Counterfactual Dynamics Forecasting — a New Setting of Quantitative Reasoning

Yanzhu Liu, Ying Sun, Joo-Hwee Lim
Institute for Infocomm Research (I²R) & Centre for Frontier AI Research (CFAR), A*STAR, Singapore
liu_yanzhu@i2r.a-star.edu.sg, suny@i2r.a-star.edu.sg, joohwee@i2r.a-star.edu.sg

Abstract
Rethinking and introspection are important elements of human intelligence. To mimic these capabilities, counterfactual reasoning has attracted attention of AI researchers recently, which aims to forecast the alternative outcomes for hypothetical scenarios (“what-if”). However, most existing approaches focused on qualitative reasoning (e.g., causal-effect relationship). It lacks a well-defined description of the differences between counterfactuals and facts, as well as how these differences evolve over time. This paper introduces a new problem formulation — counterfactual dynamics forecasting — which is described in middle-level abstraction under the structural causal models (SCM) framework and derived as ordinary differential equations (ODEs) as low-level quantitative computation. Based on it, we propose a method to infer counterfactual dynamics considering the factual dynamics as demonstration. The experimental results on two dynamical systems demonstrate the effectiveness of the proposed method.

Introduction
“It may happen that small differences in the initial conditions produce very great ones in the final phenomena” — Henri Poincaré, 1908

Looking back to the past, we often imagine how things could have been different if the situation was changed. With regret or relief, we learn the lessons for better future decisions. While the estimation of the antecedents’ development are often not accurate in human brains, in fact, optimists’ and pessimists’ assessment might be worlds apart. Recently, Artificial General Intelligence (AGI) has engaged in imitating this important capability of human intelligence. Although it is challenging to equip AI agents with the counterfactual imagination, by nature they have advantages of precise and large-scale computation.

The term “counterfactual” has been used to define several different tasks in machine learning field. Counterfactual Explanation (Verma, Dickerson, and Hines 2020) seeks to provide the explanations that what changes in the input (e.g., visual features) are needed to alter the decision of the machine (e.g., predicted class labels). The inputs can be images (Sauer and Geiger 2021), tabular vectors (Wachter, Mittelstadt, and Russell 2017), sequential actions (Tsirtsis, De, and Rodriguez 2021), graph data (Bajaj et al. 2021) and so on. The task targets on the interpretability of machine learning algorithm. Counterfactual data augmentation aims to generate more training instances by sampling from a counterfactual data distribution (Kaushik, Hovy, and Lipton 2019) (Pitis, Creager, and Garg 2020). Counterfactual treatment effect estimation infers the expected difference between outcomes with and without the treatment assignment from observational data (Lu et al. 2020). Only partial observation on potential outcomes can be made because individual instances cannot receive and refuse intervention assignment simultaneously. Moreover, the intervention is not independent. It is affected by instance features and latent exogenous variables. Counterfactual regret minimization solving Imperfect Information Games aims to plan the successive strategy under uncertainty about the opponent’s information (Li et al. 2019). However, none of these problem statements focus on simulating the procedure of human rethinking, which consists of three aspects: a) there is factual observation as reference or demonstration; b) limited differences lie between factual and counterfactual initial conditions; c) the imagination world can be described as a dynamical system of multiple subjects with continuous evolution and instantaneous interactive events between them.

Therefore, this paper defines a new problem formulation — counterfactual dynamics forecasting. In this task, a data instance is a pair of sequences: one is the factual observation, and the other is the counterfactual series to be estimated based on its first slot. The two initial slots of factual sequence and counterfactual sequence differ in a constrained extent. The research (Byrne 2019) classifies counterfactuals into four types: additive, subtractive, better-world and worse-world. Such categorization is based on the evidence of psychology and cognitive science about human reasoning. On the basis of this study, we explore further the feasible categorization in the context of quantitative reasoning. By formulating the differences between initial values of factual and counterfactual series, we characterize three types of counterfactuals as observation counterfactuals (Type I), covariate counterfactuals (Type II) and subtractive counterfactuals (Type III). Figure 1 gives an illustration of counterfactual types by using a bouncing ball system as example. Figure 1a shows the factual trajectories of three balls. The observation variables include location and velocity of each...
ball, and the invisible covariates include mass, friction coefficients and restitution coefficients. In Figure 1b, the orange ball is moving in a smaller velocity (observation counterfactual) so that it is too late to collide with the blue one. Figure 1c describes what would happen if the orange ball becomes heavier enough (covariate counterfactual). Figure 1d shows the alternative outcome if one of the balls is removed (subtractive counterfactual).

