A. Spatial distribution of the divergence error

To complement this study in Section 5, the divergence distribution at four timesteps is shown here. Note that these divergence percentiles show the spatial distribution of the divergence, but they do not give any information about the absolute value of the studied error.

Figure 23 represents the divergence distributions at 4 timesteps for the no cylinder case where no hybrid strategy is used, i. e., the divergence distribution of the raw output of the four networks. Qualitatively, all networks behave similarly, as the main divergence sources are found on the plume head and vortices. However, when closely analyzed, each network counts with its distinct behavior related to the network architecture. The MultiScale and SmallScale architectures behave similarly, showing a smooth divergence distribution, with the presence of a divergence stride close to the domain edges, probably related to the network padding, while the Unet network seems to struggle predicting the pressure field close to the walls (particularly the top and left wall). Moreover, the MonoScale network shows the smoothest divergence distribution, even if its flow topology clearly shows a faster plume propagation.

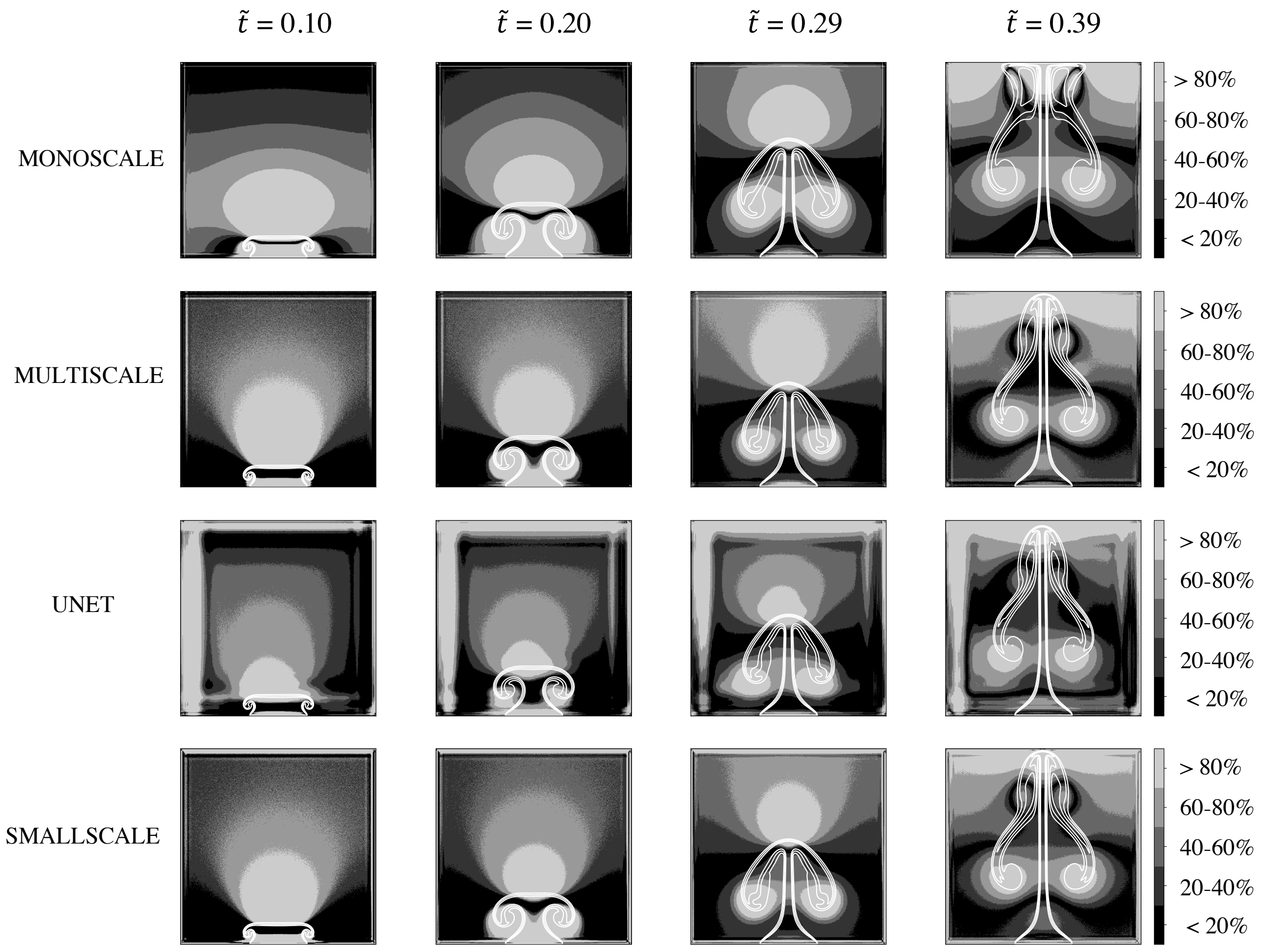

Figure 23. *Divergence percentiles and density iso-contours (in white) of the four studied networks for the no cylinder case at timesteps $\tilde{t}$ = 0.10, 0.20, 0.29 and 0.39 when the networks prediction are not coupled with the hybrid strategy.*

Similarly, Fig 24 represents the same divergence distribution but on the cylinder case. The divergence distribution follows a similar behavior as the one observed in Fig. 23. However, the presence of the cylinder highlights potential difficulties of networks to predict flow-wall interactions. In particular, it is shown that the Unet network particularly faces such a difficulty, as the major part of its error concentrates near the cylinder wall. Analyzing the flow topology shows that the MonoScale network creates larger structures, probably due to an incorrect prediction of the baroclinic-torque.

