Response to Referee 3

The authors truly appreciate your insightful comments that lead to a significant improvement of our manuscript.

1. I find the main focus of the paper somewhat unclear: is the paper primarily intended to introduce the complete FluidUrban modelling framework, as suggested by the title, or is its main contribution the incorporation of DAMO within this framework? In the latter case, has the underlying model already been described elsewhere?

The primary aim of this study is to introduce FluidUrban v1.0, a framework developed based on the open-source Fluidity solver for urban ventilation and pollutant dispersion modelling. Within this framework, the introduction of DAMO is the key numerical improvement and the central methodological contribution of this paper. In addition to the adaptive mesh strategy, FluidUrban v1.0 also includes a dedicated 3D urban geometry and mesh generator, urban-specific boundary conditions, and systematic validation against WOTAN wind-tunnel experiments and fixed-mesh benchmarks. The underlying LES and anisotropic adaptive mesh methodology are based on previously published Fluidity studies (e.g., Pain et al., 2005; Zheng et al., 2015, 2021; Wu et al., 2023), while the original contributions of this work lie in the urban-specific integration, implementation, and the quantitative evaluation presented.

Reference:

[1] Pain, C. C., Piggott, M. D., Goddard, A. J. H., Fang, F., Gorman, G. J., Marshall, D. P., Eaton, M. D., Power, P. W., and de Oliveira, C. R. E.: Three-dimensional unstructured mesh ocean modeling, Ocean Model., 10, 5–33, doi:10.1016/j.ocemod.2004.07.005, 2005.

[2] Zheng, J., Zhu, J., Wang, Z., Fang, F., Pain, C. C., and Xiang, J.: Towards a new multiscale air quality transport model using the fully unstructured anisotropic adaptive mesh technology of Fluidity (version 4.1.9), Geosci. Model Dev., 8, 3421–3440, doi:10.5194/gmd-8-3421-2015, 2015.

[3] Zheng, J., Wu, X., Fang, F., Li, J., Wang, Z., Xiao, H., Zhu, J., Pain, C., Linden, P., and Xiang, B.: Numerical study of COVID-19 spatial–temporal spreading in London, Phys. Fluids, 33, doi:10.1063/5.0048472, 2021.

[4] Wu, X., Abubakar-Waziri, H., Fang, F., Dilliway, C., Wu, P., Li, J., Yao, R., Bhavsar, P., Kumar, P., Pain, C. C., and Chung, K. F.: Modeling for understanding of coronavirus disease-2019 (COVID-19) spread and design of an isolation room in a hospital, Phys. Fluids, 35, 025111, doi:10.1063/5.0135247, 2023.

2. Could the authors please specify the spatial and temporal discretisation schemes that are used, together with the main numerical parameters employed in the simulations? In addition, some evidence that the underlying solver behaves satisfactorily on fixed meshes before introducing adaptivity would strengthen the paper.

Thank you for this important suggestion.

In the revised manuscript, we have clarified that the velocity and pressure fields are discretized using a continuous Galerkin finite-element formulation, while the scalar transport equation is solved using a control-volume scheme. Time integration is performed with a theta method ($\theta = 0.5$). For turbulence closure, the subgrid scale (SGS) stress is parameterized using a second-order Smagorinsky model ($Cs = 0.1$), and a tensor-based filter length is adopted for the anisotropic unstructured mesh. For the validation simulations, the time step is set to 1 s for both FIXM and DAMO simulations. DAMO is applied every 20 time steps, the initial mesh resolution is 5 m near building surfaces and 20 m elsewhere. During the DAMO processes, the mesh may be further refined to 2 m in regions with strong gradients, with the maximum number of mesh nodes limited to 2.0×10^5 for computing efficacy. We have added these details to Section 2.2 of the revised manuscript, specifically in lines 180–185.

Regarding the performance of the underlying solver on fixed meshes, the Fluidity framework's baseline accuracy utilizing fixed or non-adaptive unstructured meshes has been extensively validated in previous atmospheric and fluid dynamics studies (e.g., Savre et al., 2016; Li et al., 2021). Furthermore, within this manuscript, the simulations utilizing the non-uniform FIXM presented in Section 3 serve precisely as this validation baseline. By evaluating the FIXM results directly against the WOTAN wind-tunnel measurements, we demonstrate that the underlying solver behaves satisfactorily and captures the primary macroscopic flow features prior to the activation of the dynamic mesh optimization.

References:

[1] Li, J., Fang, F., Steppeler, J., Zhu, J., Cheng, Y., and Wu, X. Demonstration of a three-dimensional dynamically adaptive atmospheric dynamic framework for the simulation of mountain waves. Meteorology and Atmospheric Physics, 133(6), 1627-1645, 2021.

[2] Savre, J., Percival, J., Herzog, M., and Pain. Two-dimensional evaluation of ATHAM-fluidity, a nonhydrostatic atmospheric model using mixed continuous/discontinuous finite elements and anisotropic grid optimization. Monthly Weather Review, 144 (11), 4349-4372, 2016.

Major comments about Meshing and remeshing：

1. Section 2.1.1 details how the initial mesh is generated. Interesting techniques are mentioned, but with little detail :

(i) "a custom utility designed to extract building polygons directly from high-resolution satellite imagery utilizing a supervised Support Vector Machine (SVM) classification algorithm" seems to be quite a piece of software, could the authors please tell us more about it, or provide a reference?

We have expanded the description of the building-extraction workflow in Section 2.1.1 to clarify that it combines existing software with custom-developed scripts. The high-resolution satellite imagery was first radiometrically calibrated and atmospherically corrected. Training samples representing buildings, roads, vegetation, shadows, and other land-cover types were then manually selected in ENVI 5.3, after which a support vector machine (SVM) classifier was applied to perform pixel-wise land-cover classification. The classification results were evaluated using a confusion matrix, overall accuracy, and the Kappa coefficient, and the resulting building regions were converted into vector polygons. Finally, a custom Python program was used to read the building outlines and height information and generate the three-dimensional geometry files required by GMSH. The reference Turker & Koc-San (2015) has been added to support this workflow.

Reference:

[1] Turker M, Koc-San D. Building extraction from high-resolution optical spaceborne images using the integration of support vector machine (SVM) classification, Hough transformation and perceptual grouping[J]. International Journal of Applied Earth Observation and Geoinformation, 2015, 34: 58-69.

