PDE-Driven Spatiotemporal Disentanglement

Jérémie Donà, Jean-Yves Franceschi, Sylvain Lamprier, Patrick Gallinari

PDE-Driven Spatiotemporal Disentanglement

Jérémie Donà[*]
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
jeremie.dona@lip6.fr

Jean-Yves Franceschi[*]
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
jean-yves.franceschi@lip6.fr

Sylvain Lamprier
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
sylvain.lamprier@lip6.fr

Patrick Gallinari
Sorbonne Université, CNRS, LIP6,
F-75005 Paris, France
Criteo AI Lab, Paris, France
patrick.gallinari@lip6.fr

Abstract

A recent line of work addresses the problem of predicting high-dimensional spatiotemporal phenomena by leveraging specific tools from the differential equations theory. Following this direction, we propose in this article a novel and general paradigm for this task based on a resolution method for partial differential equations: the separation of variables. This inspiration allows to introduce a dynamical interpretation of spatiotemporal disentanglement. It induces a simple and principled model based on learning disentangled spatial and temporal representations of a phenomenon to accurately predict future observations. We experimentally demonstrate the performance and broad applicability of our method against prior state-of-the-art models on physical and synthetic video datasets.

1 Introduction

The interest of the machine learning community in physical phenomena has substantially grown for the last few years (Shi et al., 2015; Long et al., 2018; Greydanus et al., 2019). In particular, an increasing amount of works studies the challenging problem of modeling the evolution of dynamical systems, with applications in sensible domains like climate or health science, making the understanding of physical phenomena a key challenge in machine learning. To this end, the community has successfully leveraged the formalism of dynamical systems and their associated differential formulation as powerful tools to specifically design efficient prediction models. In this work, we aim at studying this prediction problem with a principled and general approach, through the prism of Partial Differential Equations (PDEs), with a focus on learning spatiotemporal disentangled representations.

Prediction via spatiotemporal disentanglement was first studied in video prediction works, in order to separate static and dynamic information (Denton & Birodkar, 2017) for prediction and interpretability purposes. Existing models are particularly complex, involving either adversarial losses or variational inference. Furthermore, their reliance on Recurrent Neural Networks (RNNs) hinders their ability to model spatiotemporal phenomena (Yıldız et al., 2019; Ayed et al., 2020; Franceschi et al., 2020). Our proposition addresses these shortcomings with a simplified and improved model by grounding spatiotemporal disentanglement in the PDE formalism.

Spatiotemporal phenomena obey physical laws such as the conservation of energy, that lead to describe the evolution of the system through PDEs. Practical examples include the conservation of energy for

[*]Equal contribution.

Preprint. Under review.

physical systems (Hamilton, 1835), or the equation describing constant illumination in a scene (Horn & Schunck, 1981) for videos that has had a longstanding impact in computer vision with optical flow methods (Finn et al., 2016; Dosovitskiy et al., 2015). We propose to model the evolution of partially observed spatiotemporal phenomena with unknown dynamics by leveraging a formal method for the analytical resolution of PDEs: the functional separation of variables (Miller, 1988). Our framework formulates spatiotemporal disentanglement for prediction as learning a separable solution, where spatial and dynamic information are represented in separate variables. Besides offering a novel interpretation of spatiotemporal disentanglement, it confers simplicity and performance compared to existing methods: disentanglement is achieved through the sole combination of a prediction objective with regularization penalties and the temporal dynamics is defined by a learned Ordinary Differential Equation (ODE). We experimentally demonstrate the applicability, disentanglement capacity, and forecasting performance of the proposed model on various spatiotemporal phenomena involving standard physical processes and synthetic video datasets against prior state-of-the-art models.

2 Related Work

Our contribution deals with two main directions of research: statiotemporal disentanglement and the coupling of neural networks and PDEs.

Spatiotemporal disentanglement. Disentangling factors of variations is an essential representation learning problem (Bengio et al., 2013). Its cardinal formulation for static data has been extensively studied, with state-of-the-art solutions, studied by Locatello et al. (2019), being essentially based on Variational Autoencoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014). As for sequential data, several disentanglement notions have been formulated, ranging from distinguishing objects in a video (Hsieh et al., 2018; van Steenkiste et al., 2018), to separating and modeling multi-scale dynamics (Hsu et al., 2017; Yingzhen & Mandt, 2018).

We focus in this work on the dissociation of the dynamics and visual aspects for spatiotemporal data. Even in this case, dissociation can take multiple forms. Examples in the video generation community include decoupling the foreground and background when generating videos (Vondrick et al., 2016), constructing structured frame representations (Villegas et al., 2017b; Minderer et al., 2019; Liu et al., 2019), extracting physical dynamics (Le Guen & Thome, 2020), or latent modeling of dynamics in a state-space manner (Franceschi et al., 2020). Closer to our work, Denton & Birodkar (2017), Villegas et al. (2017a) and Hsieh et al. (2018) introduced in their video prediction models explicit latent disentanglement of static and dynamic information obtained using adversarial losses (Goodfellow et al., 2014) or VAEs. Disentanglement has also been introduced in more restrictive models relying on data-specific assumptions (Kosiorek et al., 2018; Jaques et al., 2020), and in video generation (Tulyakov et al., 2018). We aim in this work at grounding and improving spatiotemporal disentanglement with more adapted inductive biases, as suggested by Locatello et al. (2019), by introducing a paradigm leveraging the functional separation of variables resolution method of PDEs.

Spatiotemporal prediction and PDE-based neural network models. An increasing number of works combining neural networks and differential equations for spatiotemporal forecasting have been produced for the last few years. Some of them show substantial improvements for the prediction of dynamical systems or videos compared to standard RNNs by defining the dynamics using learned ODEs (Rubanova et al., 2019; Yıldız et al., 2019; Ayed et al., 2020; Le Guen & Thome, 2020), following Chen et al. (2018), or adapting them to stochastic data (Ryder et al., 2018; Li et al., 2020; Franceschi et al., 2020). Most PDE-based spatiotemporal models exploit some prior physical knowledge. It can induce the structure of the prediction function (Brunton et al., 2016; de Avila Belbute-Peres et al., 2018) or specific cost functions, thereby improving model performances. For instance, de Bézenac et al. (2018) shape their prediction function with an advection-diffusion mechanism, and Long et al. (2018, 2019) estimate PDEs and their solutions by learning convolutional filters proven to approximate differential operators. Greydanus et al. (2019), Chen et al. (2020) and Toth et al. (2020) introduce non-regression losses by taking advantage of Hamiltonian mechanics (Hamilton, 1835), while Tompson et al. (2017) and Raissi et al. (2020) combine physically inspired constraints and structural priors for fluid dynamic prediction. Our work deepens this literature by establishing a novel link between a resolution method for PDEs and spatiotemporal disentanglement, and thereby introducing a data-agnostic model leveraging any static information in observed phenomena.

3 Background: Separation of Variables

Analytically or numerically solving high-dimensional PDEs is a difficult problem (Bungartz & Griebel, 2004). Given a decomposition of the solution, e.g., a simple combination of lower-dimensional functions, it consists in reducing the PDE to equivalent simpler differential equations, thus simplifying its resolution.

3.1 Simple Case Study

Let us introduce the idea through a standard application of this technique, with proofs in Appendix A.1, on the one-dimensional heat diffusion problem (Fourier, 1822), e.g., a bar of length L, whose temperature at time t and position x is denoted by $u(x,t)$ and satisfies:

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad u(0,t) = u(L,t) = 0, \qquad u(x,0) = f(x). \tag{1}$$

Suppose that a solution u is product-separable, i.e., it can be decomposed as: $u(x,t) = u_1(x) \cdot u_2(t)$. Combined with Equation (1), it leads to $c^2 u_1''(x)/u_1(x) = u_2'(t)/u_2(t)$. The left and right hand sides of this equation are respectively independent from t and x, thus both sides are constant, and solving both resulting ODEs gives solutions of the form, with $\mu \in \mathbb{R}$ and $n \in \mathbb{N}$:

$$u(x,t) = \mu \sin\left(\frac{n\pi}{L}x\right) \times \exp\left(-\left(\frac{cn\pi}{L}\right)^2 t\right). \tag{2}$$

The superposition principle and the unicity of solutions under smoothness constraints allow then to build the set of solutions of Equation (1) with linear combinations of separable solutions (Le Dret & Lucquin, 2016). Besides this simple example, separation of variables can be more elaborate.

3.2 Functional Separation of Variables

The functional separation of variables (Miller, 1988) generalizes this method. Let u be a function obeying a given arbitrary PDE. The functional variable separation method amounts to finding a parameterization z, a functional U, an entangling function ξ, and representations ϕ and ψ such that:

$$z = \xi\big(\phi(x), \psi(t)\big), \qquad u(x,t) = U(z). \tag{3}$$

Trivial choices $\xi = u$ and identity function as U, ϕ and ψ ensure the validity of this reformulation. Finding suitable ϕ, ψ, U, and ξ with regards to the initial PDE can facilitate its resolution by inducing separate simpler PDEs on ϕ, ψ, and U. General results on the existence of separable solutions have indeed been proven (Miller, 1983), even though their unicity highly depends on the initial problem and the choice of functional separation (Polyanin, 2020). Functional separation of variables finds applications in various physics fields, such as reaction-diffusion with non-linear sources or convection-diffusion (Polyanin, 2019; Polyanin & Zhurov, 2020), Hamiltonian physics (Benenti, 1997), or even general relativity (Kalnins et al., 1992).

As an example, consider a refinement of Equation (1) on u along with the change of variable v :

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \chi \frac{\partial^2 u}{\partial x^2}, \qquad v(x,t) = u(x,t)\mathrm{e}^{-\alpha x}\mathrm{e}^{-\beta t}. \tag{4}$$

A proper choice of constants α and β makes v satisfy Equation (1)'s PDE, which is solvable via separation of variables; see Appendix A.2 for details. Non-linear generalizations of Equations (1) and (4) also find solutions using the functional separation of variables, with for instance:

$$\frac{\partial u}{\partial t} = \nu(u)\frac{\partial^2 u}{\partial x^2} + Q(x,u)\frac{\partial u}{\partial x} + f(x,u), \tag{5}$$

for which Jia et al. (2008) exhibit conditions for the existence of solutions to this equation decomposed as follows:

$$z = \phi(x) + \psi(t), \qquad u(x,t) = U(z). \tag{6}$$

The functional decomposition of Equation (3) generalizes the separability defined in Section 3.1, as addition and product separability are recoverable by setting, respectively, $U = \mathrm{id}$ and $U = \exp$.

