Appendix

1. Implementation details

We provide the implementation details with the code and data in the supplemental material, along with the neural network architectures. We refer to the guideline (i.e., *README.md*) to reproduce the results presented in this work. The code and trained models of our experiments will be published upon acceptance.

2. Experiments

In order to acquire a training data-set for each scenario, we generate a set of solution sequences of the given PDE problem. The PDEs from our experiments work with a continuous velocity field $\mathbf{v}$ in the two-dimensional space, i.e., $\mathbf{v} = [v_x, v_y]^T$. Considering reference simulations on regularly discretized grids, we focus on exploring latent spaces (i.e., reduced representations) that are four times coarser than the reference.

2.1. Karman vortex street

This first example targets a complex PDE problem within a constrained setup, where the velocity field evolves over time while being constrained to be divergence free. We evaluate the incompressible Navier-Stokes equations for Newtonian fluids:

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} - \frac{\nabla p}{\rho} + \nu\nabla^2\mathbf{v} \quad \text{subject to} \quad \nabla\cdot\mathbf{v} = 0 \qquad (1)$$

where p is the pressure, ρ is the density, and ν is the viscosity coefficient. The reference simulation domain is discretized with 128×256 cells and a cell spacing of one using a staggered grid scheme. We use closed boundary conditions for the sides and open boundary conditions for the top of the domain; at the bottom, we set a constant inflow velocity. The continuous inflow collides with a fixed circular obstacle, which creates an unsteady wake flow that evolves differently depending on the Reynolds number. For the temporal discretization, the unit time step size is used. We generate 20 simulations of 200 steps each and randomly choose 5% of them for the validation set and the remaining 95% for the training set. We use Reynolds numbers in {90, 120, 140, 150, 160, 170, 180, 190, 200, 220, 290, 340, 390, 490, 540, 590, 690, 740, 790, 1190}, and we skip the first 2000 time-steps in order to let the flow stabilize. Both the least and most turbulent simulations of the training set are shown in Fig. 1.

In order to make our training more stable, we use pre-trained networks with eight integrated steps as warm starts for our final models. Each training uses 100 epochs with a batch size of ten. The learning rate starts from 4×10^{-4} and exponentially decays with a decaying rate of 0.9 every ten epochs. If divergence happens while training, we restart our training with a smaller learning rate. In this example, we compare all the models trained with 16 integrated steps. We also note that the encoder and adjustment models of *ATO*, the corrector of *SOL*, and the solver of *Dil-ResNet* take the Reynolds number as additional input.

The test set consists of six solution trajectories evaluated with Reynolds numbers in {450, 650, 850, 1050, 1200, 1400}. Example sequences of the test data and the inference results of different models are shown in Fig. 2, for $Re = 850$, along with the spatial distribution of the velocity error in Fig. 3.

As can be seen on Fig. 4, which shows the velocity and vorticity error improvements of each model over the baseline, *ATO* presents the best generalization capabilities. Fig. 5 shows the temporal evolution of the velocity MAE and the distance between each model's reduced space and *lerp(ref)*.

2.2. Decaying turbulence

This example tackles the same incompressible Navier-Stokes equations, but with vortices intialized all over the physical space that slowly decay over time. In this scenario, the viscosity stays constant (equal to 0.1) within the training and test data-sets. The reference simulation domain is discretized with 128^2 cells and a cell spacing of one. Both the discrete velocity and pressure values are stored at the center of each cell, and periodic boundary conditions are applied. For the temporal discretization, a time step size of 1.0 is used. The training data-set consists of 20 simulations of 200 steps each, which evolve from different initial velocity fields. We use randomly selected 5% for the validation set and the remaining 95% for the training set. An example sequence of the data is shown in Fig. 6. As in the Karman vortex street case, we first train our models with eight integrated steps as warm starts for the final models. We compare all the models trained with 16 integrated steps.

The five test trajectories evolve from different initial velocity fields on a domain identical to the training one. Fig. 7 shows the inference results of the different models for one example, and Fig. 8 shows the spatial distribution of the error for the same example. Fig. 9 and Fig. 10 (left) show that our *ATO* model improves the baseline the most in both velocity and vorticity metrics, although *SOL + SR* shows a comparable performance. As shown in Fig. 11 and Fig. 10 (right), its latent space representation is more distant from the linearly down-sampled representation than the other models', yet it shows a similar or better performance.

2.3. Forced turbulence

This case has the same experimental setup as the previous one, but with an external force sequence $\mathbf{g}(\mathbf{x},t)$ that is added to Eq. (1). This force sequence yields complex, chaotic evolutions of vortices over time. We use a different force sequence for each simulation trajectory, composed of 20 overlapping sine functions as follows:

$$\begin{aligned} g_x(\mathbf{x},t) &= \sum_{i=1}^{20} a_i \sin(k_i\alpha_i\cdot\mathbf{x} + w_i t + \phi_i) \\ g_y(\mathbf{x},t) &= \sum_{i=1}^{20} a_i \sin(k_i\alpha_i\cdot\mathbf{x} + w_i t + \phi_i) \end{aligned} \qquad (2)$$

where a_i is the amplitude, k_i is the wave number, α_i is the wave direction, w_i is the frequency, and ϕ_i is the phase shift. These values are randomly sampled from uniform distributions as follows: $a_i \in [-0.1, 0.1]$, $k_i \in \{6, 8, 10, 12\}$, $w_i \in [-0.2, 0.2]$, and $\phi_i \in [0, \pi]$. α_i is a random angle ($\in [0, 2\pi]$). The composed sine functions are, then, evaluated over the domain mapped into $[0, 2\pi]$ for each dimension.

For the temporal discretization, a time step size of 0.2 is used. The training data-set consists of 20 simulations of 200 steps each, which evolve from different initial velocity fields with different force sequences. We use randomly selected 5% for the validation set and the remaining 95% for the training set. An example sequence of the data is shown in Fig. 12. We use the models trained on the previous decaying turbulence case as warm starts for our final models, trained for 100 epochs. We compare all the models trained with 16 integrated steps. The encoder of *ATO* shares its weights for velocity and force in order to learn a unified operation for both reduced representations.

The five test trajectories evolve from both different initial velocity fields and different force field sequences. Fig. 13 shows the inference results of the different models for one example, and Fig. 14 shows the spatial distribution of the error for the same example. Fig. 15 shows that our *ATO* model improves the baseline the most in all five test cases in both velocity and vorticity metrics, while its latent space representation (Fig. 16) is more distant from the linearly down-sampled representation than the other models'.