Figures 25-28 show the divergence distribution and density iso-contours of the four studied networks, as well as the Jacobi solver with the hybrid approach. Figures 25 and 26 correspond to the no cylinder test case with $\mathcal{E} = \mathcal{E}_\infty$ (Fig. 25) and $\mathcal{E} = \mathcal{E}_1$ (Fig. 26). As the threshold level is set to the minimum value obtained by any networks without any Jacobi iterations, the Unet

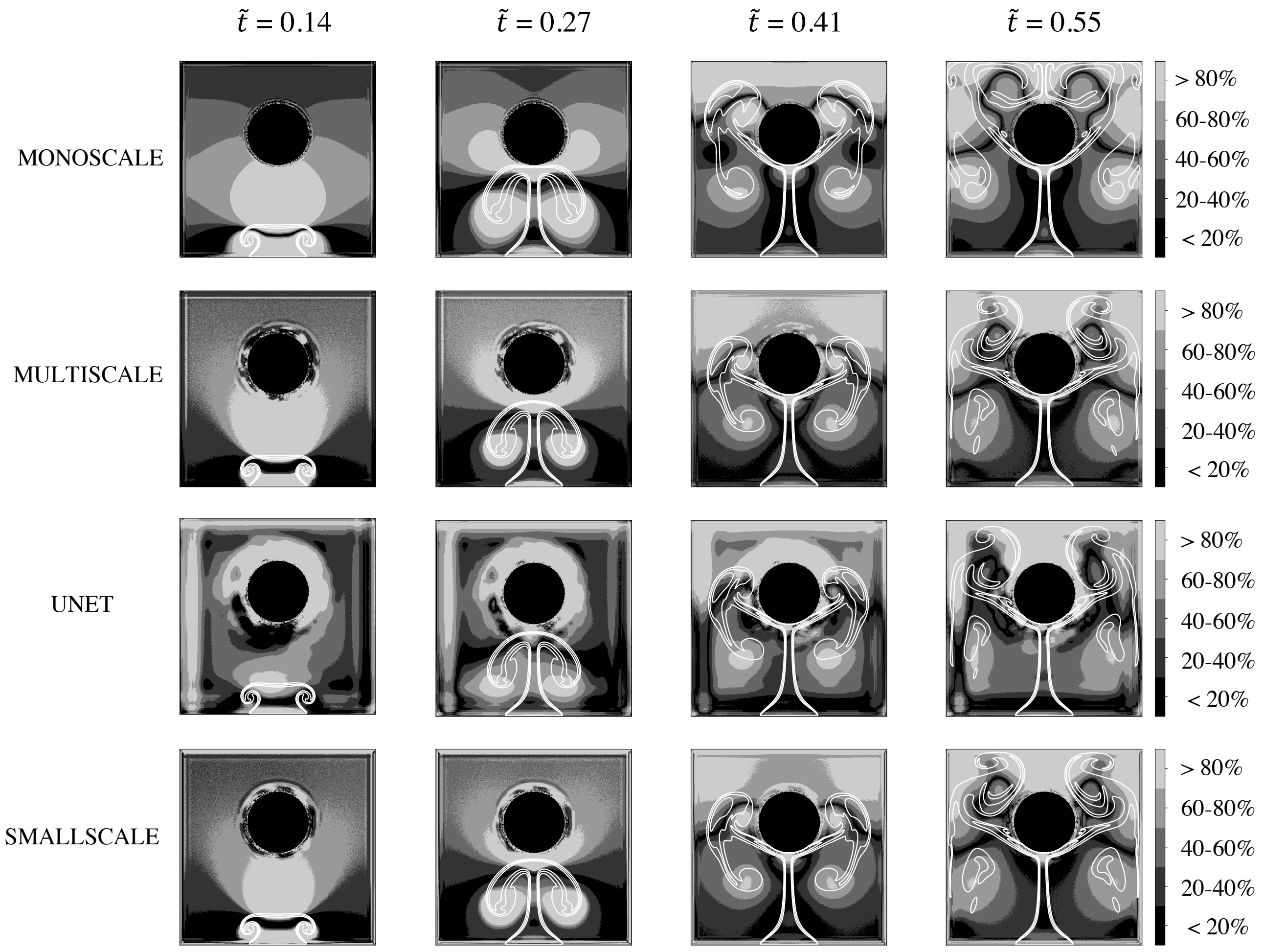

Figure 24. *Divergence percentiles and density iso-contours (in white) of the four studied networks for the cylinder case at timesteps $\tilde{t}$ = 0.14, 0.27, 0.41 and 0.55 when the networks prediction are not coupled with the hybrid strategy.*

network is usually the one needing the least number of Jacobi iterations. Thus, the Unet's output closely matches the raw output, which shows the error concentration on the top and left wall, yet the global flow topology is consistent with other networks.