(ii) I presume that the geometry recovered from the satellite images is simplified for scientific computing purposes. How is this simplification done? How much detail is preserved, how was the choice made, and how does the tool ensure that?

The simplification mainly focuses on removing classification noise and controlling the subsequent mesh complexity. The SVM supervised classification results were first analyzed in ArcMap 10.8, where anomalous polygons were removed or reclassified. The vector results were then compared with the original satellite imagery to delete redundant information and correct misclassified boundaries. To explicitly simplify the geometry for scientific computing, a geometric smoothing rule was applied: any vertices (points) within the building polygons with a distance of less than 5 m between them were deleted. This 5-m threshold was strategically chosen to eliminate jagged edges and aerodynamically negligible micro-structures, while strictly preserving the main building outlines necessary for resolving the primary urban flow topology. The initial mesh resolution is also 5 m near building surfaces. This ensures that the generated 3D geometry retains sufficient macroscopic detail to simulate building-induced vortices without wasting computational resources on unresolvable sub-grid features.

(iii) The initial mesh is apparently not uniform. What determines the local mesh size in the initial mesh? Is it entirely user-defined?

Yes, the initial unstructured mesh is generated by the GMSH software, and the parameters determining the local mesh size are entirely user-defined. Within the GMSH geometry description file (*.geo), specific mesh sizes can be prescribed at individual vertices, such as the corners of the computational domain and the building polygons. GMSH utilizes these point-wise resolution constraints to generate the corresponding mesh, automatically ensuring a smooth and uniform transition of element sizes across different spatial regions.

In the present wind-tunnel simulation, the layout contains twenty regular rectangular buildings, all 15 m in height, with the central building measuring 60 m × 30 m × 15 m. For this setup, the initial user-defined mesh resolution was explicitly set to 5 m at the bottom, 20 m at the top of the computational domain, and 2 m at the building surfaces for the FIXM simulation (5 m for the DAMO simulation). As the simulation proceeds, the DAMO framework takes over: high-gradient regions are dynamically refined to 2 m, while other regions remain at or are coarsened to 20 m. Therefore, while the initial mesh is entirely user-defined, the detailed local mesh distribution during the run is dynamically controlled by the evolving flow-field gradients.

2. Section 2.4.2 details the remeshing procedure, and should be the heart of the article. However, I found it difficult to understand the proposed procedure from the current description, and had to infer parts of it from previous publications on adaptivity in Fluidity.

(1) Does this paragraph mean that there is a priori (before the metric-based refinement) refinement based on first and second derivatives ? If so, how does it work ? Otherwise, when are gradients considered in the following procedure ?

FluidUrban v1.0 employs a fully a posteriori, metric-driven remeshing strategy. There is no separate a priori refinement step based on first or second derivatives before metric adaptation.

Instead, the spatial derivatives are considered dynamically during the simulation. At each adaptation interval, the solver first advances the flow solution on the current mesh and then computes the Hessian matrix of the target velocity components u, v and w on the current mesh.

For each target field $f \in \{u, v, w\}$, the Hessian matrix is computed as

$$\boldsymbol{H}(f) = \begin{bmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x\partial y & \partial^2 f/\partial x\partial z \\ \partial^2 f/\partial y\partial x & \partial^2 f/\partial y^2 & \partial^2 f/\partial y\partial z \\ \partial^2 f/\partial z\partial x & \partial^2 f/\partial z\partial y & \partial^2 f/\partial z^2 \end{bmatrix}.$$

The Hessian, together with the prescribed interpolation error bound ϵ, is then used to generate the corresponding anisotropic metric tensor $\boldsymbol{M}$. In the principal coordinate system of the Hessian, $\boldsymbol{H} = \boldsymbol{V}_H \boldsymbol{S}_H \boldsymbol{V}_H^T$, where $\boldsymbol{V}_H$ contains the eigenvectors and $\boldsymbol{S}_H = diag(\lambda_1, \lambda_2, \lambda_3)$ contains the eigenvalues. In this framework, directions associated with larger $|\lambda_i|$ correspond to stronger curvature and therefore require finer resolution. The three field-specific metric tensors are combined into a unified metric tensor, which is subsequently smoothed and globally rescaled to satisfy the target mesh complexity before being passed to the mesh generator.

Regarding the specific consideration of gradients (first derivatives) versus second derivatives: in our spatial discretization using linear finite elements, the leading-order interpolation error is theoretically bounded by the second derivatives. Therefore, first derivatives (gradients) are not used as an independent refinement trigger. Instead, the adaptation procedure relies strictly on the second derivatives (the Hessian matrix), which are evaluated a posteriori at each adaptation step to drive the mesh optimization.

(2) (i) could the authors please define a "metric tensor" ? How is it used ?

We have clarified the definition and role of the metric tensor in the revised manuscript. In this work, the metric tensor $\boldsymbol{M}$ is a symmetric positive-definite matrix defined at each point in space, used to prescribe the optimized local mesh size and orientation.

The fundamental goal of this metric-based mesh optimization is to ensure that all mesh edges approach a unit length when measured in the defined metric space. Consequently, the eigenvalues and eigenvectors of $\boldsymbol{M}$ dictate the physical dimensions of the mesh elements. If an eigenvalue of $\boldsymbol{M}$ is relatively large, the physical mesh edge must become correspondingly smaller (finer) along the direction of its associated eigenvector to maintain this unit length. Conversely, a smaller eigenvalue dictates a coarser mesh in that direction. By assigning different eigenvalues along different principal directions, a fully anisotropic mesh can be generated.

Once the metric tensor is constructed, it is passed to the meshing optimization algorithms, which perform local operations (such as edge splitting, collapsing, swapping, and node smoothing) until the mesh quality criteria specified by the metric field are satisfied.

(ii) could the authors please explicit how this metric tensor is obtained from the Hessian matrix ?

We sincerely thank the reviewer for outlining this precise derivation. We fully agree that explicitly detailing these steps significantly improves the clarity and rigor of the methodology section. We have incorporated this exact derivation into the revised manuscript, specifically in lines 273–291.