2.4. Smoke plume

This example represents a smoke volume of a circular shape, which is slowly rising up producing interesting swirling motions. A buoyancy force is produced by a passive marker field with a buoyancy factor of 0.25 applied vertically. The training data-set consists of 20 simulations of 200 steps each with a time-step of 0.2, which start from circular marker fields with a constant radius of 0.12, but evolve differently due to the random initialization of the markers. We use randomly selected 5% for the validation set and the remaining 95% for the training set. An example sequence of the data is shown in Fig. 17. For this case, we use more integrated solver steps than the others. We first train our model with four, eight, and 16 integrated steps as warm starts for the final model. We apply our *ATO* model trained with 32 steps, for 100 epochs. Fig. 18 shows the inferences of our *ATO* model for different initializations.

3. Neural Network Architectures

In this section, we detail our network architectures for each model. We note that the practical implementations of all the models can be found in the supplemental code.

The encoder of the *ATO* setup consists of two convolutional layers with 32 and 16 features each with a kernel size of five. Each convolutional layer is followed by the Leaky ReLU activation function. A last layer with two features and the same kernel size but without the activation infers the final encoded output. This model has approximately 15k trainable weights.

The adjustment of the *ATO* setup and the corrector of *SOL* ([UBF*20]) employ an identical network model. This model consists of a first convolutional layer with 32 features and a kernel size of five, followed by five blocks of two convolutional layers with 32 features each and a kernel size of five. Each layer is followed by the Leaky ReLU activation function, and each block is connected to the next with a skip-connection. A last layer with two features follows with the same kernel size yet without the activation. This architecture has approximately 260k trainable weights.

The decoder used for the *ATO* setup and the super-resolution model from *Dil-ResNet + SR* and *SOL + SR* are adapted from the multi-scale architecture of [FFT19], such that the total number of trainable weights is close to 97k.

The *Dil-ResNet* model is adapted from the architecture of the state-of-the-art network model proposed for turbulent flow problems [SFK*22]. This model has a first convolutional layer with 32 features with a kernel size of three and no activation. It is followed by four identical blocks of seven convolutional layers with 32 features each with a kernel size of three and varying dilation rates from one to eight (respectively: 1, 2, 4, 8, 4, 2, 1). Each layer is followed by the ReLU activation function, and each block is linked to the next via a skip-connection. A last layer with two features follows with the same kernel size yet without the activation. This model has a similar number of trainable weights as the adjustment model's (i.e., 260k). We note that a larger model did not improve the performance. Contrary to the other models, *Dil-ResNet* is trained for only one step at a time and uses a *MSE loss*.

For the Karman vortex street and smoke plume cases, we adopt zero-padding, and for the forced and decaying turbulence cases, which use periodic boundary conditions, we use periodic padding for all models.

4. Training hyper-parameters

In this section, we detail the choice of hyper-parameters for the different trained models.

Firstly, the code of the *SOL* model from [UBF*20] was released publicly, which enabled an easy reproduction of their experiments. Secondly, the learning setup of the *Dil-ResNet* model from [SFK*22] was precisely described in the article. We extensively tested the various hyper-parameters and chose the *Dil-ResNet* model over *Con-Dil-ResNet* (i.e., without the additional loss constraint) because it performed better in our physical scenarios. For training, we chose a Gaussian noise with $\sigma = 0.01$ for all scenarios. Lastly, for our *ATO* setup, we chose the depth of the models by making a compromise between performance and runtime/resources. For the learning rate and batch size, since the physics solver made the models harder to train, we chose the values that best stabilized our training. Our loss being divided in two terms of different orders of magnitude (a high-resolution term and a low-resolution one), we set λ_{hires} to 1 for all scenarios and $\lambda_{latent} = 1$ for the Karman vortex street case, $\lambda_{latent} = 1 = 10$ for the decaying turbulence, $\lambda_{latent} = 100$ for the forced turbulence and $\lambda_{latent} = 10$ for the smoke plume.

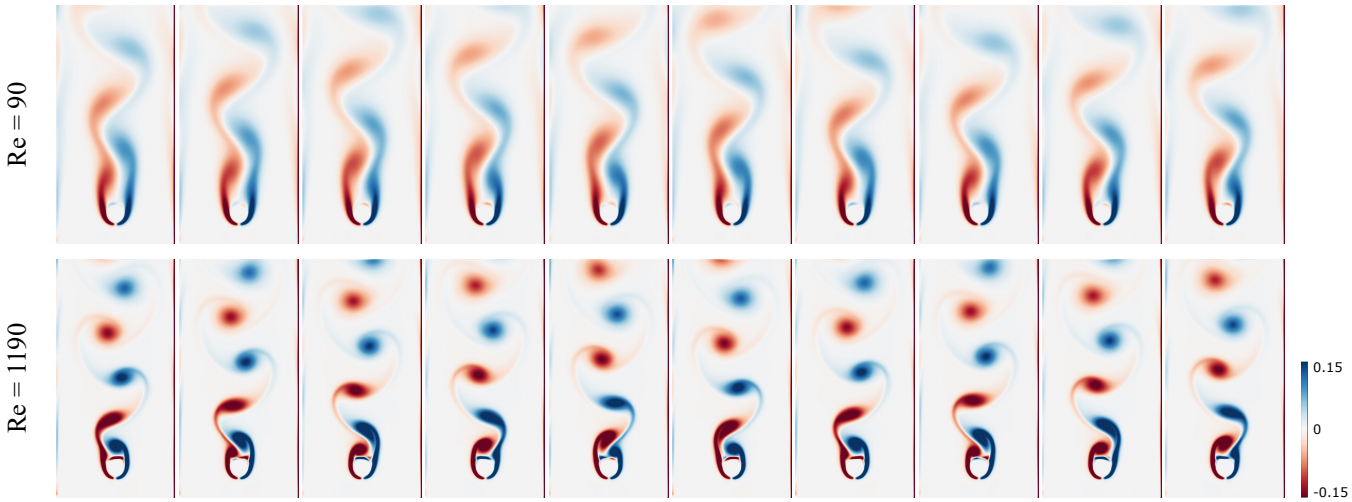

Figure 1: *Two examples from the training data-set of the Karman vortex street scenario: Re = 90 and Re = 1190.*

