

Supplemental Material

Deep Fluids: A Generative Network for Parameterized Fluid Simulations

1 EXTENDED RESULTS

1.1 2-D Smoke Plume

Additional results for the 2-D smoke plume example are shown in Figures 1, 2 and 3. Figure 1 shows vorticity plots for varying both smoke source position and width - smaller left located plumes are shown on the left, while bigger right located plumes are shown on the right. The top row shows the first frame of the simulation, while the middle and bottom rows show frames 100 and 200, respectively. In Figure 2, we show different snapshots for the time evolution of density values at an interpolated smoke source position $\hat{p}_x = 0.47$, with no correspondences on the trained data set. Finally, Figure 3 shows the result of training a network directly using density values instead of velocity fields. The motion of densities is less lively, and the smoke stream breaks up for later frames, yielding unrealistic simulation outputs.

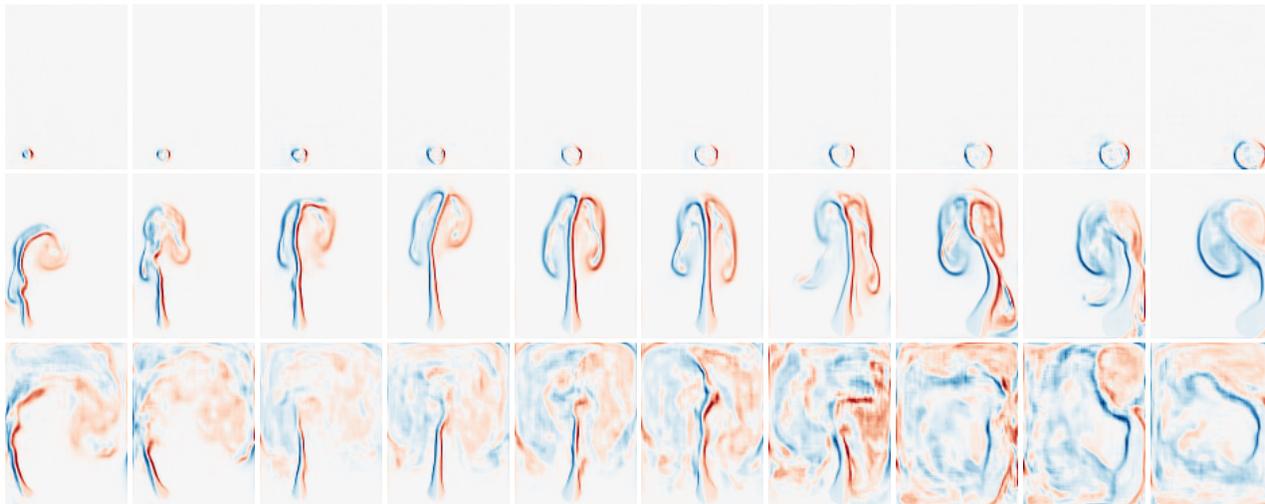


Fig. 1. Several vorticity plots for the 2-D smoke simulation. Each row shows the variation of both position and size of the initial smoke source. Top row corresponds to the first time frame, while the middle and bottom rows show frame 100 and 200, respectively.

1.2 2-D Liquid Drop

Additional results for a 2-D liquid drop are shown as velocity magnitude plots in Figure 4, Figure 5 and Figure 6. Figure 4 shows various reconstructions of varying both initial drop size and x -position. We use intervals of $[0.2, 0.8]$ and $[0.04, 0.08]$ with 10 samples for the position and 4 samples for size, respectively. In Figure 5, we compare our results side-by-side against ground truth simulations, where the top and bottom columns show time frame 37 and 60 respectively. The first, third and fifth columns show ground truth simulations while the second, fourth and sixth columns show the reconstructed counterparts. We notice small differences on splash formation, but the overall bulk of the liquid is respected.

For all data-driven approaches the quality of a trained model, and hence in our case the quality of the simulation results, depend on the training data that is used. One important factor is the sampling density that is used for the training. Especially for generating in-betweens, one has to analyze how dense input simulations must be sampled, in order to avoid artifacts in the reconstruction.