All the three types of counterfactuals have some existing applications. For example in computer vision regime, ComPhy (Baradel et al. 2020) instantiates the factual sequences as videos, where four cubes are stacked into a tower in an unstable state, and the subsequent falling down trajectories of all cubes are recorded. The observation counterfactuals are made by changing the horizontal location of the top cube. The task is to predict whether the tower will collapse or not, and estimate the corresponding trajectories of cubes. ComPhy (Chen et al. 2022) considers videos of multi-object system under covariate counterfactual scenario. The dynamical system consists of the continuous dynamics which are motions of objects and the instantaneous events which are potential collisions between objects. The goal is to predict the behaviour of objects if their mass and charge are changed. An application of Type III counterfactual is the video question answering task. CLEVRER (Yi et al. 2020) uses the same system as that of ComPhy but queries about what would happen if one of objects is removed. Furthermore, CRAFT (Ates et al. 2020) introduces static objects — ramp, platform, basket, wall and ground into the system, and extends the interactive events to include entering basket.

To simulate the procedure of human counterfactual reasoning, the proposed problem formulation supports to take factual observation as demonstration and differentiate the initial conditions of factual and counterfactual series in three constrained extents. Now the important question is how to realize the imagination world into these series. Inspired by (Russell and Norvig 2020) which treats AI system as dynamic agents, we use multi-body rigid dynamical system as the realization. It consists of multiple objects with continuous- and discrete-time dynamical processes analogous to individual evolution and interactive events in the real world. The dynamical system can be illustrated by factor graph (e.g., Figure 2). There are two types of nodes representing observational objects and latent covariates. An edge links two objects if there is an interaction between them. As shown in Figure 2, observation nodes \( (X_t,s) \) change their states separately in Markov processes depending on the previous states and unobservable covariates \( (U_t,s) \). Type I to III counterfactual happens when an intervention is made on observation nodes \( (D_h) \) and covariate nodes \( (D_{III}) \) and edges \( (D_{II}) \) respectively. Type III intervention is to remove one object. It can be realized as cutting off all edges linked to the object.

For the proposed counterfactual dynamic forecasting on multi-body systems problem, we design an intuitive solution. The continuous dynamics are modelled by a Lagrangian network (Finzi, Wang, and Wilson 2020) and instantaneous interactions are modelled by a neural contact layer (Zhong, Dey, and Chakraborty 2021). They are combined as a forecasting module to be trained end-to-end. Two such forecasting modules are trained simultaneously: one for factual series; and the other for counterfactual series. The two modules share all parameters except the covariate neurons to be intervened. Moreover, a component is installed to capture the difference between two series and coordinate them at each time step. As the problem setting is under a supervised fashion, the forecasting performance mainly benefits from using the factual series as additional information. However, successful solutions for this task will provide two new capabilities for AI agents: learning to generalize for a limited different scenario; avoiding potential risks for safe planning by estimating the counterfactual outcomes.

The two contributions of this paper are:

- Introduced a new formulation for counterfactual reasoning which provides a quantitative definition of counterfactual conditions, and formulates the task as dynamics forecasting.
- Proposed an intuitive solution based on deep networks for the new problem statement.

**Related Work**

Counterfactual treatment effect estimation investigates the effect of intervention in terms of the expected difference between outcomes with and without the treatment assignment. For instance in healthcare application, this technique is used to estimate the effect of a drug for a certain patient. (Lu et al. 2020) used a variational auto-encoder to model the causal effects of latent confounders. (Kaddour et al. 2021) extended the intervention from binary to structural, so that more general treatment assignments such as graph, image and text are supported. (Lewis and Syrgkanis 2021) extended the treatment from fixed to time-varying. (De Brouwer, Gonzalez, and Hyland 2022) studied the problem on time series data,
which is close to our problem setting. However, in their task, before each treatment assignment, common historical series for both factual and counterfactual observations are given as inputs.

*Causal inference* is a research direction with a long history, but causal relationship is not the point of focus in this paper. Approaches about manipulating the intervention on latent covariates, graph representation and dynamical data are related. (Yang et al. 2021) targeted on discovering disentangled factors of images under assumption that these factors are causally dependent. Therefore, an intervention can be assigned to a certain factor to generate desired images. (Huang, Sun, and Wang 2021) focused on social event graphs that are varying over time. They proposed a coupled neural ODEs to model the objects’ dynamics, graphs’ dynamics and the mutual influence between each other.