Figures 27 and 28 correspond to the cylinder test case for both thresholds $\mathcal{E} = \mathcal{E}_\infty$ (Fig. 27) and $\mathcal{E} = \mathcal{E}_1$ (Fig. 28). As the less accurate network, the MonoScale network is mainly controlled by the Jacobi iterations, which results in similar flow topologies between the MonoScale and the Jacobi solver. As previously mentioned, the Unet network concentrates its error around the cylinder, as it struggles to correctly predict the pressure field around objects. This results in a quite different error distribution, even when $\mathcal{E} = \mathcal{E}_1$, as even if the Unet network needs some Jacobi iterations to reach the specified divergence threshold, the resulting error field is still controlled by the error concentrated around the object and walls. However, and as previously mentioned, even if the divergence distribution is different, the flow topology perfectly matches the resulting flow of the other networks.

So to sum up, it is clear that following the maximum error with the hybrid approach yields different plume behaviors, with a too fast rising for the MonoScale network and the Jacobi method, whereas the mean error provides consistent numerical simulations whatever the network architectures. For all cases and methods, several zones of high errors can be distinguished: (i) at the inlet where the lighter fluid is injected, (ii) above the plume head, and (iii) at the core of the vortices produced by the baroclinic torque, in particular at later times ($\tilde{t}$ = 0.29 and 0.39) for the no cylinder test case. In that context, the Unet is producing additional error patterns, with high errors close to the boundaries, either near the cylinder walls or close to the CFD domain edges. Moreover, the error distribution in space is not symmetric, compared with all other methods. Note however that percentiles do not reflect the absolute error level, and the Unet has shown good overall accuracy for both cases, even in the presence of the cylindrical obstacle.