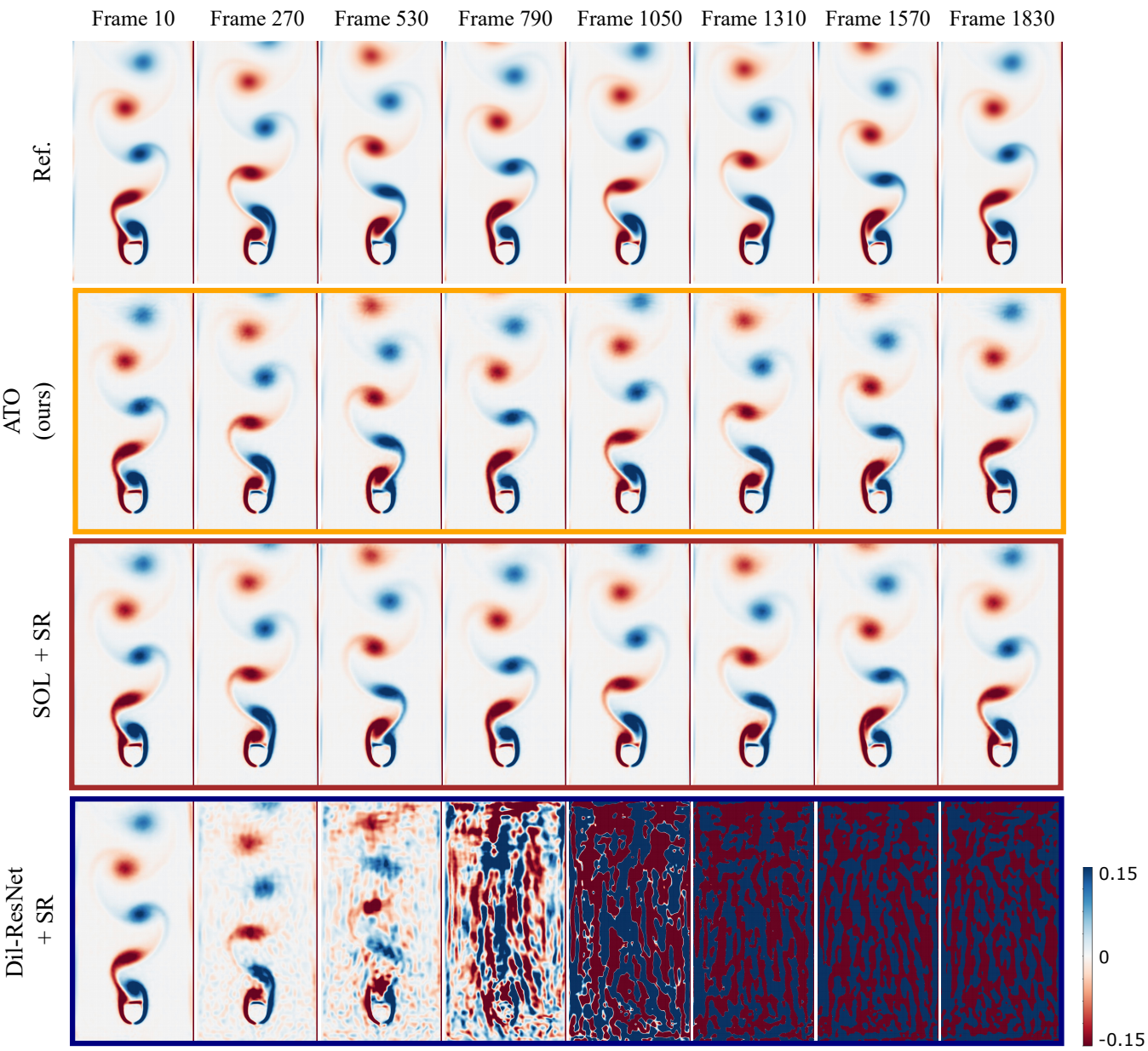

Figure 2: *Restored frames of different models for the Karman vortex street scenario with Re = 850.*

