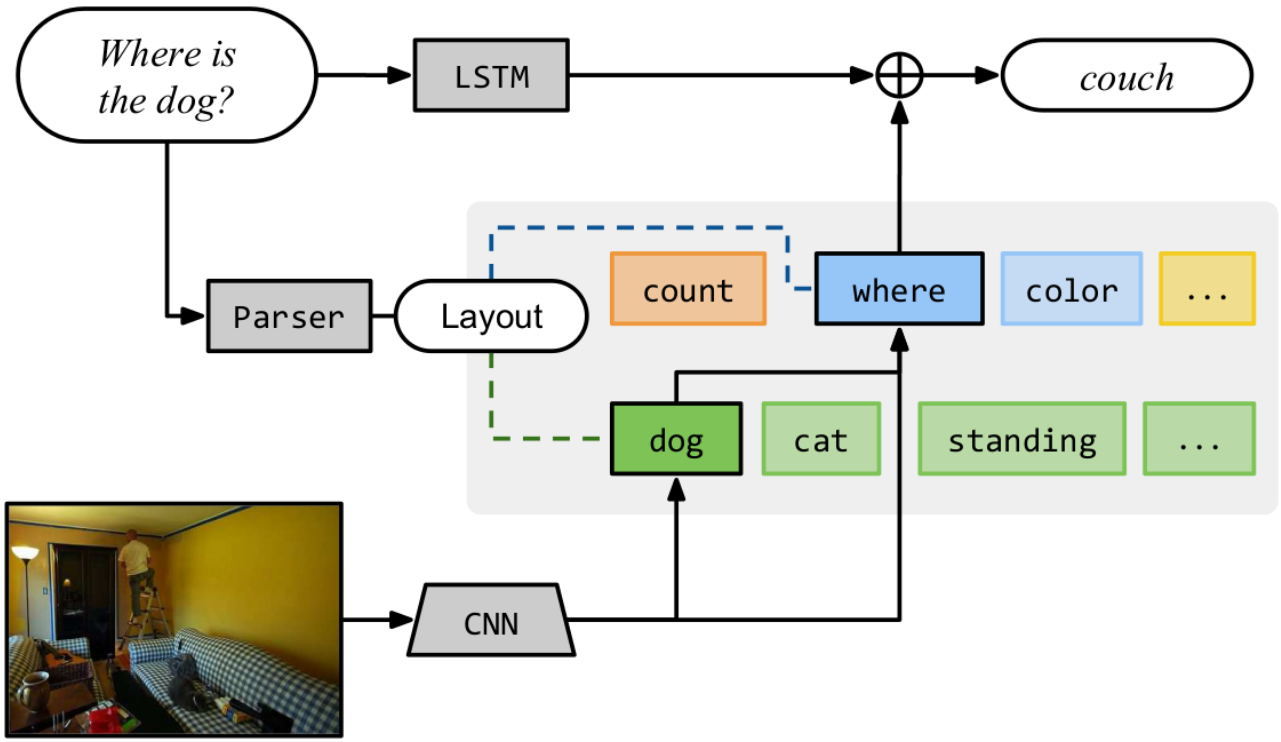
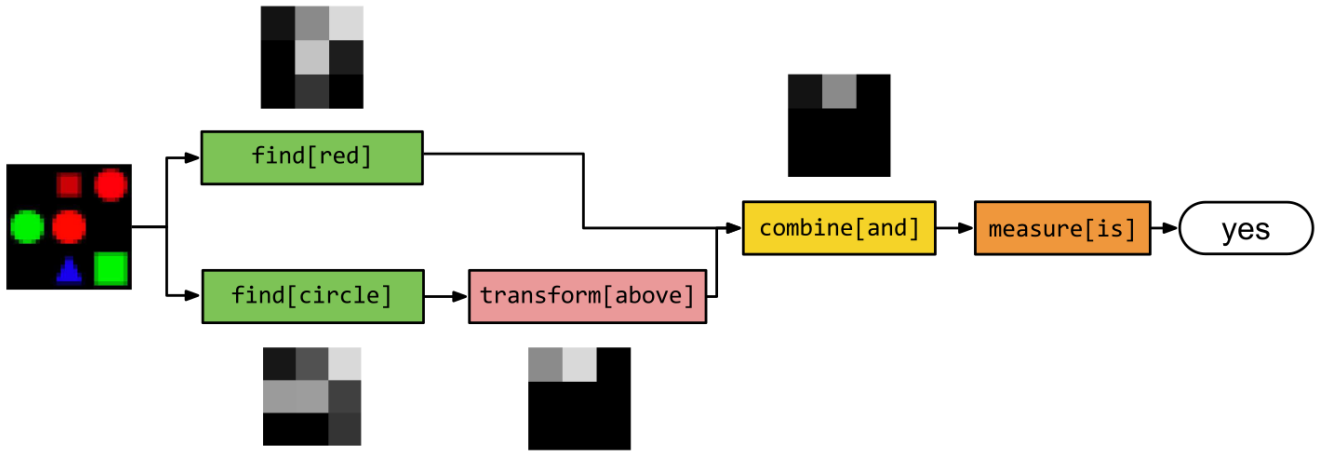


Figure: NMN Example

Learning Structured Networks III



(b) NMN for answering the question *Is there a red shape above a circle?* The two `find` modules locate the red shapes and circles, the `transform[above]` shifts the attention above the circles, the *combine* module computes their intersection, and the `measure[is]` module inspects the final attention and determines that it is non-empty.

Figure: NMN Working Example

Learning Structured Networks IV

- almost all interesting compositional phenomena occur in the space of **attentions**

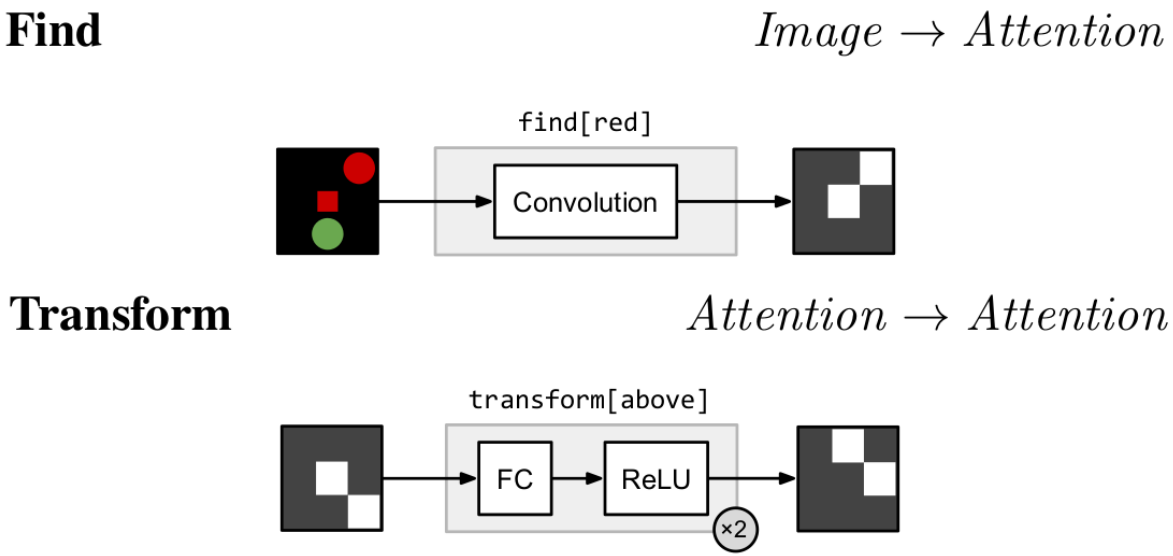
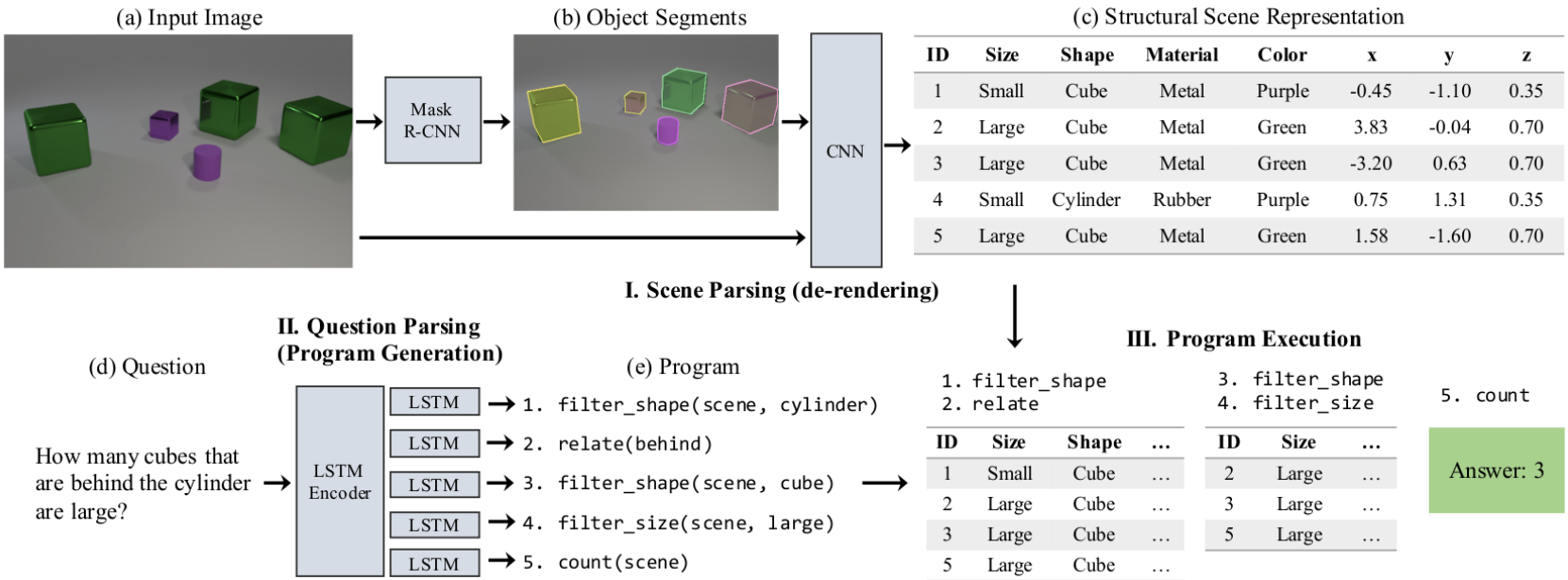


Figure: Implementation of Modules using Attention

Neural-Symbolic Visual Question Answering [98]

- disentangle reasoning from vision and language understanding



ID	Size	Shape	Material	Color	x	y	z
1	Small	Cube	Metal	Purple	-0.45	-1.10	0.35
2	Large	Cube	Metal	Green	3.83	-0.04	0.70
3	Large	Cube	Metal	Green	-3.20	0.63	0.70
4	Small	Cylinder	Rubber	Purple	0.75	1.31	0.35
5	Large	Cube	Metal	Green	1.58	-1.60	0.70

ID	Size	Shape	...
1	Small	Cube	...
2	Large	Cube	...
3	Large	Cube	...
5	Large	Cube	...

ID	Size	...
2	Large	...
3	Large	...
5	Large	...

Figure 2: Our model has three components: first, a scene parser (de-renderer) that segments an input image (a-b) and recovers a structural scene representation (c); second, a question parser (program generator) that converts a question in natural language (d) into a program (e); third, a program executor that runs the program on the structural scene representation to obtain the answer.

World Models: Learning in Abstract Concept Space

- *Supervised* Learning by Abstraction: Neural State Machines
- *Unsupervised* Relational Neural Expectation Maximization
- *Contrastive* Learning in Abstract Space

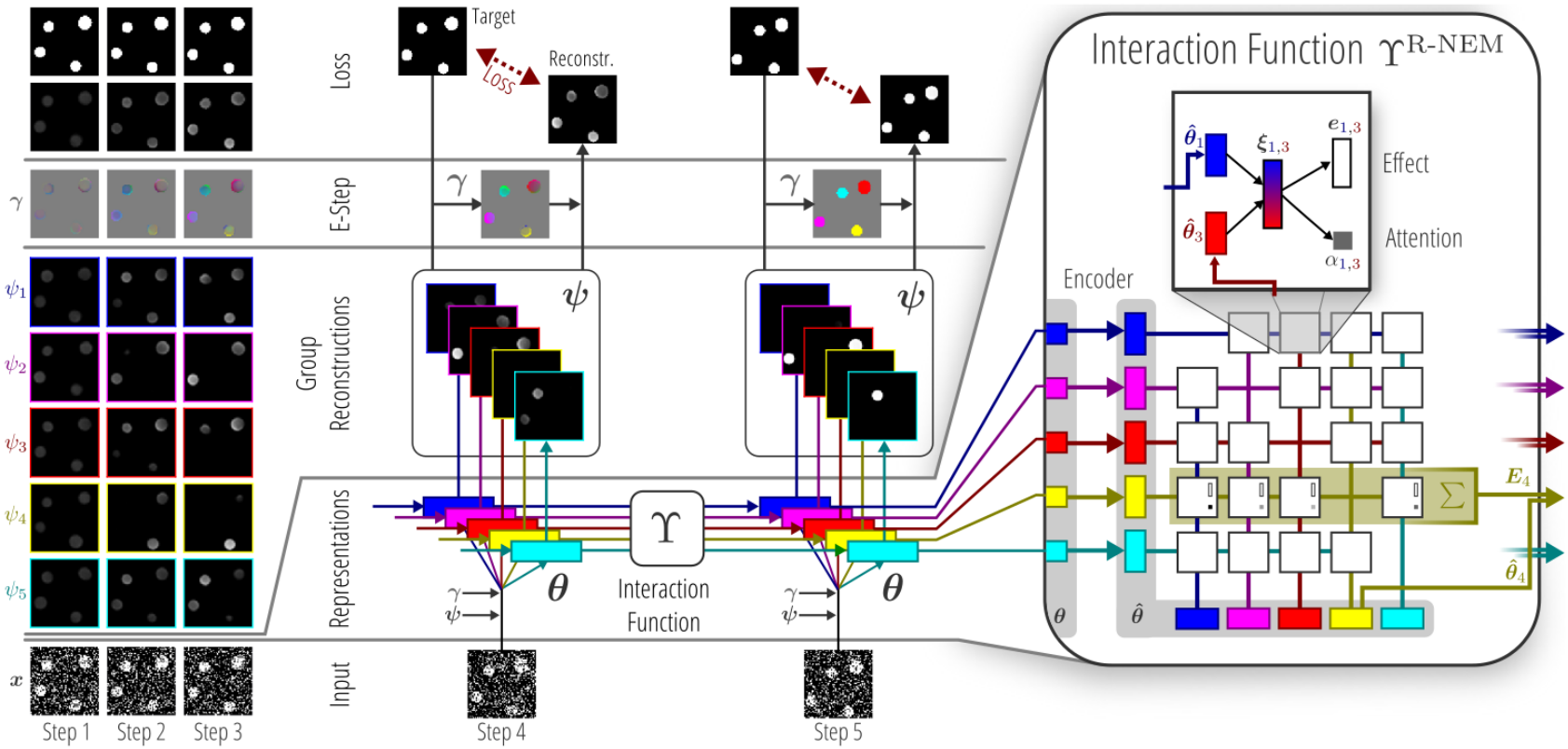
Desiderata (Revisited)

What is needed to achieve human-level cognition?
Compositional environments involve multiple interacting objects that can be manipulated independently by an agent.
Given a dynamic "compositional" environment:

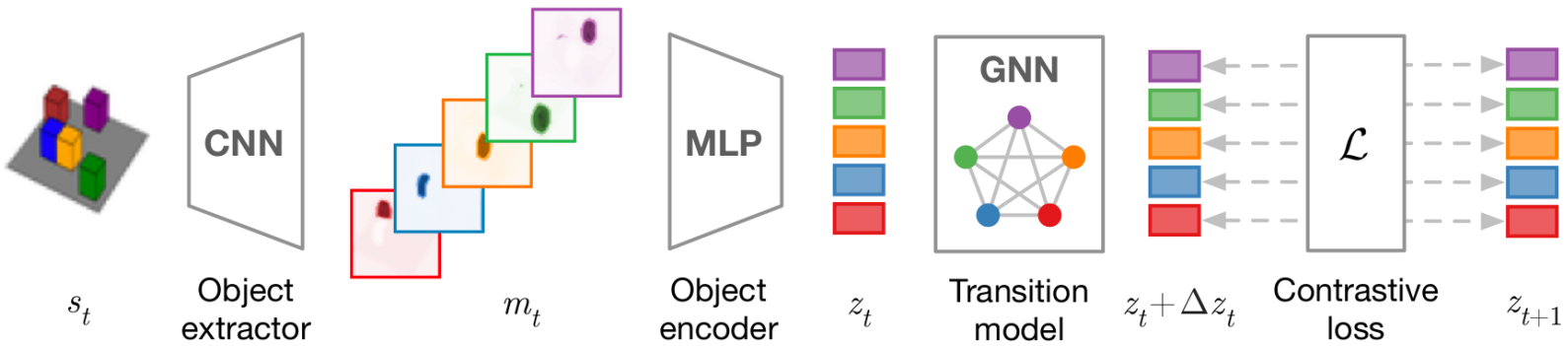
- Identify **objects**
- Identify **relations/interactions** between objects
- Identify the **properties** of the objects and their relations
- Reason on top of above points to **explain** decisions / achieve **interpretability**.
- All this needs to be enabled in a **life-long learning** setting.

In short: Ability to discover and describe a scene in terms of objects is essential for common-sense physical reasoning.

Relational Neural Expectation Maximization (R-NEM) [88]
Unsupervised Discovery of Objects and Their Interactions



Contrastive Learning of Structured World Models (C-SWM) [50]



- a CNN-based object extractor: *stimulus-response*
- an MLP-based object encoder: *compositional*
- a GNN-based relational transition model: *interaction understanding, compositional*
- an object-factorized contrastive loss: *learning in abstract space*

Key: Loss is defined in the abstract space!

Representation Learning with Contrastive Predictive Coding

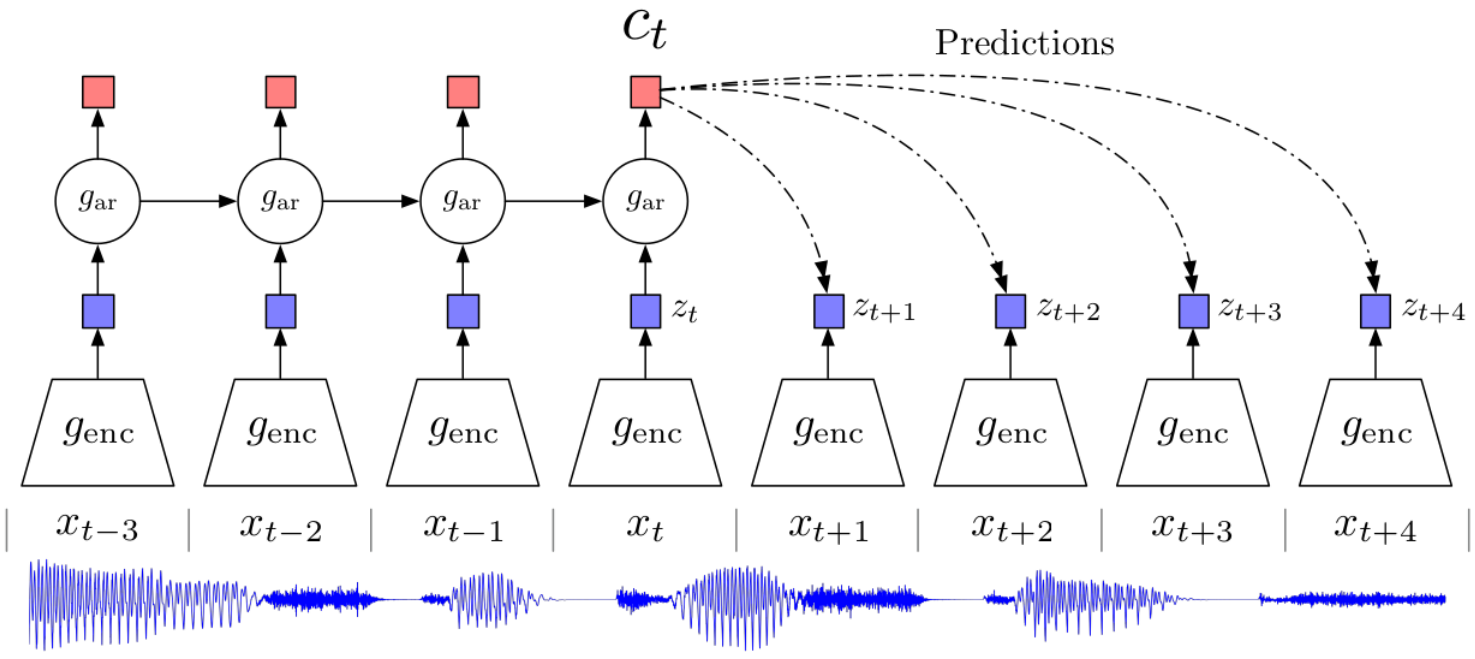


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

Key: Predict in the latent space! [69]

Again, uses ideas from Aapo's work on Noise-Contrastive Estimation.

Parts, Structure, Dynamics [94]

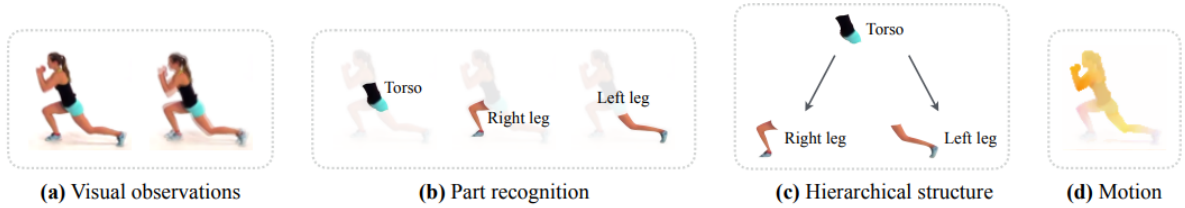


Figure 1: Observing human moving, humans are able to perceive disentangled object parts, understand their hierarchical structure, and capture their corresponding motion fields (without any annotations).

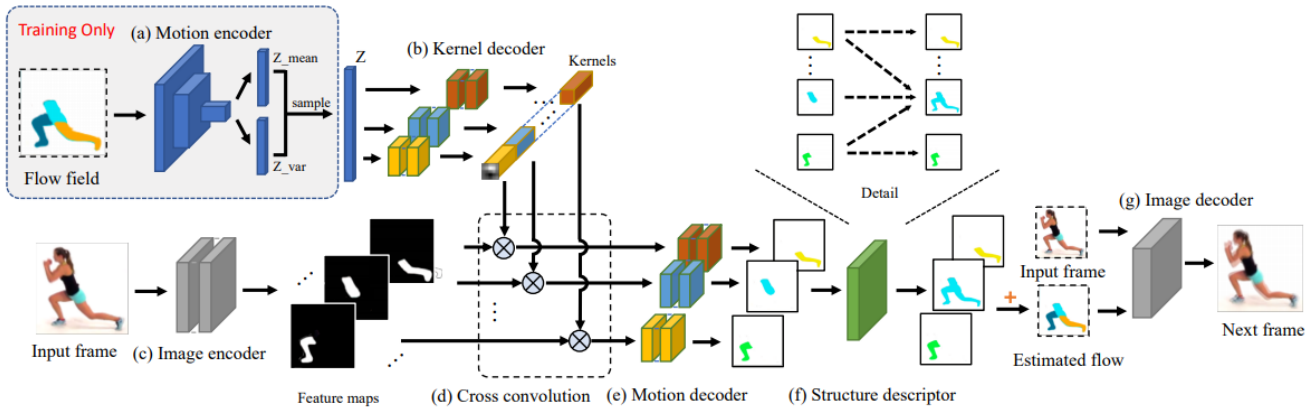


Figure 3: Our PSD model has seven components: **(a)** motion encoder; **(b)** kernel decoder; **(c)** image encoder; **(d)** cross convolution; **(e)** motion decoder; **(f)** structural descriptor; and **(g)** image decoder.

produce a structured, hierarchical object representation, and characterizes system dynamics

Contrastive state representation learning [2]

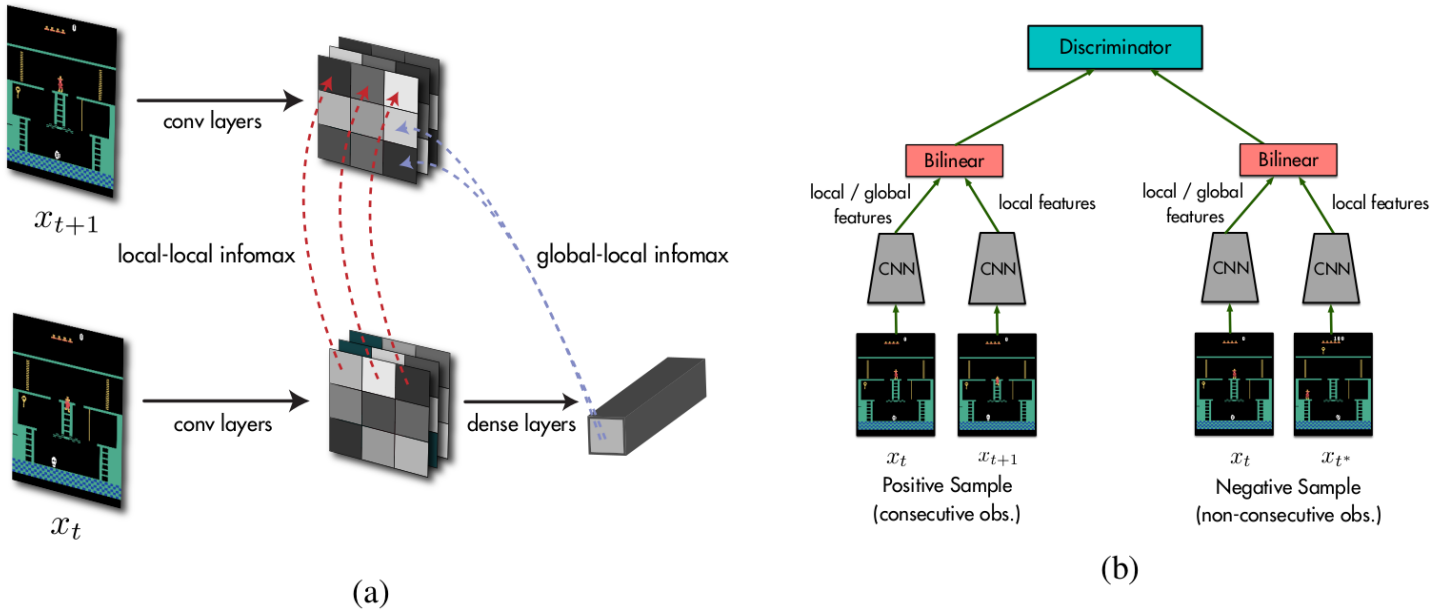
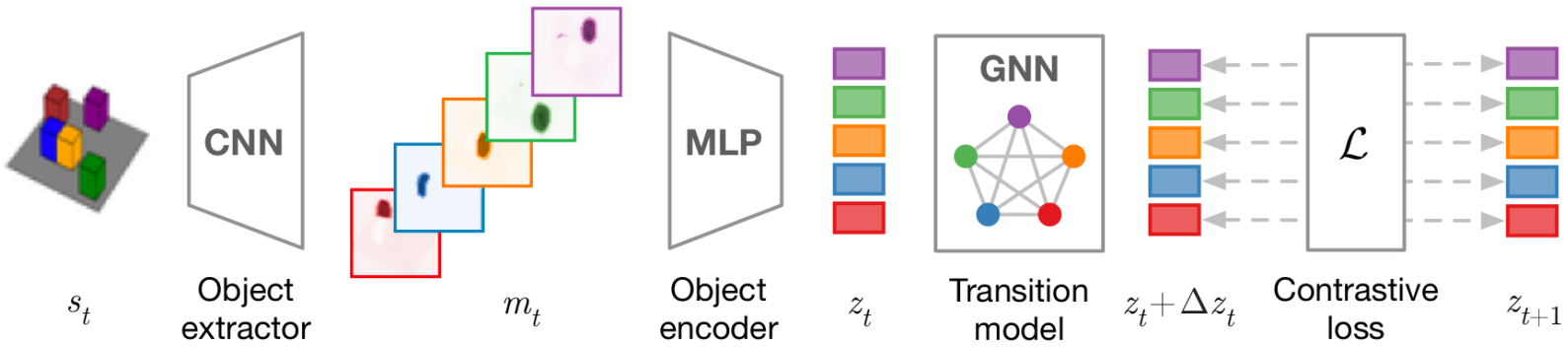


Figure 2: A schematic overview of SpatioTemporal DeepInfoMax (ST-DIM). (a) shows the two different mutual information objectives: local infomax and global infomax. (b) shows a simplified version of the contrastive task we use to estimate mutual information. In practice, we use multiple negative samples.

ST-DIM: maximize mutual information between data and learned representations

C-SWM Review I

Contrastive Learning of Structured World Models (C-SWM) [50]



- a CNN-based object extractor: *stimulus-response*
- an MLP-based object encoder: *compositional*
- a GNN-based relational transition model: *interaction understanding*, *compositional*
- an object-factorized contrastive loss: *learning in abstract space*

Key: Loss is defined in the abstract space!

C-SWM Review II

experience buffer $\mathcal{B} = \{(s_t, a_t, s_{t+1})\}_{t=1}^{T}$

states $s_t \in \mathcal{S}$

actions $a_t \in \mathcal{A}$

follow-up states $s_{t+1} \in \mathcal{S}$

encoder $E : \mathcal{S} \to \mathcal{Z}$

transition model $T : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$

abstract state space $\mathcal{Z} = \mathcal{Z}_1 \times \ldots \times \mathcal{Z}_K$, where K is the number of available object slots

object-factorized action space $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_K$

$z_{t+1} = (z_t^1 + \Delta z_t^1, \ldots, z_t^K + \Delta z_t^K)$

C-SWM Review III

State-transition function:

$$\Delta z_t = T(z_t, a_t) = \mathrm{GNN}(\{(z_t^k, a_t^k)\}_{k=1}^K)$$

Per-object transition via message passing:

$$e_t^{(i,j)} = f_{\text{edge}}([z_t^i, z_t^j])$$
$$\Delta z_t^j = f_{\text{node}}([z_t^j, a_t^j, \textstyle\sum_{i \neq j} e_t^{(i,j)}])$$

H: distance from correct frame, $\tilde{H}$: distance from random (negative) frame:

$$H = \frac{1}{K} \sum_{k=1}^{K} d(z_t^k + T^k(z_t, a_t), z_{t+1}^k), \quad \tilde{H} = \frac{1}{K} \sum_{k=1}^{K} d(\tilde{z}_t^k, z_{t+1}^k)$$

Loss function per $< s_t, a_t, s_{t+1} >$:

$$\mathcal{L} = H + \max(0, \gamma - \tilde{H})$$

C-SWM Review IV

Input:

- $50 \times 50 \times 3$ color images for the grid world environments.
- $50 \times 50 \times 6$ tensors (two concatenated consecutive frames) for the Atari and 3-body Physics environments.
- one object is acted upon at one time step
- 4 possible actions in (a) and (b). Passed as one-hot vectors to object slots. If no action, then all 0s vector is passed.
- Same one-hot encoded action vector to every object slot for (c) and (d).
- No action input for (e).

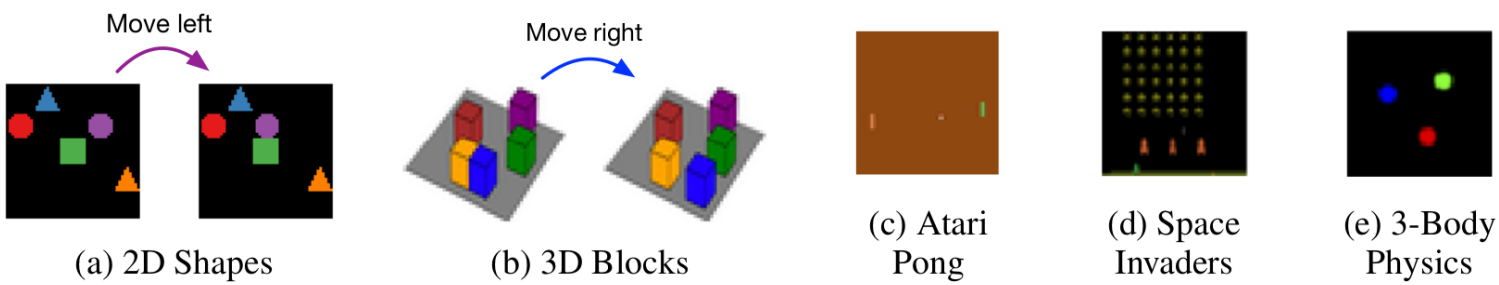
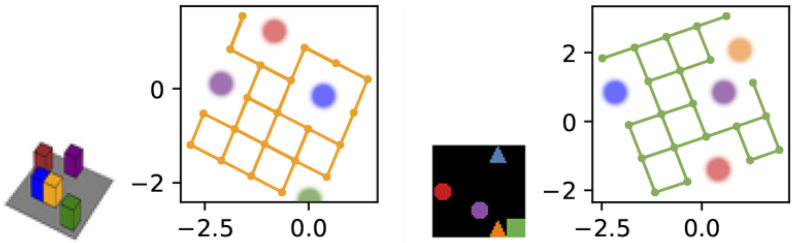


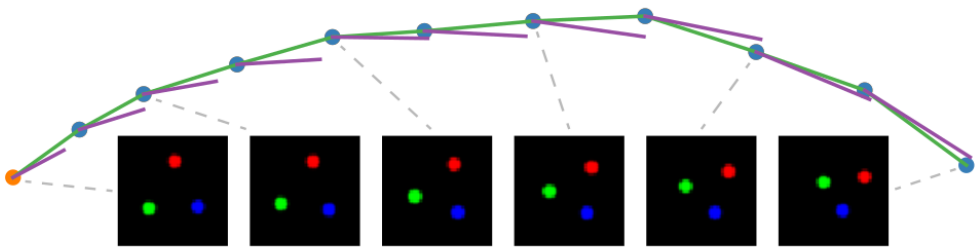
Figure: Environments

C-SWM Review V

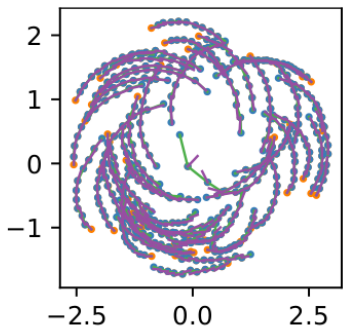
(a) Discovered object masks in a scene from the 3D cubes (top) and 2D shapes (bottom) environments.



(b) Learned abstract state transition graph of the yellow cube (left) and the green square (right), while keeping all other object positions fixed at test time.



(a) Observations from 3-body gravitational physics simulation (bottom) and learned abstract state transition graph for a single object slot (top).



(b) Abstract state transition graph from 50 test episodes for single object slot.

Figure: Results

C-SWM Review VI

- Failure modes of generative models include:
  - loss in pixel space requires trading off structural constraints on latent variables vs. accuracy of pixel-based reconstruction
  - ignoring visually small, but relevant features for predicting the future, such as a bullet in an Atari game
  - wasting model capacity on visually rich but otherwise potentially irrelevant features, such as static backgrounds.
- C-SWM learns to discriminate real vs fake experiences in the world in the form of state-action-state triples.
- Key advantages of C-SWM over prior-art:
  - Discriminative approach using contrastive learning is easier and avoids above-mentioned failure modes.
  - "think beyond autoencoder-based approaches for object-based, structured representation learning"
  - Learns a set of abstract state variables, one for each object in a particular observation.

C-SWM Review VII

- Environment transitions are modeled using a GNN that operates on latent abstract representations.
- a novel object-level contrastive loss for unsupervised learning of object-based representations
- learns to predict state transitions many steps into the future
- also proposes a novel ranking-based evaluation strategy
- learns an action-conditioned transition model of the environment that takes object representations and their interactions into account

- Possible issues in C-SWM:
  - Q: How to deal with variable number of objects? A: SQAIR [52], utilizing an iterative object encoding process such as in MONet [11].
  - Does not deal with stochastic environments. Assumes Markovian dynamics.

C-SWM Review VIII

- Does not naturally cater to continuous action spaces where multiple actions are performed on multiple objects at the same time.
- *The graph is assumed to be fully-connected with $O(K^2)$ edges which can be problematic if the number of abstract concepts K considered is large.
- Instance disambiguation: cannot disambiguate multiple instances of the same object present in one scene.
- *Action mapping to nodes in GNNs is not clear.
- Only a single object receives an action per time step. For the Atari environments, a copy of the one-hot encoded action vector is provided to every object slot, and for the 3-body physics environment, which has no actions, there is no action vector. Two concatenated consecutive frames are used as input for Atari and 3-body physics envs.

C-SWM Review IX

- *Assumes that a_t is known. For discrete action spaces, it further assumes the knowledge of the object on which the discrete action is performed.

- Questions and Thoughts:
  - C-SWM suggests that explicitly modeling the action space can be very valuable, and eases out the object disentanglement process.
  - The assumption that "a_t is known" can be restricting in practical settings. e.g. state-action-state triple can be known in RL settings from the experience buffer but not in videos.
  - C-SWM focuses on disentangling the object space but can we also disentangle the related action space?
  - Can the actions be treated as latent variables and be modeled in an identifiable manner as claimed in iVAE? Under what conditions is this possible? i. $\mathbf{f}$ has to be injective, ii... It should be possible to have an identifiable model over latent **objects as well as actions** given s_t and s_{t+1}

C-SWM Review X

- Going a step further, can the transition model be learned in the abstract space without explicit knowledge of actions? If we can do this, this will eventually help us in addressing the more challenging tasks (reasoning and counterfactuals) from e.g. CLEVRER [97].
- STOVE [53] works without knowledge of action but only for dynamic physics envs.
- How is **ST-DIM** [2] managing (constrastive learning) without explicit knowledge of actions? A: It is only learning state representations, not transition models.
- In real-valued time series, a simple approach to deal with non-stationarity is to take first-order differences. Difference across consecutive frames can improve the modeling of the action (dynamics) latent space. RW: Time-Contrastive Networks for videos (multi-view context) [80, 72], keypoint detection [65, 49].

C-SWM Review XI

- The only **generalization** C-SWM and other approaches [35, 52, 43, 53, 84] seem to enable (need to validate this) is to different states of the same set of objects. The action latent space should also be transferable to previously unseen new objects. How?
- Under what conditions can combinatorial generalization, transfer learning, or life-long learning be ensured? Can the theory of iVAE be extended along those lines? Ultimately, that is one of the key goals of disentangled representations. This may require bringing in ideas from causality.

C-SWM Review XII

- e.g. if a latent action and the duration for which its effect lasts can be identified, then it can help prediction in videos. Basically, if the **objects as well as actions are represented in the abstract space, then we need a "causal graph neural network" to process it.** Could not find much work in this area for videos: closest RW e.g. [28, 67], Bengio (2 months backs) **learning neural causal models from unknown interventions** [46], robustly disentangled causal mechanisms [86].
- Can disentangled representations help in transfer learning representations across Atari games to new objects and environments?

- C-SWM does struggle in Space Invaders and 3-body Physics experiments for longer-term predictions. Why?

C-SWM Review XIII

- C-SWM is better than PAIG [43], possibly because the loss function is entirely in abstract space in C-SWM. Even, STOVE [53] would have same limitation.
- What happens if new actions are introduced in the world and new objects are introduced, e.g. [97]?
- Markov assumption: RNNs can help but RIMs [29] or Competitive Ensembles [30] may do better.
- Instance disambiguation: need to dynamically bind individual objects to slots. Dynamic routing can help [75].

Bringing it all together...A possible direction to explore?

Proposal: Learning Structured World Models I

- Loss function in the **abstract concept space** [8], e.g. **Contrastive learning** [50]: enables **interpretable** independent (object-level) state abstractions.
- Graph Networks capture **relations** amongst disentangled abstract objects, allow **combinatorial generalization**.
- Using $< state, action, next\_state >$ is useful, and seems natural to build models of the world. How does it relate to the need of conditional factorial priors in [47]?
- C-SWM relies on Markov assumption. RIMs [29, 30] (also R-NEMs [88]?) enable modularity, **competitive ensembles** [30], non-Markovian dynamics, connection with **symbolic** knowledge representation via **naming variables**.
- Future: Disentanglement helps **life-long learning** [1].

Proposal: Learning Structured World Models II

- **Theory** [47]: Contrastive learning (TCL [40]) has connections to **causal** discovery [67], VAEs and (non-linear) ICA.
- Contrastive Learning of Disentangled Representations.
  - [47] has shown that disentangled representation is not possible without conditional factorization of priors.
  - Sequences/videos are common in real-world.
  - Disentangled factors can be learned by conditioning on the previous frame(s) and/or actions taken. Connected to TCL [40].
  - Can some theoretical claims be made on using information bottleneck as in RIMs [29] and Competitive Learning [30] to process videos?
  - Videos can also enable learning of causal factors (on datasets like CLEVRER [97]; not released yet).

Proposal: Learning Structured World Models III

- Explore connections with theory of contrastive unsupervised learning [5].
- What if there is noise or stochasticity in the environment?
  - Robust constrastive learning non-linear ICA under noise [77]

Unsupervised State Representation Learning (USRL): Capturing Latent Generative Factors of an Environment I

- Identifiable Hierarchical Latent Models via Contrastive Learning for USRL
- Existing approaches including C-SWM and ST-DIM assume a flat latent space.
- Learning hierarchical representations that disentangle blocks of variables whilst allowing for some degree of correlations within blocks is desirable [20, 36].
- Objects in a scene can have hierarchical relations or share common features inducing a hierarchy in the latent space - different objects made of same parts. What about action latent space?

Unsupervised State Representation Learning (USRL): Capturing Latent Generative Factors of an Environment II

- If the **objects as well as actions are represented in the abstract space, then we need a "causal graph neural network" to process it.**
- Rather than a flat latent space with independent components, a sparse graph may be more suitable. Can this be a causal graph as well where latent actions are the causes [46, 86], making sense of sensory input [21]?
- "Parts, Structure and Dynamics" [94] in Contrastive Learning setting with identifiability can be a good contribution.
- Need hierarchical GNN in latent space in C-SWM with sparse connections.

Unsupervised State Representation Learning (USRL): Capturing Latent Generative Factors of an Environment III

- Use cases in scene understanding, video prediction as in C-SWM, ST-DIM, CLEVRER. Can extend the data in C-SWM to have hierarchical relations. PSD is already doing it [94].
- The learned factors in latent space can be validated along the definitions of disentangled representations [18].
- To capture hierarchy, PSD [94] uses soft-edges, i.e. a fully-connected graph where the edge-weight determines the extent to which an object is a parent of another object.
- The motion encoder in PSD learns the latent action space.
- We can have attention bottleneck in GNN of C-SWM to impose hierarchy, similar to RIMs [29].
- In the context of hierarchy, what does named-variable idea of RIMs bring to the picture?

Unsupervised State Representation Learning (USRL): Capturing Latent Generative Factors of an Environment IV

- Action latent space equiv. to getting rid of the motion encoder in PSD.

Summary: Contrastive Learning of Latent Generative and Causal Dynamic Factors of An Environment

- Contrastive Learning [50, 2]
- Objects and Actions in Hierarchical Latent Space [94]
- Causal Graphs in Latent Space (causes: actions) [67, 46, 86, 21]
- Interactions processed via GNNs [50]
- With conditions for identifiability [47]

Q: Why is this an important problem?

- Videos are readily available
- Labels for objects and actions are difficult to capture
- Building blocks across objects can be same: requires hierarchy
- Capturing cause-effect relationship between actions and resulting dynamics can help with reasoning, counterfactual understanding.

Potential benchmarking on tasks from: i. C-SWM [50], ii. ST-DIM [2], iii. PSD [94], iv. CLEVRER [97], v. Bouncing balls [88, 29].

Refined Problem Statement

Problem Statement

Unsupervised Learning of Latent Generative Factors of an Environment

1. a structured, hierarchical state representation
2. characterization of system dynamics

Approach

- **Theory**: "Under what conditions and assumptions" is this possible?
- **Inductive Bias**: A sparse (causal?) graph (in sense of RIMs and GNNs) in latent space with attention-based edge-weights; implicitly learns hierarchies (b/w actions, dynamics, objects). NSMs-like mechanism to enable directional flow of attention.
- **Loss**: Contrastive, as in C-SWM and ST-DIM.
- **Evaluation**: Toy tasks in C-SWM, PSD and RIMs/R-NEM can help with initial evaluation, followed by Atari environments of ST-DIM. SQAIR [52], CLEVRER [97], CATER [26].

RW: C-SWM [50], ST-DIM [2], PSD [94], Sparse Graphs (RIM [29], WMG [61], NSM [38]), Hier.Disent. [20], Neural Causal [9, 46].

Source Codes

- C-SWM `https://github.com/tkipf/c-swm`
- ST-DIM `https://github.com/mila-iqia/atari-representation-learning`
- PSD `https://github.com/zhenjia-xu/psd`

Useful Datasets

Apart from datasets used in [50, 88, 29], following datasets may be useful:

- CLEVRER[2] [97]: Collision events for video representation and reasoning
- CATER[3] [26]: A diagnostic dataset for Compositional Actions and TEmporal Reasoning
- Disentanglement Dataset[4] [27]
- GQA [39]: dataset for real-world visual reasoning and compositional question answering
- CLEVR [45]: A diagnostic dataset for compositional language and elementary visual reasoning
- VCR [101]: Visual Commonsense Reasoning
- Testing generalization [24]

[2] http://clevrer.csail.mit.edu/
[3] http://rohitgirdhar.github.io/CATER
[4] https://github.com/rr-learning/disentanglement_dataset

Towards Disentangled Representations for Time Series I

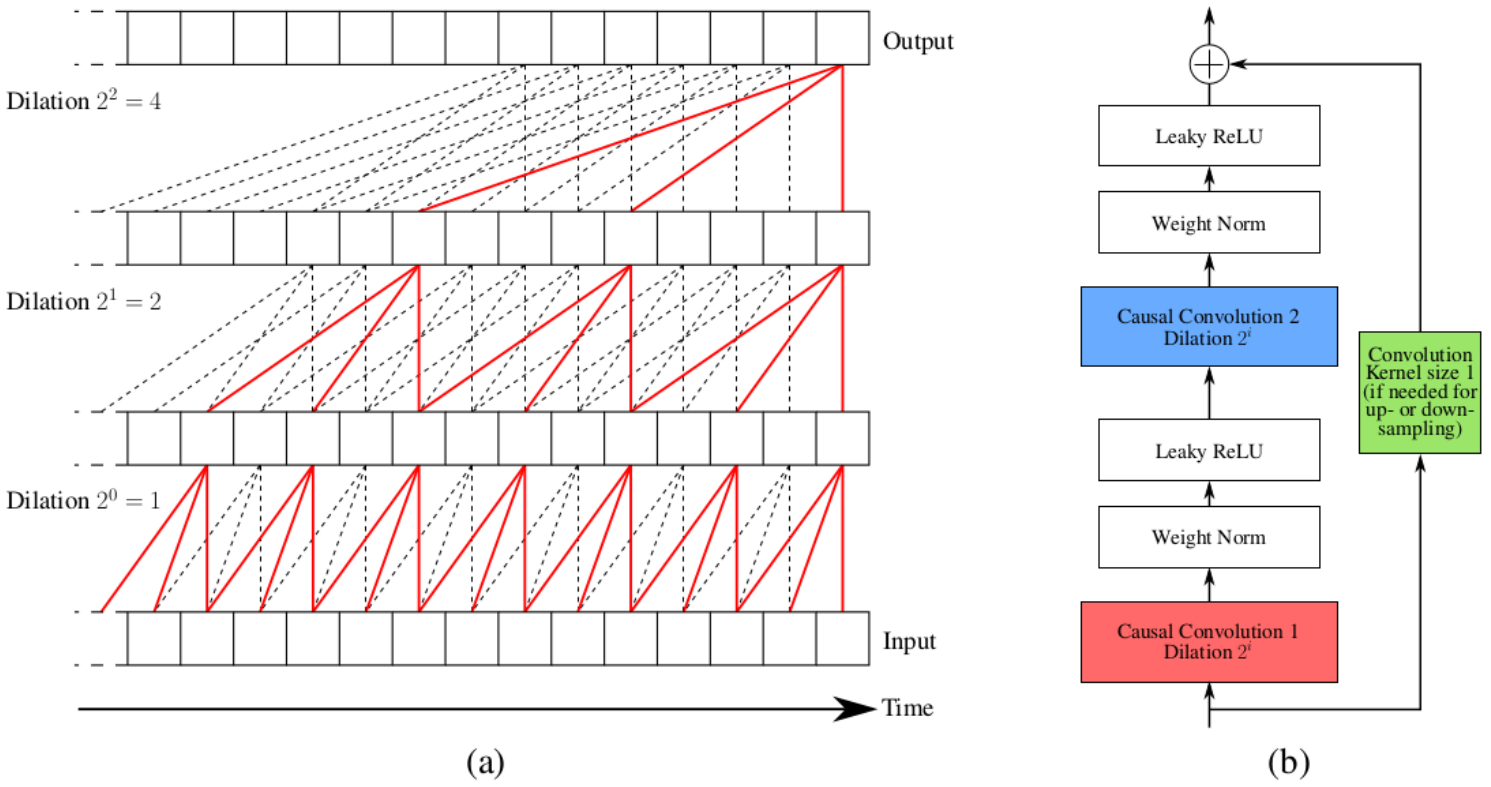


Figure 2: (a) Illustration of three stacked dilated causal convolutions. Lines between each sequence represent their computational graph. Red solid lines highlight the dependency graph for the computation of the last value of the output sequence, showing that no future value of the input time series is used to compute it. (b) Composition of the i-th layer of the chosen architecture.

Towards Disentangled Representations for Time Series II

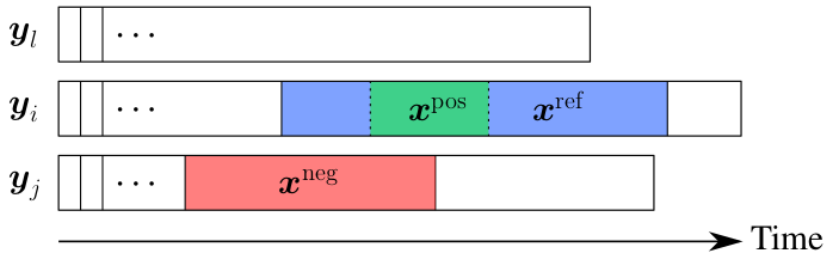


Figure 1: Choices of x^{ref}, x^{pos} and x^{neg}.

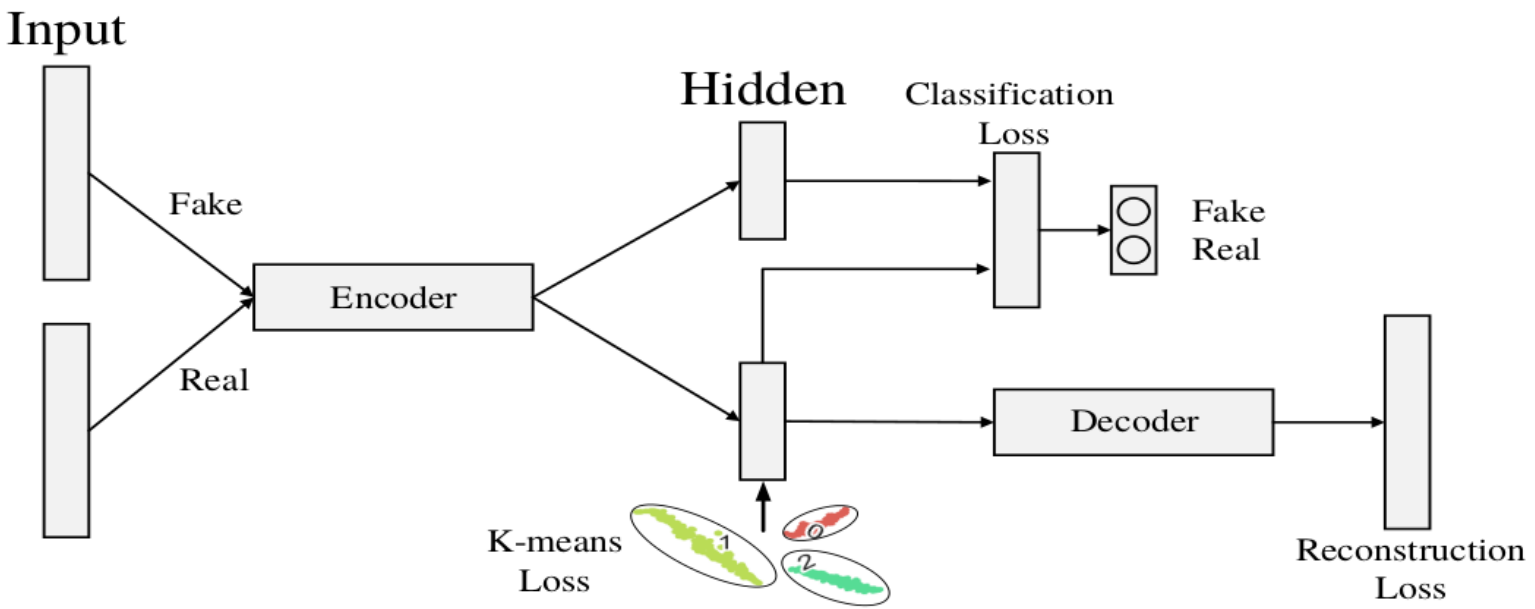


Figure 1: The general architecture of the Deep Temporal Clustering Representation (DTCR).

Towards Disentangled Representations for Time Series III

- contrastive learning for time series representations (**CLTS**) [22]; uses dilated causal CNNs
- most DL approaches for time series fail to capture high-frequency features that are important. can we prove this first on a toy dataset first?
- local-local infomax from ST-DIM [2] may help overcome this issue as it tends to capture useful features such as those from small objects.
- contrastive learning may be sub-optimal in finite-data scenarios if contrastive learning is possible without learning the true underlying distribution [47]. we don't want to learn the true underlying distribution - we only want to learn those aspects which can be interacted with and lead to changes in the environment, and those aspects of the environment which help to determine this change - rest is all static background.

Towards Disentangled Representations for Time Series IV

- why does $\mathbf{x}^{pos}$ have to be a subsequence of $\mathbf{x}^{ref}$?
- So iVAE [47] and iFlow [56] should have a role to play because they learn the underlying distribution. but rather than learning the entire distribution, learn **conditional distributions**.
- an empirically observed failure case of unsupervised methods (potentially including contrastive methods) for time series is in capturing useful high-frequency features. e.g. The output of TimeNet decoder is a smoothed version of the input time series.
- Note: ST-DIM works but not because of InfoMax [87] but because of other inductive biases. **Wasserstein Predictive Coding** is an alternative [70] - it again implicitly highlights the importance of local-local infomax, i.e. just a global or global-local infomax is not sufficient.