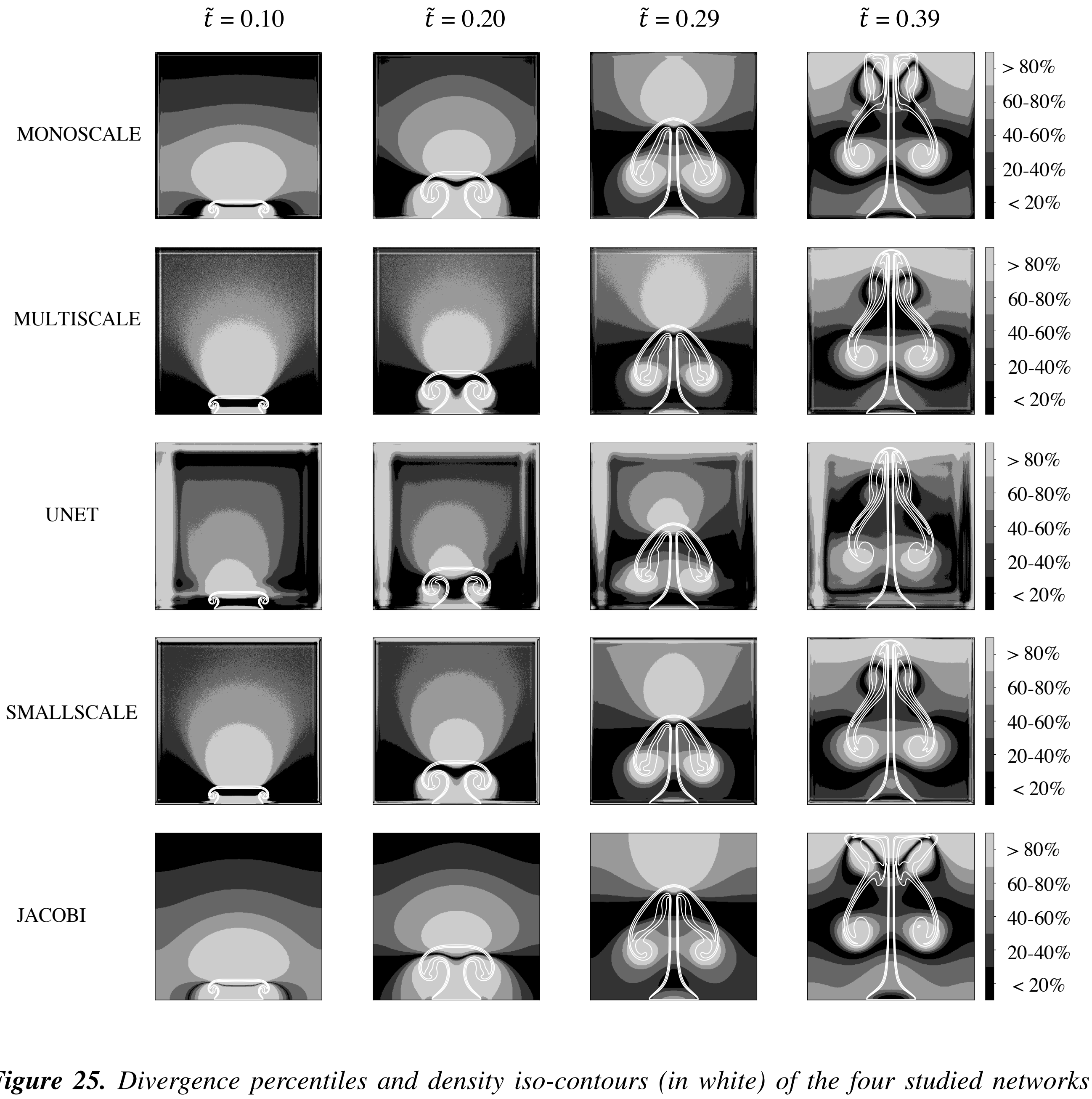

Figure 25. *Divergence percentiles and density iso-contours (in white) of the four studied networks for the no cylinder case at timesteps $\tilde{t}$ = 0.10, 0.20, 0.29 and 0.39 when the networks prediction are coupled with the hybrid strategy. The chosen threshold is $\mathcal{E} = \mathcal{E}_\infty$, where $\mathcal{E}_t = min(\mathcal{E}_\infty)$ obtained by any network at a given timestep, without being coupled with the hybrid strategy.*

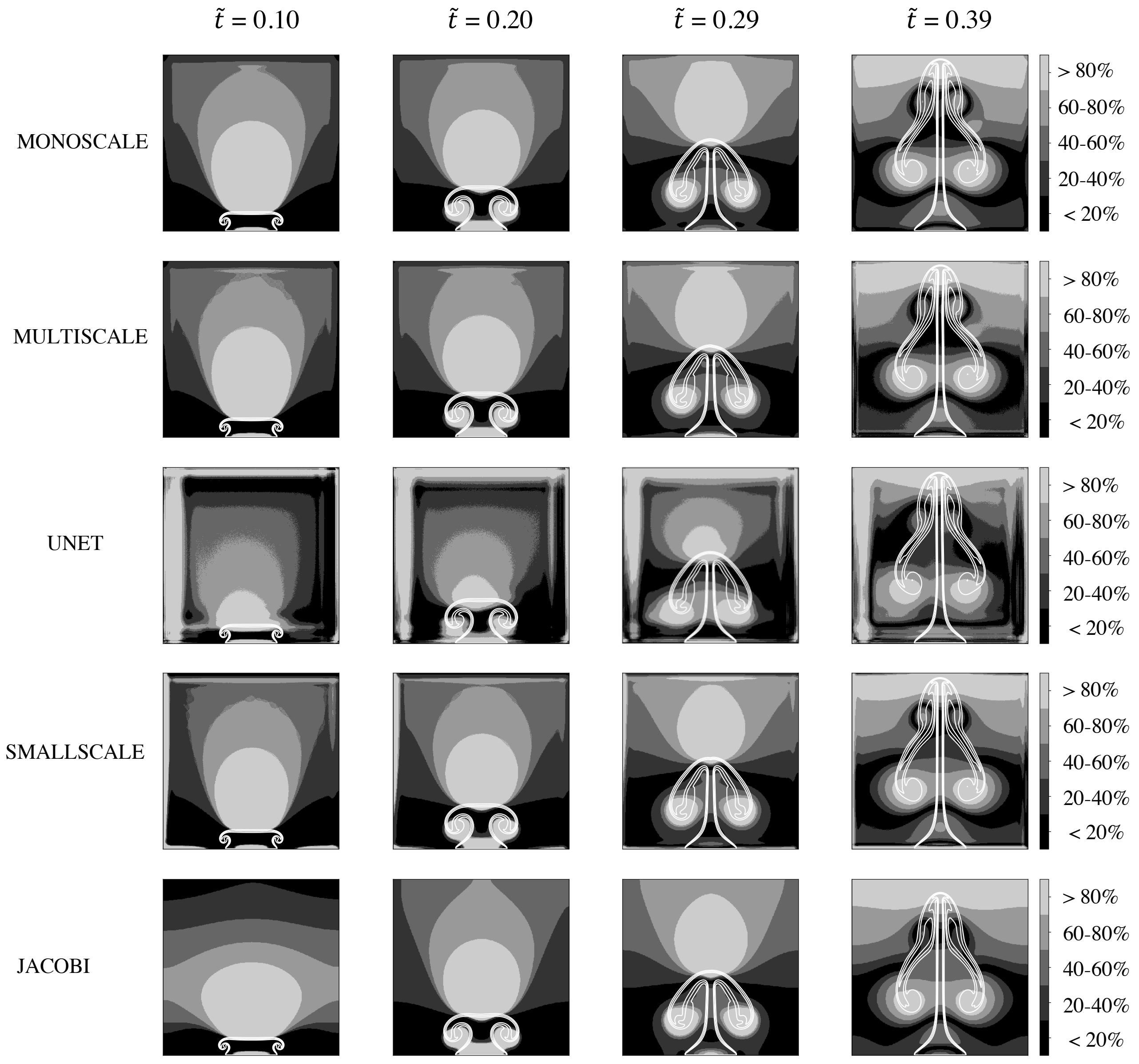

__Figure 26.__ Divergence percentiles and density iso-contours (in white) of the four studied networks for the no cylinder case at timesteps $\tilde{t}$ = 0.10, 0.20, 0.29 and 0.39 when the networks prediction are coupled with the hybrid strategy. The chosen threshold is $\mathcal{E} = \mathcal{E}_1$, where $\mathcal{E}_t = min(\mathcal{E}_1)$ obtained by any network at a given timestep, without being coupled with the hybrid strategy.