We see reparameterizations such as Equation (6) as changes of coordinates inducing a natural spatiotemporal disentanglement, and introduce in the following a relaxation of this general method.

4 Proposed Method

We propose to model spatiotemporal phenomena using the functional variable separation formalism. We first describe our notations and then derive a principled model and constraints from this method.

4.1 Problem Formulation Through Separation of Variables

We consider a distribution $\mathcal{P}$ of observed spatio-temporal trajectories and corresponding observation samples $v = (v_{t_0}, v_{t_0+\Delta t}, \dots, v_{t_1})$, with $v_t \in \mathcal{V} \subseteq \mathbb{R}^m$ and $t_1 = t_0 + \nu\Delta t$. Each sequence $v \sim \mathcal{P}$ corresponds to an observation of a dynamical phenomenon, assumed to be described by a hidden functional u_v (also denoted by u for the sake of simplicity) of space coordinates $x \in \mathcal{X} \subseteq \mathbb{R}^s$ and time t that characterizes the trajectories. More precisely, u_v describes an unobserved continuous dynamics and v corresponds to instantaneous discrete spatial measurements associated to this dynamics. Therefore, we consider that v_t results from a time-independent function ζ of the mapping $u_v(\cdot, t)$. For example, v might consist in temperatures measured at some points of the sea surface, while u_v would be the circulation ocean model. v provides a partial information about u_v and is a function (e.g. projection) of the full dynamics. We seek to learn a model which, when conditioned on prior observations, can predict future observations.

To this end, we posit that the state u of each observed trajectory v is driven by a hidden common PDE, shared among all trajectories; we discuss this assumption in details in Appendix C.1. Learning such PDE and its solutions would then allow to model observed trajectories v. We propose to do so by relying on the functional separation of variables of Equation (3), in order to leverage a potential separability of the hidden PDE. Therefore, analogously to Equation (3), we propose to formulate the problem as learning observation-constrained ϕ, ψ and U, as well as ξ and ζ, such that:

$$z = \xi\big(\phi(x), \psi(t)\big), \qquad u(x,t) = U(z), \qquad v_t = \zeta\big(u(\cdot, t)\big), \tag{7}$$

with ϕ and ψ allowing to disentangle the prediction problem. As with the formalism of the functional separation of variables, this amounts to learning a spatial ODE on ϕ, a temporal ODE on ψ, and a PDE on U, as well as their respective solutions.

4.2 Fundamental Limits and Relaxation

However, directly learning u is a restrictive choice, as it depends on the system coordinates. Indeed, learning explicit PDE solutions taking as input space and time coordinates, like Sirignano & Spiliopoulos (2018) and Raissi (2018), has major drawbacks: it requires to deal with the spatial coordinate system and to have prior knowledge about the involved PDEs, which may be unknown for complex data such as in climate modeling. We choose not to make such strong assumptions in order to maintain the generality of the proposed approach.

We overcome these issues by, instead, encoding the unknown spatial coordinate system in a spatial representation, and thus implicitly learn u by directly modeling sequences of observations thanks to representation learning. Indeed, Equation (7) induces that these spatial coordinates, hence the explicit resolution of PDEs on u or U, can be ignored, as it amounts to learning ϕ, ψ and D such that:

$$v_t = (\zeta \circ U \circ \xi)\big(\phi(\cdot), \psi(t)\big) = D\big(\phi, \psi(t)\big). \tag{8}$$

In order to manipulate functionals ϕ and ψ in practice, we respectively introduce learnable time-invariant and time-dependent representations of ϕ and ψ, denoted by S and T, such that:

$$\phi \equiv S \in \mathcal{S} \subseteq \mathbb{R}^d, \qquad \psi \equiv T\colon t \mapsto T_t \in \mathcal{T} \subseteq \mathbb{R}^p, \tag{9}$$

where the dependence of $\psi \equiv T$ on time t will be modeled using a temporal ODE following the separation of variables, and the function ϕ, and consequently its spatial ODE, are encoded into a vectorial representation S. Besides their separation of variables basis, the purpose of S and T is to capture spatial and motion information of the data. For instance, S could encode static information such as objects appearance, while T typically contains motion variables.

4.3 Parameterization of the Functional Variable Separation

S and T_{t_0}, because of their dependence on v in Equation (9), are inferred from an observation history, or conditioning frames, $V_\tau(t_0)$, where $V_\tau(t) = (v_t, v_{t+\Delta t}, \dots, v_{t+\tau\Delta t})$, using respectively encoder

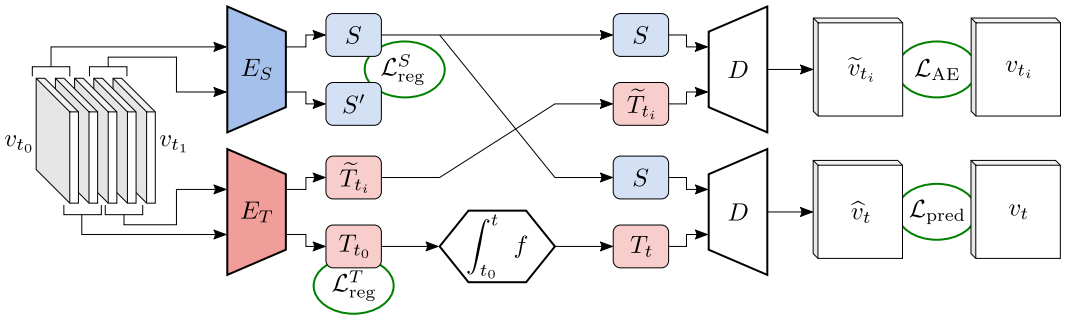

Figure 1: Computational graph of the proposed model. E_S and E_T take contiguous observations as input; time invariance is enforced on S; the evolution of T_t is modeled with an ODE and is constrained to coincide with E_T; T_{t_0} is regularized; forecasting equates to decoding from S and T_t.

networks E_S and E_T. We parameterize D of Equation (8) as a neural network that acts on both S and T_t, and outputs the estimated observation $\widehat{v}_t = D(S, T_t)$. Unless specified otherwise, S and T_t are fed concatenated into D, which then learns the parameterization ξ of their combination.

4.4 Temporal ODE on $\psi \equiv T$

We model the evolution of T_t, thereby the dynamics of our system, with a first-order ODE:

$$\frac{\partial T_t}{\partial t} = f(T_t) \qquad \Leftrightarrow \qquad T_t = T_{t_0} + \int_{t_0}^{t} f(T_{t'})\,\mathrm{d}t' \tag{10}$$

This is in accordance with the separation of variables method that induces an ODE on ψ. Note that the first-order ODE assumption can be taken without loss of generality since any ODE is equivalent to a higher-dimensional first-order ODE. Therefore, since T_t is multi-dimensional, it can model complex interactions between system variables. Following Chen et al. (2018), f is implemented by a neural network and Equation (10) is solved with an ODE resolution scheme. Suppose initial ODE conditions S and T_{t_0} have been computed with E_S and E_T. This leads to the following simple forecasting scheme, enforced by the corresponding regression loss:

$$\widehat{v}_t = D\left(S, T_{t_0} + \int_{t_0}^{t} f(T_{t'})\,\mathrm{d}t'\right), \qquad \mathcal{L}_{\text{pred}} = \frac{1}{\nu+1}\sum_{i=0}^{\nu}\frac{1}{m}\|\widehat{v}_{t_0+i\Delta t} - v_{t_0+i\Delta t}\|_2^2, \tag{11}$$

where $\nu+1$ is the number of observations, and the m is the dimension of the observed variables v.

Equation (11) ensures that the evolution of T is coherent with the observations; we now should enforce its consistency with E_T. Indeed, the dynamics of T_t is modeled by Equation (10), while only its initial condition T_{t_1} is computed with E_T. However, there is no guaranty that T_t, computed via integration, matches $E_T\big(V_\tau(t)\big)$ at any other time t, while they should in principle coincide. We introduce the following autoencoding constraint aiming at mitigating their potential divergence, thereby stabilizing the evolution of T:

$$\mathcal{L}_{\text{AE}} = \frac{1}{m}\left\|D\Big(S, E_T\big(V_\tau(t_0 + i\Delta t)\big)\Big) - v_t\right\|_2^2, \qquad i \sim \mathcal{U}\big([\![0, \nu - \tau]\!]\big). \tag{12}$$

4.5 Spatial ODE on $\phi \equiv S$

As indicated hereinabove, the spatial ODE on ϕ is assumed to be encoded into S. Nonetheless, since S is inferred from an observation history, the time independence property on S is de facto relaxed; thus, we need to explicitly enforce it. Unlike Denton & Birodkar (2017) who penalize the squared difference between two contents representation taken at random times, we adopt a simpler PDE-motivated approach. Time independence implies:

$$\frac{\partial E_S\big(V_\tau(t)\big)}{\partial t} = 0. \tag{13}$$

However, computing this derivative in practice is complex and costly; see Appendix B for more details. Moreover, observation histories may not convey identical spatial information (for example, when an object conceals another for the whole history period); thus, directly minimizing this derivative may hinder performances. Therefore, we relax this constraint thanks to a lower bound on the integral of temporal derivatives of E_S obtained with Cauchy-Schwarz inequality:

$$\int_{t_0}^{t_1-\tau\Delta t} \left\| \frac{\partial E_S\big(V_\tau(t)\big)}{\partial t} \right\|_2^2 \mathrm{d}t \geq \left\| \int_{t_0}^{t_1-\tau\Delta t} \frac{\partial E_S\big(V_\tau(t)\big)}{\partial t} \, \mathrm{d}t \right\|_2^2. \tag{14}$$

Thus, we only minimize the evolution of $E_S\big(V_\tau(t)\big)$ between two distant time steps by penalizing the right-hand side of Equation (14), where d is the dimension of S:

$$\mathcal{L}^S_{\text{reg}} = \frac{1}{d} \Big\| E_S\big(V_\tau(t_0)\big) - E_S\big(V_\tau(t_1-\tau)\big) \Big\|_2^2. \tag{15}$$

4.6 Spatiotemporal Disentanglement

Abstracting the spatial ODE into a generic representation S leads, without additional constraints, to an underconstrained problem where spatiotemporal disentanglement cannot be guaranteed. Indeed, E_S can be set to zero without breaking any prior constraint, because static information is not prevented to be encoded into T. Accordingly, information in S and T needs to be segmented.