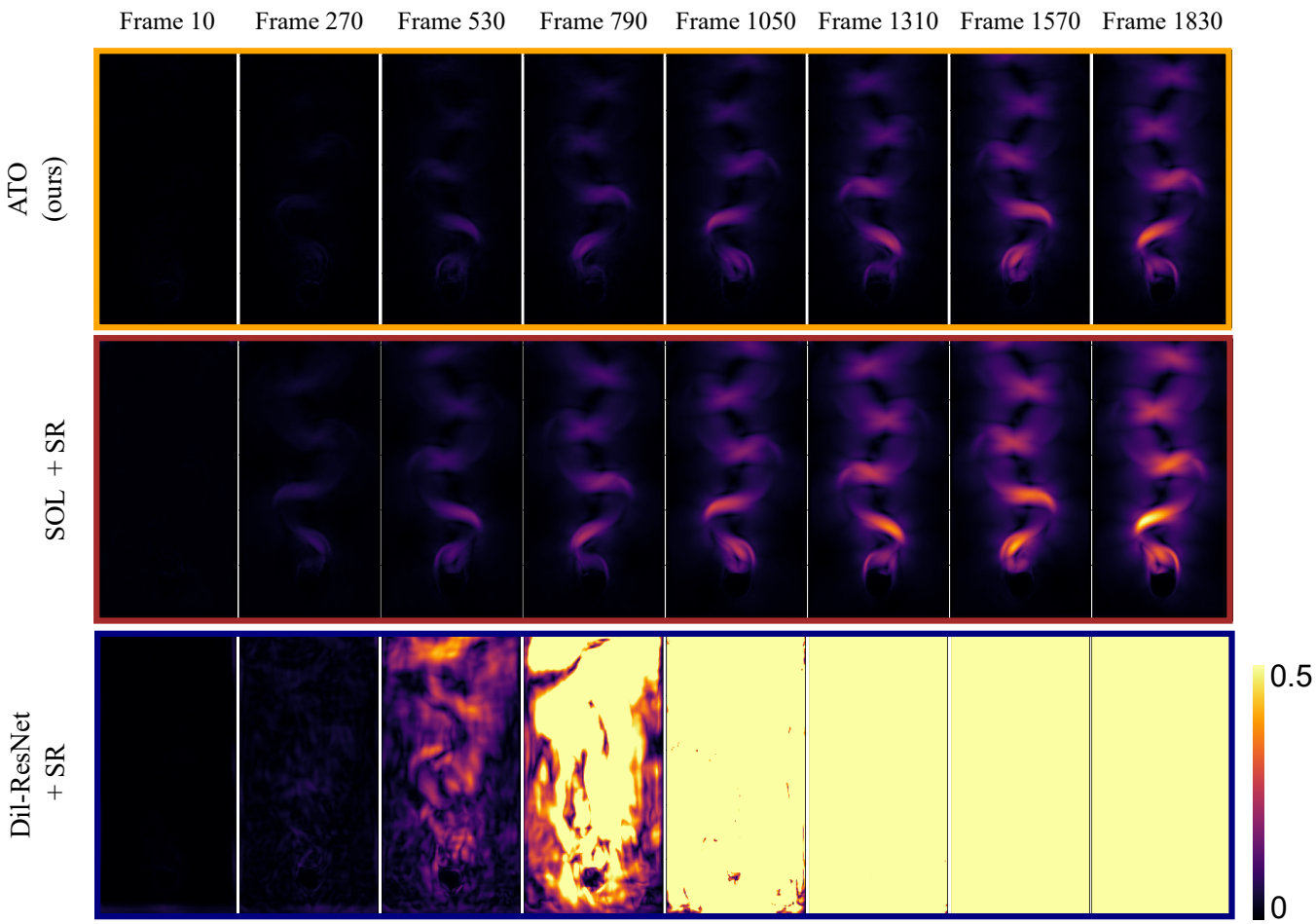

Figure 3: *Absolute error in velocity for the different models, for the Karman vortex street scenario with Re = 850.*

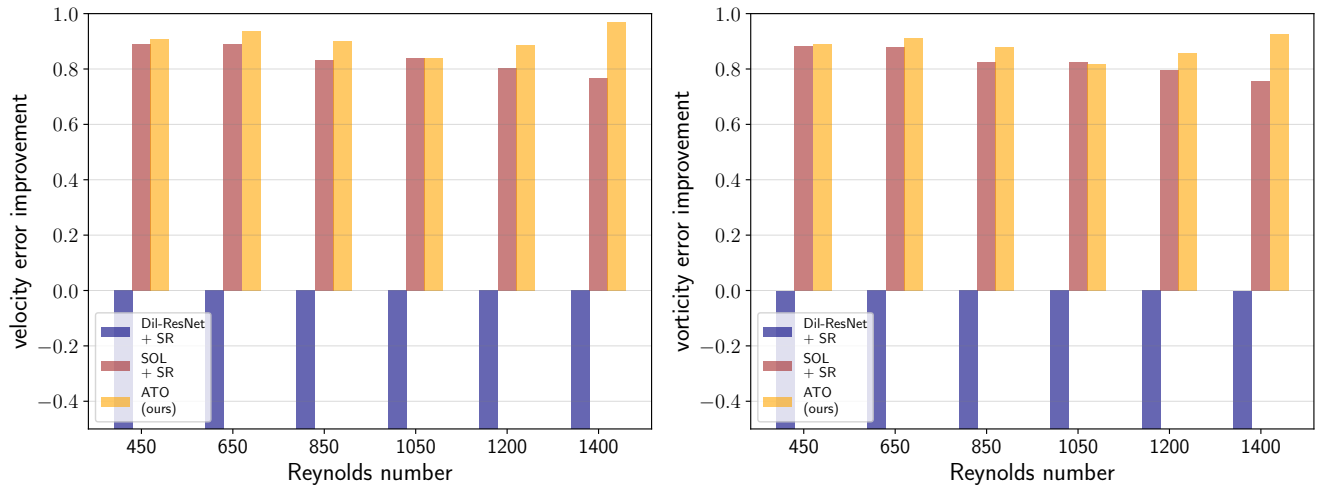

Figure 4: *Velocity (left) and vorticity (right) error improvements for six different Reynolds numbers between 450 and 1400. The highest Reynolds number used for training is 1190. The* ATO *model generalizes better than the others.*

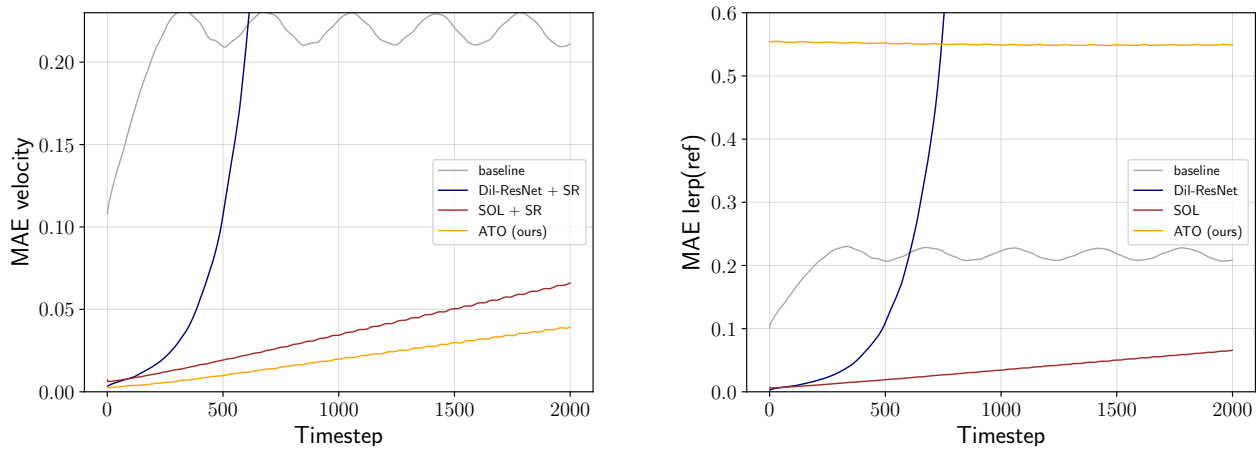

Figure 5: *MAEs of recovered velocities (left) and distances of the reduced spaces to the down-sampled reference (right) over time for the Karman vortex street scenario.*

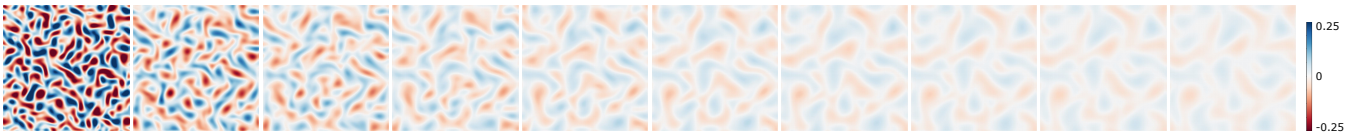

Figure 6: *Example frames from one simulation of the training data-set of the decaying turbulence scenario.*