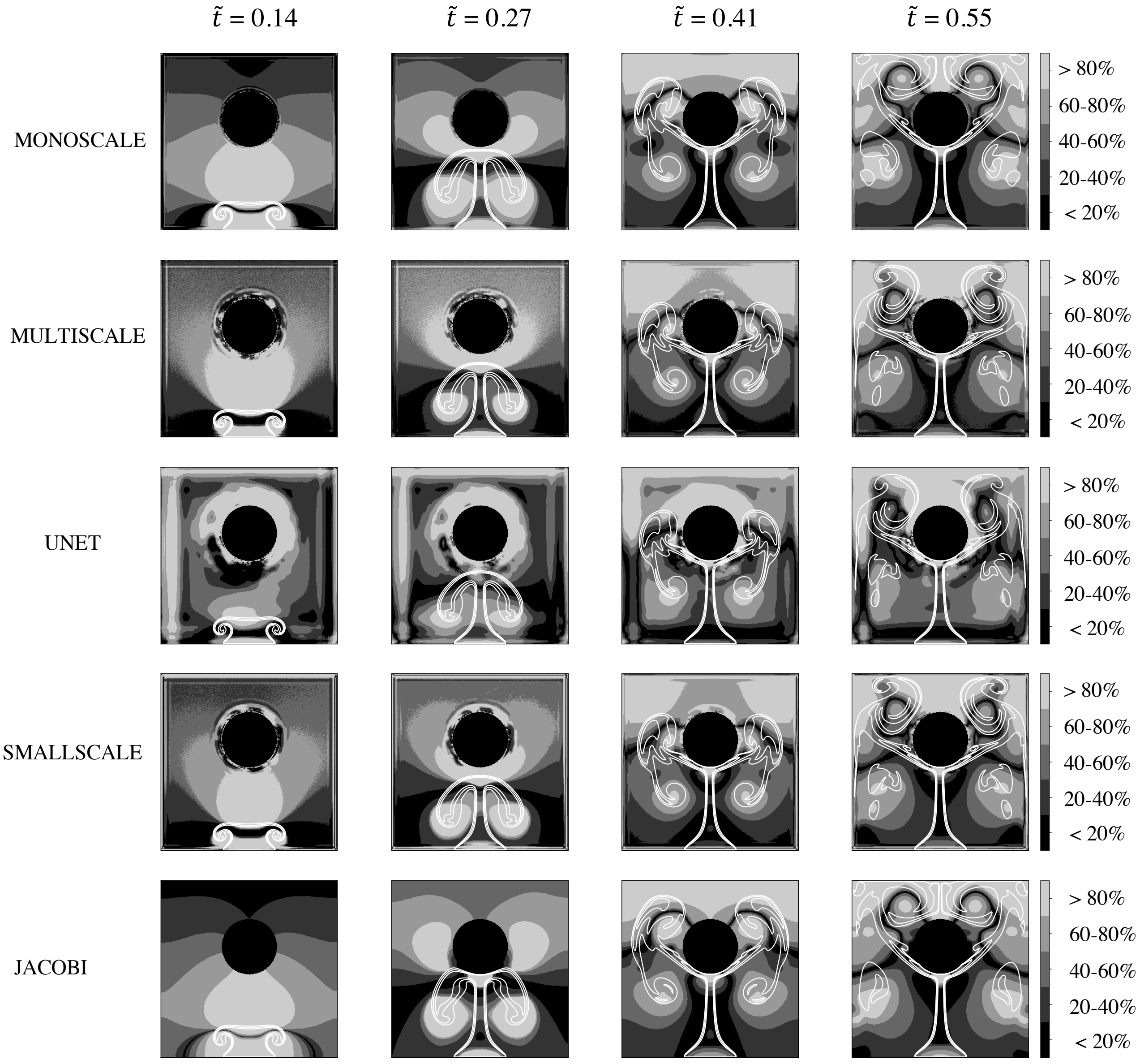

Figure 27. *Divergence percentiles and density iso-contours (in white) of the four studied networks for the cylinder case at timesteps $\tilde{t}$ = 0.14, 0.27, 0.41 and 0.55 when the networks prediction are coupled with the hybrid strategy. The chosen threshold is $\mathcal{E} = \mathcal{E}_\infty$, where $\mathcal{E}_t = min(\mathcal{E}_\infty)$ obtained by any network at a given timestep, without being coupled with the hybrid strategy.*