We have conducted a test with a liquid drop scene that has different x -positions in the training simulations. In our test, we varied the number of initial liquid drops - and hence the number of simulations in the training phase - to evaluate how this impacts the ability of the network to generate interpolated results. In particular, we want to understand when the network fails to reconstruct proper liquid interpolations. Figure 6 shows three different sampling densities with gray circles indicating the initial liquid drop in the training data set: 14 (left), 12 (middle) and 10 input samples (right). Different networks were trained with the corresponding number of samples, and intermediate liquid drop positions are generated with the trained models between the first pair of discrete samples for each scene, indicated by the red box.

The second row shows the reconstruction results where the accuracy drops as the sampling density decreases. Hence, we argue that some overlap between noticeable scene features is necessary in order to avoid reconstruction artifacts. Automatically quantifying how good an initial training data set is regarding interpolation is left as future work.

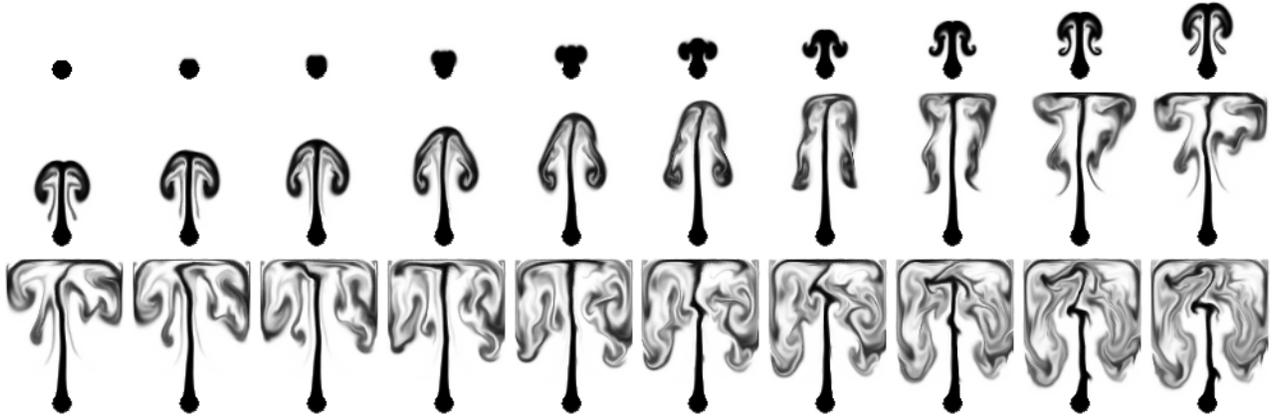


Fig. 2. Different snapshots in time where density is advected by the reconstructed velocity fields at interpolated position $\hat{p}_x = 0.47$. All outputs represent realistic fluid flows.

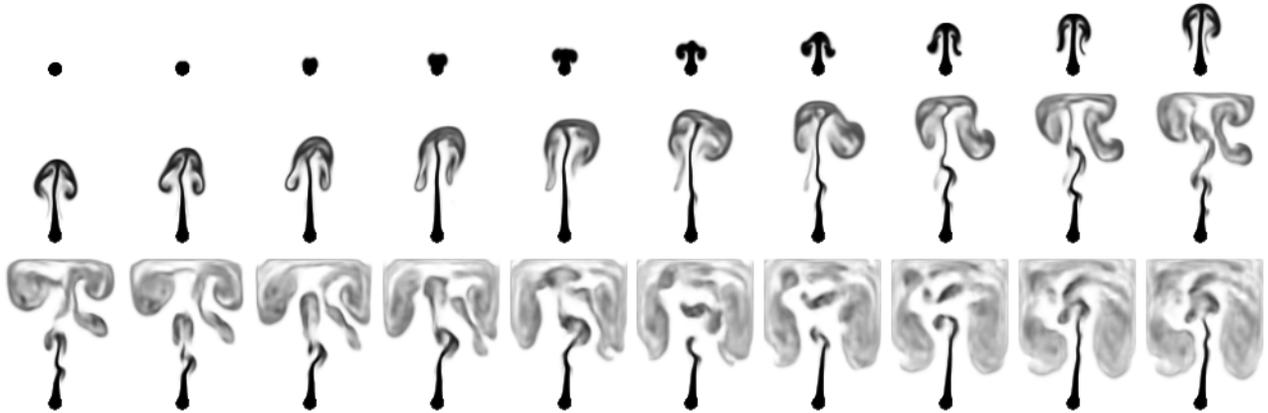


Fig. 3. Different snapshots in time for a network trained *directly* on a density-only data set. The smoke motion is less lively and towards the end of the sequence the main smoke stream breaks apart.

