In the scope of this supplementary material, we provide more details regarding the grid-based data representation and processing as well as the used architecture of the involved network. In addition, we provide more insights into our training domains and present a qualitative comparison of the $\vec{a}$-Net and $\vec{v}$-Net. For the real-time demo, we refer to the attached movie. Finally, we specify more thoroughly the benchmark problem that was used for quantitative analysis as well as the procedure to control the frequency of a Kármán vortex street.

1 MAC grid discretization

As mentioned in section 3.4 of the paper, our method relies on a staggered marker and cell grid representation for the vector potential as well as the velocity and pressure fields. In the following, more details are given on how we apply this representation to learn incompressible fluid dynamics.

To calculate the velocity field $\vec{v} = \nabla \times \vec{a}$ of a vector potential $\vec{a}$ on a MAC grid in 2D, we have to compute the curl as follows:

$$
\begin{align*}
(v_x)_{i,j} &= (a_z)_{i+1,j} - (a_z)_{i,j} \\
(v_y)_{i,j} &= (a_z)_{i,j} - (a_z)_{i,j+1}
\end{align*}
$$

(1)

If this vector potential is inserted into the divergence operator on a MAC grid, we can show that $\nabla \cdot \vec{v}_{i,j} = 0$ is indeed fulfilled:

$$
\nabla \cdot \vec{v}_{i,j} = (v_x)_{i,j+1} - (v_x)_{i,j} + (v_y)_{i+1,j} - (v_y)_{i,j} \\
= ((a_z)_{i+1,j+1} - (a_z)_{i,j+1}) - ((a_z)_{i+1,j} - (a_z)_{i,j}) \\
+ ((a_z)_{i+1,j} - (a_z)_{i+1,j+1}) - ((a_z)_{i,j} - (a_z)_{i,j+1}) \\
= 0
$$

(2)

(3)

Thus for the $\vec{a}$-Net, the incompressibility equation is automatically fulfilled and no further training on the divergence loss $L_d$ is required. However, for the $\vec{v}$-Net, the residuals of the divergence are still of importance:

$$
(R_d)_{i,j}^{t+1} = \nabla \cdot \vec{v}_{i,j}^{t+1} (= 0 \text{ for } \vec{a}\text{-Net})
$$

(5)
The residuals of the momentum equation in \(x\)-direction can be computed as follows:

\[
(R_{px})_{i,j}^{t+1} = \rho \left( \frac[(v_x)_{i,j}^{t+1} - (v_x)_{i,j}^t + (v_x)_{i,j}^t \cdot (v_x)_{i,j+1}^{t'} - (v_x)_{i,j-1}^{t'}}{2} \right) \\
+ \frac{(v_y)_{i,j}^{t'} - (v_y)_{i,j}^{t}}{2} \cdot \left( (v_x)_{i,j}^{t'} - (v_x)_{i,j}^{t} \right) + \frac{(v_y)_{i+1,j}^{t'} + (v_y)_{i-1,j}^{t'}}{2} \cdot \left( (v_x)_{i+1,j}^{t'} - (v_x)_{i,j}^{t} \right) \\
+ \left( p_{i,j}^{t+1} - p_{i,j}^{t} \right) - \mu \cdot \Delta (v_x)_{i,j}^{t} 
\]

(6)

Here, we use the following isotropic Laplace operator:

\[
\Delta s_{i,j} = \frac{1}{4} \left( 1 \ast s_{i-1,j-1} + 2 \ast s_{i,j-1} + 1 \ast s_{i-1,j+1} + 1 \ast s_{i,j+1}\right) \\
+ 2 \ast s_{i-1,j} - 2 \ast s_{i,j} + 2 \ast s_{i,j+1} \\
+ 2 \ast s_{i+1,j-1} + 2 \ast s_{i+1,j+1} \\
+ 1 \ast s_{i+1,j+1} 
\]

(7)

The derivation of the advection term for \(R_{px}\) is a bit more complex since on a MAC grid, \(v_x\) and \(v_y\) are displaced by half a pixel in \(x\)-direction and \(y\)-direction. To obtain the residuals of the momentum equation in \(y\)-direction, \((R_{py})_{i,j}\), one has to take \((R_{px})_{i,j}\) and swap \(x\) and \(y\) and the indices respectively.

Now, the discretized loss terms can be written as follows:

\[
L_d^{t+1} = \sum_{i,j} \Omega_{i,j}^{t+1} ((R_{d})_{i,j}^{t+1})^2 
\]

(8)

\[
L_p^{t+1} = \sum_{i,j} \Omega_{i,j}^{t+1} \left( ((R_{px})_{i,j}^{t+1})^2 + ((R_{py})_{i,j}^{t+1})^2 \right) 
\]

(9)

\[
L_b^{t+1} = \sum_{i,j} \Omega_{i,j}^{t+1} \left\| v_{i,j}^{t+1} - \bar{v}_{i,j}^{t+1} \right\|^2 
\]

(10)

Note, that all mentioned operations can be efficiently implemented with convolutions. To obtain the final velocities on a square grid, we project the velocity fields of the MAC grid back onto the \(\vec{a}\)-grid using linear interpolation:

\[
\bar{v} = \frac{1}{2} \left( (v_x)_{i-1,j} + (v_x)_{i,j} \right) \\
\]

(11)

### 2 Network architecture

Our fluid model is based on the U-Net architecture with a reduced number of channels (see Figure[1]). As the pressure field and vector potential can have an arbitrary offset, we always normalize the mean of the pressure \((\Delta p)\) and vector potential \(\Delta a_z\) to 0 to keep these fields well defined and prevent drifting offset values.