Thanks to the design of our model, it suffices to ensure that S and T and disentangled at initial time t_0 for them be to be disentangled at all t. Indeed, the mutual information between two variables is preserved by invertible transformations. Equation (10) is an ODE and f, as a neural network, is Lipschitz-continuous, so $T_t \mapsto T_{t'}$ is invertible. Therefore, disentanglement between S and T_t, characterized by a low mutual information between both variables, is preserved through time; see Appendix C for a detailed discussion. We thus only constrain the information quantity in T_{t_0} by using a Gaussian prior to encourage it to contain only necessary dynamic information:

$$\mathcal{L}^T_{\text{reg}} = \frac{1}{p} \|T_{t_0}\|_2^2 = \frac{1}{p} \Big\| E_T\big(V_\tau(t_0)\big) \Big\|_2^2. \tag{16}$$

4.7 Loss Function

The global loss to be minimized is a linear combination of Equations (11), (12), (15) and (16), as illustrated in Figure 1:

$$\mathcal{L}(v) = \mathbb{E}_{v\sim\mathcal{P}} \Big[\lambda_{\text{pred}} \mathcal{L}_{\text{pred}} + \lambda_{\text{AE}} \cdot \mathcal{L}_{\text{AE}} + \lambda^S_{\text{reg}} \cdot \mathcal{L}^S_{\text{reg}} + \lambda^T_{\text{reg}} \cdot \mathcal{L}^T_{\text{reg}} \Big]. \tag{17}$$

In the following, we conventionally set $\Delta t = 1$. Note that the presented approach could be generalized to irregularly sampled observation times thanks to the dedicated literature (Rubanova et al., 2019), but this is out of the scope of this paper.

5 Experiments

We describe in this section the main experimental results of our model on three physical datasets and a synthetic video prediction dataset, briefly presented in this section and in more details in Appendix D.[2] We demonstrate the relevance of our model with ablation studies, and its performance by comparing it with more complex state-of-the-art models. We refer to Appendix F for more experiments and prediction examples, and to Appendix E for training details.

5.1 Physical Datasets: Wave Equation and Sea Surface Temperature

We first investigate two toy dynamical systems and a real-world dataset in order to show the advantage of PDE-driven spatiotemporal disentanglement for forecasting.

We first lean on the wave equation, occurring for example in acoustic or electromagnetism, with source term like Saha et al. (2020), to produce the toy dataset WaveEq consisting in 64×64

[2]Code is available at `https://github.com/JeremDona/spatiotemporal_variable_separation`.

Table 1: Forecast mean squared errors on WaveEq-100, WaveEq, and SST for our model and PKnl with respect to indicated prediction horizons. Bold scores indicate the best performing method.

Models	WaveEq-100	WaveEq	SST	
	$t+40$	$t+40$	$t+6$	$t+10$
PKnl	—	—	1.28	2.03
Ours	$\mathbf{1.52 \times 10^{-5}}$	$\mathbf{4.78 \times 10^{-5}}$	**1.17**	**1.79**
Ours (without S)	1.56×10^{-4}	1.99×10^{-4}	1.60	2.38

Table 2: PSNR and SSIM scores of DrNet, DDPAE and our model on the Moving MNIST dataset for prediction and content swap tasks. Bold scores indicate the best performing method.

Models	Pred. $(t+10)$		Pred. $(t+95)$		Swap $(t+10)$		Swap $(t+95)$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DrNet	14.94	0.6596	12.91	0.5379	14.12	0.6206	12.80	0.5306
DDPAE	21.17	0.8814	13.56	0.6446	**18.44**	0.8256	13.25	0.6378
Ours	**21.74**	**0.9094**	**17.22**	**0.7867**	18.30	**0.8343**	**16.21**	**0.7600**
Ours (without S)	Failed: underflow after a few iterations							
Ours ($\lambda_{\text{AE}} = 0$)	21.51	0.9065	15.17	0.7054	18.01	0.8274	14.52	0.6884
Ours ($\lambda^S_{\text{reg}} = 0$)	15.69	0.6670	13.77	0.6770	13.76	0.5392	13.56	0.6631
Ours ($\lambda^T_{\text{reg}} = 0$)	15.06	0.7030	13.96	0.7218	14.64	0.6907	13.92	0.7208

normalized images of the physical process. We additionally build the WaveEq-100 dataset by extracting 100 pixels, chosen uniformly at random and shared among sequences, from WaveEq frames; this experimental setting can be thought of as measurements from sensors partially observing the phenomenon. In both cases, $\tau = 4$ and $\nu = 24$. Our model is also tested on the real-world dataset SST, derived from the data assimilation engine NEMO (Madec, 2008) and introduced by de Bézenac et al. (2018), consisting in 64×64 frames showing the evolution of the sea surface temperature. Modeling its evolution is particularly challenging as its dynamic is highly non-linear, chaotic, and involves several unobserved quantities (e.g., forcing terms). In this case, $\tau = 3$ and $\nu = 9$.

We compare our model on these three datasets to a version of this model with S removed and integrated into T, thus also removing $\mathcal{L}^S_{\text{reg}}$ and $\mathcal{L}^T_{\text{reg}}$. We additionally include PKnl (de Bézenac et al., 2018), a model specifically designed for SST, in the comparison. Results are compiled in Table 1 for different forecast horizons, and an example of prediction is depicted in Figure 2.

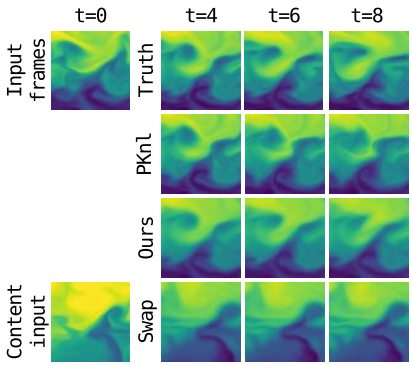

Figure 2: Example of predictions of compared models on SST.

On these three datasets, our model produces more accurate long-term prediction with S than without it. This indicates that learning an invariant component facilitates training and improves generalization on physical datasets. The influence of S can be observed on Figure 2 (swap row) where the S of a given sequence is replaced by another one extracted from another sequence, changing the aspect of the prediction. We provide in Appendix F further samples showing the influence of S in the prediction. Even though there is no evidence of separability in SST, our algorithm trained with a time-invariant component takes advantage of this feature on both tested forecast horizons. Indeed, it outperforms PKnl despite the data-specific structure of the latter, whereas removing the static component decreases performances below PKnl.

5.2 A Synthetic Video Dataset: Moving MNIST

We also assess the prediction and disentanglement performances of our model on the Moving MNIST dataset (Srivastava et al., 2015) involving MNIST digits (LeCun et al., 1998) bouncing over 64×64

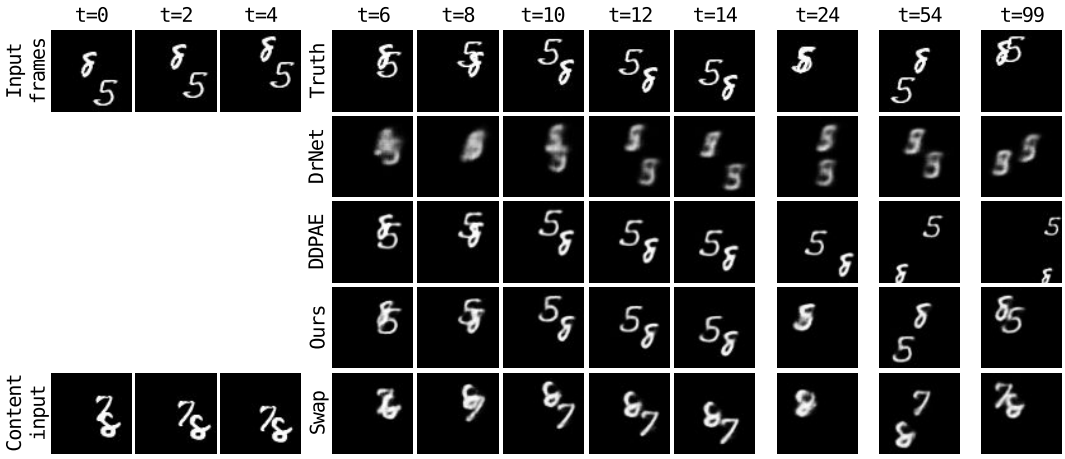

Figure 3: Example of predictions of compared models on Moving MNIST.

frame borders, with $\tau = 4$ and $\nu = 14$. We perform a full ablation study of our model, and compare it to DrNet (Denton & Birodkar, 2017) and DDPAE (Hsieh et al., 2018), which are state-of-the-art spatiotemporal disentangled prediction methods on Moving MNIST leveraging no restrictive data-specific priors. Note that DrNet and DDPAE use powerful machine learning techniques, with the former based on adversarial losses and the latter on complex VAEs.

Results reported in Table 2 and illustrated in Figure 3 correspond to two tasks: prediction and disentanglement, at both short and long-term horizons. Disentanglement is evaluated via content swapping, which consists in replacing the content representation of a sequence by the one of another sequence, which should result for a perfectly disentangled model in swapping digits of both sequences. This is done by taking advantage of the synthetic nature of this dataset that allows us to implement the ground truth content swap and compare it to the generated swaps of the model. Performances are assessed by comparing to the ground truth using standard metrics (Denton & Fergus, 2018) Peak Signal-to-Noise Ratio (PSNR, *higher is better*) and Structured Similarity (SSIM, *higher is better*).

