Parameter Identification of a Macroscopic Crowd Model Governed by a Hyperbolic PDE Using Physics-Informed Neural Networks

Ousmane Fall
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab
Grenoble, France
ousmane.fall@grenoble-inp.fr

Didier Georges
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab
Grenoble, France
didier.georges@grenoble-inp.fr

Anne-Françoise Yao
Univ. Clermont Auvergne, CNRS, LMBP
Clermont-Ferrand, France
CMAP, Ecole Polytechnique
Paris, France
anne.yao@uca.fr

***Abstract*—We address parameter identification for a macroscopic crowd model governed by a second-order hyperbolic PDE system. The proposed approach relies on Physics-Informed Neural Networks (PINNs), where unknown physical parameters are optimized jointly with the network weights through a loss functional combining data mismatch, PDE residuals, initial conditions, and boundary conditions. The framework is applied to a Bellomo–Dogbé-type model in order to estimate simultaneously the relaxation rate α and the curvature parameter c of the speed–density law. On synthetic data generated by the forward model, the identification remains accurate over all tested configurations, with relative errors below $2\,\%$. A systematic study investigates the influence of the number of collocation points, the network architecture, and the optimizer. Additional experiments indicate that the parameter estimates remain stable when additive noise is introduced in the reference configuration. The results indicate limited sensitivity to these factors, while also making clear that the validation remains restricted to synthetic data and to the assumed macroscopic model.**

***Index Terms*—parameter estimation, macroscopic crowd dynamics, hyperbolic PDE, physics-informed neural networks, system identification.**

I. Introduction

Dense crowd motion has repeatedly led to tragic events, highlighting the need for better anticipation of congestion phenomena. Examples include the Love Parade disaster in Duisburg in 2010, the 2015 Hajj stampede in Mina near Mecca, and the 2022 Itaewon crowd crush in Seoul. These events share a common feature: the formation of high-density regions where pedestrian flow becomes unstable and difficult to control.

To analyze such situations, crowd models are commonly divided into microscopic, mesoscopic, and macroscopic approaches [1]–[3]. Microscopic models, including social-force formulations [4], [5], describe each individual pedestrian and can capture rich interactions, but their computational cost becomes prohibitive at large scale. By contrast, macroscopic models represent the crowd as a continuum through density and velocity fields [6], [7]. They are therefore better suited to large gatherings and to computationally affordable estimation frameworks.

Their operational use nevertheless requires the identification of physical parameters that are usually not directly measurable. Parameter estimation for distributed systems governed by PDEs is known to be challenging [1]. Classical approaches often rely on explicit discretization and may be sensitive to initialization, observation quality, and computational burden.

Physics-Informed Neural Networks (PINNs) [8], [9] provide an appealing alternative: the governing equations are embedded in the loss functional, while automatic differentiation [10] makes it possible to propagate gradients through the full model. This paradigm has attracted significant attention for inverse problems, but its use for parameter estimation in hyperbolic macroscopic crowd models remains only sparsely documented within the literature cited here.

In this work, we focus on the identification of two parameters in a Bellomo–Dogbé-type hyperbolic macroscopic crowd model: the relaxation rate α and the parameter c controlling the curvature of the speed–density law. The main difficulty is that these parameters influence both state reconstruction and physical consistency.

The paper is organized around five contributions. First, we implement a joint optimization of (α, c) and the network weights within a multi-term loss functional. Second, we study the influence of the number of collocation points. Third, we compare several network architectures. Fourth, we assess the effect of different optimizers. Fifth, we evaluate the sensitivity of the identified parameters to additive noise in the observations. To remain aligned with the actual level of evidence provided by the experiments, we deliberately adopt a cautious interpretation throughout the paper: the results support good stability in the considered synthetic setting, but they do not yet amount to a full validation on real data or under model mismatch.

II. Macroscopic model and 2D illustration

A. Overview of crowd models

Mathematical models of crowd motion are commonly organized into three broad families [1]: *microscopic*, *mesoscopic*, and *macroscopic* models. Microscopic models track each pedestrian explicitly. Mesoscopic models adopt a statistical description in terms of distributions over velocity or direction. Macroscopic models represent the crowd through continuous density and velocity fields.