To explicitly clarify how the metric tensor is derived from the Hessian matrix for anisotropic mesh adaptation, the following steps have been detailed in the revised manuscript:

First, the Hessian matrix $\boldsymbol{H}$ of the target solution variable f is computed as:

$$\boldsymbol{H}(f) = \begin{bmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial y & \partial^2 f/\partial x \partial z \\ \partial^2 f/\partial y \partial x & \partial^2 f/\partial y^2 & \partial^2 f/\partial y \partial z \\ \partial^2 f/\partial z \partial x & \partial^2 f/\partial z \partial y & \partial^2 f/\partial z^2 \end{bmatrix}.$$

Second, its eigenvalue decomposition is performed:

$$\boldsymbol{H} = \boldsymbol{V}_H \boldsymbol{S}_H \boldsymbol{V}_H^T$$

where $\boldsymbol{V}_H$ the matrix of eigenvectors and $\boldsymbol{S}_H = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ is the diagonal matrix of eigenvalues.

Third, to ensure the resulting metric is symmetric and positive-definite, the processed Hessian norm is calculated by taking the absolute values of the eigenvalues:

$$\|\boldsymbol{H}\| = \boldsymbol{V}_{\boldsymbol{H}}\, diag\left(\left|\lambda_i^{\boldsymbol{H}}\right|\right) \boldsymbol{V}_{\boldsymbol{H}}^T$$

Finally, the Riemannian metric tensor $\boldsymbol{M}$ is constructed using the user-specified maximum numerical simulation error bound ϵ and a scaling constant γ:

$$\boldsymbol{M} = \gamma\epsilon\|\boldsymbol{H}\|$$

(iii) Could the author define the ∩ operator ? And explain the procedure for combining such operators ?

In the revised manuscript, we have clarified that the symbol ∩ denotes the metric-intersection operator, rather than a set-theoretic intersection. This operator is used when multiple solution variables or refinement indicators simultaneously require mesh adaptation. In such cases, a separate metric tensor is first constructed for each variable, and these individual tensors are then combined pointwise into a single composite metric tensor.

The intersection procedure yields the most restrictive metric among the candidates, so that the final adapted mesh satisfies all prescribed resolution requirements simultaneously. In practice, the tensors are merged by evaluating their refinement demands in a common eigenspace representation and retaining the direction-wise stricter condition (i.e., the smaller target edge length). The resulting composite metric tensor is symmetric positive definite and is then used as the target metric for mesh generation and adaptation.

(iv) is there any evidence that metric M actually results in an error of ε ? If so, could

the author provide such evidence, or reference ?

We have clarified this point in the revised manuscript. The metric tensor $\boldsymbol{M}$ is not intended to guarantee that the numerical error is exactly equal to a prescribed value ε at every location. Rather, it is constructed from finite-element interpolation error estimates so that the adapted mesh approximately equidistributes the estimated error and keeps it below the target tolerance in a local sense. In other words, $\boldsymbol{M}$ provides an error-based mesh adaptation criterion rather than an exact error certificate.

The theoretical basis of this approach is well established in adaptive finite-element methods. In particular, the metric tensor is commonly derived from interpolation error indicators, Hessian-based estimates, or adjoint-based goal-oriented error measures. We have added the relevant references to support this discussion, including Pain et al. (2001), and Power et al. (2006). These studies provide both the theoretical background and practical evidence that metric-based mesh adaptation can effectively control the discretization error.

Reference:

[1] Pain C C, Umpleby A P, de Oliveira C R E, et al. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations[J]. Computer Methods in Applied Mechanics and Engineering, 2001, 190(29-30): 3771-3796.

[2] Power P W, Piggott M D, Fang F, et al. Adjoint goal-based error norms for adaptive mesh ocean modelling[J]. Ocean modelling, 2006, 15(1-2): 3-38.

(3) (i) could the authors please specify which "composite metric tensor" they are referring to ?

We have clarified this point in the revised manuscript. The term "composite metric tensor" refers to the unified metric obtained by combining the individual metric tensors associated with multiple physical variables into a single tensor field. This composite metric is constructed so that the adapted mesh can simultaneously satisfy the resolution requirements of all variables under consideration.

In our case, the metric tensors derived from the velocity components u, v, w, and the tracer field are merged using the metric-intersection operator, i.e.,

$$M = M_u \cap M_v \cap M_w \cap M_c \ .$$

The resulting tensor represents the most restrictive adaptation demand among the individual fields and is then used as the target metric for mesh generation.

(ii) If a global scaling is applied to control the number of nodes, am I wrong to think that ε is not really reached ? if this is correct, what role should be attributed to this error bound?

Yes, we agree with the reviewer's assessment. When a global scaling procedure is applied to control the total number of nodes, ε can no longer be interpreted as a strict upper bound on the final mesh error. In this case, ε serves primarily as a nominal error tolerance used to construct the initial metric tensor.

The subsequent global scaling modifies the metric to satisfy the prescribed mesh-size or node-number constraint. Therefore, the final adapted mesh may not exactly attain the target error level ε in all regions. For this reason, ε should be understood as a baseline input parameter that dictates the relative spatial distribution of the mesh resolution, rather than as a guaranteed final error bound. If a strict error certificate is required for a specific application, the local error must be recomputed a posteriori on the adapted mesh to verify whether the target has been achieved.

(iii) How is the adapted mesh generated ? Is the meshing tool the same as in (Pain 2005) ?

We thank the reviewer for this question. The adaptive mesh generation procedure involves four main steps:

(1) Constructing a metric tensor from the relevant physical fields;

(2) Evaluating the quality of the current mesh with respect to this target metric;

(3) Improving the mesh to satisfy the metric through local topological operations,

including edge splitting, collapsing, swapping, and node smoothing;

(4) Transferring the solution fields from the old mesh to the new mesh using a conservative Galerkin projection.

Regarding the meshing tool, yes, it belongs to the exact same technical framework as that described in Pain et al. (2005). FluidUrban v1.0 is developed based on the open-source Fluidity solver, which utilizes the same core unstructured mesh adaptation libraries and methodologies pioneered in Pain et al. (2005) and further refined for conservative interpolation in Farrell et al. (2009).

Reference:

[1] Pain, C. C., Piggott, M. D., Goddard, A. J. H., Fang, F., Gorman, G. J., Marshall, D. P., Eaton, M. D., Power, P. W., and de Oliveira, C. R. E.: Three-dimensional unstructured mesh ocean modeling, Ocean Model., 10, 5–33, doi:10.1016/j.ocemod.2004.07.005, 2005.

[2] Farrell, P. E., Piggott, M. D., Pain, C. C., Gorman, G. J., and Wilson, C. R.: Conservative interpolation between unstructured meshes via super mesh construction, Comput. Methods Appl. Mech. Engrg., 198, 2632 – 2642, doi:10.1016/j.cma.2009.03.004, 2009.