1.3 Resolution Interpolation

Figure 7 shows the capability of our approach to interpolate between unusual input parameters such as the grid resolution. We simulate a rising smoke plume in 3-D with five different grid sizes: $32 \times 48 \times 32$, $48 \times 64 \times 48$, $64 \times 96 \times 64$, $80 \times 120 \times 80$ and $96 \times 144 \times 96$. We then upsample the velocities of the coarser grid resolutions with bicubic interpolation to the finest one, and feed these to our network. Grid resolutions are uniformly scaled, and we are able to map these to a single network input that varies from 0 to 1. We reduced the feature maps to 64 (half of the other examples) as otherwise the highest grid resolution was too large to train with our current hardware. Results of interpolations between different simulation resolutions can be seen in Figure 7.

2 NETWORK ANALYSIS

2.1 Analysis of velocity gradient loss

Figure 8 shows the velocity divergence evolution over the network iterations for the 2-D smoke example when the incompressible loss is *not* applied. The orange and red lines show errors at ground truth positions without and with the gradient loss (i.e., $\lambda_{\nabla \mathbf{u}} = 0/1$), respectively. The

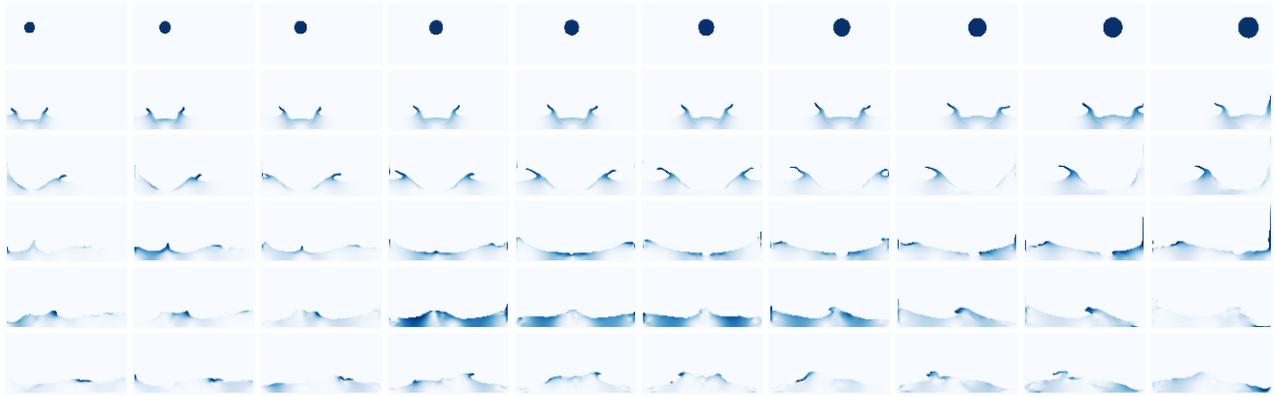


Fig. 4. Velocity magnitude plots for our CNN reconstruction on a 2-D liquid drop scenario. Each column shows the variation of both position and size of the initial liquid drop. The following rows show time frames 37, 70, 100, 150, 200, respectively.

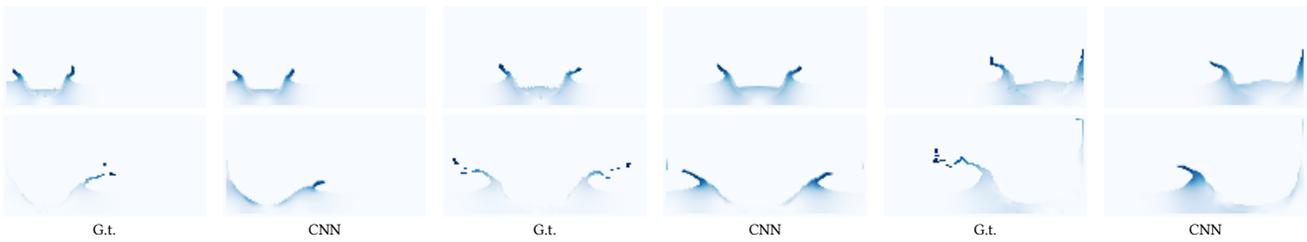


Fig. 5. Comparisons between *ground truth* and reconstructed liquid data. Odd columns (first, third and fifth) show ground truth data, while even columns (second, fourth and sixth) show the reconstruction of the CNN.

blue and green line correspond to the measured divergence at interpolated positions. The velocity gradient clearly helps the network to reduce divergence in the generated flow fields.

