Physics-Based Deep Learning for Fluid Flow

Nils Thuerey, You Xie, Mengyu Chu, Steffen Wiewel, Lukas Prantl
Technical University of Munich

1 Introduction and Related Work

Learning physical functions is an area of strongly growing interest, with applications ranging from physical models for analyzing motions in videos [3, 9], over control of robots [6], to fast approximations for numerical solvers [4, 8]. However, while humans often effortlessly predict the outcome of complex interactions of real world objects, similar effects are currently still extremely difficult to decode for machine learning approaches. The overarching goal of our work is to employ physical simulations as priors to regularize learning problems, i.e., to give learning processes a physical intuition and understanding. At the same time, these learned models have the potential to improve computational performance and accuracy of physical simulations.

Despite first steps in this direction, the often highly non-linear behavior of the underlying physical models in conjunction with large numbers of degrees of freedom pose significant challenges. In the following we will give an overview of several recent works from the area of deep learning for fluid flow, and discuss open problems as well as future directions of research.

Water and air, i.e. fluids in general, are ubiquitous in our world. At the same time, they represent a wide class of challenging physics problems, as the constantly changing motions and boundary conditions result in a complex space of motions and configurations. Although proofs for the existence of smooth solutions are still outstanding, the *Navier-Stokes* equations are an established physical model to describe such fluids. They typically take the form

$$\partial \mathbf{u}/\partial t + \mathbf{u} \cdot \nabla \mathbf{u} = -1/\rho \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}\ , \quad \nabla \cdot \mathbf{u} = 0, \tag{1}$$

where the fluid velocity $\mathbf{u}$, and p pressure and the unknown quantities. The parameters $\mathbf{g}, \rho$ and ν denote external forces, density and viscosity, respectively. The two equations combine advection and diffusion terms with hard constraints, and as such are very good representatives for large classes of physical PDEs.

The research works described in the following illustrate several core aspects of physics-based learning. More specifically:

- We will demonstrate inductive bias in the form of discretized numerical simulations for transport processes, and their importance for flow super-resolution problems.
- Within the same setting we also demonstrate how to learn realistic temporal coherence with adversarial training.
- A second project will illustrate the construction of physical latent spaces for temporal predictions. The resulting dimensionality reduction enables the inference of high-dimensional functions over time.
- Here we will also illustrate how hard constraints, such as conservation laws, can be incoporated into the learning process.

2 Transport and Temporal Coherence

We recently proposed a temporally coherent generative model addressing the super-resolution problem for fluid flows [11]. Our work represents a first approach to synthesize four-dimensional physics

Preprint. Work in progress.

fields with neural networks. The approach is based on a conditional generative adversarial network that is designed for the inference of three-dimensional volumetric data. It receives a coarsely resolved flow solution, which typically represents a passively advected quantity such as smoke, as input, and has the goal to generate a solution with a higher resolution, with a factor of 4x for our tests. In this setting the strong motions and resulting deformations cause significant difficulties for convolutional neural networks. We demonstrate that incorporating the discretized transport operator into the network leads to significantly improved results. While this transport is non-linear in general, it can be represented as a regular matrix for a known motion and time step, and in this form represents a powerful component in neural network architectures. In the following, we will denote the discretized version of this advection operator with $\mathcal{A}$.

A second, crucial component of our approach is a specialized discriminator network that focuses on temporal coherence. This network is included in addition to a regular conditional discriminator which focuses on spatial detail. Given a generator G, a simple direct L_2 based temporal loss for three consecutive frames at times $t-1$,t, and $t+1$ would take the form

$$\mathcal{L}_2 = \|G(x^t) - \mathcal{A}(G(x^{t-1}), v^{t-1})\|_2^2 + \|G(x^t) - \mathcal{A}(G(x^{t+1}), -v_x^{t+1})\|_2^2 \,. \tag{2}$$

Here x^t and v^t denote low-resolution density and velocity at time t. Despite aligned density configurations thanks to the advection with $\mathcal{A}$, we show that this loss function is clearly outperformed by a learned loss function in the form of a specialized discriminator network. This temporal discriminator network D_t is trained to minimize

$$\mathcal{L}_{D_t}(D_t, G) = \mathrm{E}_m[-\log D_t(\tilde{Y}_{\mathcal{A}})] + \mathrm{E}_n[-\log(1 - D_t(\widetilde{G}_{\mathcal{A}}(\tilde{X})))] \,. \tag{3}$$

On a high level, this discriminator is trained with a cross entropy loss to distinguish the classes for ground truth Y and generated samples X. In addition, the tilde denotes a set of three consecutive frames, i.e., $\widetilde{X} = \{x^{t-1},\, x^t, x^{t+1}\}$ that are likewise aligned by advection, similar to Eq. 2. In this way, the discriminator has the capcity to discriminate based on the temporal evolution of the generated samples in comparison to the ground truth. A temporal discriminator trained in this way successfully removes the strong temporal changes that a regular super-resolution network would produce.