### 3 Examples of Training Domains

The domains we used for training consist of \(100 \times 300\) grids. We used 3 different randomized domains as exemplary depicted in Figure[2]. First, we have boxes with randomized height and width that float on randomized paths inspired by Brownian motion in a pipe with randomized flow speed. Second, we have the same setup but replaced the boxes by cylinders with randomized radii and angular velocities in order to learn the Magnus effect. Finally, we have a folded pipe system with randomized flow speed, that is randomly flipped along the \(x\)-axis.
Figure 1: U-Net architecture with reduced number of channels.

Figure 2: The left column shows $\Omega$ (in white) / $\partial \Omega$ (in black) and the right column shows $\vec{v}_d$ for three examples of training domains. (Colors indicate direction of $\vec{v}_d$ as described in Figure 4(a))

Figure 3: Qualitative comparison of $\vec{a}$-Net and $\vec{v}$-Net
Figure 4: Screenshots of our real-time demo movie. Colors represent flow direction and speed. Grey values represent pressure (bright: high pressure, dark: low pressure).

4 Qualitative Comparison of $\vec{a}$-Net and $\vec{v}$-Net

We already demonstrated in the accompanying paper that the $\vec{a}$-Net performs quantitatively significantly better than the $\vec{v}$-Net. Here, we also want to give a qualitative example to show the benefits of using a vector potential. Figure 3 demonstrates that the $\vec{a}$-Net finds plausible solutions for the folded pipe domain while the $\vec{v}$-Net looses most of the flow in the center of the domain. The folded pipe domain is particularly difficult to learn as the flow field contains long range dependencies to the inlet and outlet (as shown in the bottom row in Figure 2).

5 Real-time demo

The attached video presents results that were generated in real-time on a $100 \times 300$ grid using the $\vec{a}$-Net. Figure 4 gives some impressions of the accompanying video, where we show how effects from fluid dynamics such as Kármán vortex streets or the Magnus effect can be reproduced. We demonstrate simulation results for different Reynolds numbers, i.e., different ratios of inertial forces to viscous forces, as well as the generalization capabilities to previously unseen object shapes (shark, car, smiley). Note that the simulation provides plausible results even for the unconnected elements of the smiley, indicating that our technique also shows potential for multiple objects within the fluid.

6 Quantitative analysis: Specifications of the benchmark problem

Figure 5 shows the domain $\Omega$ and $v_d$ on a $100 \times 100$ grid which was used as the benchmark problem for quantitative analysis. The flow speed for the inlet and outlet was set to 0.5. The timestep of the integrator was set to $dt = 4$ and the viscosity and fluid density were set to $\mu = 0.1$ and $\rho = 4$ respectively.
7 Controlling Kármán vortex streets

In section 4.3 of the paper, we already briefly outlined the pipeline that was used to control the frequency of a Kármán vortex street. In the following we want to give some more details about the domain and the loss function. Figure 6(a) shows one frame of the Kármán vortex street that we aim to control. The domain consists of a 100x200 grid and we evolve the system for 300 timesteps. The white box marks the area where we measure the mean of $v_y(t)$. Figure 6(b) and 6(c) show the evolution of $v_y(t)$ over time after optimization for low ($\hat{f} = 2$) and high ($\hat{f} = 8$) target frequencies. The system starts from zero velocity and pressure fields which can be noticed by the small oscillation amplitudes of $v_y(t)$ at the beginning of the simulations. Figure 6(d) and 6(e) show the target Gaussian functions and the normalized $\|V_y(f)\|^2$ curves that were obtained by applying a differentiable fast Fourier transform to $v_y(t)$. After around 50 optimization iterations using the Adam optimizer (lr=0.1), convergence is reached and the maxima of $\|V_y(f)\|^2$ are well aligned with the target frequencies $\hat{f}$. 

Figure 5: The left column shows $\Omega$ (in white) / $\partial\Omega$ (in black) and the right column shows $\vec{v}_d$ for the benchmark problem. (Colors indicate direction of $\vec{v}_d$ as described in Figure 4(a)).
((a)) Simulation of Kármán vortex street. The white bounding box marks the measurement area of $v_y(t)$

((b)) $v_y(t)$ for flow speed 0.221

((c)) $v_y(t)$ for flow speed 0.809

((d)) $\|V_y(f)\|^2$ for $\hat{f} = 2$

((e)) $\|V_y(f)\|^2$ for $\hat{f} = 8$

Figure 6: Controlling the frequency of Kármán vortex streets.