Additionally, we compared the performance of the gradient loss on smooth stream function and pressure fields ((a) and (b) of Figure 9, respectively). Since both are considerably smoother than velocity fields, more high-frequency network features are in the null-space of the L_1 loss minimization. Although not visually obvious when directly comparing stream function and pressures field values (Figure 9, top), artifacts clearly appear for their gradients (Figure 9, bottom). These gradients significantly influence simulation results, e.g., visible in the advected densities shown in (Figure 10).

2.2 Quality of the autoencoder and latent space integration networks

We performed additional tests regarding the reconstruction quality of our autoencoder network (Section 4 of the main document) relative to the number of parameters used for the latent space encoding. As shown in Figure 11, the quality of the reconstruction increases with the number of dimensions used for the latent space reconstruction. In Figure 12 (top) we plot the loss function for the different dimensions of Figure 11. Notice that relatively small differences of the L_1 loss impact the visual results significantly for smoke profiles.

Conversely, the latent space integration network performed better with a smaller latent space size, as shown in Figure 12. Increasing the window size w reduces long-term errors of the time advancement. We show the error plots for three different window sizes (1, 5, 30) in Figure 13 (left). In Figure 13 (right) we show advected densities by the latent space integrator network trained with $w = 1$ and $w = 30$. We notice that the errors of the advection are larger for the window size equal to 1, and some smoke samples hang mid-air (highlighted by the red square).