Microscopic models are well suited when one seeks a detailed description of local interactions, avoidance behavior, or individual trajectories. A standard example is the social-force model, in which each pedestrian is subject to a relaxation toward a desired velocity together with repulsive interactions from other pedestrians and walls [4], [5]. While behaviorally expressive, this class of models becomes computationally demanding when the number of pedestrians grows.

Macroscopic models instead rely on a continuum description. Their core equation is mass conservation:

$$\partial_t \rho + \nabla \cdot (\rho \vec{U}) = 0,$$

where $\vec{U}$ denotes an effective velocity field. The various macroscopic formulations differ mainly in how this field is defined. In first-order models, the velocity is prescribed as a function of density or of a potential. In second-order models, the velocity itself obeys an evolution equation, which makes it possible to account more naturally for inertia and relaxation effects.

Among first-order macroscopic models, the Hughes model is a major reference. It couples the conservation law with an eikonal equation in order to determine a walking direction driven by both the domain geometry and the local density [1], [7]. This formulation is well suited to congestion avoidance, but it is less direct when one wishes to represent explicit inertia or relaxation effects.

Another important family is that of congestion-constrained models, such as Maury–Roudneff-Chupin-type formulations discussed in [1]. In that setting, the desired velocity is corrected through a projection mechanism or a pressure field so as to enforce a maximum-density constraint. Such models are especially relevant when non-overlap and saturated regions play a central role.

By contrast, second-order macroscopic models, such as the one used here, introduce an evolution equation for the velocity itself. This makes it possible to represent reaction time, acceleration, and transient effects more directly. This balance between a still-manageable structure and a richer dynamics is what motivates the choice of the Bellomo–Dogbé model in the present work. The model is not claimed to be a universal description of all crowd configurations; rather, it is used as a representative hyperbolic macroscopic model for evaluating parameter identification in a controlled framework.

From the viewpoint of parameter estimation, the macroscopic setting offers two major advantages. First, it drastically reduces the problem dimension compared with individual-based descriptions. Second, measurements obtained from cameras or sensing devices can naturally be aggregated into density and average-velocity fields, which are directly compatible with the model variables.

B. Model under study

We consider the Bellomo–Dogbé hyperbolic system [2], [6] on $\Omega \times [0,T]$, with $\Omega \subset \mathbb{R}^2$:

$$\begin{cases} \partial_t \rho + \nabla \cdot (\rho\, \vec{v}) = 0, \\ \partial_t \vec{v} + (\vec{v} \cdot \nabla)\vec{v} = \alpha\big(v_e(\rho)\, \vec{\nu}_0 - \vec{v}\big), \end{cases} \tag{1}$$

where $\rho \in [0,1]$ is the normalized density, $\vec{v} = (v_x, v_y)^\top$ is the mean velocity, and $\vec{\nu}_0$ is a known unit target direction. The parameter $\alpha > 0$ measures how quickly the velocity relaxes toward equilibrium.

The equilibrium speed is chosen as

$$v_e(\rho) = v_{\max}\left(1 - \exp\left(-\frac{c(1-\rho)}{\rho}\right)\right), \tag{2}$$

with $c > 0$ and known $v_{\max} > 0$. By continuous extension, the law satisfies $v_e(0) = v_{\max}$ and $v_e(1) = 0$. The parameter c controls the transition from free flow to congestion.

The choice of this model reflects a compromise between dynamical richness and parametric simplicity. System (1) remains simple enough to permit a systematic sensitivity analysis, while preserving a hyperbolic structure able to reproduce propagation and accumulation effects. In the present paper, this balance is particularly useful because the primary goal is not to introduce a new physical model, but to assess the ability of a PINN to jointly identify physical parameters and state fields.

C. 2D simulation with obstacle

To illustrate the behavior of the forward model, a 2D simulation is performed in a $20\,\mathrm{m} \times 10\,\mathrm{m}$ corridor with an exit door and a rectangular obstacle. The parameters are $v_{\max} = 1.4\,\mathrm{m/s}$, $\alpha = 2$, $c = 1.5$, and the initial density is high in the two left corners.