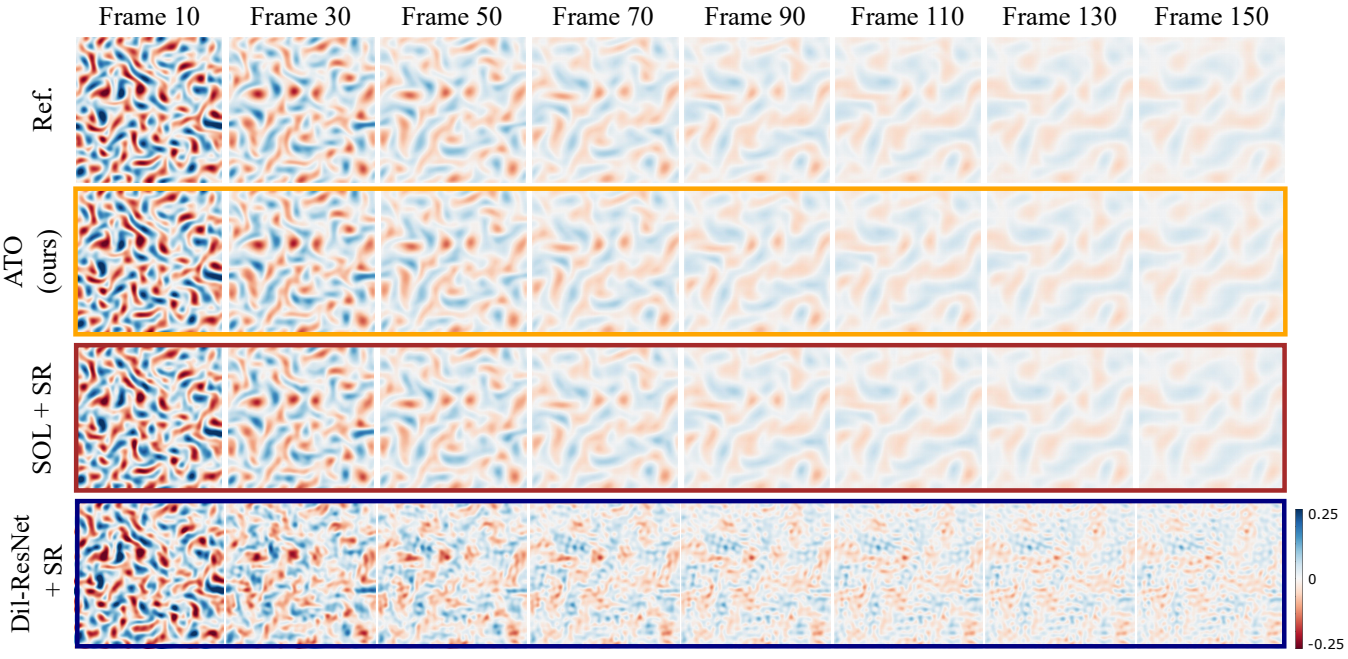

Figure 7: *Example frames of a test case for different models for the decaying turbulence scenario.*

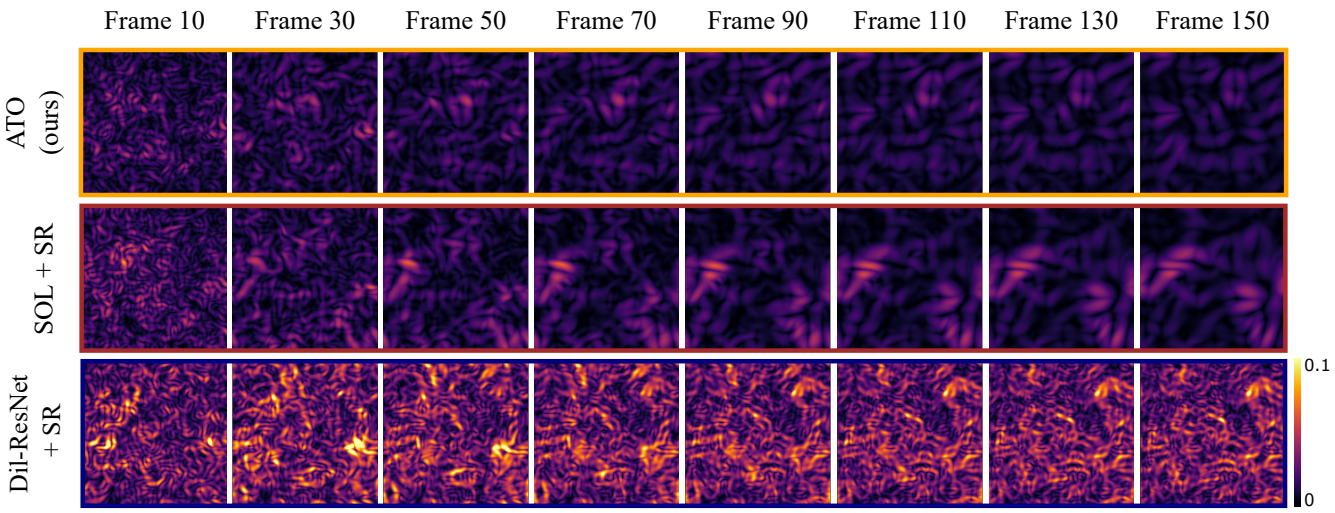

Figure 8: *Absolute error in velocity for the different models, for the decaying turbulence scenario.*

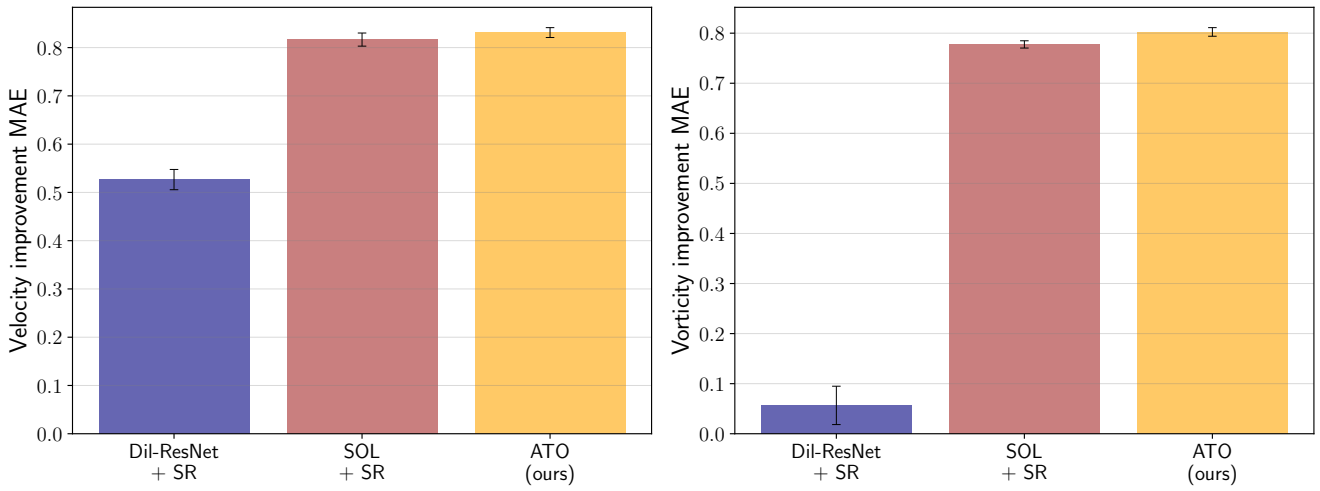

Figure 9: *Velocity (left) and vorticity (right) error improvements for the decaying turbulence scenario. The* ATO *model improves the baseline the most for every test case.*