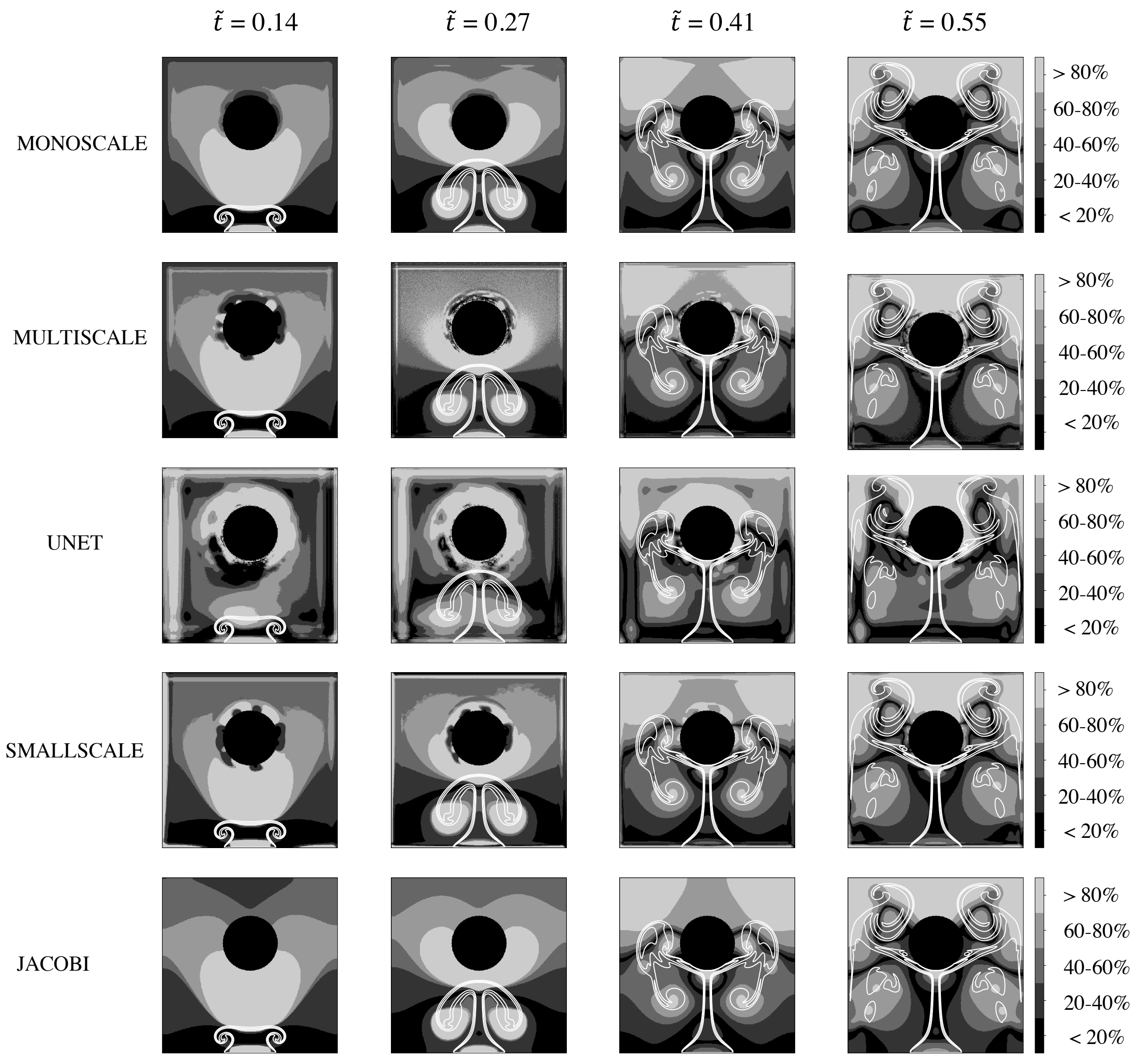

Figure 28. *Divergence percentiles and density iso-contours (in white) of the four studied networks for the cylinder case at timesteps $\tilde{t}$ = 0.14, 0.27, 0.41 and 0.55 when the networks prediction are coupled with the hybrid strategy. The chosen threshold is $\mathcal{E} = \mathcal{E}_1$, where $\mathcal{E}_t = min(\mathcal{E}_1)$ obtained by any network at a given timestep, without being coupled with the hybrid strategy.*

B. Sensitivity of the hybrid solver on the threshold value $\mathcal{E}_t$

The evolution of the head position with a lower threshold is shown in Fig. 29 for the plume test case without obstacle. If the maximum threshold level is set to $\mathcal{E}_t = 0.37$ (Fig. 29-a), which corresponds to a divergence threshold lower than the maximum threshold that could be obtained with any network (Fig. 13), the plume does not follow a homogeneous behavior. In particular, the Jacobi method and MonoScale network produce a higher divergence level. If the threshold is further decreased by a factor 2 ($\mathcal{E}_t = 0.18$, Fig. 29-b), all methods now produce simulations with the same evolution of the plume head position. To further understand the condition which dominates the flow behavior, the mean divergence of the velocity field at each time step is displayed with dotted lines in Fig. 29. It shows the evolution of $\mathcal{E}_1$ for both threshold values $\mathcal{E}_t = 0.37$ (a) and $\mathcal{E}_t = 0.18$ (b). As the maximum threshold is decreased, the spatially-averaged error gets similar for all networks, resulting in the same plume evolution in time. Note that discrepancies are still observed between these errors, the Unet outperforming all other networks.