Towards Disentangled Representations for Time Series V

- does all this help with forecasting? c-swm can play a role - days/timestep shift is the action.
- difference in timesteps can allow for better capturing the notion of time between $\mathbf{x}^{pos}$ and $\mathbf{x}^{ref}$. this will be useful especially for **forecasting** but maybe also for **classification**. for forecasting, we can even draw inspirations from contrastive predictive coding paper [69].
- fake sample generation from **DTCR** [62] can also provide useful $\mathbf{x}^{neg}$.
- $\mathbf{x}^{neg}$ taken from another time series of the same class can actually hurt performance on downstream tasks such as clustering and classification.
- if we achieve disentanglement, we should do well on **clustering** task also from DTCR [62] on the same datasets.

Towards Disentangled Representations for Time Series VI

- interaction of disentangled representations modeled via a GNN can help with forecasting.
- reconstruction error loss is also related to mutual information [34], so we don't really need a decoder if using ST-DIM?
- theory: non-stationarity helps with causal discovery [37, 40]. same should happen within a time series as well?
- need to understand linear probing for testing disentanglement.
- in the multivariate setting, does this also help with causal discovery, e.g. in activity recognition?
- multivariate time series clustering using contrastive learning is a research gap.
- from iVAE and iFlow point-of-view, we try to reconstruct $\mathbf{x} = \mathbf{x}_{t+1}$ given $\mathbf{u} = \{\mathbf{x}_{t-k}, k\}$.

Towards Disentangled Representations for Time Series VII

- what about the scenario where a small number of labeled instances are available, such that for few instances $\mathbf{u}$ can be set according to the class label?
- if given the dependent variables only, can the causal independent variables be learned?
- if given the actions and the dependent variables, we can learn disentangled state representations.
- in univariate setting, disentangled representations would be equivalent to feature maps that correspond to shapelets that characterize the time series, and their positions in a given time series.
- **RW**: time series forecasting with exogeneous variables [59]. datasets: GHL, Electricity, TEP, SWaT, SMD (Server Machine Dataset), SMAP (Soil Moisture Active Passive satellite) and MSL (Mars Science Laboratory rover).

Invertible Neural Networks

- The theory of iVAE and iFlow assumes f (the mapping from $\mathbf{z}$ to $\mathbf{x}$) to be invertible.
- i-RevNet [42] is an instance of a deep neural network that guarantees invertibility.
- These approaches have also been motivated for inverse problems [4] in natural sciences.
- Another CrevNet - Two-way autoencoder for video prediction [100].

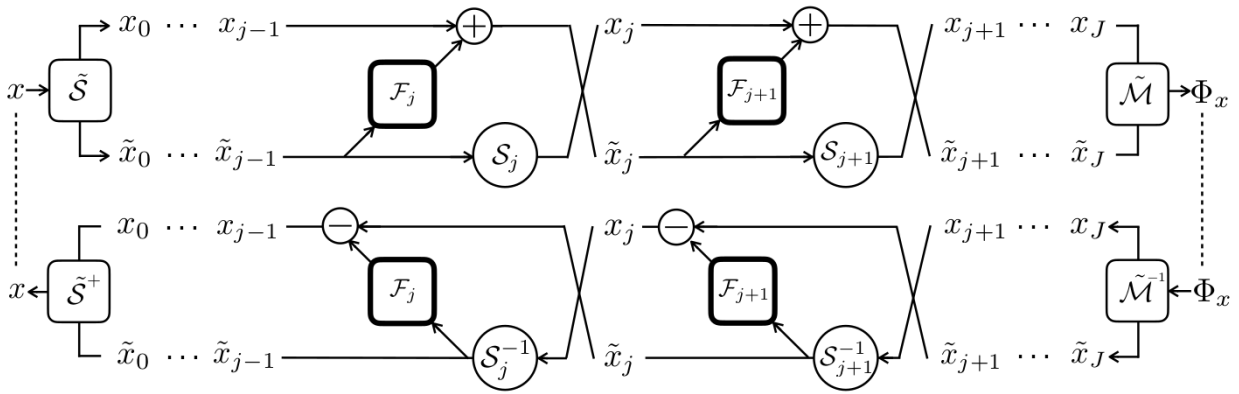


Figure 1: The main component of the i-RevNet and its inverse. RevNet blocks are interleaved with convolutional bottlenecks $\mathcal{F}_j$ and reshuffling operations $\mathcal{S}_j$ to ensure invertibility of the architecture and computational efficiency. The input is processed through a splitting operator $\tilde{\mathcal{S}}$, and output is merged through $\tilde{\mathcal{M}}$. Observe that the inverse network is obtained with minimal adaptations.

CLTS - univariate I

Since experimenting on all datasets is taking time, we can try with few select datasets to begin with e.g. TwoPatterns, Coffee, ECG200, DiatomSizeReduction, etc. Refer Fig. 3 in USSL paper [102] for the kind of shapelets expected.

1. Pooling in final layer may be interfering with the ST-DIM like losses. Try replacing it with an MLP similar to C-SWM - same MLP to process each feature map.
2. Ideally, we should not require 160/320 filters/channels in the final layer for most datasets in UCR. If the algo can actually discover useful Shapelets, 10-20 filters should suffice for most datasets [102].
3. Try only GG loss first - should observe similar results with much lesser number of (10-20) filters. Weights to be given to GG and GL losses based on their initial magnitudes. Ignore LL loss for above points.

CLTS - univariate II

4. Look at the 10-20 feature maps after addressing points 1-3. They should relate well to the known shapelets in TwoPatterns and ECG datasets.
5. Regarding defining $\mathbf{x}_{ref}$, $\mathbf{x}_{pos}$, and $\mathbf{x}_{neg}$
   - It is problematic in class imbalance datasets.
   - Shifting to get $\mathbf{x}_{pos}$ for a given $\mathbf{x}_{ref}$ would result in a similar shift in feature map affecting Local-Local DIM. Can shift the feature map also by an equal amount - need to think about padding etc. in that case.
6. Another useful task: Use transforms in frequency domain**. Given a pair of time series - <original, transformed>, task can be to identify the frequencies affected. Or, given a time series and an action to change its frequency, get the representation of the transformed time series in C-SWM-like manner.

CLTS - univariate III

7. Weights to be given to GG, GL and LL losses based on their initial magnitudes.
8. Why LL loss defined only over last layer's feature maps? Can be imposed on all layers.

Contrastive Learning for MTS Forecasting

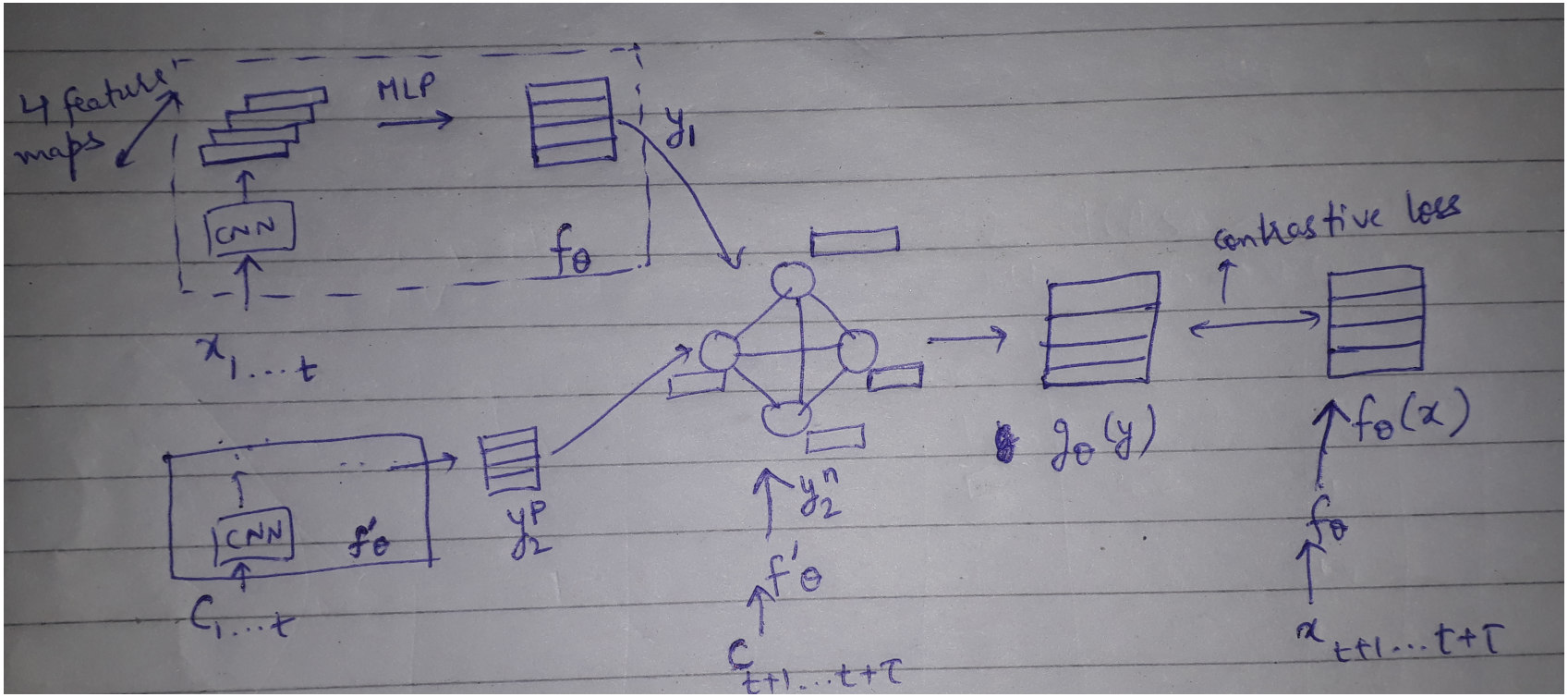


Figure: Illustrative flow diagram

Q: what would be a good decoder? hints from NRI [51]?

Representation Learning with Contrastive Predictive Coding

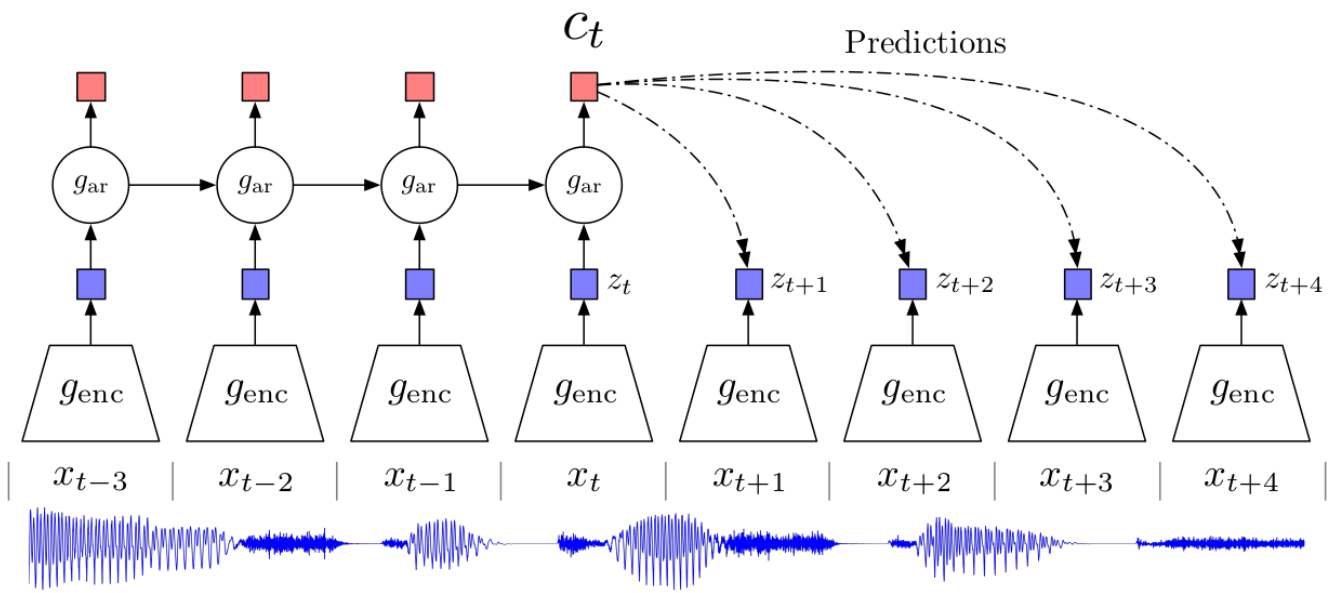
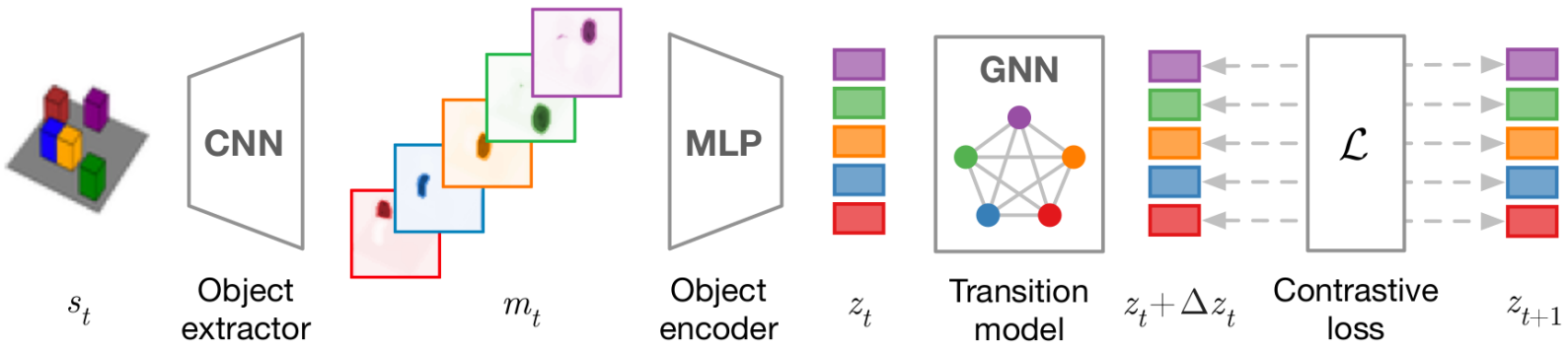


Figure 1: Overview of Contrastive Predictive Coding, the proposed representation learning approach. Although this figure shows audio as input, we use the same setup for images, text and reinforcement learning.

Contrastive Predictive Coding [69]

C-SWM (revisited) [50]

Contrastive Learning of Structured World Models (C-SWM) [50]



- a CNN-based object extractor: *stimulus-response*
- an MLP-based object encoder: *compositional*
- a GNN-based relational transition model: *interaction understanding, compositional*
- an object-factorized contrastive loss: *learning in abstract space*

Key: Loss is defined in the abstract space!

Learning Predictive Representations using Contrastive Estimation [95]

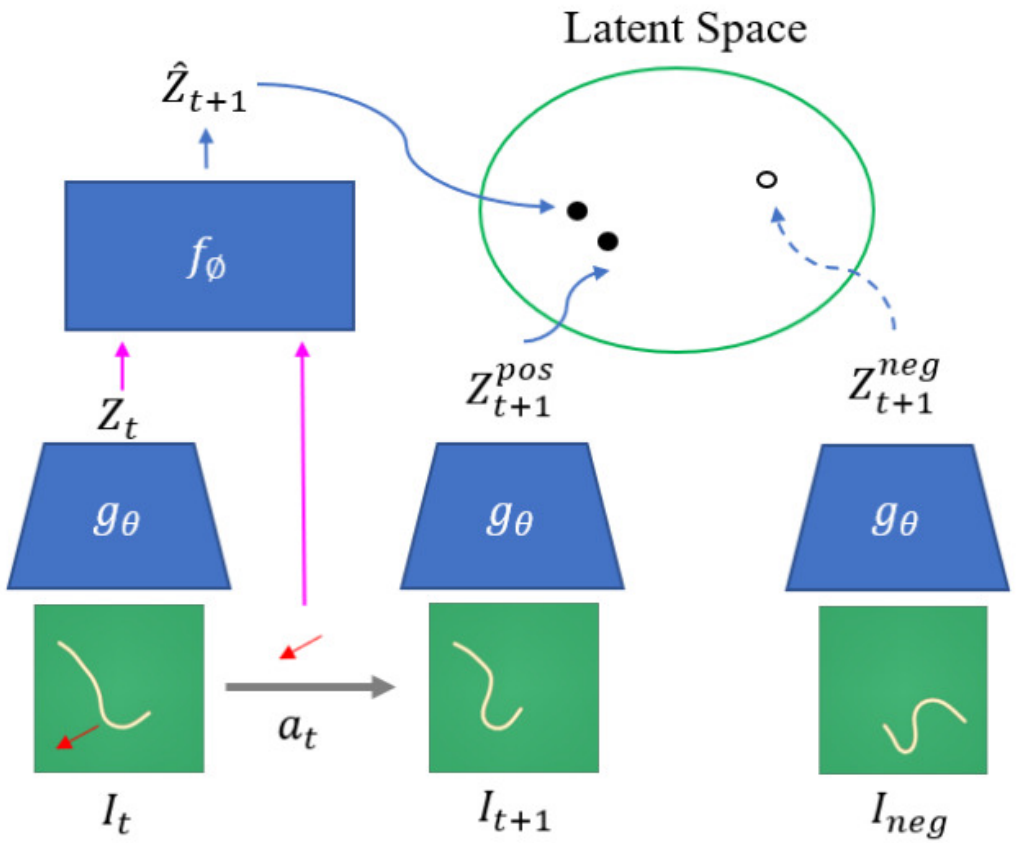


Fig. 2: Overview of our contrastive forward model. Training data consists of (image, next image, action) tuples and we learn the encoder and forward model jointly. The contrastive loss objective brings the positive embedding pairs closer together and the negative embeddings further away.

Proposal

- **Problem Statement**: System Identification from multivariate time series (MTS) data.
- **Motivation**: Unsupervised discovery of i. sub-systems, ii. their interactions, and iii. their dynamics in the latent space rather than in the input space.
- **Task**: **in**: $\mathbf{x}_{1...t}, \mathbf{c}_{1...t+\tau}$, **out**: $\mathbf{x}_{t+1...t+\tau}$
- **Relevant Lit.**: C-SWM [50], CPC [69], NRI [51], PSD [94], CFM [95], Plan2vec [96].
- **Theory**: iVAE [47], iFlow [56], GIN [82], ICE-BeeM [48], Abstract MDPs [73], Homomorphisms [91].
- **Architecture**: i. 2 1D-CNNs for processing $\mathbf{x}$ and $\mathbf{c}$, ii. GNNs for modeling interactions.
- **Baselines**: i. **Traditional**: VARMAX, SVR, GPR; ii. **DL**: NARX, LSTNet [54], DyAtNets [68], PR-SSM [19].
- **Datasets**: TEP, SWaT, GHL, DaISy, Helicopter, Quadrotor, Neuronal Dynamics, SMAP, MSL.

Example Application: Prediction and Control with Temporal Segments [66]

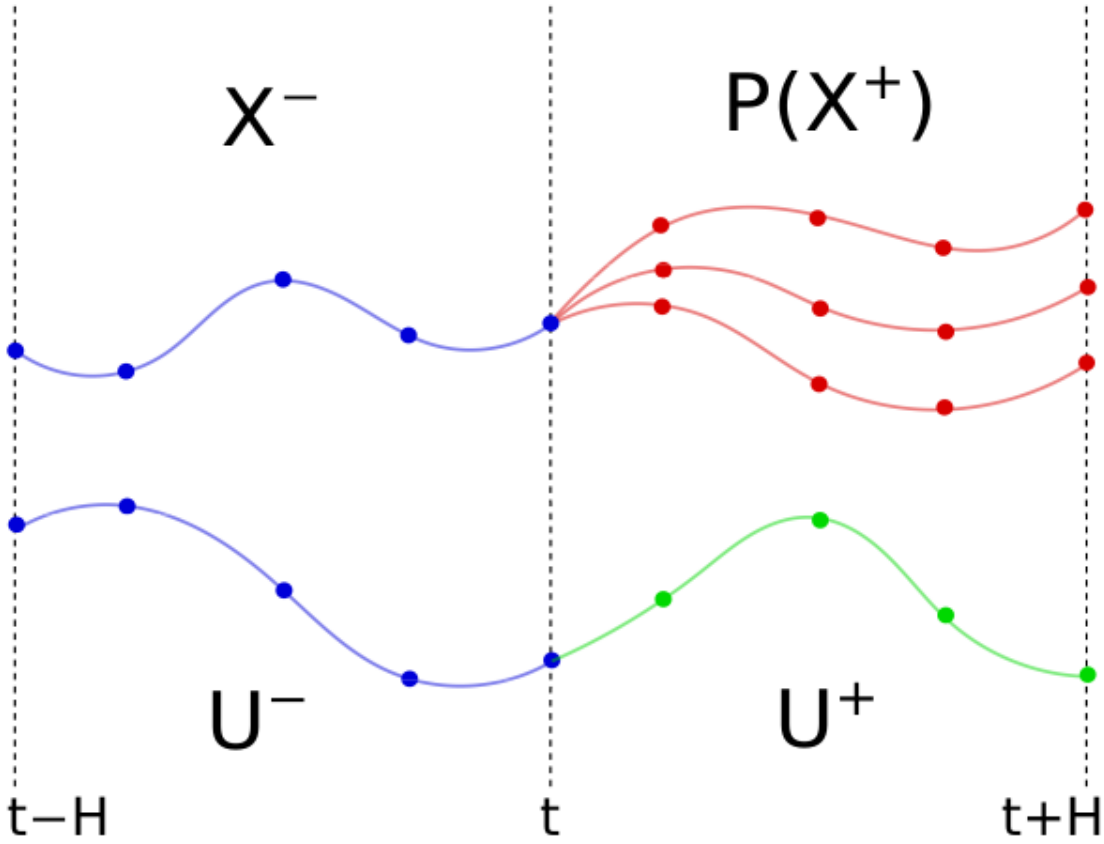


Figure 1. An overview of the probabilistic model we wish to learn. Given observed past states and actions X^-, U^- (blue), and planned future actions U^+ (green), we wish to sample possible future state trajectories X^+ (red).

Task and Architecture I

Using notation from ICE-BeeM [48]:
in: $\mathbf{y} = \mathbf{x}_{1...t}, \mathbf{c}_{1...t+\tau}$, **out**: $\mathbf{x} = \mathbf{x}_{t+1...t+\tau}$
$\mathbf{x}_i \in \mathbb{R}^m$, $\mathbf{c}_i \in \mathbb{R}^n$, Total dimensions: $m + n$, Forecasting Horizon τ.
Define conditional energy function:

$$\mathcal{E}_\theta(\mathbf{x}|\mathbf{y}) = \mathbf{f}_\theta(\mathbf{x})^T \mathbf{g}_\theta(\mathbf{y}) \tag{1}$$

Family of conditional energy-based models has the form:

$$p_\theta(\mathbf{x}|\mathbf{y}) = \frac{exp(-\mathbf{f}_\theta(\mathbf{x})^T \mathbf{g}_\theta(\mathbf{y}))}{Z(\mathbf{y};\theta)} \tag{2}$$

Estimate the model using contrastive learning, e.g. FCE [23].
Drawing parallels to C-SWM [50]:
$\mathbf{f}_\theta$ is a 1D-CNN

$$\mathbf{f}_\theta(\mathbf{x}) = \mathbf{f}_\theta(\mathbf{x}_{t+1...t+\tau}) \tag{3}$$

Task and Architecture II

$\mathbf{g}_\theta$ consists of three parts:
i. $\mathbf{f}_\theta$ to process $\mathbf{x}_{1\ldots t}$:

$$\mathbf{h}_x^p = \mathbf{f}_\theta(\mathbf{x}_{1\ldots t}) \tag{4}$$

ii. another 1D-CNN to process $\mathbf{c}_{1\ldots t+\tau}$:

$$\mathbf{h}_c^p = \mathbf{f}'_\theta(\mathbf{c}_{1\ldots t}), \quad \mathbf{h}_c^n = \mathbf{f}'_\theta(\mathbf{c}_{t+1\ldots t+\tau}) \tag{5}$$

p : *past*, n : *next*
iii. a GNN to process $\mathbf{h}_x^p$ and $\mathbf{h}_c^p, \mathbf{h}_c^n$ to get final output

$$\hat{\mathbf{h}}_x^n = \mathbf{g}_\theta(\mathbf{y}) = GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n) \tag{6}$$

$$\mathbf{h}_x^n = \mathbf{f}_\theta(\mathbf{x}_{t+1\ldots t+\tau}) \tag{7}$$

Here, we have a sparse graph over the dimensions of $\mathbf{h}_x$.

Task and Architecture III

$$\mathcal{L} = d(\mathbf{h}_x^n, \hat{\mathbf{h}}_x^n) + \sum_K max(0, \gamma - d(\hat{\mathbf{h}}_x^n, \mathbf{h}_-)) \tag{8}$$

Can we show equivariance under actions for Eq. 6 as in Abstract MDPs [73]? i.e. $T(\mathbf{x}; \mathbf{y}) \equiv T'(\mathbf{h}_x^n; \mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n)$

Task and Architecture IV

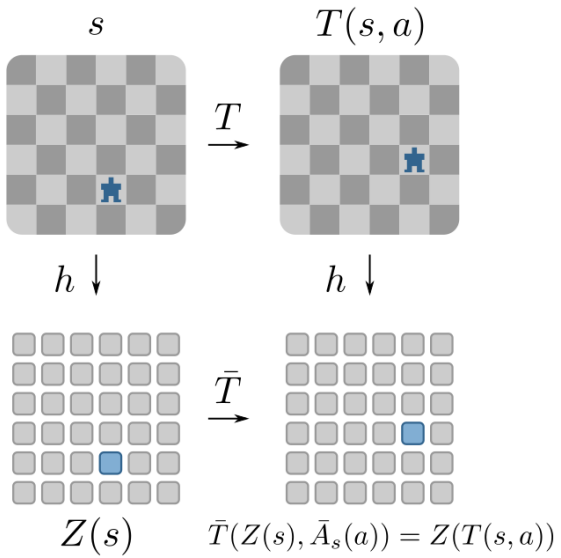


Figure 1: Visualization of the notion of equivariance under actions. We say Z is an action equivariant mapping if $Z(T(s,a)) = \bar{T}(Z(s), \bar{A}_s(a))$.

Contrastive Learning for Multivariate Time Series Forecasting (CL4MTS): Architecture Diagram (Old)

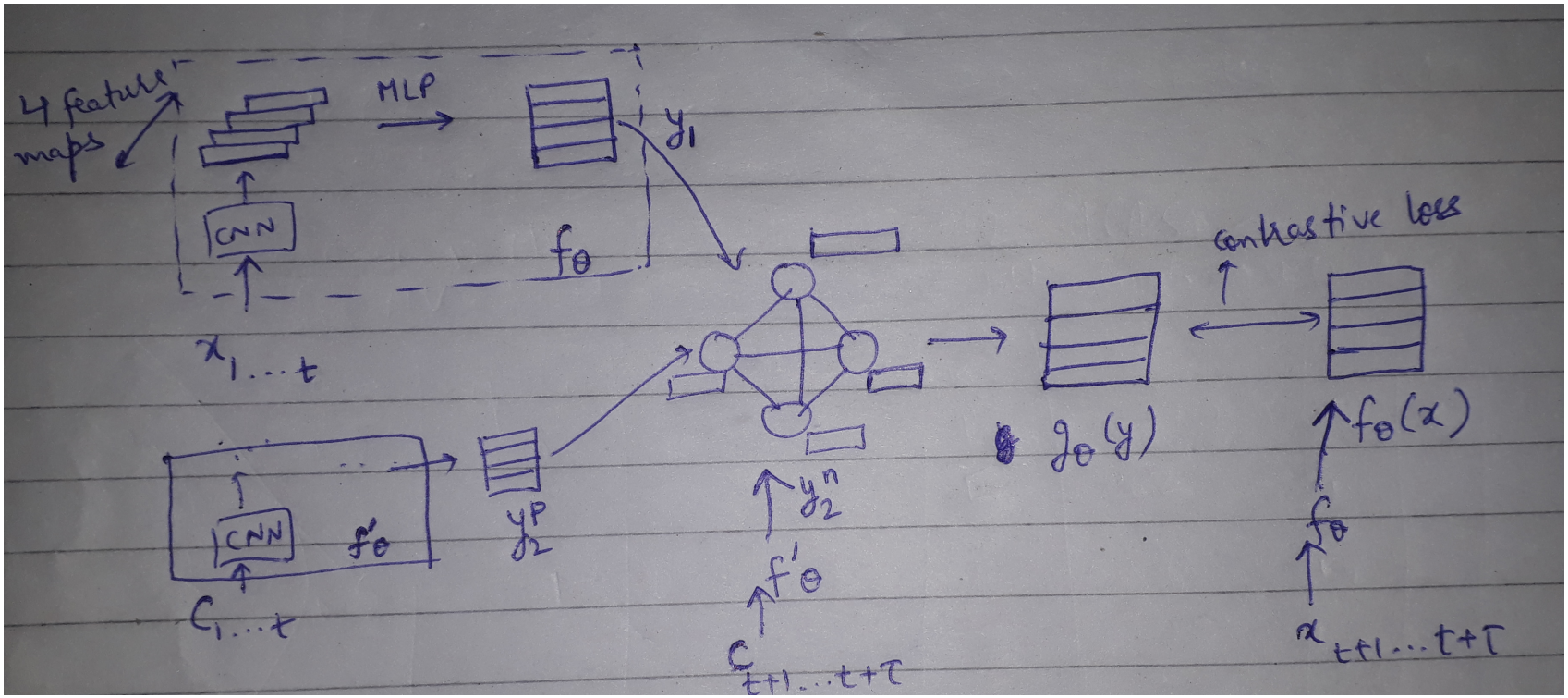
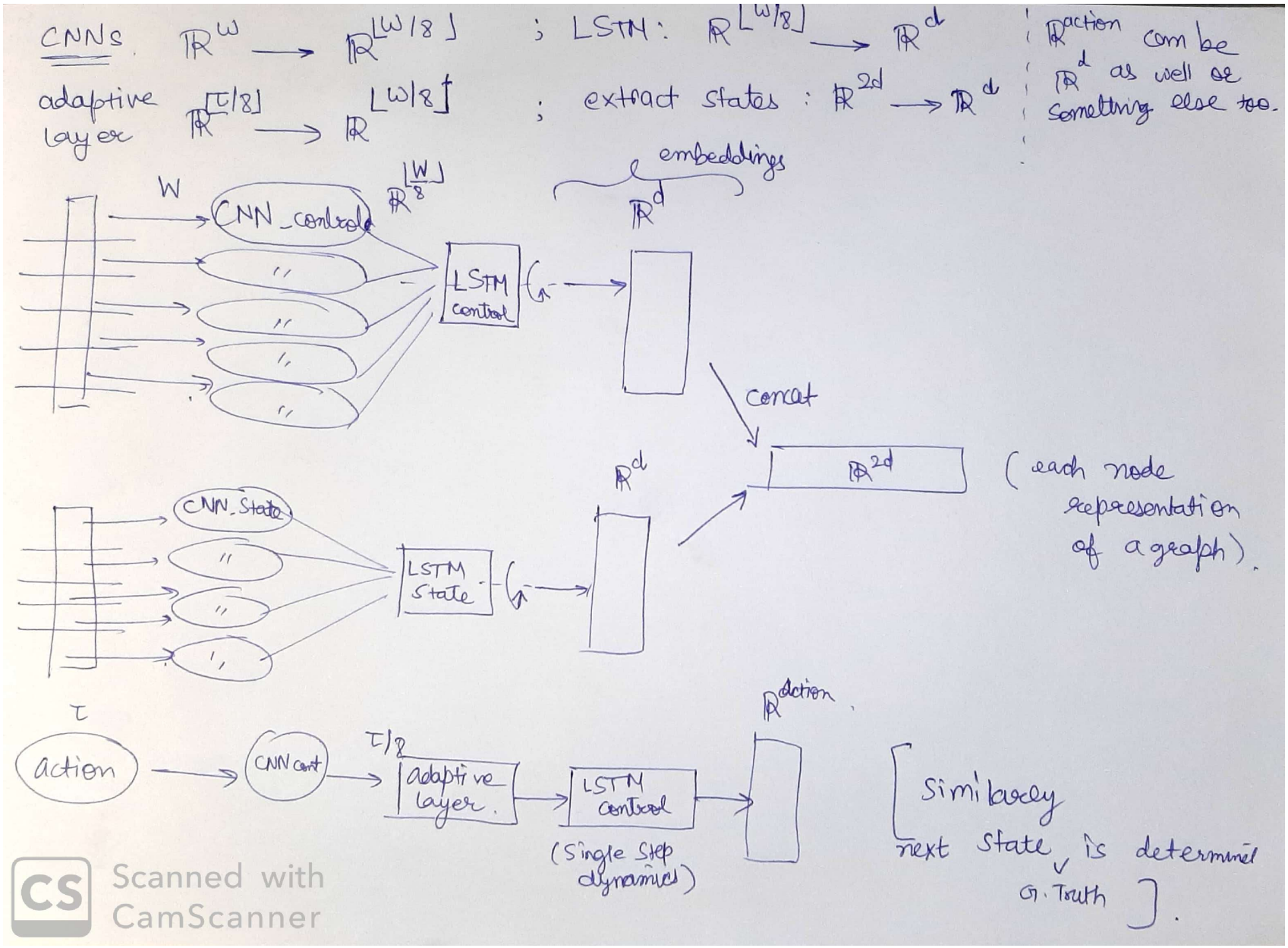


Figure: Illustrative flow diagram

CL4MTS: Architecture-I (Current)



CL4MTS: Architecture-II (Current)

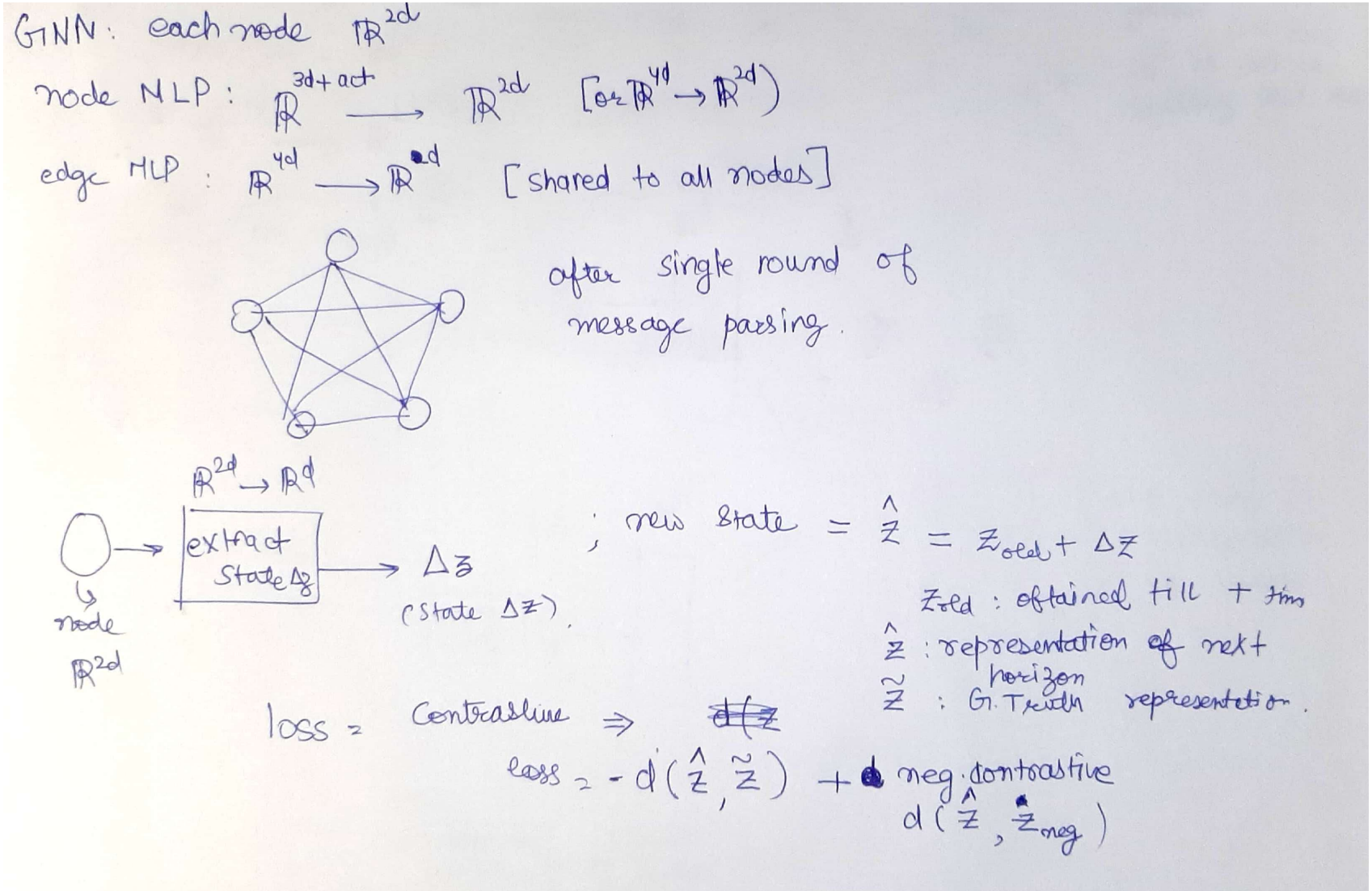


Figure: Illustrative flow diagram

Key features of proposed approach

- MLP allows for **compositionality** - extracting same features from each feature map, e.g. position of sharp rise.
- **Current** state of the system is summarized in the form of a **graph** - each **node** potentially caters to a **sub-system** of the overall complex system, each **edge** models the **interactions** across sub-systems. e.g. each node tracking a joint in a multi-DOF robotic arm. so each node carries info. of all sensors (pos., vel, acc.) for a joint.
- **Future** control variables manipulate the node representations to estimate the representation of future dependent variables. e.g. torque applied to a joint affects values of all sensors (pos., vel, acc.).
- **Contrastive learning** is used to define a loss function in the **latent space**.

Hypotheses I

- In comparison to loss in original time series space, the loss in the latent space is:
  - better at forecasting dependent variables
  - better generalization to longer forecasting horizons.
  - better at nowcasting/estimating dependent variables not used for training. ideally, by learning simple linear models over the representations.
  - leads to robust (identifiable?) representations: faster in adapting to forecast dependent variables not seen during training.
- Having manipulated variables as conditional variables leads to better models than ones relying only on dependent variables.
- Disentanglement: Able to recover manipulated variables from dependent variables.

Hypotheses II

- Graph structure over the latent variables helps, i.e. dependent latent variables sharing info. with each other are better than ones not sharing info.
- better at combinatorial generalization when a new combination of the underlying sub-systems is in place. e.g. the set-points of a sub-system are changed.
- given only the dependent variables, better at classifying which manipulated variable was changed.
- modeling all dependent variables simultaneously helps in learning robust representations.

Datasets

Dataset	m	n	ν	L	N	*Full*	Modes
TEP	44	12	1000/h	0.12M	346	T	7 ($\times$4)
GHL	5*	5*		1.5M	1	F	
SWaT	24	27	1/s	0.5M	1	T	
Helicopter	25*	4	50/s*	0.1M	20	F	20
Quadrotor	15	8	100/s	25k	54	T	> 5
Neuronal Dyn.							
Sarcos	21	7	100/s	45k	1		
steamgen	4	4		9.6k	1		
cstr	2	1		7.5k	1		
evaporator	3	3		6.3k	1		
winding	2	5		2.5k	1		
flex. struct.	28	2		8.5k	1		
CD Player Arm	2	2		2.0k	1		
Foetal ECG	8	-		2.5k	1		
Robot Arm	1	1		1k	1		

m: dependent variables, n: independent variables, ν: sampling rate, L: length of each time series, N: number of time series, *Full*: All control variables available. *approx. (not certain).

Experiments Done

- Comparing Contrastive Learning in Abstract Space vs MSE-based Learning in Original Space:
  - **CL**: Contrastive Learning of encoder, followed by MSE loss based learning of decoder.
  - **MSE**: End-to-end learning of encoder and decoder using MSE loss.
- Dataset considered: Sarcos 7DOF Robotic Arm [19]

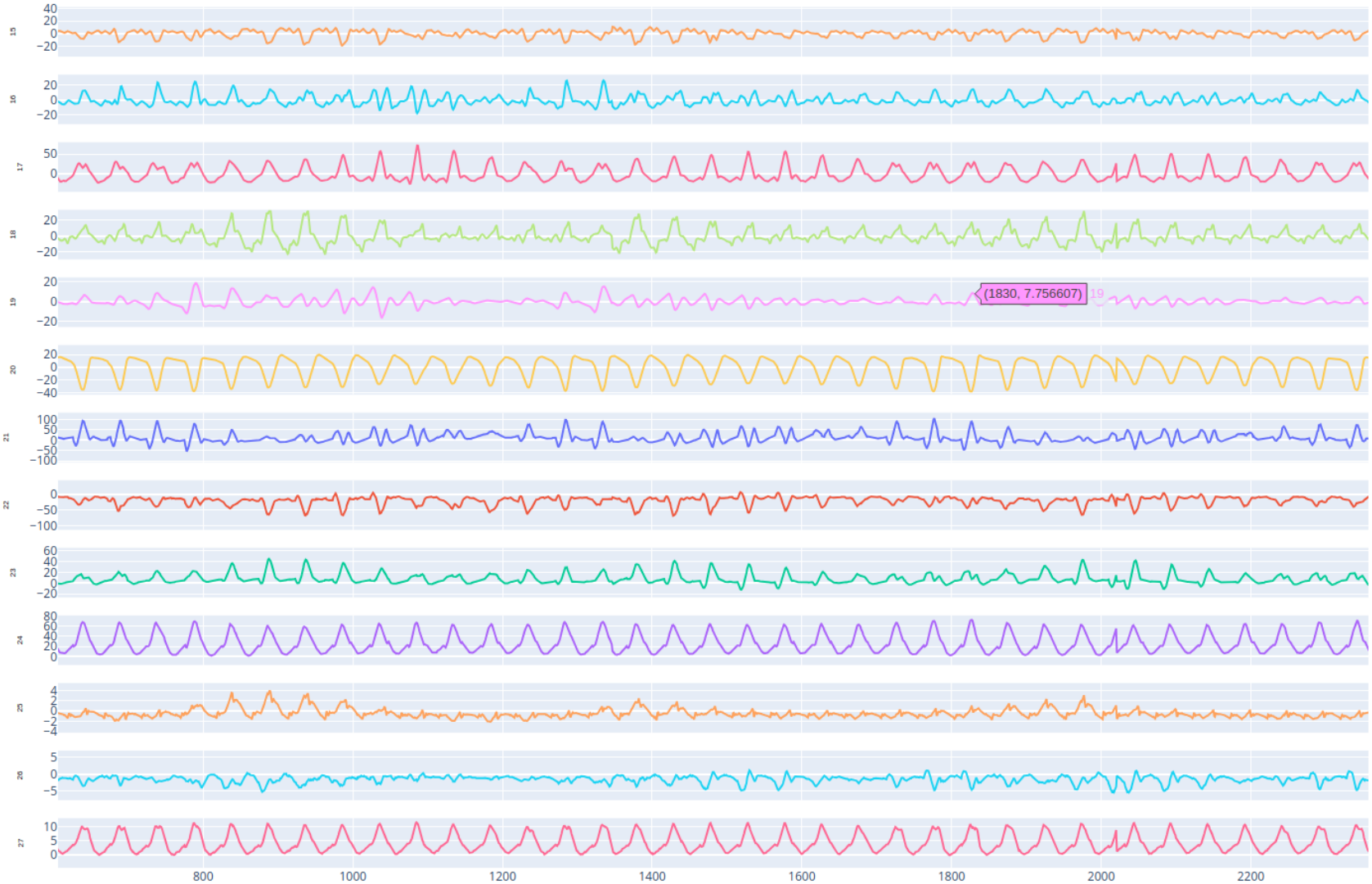
Results: CL vs MSE

						CL: Contrastive Learning followed by Decoder Training						MSE Loss (end2end training)			
t	sub-seq length	tau (horizon)	GNN Nodes	Embedding Dims	Max Epochs	H@1	MRR	Valid MSE Loss	Best Epoch Encoder	CL Loss	Best Epoch Decoder	H@1	MRR	Valid MSE Loss	Best Epoch
500	100	100	3	8	100	1	1	0.0424	78	0.0253	98	0.0057	0.0498	0.0191	100
				16		1	1	0.0442	78	0.0248	98	0.0057	0.0413	0.0148	100
				32		1	1	0.0439	24	0.0208	84	0.0057	0.0663	0.0134	98
				64		1	1	0.0495	78	0.0240	84	0.0057	0.0333	0.0177	100
			7	8		1	1	0.0385	78	0.0303	99	0.0170	0.0732	0.0134	100
				16		1	1	0.0382	30	0.0275	90	0.0057	0.0692	0.0152	100
				32		1	1	0.0397	30	0.0255	85	0.0114	0.0513	0.0134	100
				64		1	1	0.0432	24	0.0196	98	0.0000	0.0320	0.0129	100
			11	8		1	1	0.0401	78	0.0285	97	0.0114	0.0394	0.0105	100
				16		1	1	0.0437	78	0.0240	58	**0.0568**	**0.1447**	0.0112	100
				32		1	1	**0.0357**	24	0.0230	99	0.0170	0.0545	0.0090	98
				64		1	1	0.0367	24	**0.0189**	96	0.0114	0.0376	0.0112	100

Figure: Preliminary Results on Sarcos Dataset

702 training windows, 176 validation windows, each of length τ =100, with future time series of 21 dependent variables to be estimated given i. past time series of length 500 for the 21 dependent and the 7 control variables, and ii. the future time series of length 100 for the 7 control variables.

Sarcos dataset: zoomed-in view (2k points)



Results: CL vs MSE, Qualitative Analysis

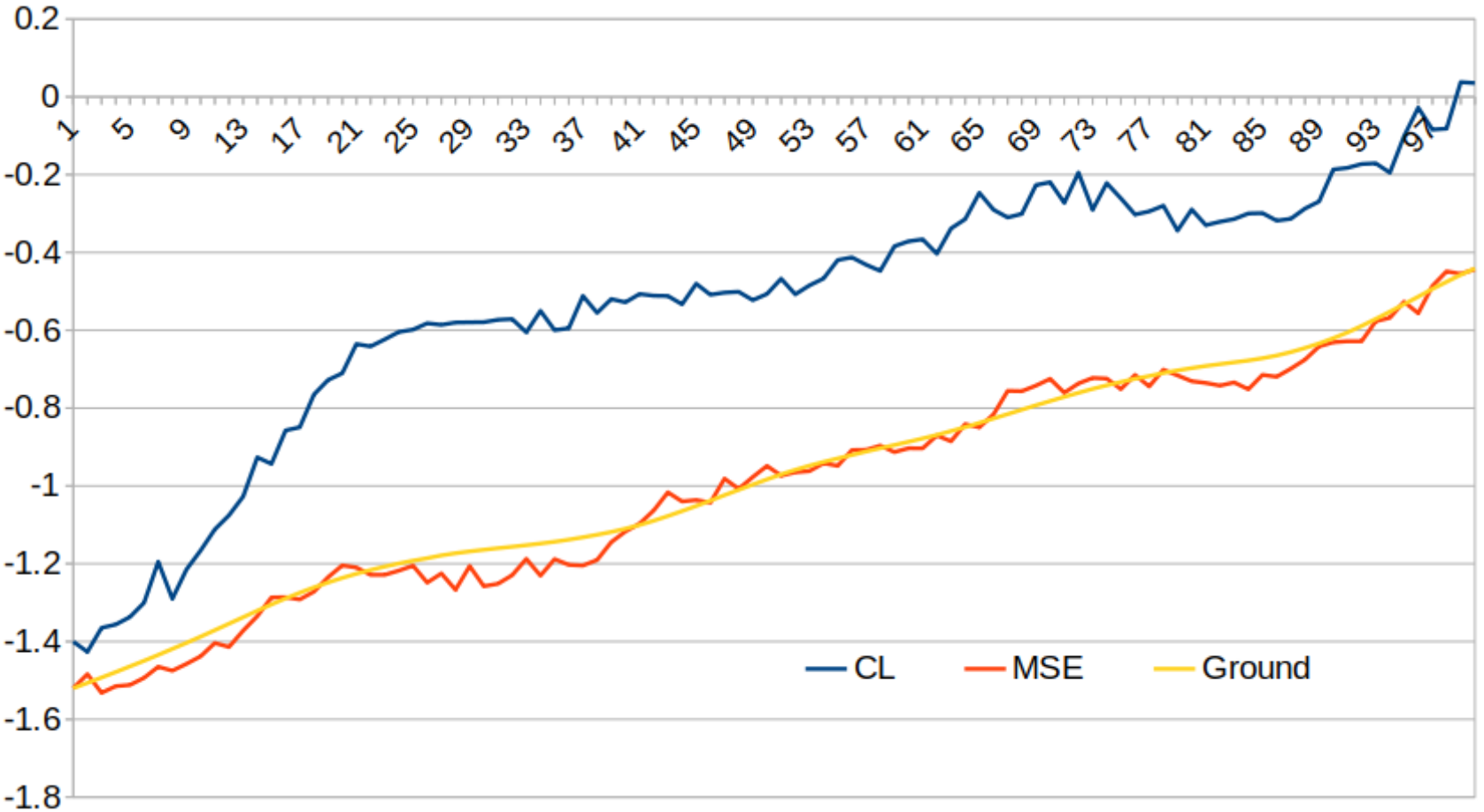


Figure: Sarcos Dataset: 1st dependent variable

Results: CL vs MSE, Qualitative Analysis

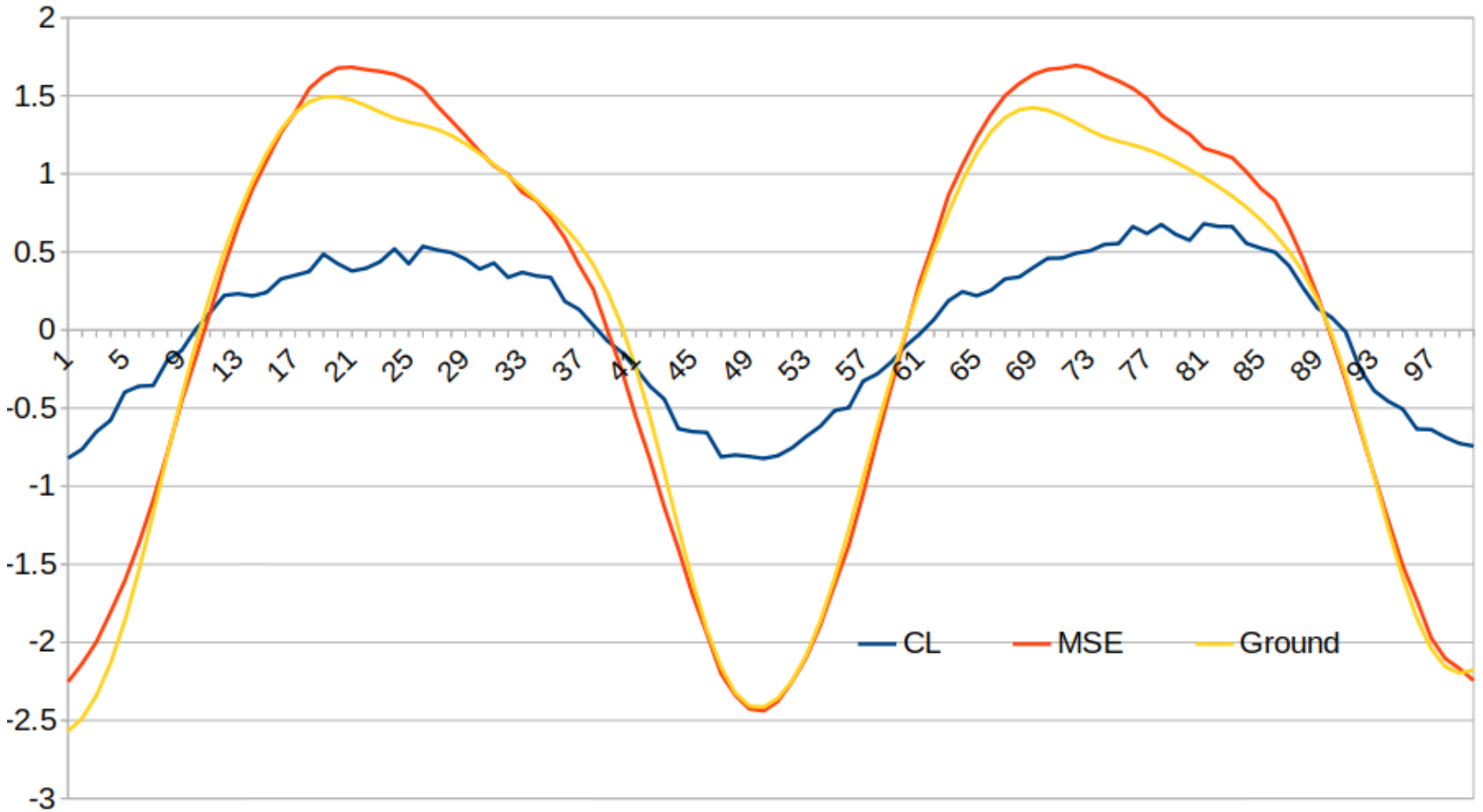
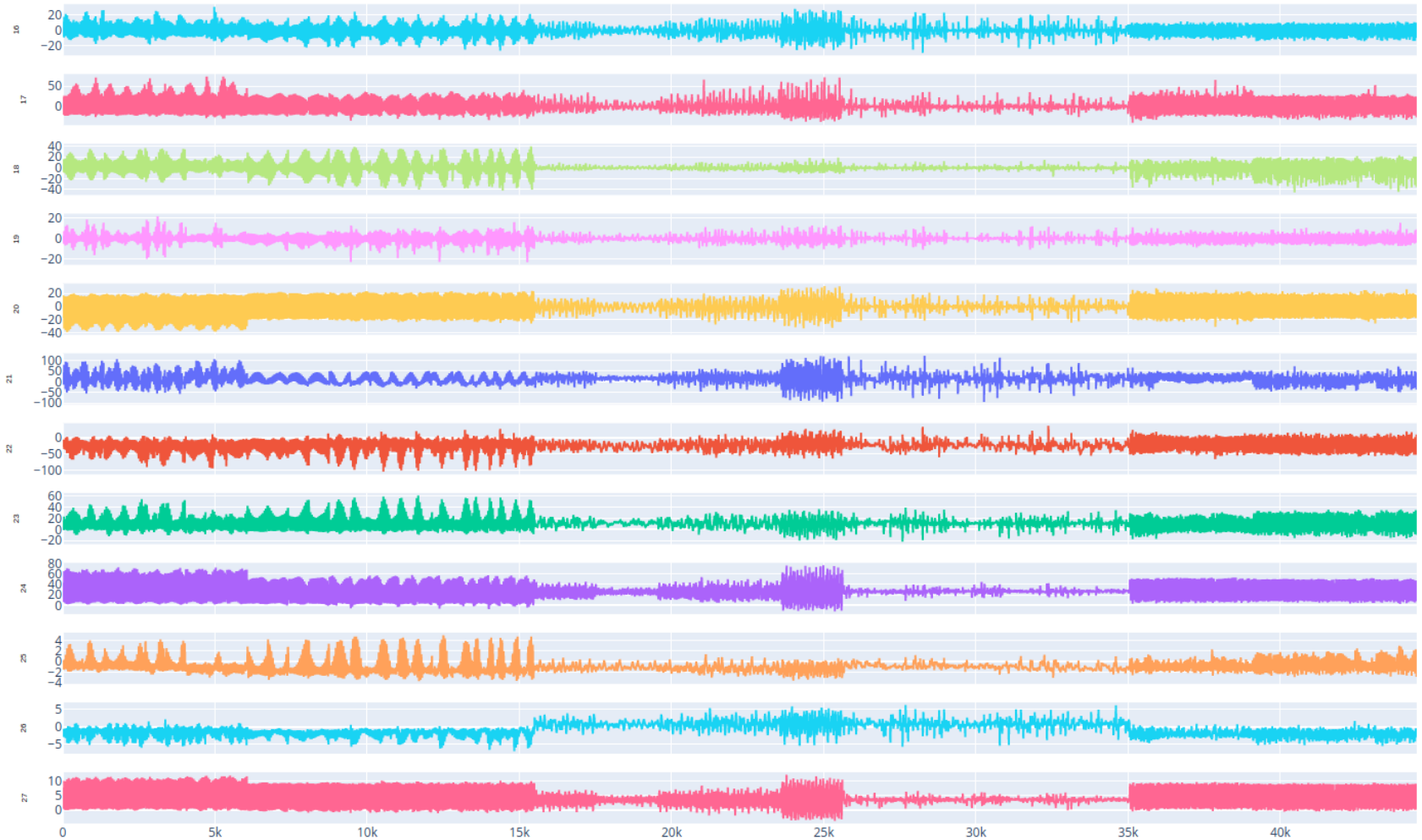


Figure: Sarcos Dataset: 21st dependent variable

Sarcos dataset: Non-stationary nature



Observations

- Further training of decoders can improve the results, i.e. training has not converged yet.
- MSE is doing better than CL-followed-by-decoder training.
- Though CL is able to capture the patterns, it is not able to capture the amplitudes well. Possibly just capturing the pattern is sufficient to get low CL loss.
- Due to the above point, CL also seems to struggle in non-stationary scenarios.
- The predicted and target embeddings of MSE loss method may not be comparable, hence poor hitrates. Range of values taken by predicted embeddings is -2 to 2, while that of target embeddings is -0.33 to 0.33.