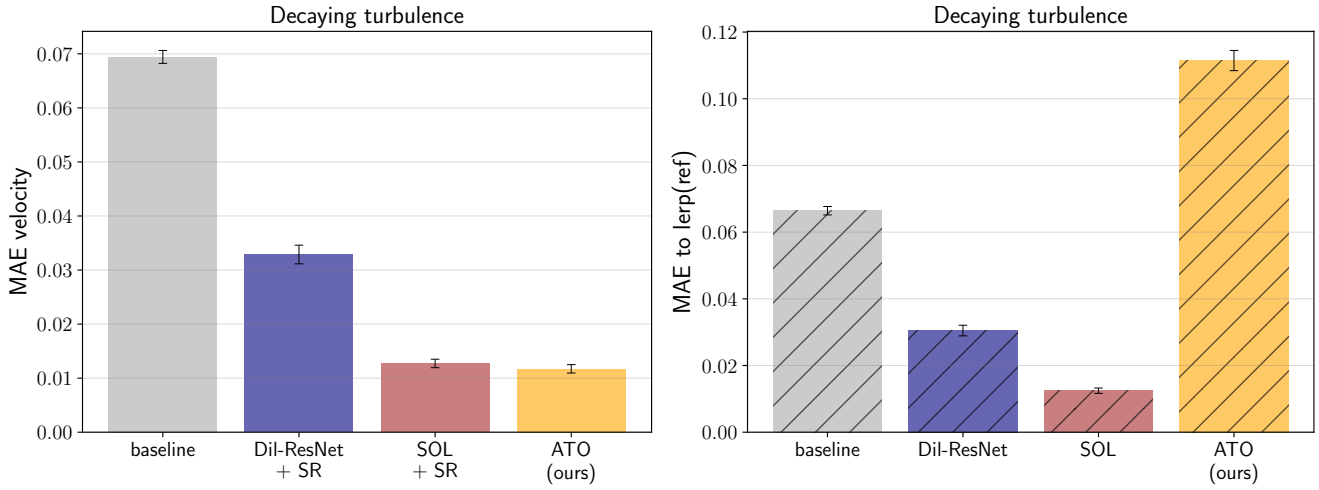

Figure 10: *MAEs of recovered velocities (left) and distances of the reduced spaces to the down-sampled reference (right) for the decaying turbulence scenario.*

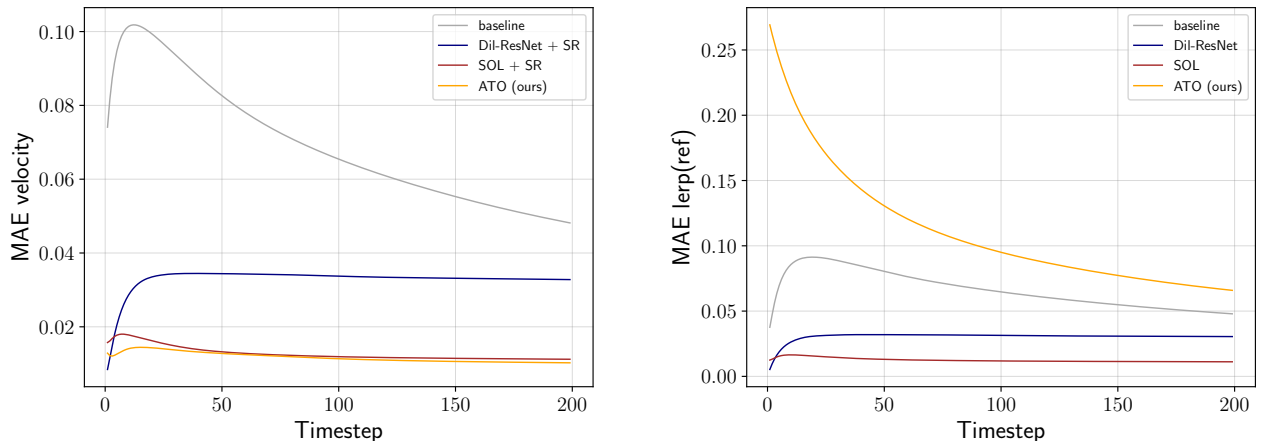

Figure 11: *MAEs of recovered velocities (left) and distances of the reduced spaces to the down-sampled reference (right) over time for the decaying turbulence scenario.*

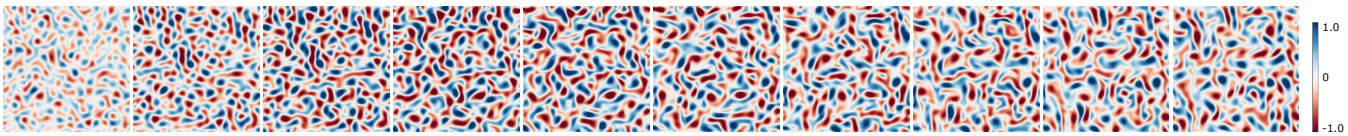

Figure 12: *Example frames from one simulation of the training data-set of the forced turbulence scenario.*