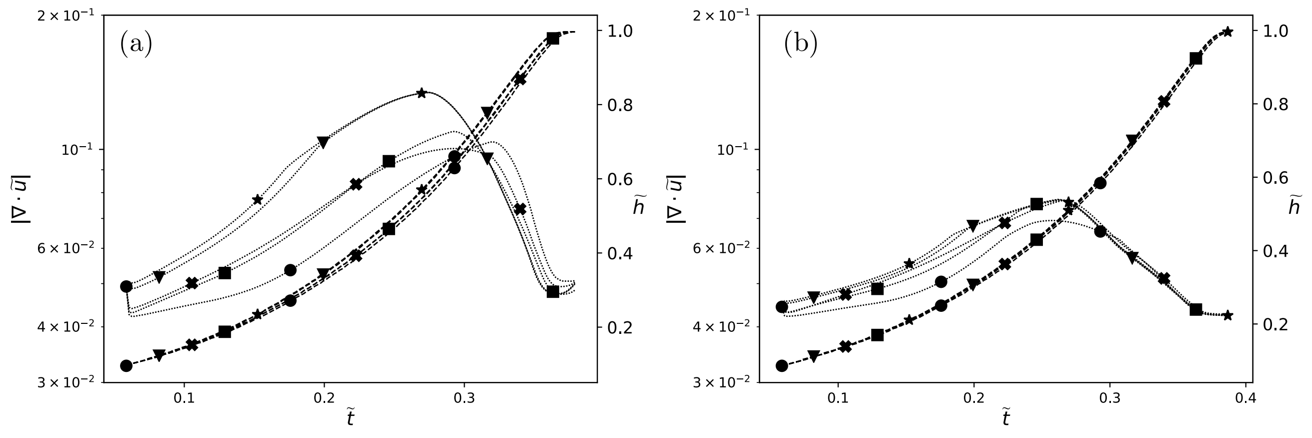

Figure 29. *Plume head position $\tilde{h}_y$ (- - -) and mean divergence of the velocity field (······) for the case with $\mathcal{E}_\infty = 0.37$ (a) and $\mathcal{E}_\infty = 0.18$ (b) without an obstacle at $R_i = 14.8$ obtained by several networks: ▼ MonoScale, ■ MultiScale, × SmallScale and • Unet as well as the ★ Jacobi solver.*

C. Evolution of the error distribution for the plume-cylinder case

Figure. 30 shows the KDE for the plume impinging a cylinder, for the cases where the threshold $\mathcal{E}$ is equal to both $\mathcal{E}_\infty$ and $\mathcal{E}_1$ for four timesteps ($\tilde{t}$ = 0.14, 0.27, 0.41 and 0.55). As previously mentioned, when $\mathcal{E} = \mathcal{E}_\infty$, the four networks, as well as the Jacobi solver, show a different behavior. This can be particularly appreciated at timestep $\tilde{t}$ = 0.27, although all the networks follow a unimodal distribution, the MonoScale network and the Jacobi solver, previously identified as the less accurate solvers, have a divergence peak around $|\nabla \cdot \mathbf{u}| = 0.2$, whereas the MultiScale and SmallScale networks behave similarly with a peak around $|\nabla \cdot \mathbf{u}| = 0.1$, and the Unet network outperforms the rest with a peak around $|\nabla \cdot \mathbf{u}| = 0.05$. When $\mathcal{E} = \mathcal{E}_1$, the MonoScale, MultiScale, SmallScale and Jacobi solver follow an almost identical distribution, showing a divergence peak at around $|\nabla \cdot \mathbf{u}| = 0.1$. The Unet behavior should be highlighted, since its distribution has a wider peak, centered on around $|\nabla \cdot \mathbf{u}| = 0.05$. This is related to the threshold level, which follows the lowest value of the mean divergence of all the studied networks. At timestep $\tilde{t}$ = 0.27, this value corresponds to the Unet network, so to achieve this threshold level, the other networks rely on Jacobi iterations which homogenize the divergence distribution, whilst the Unet's KDE corresponds to the network's output. Despite this difference, the plume flow is not affected, as macroscopically the flow structures are identical for all the cases when $\mathcal{E} = \mathcal{E}_1$.