Both qualitative and quantitative results show the advantage of our model against all baselines, despite its simplicity compared to DrNet and DDPAE. DDPAE produces accurate predictions on a short horizon but does not extrapolate well to long-term digits movements, with altered shapes of digits. DrNet fails to even generate sharp digits. Because the content variable is fixed all along the forecast, this shows that both baselines have difficulties separating content and motion. Our model instead presents consistent samples at $t + 95$, even in the content swap setting, showing that it better separates motion from content than prior methods. Accordingly, our model significantly outperforms both of these baselines in terms of prediction and disentanglement, especially at a long-term horizon.

Ablation studies confirm that this advantage is due to the constraints inspired by the separation of variables. Indeed, the model fails to train correctly without S due to numerical instabilities, and removing any non-forecasting constraint of the training loss substantially reduces performances. In particular, the invariance loss on the static component and the regularization of initial condition T_{t_0} are essential, as their absence hinders both prediction and disentanglement. Removing the auto-encoding constraints affects the prediction accuracy in a minor measure, still allowing state-of-the art performances compared to other baselines. This observation shows that it suffices to implement an ℓ_2 constraint on the first time step of the sequence only to enforce disentanglement, as described in Section 4.6. Nonetheless, the auto-encoding constraint significantly strengthens our performances at a long-term horizon, confirming its benefits to the stabilization of dynamics.

6 Conclusion

We introduce a novel method for spatiotemporal prediction inspired by the separation of variables PDE resolution technique, involving only time invariance and regression constraints. This inspiration induces simple constraints ensuring the separation of spatial and temporal information. We experi-

mentally demonstrate the benefits of the proposed model, which, despite its simplicity, outperforms prior state-of-the-art methods on physical and synthetic video datasets. We believe that this work, which provides a dynamical interpretation of spatiotemporal disentanglement and implements it in a simple method, could serve as the basis of more complex models further leveraging the PDE formalism. Another direction for future work could be extending the model with more involved tools such as VAEs to improve its performances, or adapt it to the prediction of natural stochastic videos (Denton & Fergus, 2018).

Broader Impact

Our work introduces a spatiotemporal disentanglement method for forecasting. Besides theoretical motivations, our method was designed in order to improve interpretability in machine learning prediction systems. Indeed, when using deep neural networks as predictive algorithms, exploring latent representations and evaluating their impact on the output is challenging, due to the complex geometry of the latent space. We believe that our work is a step forward in this direction. Moreover, our method provides a framework to automatically learn invariance in physical systems. The modeling of physical systems using neural networks gains momentum in the machine learning community and has potential applications in climatology and evaluation of climate change, as soon as the work results from the cooperation of experts form both fields.

However, the choice of studied datasets to learn spatiotemporal disentanglement should be carefully considered under its potential consequences. Even though our experiments only consider synthetic and physical data, separating motion from content in videos could lead to potential manipulations made possible by such disentanglement, with for instance deepfakes (Citron & Chesney, 2018), raising the broader question of the threats of machine learning technologies to privacy and society. While new advances emerge for their detection (Dolhansky et al., 2019; Güera & Delp, 2018), only little information on their implementation in mass media platforms is available, further increasing the responsibility of the experimenter in his choice of disentanglement studies.

Acknowledgments

We would like to thank all members of the MLIA team from the LIP6 laboratory of Sorbonne Université for helpful discussions and comments.

We acknowledge financial support from the LOCUST ANR project (ANR-15-CE23-0027) and the European Union's Horizon 2020 research and innovation programme under grant agreement 825619 (AI4EU). This study has been conducted using E.U. Copernicus Marine Service Information. This work was granted access to the HPC resources of IDRIS under the allocation 2020-AD011011360 made by GENCI (Grand Equipement National de Calcul Intensif).

References

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

Ayed, I., de Bézenac, E., Pajot, A., and Gallinari, P. Learning the spatio-temporal dynamics of physical processes from partial observations. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3232–3236, 2020.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 573–582, Long Beach, California, USA, June 2019. PMLR.

Benenti, S. Intrinsic characterization of the variable separation in the Hamilton-Jacobi equation. *Journal of Mathematical Physics*, 38(12):6578–6602, 1997.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, August 2013.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Bungartz, H.-J. and Griebel, M. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 6571–6583. Curran Associates, Inc., 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2172–2180, 2016.

Chen, Z., Zhang, J., Arjovsky, M., and Bottou, L. Symplectic recurrent neural networks. In *International Conference on Learning Representations*, 2020.

Citron, D. K. and Chesney, R. Deep fakes: A looming challenge for privacy, democracy, and national security. *107 California Law Review 1753 (2019); U of Texas Law, Public Law Research Paper No. 692; U of Maryland Legal Studies Research Paper No. 2018-21.*, 2018.

de Avila Belbute-Peres, F., Smith, K. A., Allen, K. R., Tenenbaum, J. B., and Kolter, J. Z. End-to-end differentiable physics for learning and control. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7178–7189. Curran Associates, Inc., 2018.

de Bézenac, E., Pajot, A., and Gallinari, P. Deep learning for physical processes: Incorporating prior scientific knowledge. In *International Conference on Learning Representations*, 2018.

Denton, E. and Birodkar, V. Unsupervised learning of disentangled representations from video. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4414–4423. Curran Associates, Inc., 2017.

Denton, E. and Fergus, R. Stochastic video generation with a learned prior. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1174–1183, Stockholmsmässan, Stockholm, Sweden, July 2018. PMLR.

Dolhansky, B., Howes, R., Pflaum, B., Baram, N., and Ferrer, C. C. The deepfake detection challenge (DFDC) preview dataset. *arXiv preprint arXiv:1910.08854*, 2019.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., and Brox, T. FlowNet: Learning optical flow with convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766, December 2015.

Finn, C., Goodfellow, I., and Levine, S. Unsupervised learning for physical interaction through video prediction. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 64–72. Curran Associates, Inc., 2016.

Fourier, J. B. J. *Théorie analytique de la chaleur*. Didot, Firmin, 1822.

Franceschi, J.-Y., Delasalles, E., Chen, M., Lamprier, S., and Gallinari, P. Stochastic latent residual video prediction. *arXiv preprint arXiv:2002.09219*, 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 15379–15389. Curran Associates, Inc., 2019.

Güera, D. and Delp, E. J. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, 2018.

Haber, E. and Ruthotto, L. Stable architectures for deep neural networks. *Inverse Problems*, 34(1): 014004, dec 2017. doi: 10.1088/1361-6420/aa9a90.

Hairer, E., Nørsett, S. P., and Wanner, G. *Solving Ordinary Differential Equations I: Nonstiff Problems*, chapter Runge-Kutta and Extrapolation Methods, pp. 129–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

Hamilton, W. R. Second essay on a general method in dynamics. *Philosophical Transactions of the Royal Society*, 125:95–144, 1835.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, June 2016.

Horn, B. K. P. and Schunck, B. G. Determining optical flow. *Artificial Intelligence*, 17(1–3):185–203, August 1981.

Hsieh, J.-T., Liu, B., Huang, D.-A., Fei-Fei, L., and Niebles, J. C. Learning to decompose and disentangle representations for video prediction. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 517–526. Curran Associates, Inc., 2018.

Hsu, W.-N., Zhang, Y., and Glass, J. Unsupervised learning of disentangled and interpretable representations from sequential data. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1878–1889. Curran Associates, Inc., 2017.

Jaques, M., Burke, M., and Hospedales, T. Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video. In *International Conference on Learning Representations*, 2020.

Jia, H., Xu, W., Zhao, X., and Li, Z. Separation of variables and exact solutions to nonlinear diffusion equations with x-dependent convection and absorption. *Journal of Mathematical Analysis and Applications*, 339(2):982–995, March 2008.

Kalnins, E. G., Miller, Jr., W., and Williams, G. C. Recent advances in the use of separation of variables methods in general relativity. *Philosophical Transactions: Physical Sciences and Engineering*, 340(1658):337–352, 1992.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kosiorek, A. R., Kim, H., Teh, Y. W., and Posner, I. Sequential attend, infer, repeat: Generative modelling of moving objects. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8606–8616. Curran Associates, Inc., 2018.

Kraskov, A., Stögbauer, H., and Grassberger, P. Estimating mutual information. *Physical Review E*, 69:066138, June 2004.

Kutta, M. W. Beitrag zur näherungweisen Integration totaler Differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 45:435–453, 1901.

Le Dret, H. and Lucquin, B. *Partial Differential Equations: Modeling, Analysis and Numerical Approximation*, chapter The Heat Equation, pp. 219–251. Springer International Publishing, Cham, 2016.

Le Guen, V. and Thome, N. Disentangling physical dynamics from unknown factors for unsupervised video prediction. *arXiv preprint arXiv:2003.01460*, 2020.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

Li, X., Wong, T.-K. L., Chen, R. T. Q., and Duvenaud, D. Scalable gradients for stochastic differential equations. *arXiv preprint arXiv:2001.01328*, 2020.

Liu, Z., Wu, J., Xu, Z., Sun, C., Murphy, K., Freeman, W. T., and Tenenbaum, J. B. Modeling parts, structure, and system dynamics via predictive learning. In *International Conference on Learning Representations*, 2019.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124, Long Beach, California, USA, June 2019. PMLR.

Long, Z., Lu, Y., Ma, X., and Dong, B. PDE-Net: Learning PDEs from data. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3208–3216, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR.

Long, Z., Lu, Y., and Dong, B. PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

Lu, Y., Zhong, A., Li, Q., and Dong, B. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. *arXiv preprint arXiv:1710.10121*, 2017.

Madec, G. *NEMO ocean engine*. Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27, 2008.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., and Wu, H. Mixed precision training. In *International Conference on Learning Representations*, 2018.