Next Steps

- Train more for MSE method.
- Directly predict the embedding, rather than the change in embedding. Do $\mathbf{h}_x^n = GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n)$ rather than $\mathbf{h}_x^n = \mathbf{h}_x^p + GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n)$.
- Currently, batch size used is 16. Increasing batch size can help in contrastive learning as more negative samples enable better learning (as shown in SimCLR [14]).
- If and when CL performs at par with RL, try:
  - longer range forecasting
  - same experiments by introducing noise in sensor readings
  - Another dataset - SWaT would be a good test bed. Can we come up with a good controllable toy experiment setup like in the C-SWM paper to test our hypothesis - e.g. using ODEs?
  - Ablations:
    - remove GNN message passing feature
    - remove factored embeddings feature

Farther into the future

- Identifiability Theory [48]
- Abstract MDPs [73]

Abstract MDPs [73]

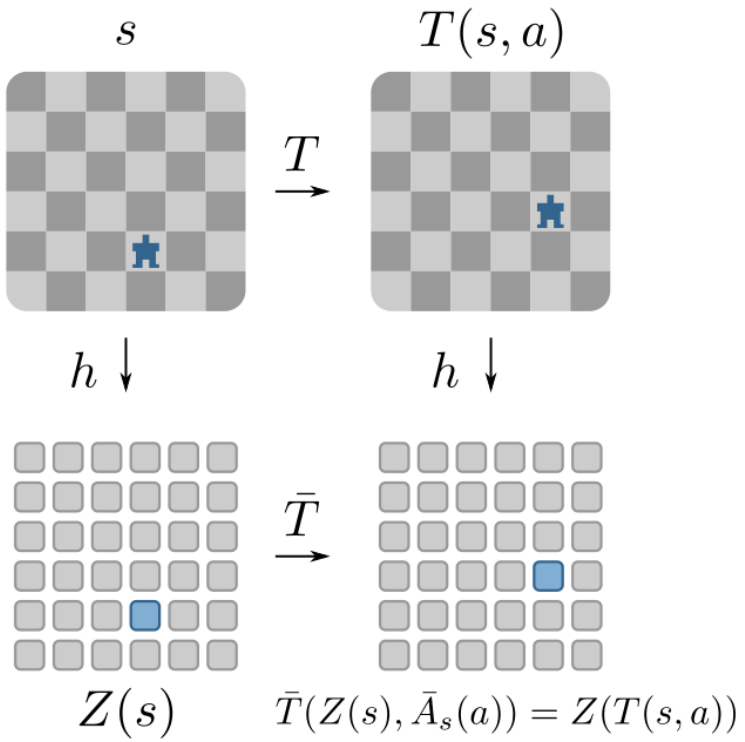


Figure 1: Visualization of the notion of equivariance under actions. We say Z is an action equivariant mapping if $Z(T(s,a)) = \bar{T}(Z(s), \bar{A}_s(a))$.

Updates 23-Apr-2020

- Experiments and Observations on Sarcos:
  - Dimension-wise forecasting error analysis
  - Analysis of embeddings from CL method
  - Sensitivity of results to hinge and sigma in CL
  - Per-dimension encoding for control variables, per-node CL
  - Varying the number of negative samples in CL
  - GNN vs MLP as Transition Model
  - Delta Transition Model vs Full Transition Model
  - Autoencoder + Transition Model
- Preliminary Results on SWaT:
  - CL vs MSE

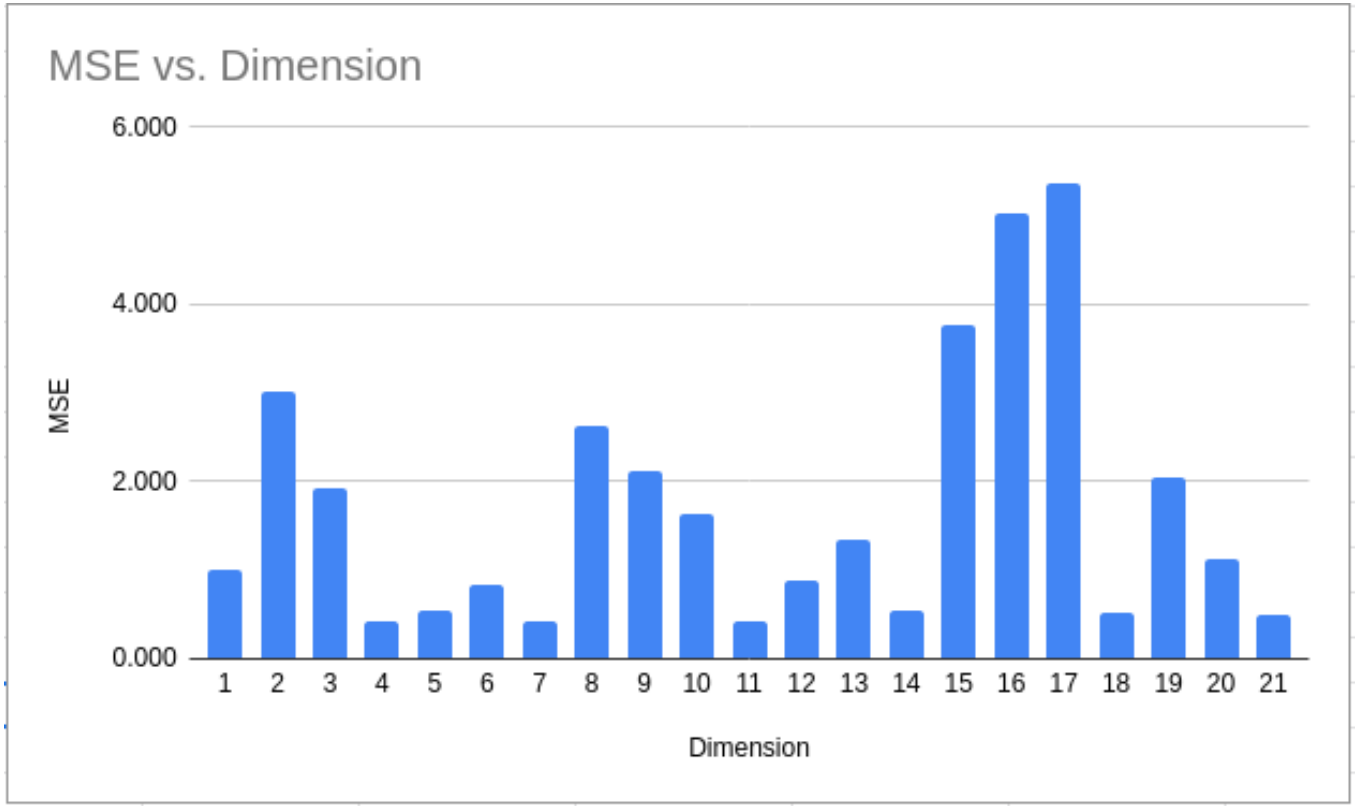
Datasets

Dataset	m	n	ν	L	N	*Full*	Modes
TEP	44	12	1000/h	0.12M	346	T	7 ($\times$4)
GHL	5*	5*		1.5M	1	F	
SWaT	24	27	1/s	0.5M	1	T	
Helicopter	25*	4	50/s*	0.1M	20	F	20
Quadrotor	15	8	100/s	25k	54	T	> 5
Neuronal Dyn.							
Sarcos	21	7	100/s	45k	1		
steamgen	4	4		9.6k	1		
cstr	2	1		7.5k	1		
evaporator	3	3		6.3k	1		
winding	2	5		2.5k	1		
flex. struct.	28	2		8.5k	1		
CD Player Arm	2	2		2.0k	1		
Foetal ECG	8	-		2.5k	1		
Robot Arm	1	1		1k	1		

m: dependent variables, n: independent variables, ν: sampling rate, L: length of each time series, N: number of time series, *Full*: All control variables available. *approx. (not certain).

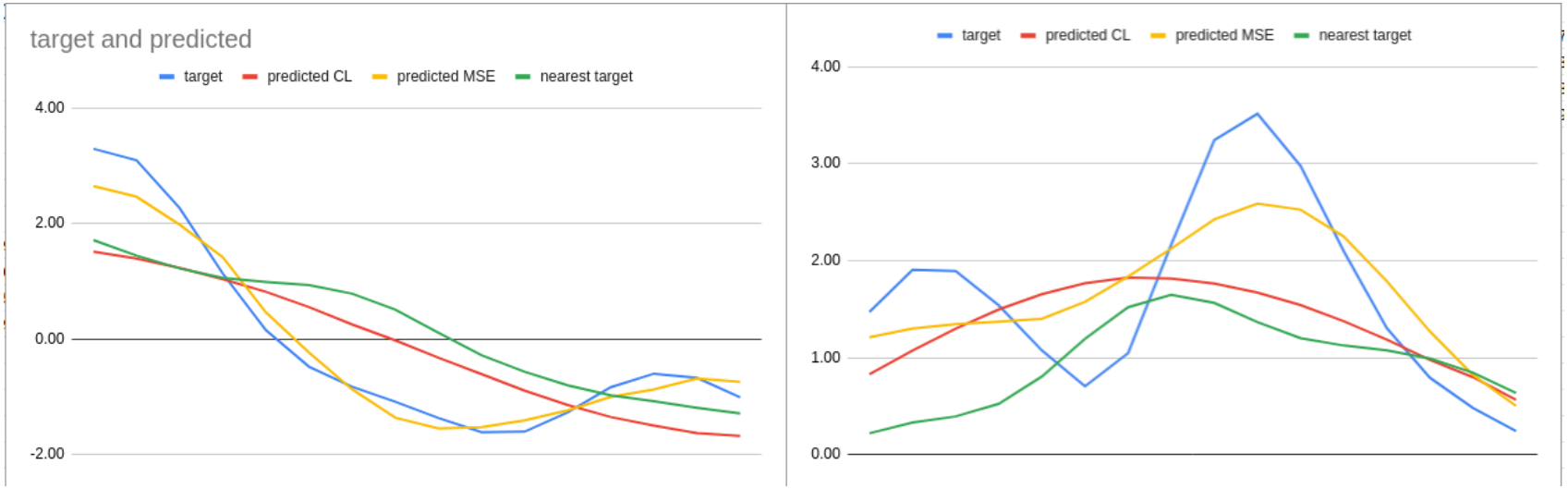
Sarcos Dataset

Dimension-wise forecasting errors from CL model



Observation: Some target dimensions are better modeled than others. Local optima (?): CL will be fine even if it captures one dimension well. Related to issues mentioned in [2, 70].

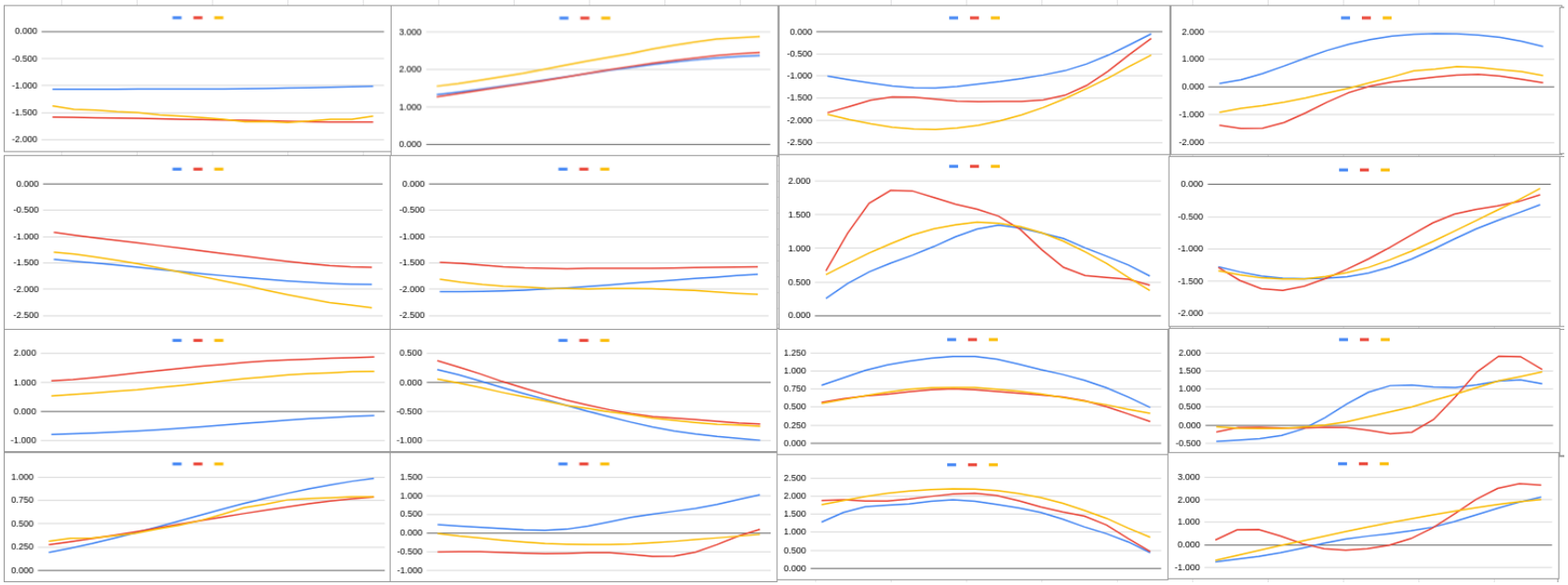
CL-vs-MSE results



nearest target: ground truth of the 2nd nearest embedding to the predicted embedding based on CL model.

Observation: The prediction is very off compared to ground truth but is similar to the 2nd nearest time series.

CL Results across dimensions (randomly chosen 16/21 dims.)

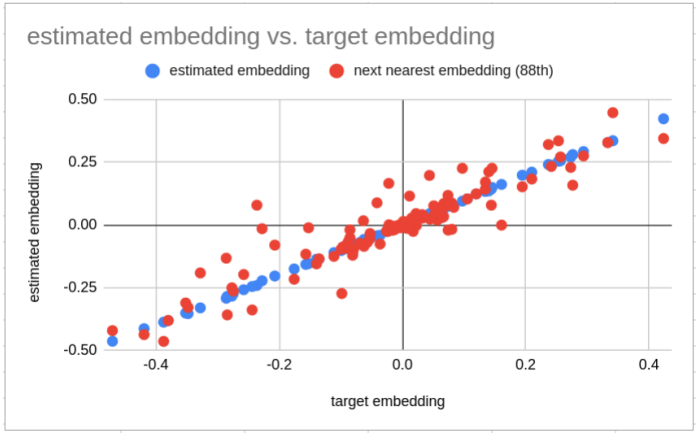


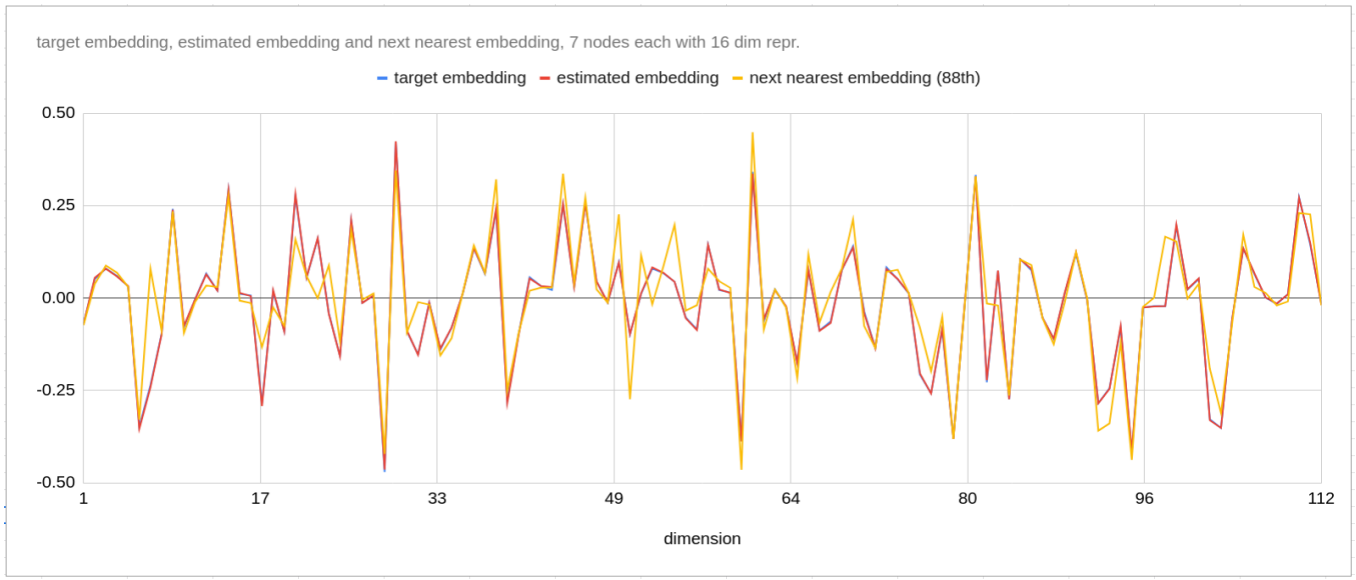
Legend: Ground Truth Prediction by CL Ground Truth of Nearest Neighbor

Observation: Some dependent variables/dimensions are perfectly modeled, others are way-off. In some cases, when red-yellow are not close, either red-blue or yellow-blue tend to be closer.

Embeddings from CL

Observation: Target and predicted embeddings are very close. Second nearest differs in some of the dimensions.





Effect of hinge (γ) and sigma (σ) on CL performance I

i. $\mathbf{f}_\theta$ to process $\mathbf{x}_{1\ldots t}$:

$$\mathbf{h}_x^p = \mathbf{f}_\theta(\mathbf{x}_{1\ldots t}) \tag{9}$$

ii. another 1D-CNN to process $\mathbf{c}_{1\ldots t+\tau}$:

$$\mathbf{h}_c^p = \mathbf{f}'_\theta(\mathbf{c}_{1\ldots t}), \quad \mathbf{h}_c^n = \mathbf{f}'_\theta(\mathbf{c}_{t+1\ldots t+\tau}) \tag{10}$$

p : *past*, n : *next*

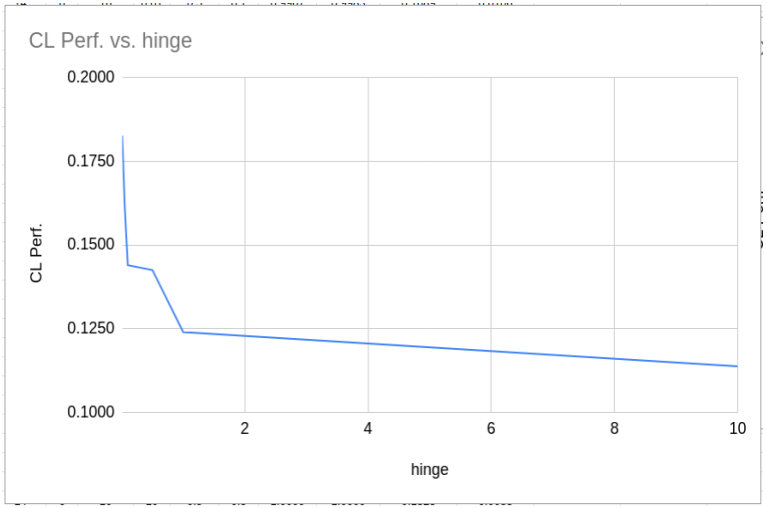
iii. a GNN to process $\mathbf{h}_x^p$ and $\mathbf{h}_c^p, \mathbf{h}_c^n$ to get final output

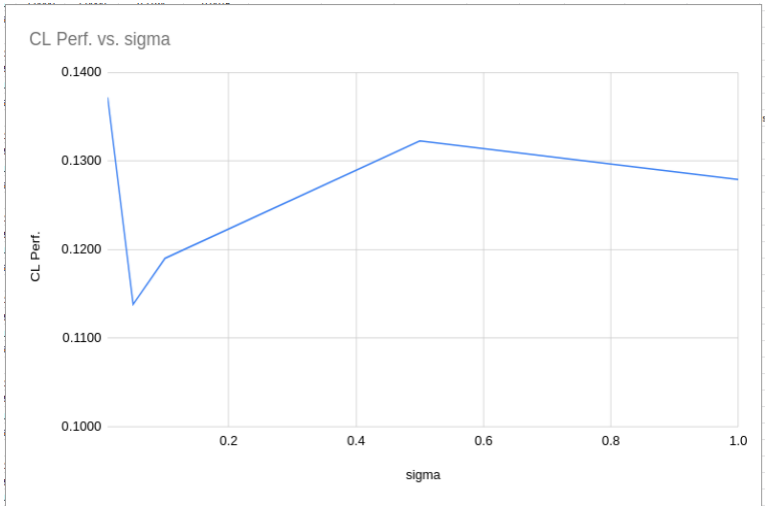
$$\mathbf{h}_x^n = \mathbf{g}_\theta(\mathbf{y}) = GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n). \tag{11}$$

$$\mathbf{h}_+ = \mathbf{f}_\theta(\mathbf{x}_{t+1\ldots t+\tau}), \mathcal{L} = d(\mathbf{h}_x^n, \mathbf{h}_+) + \sum_K max(0, \gamma - d(\mathbf{h}_x^n, \mathbf{h}_-)) \tag{12}$$

Here, $d(\mathbf{a}, \mathbf{b}) = \frac{1}{2\sigma^2}||\mathbf{a} - \mathbf{b}||_2^2$.

Effect of hinge (γ) and sigma (σ) on CL performance II





Observation: The performance is sensitive to the choice of hinge and sigma.

Separate (Control) Action Per Node vs Common Encoder

$$\mathcal{L} = \sum_{N} \left(d(\mathbf{h}_x^n, \mathbf{h}_+) + \sum_{K} max(0, \gamma - d(\mathbf{h}_x^n, \mathbf{h}_-)) \right) \tag{13}$$

N: number of nodes in graph. common control case: $\mathbf{h} \in \mathbb{R}^{Nd}$ and was obtained using all control variables as input to control encoder $\mathbf{f}'_{\theta}$.
In separate action per node case: $\mathbf{h} \in \mathbb{R}^{d}$, and one control variable is passed to $\mathbf{f}'_{\theta}$ at a time.
The idea is to force each node to learn a different aspect of the system (disentanglement).

Results

All models have 7 nodes unless specified.

All encoders trained for 50 epochs

t	Embed. Dims	Nodes	Shift	tau (horizon)
16	40	7	6	16

K	Decoder hidden dims	Control-wise	Control-wise (w/o GNN)	Common	end2end	Common (14 nodes)
5	13	0.1446	0.1568			
5	97	0.1374		0.1287		**0.1138**
10	13	0.1398	0.1579			
10	104	0.1257		0.1299*	**0.0710**	
10	156	**0.1165**	0.1216			
10	312	0.1185				

Observations:

- Results improve with increasing K for control-wise approach.
- MSE (end2end) is better than CL.
- Performance degrades on removing GNN in Control-wise approach
- Common encoder across control dimensions is better that control-wise encoder.

Control-wise Approach: GNN vs MLP as Transition Model

K	t	Embed. Dims	Nodes	Shift	tau (horizon)	hinge	control dropout	sigma	H@1	MRR	Valid MSE Loss	H@1	MRR	Valid MSE Loss
									per node loss, separate control, hd=13, 7 nodes					
									GNN			MLP		
5	16	40	7	6	16	10	0	0.05	1.0000	1.0000	0.1580	1.0000	1.0000	0.1872
5	16	40	7	6	16	10	0	0.01	1.0000	1.0000	0.1598	1.0000	1.0000	0.1801
5	16	40	7	6	16	10	0	0.1	1.0000	1.0000	0.1750	1.0000	1.0000	0.1804
5	16	40	7	6	16	20	0	0.05	1.0000	1.0000	0.1446	1.0000	1.0000	0.1663
5	16	40	7	6	16	20	0	0.01	1.0000	1.0000	0.1518	1.0000	1.0000	0.1759
5	16	40	7	6	16	20	0	0.1	1.0000	1.0000	0.1885	1.0000	1.0000	0.1848
5	16	40	7	6	16	1	0	0.05	1.0000	1.0000	0.1496	1.0000	1.0000	0.1996
5	16	40	7	6	16	1	0	0.01	0.9967	0.9983	0.2190	0.9967	0.9983	0.2847
5	16	40	7	6	16	1	0	0.1	1.0000	1.0000	0.1606	1.0000	1.0000	**0.1568**
5	16	40	7	6	16	1	0	0.2	1.0000	1.0000	0.1804	1.0000	1.0000	0.1722
5	16	40	7	6	16	1	0	0.4	1.0000	1.0000	0.1562	1.0000	1.0000	0.1889
5	16	40	7	6	16	0.5	0	0.2	1.0000	1.0000	0.1519	1.0000	1.0000	0.1685
5	16	40	7	6	16	0.5	0	0.4	1.0000	1.0000	0.1639	1.0000	1.0000	0.1839
10	16	40	7	6	16	10	0	0.05	1.0000	1.0000	0.1535	1.0000	1.0000	0.1756
10	16	40	7	6	16	10	0	0.01	1.0000	1.0000	0.1419	1.0000	1.0000	0.1740
10	16	40	7	6	16	10	0	0.1	1.0000	1.0000	0.1722	1.0000	1.0000	0.2037
10	16	40	7	6	16	20	0	0.05	1.0000	1.0000	0.1586	1.0000	1.0000	0.1772
10	16	40	7	6	16	20	0	0.01	1.0000	1.0000	0.1445	1.0000	1.0000	0.1865
10	16	40	7	6	16	20	0	0.1	1.0000	1.0000	0.1630	1.0000	1.0000	0.1724
10	16	40	7	6	16	1	0	0.05	1.0000	1.0000	**0.1398**	1.0000	1.0000	**0.1579**
10	16	40	7	6	16	1	0	0.01	0.9967	0.9983	0.1892	0.9967	0.9983	0.2692
10	16	40	7	6	16	1	0	0.1	1.0000	1.0000	**0.1382**	1.0000	1.0000	0.1671
10	16	40	7	6	16	0.5	0	0.2	1.0000	1.0000	0.1472	1.0000	1.0000	0.1876
10	16	40	7	6	16	0.5	0	0.4	1.0000	1.0000	0.1806	1.0000	1.0000	0.1764
10	16	40	7	6	16	1	0	0.2	1.0000	1.0000	0.1608	1.0000	1.0000	0.1944
10	16	40	7	6	16	1	0	0.4	1.0000	1.0000	0.1689	1.0000	1.0000	0.1863

Observations:

- GNN is better than MLP for Sarcos dataset
- K=10 is better than K=5 for GNN approach, mixed for MLP.

Reason for poor hitrates and MRR for MSE

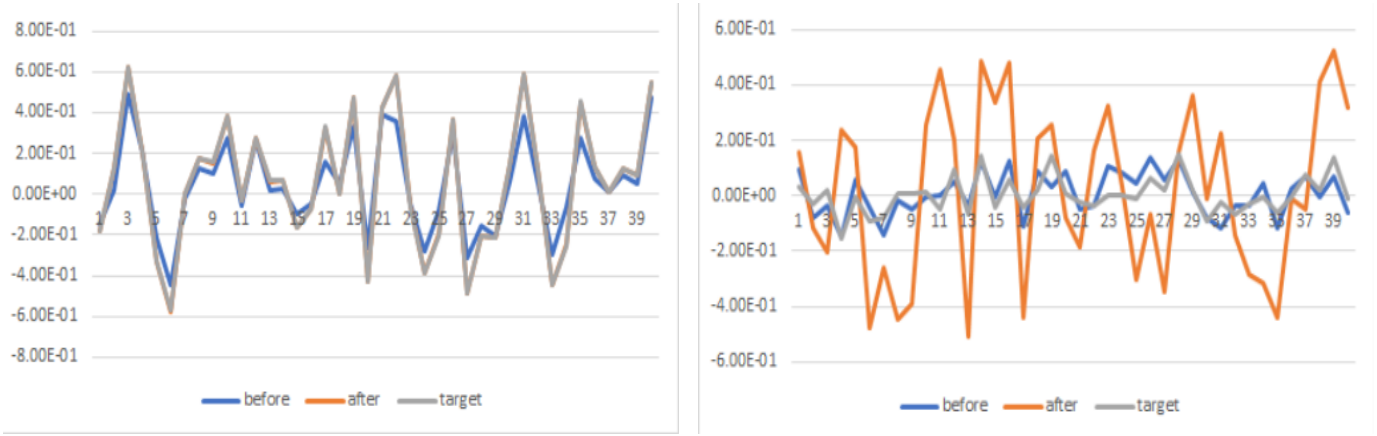


Figure: **Left: CL, Right: MSE**. Here, *before*: embedding of input dependent variables time series, *after*: embedding estimated by the transition model, *target*: embedding of target time series.

$H@1$ and MRR for the best CL models is 1.0, while for MSE models the best $H@1 = 0.0795$ and $MRR = 0.1933$.
Observation: The embeddings from encoder vs those of the target estimated via transition model lie in different regions of the space in case of MSE.

Discussion I

Why is MSE in case of decoder trained over freezed encoder(trained using CL) more than the end-to-end model trained using MSE?

- Input to the decoder is the afterGNN embedding.
- The range of beforeGNN/target is smaller than that of afterGNN as seen in endtoend.
- Given that end-to-end model performs pretty well, we observe that the range required for good reconstruction is captured in afterGNN embedding. there is no incentive to force beforeGNN and target embedding to be in the similar range. However, in case of contrastive loss(CL) based model, afterGNN is forced to be close to target embeddings. beforeGNN also comes in the similar range because target and beforeGNN come from same encoder.

Discussion II

- The distortions caused in the range of afterGNN due to CL, deteriorate its reconstruction performance.
- While the CL model allows for embeddings which can have action homomorphism, the end-to-end (MSE) model does not have that advantage.

Delta vs Full Embedding Transition Model I

Full:

$$\mathbf{h}_x^n = \mathbf{g}_\theta(\mathbf{y}) = GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n). \tag{14}$$

Delta:

$$\mathbf{h}_x^n = \mathbf{g}_\theta(\mathbf{y}) = \mathbf{h}_x^p + GNN_\theta(\mathbf{h}_x^p, \mathbf{h}_c^p, \mathbf{h}_c^n). \tag{15}$$

Results in all the previous slides are with the Delta Transition model.

The range of embeddings from the delta model may not be sufficient to reach the target embeddings when added to the input embedding because of the use of `tanh` in final layer.

Delta vs Full Embedding Transition Model II

	H@1	MRR	MSE
Full	0.811	0.896	**0.129**
Delta	**1.000**	**1.000**	0.145

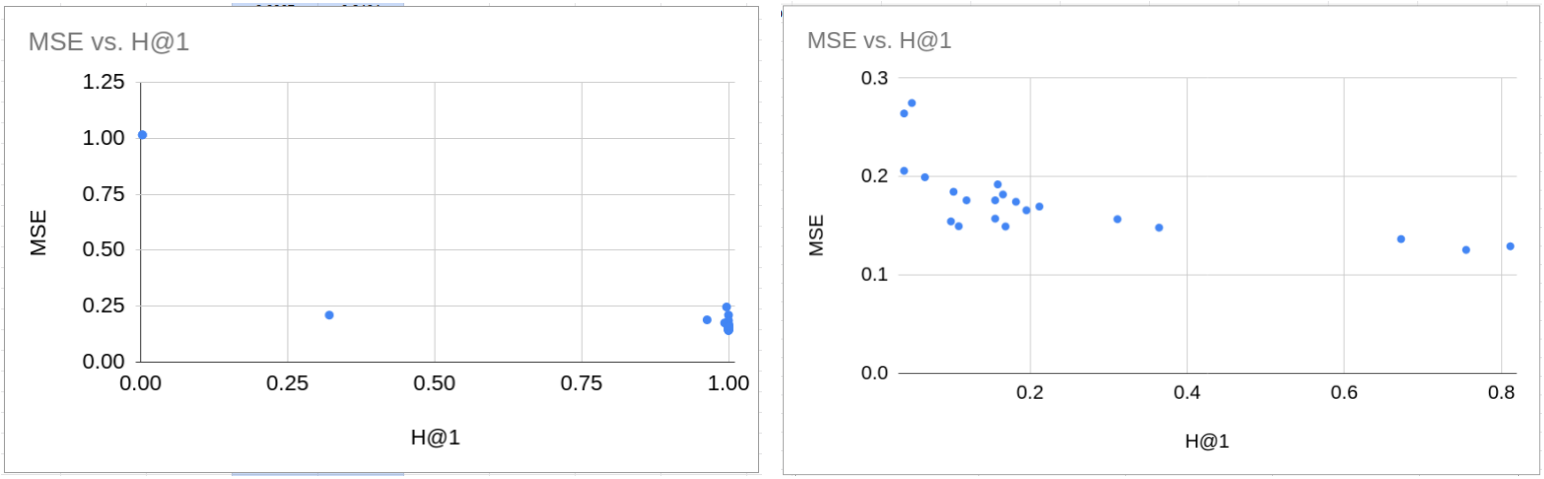
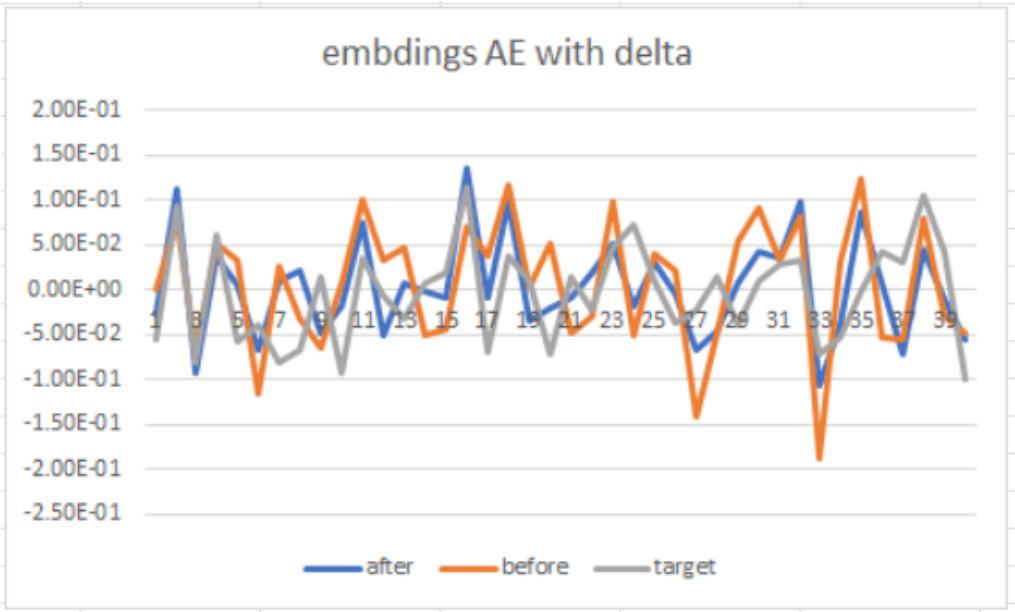


Figure: Left: Delta, Right: Full

H@1 being a relative metric where even if we have big distance from negative samples while having bigger (absolute terms) value of positive distance can give H@1 as 1.

Autoencoder + Transition Model

- Train two autoencoders: one for dependent variables, one for independent variables.
- Get embeddings for the variables from respective autoencoders
- Train a GNN-based transition model using these embeddings



Autoencoder + Transition Model Results

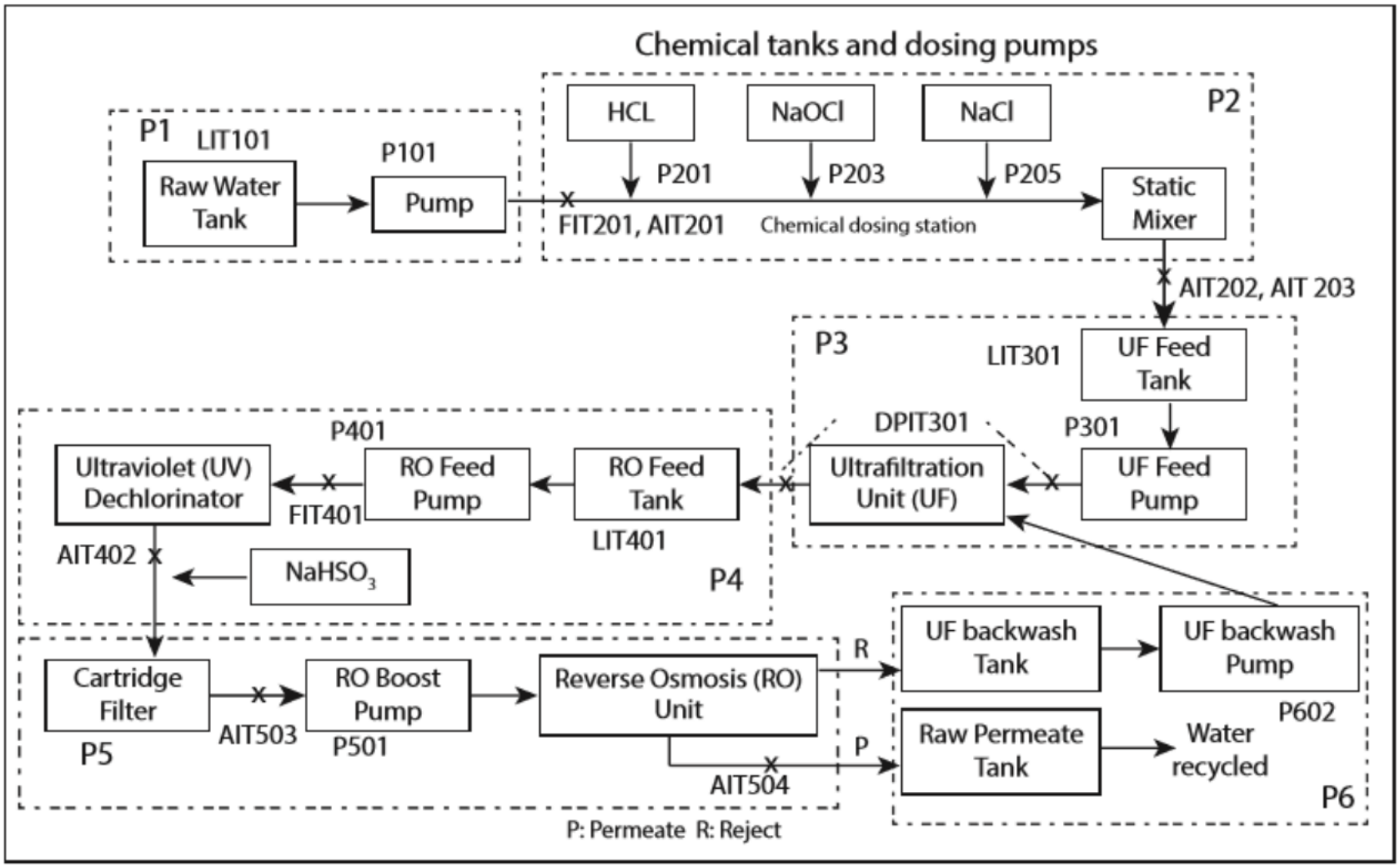
	with delta model				
			Reconstruction		Prediction
	H@1	MRR	MSE Dep.	MSE Control	MSE Dep.
AE+Transition	0.874	0.917	0.0172	0.015	0.088
CL	1	1			0.145/0.117
MSE	0.079	0.193			**0.071**

Observations:

- AE+Transition is better than CL. This again suggests CL model getting stuck in local optima and not able to capture the information for all the dependent variables.
- AE+Transition is close to MSE model in terms of prediction error (mse) but significantly better in terms of *H*@1 and *MRR*.

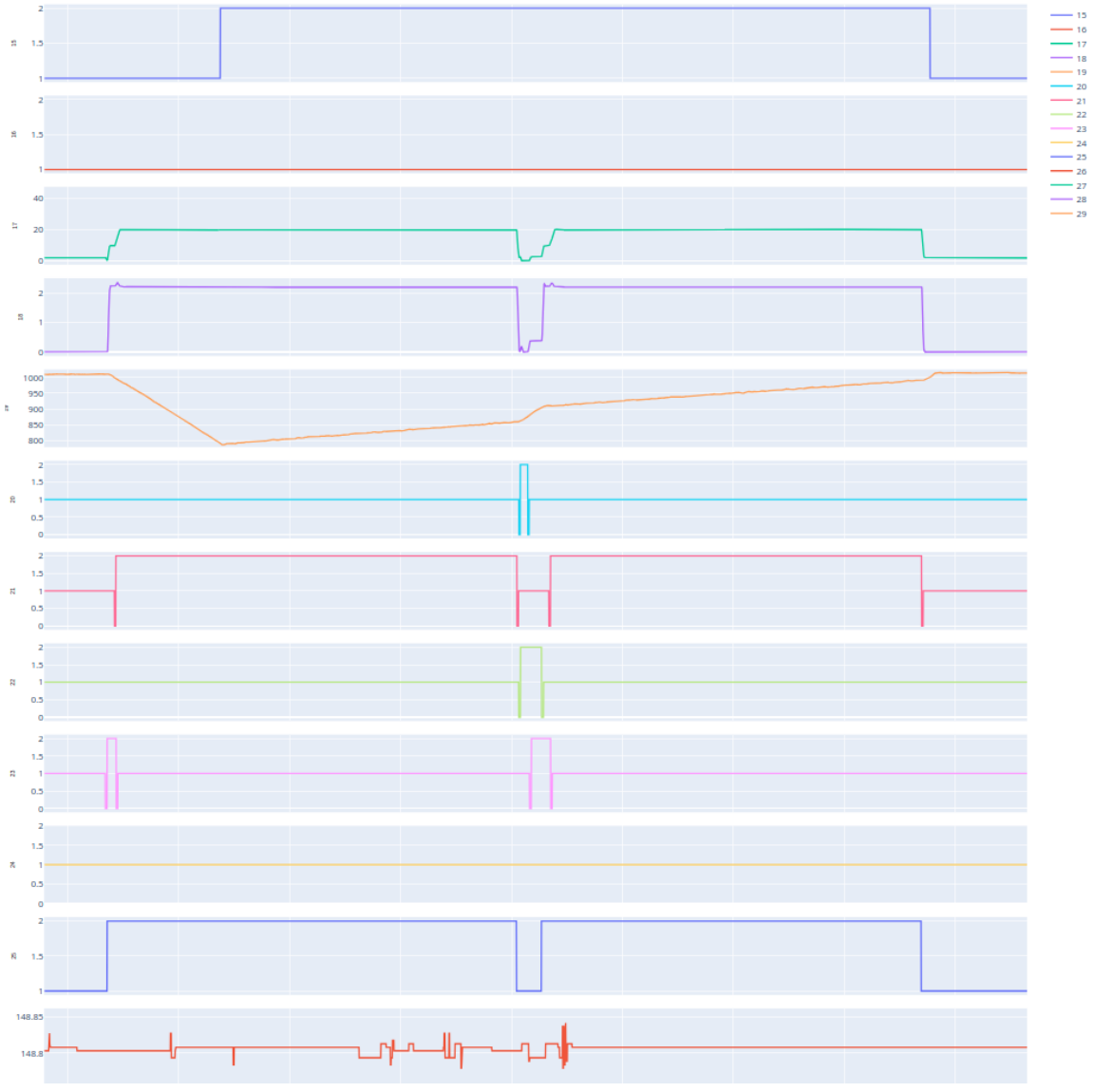
SWaT Dataset

SWaT: Secure Water Treatment Plant



Sample Time Series - SWaT 2k points

Sample time series - SWaT-2 4k points



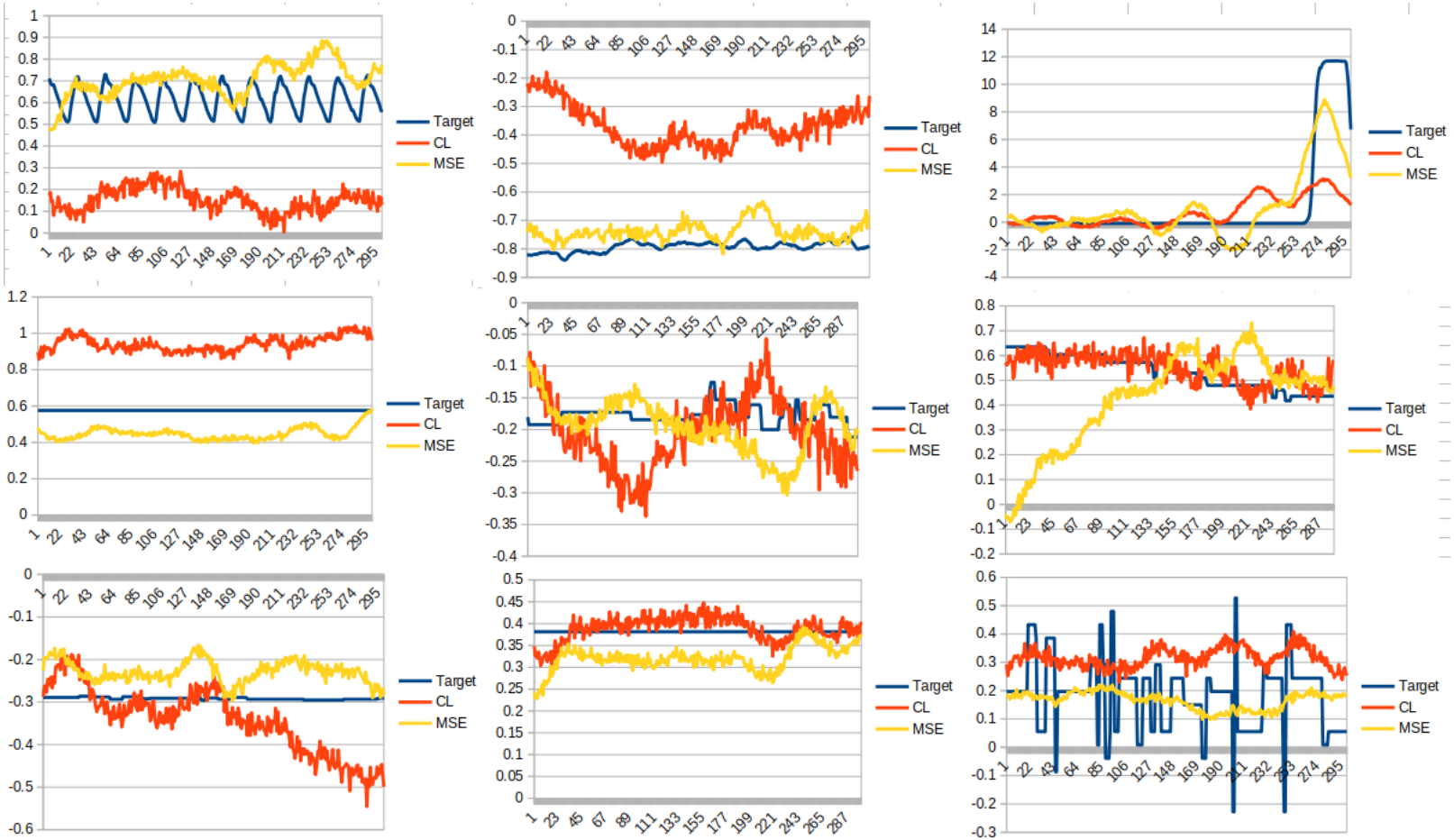
Summary of Results

							MSE with GNN			CL with GNN			CL with MLP		
Epochs	Stride	K	t	Embed. Dims	Shift	tau (horizon)	H@1	MRR	Valid MSE Loss	H@1	MRR	Valid MSE Loss	H@1	MRR	Valid MSE Loss
100	1	10	600	40	120	120	0.252	0.464	**0.091**	1	1	0.362			
	1	10	600	80	120	120	0.469	0.664	**0.161**	1	1	0.308			
	1	10	600	160	120	120	0.082	0.221	**0.076**	1	1	0.314			
50	2	10	1200	40	120	300	0.225	0.350	**0.085**	1	1	0.263			
	2	10	1200	40	240	300	0.128	0.268	**0.193**	1	1	0.311			
	2	10	600	40	120	300	0.038	0.166	**0.090**	1	1	0.289	1	1	0.255
	2	10	600	80	120	300	0.010	0.059	**0.099**	1	1	0.206	1	1	0.223
	2	10	600	40	240	300	0.135	0.255	**0.128**	1	1	0.308			
	2	10	300	40	30	300	0.033	0.081	**0.063**	1	1	0.173			
	2	10	300	40	60	300	0.289	0.431	**0.095**	1	1	0.198			
		K=10	hinge= 1 or 10	sigma=0. 05 or 0.1	Nodes=7	Dropout =0									

Observations:

- Increasing the embedding dimensions helps.
- CL with MLP performs better than CL with GNN**.
- Results improve by reducing the shift for all cases as it implies more training examples.
- Stride=2 is better than stride=1

Sample predictions across dimensions



Next Steps

- modify loss such that the encoder is forced to retain the information about the dependent variables.
- look at "full" transition model more carefully.
- autoregressive forecasting to test robustness of embeddings
- introduce noise in sensors
- perturb embeddings to see effect on different dimensions.
- Look at other toy models/tasks: Lorenz, Lotka-Volterra, SIR, DREAM

Lotka-Volterra Equations

$$\frac{dx}{dt} = \alpha x - \beta xy,$$
$$\frac{dy}{dt} = \delta xy - \gamma y,$$

where

x is the number of prey (for example, rabbits);

y is the number of some predator (for example, foxes);

$\frac{dy}{dt}$ and $\frac{dx}{dt}$ represent the instantaneous growth rates of the two populations;

t represents time;

$\alpha, \beta, \gamma, \delta$ are positive real parameters describing the interaction of the two species.

Given that both x and y influence each other, can we still put this in our current framework, e.g. assuming y to be the control, x to be the dependent variable?