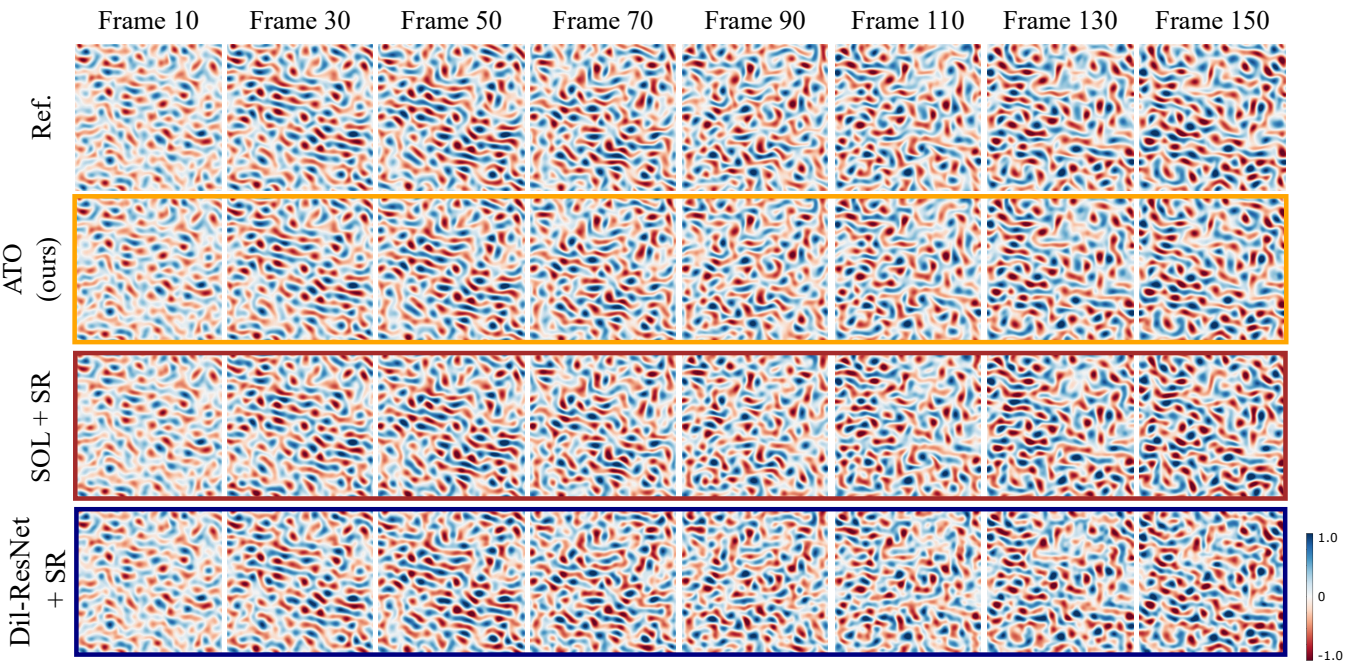

Figure 13: *Example frames of a test case for different models for the forced turbulence scenario.*

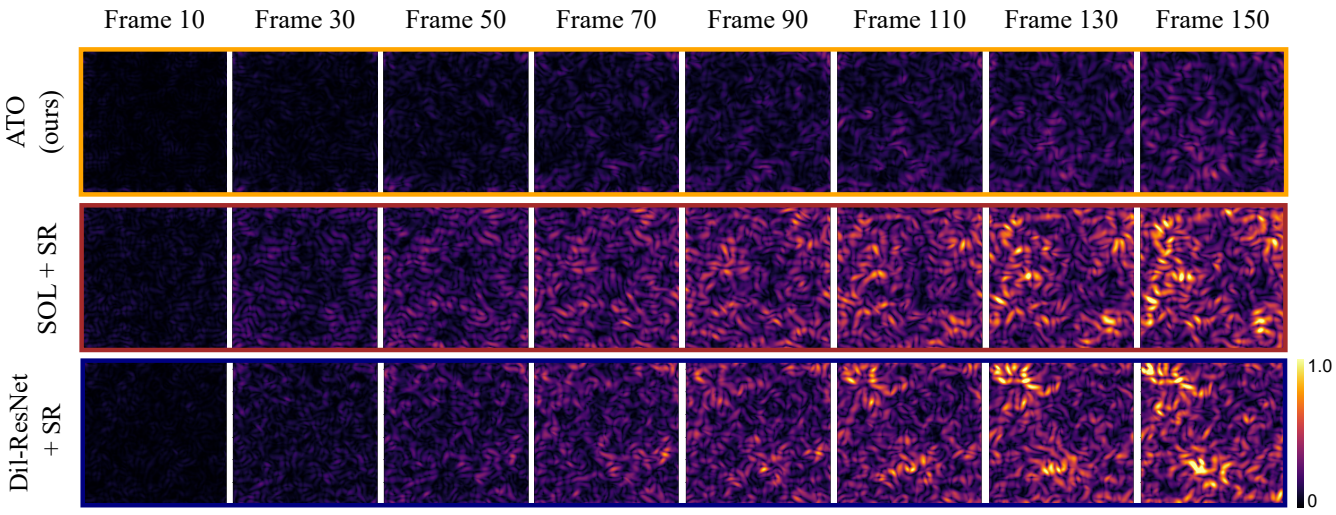

Figure 14: *Absolute error in velocity for the different models, for the forced turbulence scenario.*

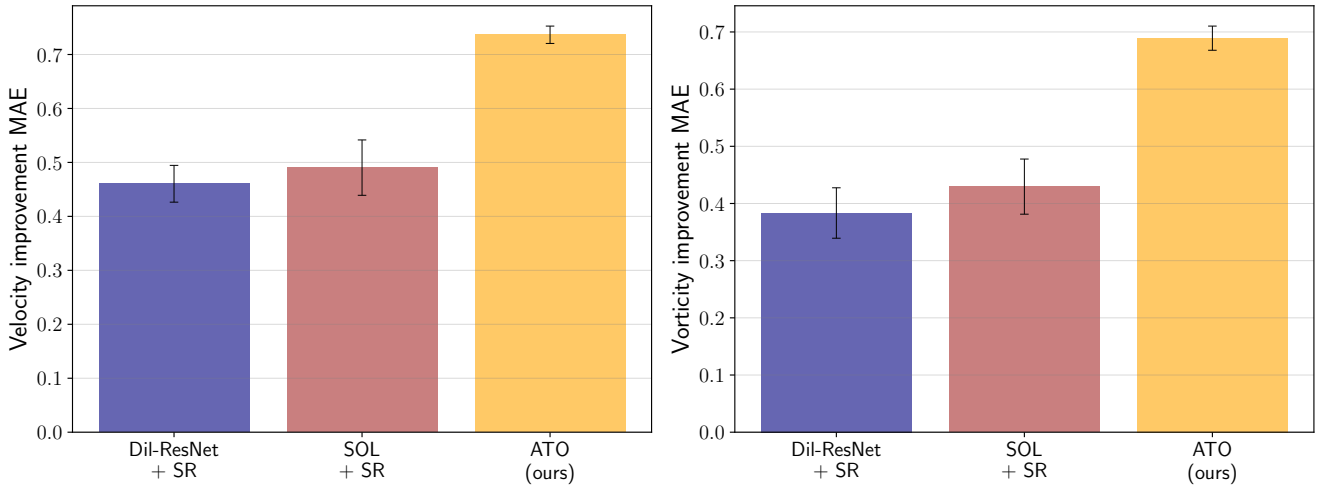

Figure 15: *Velocity (left) and vorticity (right) error improvements for the forced turbulence scenario. The* ATO *model improves the baseline the most for every test case.*