Miller, Jr., W. The technique of variable separation for partial differential equations. In Wolf, K. B. (ed.), *Nonlinear Phenomena*, pp. 184–208, Berlin, Heidelberg, 1983. Springer Berlin Heidelberg.

Miller, Jr., W. Mechanisms for variable separation in partial differential equations and their relationship to group theory. In Levi, D. and Winternitz, P. (eds.), *Symmetries and Nonlinear Phenomena: Proceedings of the International School on Applied Mathematics*, pp. 188–221, Singapore, 1988. World Scientific.

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K., and Lee, H. Unsupervised learning of object structure and dynamics from videos. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 92–102. Curran Associates, Inc., 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8026–8037. Curran Associates, Inc., 2019.

Polyanin, A. D. Functional separable solutions of nonlinear convection–diffusion equations with variable coefficients. *Communications in Nonlinear Science and Numerical Simulation*, 73:379–390, July 2019.

Polyanin, A. D. Functional separation of variables in nonlinear PDEs: General approach, new solutions of diffusion-type equations. *Mathematics*, 8(1):90, 2020.

Polyanin, A. D. and Zhurov, A. I. Separation of variables in PDEs using nonlinear transformations: Applications to reaction–diffusion type equations. *Applied Mathematics Letters*, 100:106055, February 2020.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

Raissi, M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018.

Raissi, M., Yazdani, A., and Karniadakis, G. E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278–1286, Bejing, China, June 2014. PMLR.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing.

Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 5320–5330. Curran Associates, Inc., 2019.

Ryder, T., Golightly, A., McGough, A. S., and Prangle, D. Black-box variational inference for stochastic differential equations. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4423–4432, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR.

Saha, P., Dash, S., and Mukhopadhyay, S. PhICnet: Physics-incorporated convolutional recurrent neural networks for modeling dynamical systems. *arXiv preprint arXiv:2004.06243*, 2020.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28*, pp. 802–810. Curran Associates, Inc., 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Sirignano, J. and Spiliopoulos, K. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

Srivastava, N., Mansimov, E., and Salakhudinov, R. Unsupervised learning of video representations using LSTMs. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 843–852, Lille, France, July 2015. PMLR.

Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. Accelerating Eulerian fluid simulation with convolutional networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3424–3433, International Convention Centre, Sydney, Australia, August 2017. PMLR.

Toth, P., Rezende, D. J., Jaegle, A., Racanière, S., Botev, A., and Higgins, I. Hamiltonian generative networks. In *International Conference on Learning Representations*, 2020.

Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. MoCoGAN: Decomposing motion and content for video generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1526–1535, June 2018.

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.

Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations*, 2017a.

Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. Learning to generate long-term future via hierarchical prediction. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3560–3569, International Convention Centre, Sydney, Australia, August 2017b. PMLR.

Vondrick, C., Pirsiavash, H., and Torralba, A. Generating videos with scene dynamics. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 613–621. Curran Associates, Inc., 2016.

Yingzhen, L. and Mandt, S. Disentangled sequential autoencoder. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5670–5679, Stockholmsmässan, Stockholm Sweden, July 2018. PMLR.

Yıldız, C., Heinonen, M., and Lahdesmaki, H. ODE^{2}VAE: Deep generative second order odes with Bayesian neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 13412–13421. Curran Associates, Inc., 2019.

A Proofs

A.1 Resolution of the Heat Equation

In this section, we succinctly detail a proof for the existence and uniqueness for the solution to the two-dimensional heat equation. It allows to show that product-separable solutions build the entire solution space for this problem, highlighting our interest into the research of separable solutions.

Existence through separation of variables. Consider the heat equation problem:

$$\frac{\partial u}{\partial t} = c^2 \frac{\partial^2 u}{\partial x^2}, \qquad u(0,t) = u(L,t) = 0, \qquad u(x,0) = f(x). \tag{18}$$

Assuming product separability of u with $u(x,t) = u_1(x)u_2(t)$ in Equation (18) gives:

$$c^2 \frac{u_1''(x)}{u_1(x)} = \frac{u_2'(t)}{u_2(t)}. \tag{19}$$

Both sides being independent of each other variables, they are equal to a constant denoted by $-\alpha$. If α is negative, solving the right side of Equation (19) results to non-physical solutions with exponentially increasing temperatures, and imposing border condition of Equation (18) makes this solution collapse to the null trivial solution. Therefore, we consider that $\alpha > 0$.

Both sides of Equation (19) being equal to a constant leads to a second-order ODE on u_1 and a first-order ODE on u_2, giving the solution shapes, with constants A, B and D:

$$\begin{cases} u_1(x) &= A \times \cos\left(\sqrt{\alpha}x\right) + B \sin\left(\sqrt{\alpha}x\right) \\ u_2(t) &= D \times \mathrm{e}^{-\alpha t \times c^2} \end{cases}. \tag{20}$$

Link with initial and boundary conditions Now we link the above equation to the boundary conditions of the problem. Because our separation is multiplicative, we can omit D for non-trivial solutions and set it with loss of generality to 1 (as it only scales the values of A and B).

$u(0,t) = u(L,t) = 0$ and $\forall t > 0, u_1(t) \neq 0$ gives:

$$A = 0, \qquad B \times \mathrm{e}^{-\alpha t \times c^2} \sin\left(\sqrt{\alpha}L\right) = 0, \tag{21}$$

which means that, for non-trivial solution (i.e, $B \neq 0$), we have for a given $n \in \mathbb{N}$: $\sqrt{\alpha} = n\pi/L$. We can finally express our solution to the heat equation without initial conditions as:

$$u(x,t) = B \sin\left(\frac{n\pi}{L}x\right) \times \exp\left(-\left(\frac{cn\pi}{L}\right)^2 t\right). \tag{22}$$

Considering the superposition principle, because the initial problem is homogeneous, all linear combinations of Equation (22) are solutions of the heat equation without initial conditions. Therefore, any following function is a solutions of the heat equation without initial conditions.

$$u(x,t) = \sum_{n=0}^{+\infty} B_n \sin\left(\frac{n\pi}{L}x\right) \times \exp\left(-\left(\frac{cn\pi}{L}\right)^2 t\right). \tag{23}$$

Finally, considering the initial condition $u(x,0) = f(x)$, a Fourier decomposition of f allows to set all coefficients B_n, showing that, for any initial condition f, there exists a solution to Equation (18) of the form of Equation (23).

Uniqueness We present here elements of proof for establishing the uniqueness of the solutions of Equation (18) that belong to $\mathcal{C}^2\left([0,1] \times \mathbb{R}_+\right)$. Detailed and rigorous proofs are given by Le Dret & Lucquin (2016).

The key element consists in establishing the so-called Maximum Principle which states that, considering a sufficiently smooth solution, the minimum value of the solution is reached on the boundary of the space and time domains.

For null border condition (as we have here), this means that the norm of the solution u is given by the norm of f (because of the initial condition). Finally, let us consider two smooth solutions of Equation (18) U_1 and U_2. Then, their difference $v = U_1 - U_2$ follows the heat equation with null border and initial conditions (i.e, $v(x, 0) = 0$). Because v is as regular as U_1 and U_2, it satisfies the previous fact about the norm of the solutions, i.e, the norm of v equals the norm of its initial condition: $\|v\| = 0$. Therefore, v is null and so is $U_1 = U_2$, showing the unicity of the solutions.

Finally, this show that solutions of the form of Equation (23) shape the whole set of smooth solutions of Equation (18).

A.2 Heat Equation with Advection Term (Equation (4))

We give here details for the existence of product-separable solutions to Equation (4):

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \chi\frac{\partial^2 u}{\partial x^2}, \quad \text{for} - 1 < x < 1 \text{ and } t < T, c > 0. \tag{24}$$

Let $\alpha, \beta \in \mathbb{R}$; consider the following change of variables for u:

$$u(x, t) = v(x, t)\mathrm{e}^{\alpha x+\beta t}. \tag{25}$$

The partial derivatives from Equation (4) can be rewritten as functions of the new variable v:

$$\frac{\partial u}{\partial t} = \frac{\partial v}{\partial t}\,\mathrm{e}^{\alpha x+\beta t} + v \times \beta \mathrm{e}^{\alpha x+\beta t} \tag{26}$$

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial v}\,\mathrm{e}^{\alpha x+\beta t} + \alpha v \mathrm{e}^{\alpha x+\beta t} \tag{27}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 v}{\partial x^2} \times \mathrm{e}^{\alpha x+\beta t} + 2\alpha \times \frac{\partial v}{\partial x}\,\mathrm{e}^{\alpha x+\beta t} + \alpha^2 v e^{\alpha x+\beta t} \tag{28}$$

Using these expressions in Equation (4) and dividing it by $\mathrm{e}^{\alpha x+\beta t}$ leads to:

$$\frac{\partial v}{\partial t} + \Big(\beta + c\alpha - \alpha^2\chi\Big)v + (c - 2\alpha\chi)\frac{\partial v}{\partial x} = \nu\frac{\partial^2 v}{\partial x^2}. \tag{29}$$

α and β, being dummy parameters, cen be set such that:

$$\begin{aligned} \beta + c\alpha - \alpha^2\chi &= 0 \\ c - 2\alpha\chi &= 0 \end{aligned}$$

We then retrieve the standard two-dimensional heat equation of Equation (18) given by:

$$\frac{\partial v}{\partial t} = \chi\frac{\partial^2 v}{\partial x^2}, \tag{30}$$

which is known to have product-separable solutions as explained in the previous section. This more generally shows all solutions of Equation (4) can be retrieved from solutions to Equation (18).

B Accessing Time Derivatives of S

While explicitly constraining the time derivative of $E_S\big(V_\tau(t)\big)$ seems more intuitive than imposing time invariance as explained in Section 4.5, it is a difficult matter in practice. Indeed, E_S does not take as input neither the time coordinate t nor spatial coordinates x, y as done by Raissi (2018) and Sirignano & Spiliopoulos (2018), which allows the authors to directly estimate the networks derivative thanks to automatic differentiation. In our case, E_S rather takes as input observations.