SWaT Toy

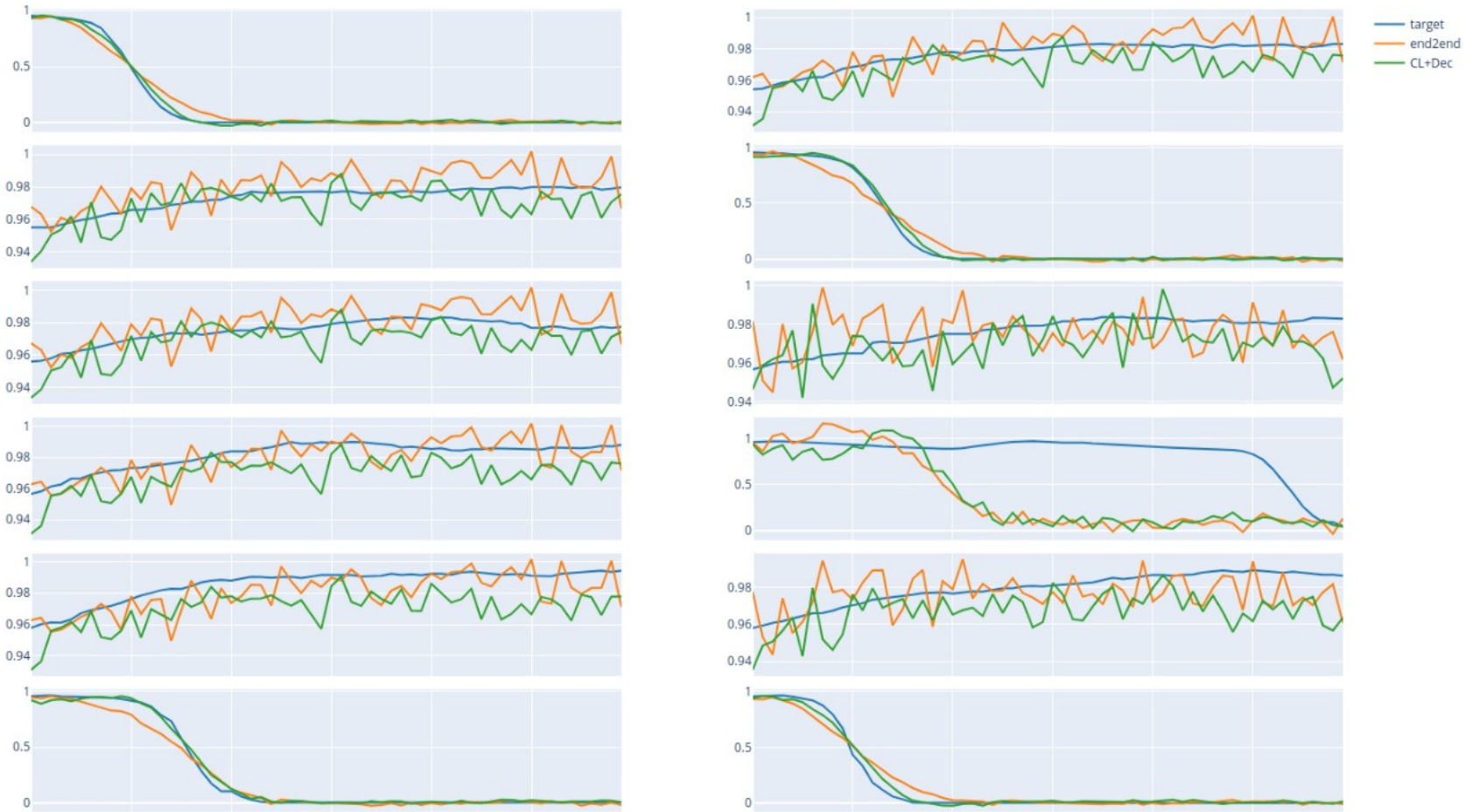
Figure: Depicting two dependent variables followed by two control variables

Summary Results (9 May 2020)

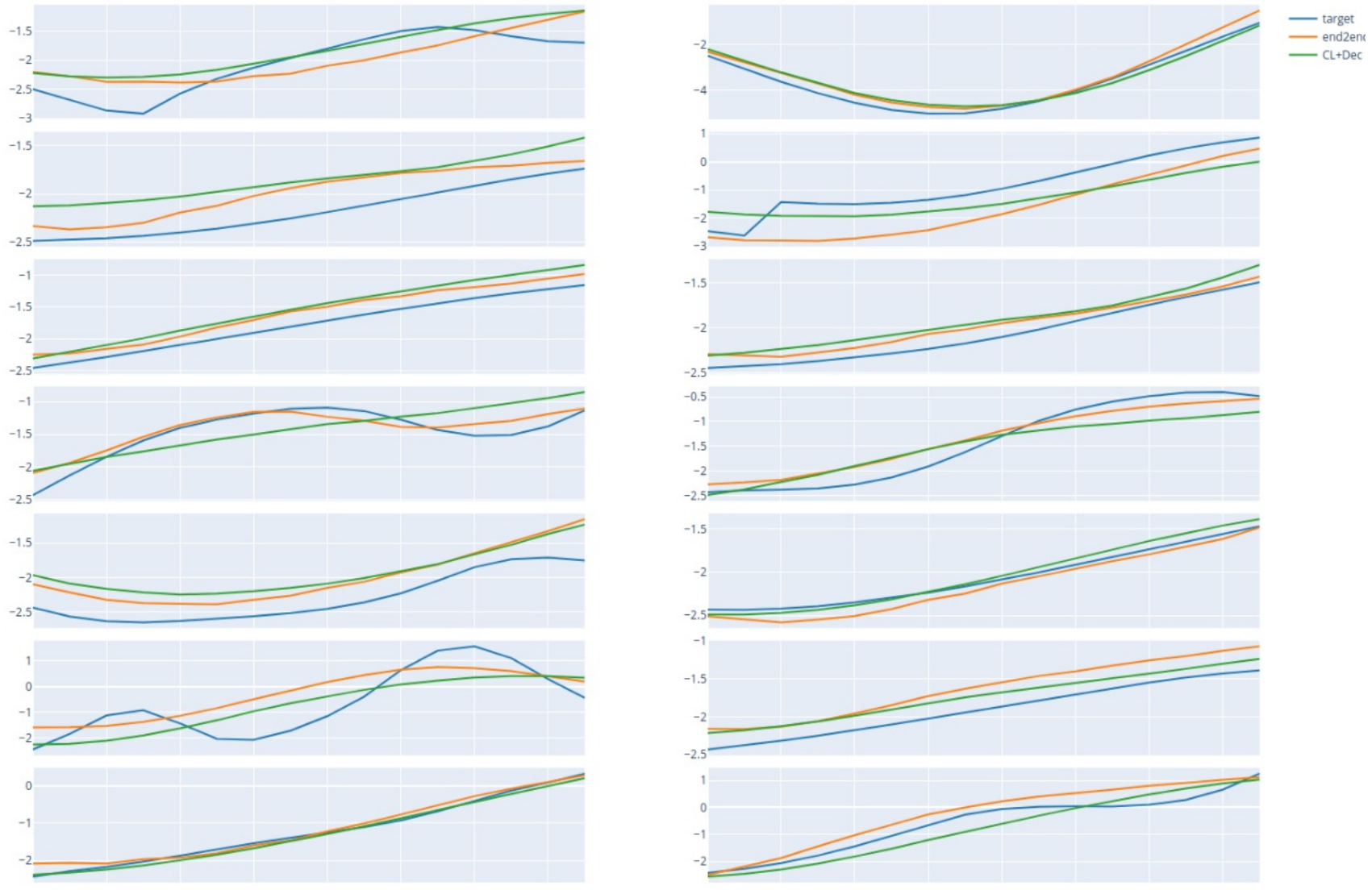
	Now						Earlier					
	SWaT_Toy (min-max norm)			Sarcos			SWaT_Full (z-norm)			Sarcos		
	H@1	MRR	MSE	H@1	MRR	MSE	H@1	MRR	MSE	H@1	MRR	MSE
end2end (upper bound)	0.0083	0.0477	0.0025	0.0000	0.0223	0.0580	0.0335	0.0807	0.0630	0.0232	0.0772	0.0628
CL+Dec	0.0909	0.2444	0.0036	0.9172	0.9482	0.0696	1.0000	1.0000	0.1730	0.7980	0.8780	0.1033
AE+Transition	On Run									0.8740	0.9170	0.0880
MSCL+Dec	Debugging											

- **end2end (upper bound)**: encoder, followed by transition model, followed by decoder. trained end2end for prediction of dependent variables using mse loss.
- **CL+Dec**: encoder, followed by transition model trained using contrastive learning. decoder is trained on the frozen encoder for prediction of dependent variables.
- **AE+Transition**: encoder-decoder pair trained using reconstruction loss (no prediction). frozen encoder is followed by transition model training using CL, followed by decoder training.
- **MSCL+Dec**: multi-scale global-local contrastive loss in CL+Dec.

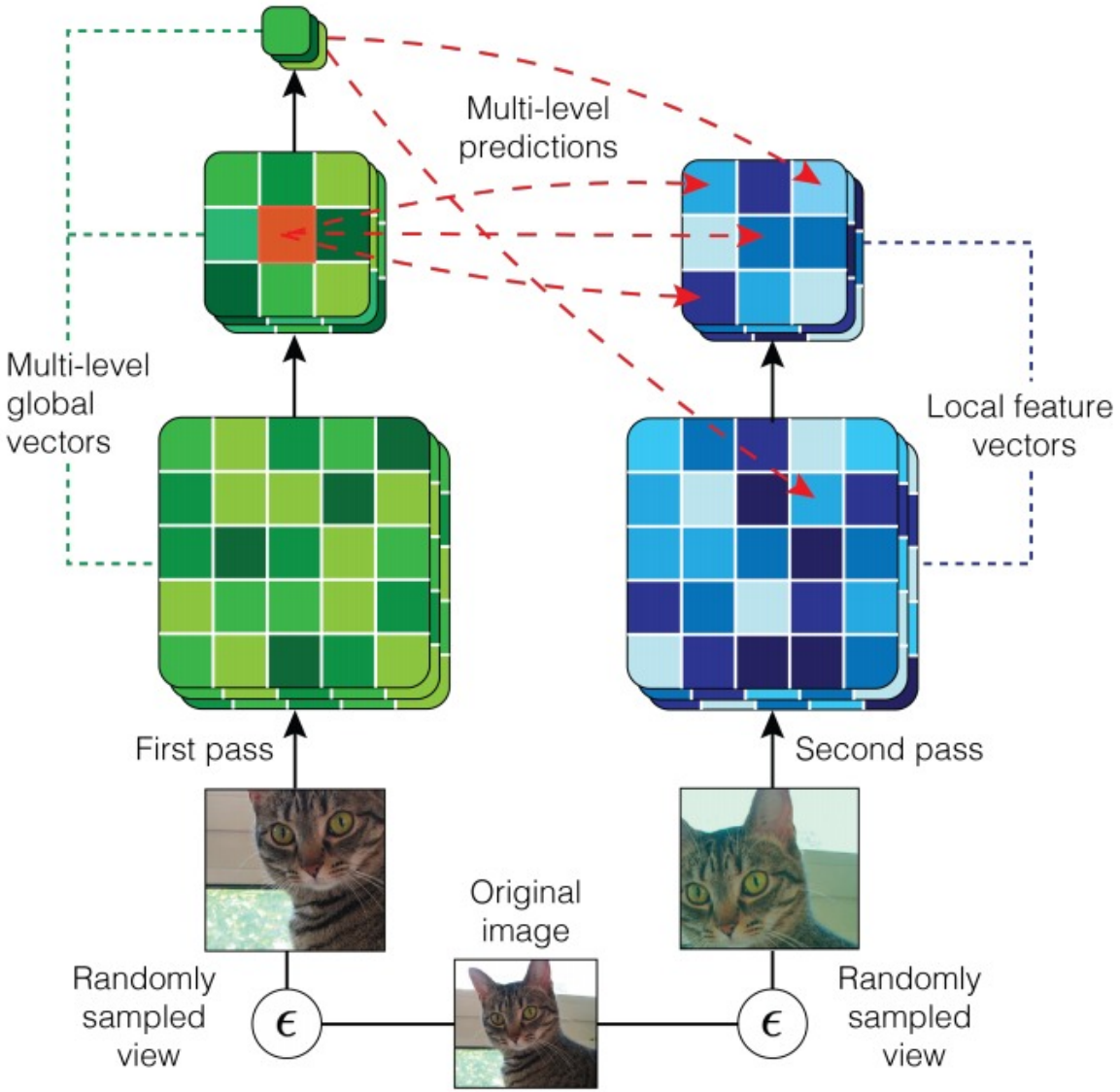
Sample predictions: SWaT



Sample predictions: Sarcos



Multi-scale global-local loss [6, 2]



Contrastive state representation learning

[2]

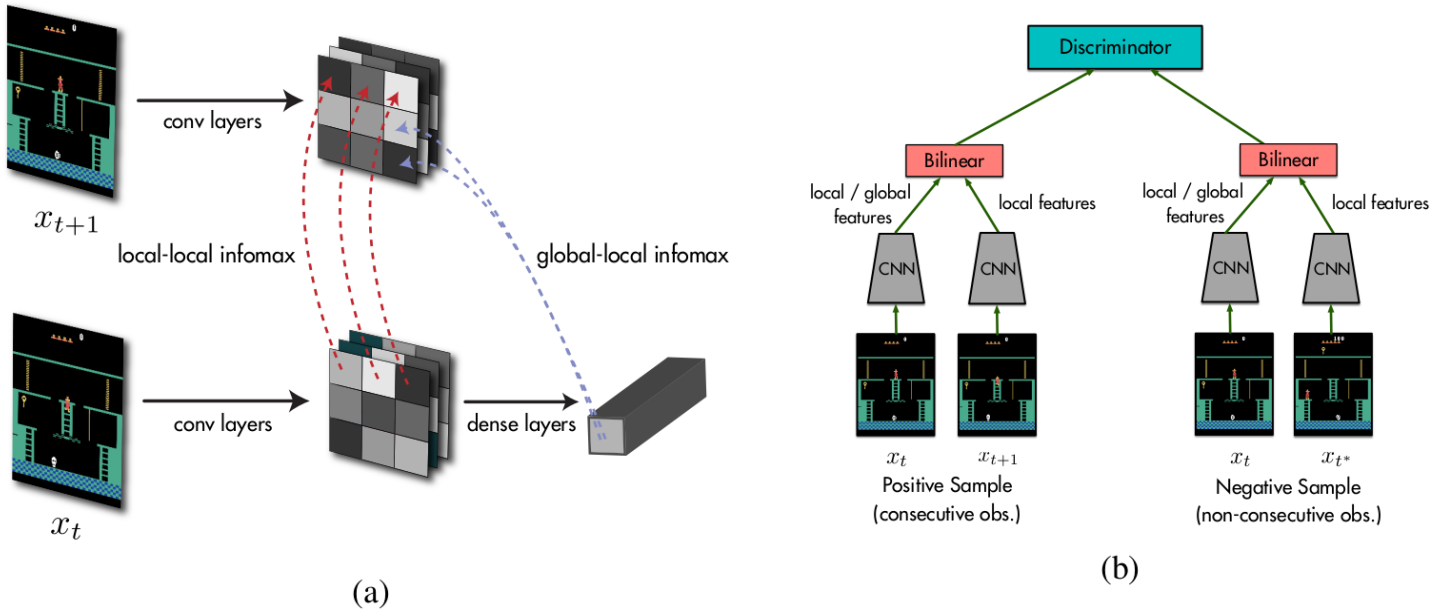
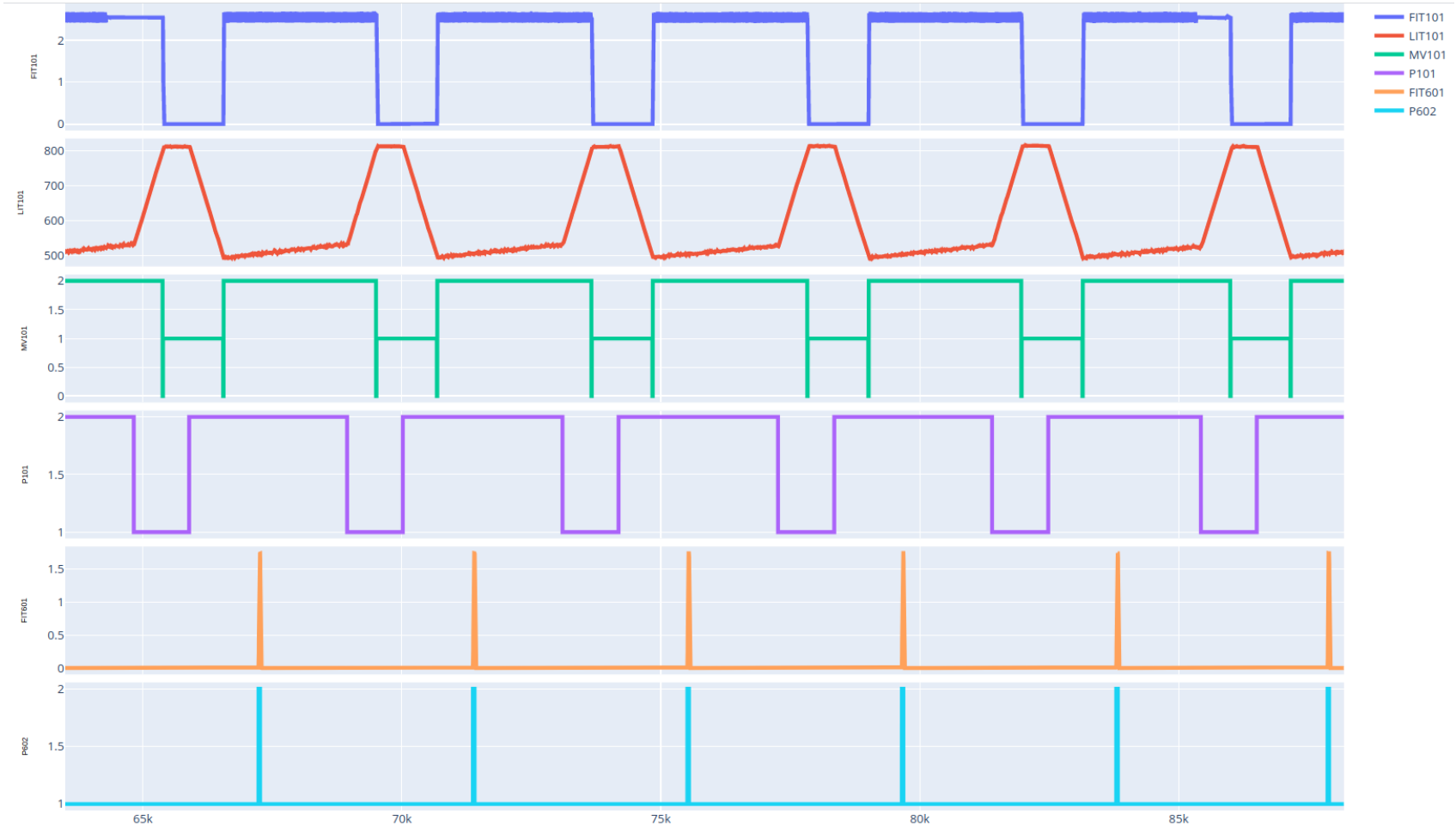


Figure 2: A schematic overview of SpatioTemporal DeepInfoMax (ST-DIM). (a) shows the two different mutual information objectives: local infomax and global infomax. (b) shows a simplified version of the contrastive task we use to estimate mutual information. In practice, we use multiple negative samples.

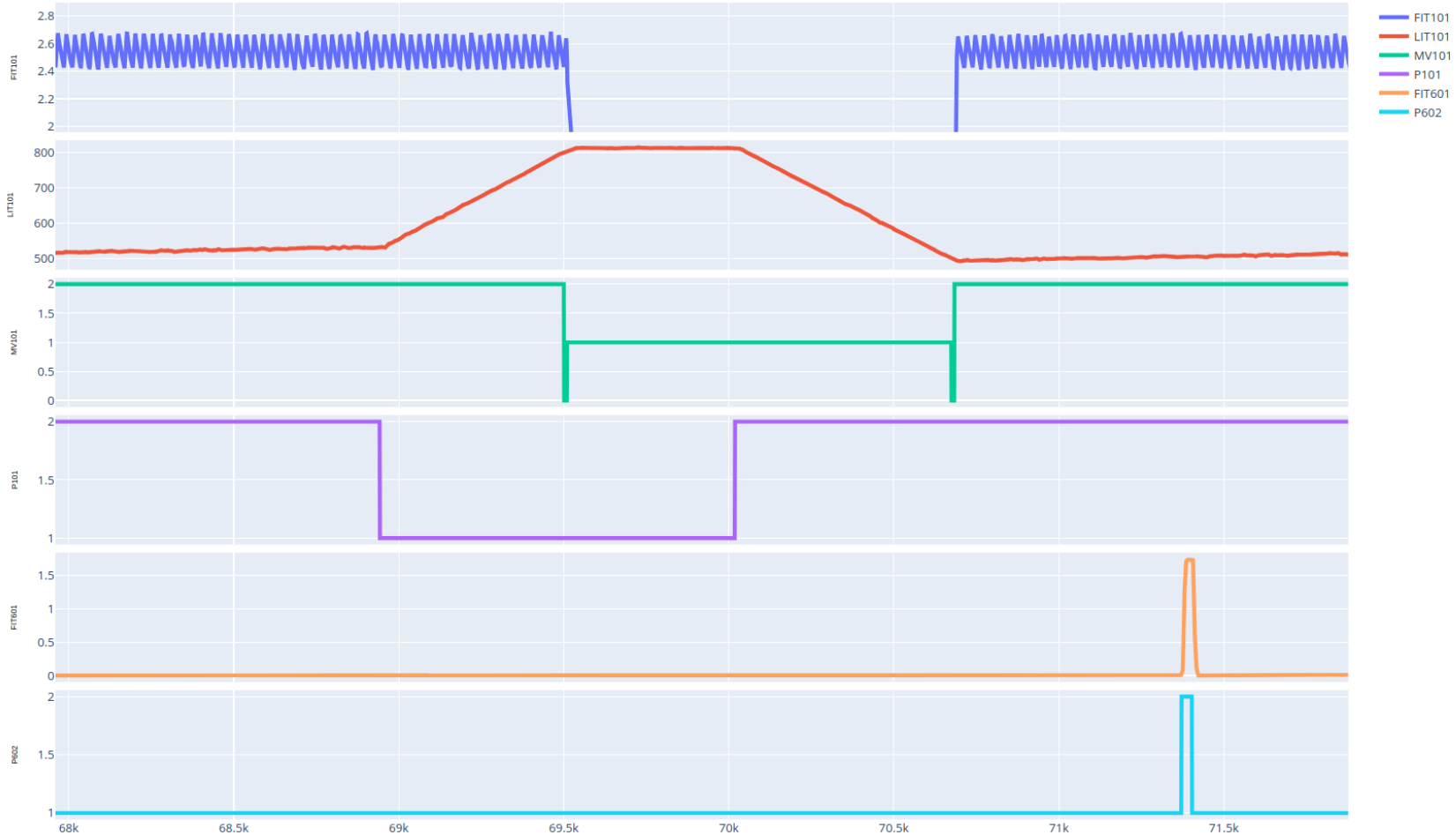
ST-DIM: maximize mutual information between data and learned representations

Updates 31-May-2020

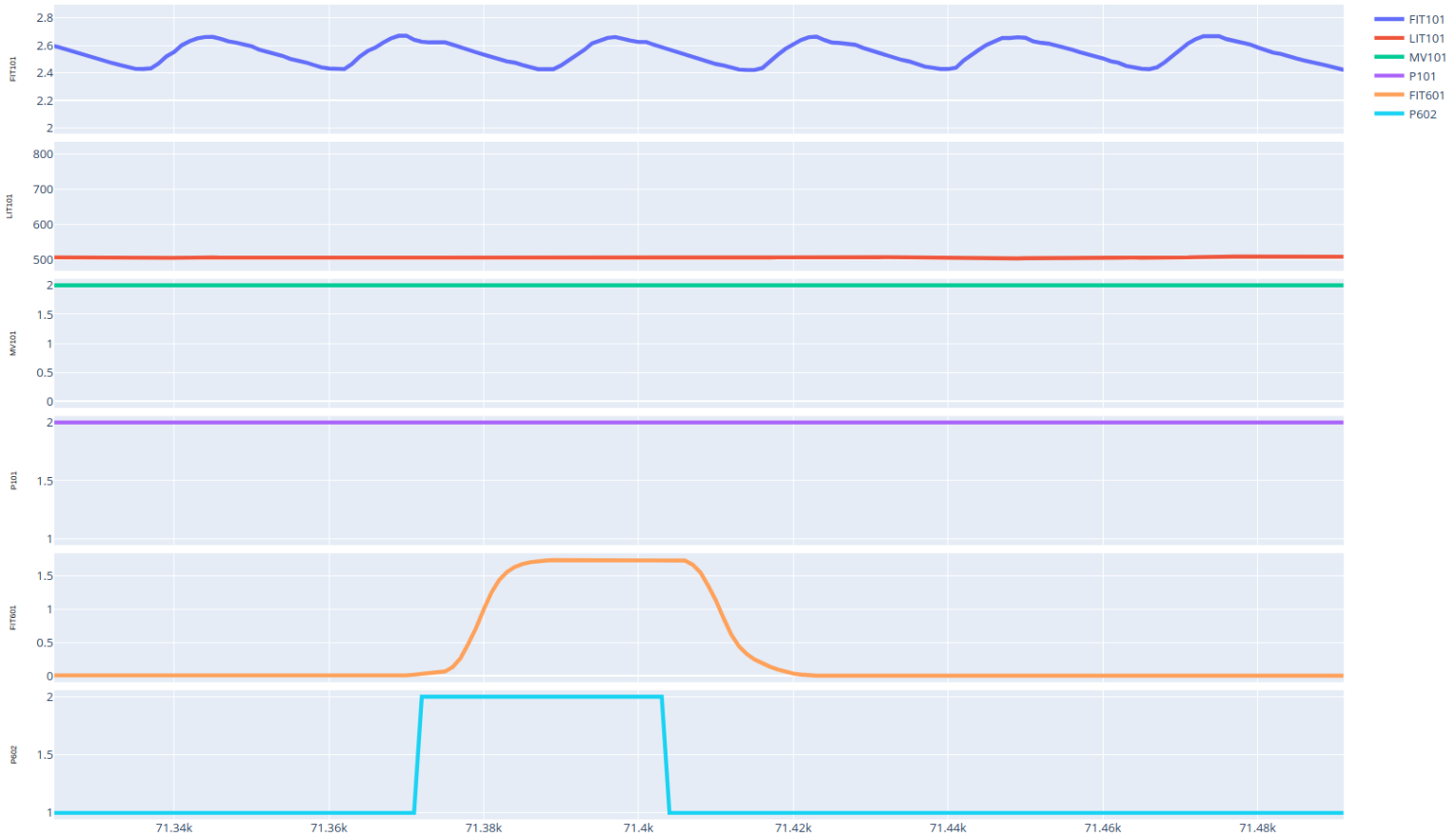
SWaT Sample Data (3 Control Variables (CVs), 3 Dependent Variables (DVs))



SWaT Sample Data (zoomed-in view (y-axis) showing sinusoidal pattern in DV0)



SWaT Sample Data (zoomed-in view (x-axis) of the short duration CV2 and DV2 change



Three dataset variants used in experiments: **SWaT1**: 2 CVs, 2 DVs from stage-1; **SWaT2**: 3 CVs, 3 DVs from stage-1 and 6; **SWaT3**: 1 CV and DV from stage-1, and 1 CV and DV from stage-6 with removed sine pattern and increased frequency of controls in second CV.

SWaT1 (2 CVs, 2 DVs) Summary Results

Training 50 epochs

						SWaT (2 CVs, 2 DVs)							
						Validation				Test			
Description	Method	SC	PC	Full vs Delta	MLP or GNN or Concat	H@1	H@10	MRR	MSE	H@1	H@10	MRR	MSE
Using only past dependents	**M0.1**	N							0.0026				0.0028
Using only past controls	**M0.2**	N							0.0023				0.0023
		Y							0.0051				0.0054
Using only future controls	**M0.3**	N							0.0053				0.0050
		Y							0.0273				0.0248
Using past dependents and past controls with common encoder for both	**M0.4.1**	N							0.1562				0.1654
		Y							0.0486				0.0491
Using past dependents (D) and past controls (C) with separate encoder for D and C	**M0.4.2**	N							0.0017				0.0018
		Y							0.0023				0.0026
MSE with common control	**M1**	N	-	F	C	0.0020	0.0238	0.0144	0.0015	0.0021	0.0205	0.0148	0.0016
MSE with separate control	**M2**	Y	-	F	C	0.0020	0.0179	0.0121	0.0013	0.0021	0.0164	0.0124	0.0015
MSE with separate control	**M2 (new SC)**	Y	-	F	C								
MSE+CL with common control	**M3**	N	Y	F	M	0.0317	0.2440	0.1073	**0.0012**	0.0349	0.2772	0.1167	**0.0012**
		N	N	F	M	0.0159	0.1488	0.0685	0.0013	0.0123	0.1581	0.0617	0.0014
MSE+CL with separate control	**M4**	Y	Y	F	M	0.0139	0.2143	0.0827	0.0014	0.0164	0.2320	0.0864	0.0015
CL with common control	**M6**	N	Y	F	M	**0.0952**	**0.5595**	**0.2279**	0.0020	**0.0945**	**0.5832**	**0.2368**	0.0021
		N	N	F	M	0.0655	0.4107	0.1831	0.0030	0.0965	0.4887	0.2276	0.0029
CL with separate control	**M7**	Y	Y	F	M	0.0556	0.4266	0.1662	0.0040	0.0575	0.5216	0.1886	0.0039
		Y	N	F	M	0.0456	0.2996	0.1222	0.0032	0.0370	0.3039	0.1199	0.0034
	M7 (new SC - with DD)	Y	Y	F	M								

Observations:

- M3 (MSE+CL) is better than M1 (only MSE) in terms of MSE.
- M6 (only CL) is worse than M1 (only MSE) in terms of MSE.
- M4 (MSE+CL with each CV passed separately through encoder) is worse than M3 (all CVs passed simultaneously through encoder), but similar to M1 and M2.
- M7 (only CL with separate encoder per control variable) is worst.

SWaT2 (3 CVs, 3 DVs) Summary Results

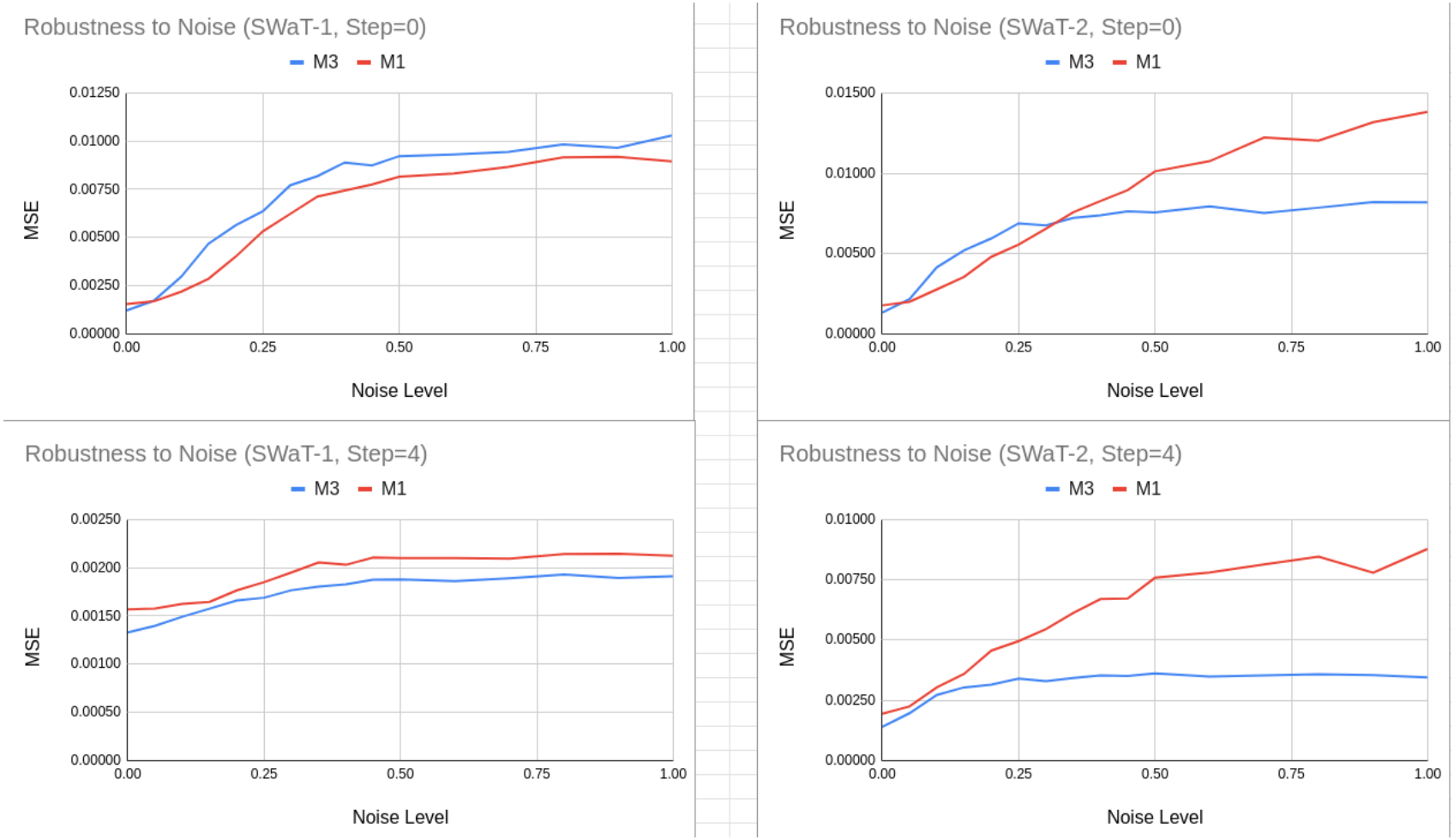
Training 50 epochs

						SWaT (3 CVs, 3 DVs)							
						Validation				Test			
Description	Method	SC	PC	Full vs Delta	MLP or GNN or Concat	H@1	H@10	MRR	MSE	H@1	H@10	MRR	MSE
Using only past dependents	M0.1	N							0.0032				0.0035
Using only past controls	M0.2	N							0.0034				0.0037
		Y							0.0104				0.0110
Using only future controls	M0.3	N							0.0038				0.0036
		Y							0.0917				0.0933
Using past dependents and past controls with common encoder for both	M0.4.1	N							0.2727				0.2741
		Y							0.2184				0.2343
Using past dependents (D) and past controls (C) with separate encoder for D and C	M0.4.2	N							0.0029				0.0032
		Y							0.0149				0.0148
MSE with common control	M1	N	-	F	C	0.0020	0.0179	0.0127	0.0015	0.0021	0.0205	0.0130	0.0018
MSE with separate control	M2	Y	-	F	C	0.0000	0.0198	0.0122	0.0081	0.0021	0.0287	0.0144	0.0087
MSE with separate control	M2 (new SC)	Y	-	F	C	0.0020	0.0198	0.0113	0.0015	0.0020	0.0202	0.0117	0.0019
MSE+CL with common control	M3	N	Y	F	M	0.0734	0.5139	0.2046	**0.0012**	0.0698	0.5175	0.2072	**0.0013**
		N	N	F	M	0.0238	0.2004	0.0789	0.0015	0.0103	0.1828	0.0701	0.0018
MSE+CL with separate control	M4	Y	Y	F	M	0.0139	0.1607	0.0696	0.0062	0.0144	0.1663	0.0707	0.0063
CL with common control	M6	N	Y	F	M	0.1171	**0.6409**	0.2687	0.0040	**0.1335**	**0.6509**	**0.2874**	0.0046
		N	N	F	M	0.0655	0.4167	0.1758	0.0041	0.0760	0.4723	0.1970	0.0044
CL with separate control	M7	Y	Y	F	M	0.0298	0.2123	0.0964	0.0043	0.0308	0.2361	0.1047	0.0050
		Y	N	F	M	0.0337	0.2718	0.1132	0.0051	0.0287	0.2957	0.1120	0.0055
	M7 (new SC - with DD)	Y	Y	F	M	0.0734	0.5337	0.2093	0.0035	0.0801	0.5647	0.2228	0.0037

Observations:

- M2 (with each CV passed separately through encoder) does not work.
- M3 (MSE+CL) is better than M1 (only MSE) in terms of MSE.
- M6 (only CL) is worse than M1 (only MSE) in terms of MSE.
- M4 (MSE+CL with each CV passed separately through encoder) is worse than M3 (all CVs passed simultaneously through encoder).
- M6 and M7 (only CL) much worse than M3 and M4 (MSE+CL).

Robustness to Noise Injected in Past DVs



Observations:

- For SWaT1 (2CVs, 2DVs), M3 worse than M1 on step=0 but better on step=4.
- For SWaT2 (3CVs, 3DVs), M3 degrades more gracefully with increasing noise levels in comparison to M1 for step=0 as well as step=4.

Multi-step (multi-window) prediction

Observation:

- Minor increase in MSE for both M3 and M1 in multi-step case without noise.

Sinusoidal pattern in DV0 (50 epochs training)

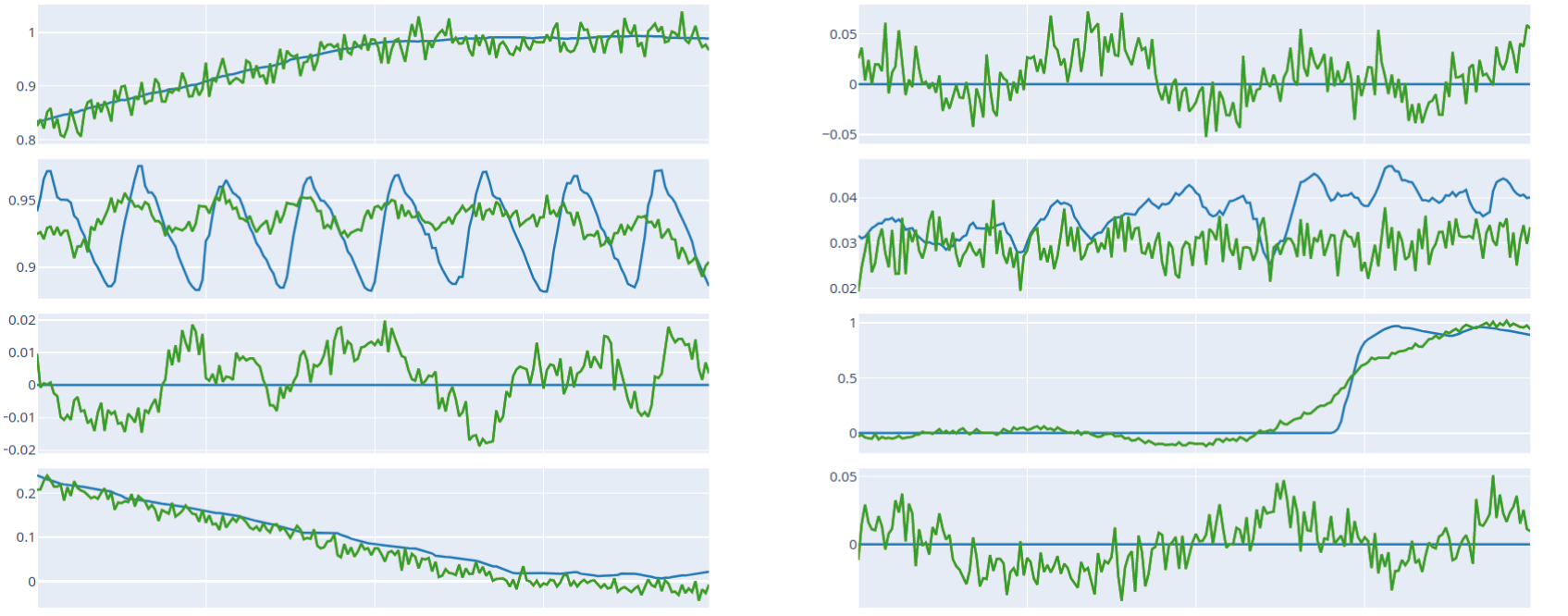


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

Observation: Sine pattern not captured in M1. Same true for M3 as well.

Sinusoidal pattern in DV0 captured by M1 (500 epochs training)

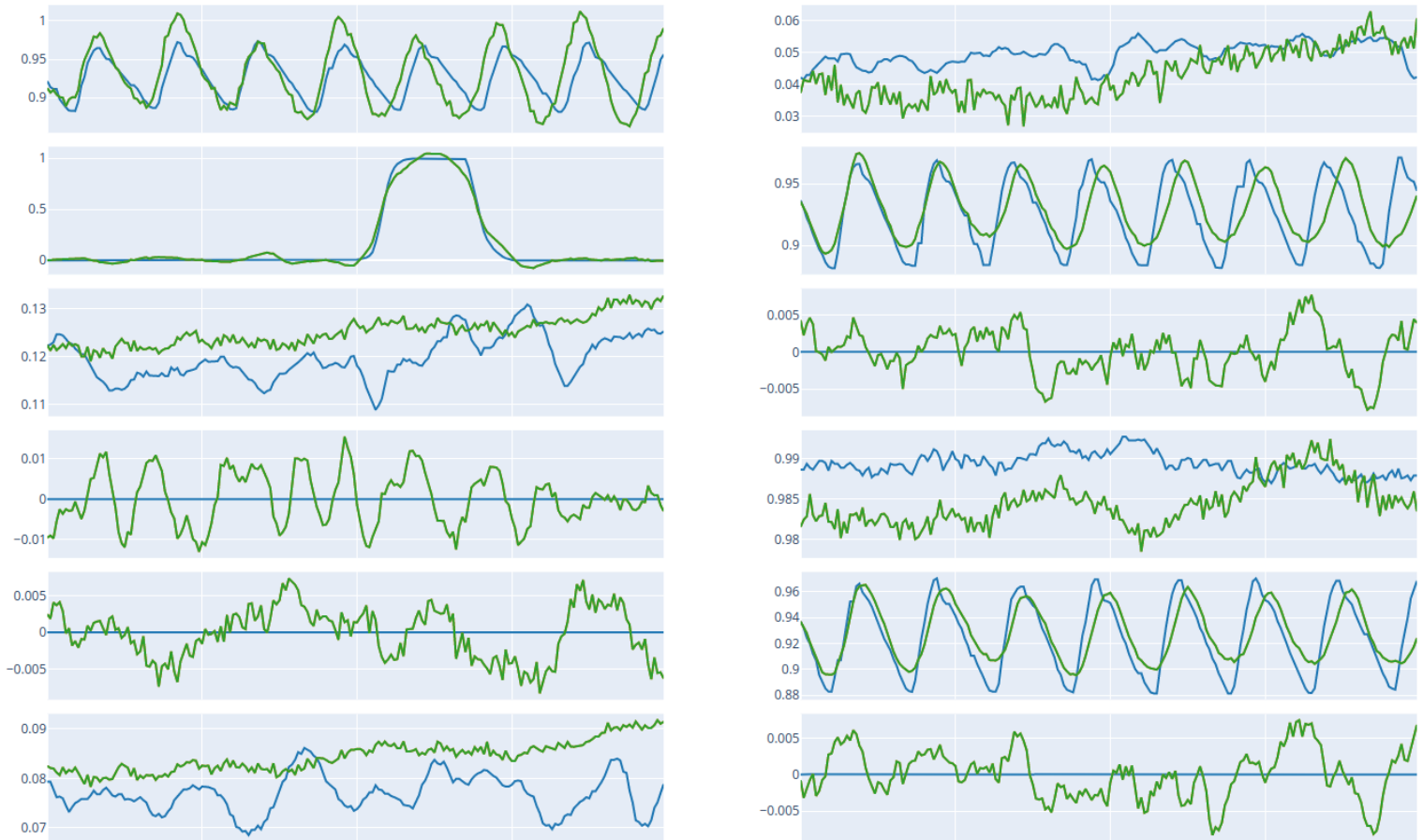


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

M3 fails to capture sinusoidal pattern even after 500 epochs of training

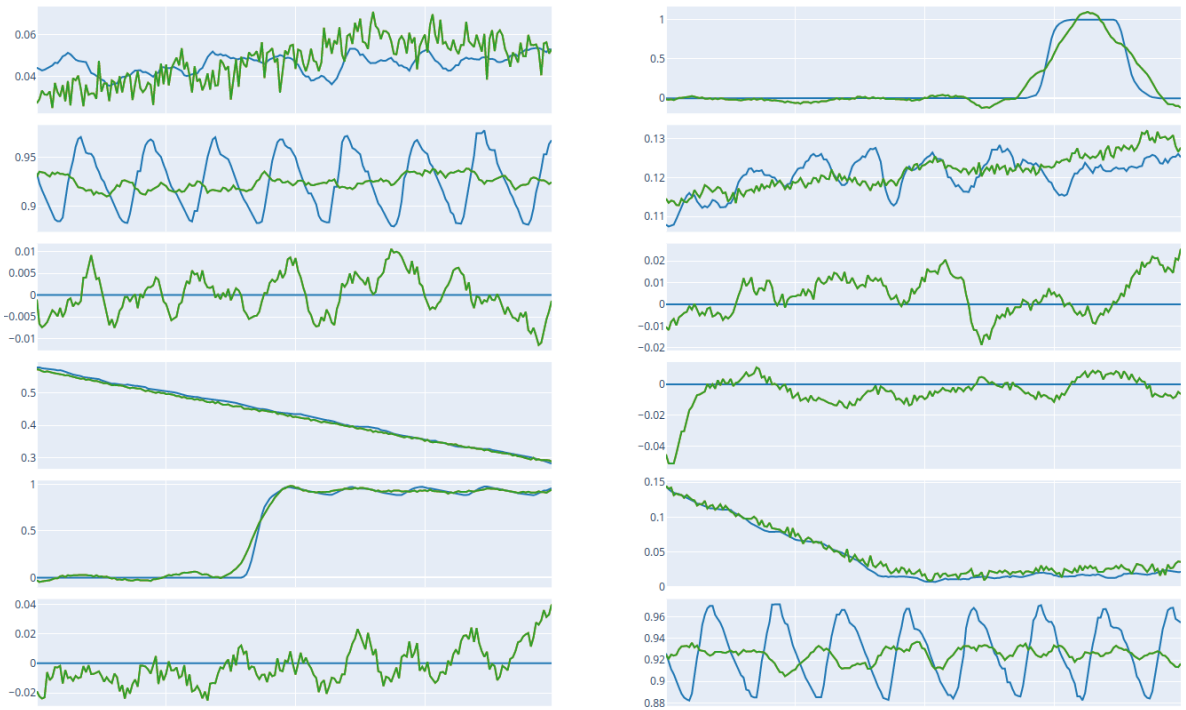


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

Hypothesis: CL discourages capturing sine pattern as it is present in all windows (positive as well as negative). Sine pattern acting as background?

M1 also struggles to predict sinusoidal pattern further into the future (step=4)

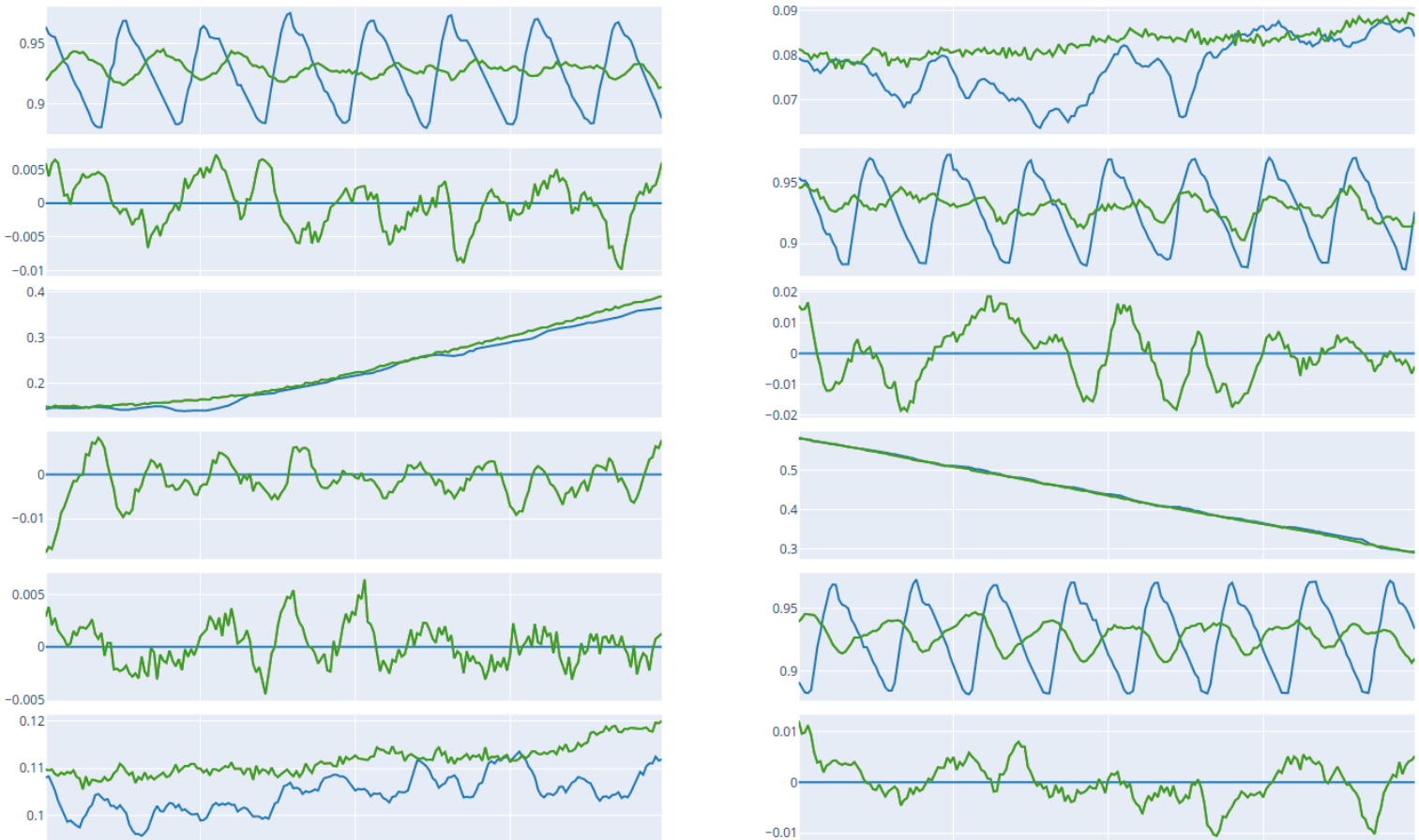
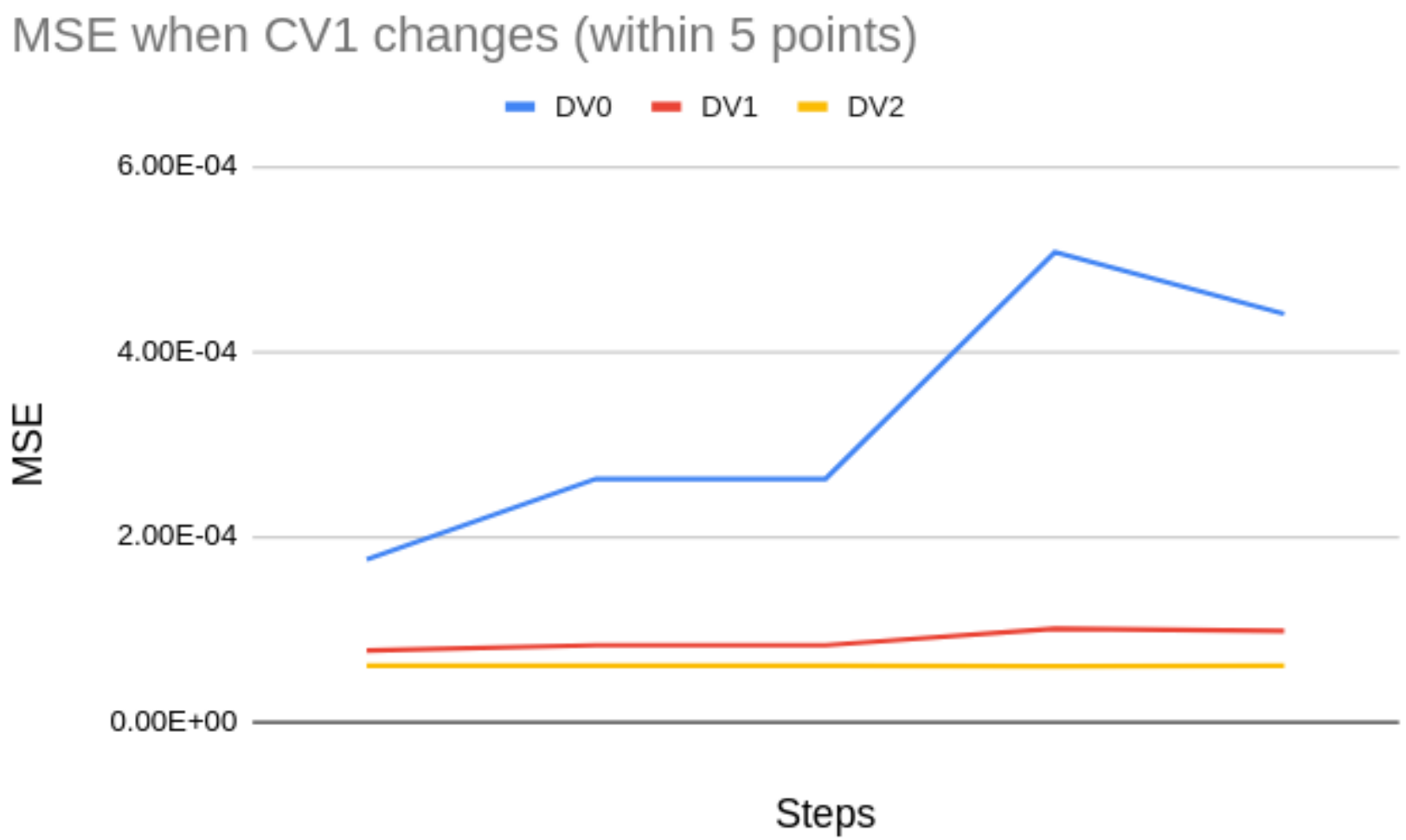


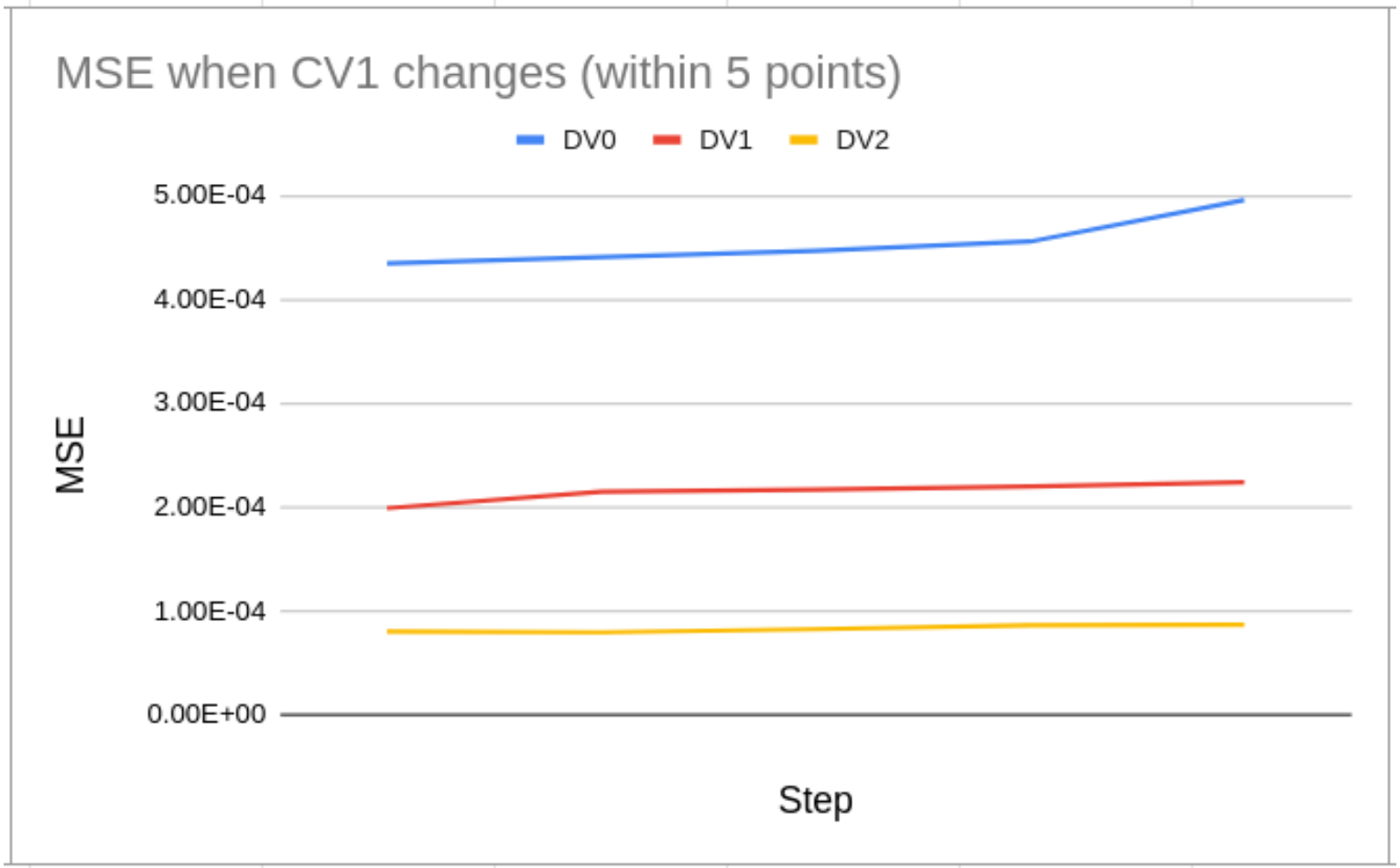
Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

M1 (MSE) fails to predict sinusoidal pattern in multi-step case



MSE+CL model in multi-step analysis

DV0 estimates from M3 are not good (compared to M1 model) from step=0 itself, indicating CL discourages detecting sine pattern.