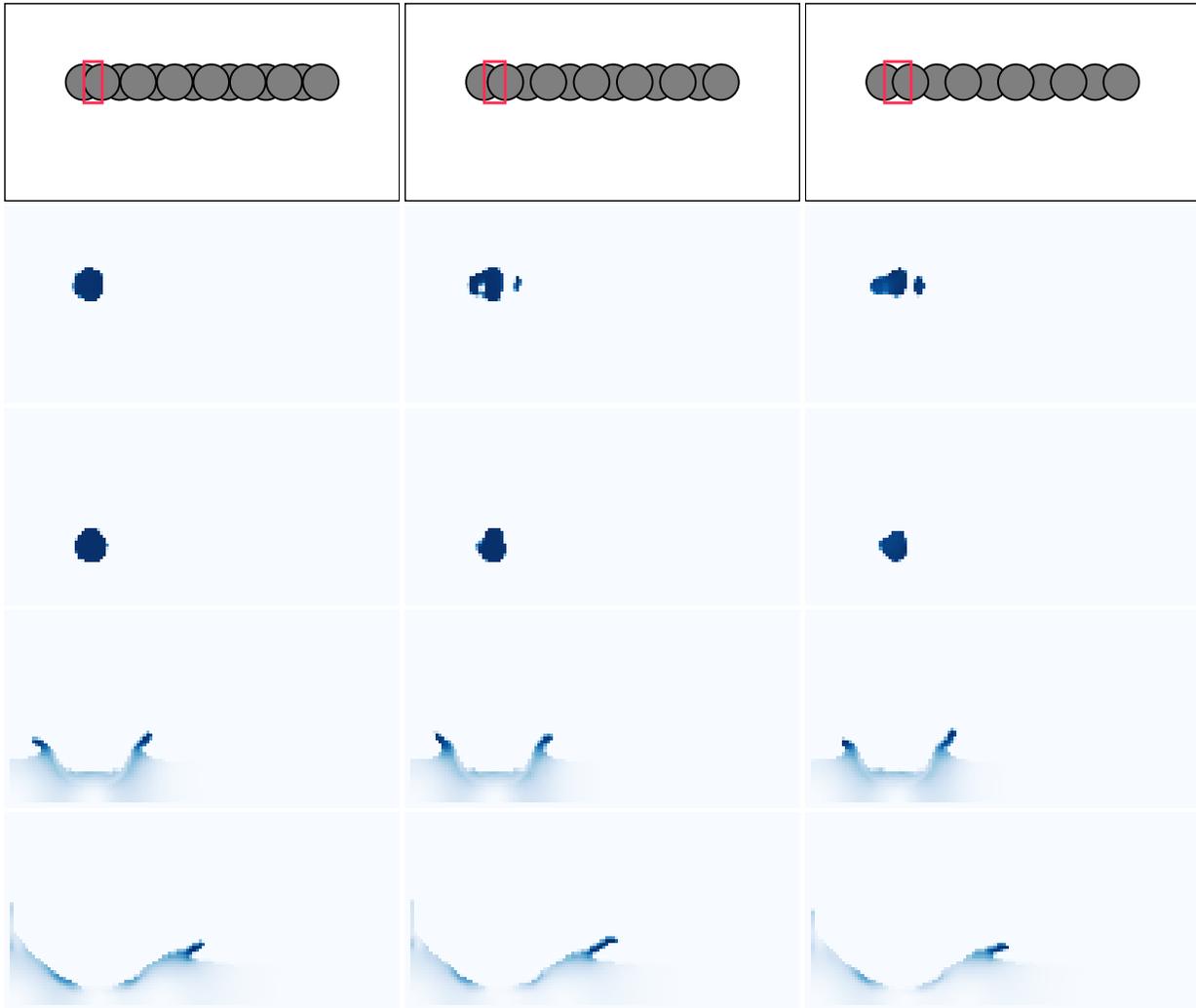


Fig. 6. The sampling density of the input simulations that are used for the training impacts the resulting reconstruction quality. Here, we increasingly vary the number of initial liquid drops on the x-axis (top) and show the interpolation result (inside the red box).



Fig. 7. Trained with five different grid resolutions, our network is able to interpolate resolutions of $40 \times 60 \times 40$ (middle column, left image) and $88 \times 132 \times 88$ (middle column, right image)

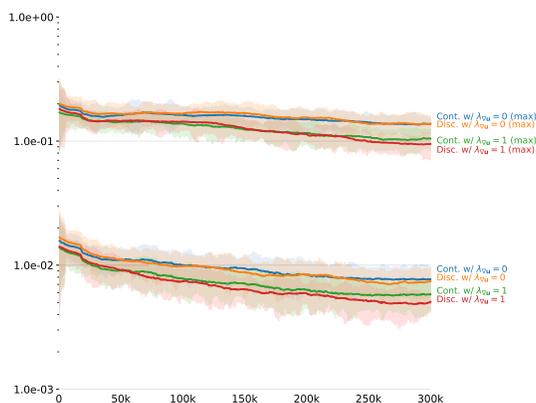


Fig. 8. Divergence max (top) and average (bottom) over number of iterations. Using the gradient velocity in the loss function ($\lambda_{\nabla \mathbf{u}} = 1$) reduces the divergence error, increasing reconstruction quality.

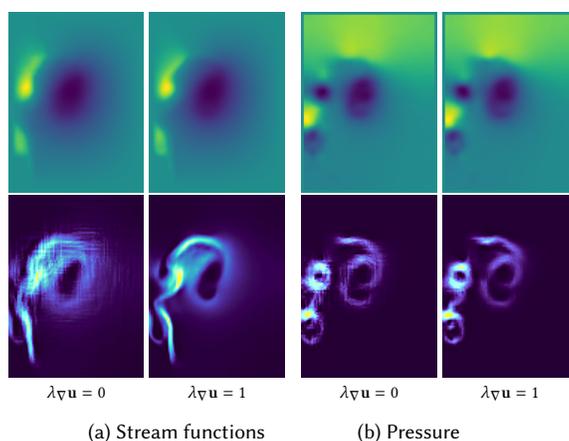


Fig. 9. Stream functions (a) and pressure (b) fields plots. Top images show original fields, while bottom ones show their gradients. We highlight that artifacts appear on those data sets when not using the gradient loss function ($\lambda_{\nabla \mathbf{u}} = 0$).

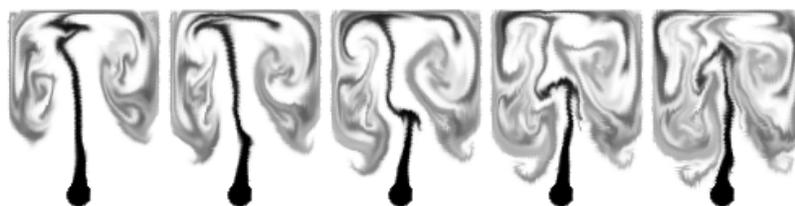


Fig. 10. Density plots advected by velocity fields from reconstructed streamfunctions of the network trained without gradient loss. Jagged artifacts are visible, e.g., near the central stem of the smoke plume.