In addition, our experiments show that the generator network is able to infer more realistic high-resolution details by using additional physical quantities, such as low-resolution velocities or vorticities. In this way, our network learns to generate advected quantities with highly detailed, realistic, and temporally coherent features. Our method works instantaneously, using only a single time-step of low-resolution fluid data, consisting of a density configuration and a flow field. To the best of our knowledge, our work was the first to demonstrate the usefulness of adversarial training in the context of space-time functions for fluid flow.

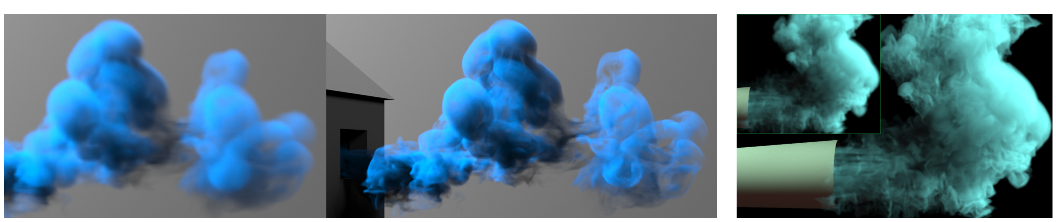

Figure 1: Two examples of high resolution flows generated by our network: on the left, a flow around an obstacle, with the coarse input on the left, and the high resolution output on the right. The output in this case has almost 200 million degrees of freedom. On the right a jet flow example with an output resolution of 512^3 is shown. In this case, the low-resolution input is shown in the top-left inset image.

3 Physical Latent-Spaces and Hard Constraints

We additionally propose a method for learning representations of physical function spaces, i.e. latent spaces of deep neural networks, and for the data-driven inference of temporal evolutions in these latent spaces [10, 2]. The central challenge in this context is the high dimensionality of Eulerian space-time data sets, which arise in many settings of physical problems.

We address this challenge by training an autoencoder to learn an encoding scheme that can adapt to the targeted space of solutions. We have found that deep-learning based encoders significantly outperforms generic methods for the compression of scientific data sets such as FPZIP [5]. For data sets of buoyant smoke plumes, we have achieved compressions ratios of $172\times$ in two-, and $356\times$ in three dimensions, while keeping the compression errors below those of FPZIP [2].

Despite the excellent performance in terms of dimensionality reduction, deep learning approaches typically do not support hard constraints. While additional terms can be added to the objective function of an optimization or learning process, there are typically no guarantee to what extent such a constraint will be minimized, and in the worst case the optimizer converges to a sub-optimal trade off between the constraints and other regularization terms or minimization targets.

However, in the context of physics simulations, it is possible to re-formulate a problem such that constraints are reflected in the space of solutions. For fluid flow, divergence-freeness of the fluid motion, i.e., conservation of mass, is a constraint of fundamental importance, see Equation 1. Here, instead of representing the flow motion $\mathbf{u}$ as a regular vector field, i.e. $\mathbf{u} \in \mathcal{R}^3$, we can re-phrase the physical model in terms of a so-called *stream function*, $\psi \in \mathcal{R}^3$, with $\mathbf{u} = \nabla \times \psi$. Thus, the velocity is given by the curl of the stream function, and by construction, $\nabla \cdot (\nabla \times \psi) = 0$, thus velocities represented in this fashion are guaranteed to be divergence free up to numerical precision of the curl operator.

Correspondigly, instead of using a loss function directly minimizing velocity for the autoencoder $L_u = |\mathbf{u} - \mathbf{a}|$, with $\mathbf{a}$ denoting the output of the autoencoder, we employ a loss function of the form $L_{sf} = |\mathbf{u} - \nabla \times \mathbf{a}|$. In this way the autoencoder learns to generate a stream function, which we can efficiently convert to a divergence free flow field by computing its curl. Examples of learned spaces of flow behavior can be found in Kim et al. [2]. While the learned models exhibit excellent mass conservation properties, it is worth mentioning that boundary conditions can be more difficult for stream function representations, hence we have focused on single-phase smoke flows with stream functions so far.

4 Temporal Prediction

For the temporal prediction we employ an hybrid LSTM-based approach to predict the changes of the pressure fields over time [10]. We propose an LSTM architecture that is combined with convolutuions in order to reduce the number of required weights.