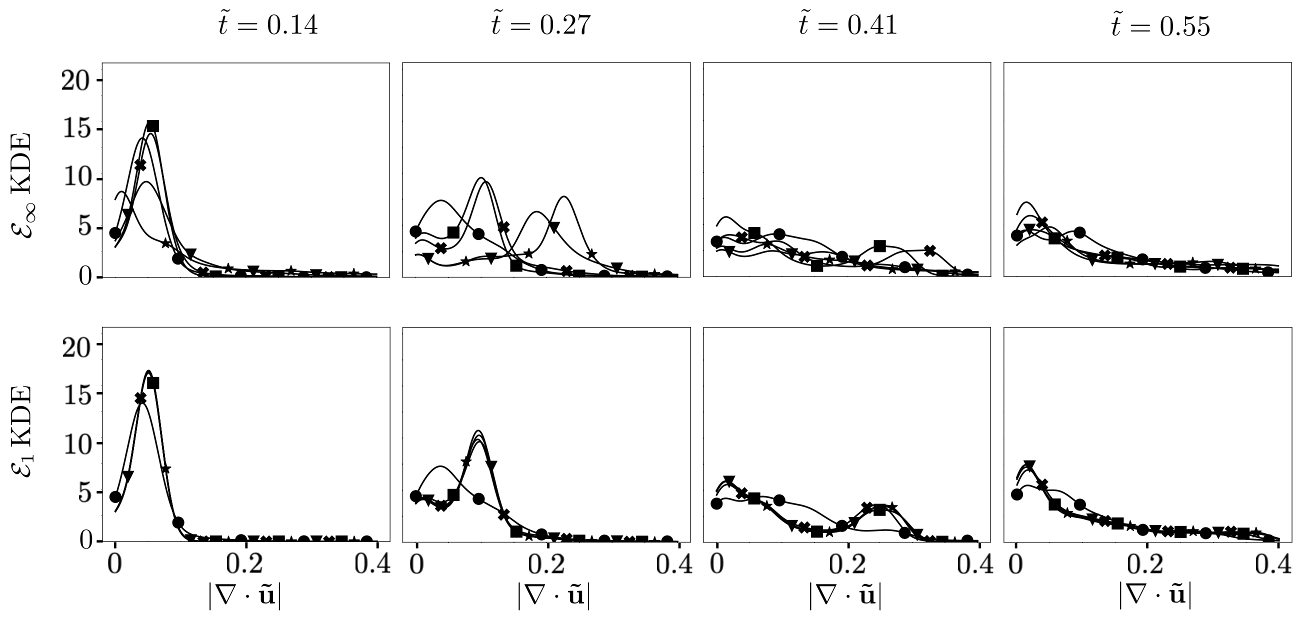

Figure 30. *KDE at 4 times ($\tilde{t}$ = 0.14, 0.27, 0.41 and 0.55) of the cases where $\mathcal{E} = \mathcal{E}_\infty$ (top) and $\mathcal{E} = \mathcal{E}_1$ (bottom) of the cylinder test case, at a $R_i = 14.8$ obtained by several networks: ▼ MonoScale, ■ MultiScale, × SmallScale, and ● Unet, as well as the ★ Jacobi solver.*

D. Further details of the training procedure

The training dataset is created following the same strategy as Tompson et al. (2017). It is generated with the open-source, GPU-adapted code: Mantaflow (Thuerey and Pfaff, 2016). The code works with uniform structured grids and it was mainly developed for computer-vision simulations. Both the training and validation datasets consist of 320 different fluid flow simulations of 128×128 cell domains. Every single simulation is propagated in time through 256 time steps, from which 64 iso-spaced time steps are saved for the simulation.

Each close-bounded domain is initialized with a wavelet turbulent noise technique, introduced by Kim et al. (2008), and contains random geometries issued from the NTU 3D Model database, which have been cropped for 2D cases. To generate more diverse fields, divergence sources are added and propagated through the domain, and random-gravity-like forces of different magnitudes and directions are applied as well. Fig. 31 shows an example of such a velocity field extracted from the training dataset.

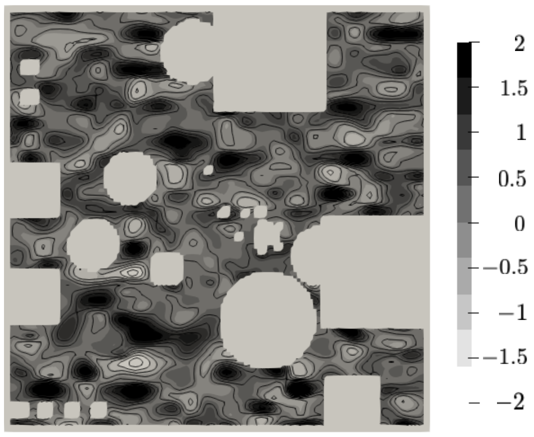

Figure 31. *Example of U_x velocity field (ms^{-1}) extracted from the training dataset.*

For the training, the *Adam* optimizer, developed by Kingma and Ba (2014) is used, with a learning rate initialized to 5×10^{-5} and a learning rate plateau scheduler. The batch size is kept to 64, (which could be slightly increased before saturating the GPU memory). The network weights are initialized using the *kaiming uniform* initializer. Converge of the network is usually reached in around 300 epochs.