3. How was the spatial accuracy of the fixed mesh (FIXM) chosen ?

The fixed mesh adopts an unstructured tetrahedral grid with a nonuniform spatial resolution distribution. To characterize the overall spatial resolution, we use an equivalent resolution metric, namely by converting the area of each triangular element into the side length of an equilateral triangle with the same area, and then computing the area-weighted average over all elements. Therefore, the equivalent resolution of the fixed mesh near building surfaces is 2 m, while that in other regions is 20 m.

In addition, the underlying dynamical framework used in this study is the Fluidity model, whose spatial accuracy has been validated in previous work; see the following two references for details. The numerical discretization method adopted in this study is a hybrid scheme combining continuous finite elements and control-volume methods:

pressure and velocity are discretized using linear continuous finite elements on the tetrahedral mesh, while other prognostic variables are discretized using a control-volume method based on the dual mesh formed by tetrahedral cell centroids.

Reference:

[1] Li J, Fang F, Steppeler J, et al. Demonstration of a three-dimensional dynamically adaptive atmospheric dynamic framework for the simulation of mountain waves[J]. Meteorology and Atmospheric Physics, 2021, 133: 1627-1645.

[2] Savre J, Percival J, Herzog M, et al. Two-dimensional evaluation of ATHAM-fluidity, a nonhydrostatic atmospheric model using mixed continuous/discontinuous finite elements and anisotropic grid optimization[J]. Monthly Weather Review, 2016, 144(11): 4349-4372.

Some more questions:

1. Section 3: I don't think the mesh sizes (number of elements, min size, max size...) are specified for both adaptive and fixed meshes. Could the author give those figures ?

To address the insufficient description of the mesh information in Section 3, we have added the mesh size distribution plot (see **Figure. R1**), which illustrates the spatial resolution characteristics of both the adaptive and fixed meshes under different conditions, together with the corresponding number of mesh cells and the minimum and maximum mesh sizes.

It should be noted that the mesh size in **Figure. R1** is defined in terms of edge length. To generate this data, a custom routine reads the three-dimensional coordinates of all mesh points in the vtu output files and iterates over each cell to extract all of its edges. The length of each edge is calculated using the Euclidean distance formula:

$$L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}.$$

For edges shared by adjacent cells, we ensure that each edge is counted only once by sorting the endpoints and removing duplicates using a set-based operation. The resulting distribution of all unique edge lengths is then used for the mesh size statistics and plotting. Therefore, **Figure R1** now provides a clear, quantitative representation of the relevant mesh parameters and their specific distributions.

We have added these details to Section 3 of the revised manuscript, specifically in lines 385–400.

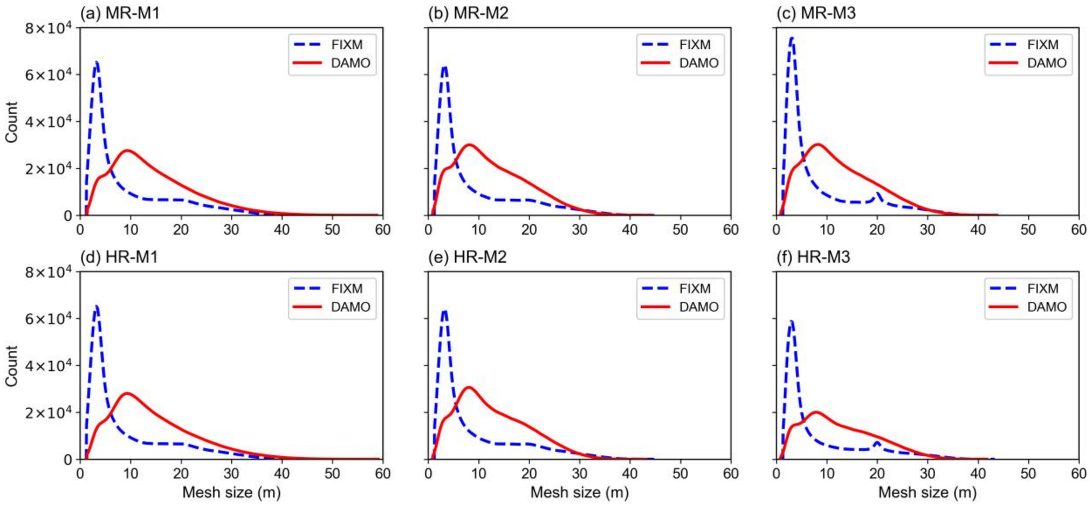

Figure. R1 Mesh size distribution statistics under the medium roughness (upper row) and high roughness (lower row) conditions. The blue dashed line represents the FIXM simulation results, and the red solid line represents the DAMO simulation results.

2. Figure 6: I would have expected more turbulent variability behind the building. Could the adaptation process selected introduce some dissipation that would filter part of the turbulence structures ?

This comment is very important to us.

The mesh adaptation process, particularly variable interpolation and remeshing, does introduce a certain level of numerical dissipation. This can smooth small-scale, instantaneous turbulent structures and, to some extent, reduce the apparent fluctuation intensity in the wake region behind the building.

To visually demonstrate the physical continuity of the wake circulation during this process, we have added a sequence of instantaneous streamline plots (**Figure R2**). Dynamic mesh adaptation is executed exactly at t=600 s (c) and t=605 s (h). Despite the complete updating of the mesh topology and subsequent variable interpolation at these steps, the primary recirculation vortices behind the buildings and the flow separation structures transition smoothly. The conservative Galerkin projection ensures that the remeshing does not unphysically fragment or destructively filter out these macroscopic wake circulations.