SWaT2 Summary Results (500 epochs, separate encoder in SC case)

					SWaT (3 CVs, 3 DVs)							
					Validation				Test			
Description	Method	SC	PC	MLP or GNN or Concat	H@1	H@10	MRR	MSE	H@1	H@10	MRR	MSE
MSE with common control	**M1**	N	-	C				**2.04E-04**				**2.56E-04**
MSE with separate control	**M2**	Y	-	C				2.59E-04				3.21E-04
MSE+CL with common control	**M3**	N	Y	M	1.35E-01	7.00E-01	3.06E-01	4.53E-04	1.50E-01	7.08E-01	3.23E-01	5.10E-04
				G	1.51E-01	6.88E-01	3.16E-01	4.18E-04	1.62E-01	5.32E-01	3.34E-01	4.70E-04
MSE with common control **w2=0.0**	**M3**	N	Y	M	1.98E-03	1.98E-02	1.29E-02	***2.06E-04***	2.05E-03	2.46E-02	1.45E-02	***2.45E-04***
				G	0.00E+00	9.92E-03	6.62E-03	2.15E-04	0.00E+00	8.21E-03	6.81E-03	2.68E-04
MSE+CL with separate control	**M4**	Y	Y	M	4.37E-02	3.71E-01	1.44E-01	3.77E-04	6.16E-02	4.19E-01	1.67E-01	4.70E-04
				G	9.72E-02	5.77E-01	2.44E-01	4.71E-04	1.21E-01	6.26E-01	2.75E-01	5.14E-04
MSE with separate control **w2=0.0**	**M4**	Y	Y	M	1.98E-03	9.92E-03	1.12E-02	2.18E-04	2.05E-03	2.05E-02	1.13E-02	2.72E-04
				G	0.00E+00	1.98E-02	9.07E-03	3.04E-04	2.05E-03	2.05E-02	1.22E-02	3.18E-04

M3: $loss = w1 \times MSE + w2 \times CL, w1 = 1.0, w2 = 0.005$.
Observations:

- M3 (w2=0.005) worse than M1.
- M3 and M4 with w2=0.0 slightly better than M1 and M2, respectively. Implies better architecture (node-wise processing gives favorable inductive bias)?

M4 with w2=0.0 on SWaT2, DV-wise noise injection

noise=0.9

change	affect	DV0	DV1	DV2
	DV0	429.7%	1712.9%	429.0%
	DV1	326.8%	1684.8%	288.7%
	DV2	317.7%	1651.9%	286.6%
	All	408.0%	2164.3%	496.2%

noise=0.45

change	affect	DV0	DV1	DV2
	DV0	278.5%	998.4%	311.0%
	DV1	253.6%	1054.0%	119.8%
	DV2	213.7%	922.5%	180.4%
	All	350.3%	1523.1%	285.3%

noise=0.2

change	affect	DV0	DV1	DV2
	DV0	167.4%	557.8%	26.7%
	DV1	117.8%	509.9%	17.5%
	DV2	100.5%	305.7%	11.7%
	All	212.0%	772.6%	99.6%

noise=0.9

change	affect	DV0	DV1	DV2
	DV0	0.00220	0.00105	0.00080
	DV1	0.00167	0.00103	0.00054
	DV2	0.00163	0.00101	0.00053
	All	0.00209	0.00133	0.00092

noise=0.45

change	affect	DV0	DV1	DV2
	DV0	0.00143	0.00061	0.00058
	DV1	0.00130	0.00065	0.00022
	DV2	0.00109	0.00056	0.00034
	All	0.00179	0.00093	0.00053

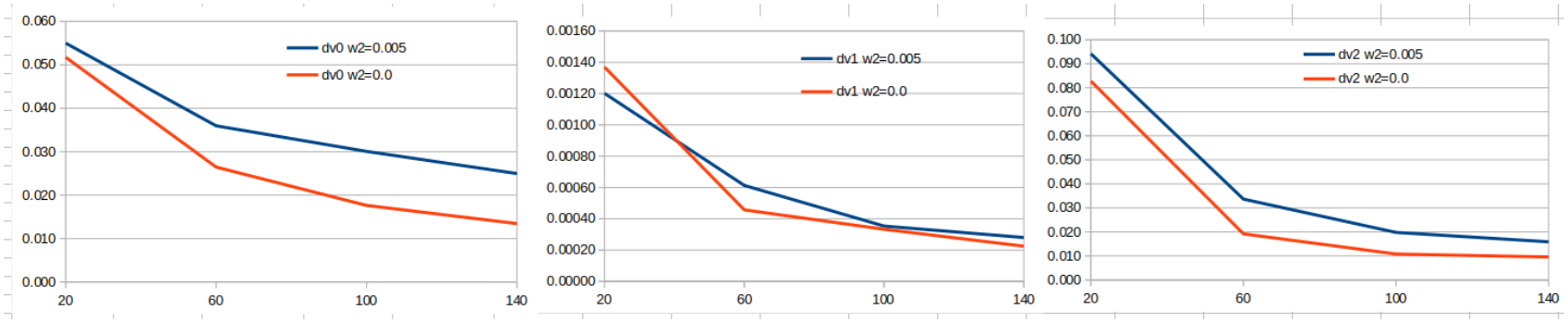
noise=0.2

change	affect	DV0	DV1	DV2
	DV0	0.00086	0.00034	0.00005
	DV1	0.00060	0.00031	0.00003
	DV2	0.00051	0.00019	0.00002
	All	0.00109	0.00047	0.00019

Additional Experiments

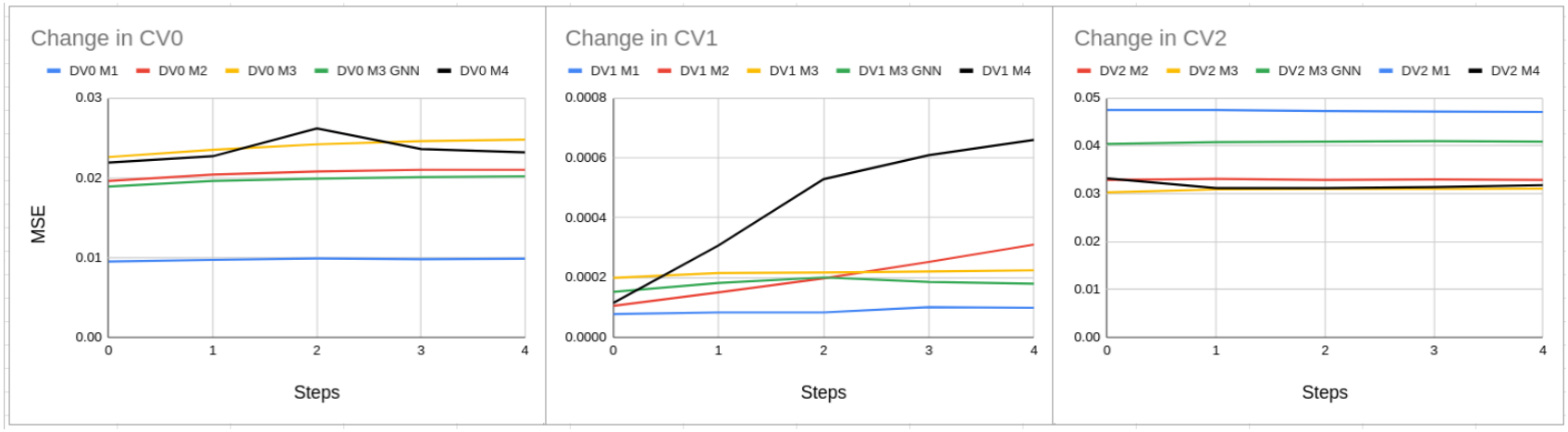
- New CL loss: Negative examples obtained using same past CVs and DVs but a randomly chosen future CV. This improves MSE on M3 (MLP) from 5.1x10-4 to 4.9x10-4, and M3 (GNN) from 4.7x10-4 to 4.4x10-4 on SWaT2. This indicates that the way to choose negative samples can be critical to the performance of CL methods.
- Similarly for SWaT3, new CL M4 (MLP transition model) gives 3.8x10-4 which is at par with M1 3.8x10-4. Furthermore, **new CL M4 (GNN) gives 3.7x10-4**. These are with common MLP per node.
- Since the effect of CVs on DVs can be different, tried a separate MLP per node in the transition model. This improves the results for M4 (MLP) from 4.7x10-4 to 4.3x10-4 for SWaT2.
- On SWaT2, M4 with **separate MLP** per node, **new CL**, w2=0.005 gives MSE of **2.9x10-4**, much better than 4.3x10-4 obtained using separate MLP per node but old CL method, indicating that **new CL is playing an important role**.
- On SWaT2, M4 with separate MLP per node and w2=0.0 gives MSE of **2.53x10-4** which is **better than M1 and M2**. Earlier with common MLP per node, this was 2.72x10-4.

M3 (w2=0.0 vs w2=0.005)



Observation: Adding CL loss to the MSE hurts performance on MSE even on points immediately after the control variable changes.

SWaT2: Comparison of all models for prediction just after change points



Observations:

- M1 is best on DV0 and DV1 on the points immediately after change in respective CV. M3 is best on DV2 when CV2 changes.
- M1 is better on DV0 because it captures sine pattern while M3 ignores it.

Sarcos Results for M1

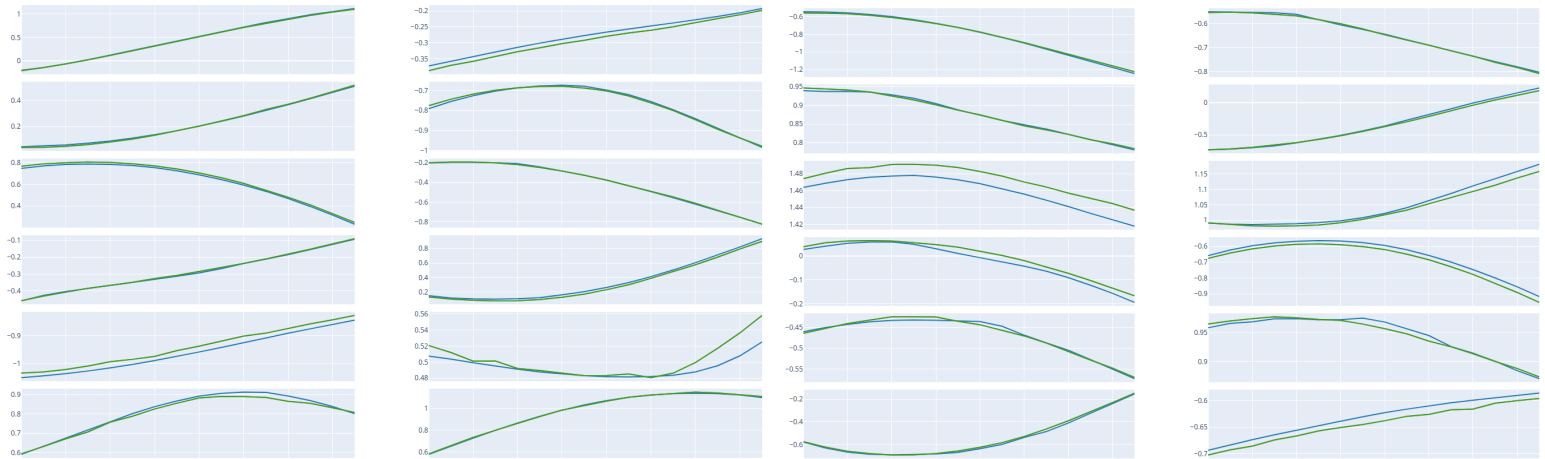


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

M1 MSE: 3.2x10-4, M3 MLP MSE: 5.0x10-4, M6 GNN MSE: 0.24, M6 MLP MSE: 0.05

Sarcos Results for M6 (only CL) GNN Transition Model

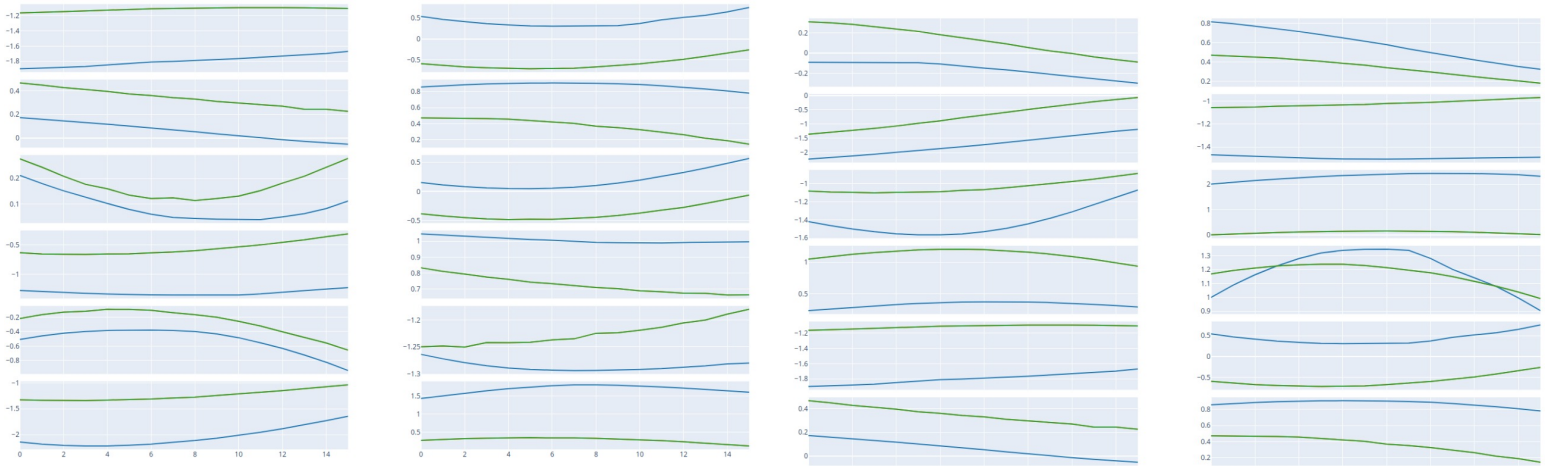


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

M1 MSE: 3.2x10-4, M3 MLP MSE: 5.0x10-4, M6 GNN MSE: 0.24, M6 MLP MSE: 0.05

Sarcos Results for M6 (only CL) MLP Transition Model

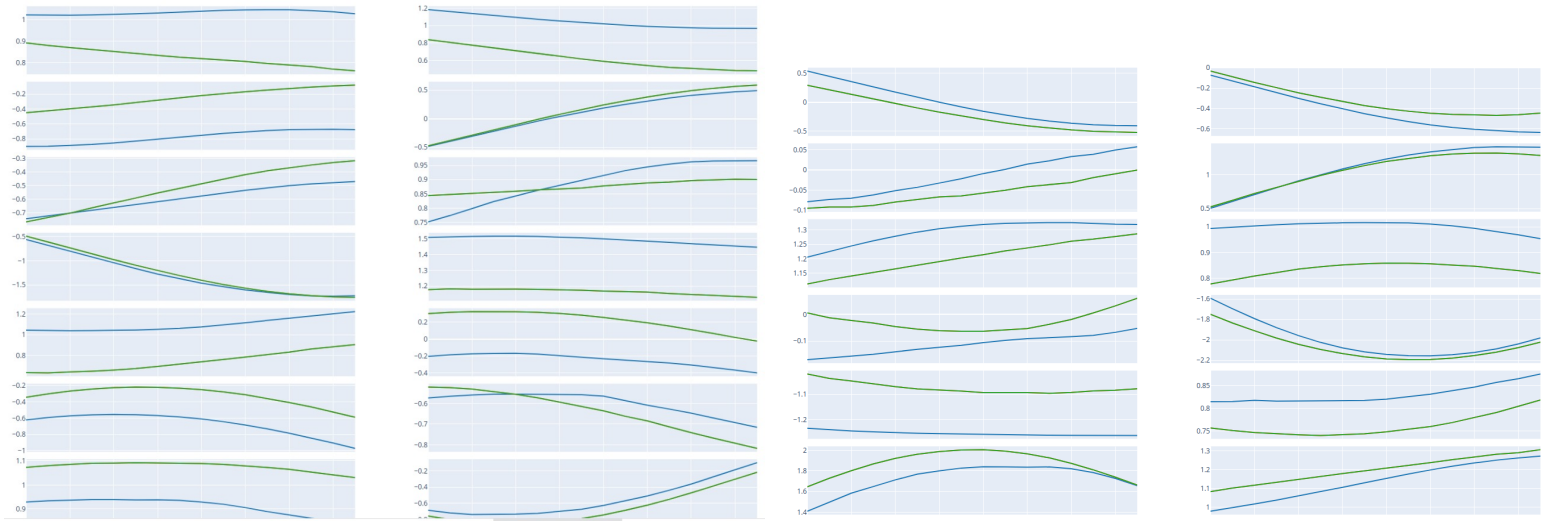


Figure: x-axis: time, y-axis: normalized sensor (DV) reading, Legend: prediction, target.

M1 MSE: 3.2x10-4, M3 MLP MSE: 5.0x10-4, M6 GNN MSE: 0.24, M6 MLP MSE: 0.05

Key aspects that we intend to explore (revisited)

- Contrastive Learning
  - More **robust to noise** compared with pure MSE based methods. This allows better multi-step ahead forecasting.
  - Better at **capturing subtle patterns**/changes that might get missed by MSE due to noise or assigning all resources to modeling the trivial patterns/background.
- Inductive bias in structure of neural network
  - attach each control to a different latent node (C-SWM [50], ICE-BeeM, IMCA [48])
  - nodes interact sparingly (NRI [51], PSD [94])
  - group-wise lasso while mapping latent embeddings to targets
  - Leads to **identifiable disentangled representations**
    - **combinatorial generalization** and **faster transfer**
    - Combinatorial generalization over controls (under certain assumptions).
    - Effect of noise restricted to fewer DVs; helps diagnosis in anomaly detection?
- Action Homomorphism [73]

Current Challenges and Thoughts

CL-related I

- CL getting stuck in local optimum. M6/M7 are much worse than M1-M4. Explore global-local loss [2, 6] or [70].
- In SWaT2 and SWaT3, adding CL (with a small weightage, w2=0.005) hurts mse performance. In SWaT2 it can be due to sine pattern, but why in SWaT3? Possibly because of poor neg. sampling.
- MSE+CL (M3) was doing well than MSE (M1) when trained for 50 epochs. Does it learn something faster than MSE? Also, it is doing better for DV2 near change points.
- CL is not able to model the abrupt changes well. Possibly the choice of pos./neg. samples is hurting. Have shifted versions of the time series as negative samples - then, it should learn faster and better. Also, this might not be an issue in GHL, as controls keep changing more frequently.

CL-related II

- CL can separate out background from what really matters. Background is something which changes normally but is not affected by any of the control variables. but, if change in a CV affects everything, then CL cannot outperform MSE.
- currently in swat, the change in DVs caused by change in CVs is huge and significant. therefore, CL won't have any advantage over MSE.
- Why does CL help in speech application in CPC without needing external control? is multi-step ahead the key (with diff. W_k for every k-th step)? they have evaluated for speaker classification and H@1 but never for forecasting. they also choose the neg. samples in non-random ways.
- also see CPC-v2 [33]. they say that BN hurts CPC while LN can be useful.

CL-related III

- Finding failure cases of MSE:
  - Can we first look at failure cases of MSE in SWaT, and try to address them? e.g. sine pattern not predicted well in multi-step - increasing MSE with steps.
  - Try on a noisy dataset where noise starts acting like background. Sine wave is acting like background in DV0.
  - We have not tried "training" on noisy setups yet. The behavior of M1-M2 may be completely different there, and they might struggle to learn properly in comparison to M3-M4.
  - Have very high-dimensional time series with large number of DVs and CVs. Then, change in one CV will affect only a few DVs with the rest of the DVs acting as background. What happens when CVs are unknown?

CL on its own not good enough (CPCv2 [33] ICML'20)

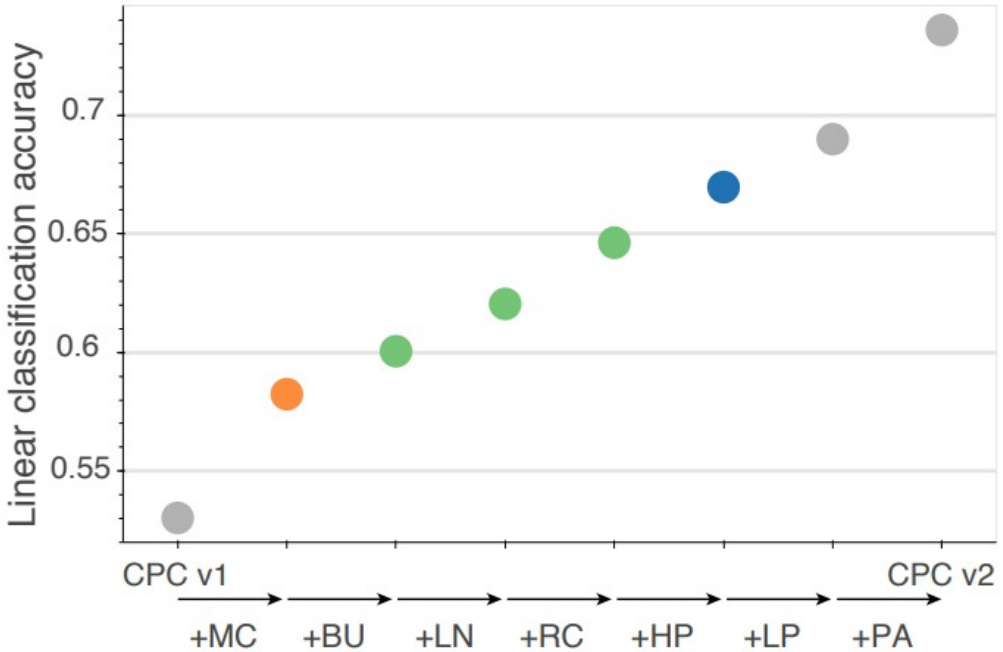


Figure 3: Linear classification performance of new variants of CPC, which incrementally add a series of modifications. MC: model capacity. BU: bottom-up spatial predictions. LN: layer normalization. RC: random color-dropping. HP: horizontal spatial predictions. LP: larger patches. PA: further patch-based augmentation. We use color to indicate the number of spatial predictions used (orange, green, blue for 1, 2 and 4 directions). Note that these accuracies are evaluated on a custom validation set and are therefore not directly comparable to the results we report and compare to.

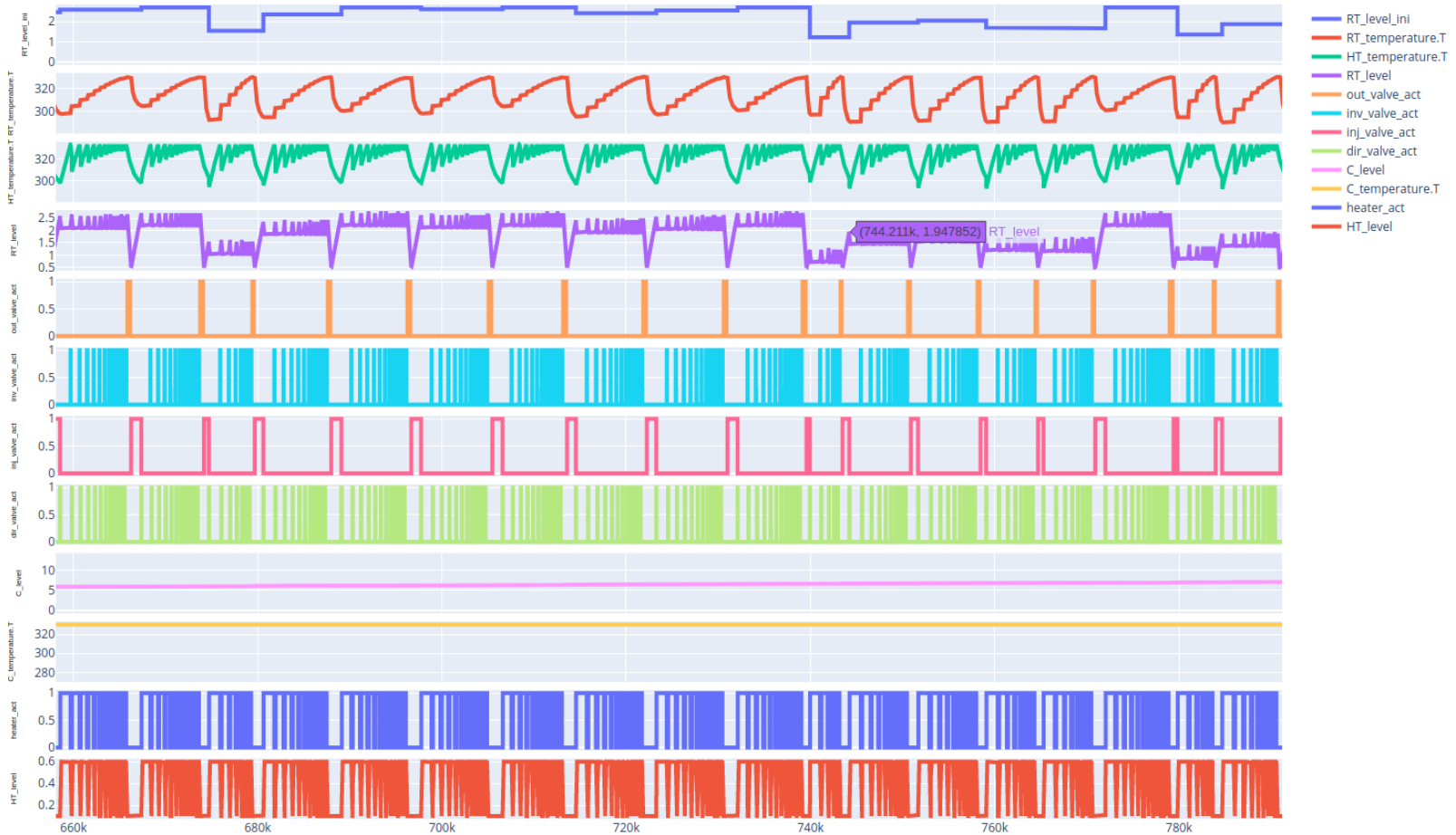
Architecture-related

- Separate MLP per node as the effects of CVs on DVs is of different kinds.
- control encoder timing (causal) issues - effect of change in control gets reflected earlier.
- Autoregressive model in the latent space is needed. iteratively call the MLP over estimated embeddings (as in NRI [51]). then, the control can be applied independently at each time step - this will get rid of timing and control encoder related issues.
- SC vs CC: SC should help with disentanglement which should help with robustness to noise, multi-step predictions, OOD generalization. baselines: i. no notion of nodes (M1,M2), ii. notion of nodes but common control (M3).
- VAE vs AE - why does Temporal Segments [66] use VAE? even NRI uses VAE formulation.
- does disentanglement make sense in our current non-generative models?

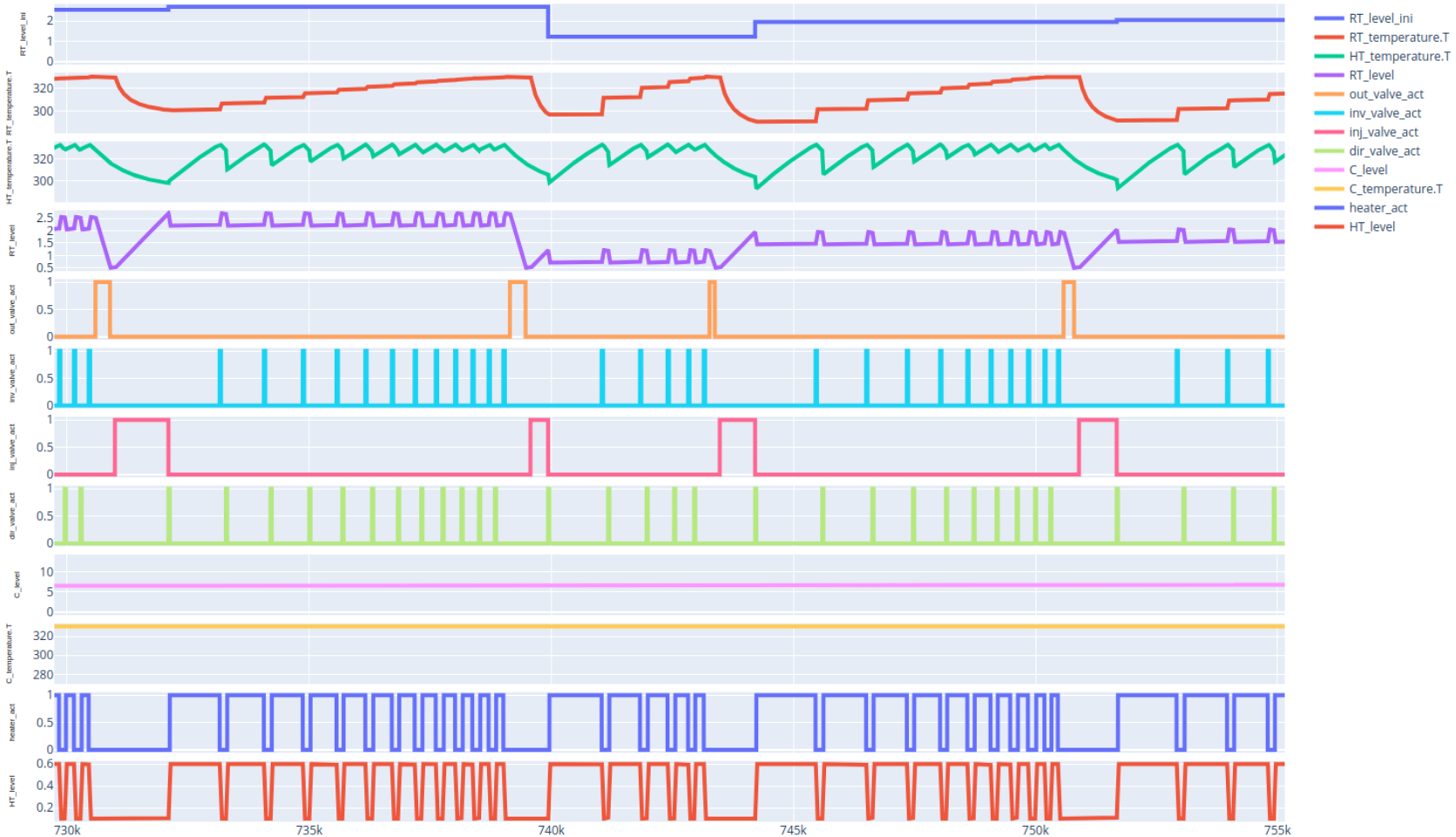
Datasets-related

- SWaT not suited for CL as-it-is unless we look at the DVs with subtler changes like the ones in LIT301, FIT301, DPIT301.
- Look at GHL. GHL has these subtler patterns which may be non-trivial to capture for MSE model.
- Look at electrical motor usecase [89].
- own toy dataset - high-dimensional. can enhance GHL also by adding delays, negative correlations, etc.

GHL-1


RT_level_ini
RT_temperature.T
HT_temperature.T
RT_level
out_valve_act
inv_valve_act
inj_valve_act
dir_valve_act
C_level
C_temperature.T
heater_act
HT_level
(744.211k, 1.947852) RT_level
660k
680k
700k
720k
740k
760k
780k

GHL-2


RT_level_ini
RT_temperature.T
HT_temperature.T
RT_level
out_valve_act
inv_valve_act
inj_valve_act
dir_valve_act
C_level
C_temperature.T
heater_act
HT_level
730k
735k
740k
745k
750k
755k

Combinatorial Generalization Testing

- **Goal**: generalize well on forecasting dependent variables (DVs) under previously unseen combinations of states.
- Scenarios
  1. CVs observed
  2. CVs unobserved
- DVs are by-definition observed in both scenarios.
- Start with Scenario-1 - should be easier.
- **Hypotheses**:
  1. M4 (without CL, i.e. w2=0) is better than M1 and M2.
  2. M2 is better than M1.

Causal Time Series Generation with Exogeneous Variables[5]

Synthetic data: We first obtain the S exogenous time series through the following Non-linear Autoregressive Moving Average (NARMA) (Atiya & Parlos, 2000) generators:

$$x_{i,t}^s = \alpha_s x_{i,t-1}^s + \beta_s x_{i,t-1}^s \sum_{j=1}^{d} x_{i,t-j}^s + \gamma_s \varepsilon_{i,t-d} \varepsilon_{i,t-1} + \varepsilon_{i,t}, \tag{14}$$

where ε_t are zero-mean noise terms of 0.01 variance, d is the order of non-linear interactions, and α_s, β_s and γ_s are parameters specific to variable s, generated from $\mathcal{N}(0, 0.1)$. Then, we generate the target series from the generated exogenous series via the formula:

$$y_{i,t} = \sum_{s=1}^{S} \omega_i^s (\boldsymbol{\eta}_i^s)^\mathsf{T} \tanh\left(\boldsymbol{x}_{i,t-p:t-1}^s\right) + \varepsilon_{i,t}, \tag{15}$$

where $\omega_i^s \in \{0, 1\}$ with 0.6 probability of being zero that controls the underlying causal relationship from the s-th variable to the target variable, $\boldsymbol{\eta}_i^s \in \mathbb{R}^p$ controls the causal strength sampling from Unif$\{-1, 1\}$, and $\boldsymbol{x}_{i,t-p:t-1}^s = (x_{i,t-p}^s, x_{i,t-p+1}^s, \ldots, x_{i,t-1}^s)^\mathsf{T} \in \mathbb{R}^p$ represents the last p historical values of variable s of sample i. The 0-1 indicator vector $\boldsymbol{\omega}_i = (\omega_i^1, \omega_i^2, \ldots, \omega_i^S)^\mathsf{T} \in \mathbb{R}^S$ is the ground-truth causal structure of i-th individual.

[5] Granger Causal... Heterogeneus Multivariate Time Series https://openreview.net/pdf?id=SJxyCRVKvB

Gaussian Copula Processes [76]

Synthetic experiment. We first perform an experiment on synthetic data demonstrating that our approach can recover complex time-varying low-rank covariance patterns from multi-dimensional observations. An artificial dataset is generated by drawing T observations from a normal distribution with time-varying mean and covariance matrix, $\mathbf{z}_t \sim \mathcal{N}(\rho_t \mathbf{u}, \Sigma_t)$ where $\rho_t = \sin(t)$, $\Sigma_t = U S_t U^T$ and

$$S_t = \begin{bmatrix} \sigma_1^2 & \rho_t \sigma_1 \sigma_2 \\ \rho_t \sigma_1 \sigma_2 & \sigma_2^2 \end{bmatrix}$$

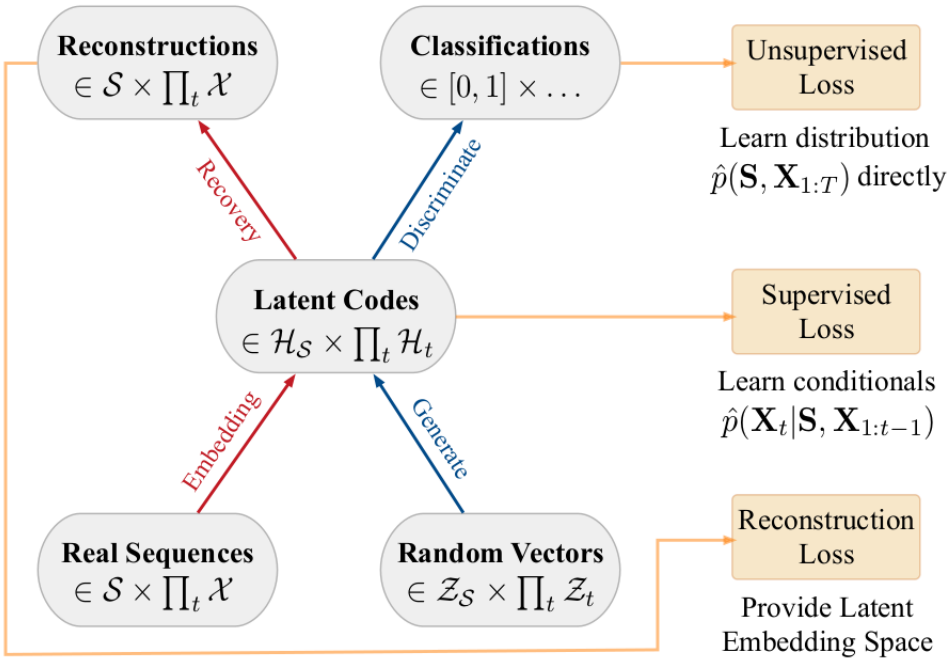
The coefficients of $\mathbf{u} \in \mathbb{R}^{N \times 1}$ and $U \in \mathbb{R}^{N \times r}$ are drawn uniformly in $[a, b]$ and σ_1, σ_2 are fixed constants. By construction, the rank of Σ_t is 2. Both the mean and correlation coefficient of the two underlying latent variables oscillate through time as ρ_t oscillates between -1 and 1. In our experiments, the constants are set to $\sigma_1 = \sigma_2 = 0.1, a = -0.5, b = 0.5$ and $T = 24,000$.

Combinatorial Generalization Testing - Toy Simulation

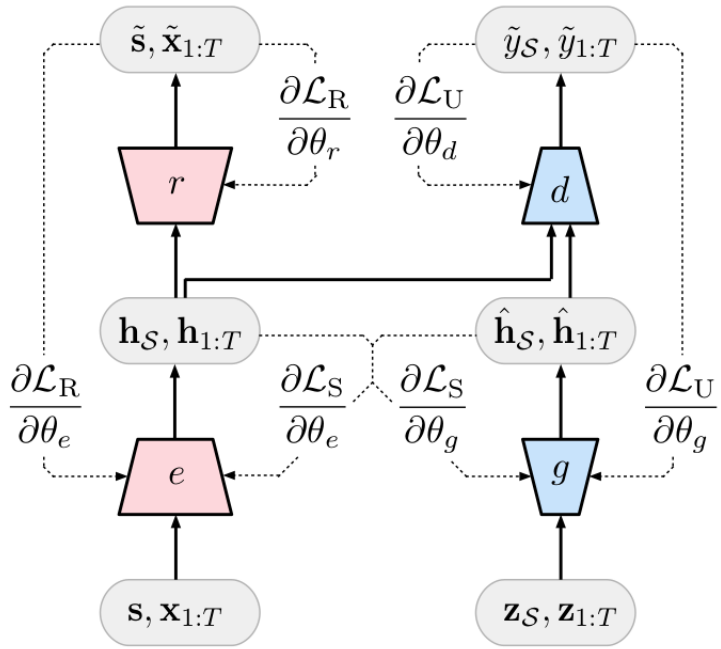
Enhance using ideas from previous slides

- 4 CVs, each CV can have 4 possible ordinal values, say (0,1,2,3).
- Therefore, there are total $4^4 = 256$ states.
- Some of them (say, 200 (80%)) are encountered in the training+validation set. Test on remaining 20% states
- Each CV affects mutually exclusive set of DVs (each set with cardinality $n = 1, 2, 4$).
- Each DV is affected by only one CV
- The n DVs associated with a CV are related
  - i. -ve correlation, ii. +ve correlation, iii. +ve correlation with different levels of noise.
- Given its CV, the dynamics of a DV are independent of other DVs (though they may still be correlated).
- Effect of a CV on a DV (inspired by SWaT):
  - i. Sudden, ii. Slope Change, iii. Transient, iv. Delayed, (later: v. Frequency change)

TimeGAN-I [99]

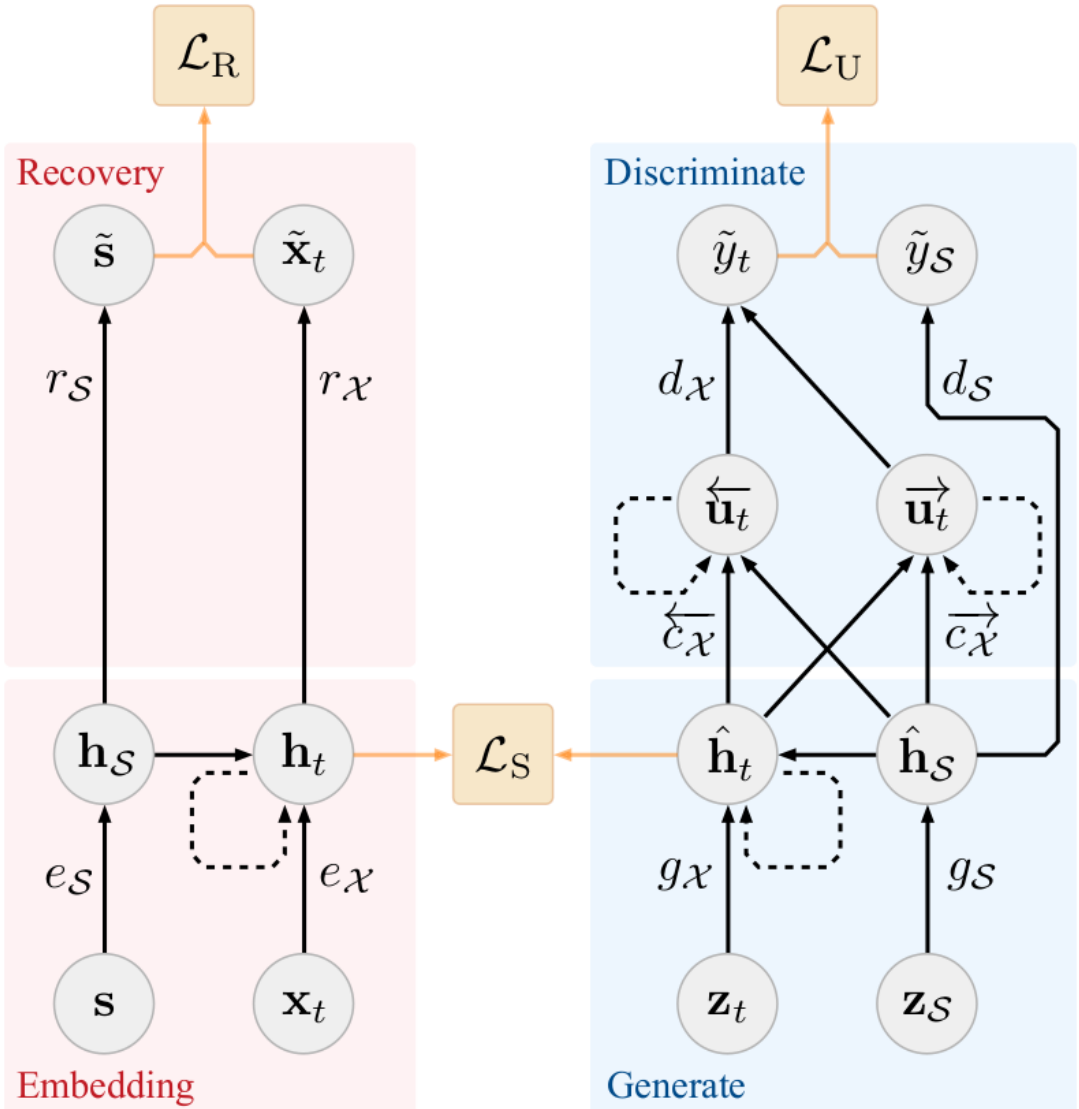


(a) Block Diagram



(b) Training Scheme

TimeGAN-II



(a) TimeGAN

Simulations used in TimeGAN

follows: $\mathbf{x}_t = \phi\mathbf{x}_{t-1} + \mathbf{n}$, where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{1} + (1-\sigma)\mathbf{I})$. The coefficient $\phi \in [0, 1]$ allows us to control the correlation across time steps, and $\sigma \in [-1, 1]$ controls the correlation across features.

Figure: Autoregressive Gaussian Models

(1) Sines. We simulate multivariate sinusoidal sequences of different frequencies η and phases θ, providing continuous-valued, periodic, multivariate data where each feature is independent of others. For each dimension $i \in \{1, ..., 5\}$, $x_i(t) = \sin(2\pi\eta t + \theta)$, where $\eta \sim \mathcal{U}[0, 1]$ and $\theta \sim \mathcal{U}[-\pi, \pi]$.

Figure: Sines

Sample real (input) time series in TimeGAN

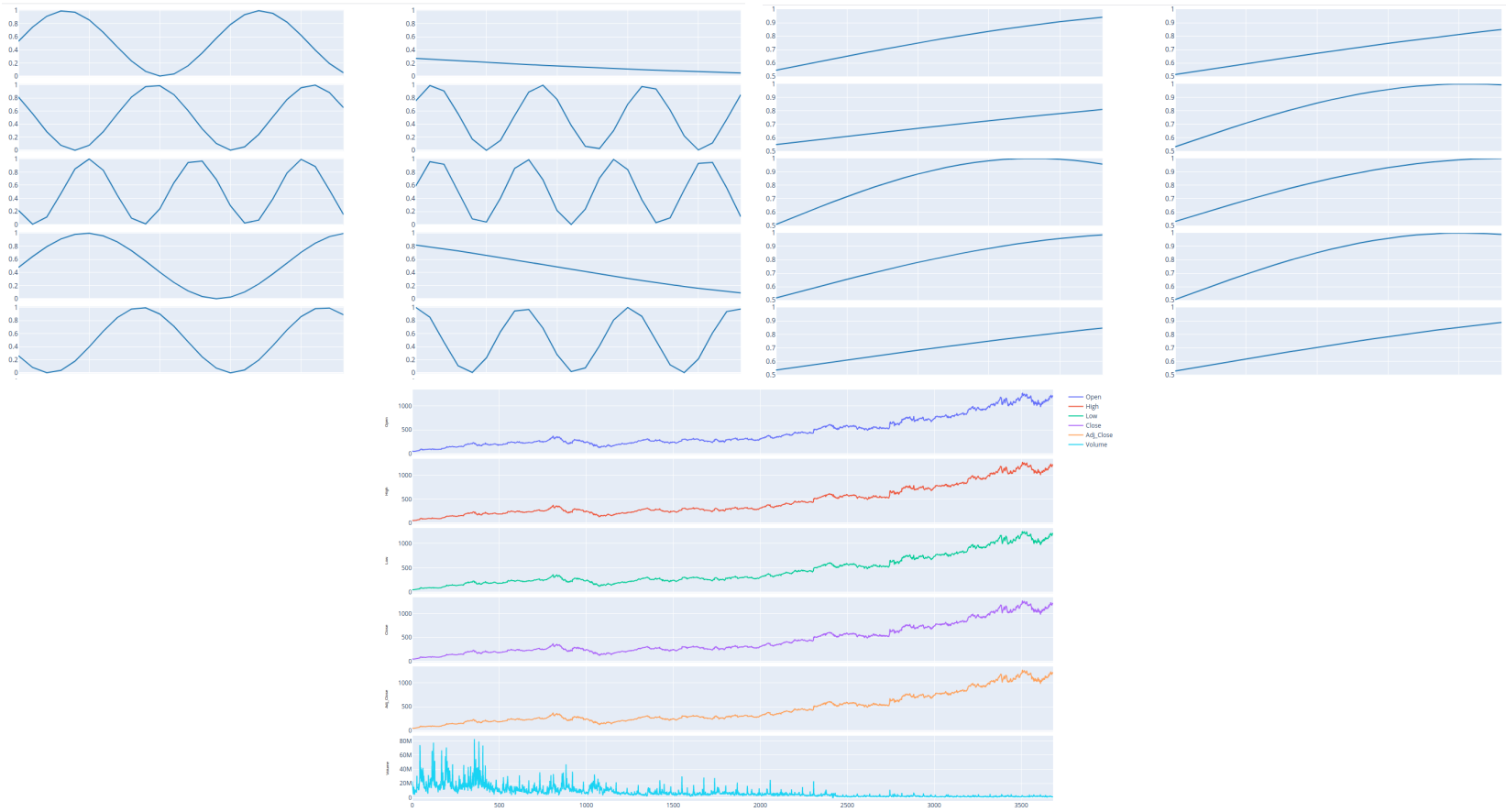
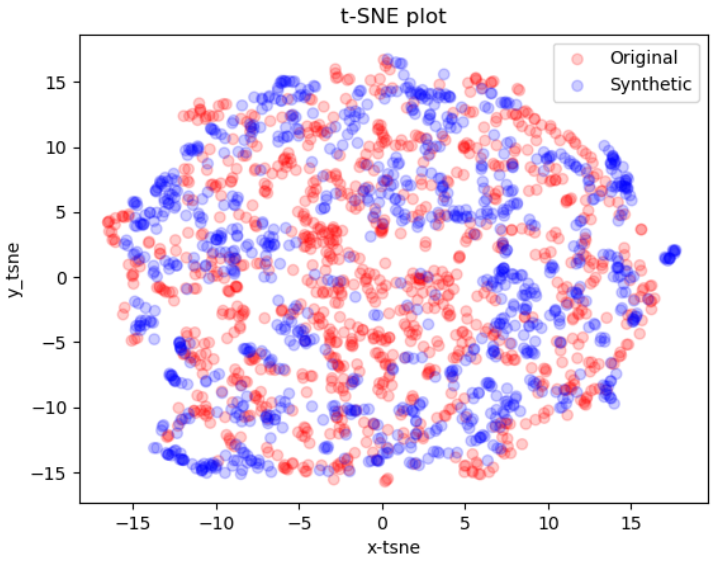


Figure: Top: Sines, Bottom: Stocks

TimeGAN - exp-1

(1) Sines. We simulate multivariate sinusoidal sequences of different frequencies η and phases θ, providing continuous-valued, periodic, multivariate data where each feature is independent of others. For each dimension $i \in \{1, ..., 5\}$, $x_i(t) = \sin(2\pi\eta t + \theta)$, where $\eta \sim \mathcal{U}[0, 1]$ and $\theta \sim \mathcal{U}[-\pi, \pi]$.



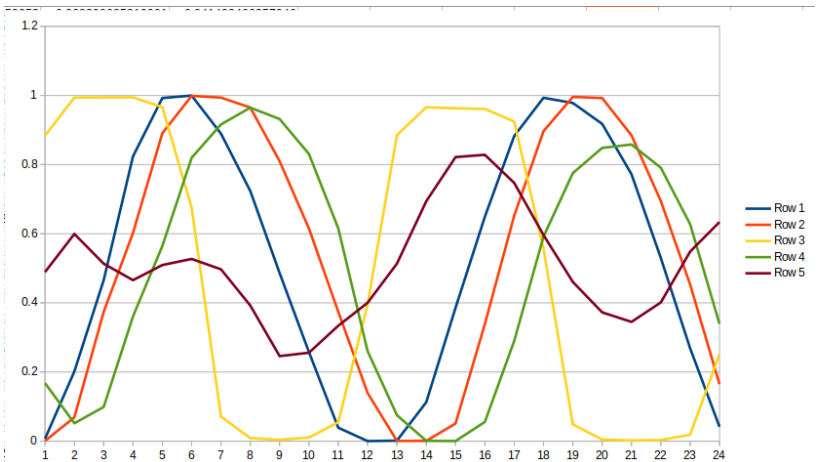


Figure: Sines, tSNE for $\eta \sim \mathcal{U}[0.4, 0.6]$.

freq. 0.4-0.6, bs 128, N=10000, dims=5, 50k iterations, Discriminative Score - 0.371, Predictive Score - 0.3888

TimeGAN - exp-2

(1) Sines. We simulate multivariate sinusoidal sequences of different frequencies η and phases θ, providing continuous-valued, periodic, multivariate data where each feature is independent of others. For each dimension $i \in \{1, ..., 5\}$, $x_i(t) = \sin(2\pi\eta t + \theta)$, where $\eta \sim \mathcal{U}[0, 1]$ and $\theta \sim \mathcal{U}[-\pi, \pi]$.

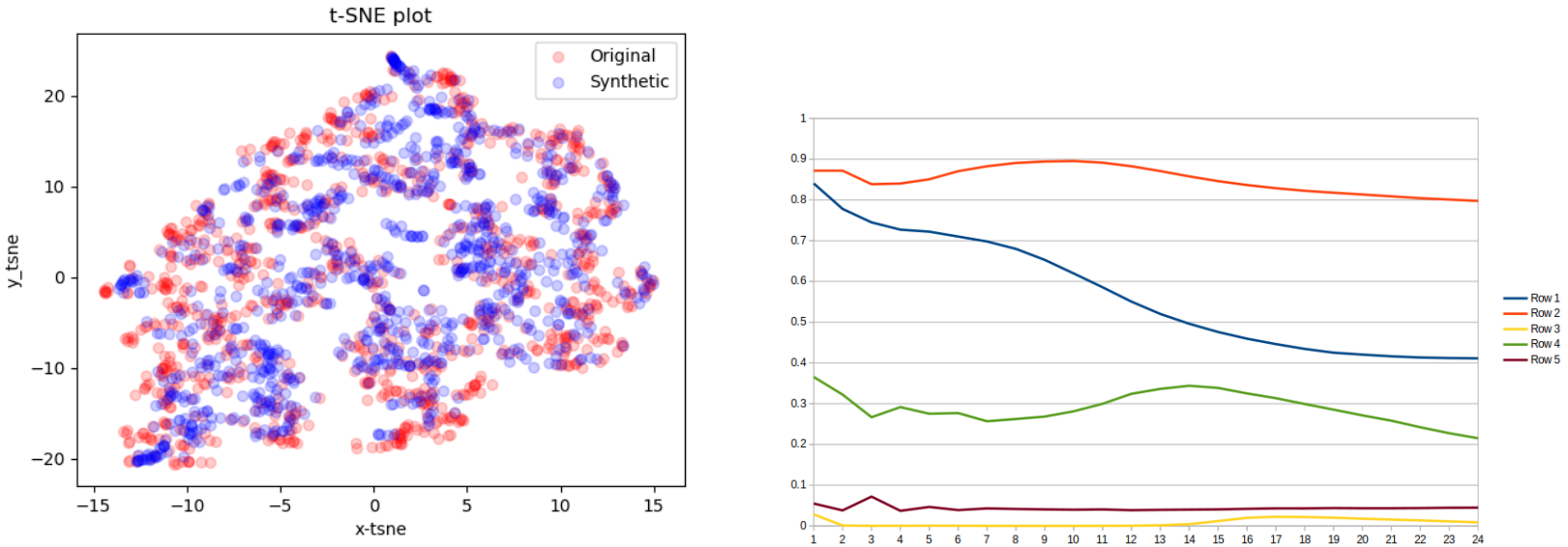
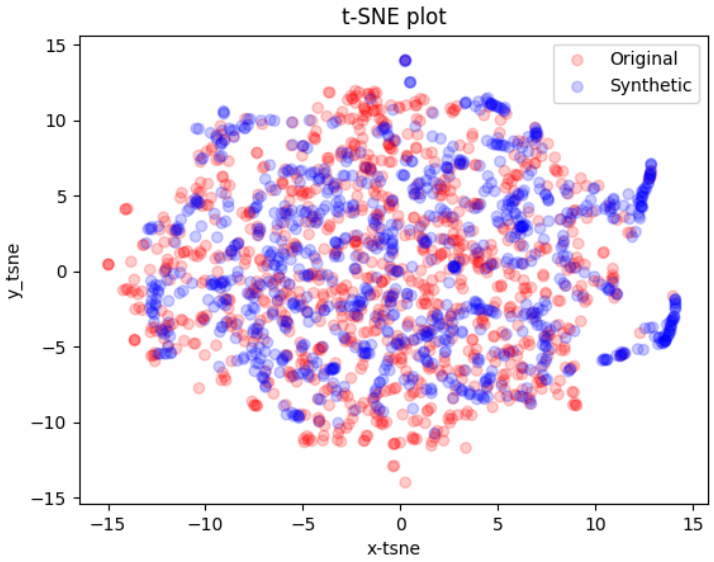


Figure: Sines, tSNE for $\eta \sim \mathcal{U}[0.0, 0.1]$.

freq. 0.0-0.1, bs 128, N=10000, dims=5, 50k iterations, Discriminative Score: 0.037, Predictive Score: 0.344

TimeGAN - exp-3

(1) Sines. We simulate multivariate sinusoidal sequences of different frequencies η and phases θ, providing continuous-valued, periodic, multivariate data where each feature is independent of others. For each dimension $i \in \{1, ..., 5\}$, $x_i(t) = \sin(2\pi\eta t + \theta)$, where $\eta \sim \mathcal{U}[0, 1]$ and $\theta \sim \mathcal{U}[-\pi, \pi]$.