*Neural ODEs and Hamiltonian/Lagrangian networks* model the physical dynamics based on deep neural networks. Neural ODEs (Chen et al. 2018) made a big step towards neural learning of ODEs. The integration solver of initial value problem is adapted to be differentiable so that ODEs can be learned as a network layer end-to-end. Following that, many variants are invented. Hamiltonian (Greydanus, Dzamba, and Yosinski 2019) and Lagrangian networks (Lutter, Ritter, and Peters 2018) take into account the energy conservation of systems in network training. They optimize scalar functions (Hamiltonians and Lagrangians) instead of the differential equations. Furthermore, constrained Hamiltonian/Lagrangian networks (Finzi, Wang, and Wilson 2020) simplified the constraints of systems in generalized coordinates into Cartesian coordinates. However, all above methods are only able to model the continuous dynamics. Learning instantaneous interaction between objects in dynamical systems by neural networks are challenging. A recent work (Zhong, Dey, and Chakraborty 2021) developed a differentiable contact model by using a convex optimization layer, so that the continuous and instantaneous dynamics can be learned together end-to-end.

**The Problem Statement**

To instantiate quantitative counterfactual reasoning, we consider the sequential forecasting task for uncontrolled Markov processes under the supervised learning scheme. The training dataset has $N$ samples, and each sample is a pair of multivariate time series $\langle X(t), X'(t) \rangle : t = 1, \ldots, T$, where $X(t) \in \mathbb{R}^{m\times d}, X'(t) \in \mathbb{R}^{m'\times d}$. The observation vector $X(t)$ at the time point $t$ consists of $m$ independent $d$-dimensional subvectors $X_i(t)$ for $m$ objects. For example, in a 2d 3-body rigid dynamical system, $X(t) \in \mathbb{R}^6$ are the concatenated 2d coordinates of the three bodies $X_1(t), X_2(t), X_3(t) \in \mathbb{R}^2$. $X(t) : t = 1, \ldots, T$ is the factual series. The first slot of $X'(t)$ (i.e., $X'(1)$) is defined as the counterfactual initial in relation to the first slot of factual series, $X(1)$. In the testing phase, given the available observation $\langle X^*(t = 1, \ldots, T), X'^*(1) \rangle$, the goal of the counterfactual dynamics forecasting is to estimate the future trajectory starting from $X^*(1)$, i.e., $X'^*(t = 2, \ldots, T)$. Thus, in each data sample, the factual dynamics $X(t)$ is given as a demonstration, and the target is to predict the counterfactual outcomes specified by $X'(1)$.

The difference between $X(1)$ and $X'(1)$ describes how far the imaginary condition is away from the fact. Based on three possible types of the differences, three respective counterfactual settings are defined and studied in this paper, which are formulated in a unified framework by using deterministic structural causal models (SCM) (Pearl 2009). The task can be represented by a SCM $\mathcal{M} = \langle V, U, F \rangle$ with directed acyclic graph $G$:

- $X_t = \{X_i(t), X_j(t+1)\}_{i=1}^m$ are the observational variables (nodes of $G$).
- $U = \{U_i\}_{i=1}^m$ is the set of unobservable covariate vectors (nodes of $G$), one for each object. It is assumed that the covariates of any one object are independent from those of others, i.e., $U_i \perp U_j \forall i, j$.
- $E = \{E_{ij}\}_{i,j=1; i \neq j}^{m}$ is the set of unobservable contact covariates (associated to the edges of $G$), where $E_{i,j}$ is related to both object $i$ and $j$, e.g., friction coefficients of contacts between object $i$ and $j$.
- $F = \{f_i : \{X_i(t+1) \mapsto X_i(t+1)\}\}$ is the set of “structural equations” (incoming edges to node $X_i(t+1)$ in $G$), where $Pa(X_i(t+1))$ are parents of $X_i(t+1)$ in $G$.

The graphic model is shown in Figure 2a, where $X_i(t)$ can be the coordinates of an object in multi-body system. Due to the Markov property, $X_i(t+1)$ depends on $X_i(t)$ but not further previous time points. Possible interactions between objects in the system lead to potential connections.
between nodes, which are drawn as dotted arrows in the figure. All $X_i(t), X_i(t+1)$ variables are observable, which are displayed in grey colour. Unobservable covariates $U_i$ (e.g., mass) and $E_{ij}$ (e.g., coefficients of friction) determining the dynamics of $X_i(t)$ are assumed to be fixed over time.