In this way we demonstrate that dense 3D+time functions of physics system can be predicted within the latent spaces of neural networks, and we arrive at a neural-network based simulation algorithm with significant practical speed-ups. We highlight the capabilities of our method with a series of complex liquid simulations, and with a set of single-phase buoyancy simulations, examples of which are shown in Figure 3. With a trained network, our method very significantly outperforms a regular pressure solver: The pressure inference by the LSTM network for a 128^3 volume with more than 2 million entries takes 9.5ms, on average. Including the times to encode and decode the field, which we measured at 4.1ms and 3.3ms, respectively, this represents a $155\times$ speedup compared to a parallelized state-of-the-art iterative pressure solver [1], running with eight threads.

We also studied the accuracy of the predicted solutions [10, 7]. Two examples of steady-state flow predictions for varying network sizes and training data amounts can be found in Figure 2, which yield relative errors of less than 3% for previously unseen configurations. Detailed evaluations of the trained LSTM models discussed above are given in [10]. There we measure errors with respect to the inferred quantities, e.g. pressure, and for the resulting liquid surface position. In general, our LSTM introduces a certain amount of drift per prediction step, which however, is significantly reduced by employing a traditional solving step in intervals. Interestingly, the LSTM can look ahead in time, i.e., predict multiple future steps at once, with only a negligible loss in accuracy and increase in runtime.

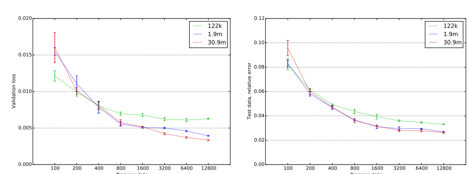

Figure 2: Validation (l) and test accuracy (r).

In the future, networks such as the one we have discussed for predictions of pressure functions could provide pre-trained building blocks for other networks. These pre-trained networks are differentiable, and in contrast to black-box numerical solvers allow for back-propagation without the need to manually specifiy gradients. Thus, they could provide capabilities to predict how a physical system will change and react, i.e., they can yield a physical intuition for deep learning methods.

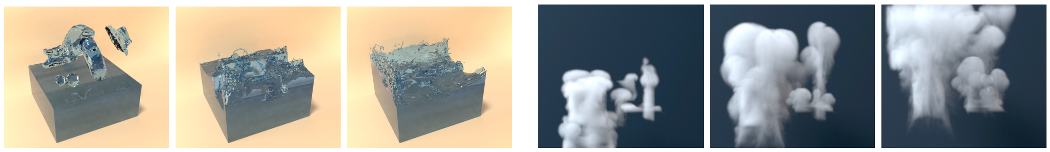

Figure 3: Examples of temporal predictions for a liquid simulation (left), and a buoyant smoke simulation (right).

5 Conclusions

We have discussed two approaches for deep learning in the context of physically-based flow simulations. Despite outperforming traditional techniques, we believe that these methods are only first steps towards a tighter integration of physical models and deep learning. A seamless integration of both fields has the potential for huge impact on both sides: on the one hand it could provide learning methods with a reliable way to predict physical behavior, and as such help us to analyze and improve our models of cognition and learning algorithms. On the other hand, deep learning techniques also have the potential to uncover previously unknown patterns in physical data sets, and in this way help humans to better understand nature.

References

[1] R. Bridson. *Fluid Simulation for Computer Graphics*. CRC Press, 2015.

[2] B. Kim, V. C. Azevedo, N. Thuerey, T. Kim, M. Gross, and B. Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. *arXiv preprint arXiv:1806.02071*, 2018.

[3] N. Kyriazis and A. Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 9–16. IEEE, 2013.

[4] L. Ladicky, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross. Data-driven fluid simulations using regression forests. *ACM Trans. Graph.*, 34(6):199, 2015.

[5] P. Lindstrom and M. Isenburg. Fast and efficient compression of floating-point data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1245–1250, 2006.

[6] C. Schenck and D. Fox. Reasoning about liquids via closed-loop simulation. *arXiv:1703.01656*, 2017.

[7] N. Thuerey, K. Weissenow, H. Mehrotra, N. Mainali, L. Prantl, and X. Hu. Well, how accurate is it? a study of deep learning methods for reynolds-averaged navier-stokes simulations. *arXiv:1810.08217*, 2018.

[8] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *International Conference on Machine Learning*, pages 3424–3433, 2017.

[9] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *Advances in Neural Information Processing Systems*, pages 4542–4550, 2017.

[10] S. Wiewel, M. Becher, and N. Thuerey. Latent-space physics: Towards learning the temporal evolution of fluid flow. *arXiv:1801*, 2018.

[11] Y. Xie, E. Franz, M. Chu, and N. Thuerey. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.*, 37(4), 2018.