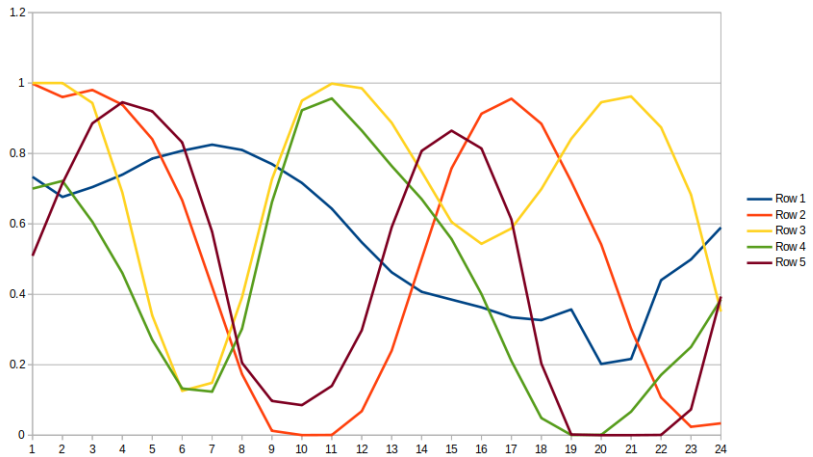


Figure: Sines, tSNE for $\eta \sim \mathcal{U}[0.0, 1.0]$.

freq. 0.0-1.0, bs 128, N=10000, dims=5, 50k iterations, Discriminative Score - 0.2178, Predictive Score - 0.3515

Electric Dynamic Motor [89]

Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Conv1	10	3	100	32	91
Conv2	7	32	91	64	85
Conv3	5	64	85	128	81
Conv4	3	128	81	256	79

Encoder Parameters

Layer	Kernel Size	#Input Features	Input Length	#Ouput Features	Output Length
Dconv1	10	96	91	3	100
Dconv2	7	192	85	32	91
Dconv3	5	384	81	64	85
Dconv4	3	768	79	128	81

Decoder Parameters

Layer	Hidden Vector Size	#Input Features	Input Length	#Ouput Features	Output Length
RNN1	32	32	91	64	91
RNN2	64	64	85	128	85
RNN3	128	128	81	256	81
RNN4	256	256	79	512	79

Recurrent Skip Connection Parameters

All RNNs are bidirectional

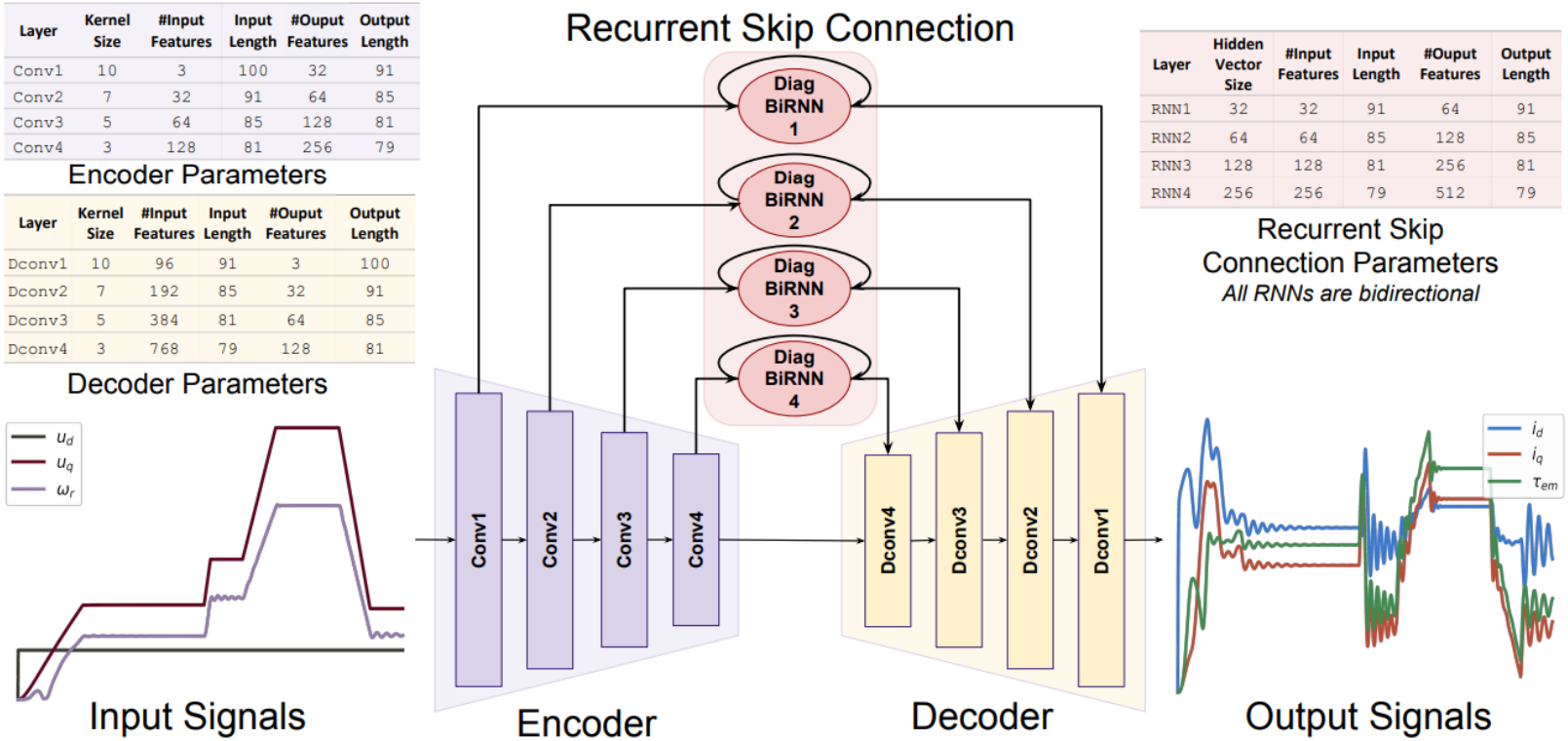


Figure 2: Proposed architecture.

Generative Models for Multivariate Time Series (Extending TimeGAN)

- TimeGAN [99] as such looks like a special case of FlexAE, except for the temporal aspect. Should be possible to extend theory.
- Enhance TimeGAN along: i. inductive bias in the latent space, ii. combinatorial generalization under conditional generation, iii. identifiability under conditional generation.
- TimeGAN paper does not show any simulated time series - there may a catch in that. Need to first check if their results are reliable.
- Analyze the failure cases for TimeGAN. It may not work well on some of these datasets: SWaT, GHL, Causal Toy Simulations, Electric Motors (difficult as per [89]) datasets without control variables, and on electricity demand forecasting dataset (inspired from MTGNN [92] as GNN helps).

Evaluation metrics and loss function-related

- MSE is not a good evaluation metric when:
  - time series is noisy
  - there are "background" patterns that are not critical and need not be modeled. In other words, background is something for which poor mse does not hurt the downstream tasks.
- In multi-step evaluation, getting the critical predictions wrong should hurt on subsequent predictions. in current swat setups, doing badly on first step may or may not hurt on second step.
- Other loss functions to handle abrupt changes [90, 89].
- normalized mse: smape as used in [89]

General

- think from perspective of multivariate time series forecasting [79] (without control variables). in very high-dimensional scenario, CL may have some advantages, and so would node-level architectures.
- think from perspective of anomaly detection. sensor relevance ranking evaluation in anomaly detection.
- what if the control variables are unknown? can they be learned from the data? then it becomes a real identifiability problem. Are TCL [40, 41], ICE-BeeM [48] attempting to solve this?
- relation of all this to causality and counterfactual reasoning [58].
- Ideas from Time2Graph [15].
- PSD idea (relational inference as in NRI [51], [57])
- Application of Predictive Coding in Control [81]

Overview

- Explore inductive biases in the neural network design to enable better disentanglement for multivariate time series data
- Two key ideas:
  - Have a separate encoder for each control variable
  - Have a separate node (transition model) in the latent space for each control variable

Experimental Setup

- **Data**:
  - Generated time series with two control variables (CVs) and two dependent variables (DVs) as per the following equation:

$$y^{(i)}(t) = \alpha_1^{(i)} y^{(i)}(t-1) + \alpha_2^{(i)} u^{(i)}(t) u^{(i)}(t-10) + \alpha_3^{(i)}, i = 0, 1 \tag{16}$$

  - So, one DV (y) depends on only one CV (u).
  - $\alpha_1^{(0)} = 0.3, \alpha_2^{(0)} = 1.5, \alpha_3^{(0)} = 0.1$
  - $\alpha_1^{(1)} = 0.15, \alpha_2^{(1)} = 0.55, \alpha_3^{(1)} = 0.2$
- Input: 100 length time series for DVs, 100+5 for CVs, Output: 5-step ahead forecasts, MSE loss.
- **Hyperparameters**: learning rate:0.001, Adam optimizer, L1 regularizer weight: 0.001, batch size:128, training instances: 8k, test instances: 2k.

Sample Data

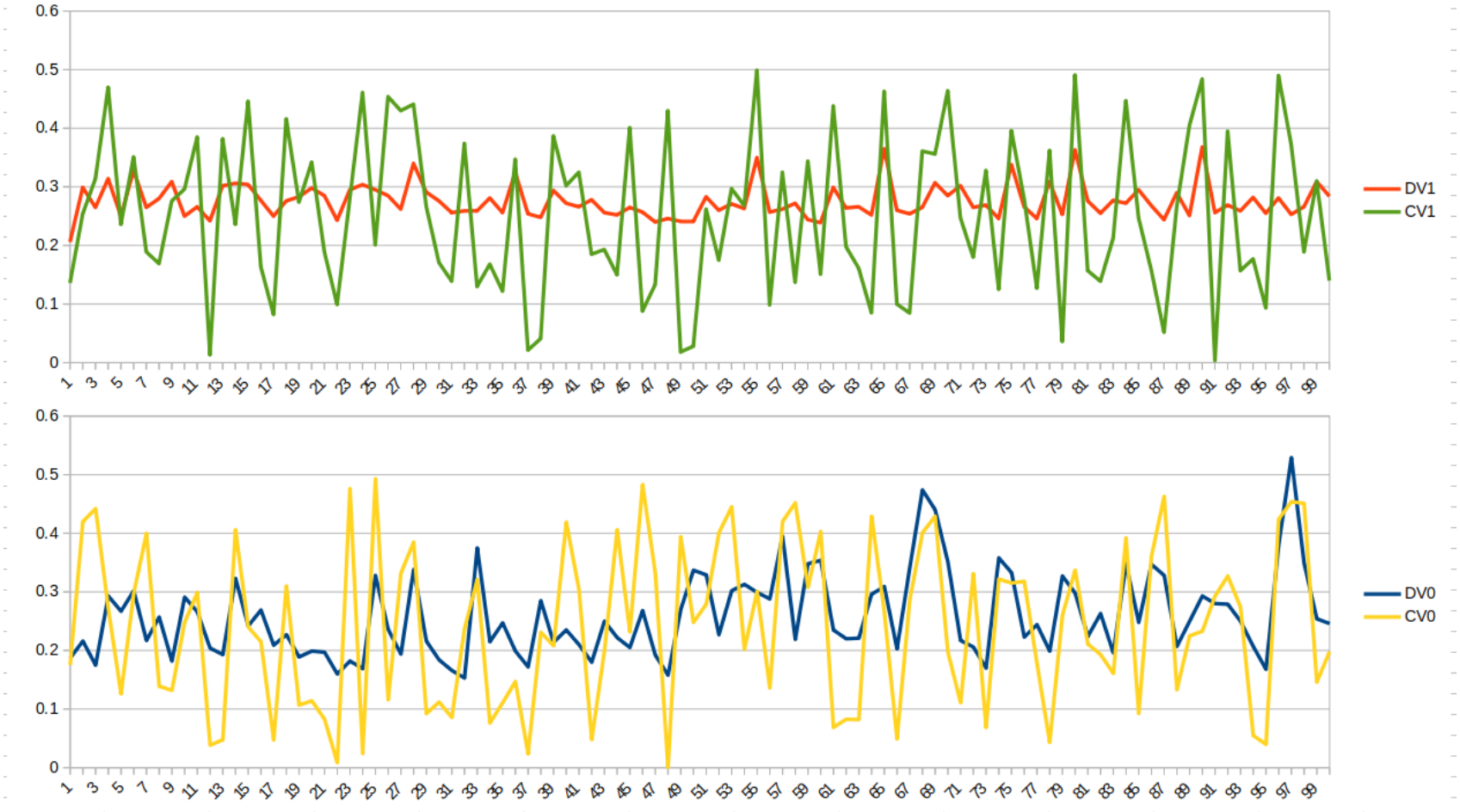


Figure: Sample Data

Vanilla Approach for Forecasting

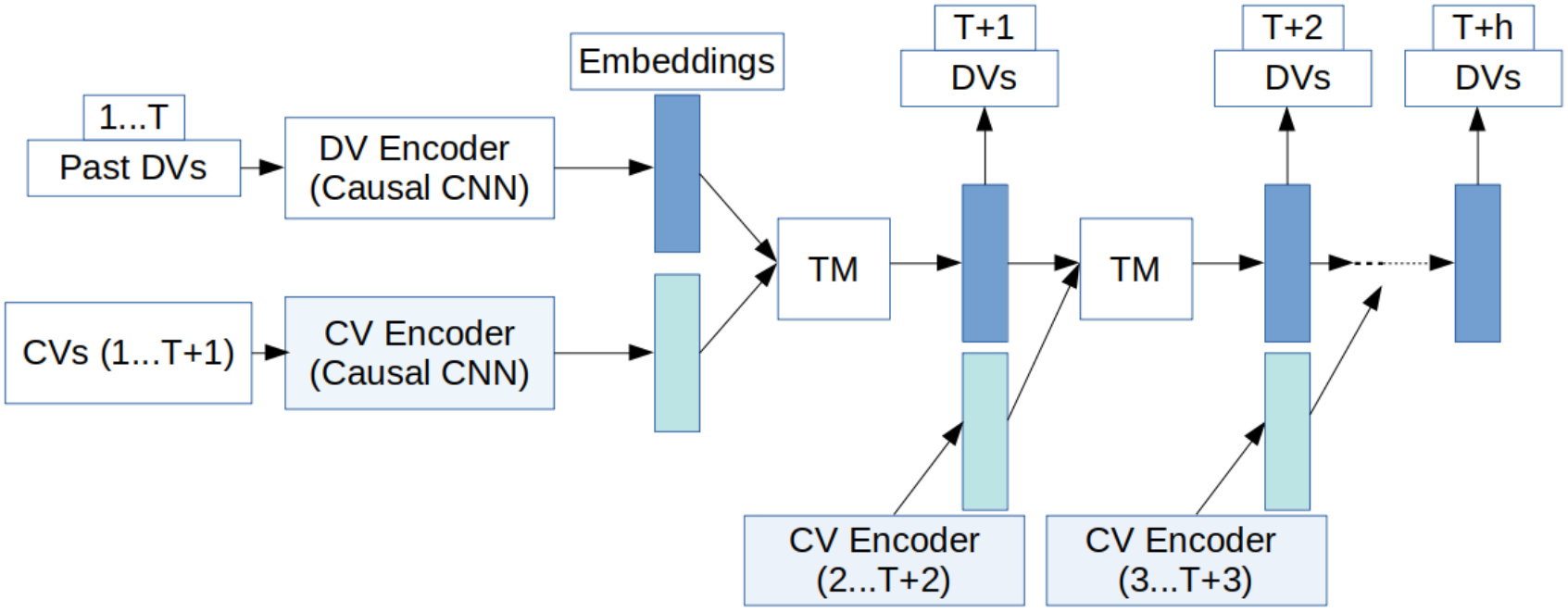


Figure: Vanilla Approach

Treating Each Control Variable Separately

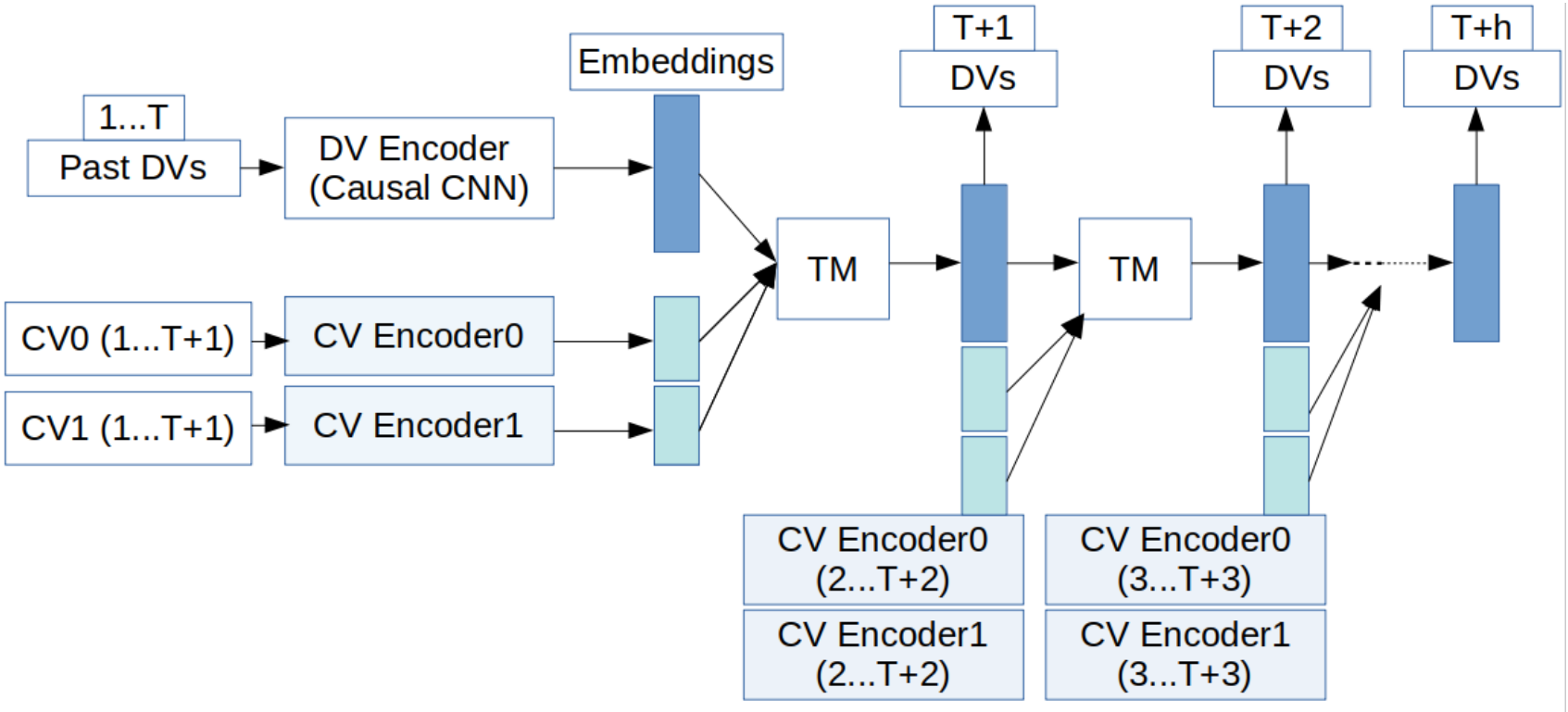


Figure: Separate Encoders for each control variable

Our Approach

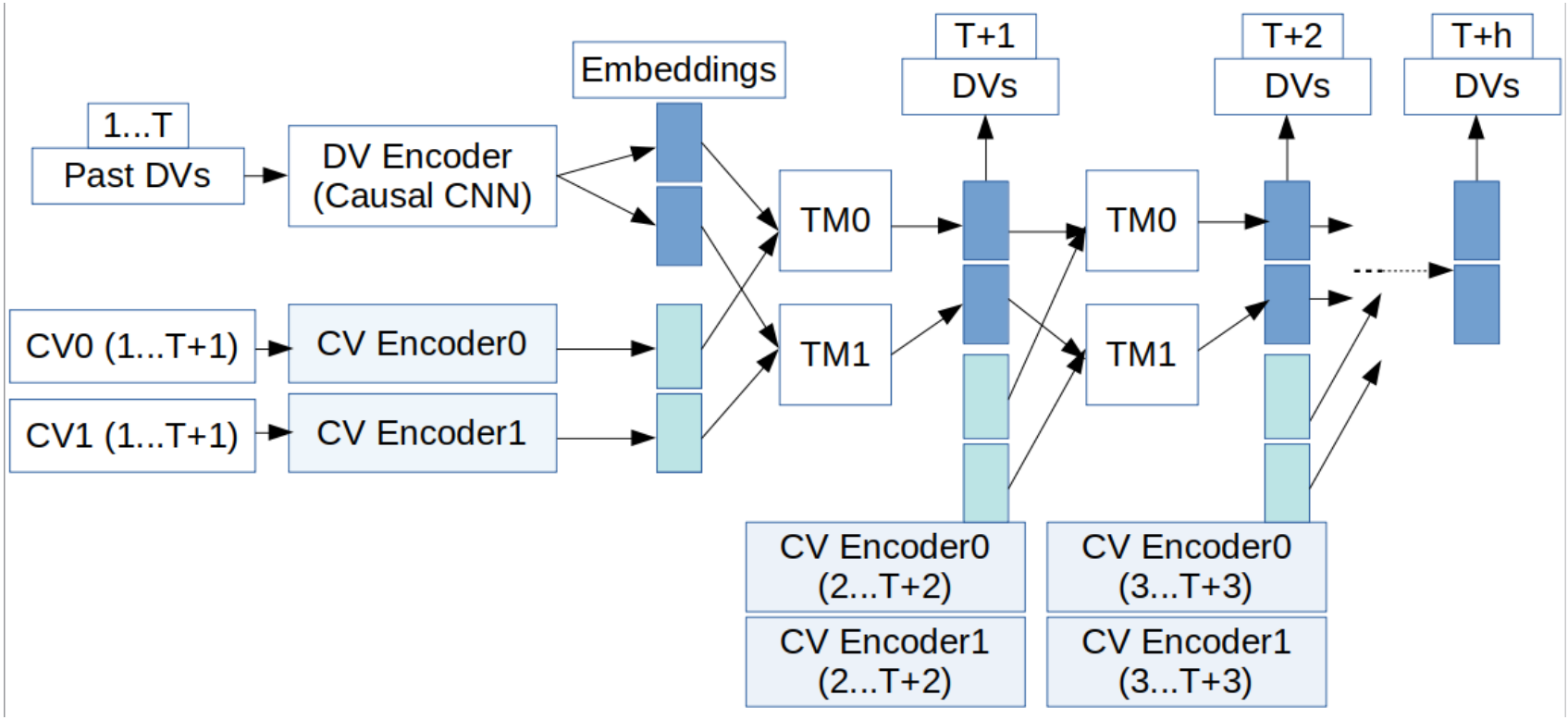


Figure: Our Approach with i. separate encoder for each control variable, ii. separate transition model for each control variable

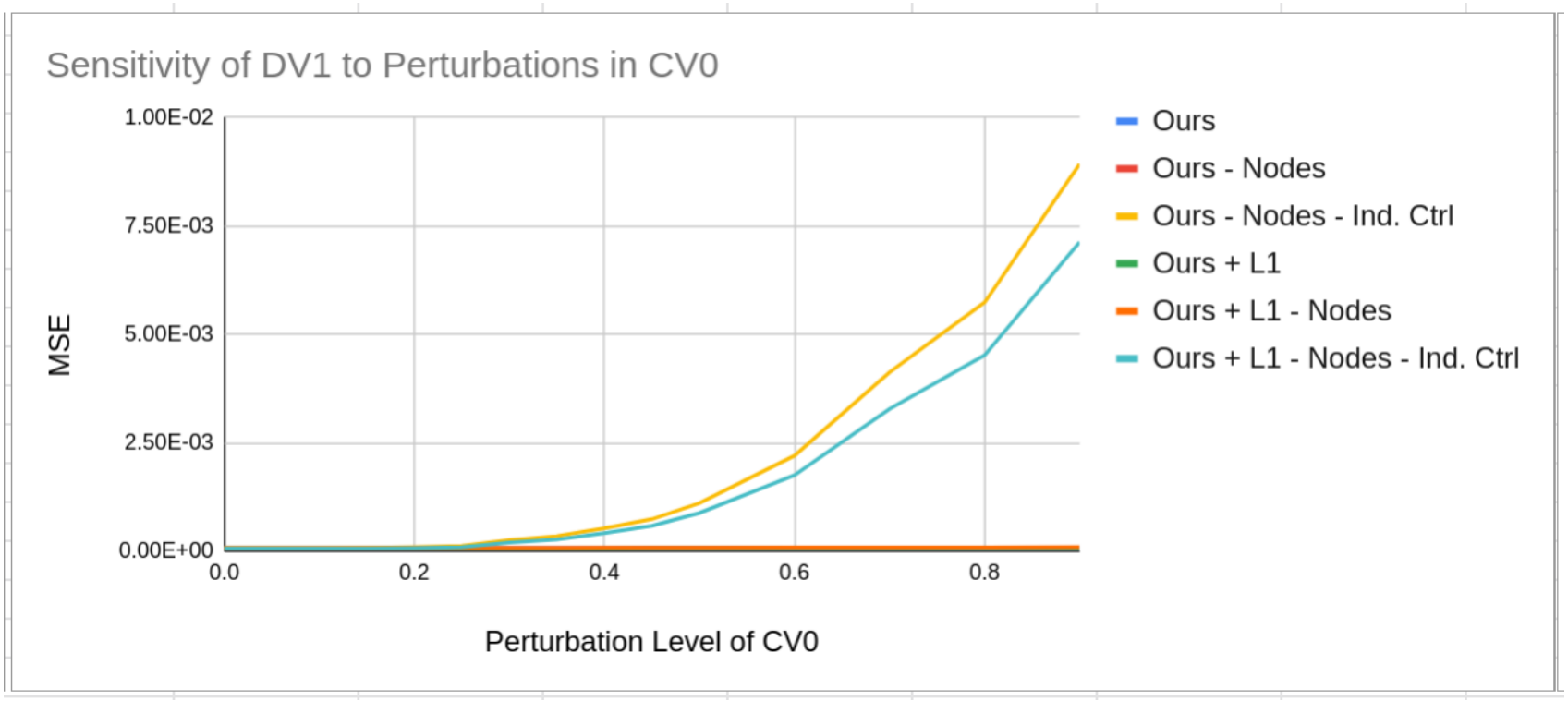
Results (5-step ahead forecasting)

Description	Method	Valid. MSE	Test MSE
w/ nodes and w/ separate channel per control	**Ours**	6.560E-05	6.410E-05
	Ours + L1	6.420E-05	6.250E-05
w/o nodes, w/ separate channel per control	**Ours - Nodes**	6.909E-05	6.733E-05
	Ours + L1 - Nodes	7.046E-05	6.908E-05
Standard Approaches w/o structural biases	**Ours - Nodes - Ind. Ctrl.**	6.585E-05	6.449E-05
	Ours + L1 - Nodes - Ind. Ctrl.	6.390E-05	6.188E-05

Key Observation: All approaches have similar results on the task they are trained for. However, our method with structural biases learns the correct dependencies and correlations almost perfectly, while other methods capture spurious correlations in the data as shown by sensitivity analysis.

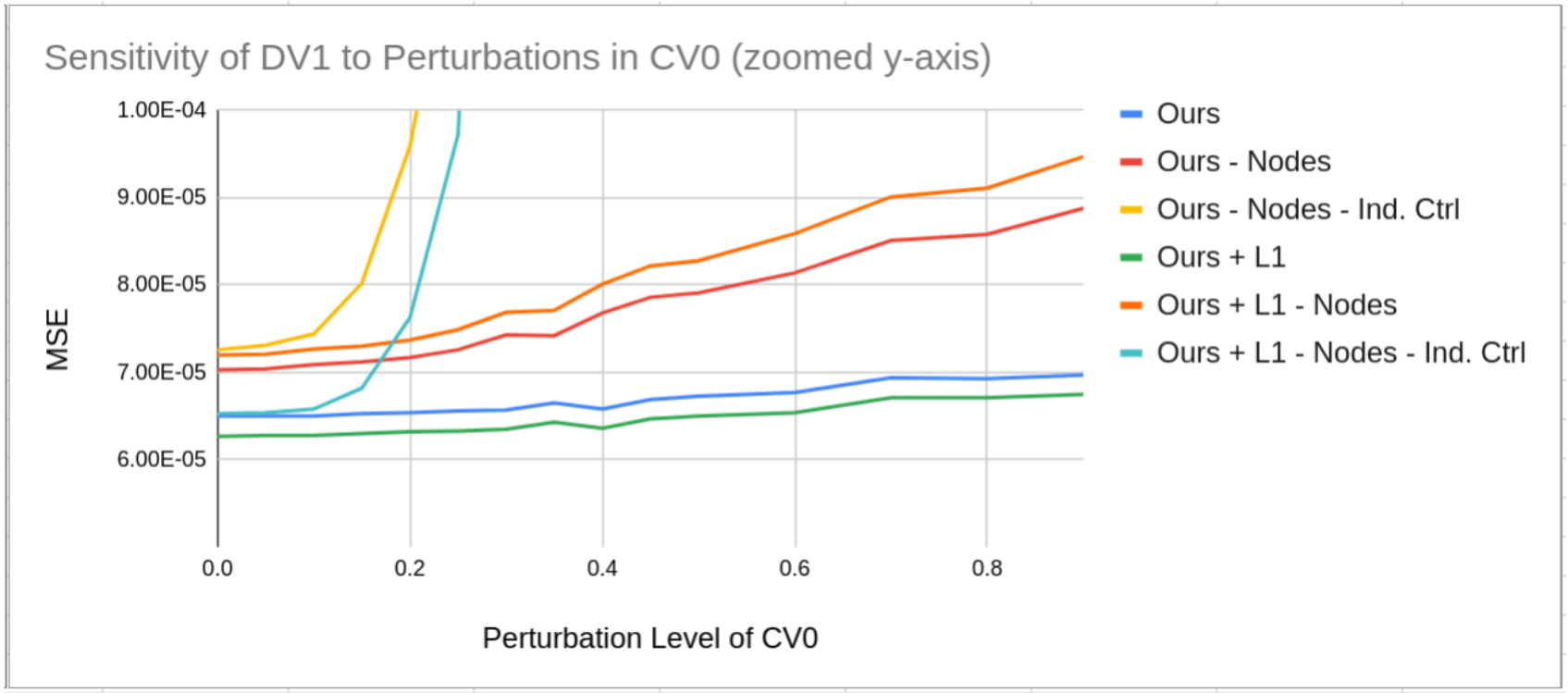
L1: LASSO regularizer on final layer.

Sensitivity Analysis (CV0 on DV1)



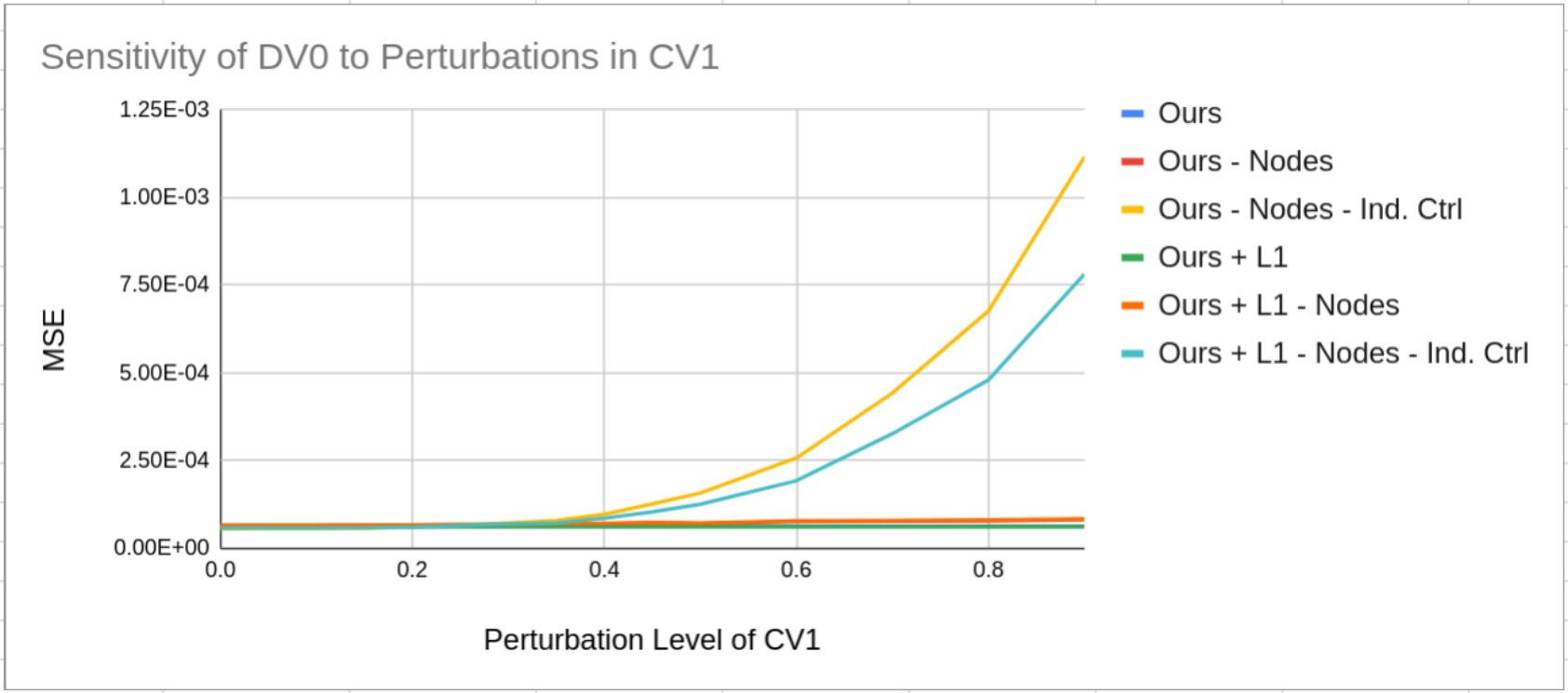
Observation: Vanilla approach learns wrong sensitivities (due to spurious correlations).

Sensitivity Analysis (CV0 on DV1)



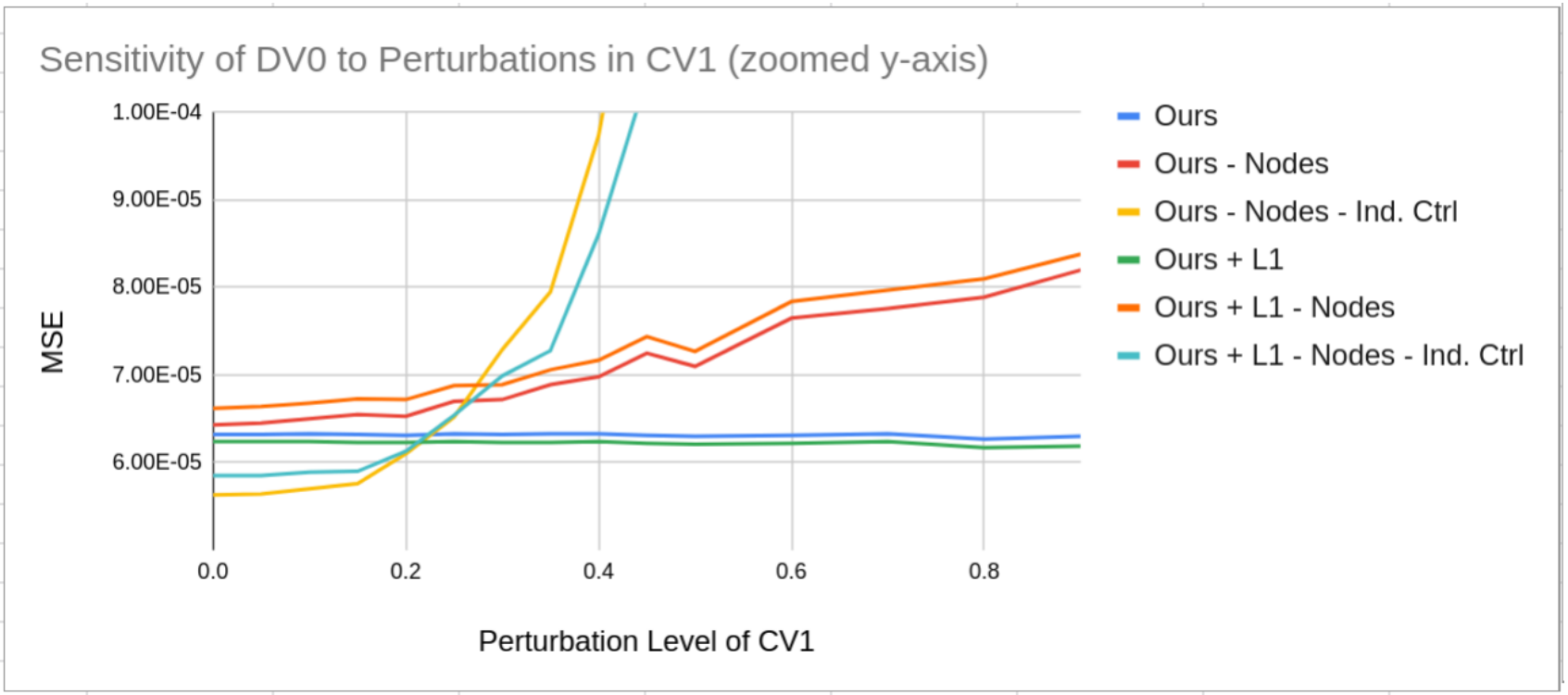
Observation: Having nodes in the latent space further improves upon separate control case.

Sensitivity Analysis (CV1 on DV0)



Observation: Vanilla approach learns wrong sensitivities.

Sensitivity Analysis (CV1 on DV0)



Observation: Having nodes in the latent space further improves upon separate control case.

Updates 22-Jul-2020

Overview

- Explore inductive biases in the neural network design to enable better disentanglement for multivariate time series data
- Key ideas:
  - Have a separate encoder for each variable already known to be independent (e.g. control variables)
  - Have a separate node (transition model) in the latent space for each control variable
  - Hierarchical latent space with sparse message passing

Experimental Setup

Generated time series with two control variables (CVs) and two dependent variables (DVs) as per the following equation:

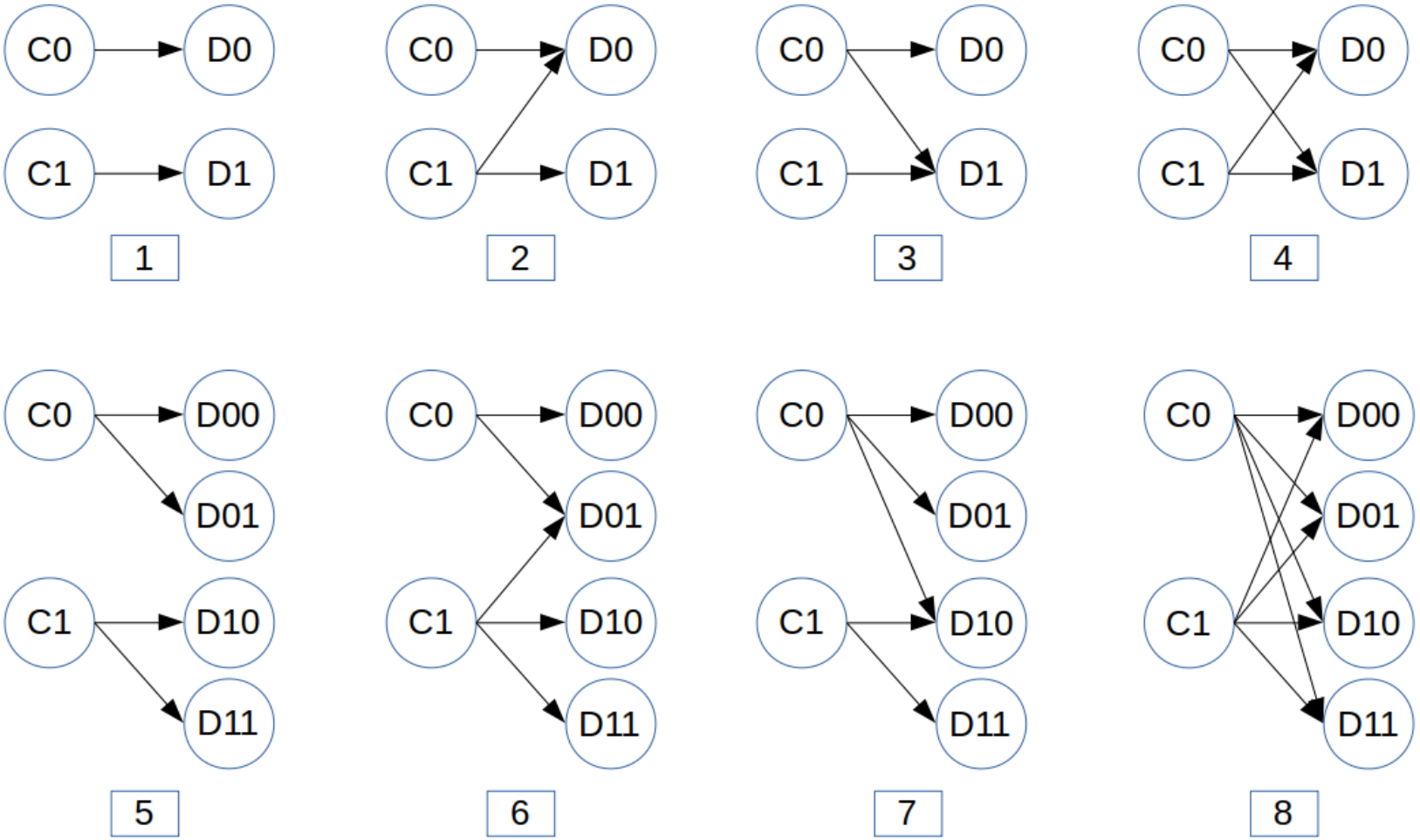
$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-1}(\sum_{j=t-10}^{t-1} y_j) + \alpha_3 u_t u_{t-10} + \alpha_4 \quad (17)$$

CV: u_t is sampled from $\mathcal{U}(0, 0.5)$, DV: y_t.

- Input: 100 length time series for DVs, 100+5 for CVs, Output: 5-step ahead forecasts, MSE loss.
- Architecture based on 1D CNNs (more details in next slides).
- **Hyperparameters**: learning rate:0.001, ADAM optimizer, L1 regularizer weight: 0.001, batch size:128, training instances: 8k, test instances: 2k.

Possible Scenarios

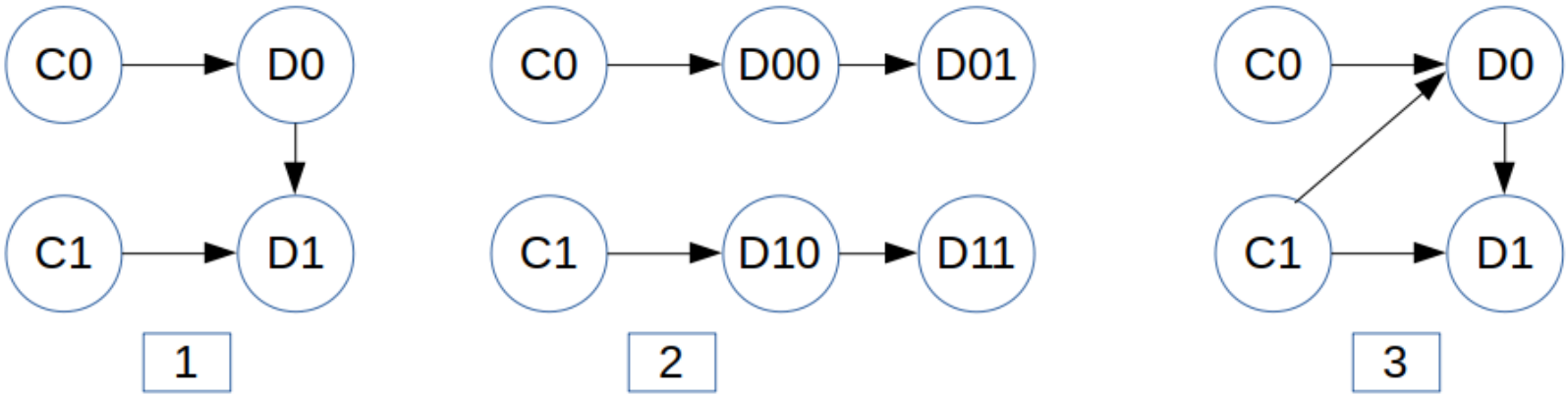
C: control variable, D: dependent variable



Experimented with Scenarios 1-4 so far.

Other Possible Scenarios

C: control variable, D: dependent variable



Note: While defining these scenarios, also need to consider the notion of time, e.g. in [3] here, first C1 affects D0, then at next time step D0 affects D1, etc.

Sample Data

CV: control variable, DV: dependent variable

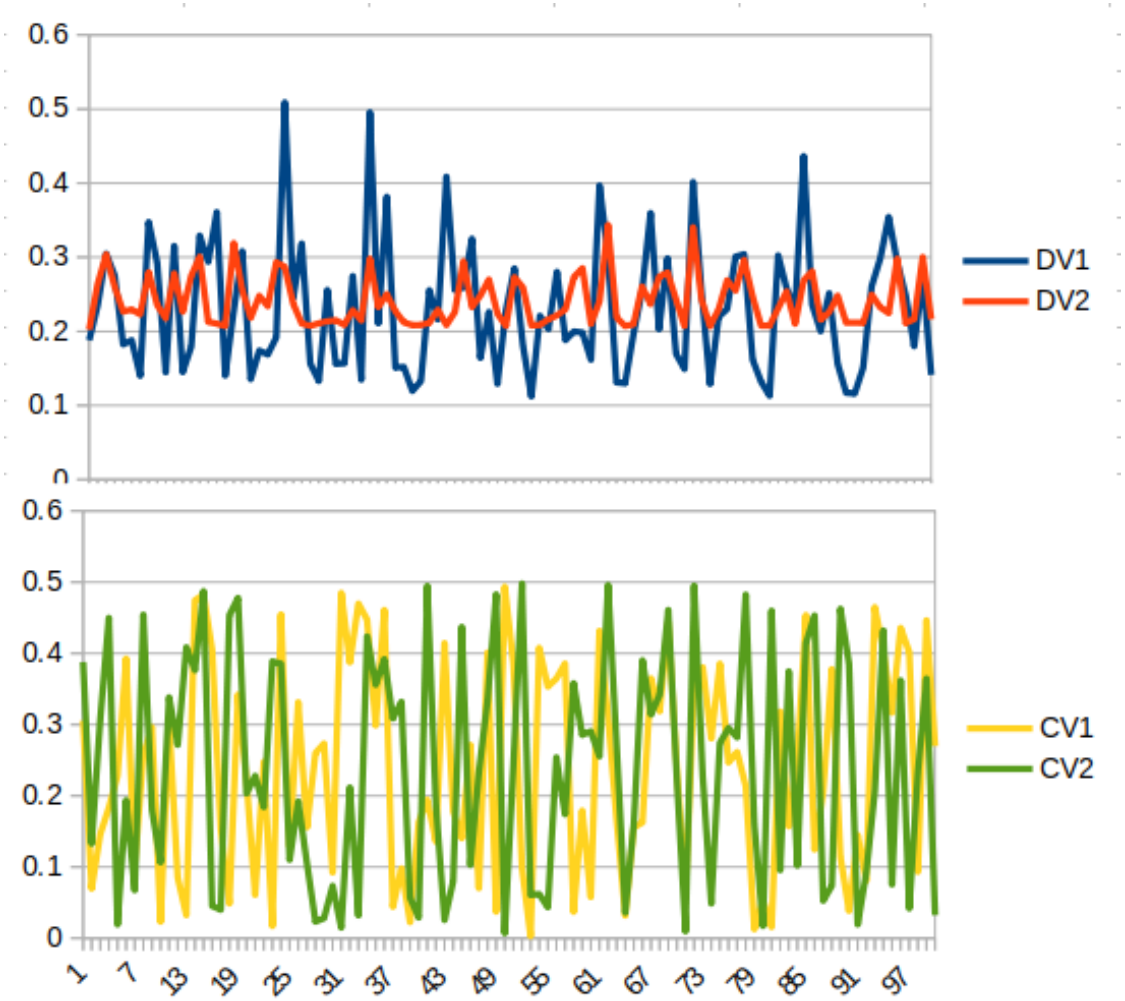


Figure: Scenario-2: CV1 affects DV1, CV2 affects DV1, DV2.

Vanilla Approach for Forecasting

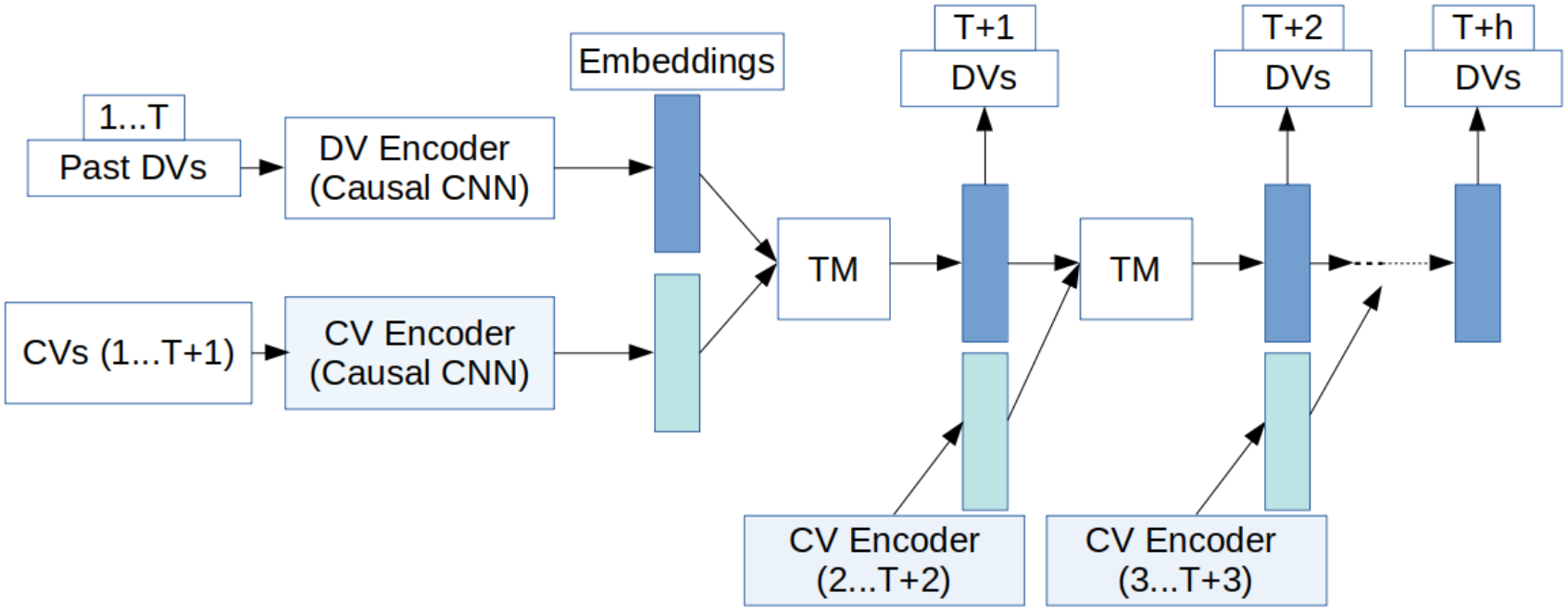


Figure: Vanilla Approach

Both CVs are passed through a common 2-input-channels 1D-CNN. TM: Transition Model

Treating Each Control Variable Separately

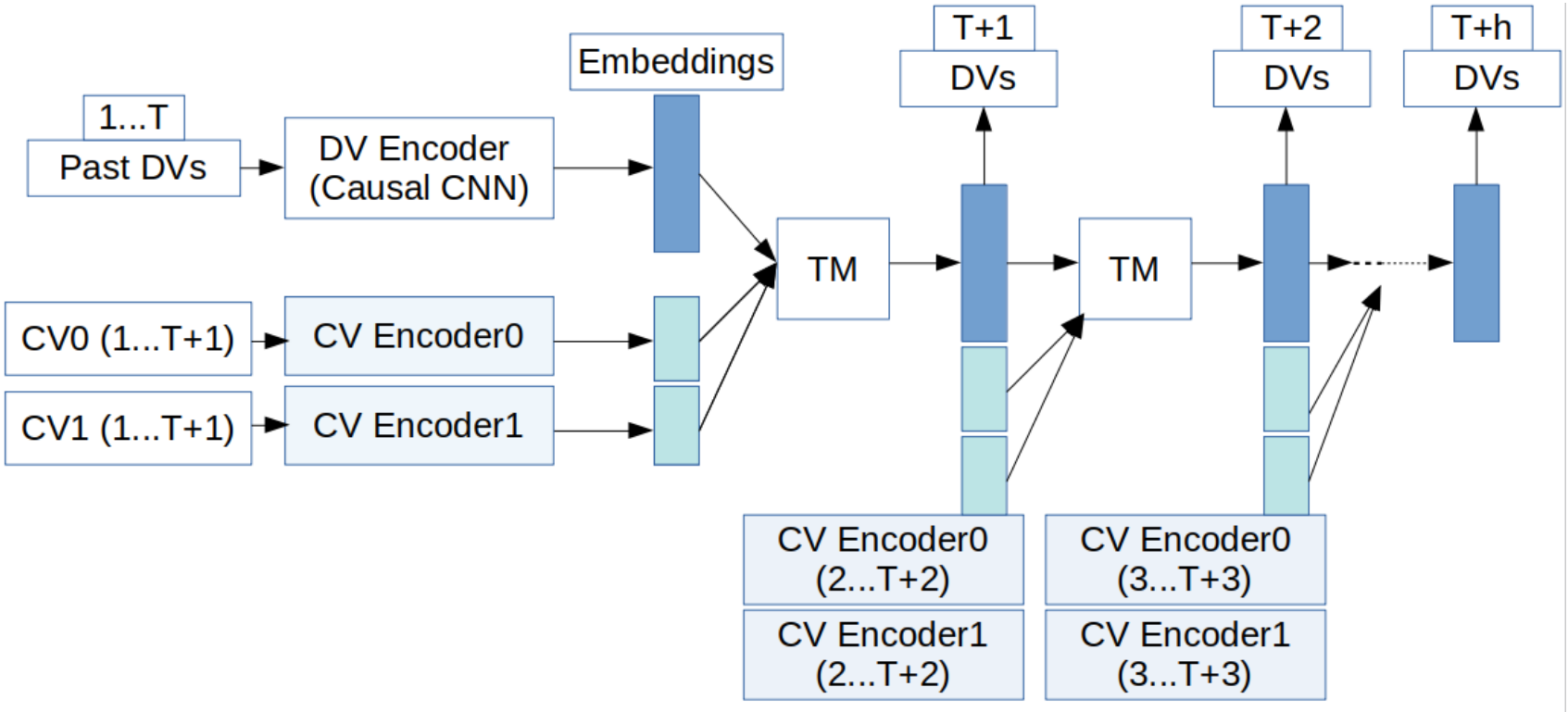


Figure: Separate Encoders for each control variable

Each CV has its own encoder.