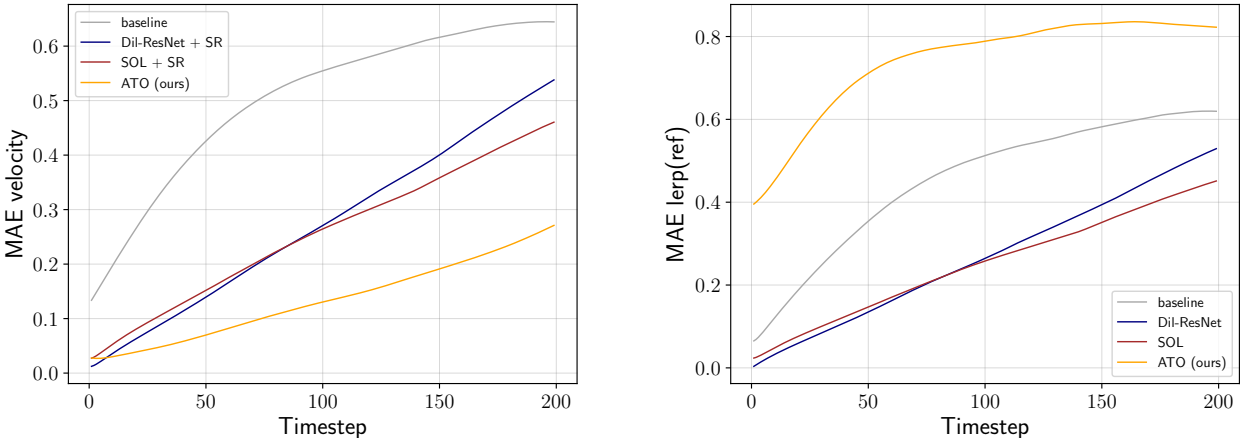

Figure 16: *MAEs of recovered velocities (left) and distances of the reduced spaces to the down-sampled reference (right) over time for the forced turbulence scenario.*

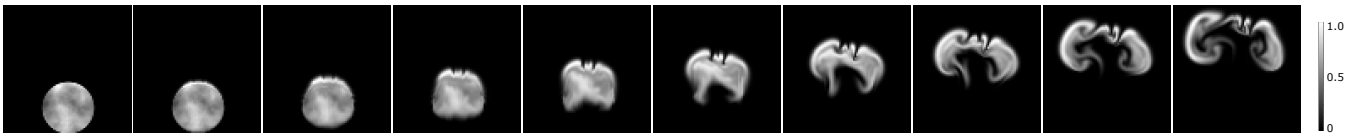

Figure 17: *Example frames from one simulation of the training data-set of the smoke plume scenario.*

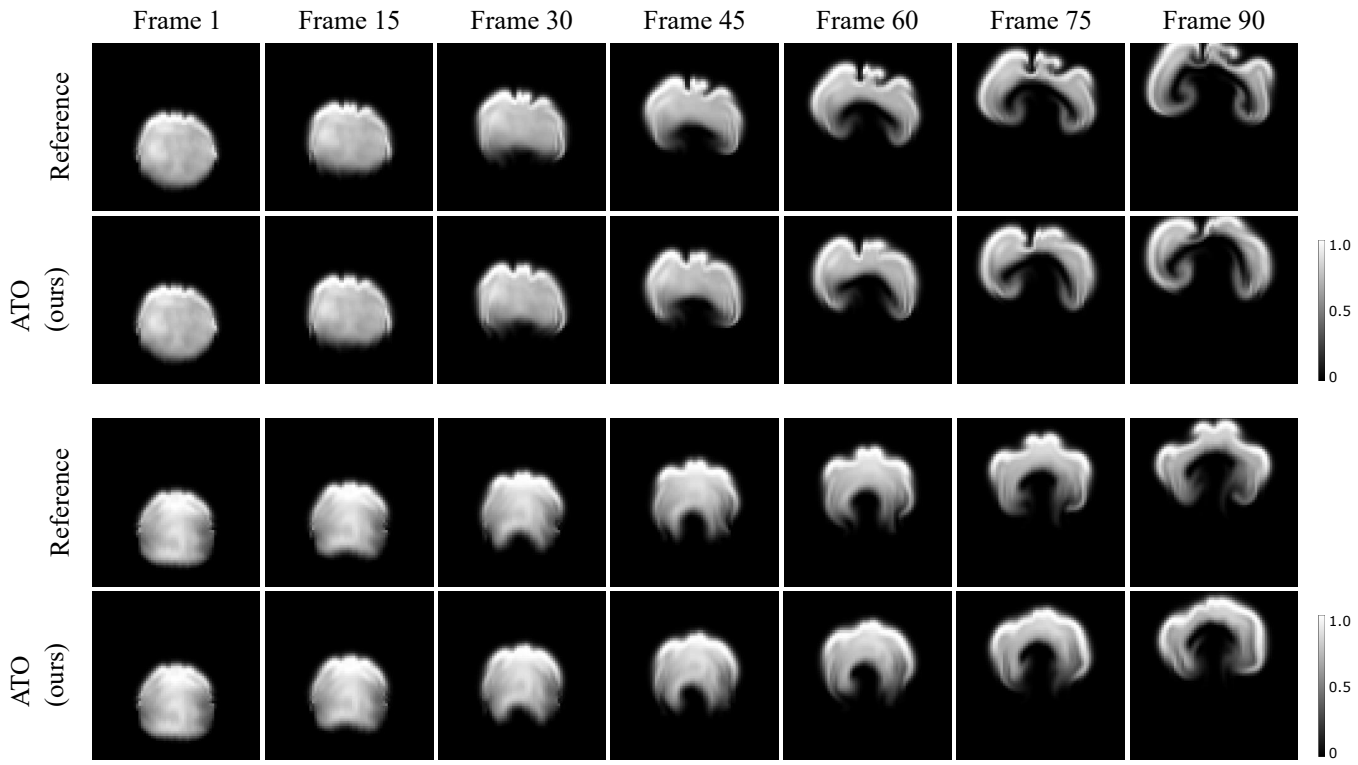

Figure 18: *Example frames from our* ATO *model for two test cases of the smoke plume scenario.*