As discussed in Section 4, a possible but costly solution to impose time invariance would be to discretize the left hand side of Equation (14), so that the following quantity would be minimized:

$$\mathcal{L}^S_{\text{first order}} = \frac{1}{\nu - \tau}\sum_{i=1}^{\nu-\tau}\left\| E_S\big(V_\tau(t_0 + i\Delta t)\big) - E_S\Big(V_\tau\big(t_0 + (i-1)\Delta t\big)\Big)\right\|_2^2. \tag{31}$$

Another alternative is the loss introduced by Denton & Birodkar (2017) that instead minimized the difference between spatial representations taken at two random steps i and j uniformly sampled in $[\![0, \nu - \tau]\!]$:

$$\mathcal{L}^S_{\text{random}} = \left\| E_S\big(V_\tau(t_i)\big) - E_S\Big(V_\tau\big(t_j\big)\Big) \right\|_2^2. \tag{32}$$

However, both alternatives hinder performances, as analyzed in Appendix F, since they implement an overly strong constraint, as explained in Section 4.5.

Another workaround would be to model explicitly the evolution of $E_S\big(V_\tau(t)\big)$ with respect to time thanks to an integrator and a regression loss, similarly to T. It would give access to an estimate of the evolution of $E_S\big(V_\tau(t)\big)$ through time, enabling a direct control of the left hand side of Equation (14). This estimate could be loose and take into account the variation of spatial information due to potential hidden spatial features in the observations, allowing to relax the overly strong penalizing constraint.

To investigate this possibility, we propose to model the evolution of $E_S\big(V_\tau(t)\big)$ using a residual network, denoted by R_S. Indeed, residual networks have been shown to implement ODE resolution schemes (Lu et al., 2017), thus assimilating their residual blocks to the true time derivatives of the system. Then, the regularisation loss on S, previously denoted $\mathcal{L}^S_{\text{reg}}$ is replaced by two components. The first component is a regression loss ensuring that our residual network R_S accurately models the evolution of S:

$$\mathcal{L}^S_{\text{ODE}} = \left\| R_S\Big(E_S\big(V_\tau(t_0)\big)\Big) - E_S\big(V_\tau(t_1 - \tau\Delta t)\big) \right\|_2^2. \tag{33}$$

The second component is the regularisation of the residuals. We propose to minimize the ℓ_2-norm of the residuals as a proxy to minimise the true time derivative of $E_S\big(V_\tau(t)\big)$. If $r_{S,1}, \ldots, r_{S,l_S}$ are the residual blocks of R_S, we define the regularization implementing the left hand side of Equation (14) as:

$$\mathcal{L}^S_{\text{resblock}} = \sum_{h=1}^{l_S} \left\| \Big(r_{S,h} \circ \big(\text{id} + r_{S,h-1}\big) \circ \ldots \circ \big(\text{id} + r_{S,1}\big)\Big)\Big(E_S\big(V_\tau(t_0)\big)\Big) \right\|_2^2. \tag{34}$$

We show in Appendix F that this workaround is a viable alternative, as using such constraint leads to results that are numerically similar to the originally proposed method. Yet, even though it achieves slightly better results, this alternative is computationally less efficient than our method, and requires one more hyperparameter to tune (the coefficient in front of $\mathcal{L}^S_{\text{ODE}}$), making its use more complex. Therefore, it is an interesting option to study that also provides state-of-the-art results.

C Of Spatiotemporal Disentanglement

C.1 Modeling Spatiotemporal Phenomena with Differential Equations

Besides their increasing popularity to model spatiotemporal phenomena (see Section 2), the ability of residual networks to facilitate learning (Haber & Ruthotto, 2017) along with the success of their continuous counterpart (Chen et al., 2018) motivates our choice. Indeed, learning ODEs or discrete approximations as residual networks has become standard for a variety of tasks such as classification, inpainting, and generative models. Consequently, their application to forecasting physical processes and videos is only a natural extension of its already broad applicability discussed in Section 2. Furthermore, they present interesting properties, as detailed below.

C.2 Separation of Variables Preserves the Mutual Information of S and T through Time

C.2.1 Invertible Flow of an ODE

We first highlight that the general ODE Equation (10) admits, according to the Cauchy–Lipschitz theorem, exactly one solution for a given initial condition, since f is implemented with a standard neural network (see Appendix E), making it Lipschitz-continuous. Consequently, the flow of this ODE, denoted by Φ_t and defined as:

$$\begin{aligned} \Phi\colon \mathbb{R} \times \mathbb{R}^p &\to \mathbb{R}^p \\ (t_0, T_{t_0}) &\mapsto \Phi_t(T_{t_0}) = T_{t_0+t} \end{aligned}$$

is a bijection for all t. Indeed, let T_{t_0} be fixed and t_0, t_1 be two timesteps; thanks to the existence and unicity of the solution to the ODE with this initial condition: $\Phi_{t_0+t_1} = \Phi_{t_0} \circ \Phi_{t_1} = \Phi_{t_1} \circ \Phi_{t_0}$. Therefore, Φ_t is a bijection and $\Phi_t^{-1} = \Phi_{-t}$. Moreover, the flow is differentiable if f is continuously differentiable as well, which is not a restrictive assumption if it is implemented by a neural network with differentiable activation functions.

C.2.2 Preservation of Mutual Information by Invertible Mappings

A proof of the following result is given by Kraskov et al. (2004). We indicate below the major steps of the proof. Let X and Y be two random variables with marginal density μ_X, μ_Y. Let F be a diffeomorphism acting on Y, $Y' = F(Y)$. If J_F is the determinant of the Jacobian of F, we have:

$$\mu'\big(x, y'\big) = \mu(x, y) J_F\big(y'\big).$$

Then, expressing the mutual information I in integral form, with the change of variables $y' = F(y)$ (F being a diffeomorphism), results in:

$$\begin{aligned} I\big(X, Y'\big) &= \iint_{x,y'} \mu'\big(x, y'\big) \log \frac{\mu'\big(x, y'\big)}{\mu_X(x) \times \mu_{Y'}(y')} \,\mathrm{d}x \,\mathrm{d}y' \\ &= \iint_{x,y} \mu(x, y) \log \frac{\mu(x, y)}{\mu_X(x) \times \mu_Y(y)} \,\mathrm{d}x \,\mathrm{d}y \\ I\big(X, Y'\big) &= I(X, Y). \end{aligned}$$

C.3 Ensuring Disentanglement at any Time

As noted by Chen et al. (2016) and Achille & Soatto (2018), mutual information I is a key metric to evaluate disentanglement. We show that our model logically conserves the mutual information between S and T through time thanks to the flow of the learned ODE on T. Indeed, with the result of mutual information presevation by diffeomorphisms, and Φ_t being a diffeomorphism as demonstrated above, we have, for all t and t':

$$I(S, T_t) = I\big(X, \Phi_{t'-t}(T_t)\big) = I(S, T_{t'}). \tag{35}$$

Hence, if S and T_t are disentangled, then so are S and $T_{t'}$.

The flow Φ_t being dicretized in practice, its invertibility can no longer be guaranteed in general. Some numerical schemes (Chen et al., 2020) or residual networks with Lipschitz-constrained residual blocks (Behrmann et al., 2019) provide sufficient conditions to concretely reach this invertibility. In our case, we did not observe the need to enforce invertibility. We can also leverage the data processing inequality to show that, for any $t \geq t_0$:

$$I(S, T_{t_0}) \geq I(S, T_t), \tag{36}$$

since T_t is always a deterministic function of T_{t_0}. Since we constrain the very first T value T_{t_0} (i.e., we do not need to go back in time), there is no imperative need to enforce the invertibility of Φ_t in practice: the inequality also implies that, if S and T_{t_0} are disentangled, then so are S and T_t for $t \geq t_0$. Nevertheless, should the need to disentangle for $t < t_0$ appear, the aforementioned mutual information conservation properties could allow, with further practical work to ensure the effective invertibility of Φ_t, to still regularize T_{t_0} only. This is, however, out of the scope of this paper.

D Datasets

D.1 WaveEq and WaveEq-100

These datasets are based on the two-dimensional wave equation on a functional $w(x, y, t)$:

$$\frac{\partial^2 w}{\partial t^2} = c^2 \nabla^2 w + f(x, y, t), \tag{37}$$

where ∇^2 is the Laplacian operator, c denotes the wave celerity, and f is an arbitrary time-dependent source term. It has several application in physics, modeling a wide range of phenomena ranging from

mechanical oscillations to electromagnetism. Note that the homogeneous equation, where $f = 0$, admits product-separable solutions.

We build the WaveEq dataset by solving Equation (37) for $t \in [0, 0.298]$ and $x, y \in [0, 63]$. Sequences are generated using c drawn uniformly at random in $[300, 400]$ for each sequence to imitate the propagation of acoustic waves, with initial and Neumann boundary conditions:

$$w(x, y, 0) = w(0, 0, t) = w(32, 32, t) = 0, \tag{38}$$

and, following Saha et al. (2020), we make use of the following source term:

$$f(x, y, t) = \begin{cases} f_0 \mathrm{e}^{-\frac{t}{T_0}} & \text{if } (x, y) \in \mathcal{B}\big((32, 32), 5\big) \\ 0 & \text{otherwise} \end{cases}, \tag{39}$$

with $T_0 = 0.05$ and $f_0 \sim \mathcal{U}\big([1, 30]\big)$. The source term is taken non-null in a circular central zone only in order to avoid numerical differentiation problems in the case of a punctual source.

We generate 300 sequences of 64×64 frames of length 150 from this setting by assimilating pixel $(i, j) \in [\![0, 63]\!] \times [\![0, 63]\!]$ to a point $(x, y) \in [0, 63] \times [0, 63]$ and selecting a frame per time interval of size 0.002. This discretization is used to solve Equation (37) as its spatial derivatives are estimated thanks to finite differences; once computed, they are used in an ODE numerical solver to solve Equation (37) on t. Spatial derivatives are estimated with finite differences of order 5, and the ODE solver is the fourth-order Runge-Kutta method with the $3/8$ rule (Kutta, 1901; Hairer et al., 1993) and step size 0.001. The data are finally normalized following a min-max $[0, 1]$ scaling per sequence.