Our Approach - 1 (No communication between TMs)

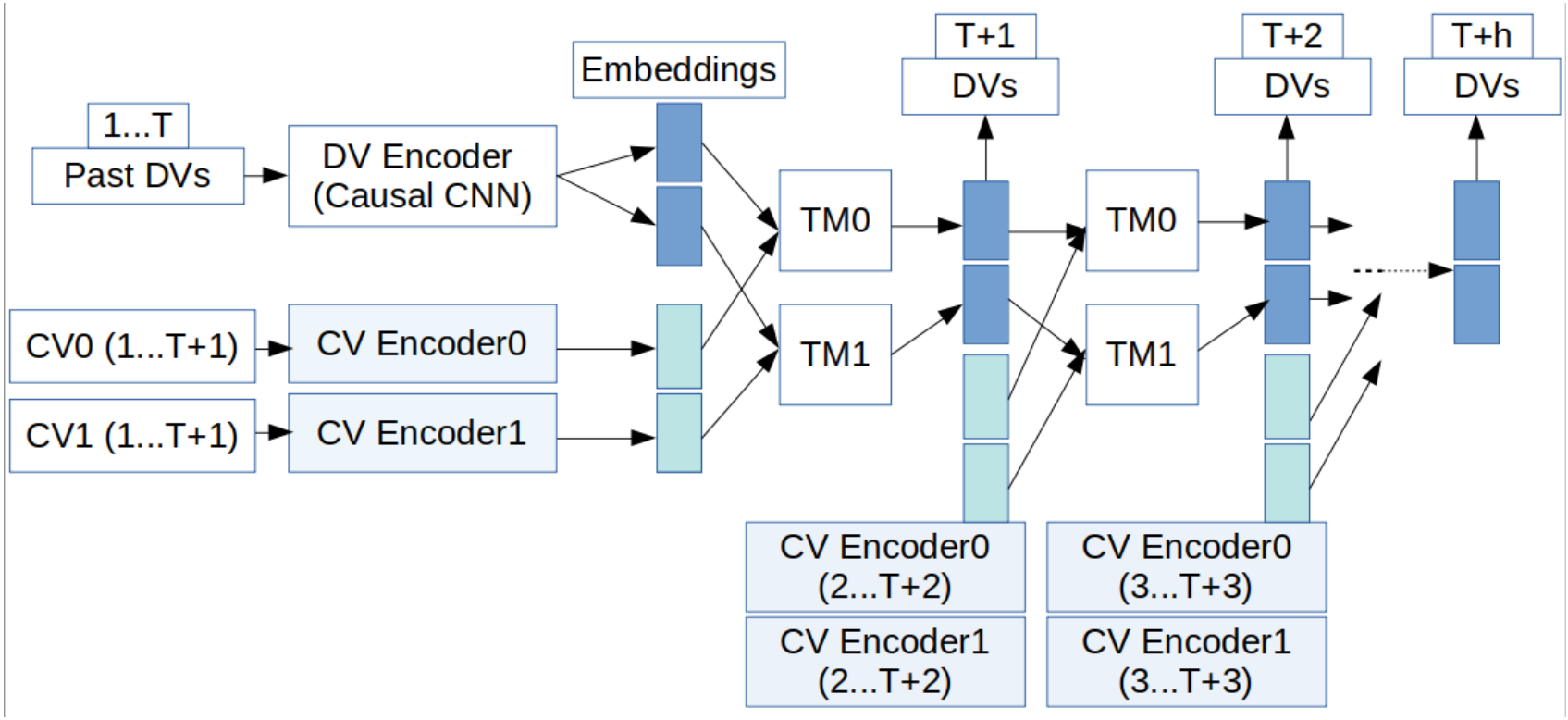


Figure: Our Approach with i. separate encoder for each control variable, ii. separate transition model for each control variable

Our Approach - Hierarchical Latent Space

sparse communication between TMs, hard sparsity on output layer

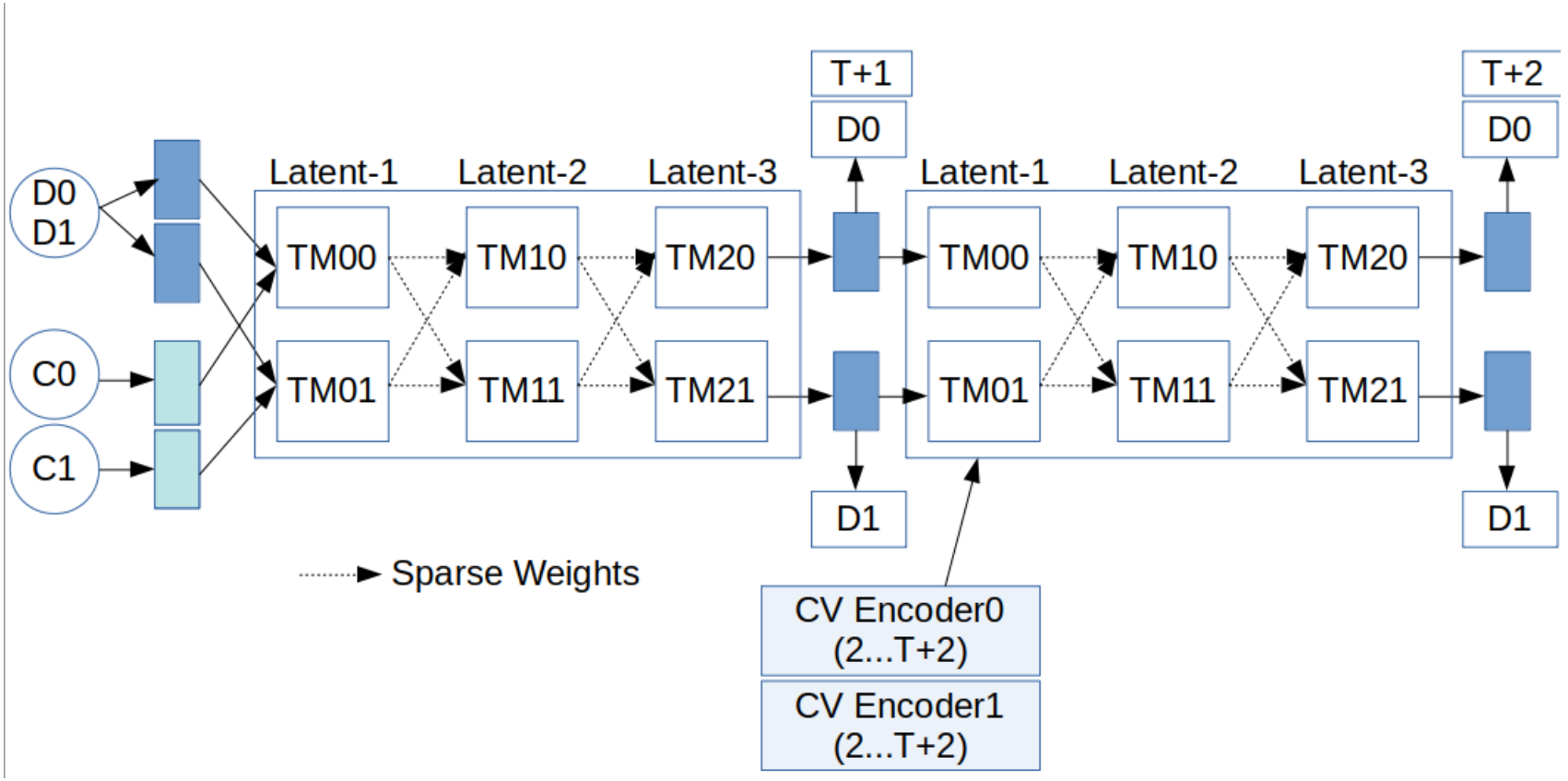
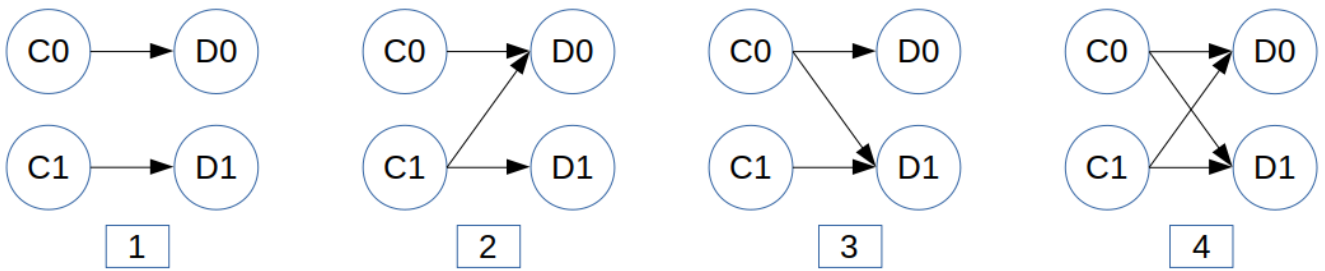


Figure: Our Approach with i. separate encoder for each control variable, ii. multi-node transition model, iii. sparse access to embeddings across multi-layer transition models.

Summary Results (5-step ahead forecasting)

				Without DV-lag (alpha_2=0)							
				Scenario-1		Scenario-2		Scenario-3		Scenario-4	
Approach	Sep. Control	Nodes	Hierarchical	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE
A1	T	F	F	4.49E-05	4.55E-05	5.09E-05	5.20E-05	5.40E-05	5.47E-05	6.05E-05	6.02E-05
A2	F	F	F	6.01E-05	6.26E-05	4.80E-05	5.03E-05	4.87E-05	4.85E-05	4.24E-05	4.40E-05
A3	T	T	F	5.00E-05	5.15E-05	4.89E-05	5.01E-05	4.70E-05	4.75E-05	4.11E-05	4.22E-05

				With DV-lag							
				Scenario-1		Scenario-2		Scenario-3		Scenario-4	
Approach	Sep. Control	Nodes	Hierarchical	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE
A1	T	F	F	6.39E-05	6.43E-05	7.37E-05	7.56E-05	8.02E-05	8.20E-05	6.71E-05	6.91E-05
A2	F	F	F	1.00E-04	1.02E-04	8.34E-05	8.55E-05	7.39E-05	7.51E-05	9.63E-05	9.56E-05
A3	T	T	F	7.03E-05	7.02E-05	1.25E-04	1.25E-04	6.37E-05	6.59E-05	1.22E-04	1.23E-04
A4	T	T	T	8.51E-05	8.56E-05	8.18E-05	8.49E-05	6.75E-05	6.69E-05	9.35E-05	9.56E-05



$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-1}\left(\sum_{j=t-10}^{t-1} y_j\right) + \alpha_3 u_t u_{t-10} + \alpha_4 \qquad (18)$$

Summary Results (5-step ahead forecasting)

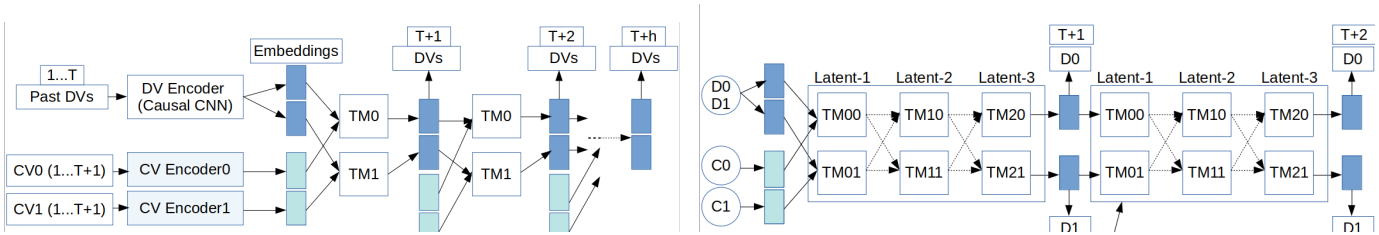


Figure: Left: Approach A3, Right: Approach A4

				Without DV-lag (alpha_2=0)							
				Scenario-1		Scenario-2		Scenario-3		Scenario-4	
Approach	Sep. Control	Nodes	Hierarchical	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE
A1	T	F	F	4.49E-05	4.55E-05	5.09E-05	5.20E-05	5.40E-05	5.47E-05	6.05E-05	6.02E-05
A2	F	F	F	6.01E-05	6.26E-05	4.80E-05	5.03E-05	4.87E-05	4.85E-05	4.24E-05	4.40E-05
A3	T	T	F	5.00E-05	5.15E-05	4.89E-05	5.01E-05	4.70E-05	4.75E-05	4.11E-05	4.22E-05

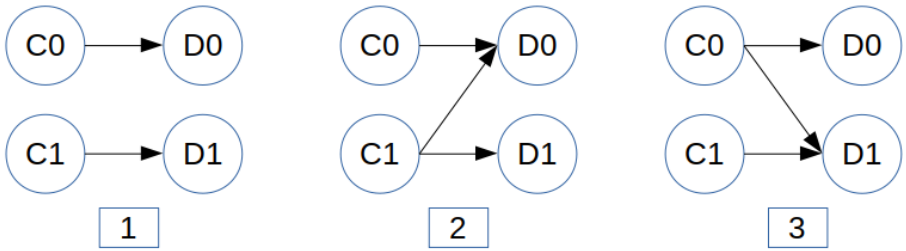
				With DV-lag							
				Scenario-1		Scenario-2		Scenario-3		Scenario-4	
Approach	Sep. Control	Nodes	Hierarchical	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE	Valid. MSE	Test MSE
A1	T	F	F	6.39E-05	6.43E-05	7.37E-05	7.56E-05	8.02E-05	8.20E-05	6.71E-05	6.91E-05
A2	F	F	F	1.00E-04	1.02E-04	8.34E-05	8.55E-05	7.39E-05	7.51E-05	9.63E-05	9.56E-05
A3	T	T	F	7.03E-05	7.02E-05	1.25E-04	1.25E-04	6.37E-05	6.59E-05	1.22E-04	1.23E-04
A4	T	T	T	8.51E-05	8.56E-05	8.18E-05	8.49E-05	6.75E-05	6.69E-05	9.35E-05	9.56E-05

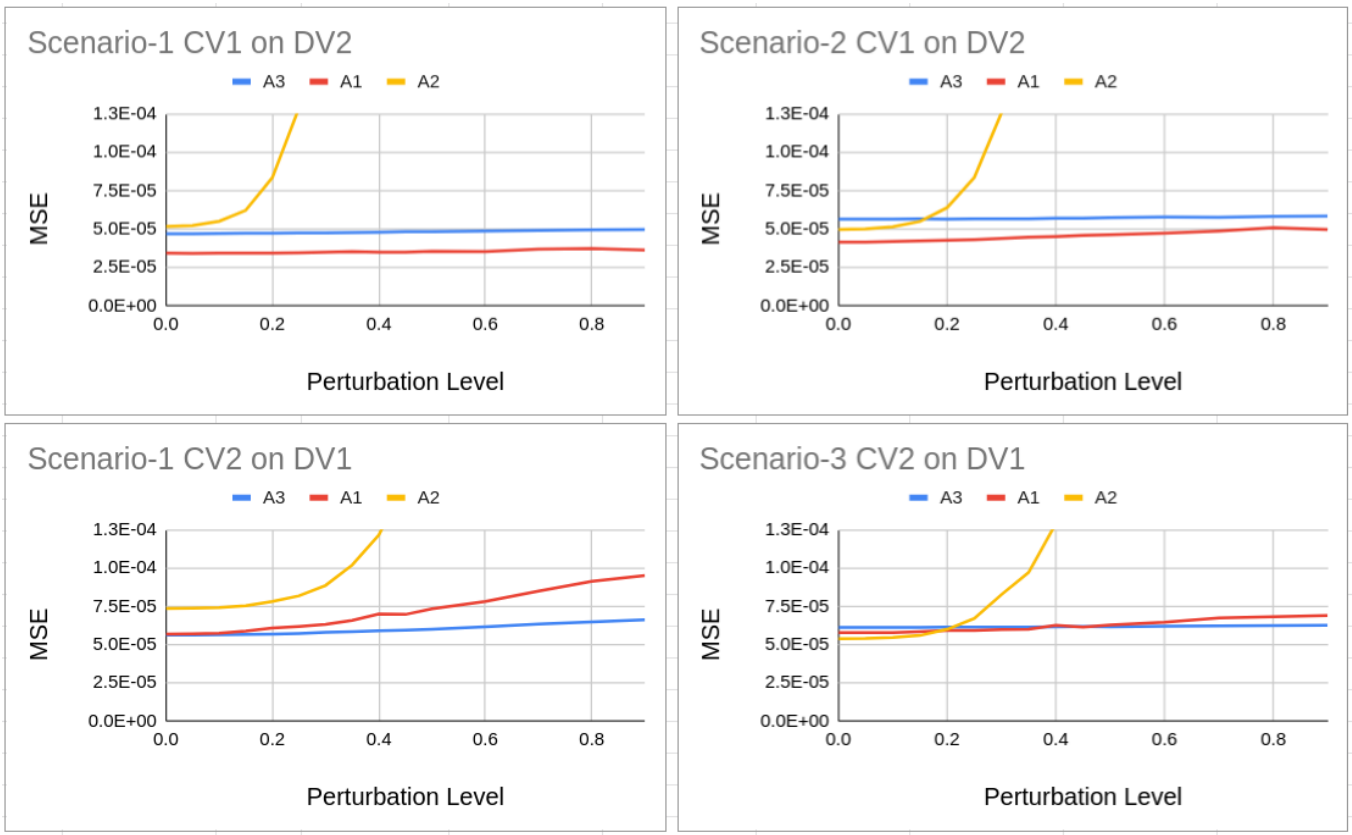
Observation: Approach A3 struggles on Scenarios 2 and 4, while A4 is close to A1 and A2 for the DV-lag case.

Sensitivity to perturbations (w/o DV-lag case)

A3 shows best “disentanglement”.

Approach	Sep. Control	Nodes	Hierarchical
A1	T	F	F
A2	F	F	F
A3	T	T	F
A4	T	T	T

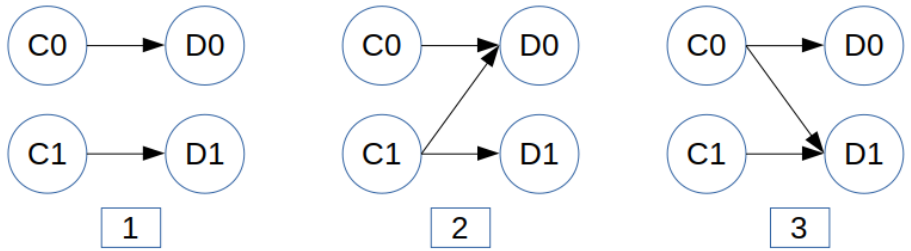


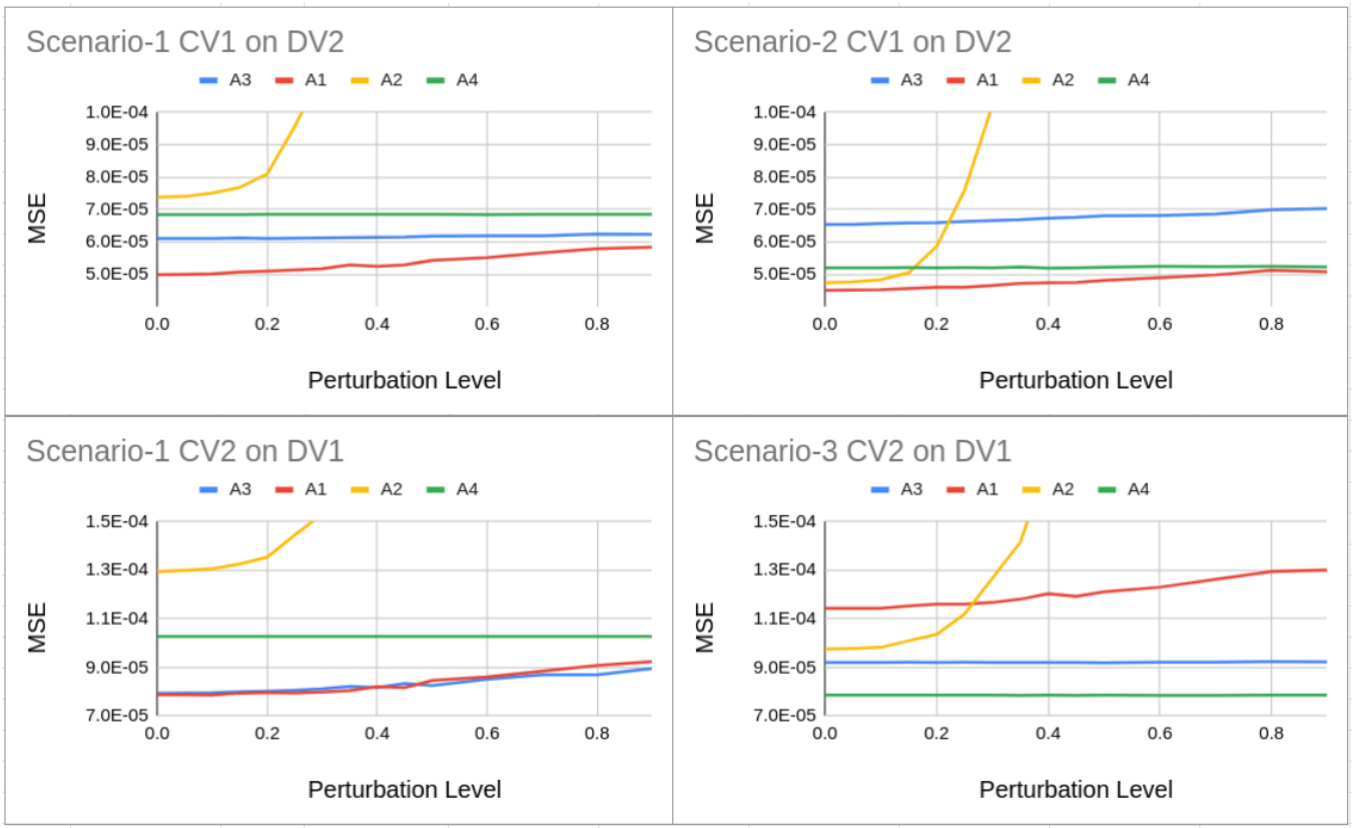


Sensitivity to perturbations (DV-lag case)

A4 shows best “disentanglement”.

Approach	Sep. Control	Nodes	Hierarchical
A1	T	F	F
A2	F	F	F
A3	T	T	F
A4	T	T	T





Hierarchical Latent Space - Soft vs Hard Sparsity

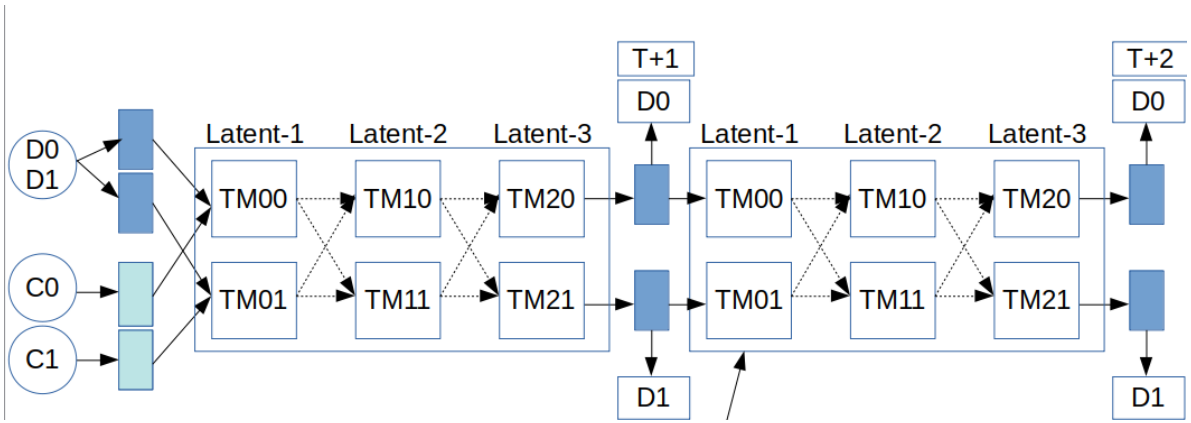


Figure: Hard Decoder

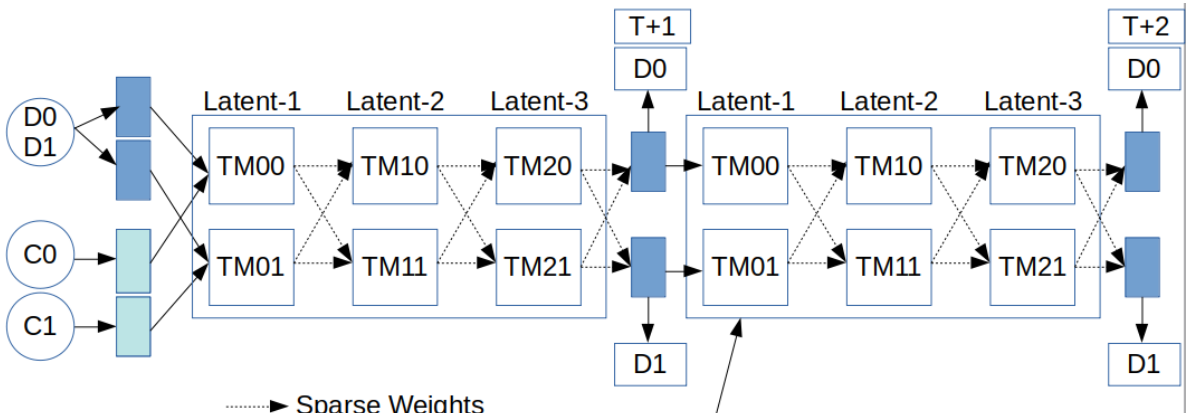
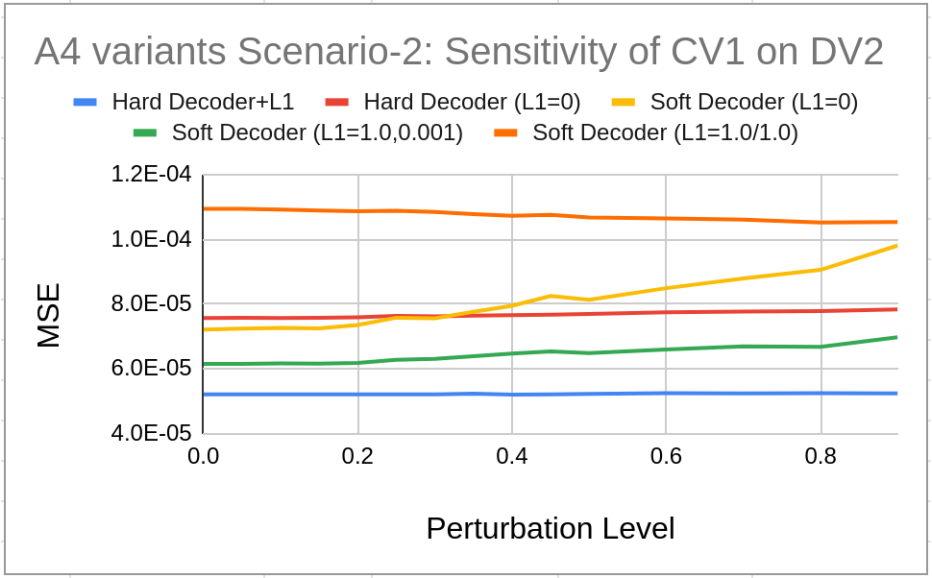


Figure: Soft Decoder

Hierarchical Latent Space (A4) - Soft vs Hard Sparsity



A4 Variant	Test MSE
Hard Decoder+L1	8.49E-05
Hard Decoder (L1=0)	1.16E-04
Soft Decoder (L1=0)	1.46E-04
Soft Decoder (L1=1.0,0.001)	1.04E-04
Soft Decoder (L1=1.0/1.0)	1.47E-04

Observations:

- Sparsity in the latent space improves results
- Hard sparsity at the output improves results

Next Steps

- Group LASSO in latent space
- Detailed sensitivity analysis of the latent space to see what edges are being formed
- Real datasets: GHL, SWaT, Sarcos, NetSim, DREAM-3.

Recent Relevant Literature

Discovering Symbolic Models from Deep Learning with Inductive Biases

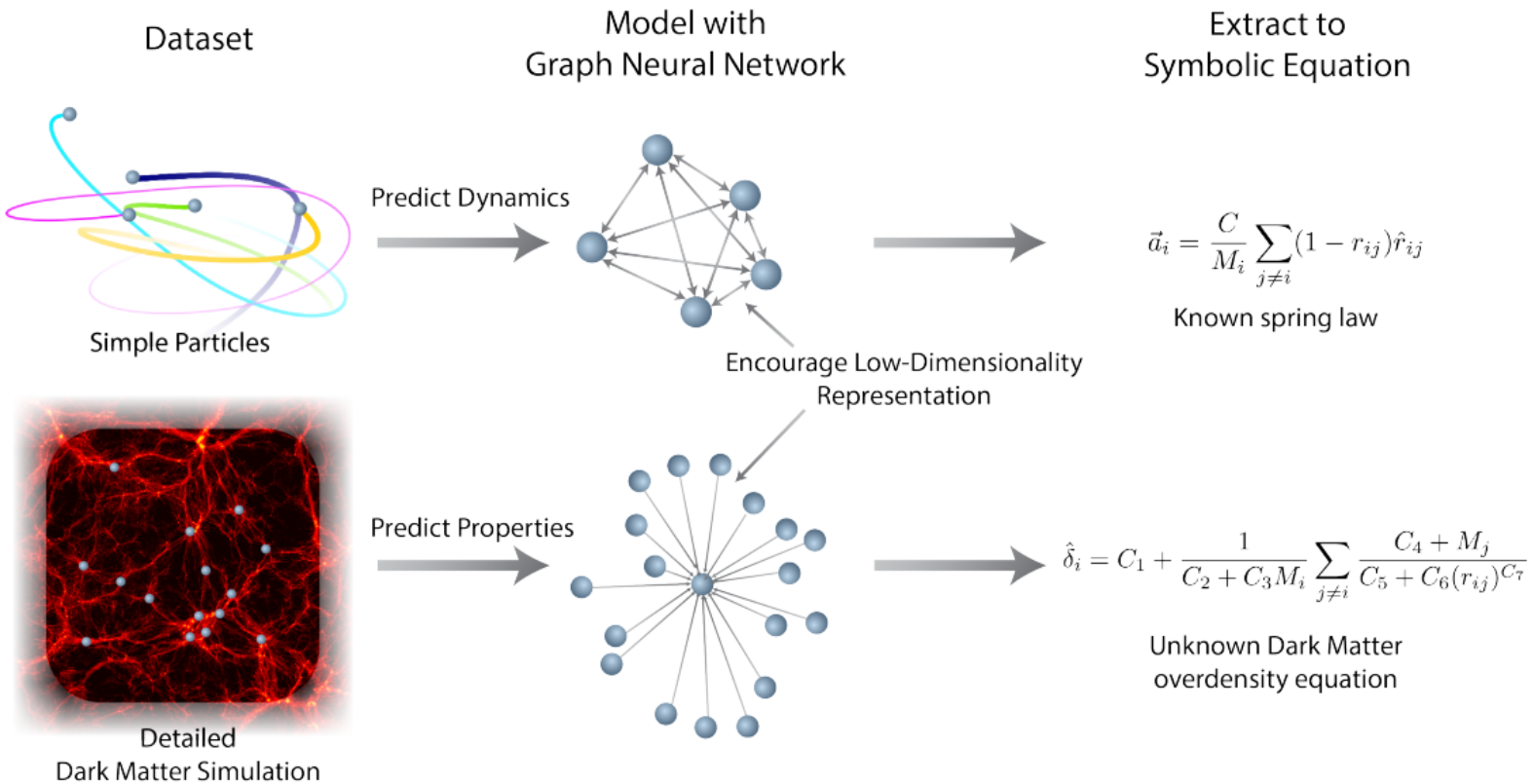


Figure 1: A cartoon depicting how we extract physical equations from a dataset.

Discovering Symbolic Models from Deep Learning with Inductive Biases

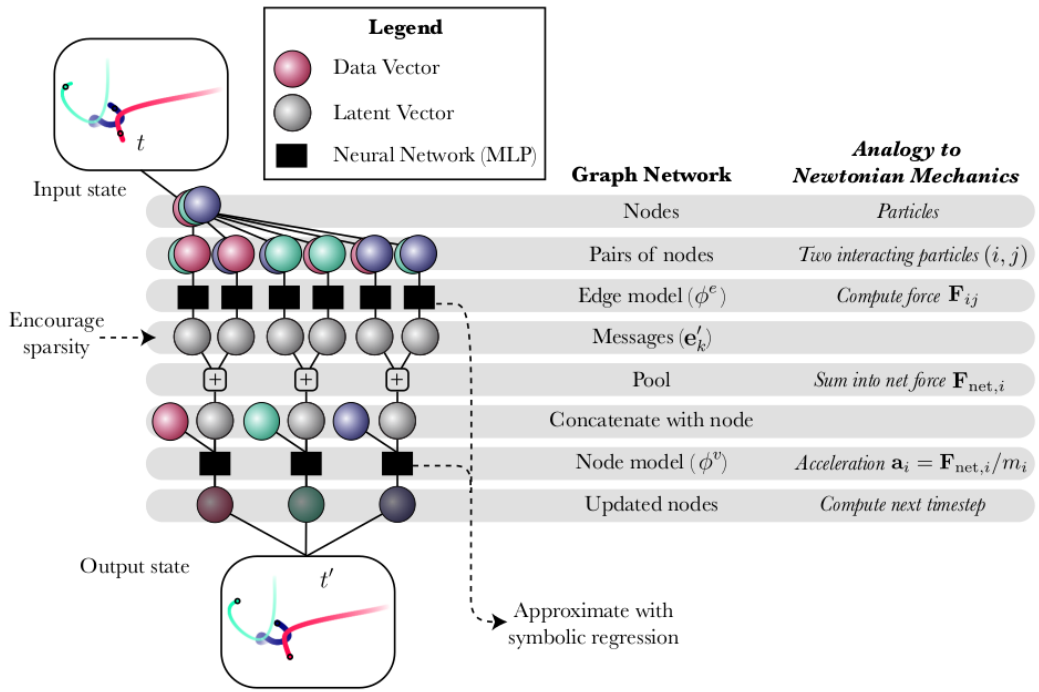


Figure 2: An illustration of the internal structure of the graph neural network we use in some of our experiments. Note that the comparison to Newtonian mechanics is purely for explanatory purposes, but is not explicit. Differences include: the "forces" (messages) are often high dimensional, the nodes need not be physical particles, ϕ^e and ϕ^v are arbitrary learned functions, and the output need not be an updated state. However, the rough equivalency between this architecture and physical frameworks allows us to interpret learned formulas in terms of existing physics.

Granger Causality

2.1 Granger Causality

Granger causality [18] is one of the most commonly used approaches to infer causal relations from observational time-series data. Its central assumption is that causes precede their effects: if the prediction of the future of time-series Y can be improved by knowing past elements of time-series X, then X "Granger causes" Y. Originally, Granger causality was defined for linear relations; we follow the more recent definition of Tank et al. [49] for non-linear Granger causality:

Definition 2.1. *Non-Linear Granger Causality*: Given N stationary time-series $\boldsymbol{x} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_N\}$ across time-steps $t = \{1, ..., T\}$ and a non-linear autoregressive function g_j, such that

$$\boldsymbol{x}_j^{t+1} = g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_N^{\leq t}) + \boldsymbol{\varepsilon}_j^t \quad , \tag{1}$$

where $\boldsymbol{x}_j^{\leq t} = (..., \boldsymbol{x}_j^{t-1}, \boldsymbol{x}_j^t)$ denotes the present and past of series j and $\boldsymbol{\varepsilon}_j^t$ represents independent noise. In this setup, time-series i Granger causes j, if g_j is not invariant to $\boldsymbol{x}_i^{\leq t}$, i.e. if $\exists \boldsymbol{x}_i'^{\leq t} \neq \boldsymbol{x}_i^{\leq t} : \; g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_i'^{\leq t}, ..., \boldsymbol{x}_N^{\leq t}) \neq g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_i^{\leq t}, ...\boldsymbol{x}_N^{\leq t})$.

Granger causal relations are equivalent to causal relations in the underlying directed acyclic graph (DAG) if all relevant variables are observed and no instantaneous[2] connections exist [40].

Learning to Infer Causal Graphs from Time-Series Data (Max Welling, June 2020, Under Review NeurIPS)

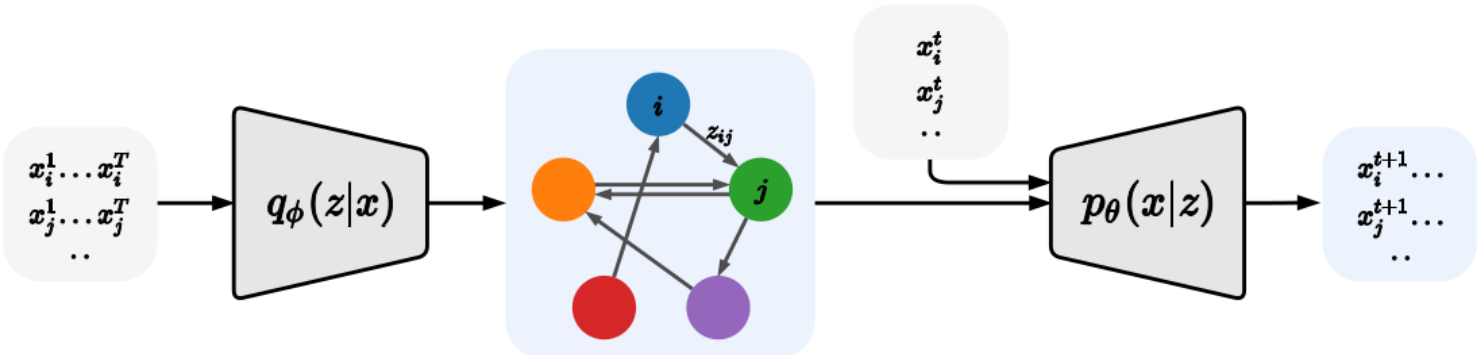


Figure 2: ACD-NRI: A Probabilistic Approach to Amortized Causal Discovery. An amortized encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ predicts the causal relations between the input time-series $\boldsymbol{x}$. A decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ models the dynamics of the future trajectories $\boldsymbol{x}^{t+1}$ given their current values $\boldsymbol{x}^t$ and the predicted relations $\boldsymbol{z}$. This separation between causal relation prediction and modeling lets us train the model across samples with different underlying causal graphs but shared dynamics.

Recent Relevant Literature

Economic Statistical Recurrent Unit for Inferring Non-linear Granger Causality (ICLR20)

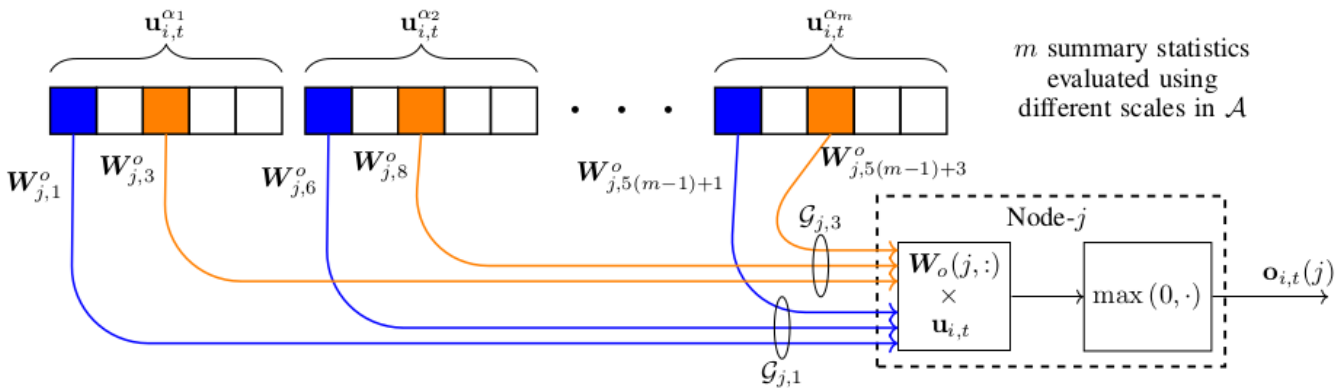
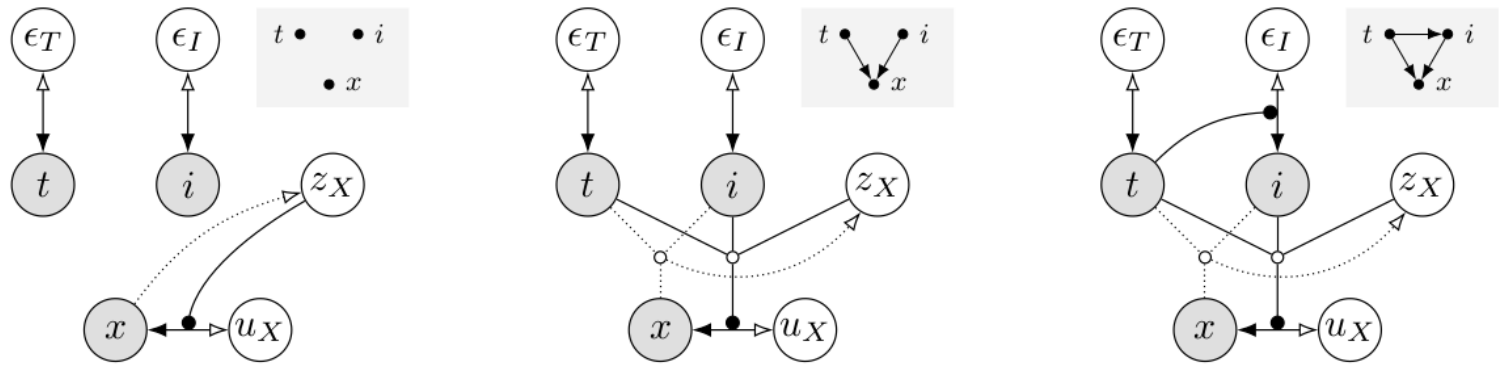


Figure 3: An illustration of the proposed group-wise mixing of the multi-timescale summary statistics $\mathbf{u}_{i,t}$ in the i^{th} SRU (with $d_\phi = 5$) towards generating the j^{th} predictive feature in $\mathbf{o}_{i,t}$. The weights corresponding to the same colored connections belong to the same group.

Note: Also evaluates on DREAM-3 dataset and an fMRI dataset

Recent Relevant Literature

Deep Structural Causal Models for Tractable Counterfactual Inference (Under Review, NeurIPS 20)



(a) Independent (b) Conditional (c) Full

Figure 2: Computational graphs of the structural causal models for the Morpho-MNIST example. The image is denoted by x, stroke thickness by t, and image intensity by i. The corresponding causal diagrams are displayed in the top-right corners.

Recent Relevant Literature

Deep Structural Causal Models for Tractable Counterfactual Inference - Example

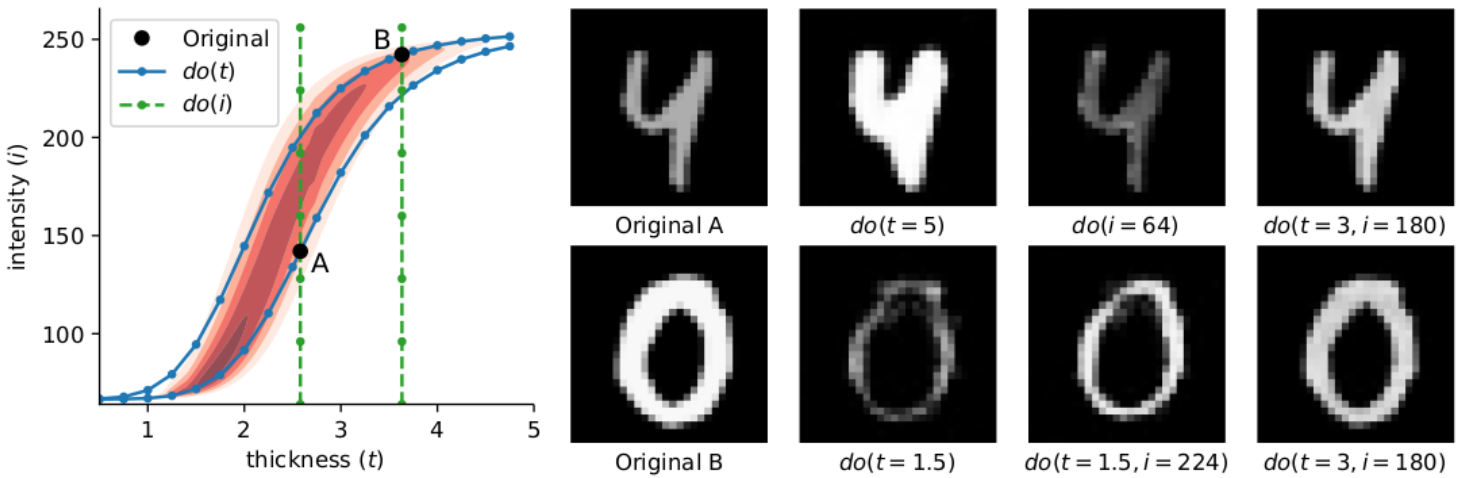


Figure 4: Counterfactuals generated by the full model. (left) Counterfactual 'trajectories' of two original samples, A and B, as their thickness and intensity are modified, overlaid on the learned joint density $p(t, i)$. (right) Original and counterfactual images corresponding to samples A and B.

Combinatorial Generalization Testing

$$y_t^i = \sum_j \alpha_1^j y_{t-1}^j + \alpha_2^j y_{t-1}^j \Big(\sum_{k=t-10}^{t-1} y_k^j \Big) + \alpha_3^j u_t^j u_{t-10}^j + \alpha_4^j, \quad (19)$$

where $i, j \in \{0, 1\}$. For now, we are considering $\alpha_1^j, \alpha_2^j, \alpha_4^j = 0$ for $j \neq i$ for all scenarios, while $\alpha_3^j \neq 0$ depending upon the scenario.

- At each time step, switch regime with probability $p_s = 0.2$, and choose a new regime r sampled uniformly from $\{1, 2, \ldots, n_1\}$ for training set and from $\{n_1 + 1, \ldots, n_1 + n_2\}$ for testing set, such that there are total of n_1 regimes in train and n_2 regimes in OOD test set.
- In any regime r, $u^1 = \beta_{r,1} u^0 + \beta_{r,2} (u^0)^2 + \beta_{r,3}$, where $\beta_{r,m} \sim \mathcal{U}(0, 0.5)$ for $m = 1, 2$.

Note: Also create IID test set from the n_1 training regime. Hypothesis is that while all M5-M7 and M9 do well on IID test set, M7 should struggle on OOD, M5,M6,M9 should generalize to OOD test set. Furthermore, M5,M6,M9 with group LASSO / L1 should generalize even better than vanilla M5,M6,M9.

Combinatorial Generalization Testing - II

- For a regime r for ith control variable CVi:

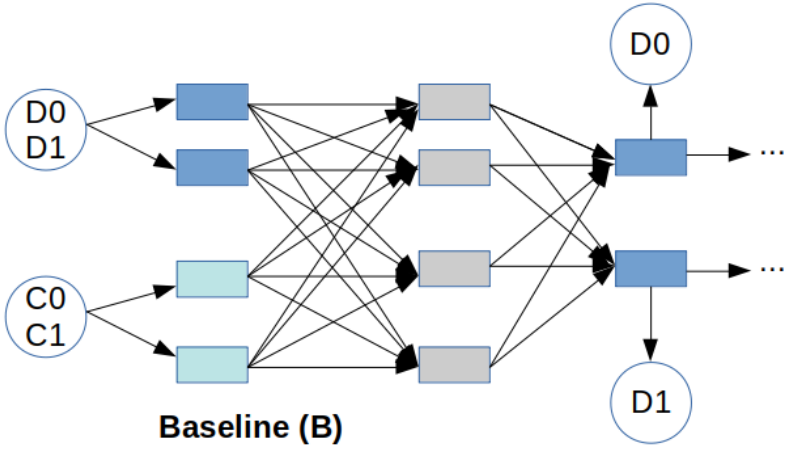
$$u_i = \beta_{i1} u'_i + \beta_{i2} \tag{20}$$

where $\beta_{i1} \sim \mathcal{U}(0.5, 1.0)$ and $\beta_{i2} \sim \mathcal{U}(-0.2, 0.2)$ define a regime for i-th CV. $u'_i \sim \mathcal{U}(0, 1.0)$ is sampled iid at each time step.

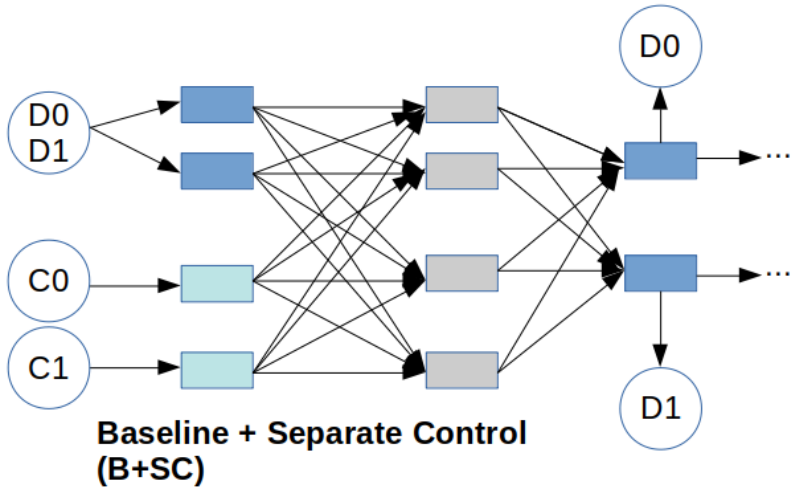
- Suppose there are n regimes for CV1 and CV2 each. Then, there are total n^2 possible regimes for the overall system. Let, first $n/2$ regimes of CV1 co-occur with first $n/2$ regimes of CV2, and same for second half. Then, total regimes seen during training is $n^2/2$ $(= 2 * (n/2) * (n/2))$. Remaining $n^2/2$ regimes are OOD.
- We can try with $n = 8$ such that there are 64 possible combinations, 32 are used for training (and iid testing), while the other 32 are used for OOD testing.

Updates 05-Aug-2020

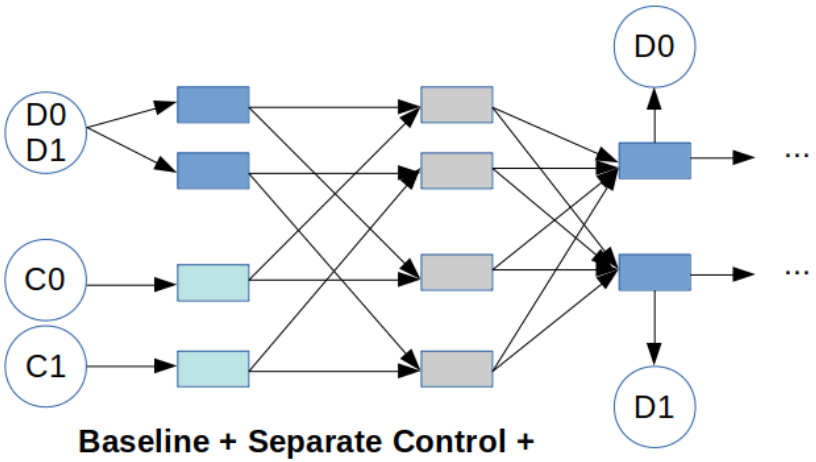
Architectures



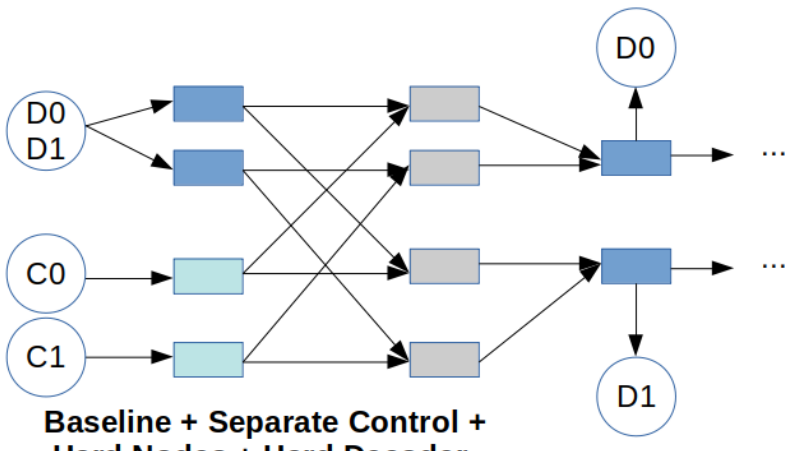
Baseline (B)



Baseline + Separate Control (B+SC)



Baseline + Separate Control + Hard Nodes (B+SC+HN)



Baseline + Separate Control + Hard Nodes + Hard Decoder (B+SC+HN+HD)

Experimental Setup

Generated time series with two control variables (CVs) and two dependent variables (DVs) as per the following equation:

$$y_t = \alpha_1 y_{t-1} + \alpha_2 y_{t-1}\left(\sum_{j=t-10}^{t-1} y_j\right) + \alpha_3 u_t u_{t-10} + \alpha_4 \tag{21}$$

CV: u_t is sampled from $\mathcal{U}(0, 0.5)$, DV: y_t.

- Input: 11 length time series for DVs, 11+5 for CVs, Output: 5-step ahead forecasts, MSE loss.
- Architecture based on 1D CNNs (more details in next slides).
- **Hyperparameters**: learning rate:0.001, ADAM optimizer, batch size:128, training instances: 8k, test instances: 2k.

Combinatorial Generalization Testing

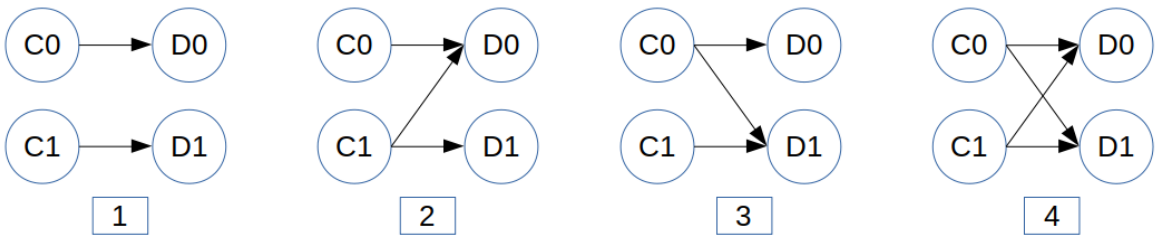
- For a regime r for ith control variable CVi:

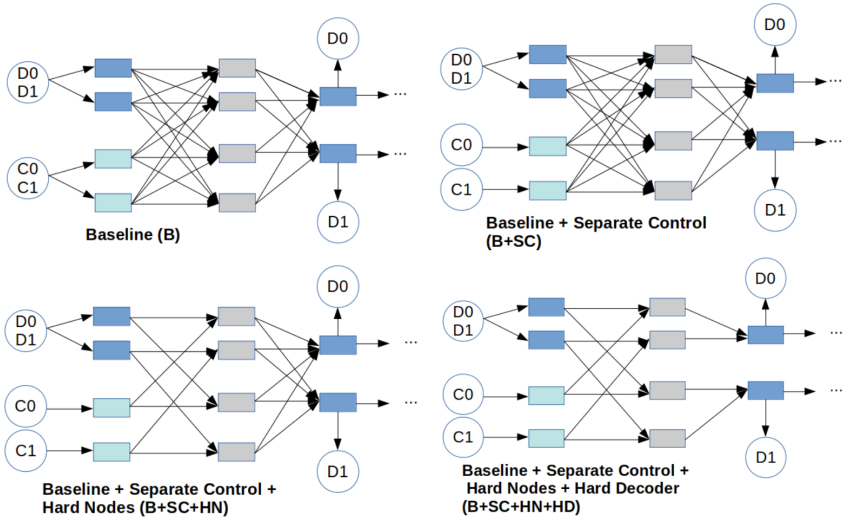
$$u_i = \beta_{i1} u_i' + \beta_{i2} \tag{22}$$

  where $\beta_{i1} \sim \mathcal{U}(0.5, 1.0)$ and $\beta_{i2} \sim \mathcal{U}(-0.2, 0.2)$ define a regime for i-th CV. $u_i' \sim \mathcal{U}(0, 1.0)$ is sampled iid at each time step.
- Then, DVs are generated as per Slide 18 (Experimental Setup slide).
- Suppose there are n regimes for CV1 and CV2 each. Then, there are total n^2 possible regimes for the overall system.
- We trained on 8 regime-combinations and tested on 32 unseen regime-combinations. Train on 1-1, 2-2, ..., 8-8. Test on 1-5,1-6,...,8,3,8-4.