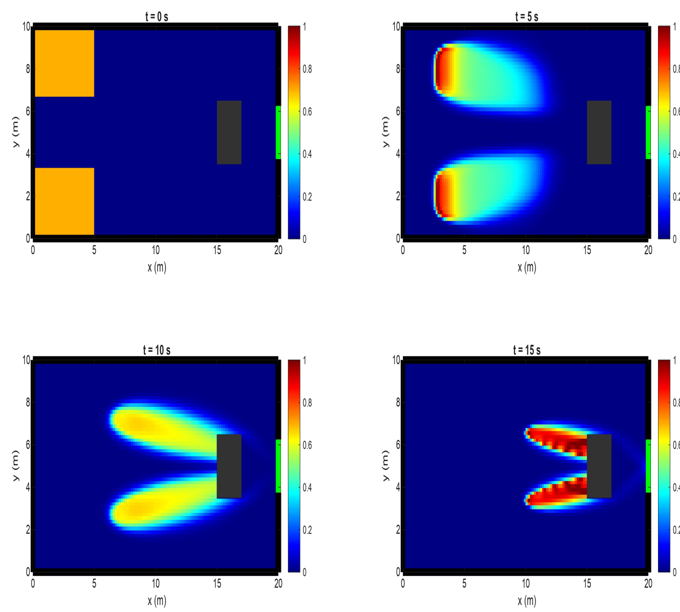

Figure 1. 2D simulation of (1): time evolution of the density $\rho(x, y, t)$ at $t = 0$, 5, 10, and 15 s. The figure illustrates channeling around the obstacle and a high-density region near the exit.

Fig. 1 is intended as a qualitative illustration only. It shows that the forward model generates plausible structures, such as accumulation near the door and obstacle-induced channeling. It is not used as a quantitative validation of the estimation procedure. The identification results reported below rely on controlled experiments with synthetic data generated by the forward model.

III. PINN METHODOLOGY

A. Approximation and residuals

Let $\{x_u^i, t_u^i, \rho^i, \vec{v}^{\,i}\}_{i=1}^{N_u}$ denote the observations and $\{x_f^j, t_f^j\}_{j=1}^{N_c}$ the collocation points. A multilayer perceptron with parameters $\boldsymbol{\theta}$ approximates the solution:

$$\big(\widehat{\rho}_{\boldsymbol{\theta}}(x,t), \widehat{\vec{v}}_{\boldsymbol{\theta}}(x,t)\big) \approx \big(\rho(x,t), \vec{v}(x,t)\big). \tag{3}$$

The physical parameters $\boldsymbol{\lambda} = (\alpha, c)$ are optimized jointly with $\boldsymbol{\theta}$.

The physics residuals associated with (1) are

$$f_1 := \partial_t \widehat{\rho}_{\boldsymbol{\theta}} + \nabla \cdot \big(\widehat{\rho}_{\boldsymbol{\theta}}\, \widehat{\vec{v}}_{\boldsymbol{\theta}}\big), \tag{4}$$

$$f_2 := \partial_t \widehat{\vec{v}}_{\boldsymbol{\theta}} + \big(\widehat{\vec{v}}_{\boldsymbol{\theta}} \cdot \nabla\big)\widehat{\vec{v}}_{\boldsymbol{\theta}} - \alpha\Big(v_e(\widehat{\rho}_{\boldsymbol{\theta}})\,\vec{\nu}_0 - \widehat{\vec{v}}_{\boldsymbol{\theta}}\Big), \tag{5}$$

and are computed by automatic differentiation [10].

B. Loss functional

The estimation of $(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is obtained by minimizing

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathcal{L}_{\text{data}} + \mathcal{L}_f + \mathcal{L}_{\text{IC}} + \mathcal{L}_{\text{BC}}, \tag{6}$$

where

$$\mathcal{L}_{\text{data}} = \frac{1}{N_u}\sum_{i=1}^{N_u}\Big[\big(\widehat{\rho}_{\boldsymbol{\theta}}(x_u^i, t_u^i) - \rho^i\big)^2 + \big\|\widehat{\vec{v}}_{\boldsymbol{\theta}}(x_u^i, t_u^i) - \vec{v}^{\,i}\big\|^2\Big],$$

$$\mathcal{L}_f = \frac{1}{N_c}\sum_{j=1}^{N_c}\Big[f_1(x_f^j, t_f^j)^2 + \big\|f_2(x_f^j, t_f^j)\big\|^2\Big].$$

The terms $\mathcal{L}_{\text{IC}}$ and $\mathcal{L}_{\text{BC}}$ enforce initial and boundary conditions and are defined by