The dataset is then split into training (240 sequences) and testing (60 sequences) sets. Sequences sampled during training are random chuncks of length $\nu + 1 = 25$, including $\tau + 1 = 5$ conditioning frames, of full-size training sequences. Sequences used during testing are all possible chunks of length $\tau + 1 + 40 = 45$ from full-size testing sequences.

Finally, WaveEq-100 is created from WaveEq by selecting 100 pixels uniformly at random. The extracted pixels are selected before training and are fixed for both training and test. Therefore train and test sequences for WaveEq-100 consist of vector of size 100 extracted from WaveEq frames. Training and testing sequences are chosen to be the same as those of WaveEq.

D.2 Sea Surface Temperature

SST is composed of sea surface temperatures of the Atlantic ocean generated using E.U. Copernicus Marine Service Information thanks to the state-of-the-art simulation engine NEMO. The use of a so-called reanalysis procedure implies that these data accurately represent the actual temperature measures. For more information, we refer to the complete description of the data by de Bézenac et al. (2018). The data history of this engine is available online.[3] Unfortunately, due to recent maintenance, data history is limited to the last three years; prior histories should be manually requested.

The dataset uses daily temperature acquisitions from Thursday 28th December, 2006 to Wednesday 5th April, 2017 of a 481×781 zone, from which 29 zones of size 64×64 zones are extracted. We follow the same setting as de Bézenac et al. (2018) by training all models with $\tau + 1 = 4$ conditioning steps and $\nu - \tau = 6$ steps to predict, and evaluating them on only zones 17 to 20. These zones are particularly interesting since they are the places where cold waters meet warm waters, inducing more pronounced motion.

We normalize the data in the same manner as de Bézenac et al. (2018). Each daily acquisition of a zone is first normalized using the mean and standard deviation of measured temperatures in this zone computed for all days with the same date of the year from the available data (daily history climatological normalization). Each zone is then normalized so the mean and variance over all acquisitions correspond to those of a standard Gaussian distribution. These normalized data are finally fed to the model; MSE scores reported in Table 1 are computed once the performed normalization of the data and model prediction is reverted to the original temperature measurement space, in order to compute physically meaningful scores.

[3] https://resources.marine.copernicus.eu/?option=com_csw&view=details&product_id=GLOBAL_ANALYSIS_FORECAST_PHY_001_024.

Training sequences correspond to randomly selected chunks of length $\nu = 10$ in the first 2987 acquisitions (corresponding to 80% of total acquisitions), and testing sequences to all possible chunks of length $\nu = 10$ in the remaining 747 acquisitions.

D.3 Moving MNIST

This dataset involves two MNIST digits (LeCun et al., 1998) of size 28×28 that linearly move within 64×64 frames and deterministically bounce against frame borders following reflection laws. We use the modified version of the dataset proposed by Franceschi et al. (2020) instead of the original one (Srivastava et al., 2015). We train all models in the same setting as Denton & Birodkar (2017), with $\tau + 1 = 5$ conditioning frames and $\nu - \tau = 10$ frames to predict, and test them to predict either 10 or 95 frames ahead. Training data consist in trajectories of digits from the MNIST training set, randomly generated on the fly during training. Test data are produced by computing a trajectory for each digit of the MNIST testing set, and randomly pairwise combining them, thus producing 5000 sequences.

To evaluate disentanglement with content swapping, we report PSNR and SSIM metrics between the swapped sequence produced by our model and a ground truth. However, having two digits in the image, there is an ambiguity as to in which order target digits should be swapped in the ground truth. To account for this ambiguity and thanks to the synthetic nature of the dataset, we instead build two ground truth sequences for both possible digit swap permutations, and report the lowest metric between the generated sequence and both ground truths (i.e., we choose the closest ground truth to compare to with respect to the considered metric).

E Training Details

Along with the code, we provide here sufficient details in order to replicate our results.

E.1 Reproduction of PKnl, DrNet and DDPAE

PKnl. We retrained PKnl (de Bézenac et al., 2018) on SST using their official implementation and the same hyperparameters they indicate.

DrNet. We trained DrNet (Denton & Birodkar, 2017) on our version of Moving MNIST using the same hyperparameters originally used for the alternative version of the dataset on which it was originally trained (with digits of different colors). To this end, we reimplemented the official Lua implementation into a Python code in order to train it with a more recent infrastucture.

DDPAE. We trained DDPAE (Hsieh et al., 2018) on our version of Moving MNIST using the official implementation and the same hyperparameters they used for the original version of Moving MNIST.

E.2 Model Specifications

E.2.1 Implementation

We used Python 3.8.1 and PyTorch 1.4.0 (Paszke et al., 2019) to implement our model. Each model was trained on an Nvidia GPU with CUDA 10.1. Training is done with mixed-precision training (Micikevicius et al., 2018) thanks to the Apex library.[4]

E.2.2 Architecture

Combination of S and T. As explained in Section 4, the default choice of combination of S and T as decoder inputs is the concatenation of both vectorial variables: it is generic, and allows the decoder to learn an appropriate combination function ζ as in Equation (7).

Nonetheless, further knowledge of the studied dataset can help to narrow the choices of combination functions. Indeed, we choose to multiply S and T before giving them as input to the decoder for

[4] `https://github.com/nvidia/apex`.

both datasets WaveEq and WaveEq-100, given the knowledge of the existence of product-separable solutions to the homogeneous version of equation (i.e., without source). This shows that it is possible to change the combination function of S and T, and that existing combination functions in the PDE literature could be leveraged for other datasets.

Encoders E_S and E_T, and decoder D. For WaveEq, the encoder and decoder outputs are considered to be vectors; images are thus reshaped before encoding and after encoding to 64×64 frames. The encoder is a MultiLayer Perceptrons (MLP) with two hidden layers of size 1200 and internal ReLU activation functions. The decoder is an MLP with three hidden layers of size 1200, internal ReLU activation functions, and a final sigmoid activation function for the decoder. The encoder and decoder used for WaveEq-100 are similar to those used for WaveEq, but with two hidden layers each, of respective sizes 2400 and 150.

We used for SST a VGG16 architecture (Simonyan & Zisserman, 2015), mirrored between the encoder and the decoder, complemented with skip connections integrated into S (Ronneberger et al., 2015) from all internal layers of the encoder to corresponding decoder layers, also leveraged by de Bézenac et al. (2018) in their PKnl model. For Moving MNIST, the encoder and its mirrored decoder are shaped with the DCGAN discriminator and generator architecture (Radford et al., 2016), with an additional sigmoid activation after the very last layer of the decoder; this encoder and decoder DCGAN architecture is also used by DrNet and DDPAE. We highlight that we leveraged in both SST and Moving MNIST architectural choices that are also used in compared baselines, enabling fair comparisons.

Encoders E_S and E_T taking as input multiple observations, we combine them by either concatenating them for the vectorial observations of WaveEq-100, or grouping them on the color channel dimensions for the other datasets where observations are frames. Each encoder and decoder was initialized from a normal distribution with standard deviation 0.02.

ODE solver. Following the recent line of work assimilating residual networks (He et al., 2016) with ODE solvers (Lu et al., 2017; Chen et al., 2018), we use a residual network as an integrator for Equation (10). This residual network is composed of a given number K of residual blocks, each block $i \in [\![1, K]\!]$ implementing the application $\mathrm{id} + g_i$, where g_i is an MLP with a two hidden layers of size H and internal ReLU activation functions. The parameter values for each dataset are:

- WaveEq and WaveEq-100: $K = 3$ and $H = 512$;
- SST: $K = 5$ and $H = 1024$;
- Moving MNIST: $K = 1$ and $H = 512$.

Each MLP is orthogonally initialized with the following gain for each dataset:

- WaveEq, WaveEq-100 and SST: 0.71;
- Moving MNIST: 1.41.

Latent variable sizes. S and T have the following vectorial dimensions for each dataset:

- WaveEq and WaveEq-100: 32;
- SST: 256;
- Moving MNIST: respectively, 128 and 20.

Note that, in order to perform fair comparisons, the size of T for baselines without static component S is chosen to be the sum of the vectorial sizes of S and T in the full model. The skip connections of S for SST cannot, however, be integrated into T, as its evolution is only modeled in the latent, and it is out of the scope of this paper to leverage low-level dynamics.

E.3 Optimization

Optimization is performed using the Adam optimizer (Kingma & Ba, 2015) with initial learning rate 4×10^{-4} for WaveEq, WaveEq-100 and Moving MNIST and 2×10^{-4} for SST, and with decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.99$.

Loss function. Chosen coefficients values of λ_{pred}, λ_{AE}, λ_{reg}^{S}, and λ_{reg}^{T} are the following:

- $\lambda_{\text{pred}} = 45$;
- $\lambda_{\text{AE}} = 10$ for SST and Moving MNIST, and 1 for WaveEq and WaveEq-100;
- $\lambda_{\text{reg}}^{S} = 45$ for WaveEq, WaveEq-100 and Moving MNIST, and 1500 for SST;
- $\lambda_{\text{reg}}^{T} = \frac{1}{2}p \times 10^{-3}$, where p is the dimension of T.

The batch size is chosen to be 128 for WaveEq, WaveEq-100 and Moving MNIST, and 100 for SST.

Training length. The number of training epochs for each dataset is:

- WaveEq and WaveEq-100: 250 epochs;
- SST: 200 epochs for the full model, and 75 epochs for the model without S (the latter tending to overfit for higher number of epochs);
- Moving MNIST: 800 epochs, with an epoch corresponding to 200 000 trajectories (the dataset being infinite), with the learning rate successively divided by 2 at epochs 300, 400, 500, 600, and 700.

These correspond to the following appoximate training times of an Nvidia Titan V GPU:

- WaveEq and WaveEq-100: two hours;
- SST: a day;
- Moving MNIST: two days and a half.