**Definition 1 Observation counterfactuals (Type I)** An intervention $D_j$ (as shown in Figure 2b) is given to one or more observable variables $X_i(t)\ at\ t = 1$, so that $\Delta X_i = X_i'(1) - X_i(1) \neq 0$ and the number of objects $m = m'$. It should be pointed out that only the treatment on $t = 1$ is considered, but not on every time $t$ in the graph model of Figure 2b.

**Intuitive Example 1** This Type I counterfactuals respect to human’s imagination about the observations, for example, to answer the question: “What would happen if the cup is moved closer to the boundary of the table?” or “To what extent, the cup will not drop off if it is placed close to the boundary of the table?”

**Definition 2 Covariate counterfactuals (Type II)** An intervention $D_m$ is given to one of the unobservable variables $U_i$, i.e., $\Delta U_i = U'_i - U_i \neq 0$. The changes of latent factors $U$ will affect the dynamical behaviours of objects.

**Intuitive Example 2** An example question is “Could we avoid sliding if the plastic was replaced by a blanket?”

**Definition 3 Subtractive counterfactuals (Type III)** The intervention $D_m$ in Figure 2b is to remove one of objects in the system, i.e., $m' = m - 1$. This treatment is realized by masking out all edges linked into and out from the object $i$.

**Intuitive Example 3** Imagine that “If the car had not swerved and hit the wall, would the passenger have been injured?”

**Algorithm**

The generative relationship $X_i(t+1)\mid X_i(t), U_i, X_j(t), E_{ij}$ in Figure 2 is represented by the structural equation $f_i : U_i \times E_{ij} \times Pa(X_i(t+1)) \rightarrow X_i(t+1)$}. This paper uses ODEs to realize $f_i$. Following the common practice (Greydanus, Dzamba, and Yosinski 2019) (Lutter, Ritter, and Peters 2018), in addition to object state $p$, first-order derivative $v := \dot{p}$ is also included as observation, i.e., $X(t) = \begin{pmatrix} p(t) \\ v(t) \end{pmatrix}$. For instance, in the system of Figure 1, $p(t)$ is the object location in $X$-Y plane, and $v(t)$ is the velocity vector in $X$-$Y$ directions.

Let $h_0$ be the differential equation of continuous-time dynamics implemented by a neural network with parameter $\theta$. Eq. 1 defines the ODEs describing the continuous-time dynamics of the system.

$$\begin{align*}
\begin{bmatrix} \hat{p}(t+1) \\ \hat{v}(t+1) \end{bmatrix} &= \begin{bmatrix} p(t) \\ v(t) \end{bmatrix} + \int_t^{t+1} h_0(p(t), v(t), U(t))dt \\
\end{align*}
$$

where $p(t), v(t) \in \mathbb{R}^{m \times d}$ are concatenated observations of all objects $i = 1, \ldots, m$ at the time step $t$, and $\hat{p}(t+1), \hat{v}(t+1)$ are estimated states of the next time step. $U$ is learnable parameter representing the invisible covariates.

Before evolving to the next time step $t + 1$, a differentiable contact layer $\text{contact}()$ (Zhong, Dey, and Chakraborty 2021) is embedded to update $v$ as discrete-time dynamics:

$$\begin{align*}
\hat{v}(t+1) &= \hat{v}(t+1) + \text{contact}(\hat{p}(t+1), \hat{v}(t+1), E)
\end{align*}
$$

where $E$ is a matrix with learnable elements $E_{ij}$ as the contact covariates. $\text{contact}()$ function detects possible interactions based on $\hat{p}(t+1)$ first. If there is no interaction, it returns zero. Otherwise, it computes the velocity changes before and after the interaction. Denoting $X(t+1) := \begin{pmatrix} \hat{p}(t+1) \\ \hat{v}(t+1) \end{pmatrix}$, the composite function of Eq. 1 and Eq. 2 is a vector function $f : U \times E \times X(t) \rightarrow \hat{X}(t+1)$ where each $f_i(U_i, E_i, X_i(t))$ is the structural equation for object $i$.