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{\text{IC}}}\sum_{k=1}^{N_{\text{IC}}}\big\|\widehat{\mathbf{q}}_{\boldsymbol{\theta}}(x_{\text{IC}}^k, 0) - \mathbf{q}_0(x_{\text{IC}}^k)\big\|^2, \tag{7}$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_{\text{BC}}}\sum_{\ell=1}^{N_{\text{BC}}}\big\|\mathcal{B}[\widehat{\mathbf{q}}_{\boldsymbol{\theta}}](x_{\text{BC}}^\ell, t_{\text{BC}}^\ell) - \mathbf{g}(x_{\text{BC}}^\ell, t_{\text{BC}}^\ell)\big\|^2, \tag{8}$$

where $\mathbf{q} = (\rho, v_x, v_y)^\top$, $\mathbf{q}_0$ is the prescribed initial state, $\mathcal{B}$ denotes the boundary operator, and $\mathbf{g}$ is the prescribed boundary datum. These two definitions make explicit the loss terms appearing in (6).

C. Reference setting and optimizers

The reference setting uses Adam, $N_c = 1024$ collocation points, and an MLP with three hidden layers of 64 neurons. The activation is $\tanh$, the input is $(x, y, t) \in \mathbb{R}^3$, and the output is $(\rho, v_x, v_y) \in \mathbb{R}^3$. The number of observation points is $N_u = 10\,000$. Training is carried out for $15\,000$ iterations. In the following, this combination is referred to as the reference configuration.

The physical parameters are initialized at $\alpha_0 = c_0 = 0.5$. Four optimization strategies are compared: Adam [11], NatGrad [12], K-FAC [13], and EK-FAC [14]. The goal is not to introduce a new optimizer, but to assess how strongly the learning strategy affects the final identification.

D. Specific features of the compared optimizers

Adam. Adam is a first-order optimizer based on adaptive estimates of the first and second moments of the gradient. Let $g_t = \nabla\mathcal{L}(\theta_t)$. The update reads

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{9}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t, \tag{10}$$

followed by the bias-corrected step

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon}. \tag{11}$$

In the present context, Adam is a natural baseline because it is simple to implement, memory-efficient, and often robust in practice.

Natural Gradient. Natural-gradient methods attempt to account for the geometry of the parameter space by preconditioning the gradient with the inverse Fisher matrix:

$$\theta_{t+1} = \theta_t - \eta F(\theta_t)^{-1}\nabla\mathcal{L}(\theta_t). \tag{12}$$

The rationale is to adapt the descent direction to the local curvature of the problem. In practice, however, constructing and inverting the relevant matrix can be costly, which motivates structured approximations in high dimension.

K-FAC. K-FAC (*Kronecker-Factored Approximate Curvature*) relies on a layerwise approximation of the Fisher matrix. For each layer, the exact curvature is replaced by a Kronecker product of lower-dimensional factors that typically encode activation and gradient covariances. The resulting update can be written schematically as

$$\theta_{t+1} = \theta_t - \eta\, \widetilde{F}_t^{-1}\nabla\mathcal{L}(\theta_t), \tag{13}$$

where $\widetilde{F}_t$ denotes the K-FAC approximation of the curvature. The aim is to capture useful second-order information without incurring the cost of a fully exact method.

EK-FAC. EK-FAC refines K-FAC by correcting eigenvalues in the approximate curvature basis. In schematic form, the update keeps the same structure,

$$\theta_{t+1} = \theta_t - \eta\, \widetilde{F}_t^{\text{EK}\,-1}\nabla\mathcal{L}(\theta_t), \tag{14}$$

but uses a more faithful approximation of the curvature than standard K-FAC. In practice, EK-FAC seeks to improve the quality of the preconditioning at the price of a moderate overhead.

In this paper, these four optimizers are compared not to establish a universal ranking, but to determine whether the considered identification problem is strongly dependent on the training algorithm. This comparison helps clarify the trade-off between ease of implementation and the sophistication of the curvature information being exploited.

IV. Results

Observation data are generated numerically from the forward model with exact parameters $\alpha^\star = 2$ and $c^\star = 1.5$, then sampled over the spatio-temporal domain.

A. Reference result

For the reference configuration (Adam, $N_c = 1024$, architecture 3×64), the network converges to

$$\hat{\alpha} = 1.9760, \qquad \hat{c} = 1.5158,$$

which corresponds to relative errors of $1.20\,\%$ and $1.05\,\%$, respectively.