E.4 Prediction Offset for SST

Using the formalism of our work, our algorithm trains to reconstruct $v = (v_{t_0}, ..., v_{t_1})$ from conditioning frames $V_\tau(t_0)$. Therefore, it first learns to reconstruct V_τ.

However, the evolution of SST data is chaotic and predicting above an horizon of 6 with coherent and sharp estimations is challenging. Therefore, for the SST dataset only, we chose to supervise the prediction from $t = t_0 + (\tau + 1)\Delta t$, i.e, our algorithm trains to forecast $v_{t_0+(\tau+1)\Delta t}, \ldots, v_{t_1}$ from $V_\tau(t_0)$. It simply consists in making the temporal representation $E_T\big(V_\tau(t_0)\big)$ match the observation $v_{t_0+(\tau+1)\Delta t}$ instead of v_{t_0}. This index offset does not change our interpretation of spatiotemporal disentanglement through separation of variables.

F Additional Results and Samples

F.1 Additional Results on Moving MNIST

We compare results on the Moving MNIST dataset in Table 3 for the several variants to impose time invariance as detailed in Appendix B.

We can conclude from Table 3 that our proposed method to enforce time invariance for minimizing the time derivative is significantly more efficient than directly minimizing the left hand side of Equation (14) ($\mathcal{L}^{S}_{\text{first order}}$). Indeed, our method provides more consistent long-term forecasts than those produced using $\mathcal{L}^{S}_{\text{first order}}$. Furthermore, it performs better in both long and short-term forecasts than the $\mathcal{L}^{S}_{\text{random}}$ proposed by Denton & Birodkar (2017). Finally, compared to both $\mathcal{L}^{S}_{\text{first order}}$ and $\mathcal{L}^{S}_{\text{random}}$, our method to impose time invariance strengthens the disentanglement ability of our algorithm providing better results in the swap experiment at $t + 95$. These results confirm the analysis of Appendix B.

Finally, modeling the evolution of the spatial content and minimizing the ℓ_2-norm of the residuals is a competitive alternative compared to our approach in both prediction and disentanglement, but is more complex and computationally heavier as its execution time is increased by about 20%.

Table 3: PSNR and SSIM scores of DrNet, DDPAE and our model on the Moving MNIST dataset for prediction and content swap tasks. The first part of the table reports results of Table 2, and the second half report additional results for alternative invariance losses. Bold scores indicate the best performing method in each part of the table.

Models	Pred. $(t+10)$		Pred. $(t+95)$		Swap $(t+10)$		Swap $(t+95)$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DrNet	14.94	0.6596	12.91	0.5379	14.12	0.6206	12.80	0.5306
DDPAE	21.17	0.8814	13.56	0.6446	**18.44**	0.8256	13.25	0.6378
Ours	**21.74**	**0.9094**	**17.22**	**0.7867**	18.30	**0.8343**	**16.21**	**0.7600**
Ours (without S)	Failed: underflow after a few iterations							
Ours ($\lambda_{\text{AE}} = 0$)	21.51	0.9065	15.17	0.7054	18.01	0.8274	14.52	0.6884
Ours ($\lambda^S_{\text{reg}} = 0$)	15.69	0.6670	13.77	0.6770	13.76	0.5392	13.56	0.6631
Ours ($\lambda^T_{\text{reg}} = 0$)	15.06	0.7030	13.96	0.7218	14.64	0.6907	13.92	0.7208
Ours ($\mathcal{L}^S_{\text{resblock}}, \mathcal{L}^S_{\text{ODE}}$)	**21.76**	**0.9080**	**17.89**	**0.8130**	18.30	0.8327	**16.68**	**0.7793**
Ours $\mathcal{L}^S_{\text{first order}}$	21.49	0.9054	15.80	0.7411	17.96	0.8242	15.11	0.7225
Ours $\mathcal{L}^S_{\text{random}}$	21.67	0.9071	16.56	0.7648	**18.39**	**0.8351**	15.74	0.7432

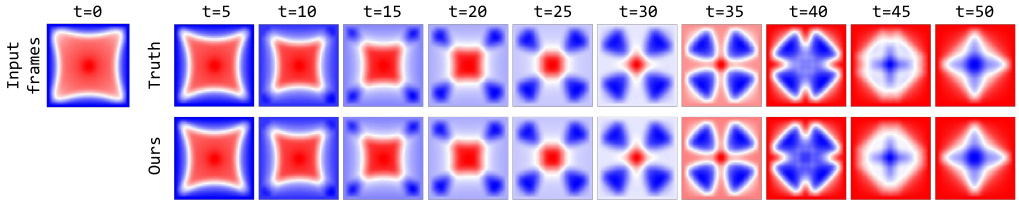

Figure 4: Example of predictions of our model on WaveEq.

F.2 Additional Samples

F.2.1 WaveEq

We provide in Figure 4 a sample for the WaveEq dataset, highlighting the long-term consistency in the forecasts of our algorithm.

We also show in Figure 5 the effect in forecasting of changing the spatial code S from the one of another sequence.

F.2.2 SST

We provide an additional sample for SST in Figure 6.

F.2.3 Moving MNIST

We provide two additional samples for Moving MNIST in Figures 7 and 8.

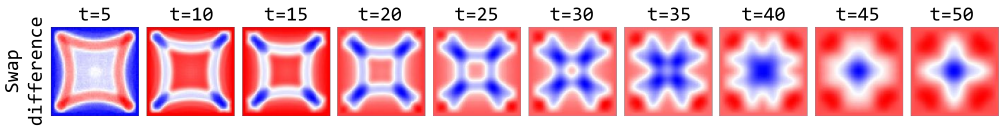

Figure 5: Evolution of the scaled difference between the forecast of a sequence and the same forecast with a spatial code coming from another sequence for the WaveEq dataset.

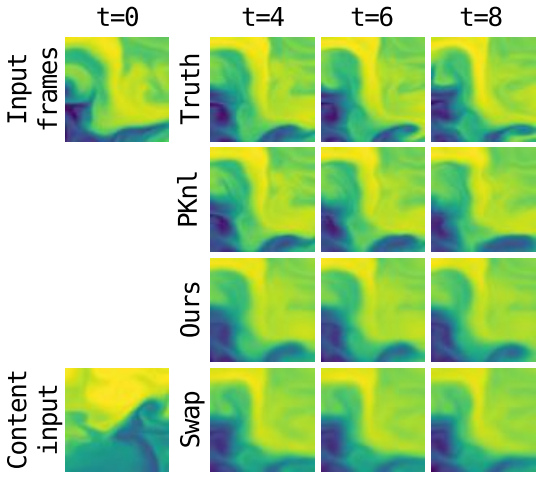

Figure 6: Example of predictions of compared models on SST.

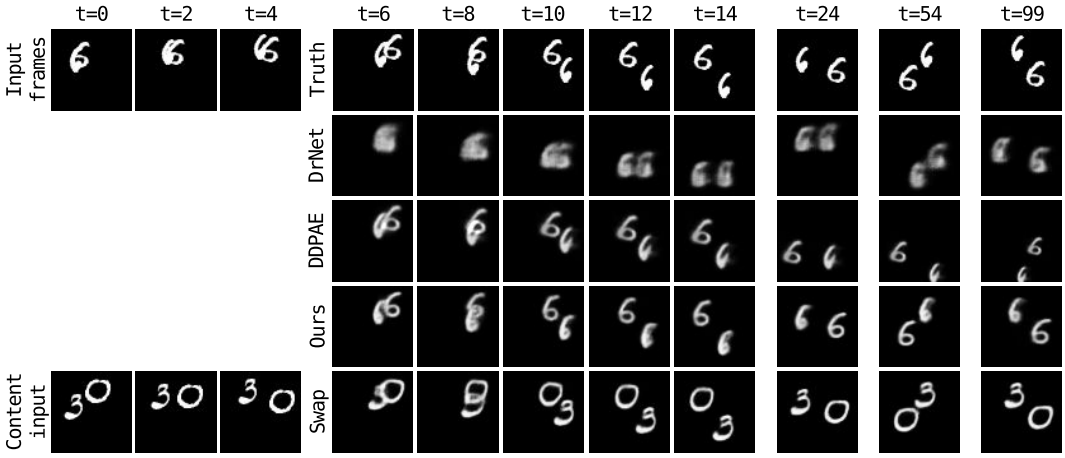

Figure 7: Example of predictions of compared models on Moving MNIST.

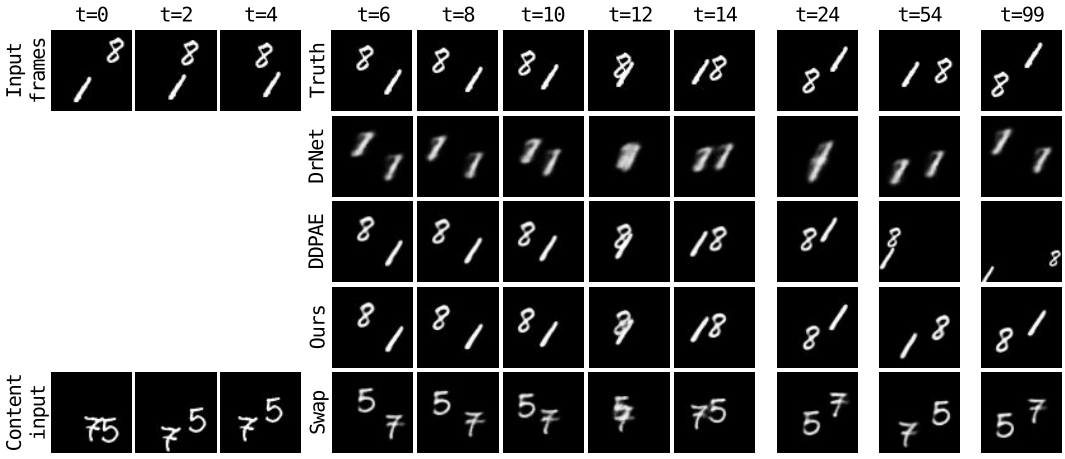

Figure 8: Example of predictions of compared models on Moving MNIST.