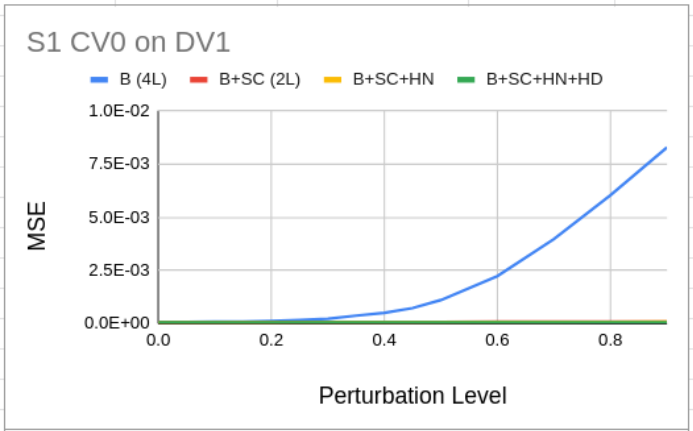
Results

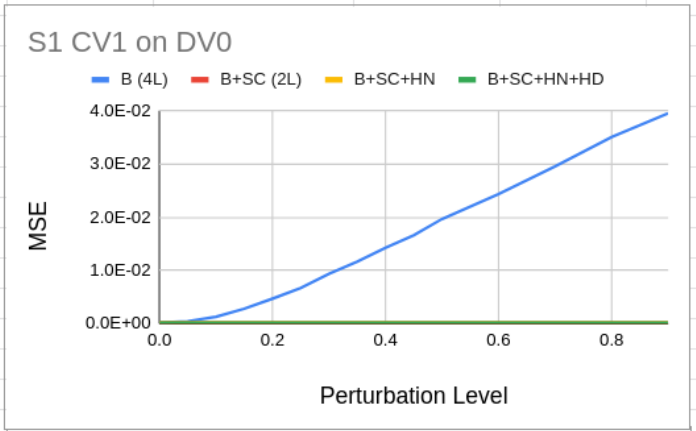
	B	Baseline		SC	SepCTRL		HN	Hard Nodes		HD	Hard Decoder	
	IID				OOD				(OOD-IID)/IID			
Scenario	B	B+SC	B+SC+HN	B+SC+HN+HD	B	B+SC	B+SC+HN	B+SC+HN+HD	B	B+SC	B+SC+HN	B+SC+HN+HD
S1	1.5E-04	1.2E-04	**1.1E-04**	1.2E-04	2.6E-04	1.8E-04	**1.6E-04**	1.9E-04	0.76	**0.47**	0.48	0.55
S2	2.0E-04	1.4E-04	1.5E-04	1.7E-04	3.8E-04	2.4E-04	2.6E-04	2.8E-04	0.87	0.72	0.75	**0.66**
S3	1.8E-04	**1.2E-04**	1.3E-04	1.4E-04	2.9E-04	1.8E-04	2.5E-04	**1.8E-04**	0.64	0.51	0.92	**0.32**
S4	**1.4E-04**	1.6E-04	1.5E-04	1.6E-04	2.5E-04	2.4E-04	2.9E-04	**2.2E-04**	0.74	0.52	0.93	**0.39**
Rank S1	4	2	1	2	4	2	1	3	4	1	2	3
Rank S2	4	1	2	3	4	1	2	3	4	2	3	1
Rank S3	4	1	2	3	4	2	3	1	3	2	4	1
Rank S4	1	4	2	3	3	2	4	1	3	2	4	1
Average Rank	3.25	2	**1.75**	2.75	3.75	**1.75**	2.5	2	3.5	1.75	3.25	**1.5**

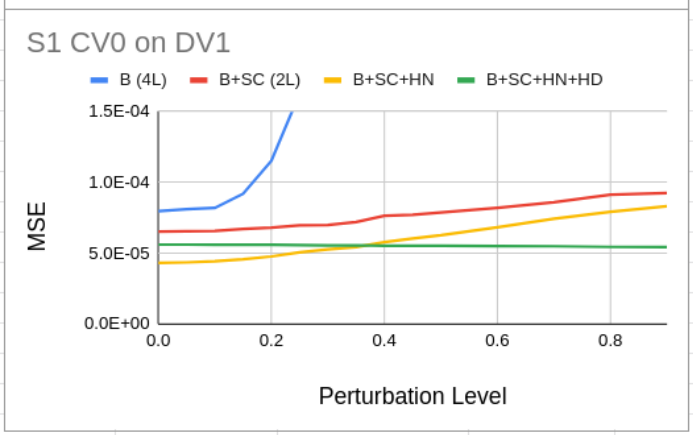


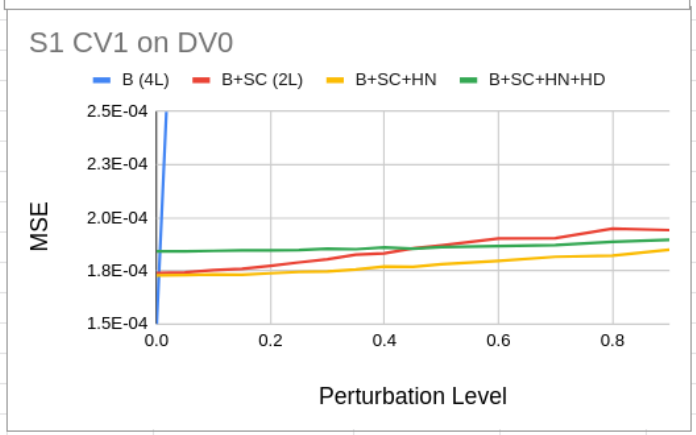


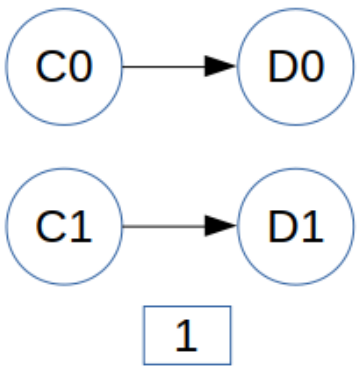
Sensitivity Analysis for Scenario-1



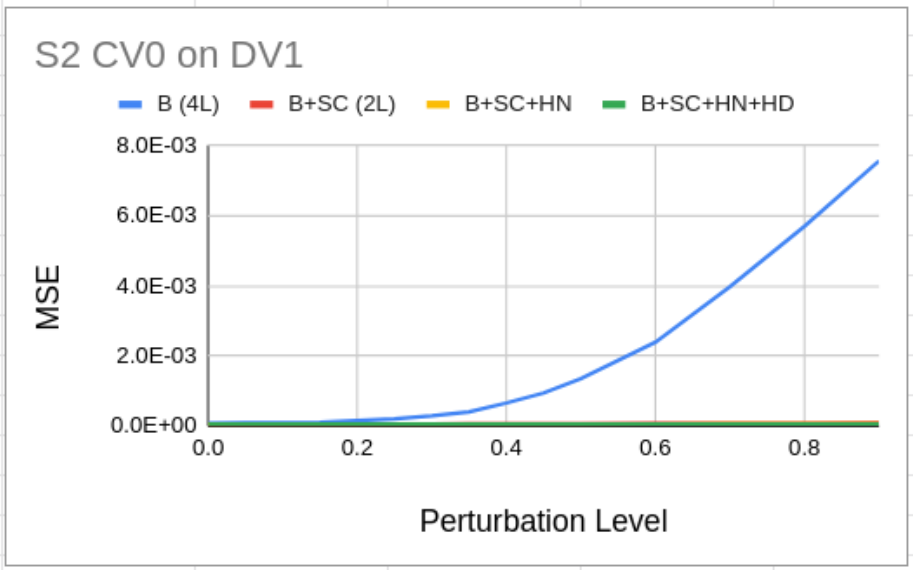


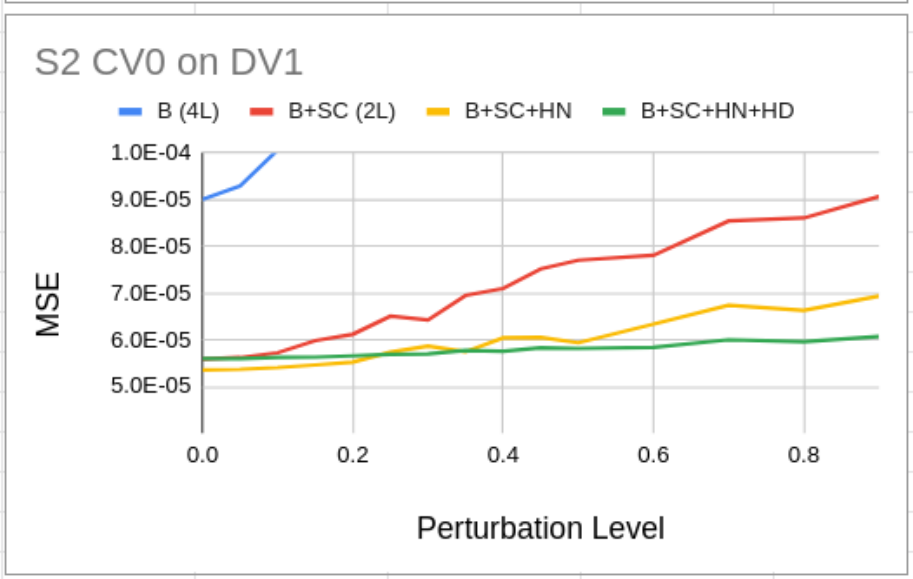


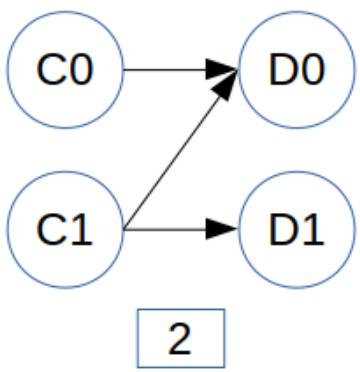




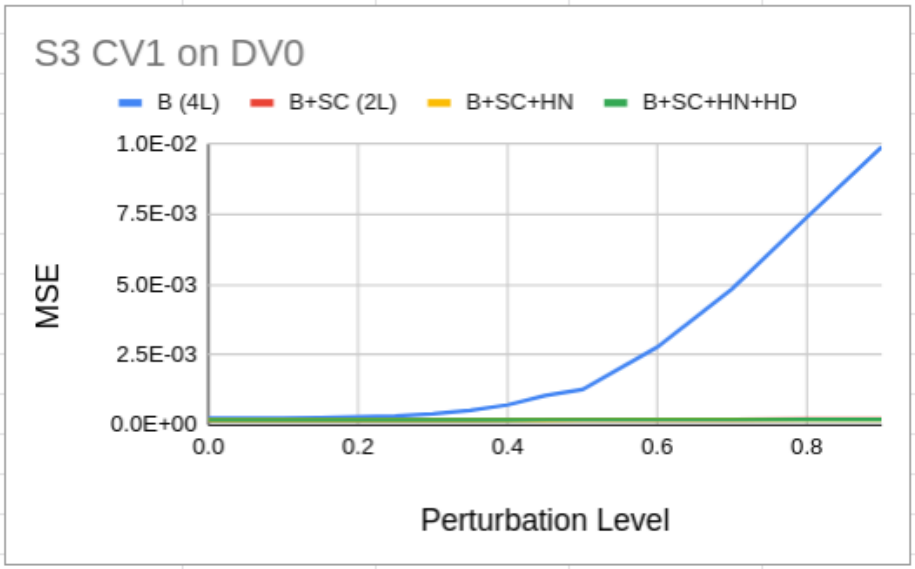
Sensitivity Analysis for Scenario-2

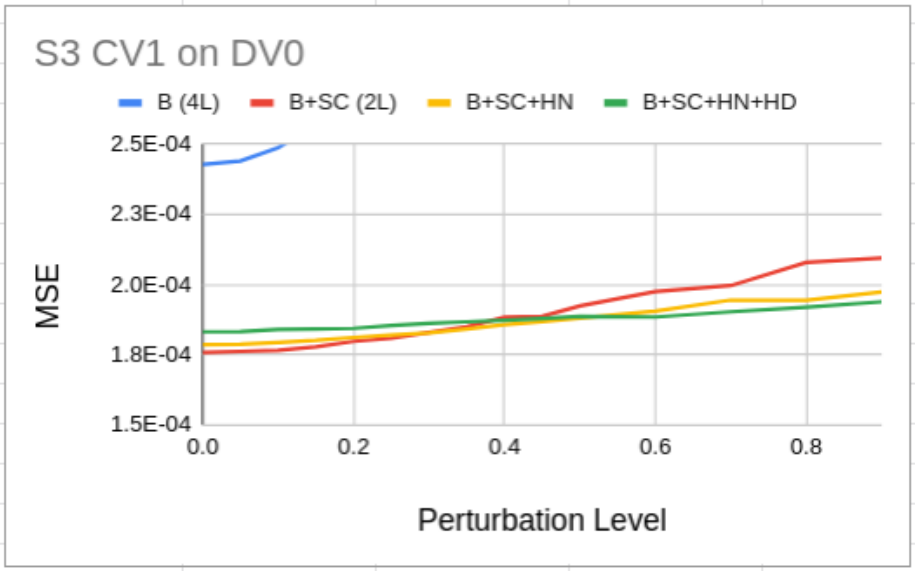


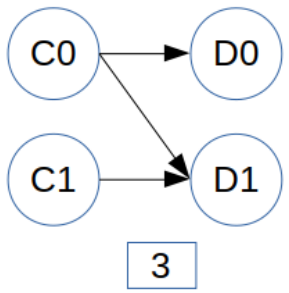




Sensitivity Analysis for Scenario-3





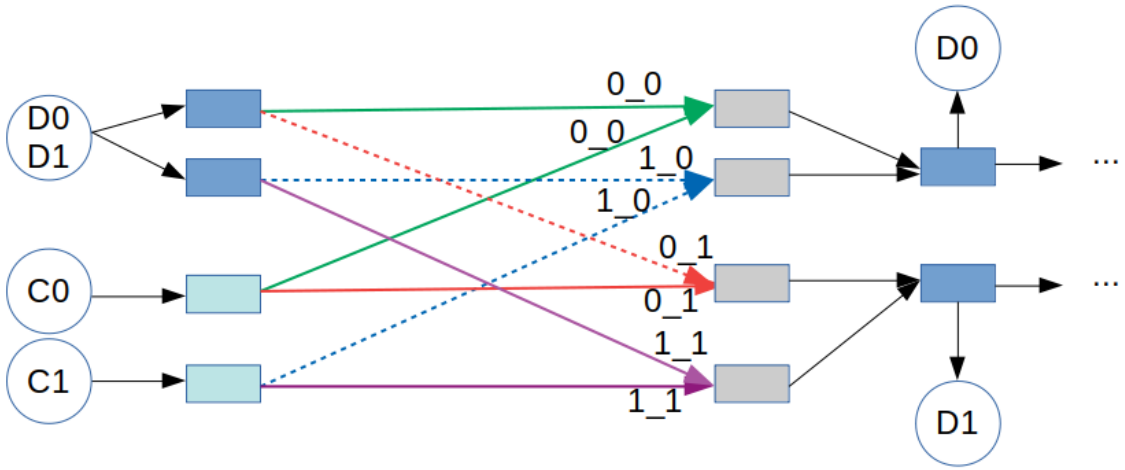


Group LASSO

$$S_{\lambda_1 \eta}(\boldsymbol{w}) \triangleq \begin{cases} \boldsymbol{w} - \lambda_1 \eta \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}, & \|\boldsymbol{w}\|_2 > \lambda_1 \eta \\ 0, & \|\boldsymbol{w}\|_2 \leq \lambda_1 \eta \end{cases}, \quad \forall \boldsymbol{w} \in \mathbb{R}^n.$$

We consider applying group LASSO on the connections in the hidden layers to encourage structure learning in the latent space.

Example: Group LASSO in Scenario S3



GL	Scenario	Norm 0_0		Norm 0_1		Norm 1_0		Norm 1_1		IID MSE	OOD MSE
0	S3	3.96	4.70	3.10	4.22	3.31	4.04	3.32	4.09	1.3E-04	2.8E-04
0.01	S3	3.38	5.28	3.25	4.58	2.93	3.07	3.42	4.82	1.3E-04	2.1E-04
0.5	S3	1.53	1.79	0.17	1.78	0.87	0.23	1.50	1.69	2.0E-04	3.2E-04
1	S3	1.00	0.93	0.18	0.91	0.00	0.00	0.95	0.89	2.5E-04	3.6E-04
1	S3	2.50	2.85	1.52	2.61	1.39	1.08	2.20	2.67	1.2E-04	2.1E-04

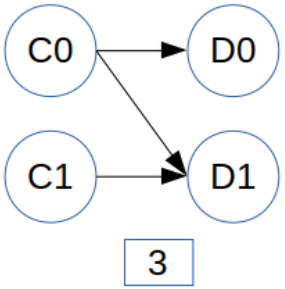
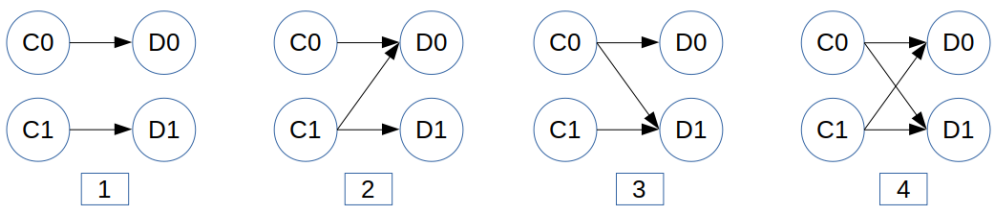


Figure: B+SC+HN+HD

Group LASSO Results Summary

Group LASSO	Scenario	Norm 0_0		Norm 0_1		Norm 1_0		Norm 1_1		IID MSE	OOD MSE
0	S1	3.89	5.18	3.03	4.34	3.19	4.00	3.23	4.21	1.1E-04	1.8E-04
0.01	S1	3.28	5.79	3.12	4.06	2.61	2.53	3.36	4.67	1.2E-04	2.2E-04
0.5	S1	0.79	1.97	0.40	1.32	0.77	0.33	1.55	1.70	2.1E-04	2.7E-04
1	S1	1.17	1.20	0.13	0.06	0.00	0.00	0.09	0.92	2.1E-04	3.3E-04
1	S1	2.16	3.46	1.72	2.31	1.38	1.00	2.59	2.90	1.1E-04	1.9E-04
GL	Scenario	Norm 0_0		Norm 0_1		Norm 1_0		Norm 1_1		IID MSE	OOD MSE
0	S2	3.53	4.78	3.13	4.92	3.84	4.30	3.33	4.55	1.5E-04	3.0E-04
0.01	S2	3.25	4.88	3.18	4.00	2.69	3.83	3.31	4.42	1.7E-04	2.8E-04
0.5	S2	0.74	1.81	1.06	3.28	0.45	0.70	2.07	1.96	2.2E-04	4.0E-04
1	S2	2.50	1.17	0.00	0.00	0.04	0.59	0.19	1.17	2.7E-04	4.7E-04
1	S2	2.38	3.24	1.80	2.42	1.32	2.19	2.39	2.63	1.8E-04	3.2E-04
GL	Scenario	Norm 0_0		Norm 0_1		Norm 1_0		Norm 1_1		IID MSE	OOD MSE
0	S3	3.96	4.70	3.10	4.22	3.31	4.04	3.32	4.09	1.3E-04	2.8E-04
0.01	S3	3.38	5.28	3.25	4.58	2.93	3.07	3.42	4.82	1.3E-04	2.1E-04
0.5	S3	1.53	1.79	0.17	1.78	0.87	0.23	1.50	1.69	2.0E-04	3.2E-04
1	S3	1.00	0.93	0.18	0.91	0.00	0.00	0.95	0.89	2.5E-04	3.6E-04
1	S3	2.50	2.85	1.52	2.61	1.39	1.08	2.20	2.67	1.2E-04	2.1E-04
GL	Scenario	Norm 0_0		Norm 0_1		Norm 1_0		Norm 1_1		IID MSE	OOD MSE
0	S4	3.63	4.43	3.25	4.43	3.92	4.55	3.22	4.44	1.5E-04	3.5E-04
0.01	S4	3.31	5.07	3.37	4.59	2.91	3.79	3.38	4.20	1.6E-04	3.2E-04
0.5	S4	0.42	1.66	0.96	2.38	0.52	0.62	1.91	1.82	2.1E-04	4.1E-04
1	S4	0.73	0.81	0.29	1.13	0.77	0.63	0.24	0.99	4.3E-04	6.2E-04
1	S4	2.14	2.93	1.65	2.86	1.92	2.22	2.40	2.43	1.8E-04	3.6E-04



*last row for each scenario corresponds: group LASSO was applied only for first 120 out of 400 epochs.

Testing ability to handle spurious correlation

- $CV2 = \alpha CV1 + (1 - \alpha)\mathcal{U}(0, 1)$, $\alpha \in [0, 1]$
- α changes every 100 timesteps with probability 0.5.
- Increased α means increased spurious correlation.
- Setup-1: Training and IID testing $\alpha \sim \mathcal{U}(0.1, 0.3)$, OOD Testing $\alpha \sim \mathcal{U}(0.0, 0.1)$
- Setup-2: Training and IID testing $\alpha \sim \mathcal{U}(0.2, 0.5)$, OOD Testing $\alpha \sim \mathcal{U}(0.0, 0.1)$
- Setup-3: Training and IID testing $\alpha \sim \mathcal{U}(0.4, 0.7)$, OOD Testing $\alpha \sim \mathcal{U}(0.0, 0.1)$

Scatter Plots depicting Spurious Correlations between CVs

$$CV2 = \alpha CV1 + (1 - \alpha)\mathcal{U}(0, 1),\ \alpha \in [0, 1]$$

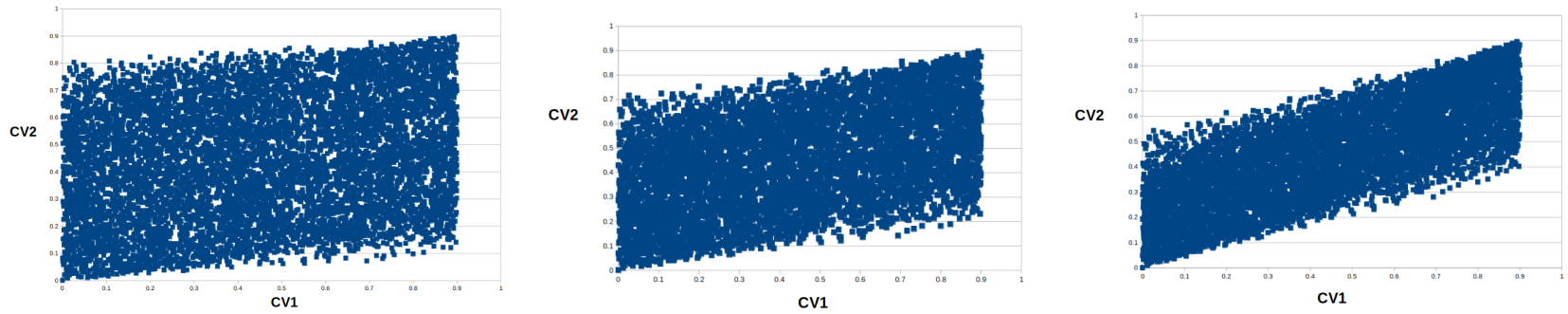


Figure: Training Left: $\alpha \sim \mathcal{U}(0.1, 0.3)$, Center: $\alpha \sim \mathcal{U}(0.2, 0.5)$, Right: $\alpha \sim \mathcal{U}(0.4, 0.7)$

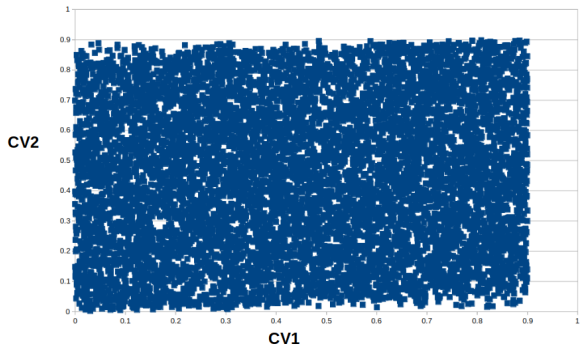
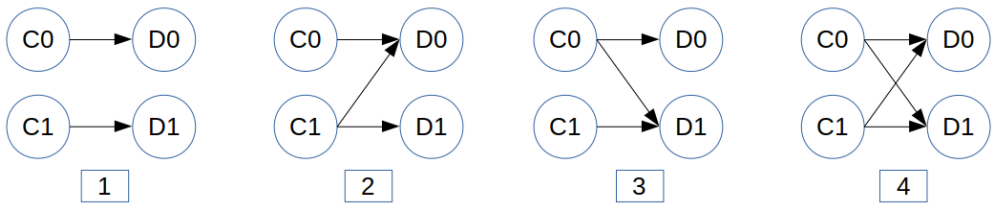


Figure: Testing $\alpha \sim \mathcal{U}(0.0, 0.1)$

Results for Spurious Correlation Testing - 1

	B	Baseline		SC	SepCTRL		HN	Hard Nodes		HD	Hard Decoder	
	IID (alpha ~ U(0.1,0.3))				OOD (alpha ~ U(0.0,0.1))				(OOD-IID)/IID			
Scenario	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**
S1	6.3E-05	5.4E-05	**4.3E-05**	4.4E-05	8.4E-05	7.8E-05	**5.4E-05**	6.1E-05	0.33	0.44	0.26	0.39
S2	7.9E-05	7.3E-05	**6.3E-05**	7.3E-05	1.2E-04	1.1E-04	**8.6E-05**	1.5E-04	0.57	0.54	0.37	1.01
S3	8.7E-05	6.5E-05	**5.8E-05**	6.2E-05	1.4E-04	7.9E-05	**7.1E-05**	8.0E-05	0.65	0.22	0.23	0.29
S4	9.1E-05	9.3E-05	**6.8E-05**	7.5E-05	1.3E-04	1.4E-04	**8.5E-05**	1.5E-04	0.41	0.46	0.24	0.96
Rank S1	4	3	1	2	4	3	1	2	2	4	1	3
Rank S2	4	3	1	2	3	2	1	4	3	2	1	4
Rank S3	4	3	1	2	4	2	1	3	4	1	2	3
Rank S4	3	4	1	2	2	3	1	4	2	3	1	4
Average Rank	3.75	3.25	**1**	2	3.25	2.5	**1**	3.25	2.75	2.5	**1.25**	3.5

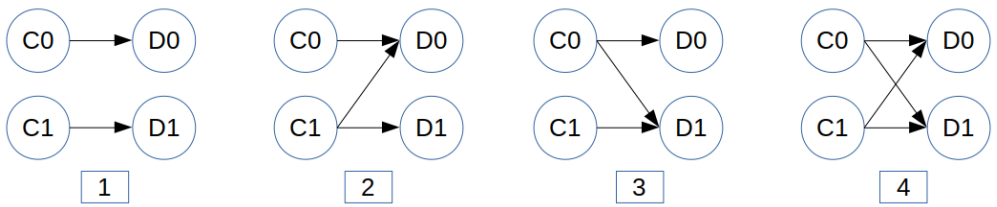


Observations:

- Separate Control and Hard Nodes help both in IID and OOD in all scenarios S1-S4.
- Baseline model is the worst in IID setting.
- Not sure why Hard Decoder is hurting performance.

Results for Spurious Correlation Testing - 2

	B	Baseline		SC	SepCTRL		HN	Hard Nodes		HD	Hard Decoder	
	IID (alpha ~ U(0.2,0.5))				OOD (alpha ~ U(0.0,0.1))				(OOD-IID)/IID			
Scenario	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**
S1	5.3E-05	5.2E-05	4.4E-05	**4.1E-05**	1.7E-04	1.2E-04	9.6E-05	**9.0E-05**	2.28	1.25	**1.20**	1.22
S2	6.8E-05	6.8E-05	7.8E-05	**6.6E-05**	1.6E-04	1.9E-04	1.7E-04	**1.5E-04**	1.41	1.83	**1.14**	1.21
S3	5.9E-05	6.3E-05	**5.7E-05**	6.4E-05	2.2E-04	1.1E-04	**8.7E-05**	1.2E-04	2.77	0.73	**0.52**	0.85
S4	**7.5E-05**	8.5E-05	7.7E-05	8.6E-05	2.2E-04	2.0E-04	**1.2E-04**	1.6E-04	1.95	1.36	**0.55**	0.86
Rank S1	4	3	2	1	4	3	2	1	4	3	1	2
Rank S2	2	3	4	1	2	4	3	1	3	4	1	2
Rank S3	2	3	1	4	4	2	1	3	4	2	1	3
Rank S4	1	3	2	4	4	3	1	2	4	3	1	2
Average Rank	2.67	3	2.33	**2**	3.33	3	2	**1.67**	3.67	3	**1**	2.33

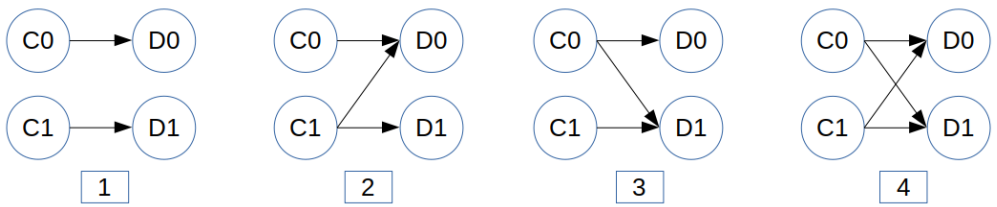


Observations:

- All methods have similar performance on IID but not on OOD.
- With increased α at train time, the performance of Baseline B for OOD test degrades further.
- Unlike in previous results, Hard Decoder is also helping.

Results for Spurious Correlation Testing - 3

	B	Baseline		SC	SepCTRL		HN	Hard Nodes		HD	Hard Decoder	
	IID (alpha ~ U(0.4,0.7))				OOD (alpha ~ U(0.0,0.1))				(OOD-IID)/IID			
Scenario	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**
S1	5.6E-05	4.0E-05	4.0E-05	**3.9E-05**	1.5E-03	1.2E-04	**1.1E-04**	1.3E-04	25.66	2.12	**1.90**	2.27
S2	7.4E-05	**5.5E-05**	7.4E-05	6.0E-05	3.7E-04	1.8E-04	**1.3E-04**	1.9E-04	3.92	2.25	**0.77**	2.24
S3	8.6E-05	5.9E-05	**5.6E-05**	6.6E-05	5.6E-04	1.8E-04	**1.1E-04**	1.9E-04	5.49	2.13	**0.96**	1.80
S4	1.1E-04	**6.9E-05**	8.1E-05	8.0E-05	1.6E-03	**1.3E-04**	1.4E-04	2.2E-04	13.29	0.95	**0.76**	1.70
Rank S1	4	3	2	1	4	2	1	3	4	2	1	3
Rank S2	4	1	3	2	4	2	1	3	4	3	1	2
Rank S3	4	2	1	3	4	2	1	3	4	3	1	2
Rank S4	4	1	3	2	4	1	2	3	4	2	1	3
Average Rank	4	1.75	2.25	2	4	1.75	1.25	3	4	2.50	1.00	2.50



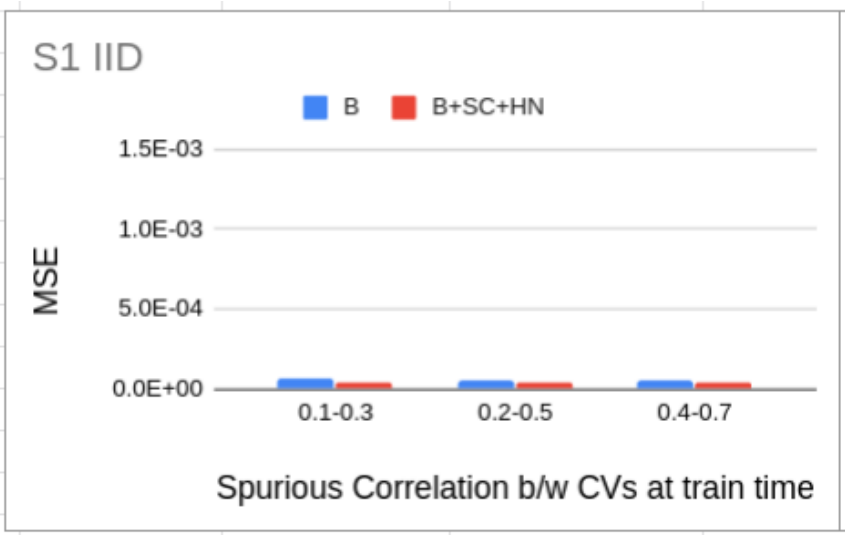
Summary results with varying levels of spurious correlation

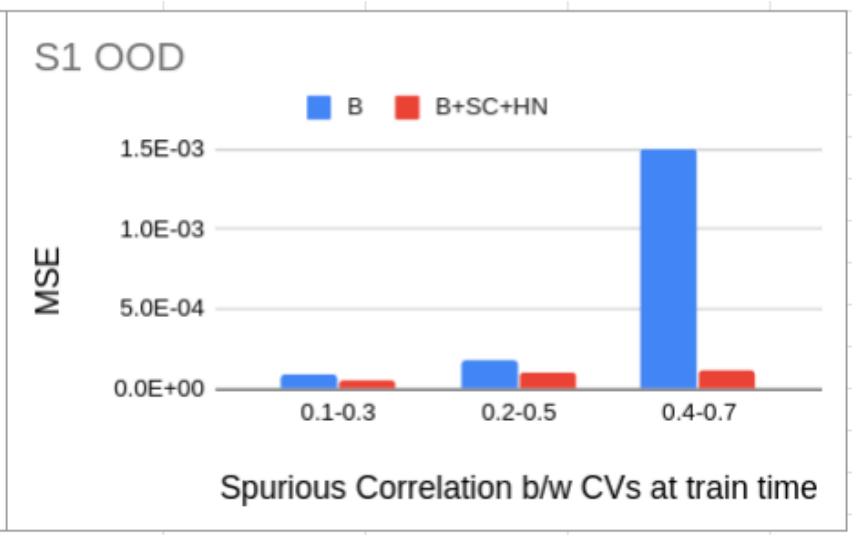
	S1 IID		S1 OOD	
alpha	B	B+SC+HN	B	B+SC+HN
0.1-0.3	6.3E-05	4.3E-05	8.4E-05	5.4E-05
0.2-0.5	5.3E-05	4.4E-05	1.7E-04	9.6E-05
0.4-0.7	5.6E-05	4.0E-05	1.5E-03	1.1E-04

	S2 IID		S2 OOD	
alpha	B	B+SC+HN	B	B+SC+HN
0.1-0.3	7.9E-05	6.3E-05	1.2E-04	8.6E-05
0.2-0.5	6.8E-05	7.8E-05	1.6E-04	1.7E-04
0.4-0.7	7.4E-05	7.4E-05	3.7E-04	1.3E-04

	S3 IID		S3 OOD	
alpha	B	B+SC+HN	B	B+SC+HN
0.1-0.3	8.7E-05	5.8E-05	1.4E-04	7.1E-05
0.2-0.5	5.9E-05	5.7E-05	2.2E-04	8.7E-05
0.4-0.7	8.6E-05	5.6E-05	5.6E-04	1.1E-04

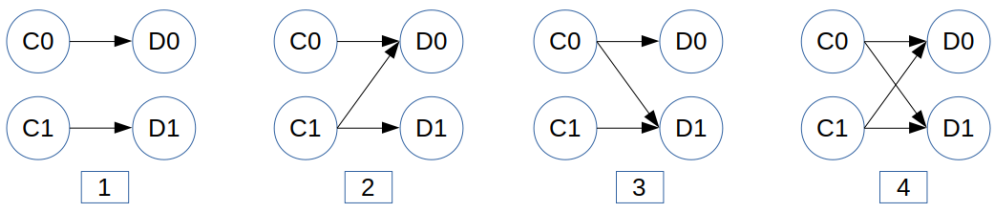
	S4 IID		S4 OOD	
alpha	B	B+SC+HN	B	B+SC+HN
0.1-0.3	9.1E-05	6.8E-05	1.3E-04	8.5E-05
0.2-0.5	7.5E-05	7.7E-05	2.2E-04	1.2E-04
0.4-0.7	1.1E-04	8.1E-05	1.6E-03	1.4E-04





Results for Spurious Correlation Testing - 3, Using ReLU instead of tanh in CNNs (and LSTM removed)

	B	Baseline		SC	SepCTRL		HN	Hard Nodes		HD	Hard Decoder	
	IID (alpha ~ U(0.4,0.7))				OOD (alpha ~ U(0.0,0.1))				(OOD-IID)/IID			
Scenario	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**	**B**	**B+SC**	**B+SC+HN**	**B+SC+HN+HD**
S1	1.2E-04	8.0E-05	8.9E-05	7.1E-05	2.3E-03	3.1E-04	2.2E-04	1.8E-04	18.82	2.89	1.50	1.62
S2	1.6E-04	2.0E-04	1.5E-04	1.2E-04	2.3E-03	6.0E-04	3.9E-04	3.4E-04	13.54	1.97	1.65	1.78
S3	1.6E-04	1.6E-04	1.3E-04	1.1E-04	1.9E-03	5.0E-04	3.3E-04	2.9E-04	10.86	2.03	1.65	1.72
S4	7.2E-04	2.0E-04	1.5E-04	1.4E-04	7.4E-03	6.4E-04	4.9E-04	3.8E-04	9.32	2.15	2.28	1.62
Rank S1	4	2	3	1	4	3	2	1	4	3	1	2
Rank S2	3	4	2	1	4	3	2	1	4	3	1	2
Rank S3	3	4	2	1	4	3	2	1	4	3	1	2
Rank S4	4	3	2	1	4	3	2	1	4	2	3	1
Average Rank	4	3.25	2.25	1	4	3	2	1	4	2.75	1.50	1.75



Summary

- Baseline model without suitable structural biases struggles to learn well in the IID case. On OOD case, its performance degrades further in comparison to architectures with structural bias.
- Separate Control, Hard Nodes and Hard Decoder all contribute to learning the correct sensitivities.
- A combination of Separate Control, Hard Nodes and Hard Decoder leads to good generalization on unseen regimes.
- Group LASSO leads to learning the correct structure in the latent space but struggles to perform well on forecasting.
- Structural biases can help deal with spurious correlations across dimensions.
- Having large window length leads to spurious correlations within a CV in case of separate control models in the current setup as each timestep is sampled IID from same distribution. Need to revisit this.

Next Steps

- Evaluate **explicitly in the "unseen" (OOD) region** of the CV1-CV2 space.
- **Quantify sensitivity analysis results**, e.g. using autograd.
- Look at **activation maps of latent nodes**. Although the norms of cross-connection weights are non-zero, information flow may not be there as suggested by the sensitivity plots.
- Finding suitable **real-world datasets or more realistic simulation environments** used in literature. GHL, SWaT, etc. may not be suitable for testing under our scenarios as entire data follows same distribution.
- Trying L1 **regularizer**. Look at how Group LASSO can be made to work well for learning the right latent structure while not hurting forecasting performance.
- Explore further along lines of **neural network attribution** [13].

References I

[1] Alessandro Achille et al. "Life-long disentangled representation learning with cross-domain latent homologies". In: *Advances in Neural Information Processing Systems*. 2018, pp. 9873–9883.

[2] Ankesh Anand et al. "Unsupervised State Representation Learning in Atari". In: *arXiv preprint arXiv:1906.08226* (2019).

[3] Jacob Andreas et al. "Neural module networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 39–48.

[4] Lynton Ardizzone et al. "Analyzing inverse problems with invertible neural networks". In: *arXiv preprint arXiv:1808.04730* (2018).

[5] Sanjeev Arora et al. "A theoretical analysis of contrastive unsupervised representation learning". In: *arXiv preprint arXiv:1902.09229* (2019).

[6] Philip Bachman, R Devon Hjelm, and William Buchwalter. "Learning representations by maximizing mutual information across views". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15509–15519.

[7] Peter W Battaglia et al. "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1806.01261* (2018).

[8] Yoshua Bengio. "The consciousness prior". In: *arXiv preprint arXiv:1709.08568* (2017).

[9] Yoshua Bengio et al. "A meta-transfer objective for learning to disentangle causal mechanisms". In: *arXiv preprint arXiv:1901.10912* (2019).

[10] William Bialek and Naftali Tishby. "Predictive information". In: *arXiv preprint cond-mat/9902341* (1999).

[11] Christopher P Burgess et al. "Monet: Unsupervised scene decomposition and representation". In: *arXiv preprint arXiv:1901.11390* (2019).

References II

[12] Hugo Caselles-Dupré, Michael Garcia Ortiz, and David Filliat. "Symmetry-Based Disentangled Representation Learning requires Interaction with Environments". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 4608–4617. URL: http://papers.nips.cc/paper/8709-symmetry-based-disentangled-representation-learning-requires-interaction-with-environments.pdf.

[13] Aditya Chattopadhyay et al. "Neural network attributions: A causal perspective". In: *arXiv preprint arXiv:1902.02302* (2019).

[14] Ting Chen et al. "A simple framework for contrastive learning of visual representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[15] Ziqiang Cheng et al. "Time2Graph: Revisiting Time Series Modeling with Dynamic Shapelets". In: *arXiv preprint arXiv:1911.04143* (2019).

[16] Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. "The evolutionary origins of modularity". In: *Proceedings of the Royal Society b: Biological sciences* 280.1755 (2013), p. 20122863.

[17] Robert Desimone and John Duncan. "Neural mechanisms of selective visual attention". In: *Annual review of neuroscience* 18.1 (1995), pp. 193–222.

[18] Kien Do and Truyen Tran. "Theory and evaluation metrics for learning disentangled representations". In: *arXiv preprint arXiv:1908.09961* (2019).

[19] Andreas Doerr et al. "Probabilistic recurrent state-space models". In: *arXiv preprint arXiv:1801.10395* (2018).

[20] Babak Esmaeili et al. "Hierarchical disentangled representations". In: *stat* 1050 (2018), p. 12.

[21] Richard Evans et al. "Making sense of sensory input". In: *arXiv preprint arXiv:1910.02227* (2019).

References III

[22] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. "Unsupervised Scalable Representation Learning for Multivariate Time Series". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 4652–4663. URL: http://papers.nips.cc/paper/8713-unsupervised-scalable-representation-learning-for-multivariate-time-series.pdf.

[23] Ruiqi Gao et al. "Flow Contrastive Estimation of Energy-Based Models". In: *arXiv preprint arXiv:1912.00589* (2019).

[24] Robert Geirhos et al. "Generalisation in humans and deep neural networks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7538–7550.

[25] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *arXiv preprint arXiv:1811.12231* (2018).

[26] Rohit Girdhar and Deva Ramanan. "CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning". In: *arXiv preprint arXiv:1910.04744* (2019).

[27] Muhammad Waleed Gondal et al. "On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset". In: *arXiv preprint arXiv:1906.03292* (2019).

[28] Olivier Goudet et al. "Causal generative neural networks". In: *arXiv preprint arXiv:1711.08936* (2017).

[29] Anirudh Goyal et al. "Recurrent independent mechanisms". In: *arXiv preprint arXiv:1909.10893* (2019).

[30] Anirudh Goyal et al. "Reinforcement Learning with Competitive Ensembles of Information-Constrained Primitives". In: *arXiv preprint arXiv:1906.10667* (2019).

[31] Stephen Grossberg. "Contour enhancement, short term memory, and constancies in reverberating neural networks". In: *Studies of mind and brain*. Springer, 1982, pp. 332–378.

[32] Kevin Gurney, Tony J Prescott, and Peter Redgrave. "A computational model of action selection in the basal ganglia. I. A new functional anatomy". In: *Biological cybernetics* 84.6 (2001), pp. 401–410.

References IV

[33] Olivier J Hénaff et al. "Data-efficient image recognition with contrastive predictive coding". In: *arXiv preprint arXiv:1905.09272* (2019).

[34] R Devon Hjelm et al. "Learning deep representations by mutual information estimation and maximization". In: *arXiv preprint arXiv:1808.06670* (2018).

[35] Jun-Ting Hsieh et al. "Learning to decompose and disentangle representations for video prediction". In: *Advances in Neural Information Processing Systems*. 2018, pp. 517–526.

[36] Wei-Ning Hsu, Yu Zhang, and James Glass. "Unsupervised learning of disentangled and interpretable representations from sequential data". In: *Advances in neural information processing systems*. 2017, pp. 1878–1889.

[37] Biwei Huang et al. "Causal Discovery and Forecasting in Nonstationary Environments with State-Space Models". In: *arXiv preprint arXiv:1905.10857* (2019).

[38] Drew Hudson and Christopher D Manning. "Learning by abstraction: The neural state machine". In: *Advances in Neural Information Processing Systems*. 2019, pp. 5901–5914.

[39] Drew A Hudson and Christopher D Manning. "Gqa: A new dataset for real-world visual reasoning and compositional question answering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6700–6709.

[40] Aapo Hyvarinen and Hiroshi Morioka. "Unsupervised feature extraction by time-contrastive learning and nonlinear ICA". In: *Advances in Neural Information Processing Systems*. 2016, pp. 3765–3773.

[41] AJ Hyvarinen and Hiroshi Morioka. "Nonlinear ICA of temporally dependent stationary sources". In: Proceedings of Machine Learning Research. 2017.

[42] Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. "i-revnet: Deep invertible networks". In: *arXiv preprint arXiv:1802.07088* (2018).

References V

[43] Miguel Jaques, Michael Burke, and Timothy Hospedales. "Physics-as-Inverse-Graphics: Joint Unsupervised Learning of Objects and Physics from Video". In: *arXiv preprint arXiv:1905.11169* (2019).

[44] Jason Jo and Yoshua Bengio. "Measuring the tendency of CNNs to learn surface statistical regularities". In: *arXiv preprint arXiv:1711.11561* (2017).

[45] Justin Johnson et al. "Clevr: A diagnostic dataset for compositional language and elementary visual reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2901–2910.

[46] Nan Rosemary Ke et al. "Learning Neural Causal Models from Unknown Interventions". In: *arXiv preprint arXiv:1910.01075* (2019).

[47] Ilyes Khemakhem, Diederik P Kingma, and Aapo Hyvärinen. "Variational autoencoders and nonlinear ica: A unifying framework". In: *arXiv preprint arXiv:1907.04809* (2019).

[48] Ilyes Khemakhem et al. "ICE-BeeM: Identifiable Conditional Energy-Based Deep Models". In: *arXiv preprint arXiv:2002.11537* (2020).

[49] Yunji Kim et al. "Unsupervised Keypoint Learning for Guiding Class-Conditional Video Prediction". In: *Advances in Neural Information Processing Systems*. 2019, pp. 3809–3819.

[50] Thomas Kipf, Elise van der Pol, and Max Welling. "Contrastive Learning of Structured World Models". In: *arXiv preprint arXiv:1911.12247* (2019).

[51] Thomas Kipf et al. "Neural relational inference for interacting systems". In: *arXiv preprint arXiv:1802.04687* (2018).

[52] Adam Kosiorek et al. "Sequential attend, infer, repeat: Generative modelling of moving objects". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8606–8616.

[53] Jannik Kossen et al. "Structured Object-Aware Physics Prediction for Video Modeling and Planning". In: *arXiv preprint arXiv:1910.02425* (2019).

References VI

[54] Guokun Lai et al. "Modeling long-and short-term temporal patterns with deep neural networks". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 95–104.

[55] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[56] Shen Li, Bryan Hooi, and Gim Hee Lee. "Identifying through Flows for Recovering Latent Representations". In: *arXiv preprint arXiv:1909.12555* (2019).

[57] Shuheng Li, Dezhi Hong, and Hongning Wang. "Relation Inference among Sensor Time Series in Smart Buildings with Metric Learning". In: *AAAI* (2020).

[58] Bryan Lim. "Forecasting treatment responses over time using recurrent marginal structural networks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 7483–7493.

[59] Bryan Lim, Stefan Zohren, and Stephen Roberts. "Recurrent Neural Filters: Learning Independent Bayesian Filtering Steps for Time Series Prediction". In: *arXiv preprint arXiv:1901.08096* (2019).

[60] Francesco Locatello et al. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *arXiv preprint arXiv:1811.12359* (2018).

[61] Ricky Loynd et al. "Working Memory Graphs". In: *arXiv preprint arXiv:1911.07141* (2019).

[62] Qianli Ma et al. "Learning Representations for Time Series Clustering". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 3776–3786. URL: http://papers.nips.cc/paper/8634-learning-representations-for-time-series-clustering.pdf.

[63] David A McAllester. "Some pac-bayesian theorems". In: *Machine Learning* 37.3 (1999), pp. 355–363.

[64] Risto Miikkulainen et al. "Evolving deep neural networks". In: *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Elsevier, 2019, pp. 293–312.

References VII

[65] Matthias Minderer et al. "Unsupervised Learning of Object Structure and Dynamics from Videos". In: *arXiv preprint arXiv:1906.07889* (2019).

[66] Nikhil Mishra, Pieter Abbeel, and Igor Mordatch. "Prediction and control with temporal segment models". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2459–2468.

[67] Ricardo Pio Monti, Kun Zhang, and Aapo Hyvarinen. "Causal Discovery with General Non-Linear Relationships Using Non-Linear ICA". In: *arXiv preprint arXiv:1904.09096* (2019).

[68] Nikhil Muralidhar, Sathappah Muthiah, and Naren Ramakrishnan. "DyAt nets: dynamic attention networks for state forecasting in cyber-physical systems". In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press. 2019, pp. 3180–3186.

[69] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[70] Sherjil Ozair et al. "Wasserstein dependency measure for representation learning". In: *Advances in Neural Information Processing Systems*. 2019, pp. 15578–15588.

[71] Giambattista Parascandolo et al. "Learning independent causal mechanisms". In: *arXiv preprint arXiv:1712.00961* (2017).

[72] Sören Pirk et al. "Online Object Representations with Contrastive Learning". In: *arXiv preprint arXiv:1906.04312* (2019).

[73] Elise van der Pol et al. "Plannable Approximations to MDP Homomorphisms: Equivariance under Actions". In: *arXiv preprint arXiv:2002.11963* (2020).

[74] Rajesh PN Rao and Dana H Ballard. "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects". In: *Nature neuroscience* 2.1 (1999), p. 79.

References VIII

[75] Clemens Rosenbaum et al. "Routing networks and the challenges of modular and compositional computation". In: *arXiv preprint arXiv:1904.12774* (2019).

[76] David Salinas et al. "High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 6827–6837. URL: http://papers.nips.cc/paper/8907-high-dimensional-multivariate-forecasting-with-low-rank-gaussian-copula-processes.pdf.

[77] Hiroaki Sasaki et al. "Robust contrastive learning and nonlinear ICA in the presence of outliers". In: *arXiv preprint arXiv:1911.00265* (2019).

[78] Bernhard Schölkopf. "Causality for Machine Learning". In: *arXiv preprint arXiv:1911.10500* (2019).

[79] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4838–4847.

[80] Pierre Sermanet et al. "Time-contrastive networks: Self-supervised learning from video". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1134–1141.

[81] Rui Shu et al. "Predictive Coding for Locally-Linear Control". In: *arXiv preprint arXiv:2003.01086* (2020).

[82] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. "Disentanglement by Nonlinear ICA with General Incompressible-flow Networks (GIN)". In: *arXiv preprint arXiv:2001.04872* (2020).

[83] Olaf Sporns and Richard F Betzel. "Modular brain networks". In: *Annual review of psychology* 67 (2016), pp. 613–640.

[84] Aleksandar Stanić and Jürgen Schmidhuber. "R-SQAIR: Relational Sequential Attend, Infer, Repeat". In: *arXiv preprint arXiv:1910.05231* (2019).

References IX

[85] Sjoerd van Steenkiste et al. "Are Disentangled Representations Helpful for Abstract Visual Reasoning?" In: *arXiv preprint arXiv:1905.12506* (2019).

[86] Raphael Suter et al. "Robustly Disentangled Causal Mechanisms: Validating Deep Representations for Interventional Robustness". In: *International Conference on Machine Learning*. 2019, pp. 6056–6065.

[87] Michael Tschannen et al. "On mutual information maximization for representation learning". In: *arXiv preprint arXiv:1907.13625* (2019).

[88] Sjoerd Van Steenkiste et al. "Relational neural expectation maximization: Unsupervised discovery of objects and their interactions". In: *arXiv preprint arXiv:1802.10353* (2018).

[89] Sagar Verma et al. "Modeling electrical motor dynamics using encoder-decoder with recurrent skip connection". In: *AAAI 2020: 34th AAAI conference on artificial intelligence*. 2020, pp. 1–8.

[90] LE Vincent and Nicolas Thome. "Shape and time distortion loss for training deep time series forecasting models". In: *Advances in Neural Information Processing Systems*. 2019, pp. 4191–4203.

[91] Daniel E Worrall et al. "Interpretable transformations with encoder-decoder networks". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 5726–5735.

[92] Zonghan Wu et al. "Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks". In: *arXiv preprint arXiv:2005.11650* (2020).

[93] Saining Xie et al. "Exploring randomly wired neural networks for image recognition". In: *arXiv preprint arXiv:1904.01569* (2019).

[94] Zhenjia Xu et al. "Unsupervised Discovery of Parts, Structure, and Dynamics". In: *arXiv preprint arXiv:1903.05136* (2019).

[95] Wilson Yan et al. "Learning Predictive Representations for Deformable Objects Using Contrastive Estimation". In: *arXiv preprint arXiv:2003.05436* (2020).

References X

[96] Ge Yang et al. "Plan2Vec: Unsupervised Representation Learning by Latent Plans". In: *arXiv preprint arXiv:2005.03648* (2020).

[97] Kexin Yi et al. "Clevrer: Collision events for video representation and reasoning". In: *arXiv preprint arXiv:1910.01442* (2019).

[98] Kexin Yi et al. "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding". In: *Advances in Neural Information Processing Systems*. 2018, pp. 1031–1042.

[99] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. "Time-series Generative Adversarial Networks". In: *Advances in Neural Information Processing Systems 32*. 2019, pp. 5508–5518.

[100] Wei Yu et al. "EFFICIENT AND INFORMATION-PRESERVING FUTURE FRAME PREDICTION AND BEYOND". In: (2020).

[101] Rowan Zellers et al. "From recognition to cognition: Visual commonsense reasoning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6720–6731.

[102] Qin Zhang et al. "Salient subsequence learning for time series clustering". In: *IEEE transactions on pattern analysis and machine intelligence* 41.9 (2018), pp. 2193–2207.

The End