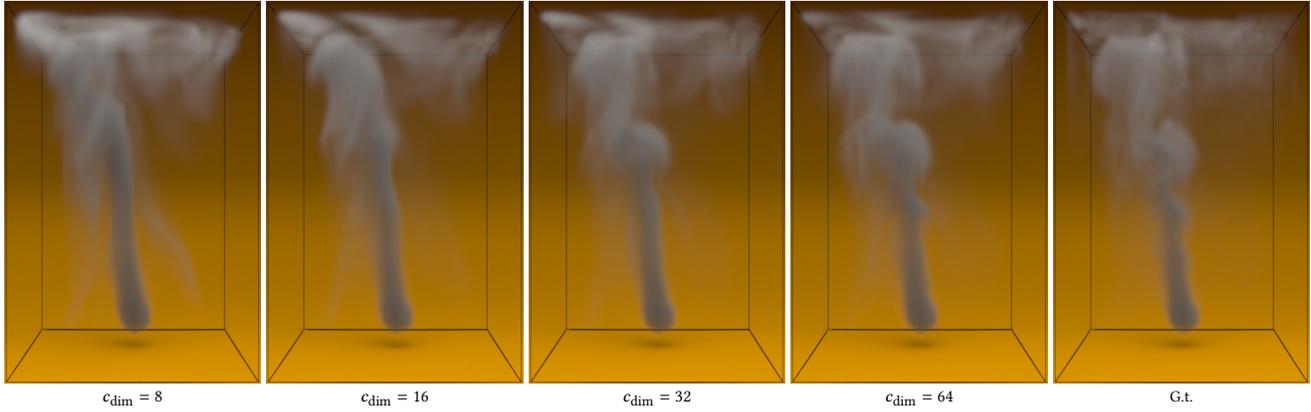


Fig. 11. Comparisons of the quality of autoencoder reconstruction with varying sizes for the latent space c_{dim} . As the latent space dimensionality increases, the quality of the reconstructed smoke plume also increases.

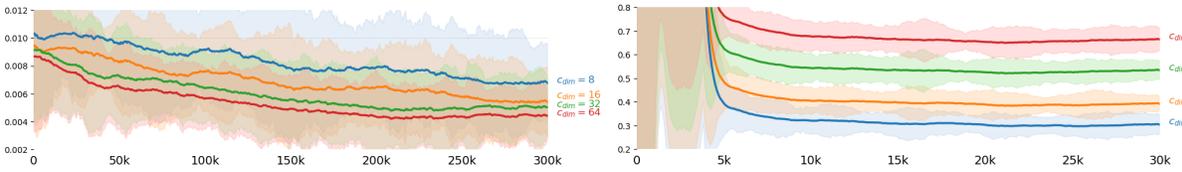


Fig. 12. L_1 loss plot of the autoencoder training (left) and L_2 loss plot of the latent space integration network training (right). Increasing the dimension of the latent space c_{dim} reduces reconstruction errors, while it degrades the accuracy of the integration.

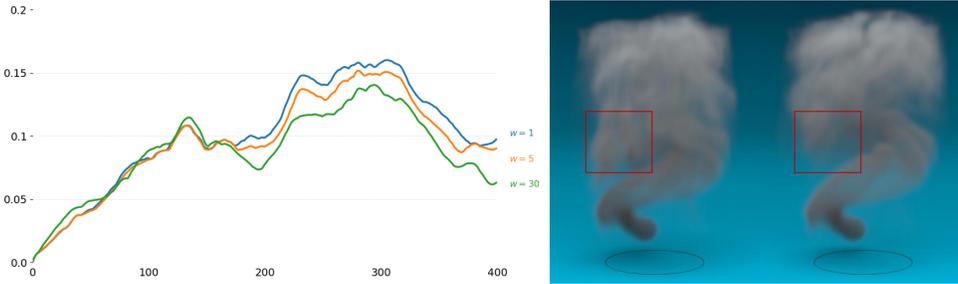


Fig. 13. We show the effect of the training with varying sizes of windows (1, 5 and 30) for the cyclic smoke example. On the left, the L_2 loss plot of the latent space integration network is shown. On the right, we show the visual impact for window sizes of 1 and 30, respectively. Note that there are densities left hanging in mid air in the red box, which are not advected properly due to wrong velocity states recovered by the latent space integration network.