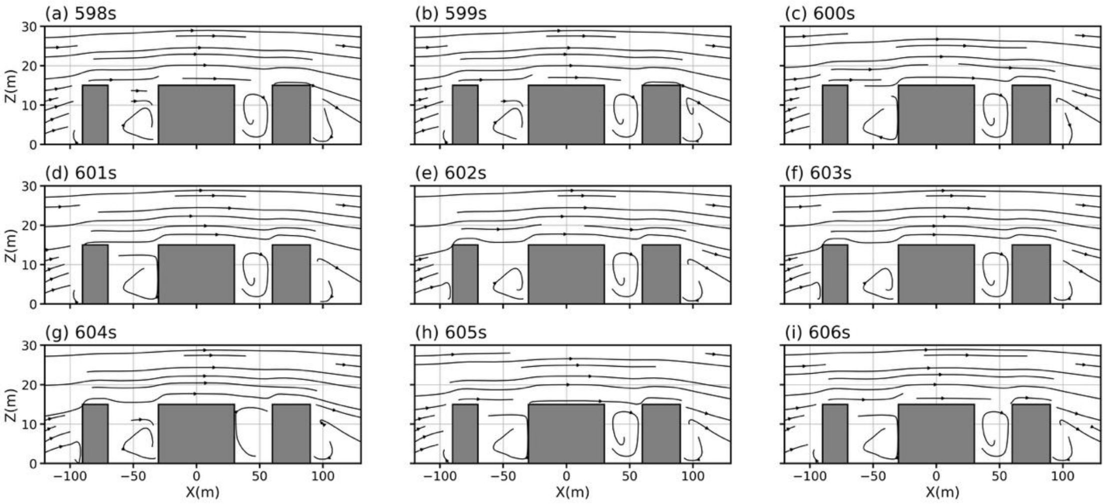

Figure R2: Instantaneous flow streamlines on a vertical cross-section across the buildings from $t = 598$ s to $t = 606$s (a–i). Dynamic mesh adaptation is executed at $t = 600$ s (c) and $t = 605$ s (h), illustrating the evolution of the flow field and the continuity of the vortex structures during the remeshing and interpolation processes.

To quantitatively examine the potential dissipation effect, we have added an evaluation of the turbulence intensities I_u and I_v in the revised manuscript to further verify the model's ability to characterize velocity fluctuations.

Specifically, for the simulation results, we first extracted the velocity components u and v at each measurement point and then calculated I_u and I_v according to the corresponding turbulence intensity formulas. For the wind tunnel experimental data, we recalculated I_u and I_v using the same time interval as the simulation results to ensure consistency and comparability in the temporal scale. Meanwhile, we added two types of figures: the vertical profile I_u and I_v between the simulation and wind tunnel experiment at observation points (**Figure. R3**), and the error distribution plots (**Figure. R4**). Here, the error is defined as “simulation result minus wind tunnel result,” so an error below 0 indicates that the simulation underestimates the turbulence intensity.

The vertical profiles of both I_u and I_v turbulence intensities reveal that the FIXM configuration consistently and severely underestimates turbulence across almost all heights and inflow scenarios. This is visually evident as the FIXM results (red dashed lines in **Figure R3**) fall far to the left of the wind-tunnel measurements (blue open circles in **Figure R3**). This pronounced underestimation occurs because the static mesh relies on a priori refinement, which introduces excessive numerical dissipation in evolving wake regions and artificially smooths out critical velocity fluctuations.

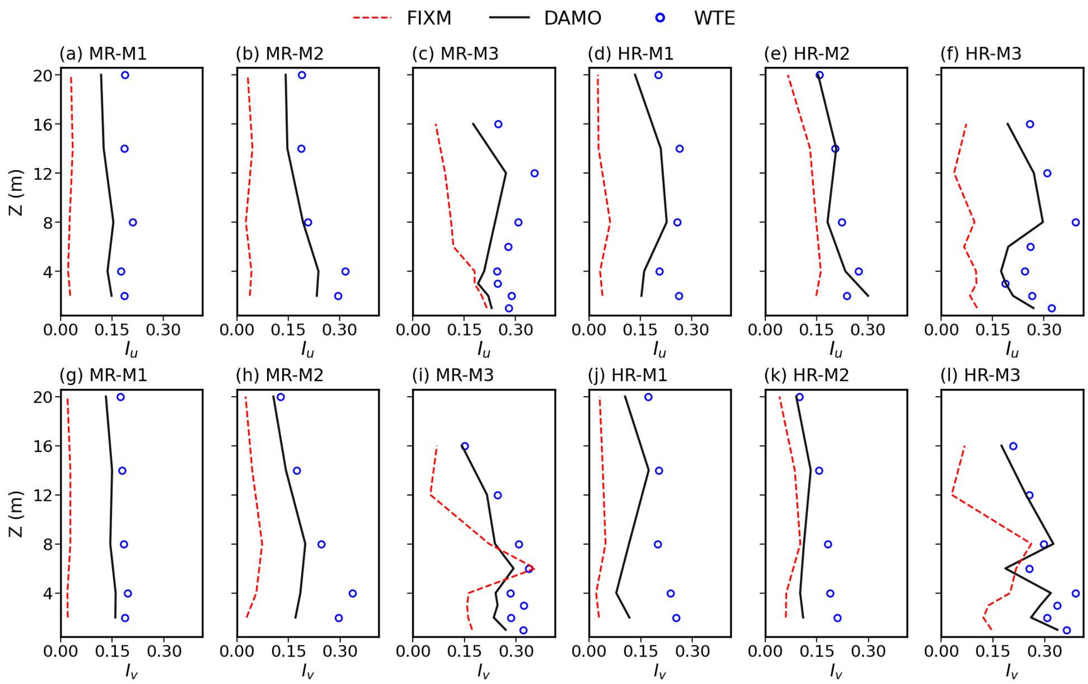

Figure. R3 Comparison of the vertical turbulence intensity profiles between the FluidUrban simulations and wind-tunnel experiments (WTE). The upper row (a–f) presents the streamwise turbulence intensity (I_u), and the lower row (g–l) presents the lateral turbulence intensity (I_v). Results are shown for the FIXM model (red dashed lines), the DAMO model (black solid lines), and the WTE observational data (blue open circles) across the medium (MR) and high (HR) roughness test cases.

Conversely, the DAMO framework (black solid lines in **Figure R3**) captures the magnitude and vertical trends of the turbulence intensity much more accurately. By dynamically tracking and refining the grid across high-gradient regions, such as separated shear layers and building wakes, DAMO explicitly resolves smaller eddies. This physical-driven allocation of computational resources effectively limits numerical diffusion, preserving the turbulent kinetic energy necessary for resolving complex multi-scale flow fields.

Figure R4 provide a robust quantitative comparison of the signed errors of I_u and I_v, where negative values indicate an underestimation relative to the observation data. While both numerical configurations exhibit overall negative biases, DAMO (red boxes) shows median errors that are consistently and significantly closer to the zero-error baseline compared to FIXM (blue boxes). Furthermore, the interquartile ranges (the height of the boxes) and the overall data spread (whiskers) are noticeably narrower and more concentrated for DAMO. In contrast, FIXM shows a more dispersed error distribution. This tighter distribution confirms that DAMO not only minimizes systematic negative bias but also yields much more stable, reliable, and spatially consistent predictions of turbulence fluctuation amplitudes across the entire complex urban morphology.