The basic assumption in counterfactual reasoning is that the factual and counterfactual evolution obey the same rules. Therefore, in the realized dynamical system, $X(t)$ and $\hat{X'(t)}$ follow the same ODEs. From the learning perspective, the only difference between these two series is that $X(t) : t = 2, \ldots, T$ are given as inputs while $X'(t) : t = 2, \ldots, T$ are values to be predicted. Therefore, for $X(t), f' : U' \times E' \times \hat{X}'(t) \rightarrow \hat{X}'(t+1)$ takes the previous step estimation $\hat{X}(t)$ as input and is composed by the two steps in Eq. 3:

$$\begin{align*}
\begin{bmatrix} \hat{p}'(t+1) \\ \hat{v}'(t+1) \end{bmatrix} &= \begin{bmatrix} \hat{p}'(t) \\ \hat{v}'(t) \end{bmatrix} + \int_t^{t+1} h_0(\hat{p}'(t), \hat{v}'(t), U(t))dt \\
\hat{v}'(t+1) &= \hat{v}'(t+1) + \text{contact}(\hat{p}'(t+1), \hat{v}'(t+1), E')
\end{align*}
$$

where $\hat{p}'(t) = p'(t)$ and $\hat{v}'(t) = v'(t)$.

To benefit from the demonstration of $X(t)$ during training, the evolution of difference between $X(t)$ and $\hat{X}'(t)$ is modelled by Eq. 4:

$$\begin{align*}
\begin{bmatrix} \hat{p}'_\Delta(t+1) \\ \hat{v}'_\Delta(t+1) \end{bmatrix} &= \begin{bmatrix} p(t+1) \\ v(t+1) \end{bmatrix} + \int_t^{t+1} g_\phi(\hat{p}'(t) - p(t), \hat{v}'_\Delta(t) - v(t), U(t), U'(t))dt \\
\hat{v}'_\Delta(t+1) &= \hat{v}'_\Delta(t+1) + \text{contact}(\hat{p}'(t+1), \hat{v}'_\Delta(t+1), E')
\end{align*}
$$

where $g_\phi$ is a neural network and $\hat{p}'_\Delta(t) = p'(t), \hat{v}'_\Delta(t) = v'(t)$. In a simple case, if $g_\phi(\hat{p}'(t) - p(t), \hat{v}'_\Delta(t) - v(t), U, U') := h_0(\hat{p}'(t), \hat{v}'_\Delta(t), U(t), U(t))$, for interaction free periods, i.e., $\hat{v}'_\Delta(t) = \hat{v}_\Delta(t), Eq. 4$ can be rewritten as $\hat{X}'_\Delta(t+1) = X(t+1) - X(t)$.
Algorithm 1: Counterfactual Dynamics Forecasting

Input: \( \{(X(t), X'(t))\}, (X^*(t), X^*(t+1)) : t = 1, \ldots, T \)
Output: \( \tilde{X}^*(t) : t = 2, \ldots, T \)

1: Create learnable parameter \( U, E \).
2: if Type I then
3:   Set \( U = U, E' = E \)
4: else if Type II then
5:   Create learnable parameter \( U'; \) Set \( E' = E \).
6: else
7:   Create learnable parameter \( U' \);
8:   \( X'(t) = \text{mask}(X'(t)); E' = \text{mask}(E) \)
9: end if
10: Initialize network parameter \( \theta, \phi \) for \( h_\theta \) and \( g_\phi \).
11: while epoch < MAX do
12:    while \( t < T \) do
13:       Compute \( \tilde{X}(t+1) \) by Eq. 1 and Eq. 2.
14:       Compute \( \tilde{X}'(t+1) \) by Eq. 3.
15:    end while
16:    Compute \( L \) by Eq. 5.
17:    Back propagate and update \( \theta, \phi, U, U', E \) and \( E' \).
18: end while
19: return \( \tilde{X}^*(t) = f'(\tilde{X}^*(t-1), U', E') \).

\[
\int_t^{t+1} h_\theta(X(t), U)dt + \tilde{X}'(t) = \int_t^{t+1} h_\theta(X'(t), U')dt
\]
where \( \tilde{X}'(t) = \left( \begin{array}{c} \tilde{p}'(t) \\ \tilde{v}'(t) \end{array} \right) \). This adds an additional supervision at each time \( t \) as \( X(t), X(t+1) \) are known inputs.