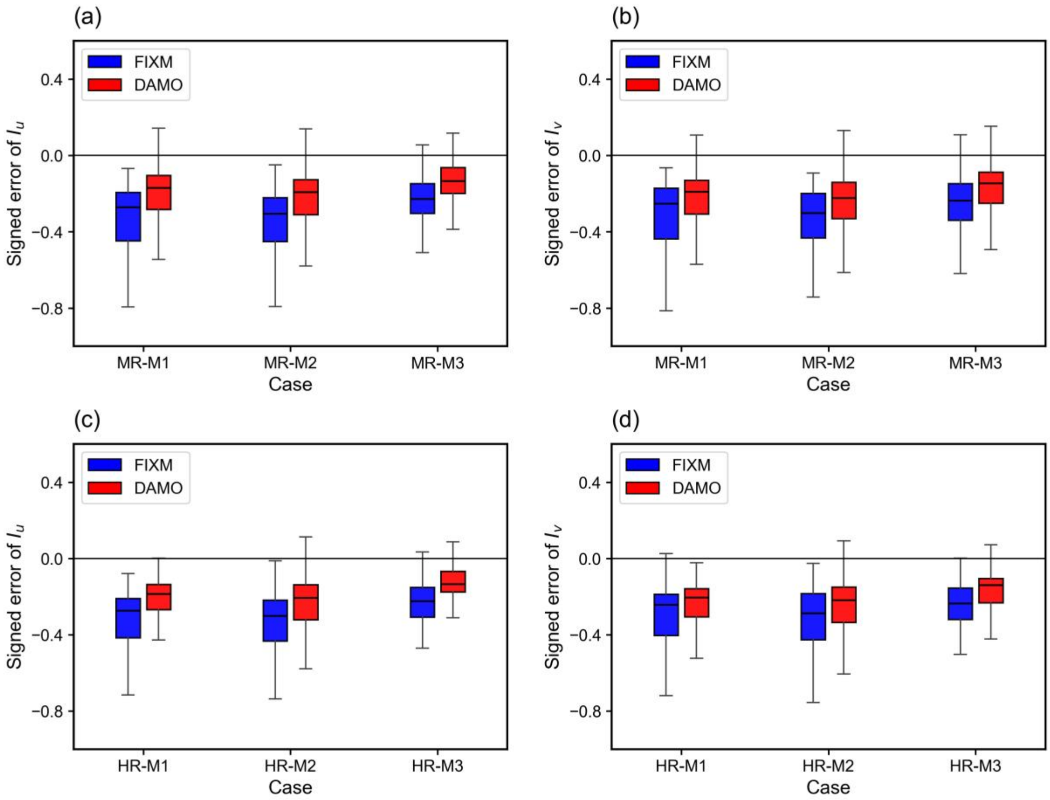

Figure. R4 Boxplots of the signed errors for the simulated streamwise (I_u) and lateral (I_v) turbulence intensities under medium roughness (MR, upper row) and high roughness (HR, lower row) conditions. For each boxplot, the box indicates the interquartile range, the horizontal line within the box represents the median, and the whiskers denote the overall data range. The solid horizontal black line marks the zero-error baseline.

3. Table 3 and others : have the authors looked at statistical dispersion indicators as well ?

Regarding whether statistical-dispersion indicators were considered for Table 3 and the other results, we have added an analysis of statistical dispersion indicators to further evaluate the model performance. Specifically, **Figure. R4** compares the simulated and wind-tunnel-measured distributions of the turbulence statistics I_u and I_v. The results show that both DAMO and FIXM generally underestimate these quantities, indicating that both methods are affected to some extent by numerical dissipation.

4. Figure 9 : is DAMO better than FIXM here ?

Yes, DAMO performs better than FIXM, especially in oblique-wind scenario (30º inlet boundary condition). As highlighted by the red boxes in Figure 9(c) and (f), the wind direction simulated by the FIXM model deviates substantially from the wind-tunnel measurements, even exhibiting an approximately opposite flow direction in certain localized areas. By contrast, DAMO reproduces the local wind-direction distribution much more accurately, yielding a flow topology that is much more consistent with the experimental observations.

To further quantify these visual deviations, we have included the mean absolute error (MAE) of the flow fields in the revised manuscript (see Figure 11). The statistical results confirm that DAMO achieves a smaller MAE than FIXM across all test cases, firmly demonstrating its higher overall accuracy in resolving complex urban wind directions.

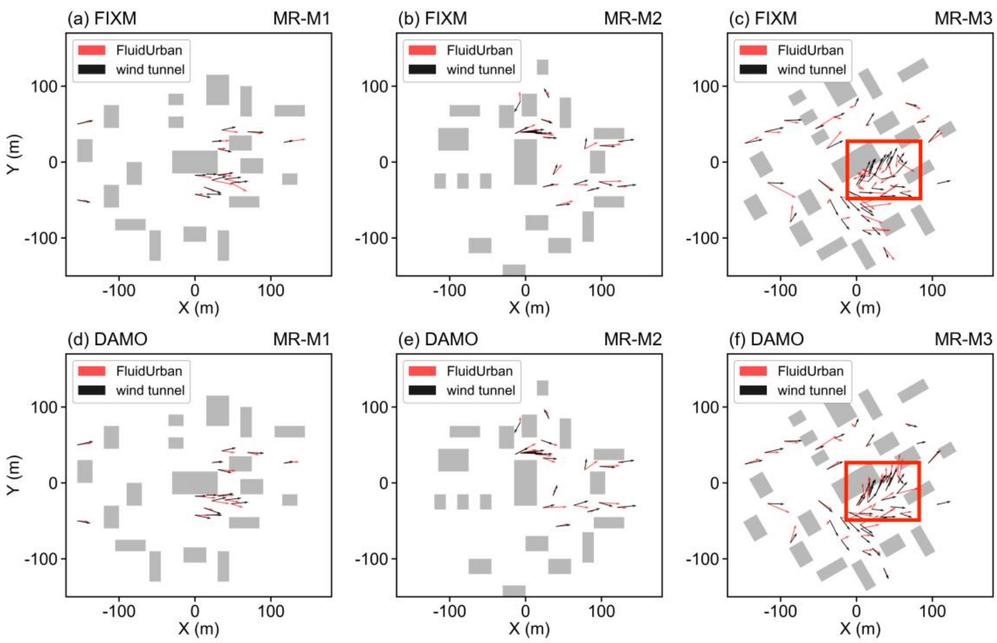

Figure 9: 2 m horizontal wind vectors under the medium roughness (MR) condition of the simulation (red arrows) with wind-tunnel experiment (black arrows). (a–c) present the FIXM simulation results, and (d–f) mean the DAMO simulation results, with inlet flow direction of 0° (left column), 90° (middle column) and 30° (right column).

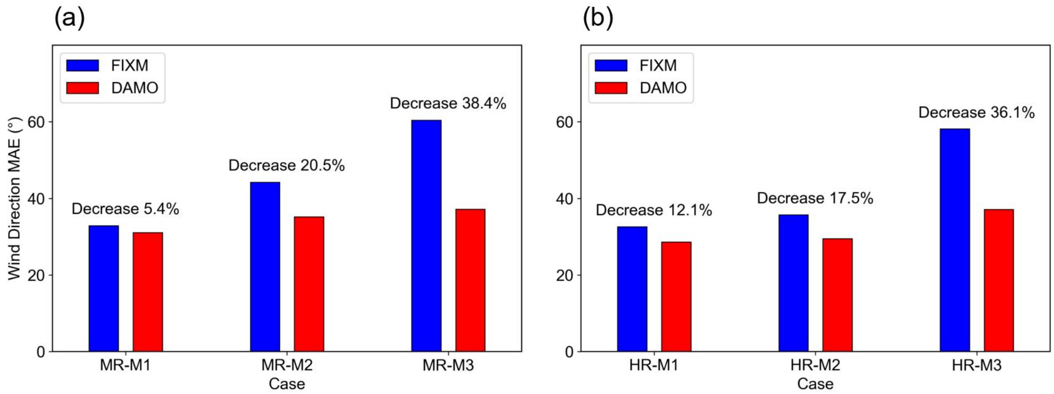

Figure 11: The spatially averaged Mean Absolute Error (MAE) of 2 m horizontal wind direction under (a) Medium roughness condition (MR) and (b) High roughness condition (HR). The labels "Decrease" and the associated percentages represent the reduction in MAE achieved by DAMO relative to FIXM.

5. Figure 12 : the flow for a fixed mesh (c) seems fairly different from measurements. Why does the model not perform better on this case ? Why is mesh adaptation guaranteed to perform better ?

The relatively large discrepancy between the fixed-mesh results and the measurements in Figure 12(c) is a direct consequence of the inaccurate wind field prediction shown earlier in Figure 9(c). Pollutant transport is fundamentally driven by the underlying flow field. In this oblique-wind scenario, dispersion is jointly controlled by the angled inflow and complex building-induced flow separation, which generate pronounced wake deflection and lateral entrainment. Because FIXM fails to accurately capture the correct flow topology, even simulating opposite wind directions in certain wake regions (as highlighted in Figure 9c), it inevitably fails to reproduce the correct plume deflection direction and overall dispersion pattern in Figure 12(c).

This failure stems from the limitations of the fixed mesh. Although the fixed mesh was locally refined near the buildings at the initial setup stage, its resolution distribution remains static. It cannot dynamically adjust to the spatial evolution of the transient separation zones and the moving plume front. Consequently, insufficient resolution occurs in these critical high-gradient regions, leading to stronger numerical diffusion, excessive smoothing of the plume structure, and fundamentally altered local flow vectors.

In contrast, DAMO dynamically refines the wake region, building corners, and the moving plume front based on the continuously evolving metrics of both the velocity and concentration fields. This dynamic realignment ensures that the underlying wind vectors (Figure 9f) remain highly accurate, which in turn enables a much more precise representation of pollutant entrainment and plume-front evolution (Figure 12f).

Finally, it should be emphasized that DAMO does not theoretically guarantee better performance in every conceivable scenario. Rather, it is designed to provide significantly improved accuracy when the flow field is strongly heterogeneous, exhibits sharp transient gradients, and contains moving physical structures. The oblique-wind dispersion case considered here perfectly exemplifies this category. By continuously resolving these critical local features, the adaptive mesh remains strictly consistent with the underlying flow physics, thereby demonstrating higher accuracy in this specific context.

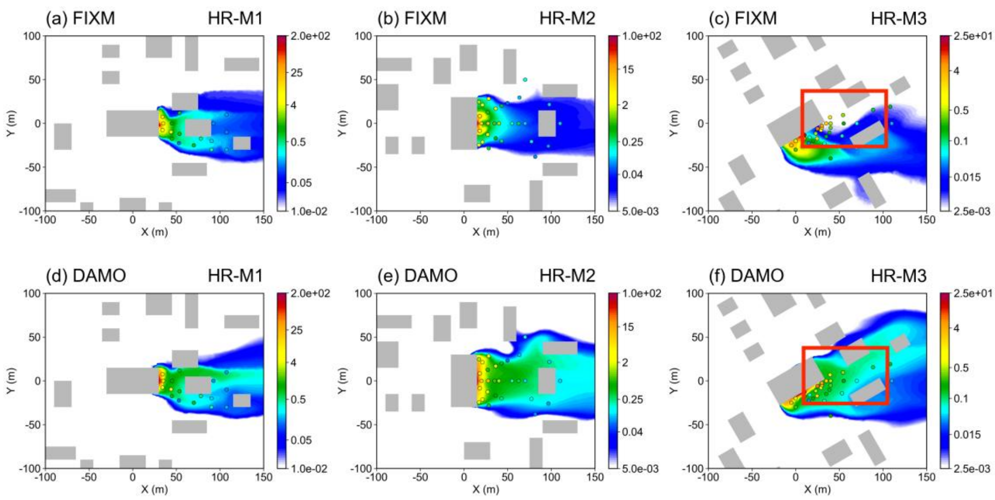

Figure 12: Pollutant gas concentration slice at 2 m above ground under medium surface roughness (shaded, units: kg m^{-3}). Gray boxes represent buildings, and the circles represent experimental measuring points. (a–c) the simulation results of FIXM, and (d–f) the simulation

results of DAMO.