The final estimations on \( X(t) \) and \( X'(t) \) are optimized by the mean absolute error:

\[
L = \sum_{(X,X') \in D} \sum_{t=2}^{T} ||\tilde{X}(t) - X(t)||_1 + ||\tilde{X}'(t) - X'(t)||_1 + ||\tilde{X}'(t) - X'(t)||_1 = \sum_{(X,X') \in D} \sum_{t=2}^{T} \frac{1}{2} \left( ||\tilde{X}(t) - X(t)||_1 + ||\tilde{X}'(t) - X'(t)||_1 + ||\tilde{X}'(t) - X'(t)||_1 \right)
\]

where \( || \cdot ||_1 \) is \( l_1 \) norm.
4a). The state $X(t)$ consists of 2d locations and velocities of all balls at time $t$, i.e., $X(t) \in \mathbb{R}^{5 \times 4}$. Type I counterfactuals are drawn from different initial states of all balls. Therefore, the samples $\langle X(t), X'(t) \rangle$ are made of the pair of trajectories as Figure 4a ($X(t)$) and Figure 4b ($X'(t)$). In Type II counterfactuals, the mass of one object $U_i$ is changed. The initial observation values $X_0(t)$ are randomly sampled differently from $X(t)$ to avoid that if no collision occurs in the testing phase, the predictions $X_0(t) = 2, \ldots, T)$, which are known as inputs. In Type III counterfactuals, one ball is removed from the system, e.g., $X(t)$ as shown in Figure 4a and $X'(t)$ as shown in Figure 4d are paired as an instance. The dataset for Chained Pendulums systems is constructed by the same strategy. For each dataset, 800, 100 and 100 pairs of trajectories are used as the training, validation and testing set respectively following (Zhong, Dey, and Chakraborty 2021).

### Evaluation Metrics

Root mean squared error (RMSE) and mean absolute percentage error (MAPE) are used as performance indexes in the experiments. The error is computed over the 50 forecasting time steps as well as on the whole predicted trajectories. Eq.6 lists their definitions, where $\hat{X}'(t)_i$ is the predicted value for $i$-dimension of $X'(t)$.

#### RMSE:

$$\text{RMSE}_t = \frac{1}{|D|} \sum_{X' \in D} \frac{1}{m \times d} \sum_{i=1}^{m \times d} \left( X'(t)_i - \hat{X}'(t)_i \right)^2$$

#### MAPE:

$$\text{MAPE}_t = \frac{1}{|D|} \sum_{X' \in D} \frac{1}{m \times d} \sum_{i=1}^{m \times d} \left| \frac{X'(t)_i - \hat{X}'(t)_i}{X'(t)_i} \right|$$

### Baselines and Implementation

Three forecasting baselines are compared:

- **CLNNwC** (Zhong, Dey, and Chakraborty 2021) models physical systems by a constrained Lagrangian neural network (LNN) and a differentiable contact layer. The former tackles the continuous evolution, the latter handles instantaneous interaction. They are trained end-to-end.
in Figures 6a and 6b. The points, the error of MLP-CLNN goes out of scope as shown in Figure 11. Therefore, it can be observed that at some pendulum strings, which will dramatically affect future constraints. Therefore, the proposed method outperforms all baselines.

**Results for Covariate Counterfactual**

The Type II counterfactuals make changes on latent variables to affect the dynamics. In the experiments, we only change the mass of one object, and keep other properties are remained same. The unobservable covariates in the counterfactuals are learned by supervising both factual and counterfactual dynamics. Figure 7 and Figure 10 show the performance comparison. It is observed that the proposed method produces several obvious peak errors, but in the earlier time steps the curves are smoother than those of CLNNwC. The benefit comes from one-step supervision on $X(t)$ prediction. To estimate $X(t + 1)$ by $H_0(\cdot)$ on each step $t$, the input is the ground truth of $X(t)$ and whether there is interaction in $X(t)$ at this time step $t$ is known, which enhances the learning of the continuous part of the dynamics.

**Conclusion and Future Work**

Counterfactual imagination is a powerful reasoning capability. This paper made effort to formulate it in quantitative settings so that AI algorithms can be applied and evaluated. Based on deep networks, we proposed an approach to solve the task in a dynamical forecasting problem and the effectiveness is demonstrated in the experiments. The limitations of current study is the lack of real-world dataset. Downstream tasks such as video prediction and video QA as discussed in introduction are potential applications.
References


De Brouwer, E.; Gonzalez, J.; and Hyland, S. 2022. Predicting the impact of treatments over time with uncertainty aware neural differential equations. In International Conference on Artificial Intelligence and Statistics, 4705–4722. PMLR.


Huang, Z.; Sun, Y.; and Wang, W. 2021. Coupled Graph ODE for Learning Interacting System Dynamics. In KDD.