6. Section 3 in general : improvements like MAE changing from 0.214 to 0.187 ar reported. As LES is inherently stochastic, how are the authors sure that these differences are not within expected sampling variability

We completely agree with the reviewer that LES is inherently stochastic, and therefore the interpretation of error metrics must not rely on instantaneous snapshots.

In this study, the MAE was not computed from instantaneous flow fields. Instead, it was evaluated using the time-averaged flow fields calculated over a long, statistically stationary time interval (specifically, a 1400-second data window from $t = 600s$ to $t = 2000\,\mathrm{s}$), and then spatially averaged across all observational points. This rigorous time-averaging effectively eliminates the influence of short-term random turbulent fluctuations, ensuring that the evaluated MAE represents the converged mean flow statistics rather than stochastic sampling noise.

You are absolutely right that LES has inherent temporal fluctuations, and therefore the interpretation of error metrics should avoid relying on a single instantaneous snapshot. In this study, the MAE was not computed from a single instantaneous field. Instead, it was evaluated over a statistically stable time interval, namely from $t = 600\,s$ to $t = 2000\,s$ corresponding to 1400 seconds of data, and then further averaged over all observation points in space. This treatment can effectively reduce the influence of short-term random fluctuations, making the results more representative of the mean flow statistics rather than instantaneous noise.

Furthermore, the reduction in MAE achieved by DAMO is not an isolated occurrence. It demonstrates a consistent, repeatable trend across different surface roughness conditions (both MR and HR) and various inflow angles, with particularly pronounced improvements in the highly complex oblique-inflow cases. Because this consistent reduction in error is observed across multiple independent scenarios after long-term time-averaging, we are confident that the differences reflect a systematic physical improvement derived from the adaptive mesh resolution, rather than expected sampling variability.

7. Section 3.4 performs a performance assessment on fixed and adaptive meshes with a similar mesh size. What would the comparison results be if, instead, the meshes were chosen to as to provide a similar accuracy ? I believe the "superior balance between accuracy and computation cost" (conclusion) would be more convincingly demonstrated by fixing alternatively the two measures.

We completely agree with the reviewer that conducting a comparison under a strictly constrained “similar accuracy” framework would provide a more rigorous and convincing assessment of the accuracy-efficiency balance.

In the context of transient LES for urban flows, however, designing a static mesh that guarantees a specific, global accuracy equivalence to a dynamic mesh is quite challenging in practice. Because the local error distribution is driven by continuously evolving flow structures (e.g., shedding vortices and moving shear layers), the required optimal resolution cannot be perfectly known a priori. Given that the core characteristic of the Fluidity framework is the dynamic tracking of these transient, multi-scale features, we followed the common practice of evaluating the adaptive method by constraining the initial computational budget (using a similar nominal mesh size) and observing the resulting improvements in physical accuracy.

Nevertheless, we fully acknowledge that a fixed-accuracy comparison is a highly meaningful metric. Constructing such a framework would require systematic parametric testing to adjust the adaptive mesh’s error threshold to exactly match the fixed mesh’s error level. We plan to incorporate these systematic mesh-convergence experiments in our future work to more comprehensively quantify the method’s performance limits, and we have noted this limitation in the revised manuscript, and will add related discussion in the end of the manuscript.

8. The paper argues that DAMO improves the accuracy-efficiency balance. Yet, the only example where performance is evaluated exhibits at 17% overhead for DAMO, which is only moderately convincing. Could the authors provide evidence that the *accuracy-efficiency trade-off becomes favourable for larger or more complex problems?*

We sincerely thank the reviewer for this important reminder. We agree that the 17.1% computational overhead observed in the current WOTAN benchmark represents a specific result for a moderately sized domain, and it is not sufficient to definitively prove a universally favorable accuracy-efficiency trade-off for all problem sizes.

The core characteristic of the Fluidity framework, upon which FluidUrban v1.0 is built, is the precise resolution of multi-scale flow fields and complex physical processes, rather than prioritizing extreme high-performance computing optimization. Within this context, the 17.1% runtime increase is viewed as a necessary computational investment to achieve the improved physical accuracy demonstrated in this study, specifically, the correction of anomalous wind directions (Figure 9) and the accurate capturing of pollutant plume deflection (Figure 12).

Regarding larger and more complex problems, our expectation of a favorable trade-off is based on theoretical mesh scaling behavior. In extensive urban microclimates, severe flow heterogeneity occupies a relatively small volume fraction of the domain. If a static mesh were used to guarantee the same high local resolution (e.g., 2 m) across a much larger domain, the total number of mesh elements would scale cubically. DAMO theoretically mitigates this by dynamically concentrating resolution strictly where the flow physics demand it. However, we humbly acknowledge that this remains a theoretical inference at this stage. We have revised the manuscript to moderate our conclusions, explicitly stating that future studies involving ultra-large-scale domains and longer integrations are required to fully verify this efficiency advantage in practice.

9. I am curious to know whether FluidUrban has been assessed on additional academic or realistic benchmark cases, and whether a sensitivity analysis of the adaptive parameters has been carried out. Could the author please comment and provide more validation cases ?

We sincerely appreciate the reviewer's constructive comment. We agree that incorporating additional benchmark cases and conducting a systematic sensitivity analysis of the adaptive parameters are essential steps to further establish the robustness and generalizability of the framework.

In the present manuscript, our primary validation relies on the WOTAN wind-tunnel dataset because it provides comprehensive, high-quality three-dimensional flow and pollutant-dispersion measurements under multiple inflow angles and surface roughness conditions, making it particularly suitable for validating 3D adaptive mesh modeling. To complement this academic benchmark, Section 4 presents an application of FluidUrban in a realistic, complex industrial urban setting, demonstrating the framework's practical feasibility in handling intricate real-world geometries and multi-scale dispersion.

However, we acknowledge that the current study does not include additional standardized academic benchmarks (such as the CEDVAL or AIJ cases), nor does it present a comprehensive parametric sensitivity analysis regarding adaptation frequency, error tolerance (ϵ), and target mesh complexity. Because the primary objective of this manuscript is to introduce the architecture of FluidUrban v1.0 and validate its baseline physical performance, we focused on demonstrating the core methodology rather than conducting exhaustive parameter optimization.

We have updated the manuscript to explicitly state these limitations. Moving forward, systematically evaluating parameter sensitivities and applying the model across a wider variety of urban morphology benchmarks will be a primary focus of our subsequent research.