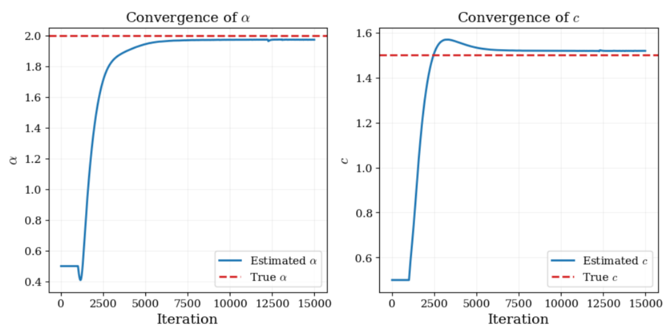

Figure 2. Convergence of $\hat{\alpha}$ and $\hat{c}$ toward the target values $\alpha^\star = 2$ and $c^\star = 1.5$ in the reference configuration. The estimate of α becomes smooth after the initial transient, while c exhibits a mild overshoot before stabilization.

Fig. 2 shows that the estimate of α evolves in a nearly monotonic manner after the first iterations. The parameter c undergoes a moderate overshoot above 1.5 and then stabilizes near its final estimate. In this experiment, this behavior is consistent with a progressive trade-off between data fitting and physics enforcement.

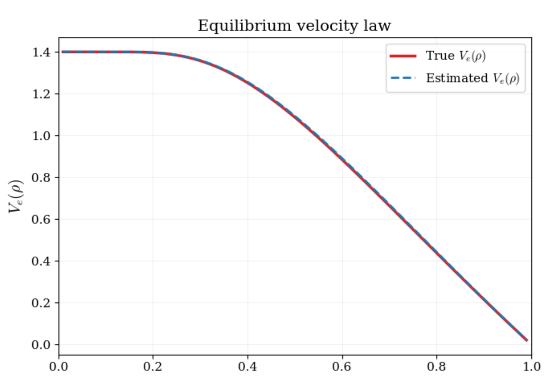

Figure 3. Equilibrium speed law $v_e(\rho)$ obtained with $c^\star = 1.5$ and $\hat{c} = 1.5158$. The two curves remain very close over the full interval $\rho \in [0, 1]$.

Fig. 3 confirms that the reconstructed equilibrium law remains close to the reference one over the full density range. This is consistent with the small error obtained for parameter c.

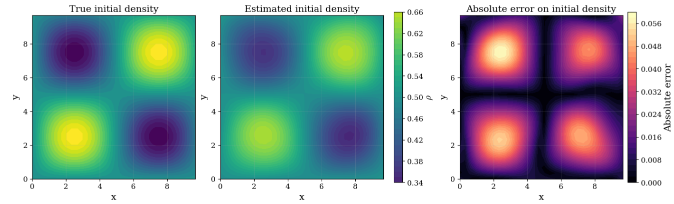

Figure 4. Initial density field: reference (left), PINN reconstruction (center), and absolute error (right). The largest discrepancies are concentrated in regions with the strongest spatial variations.

Fig. 4 indicates that the spatial structure of the initial density field is well reconstructed. The absolute error is mainly localized near extrema, that is, in regions where the gradients are largest. This is consistent with the expected behavior of a smooth neural approximation of a multi-peak profile.

B. Influence of the number of collocation points

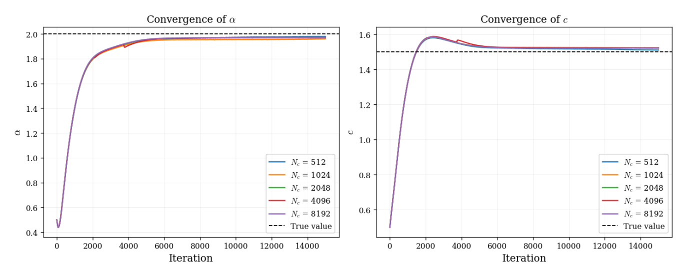

Figure 5. Convergence of $\hat{\alpha}$ and $\hat{c}$ for five values of N_c. The final trajectories are close, with only modest differences during the transient regime.

Table I
Influence of the number of collocation points N_c.

N_c	$\hat{\alpha}$	$\varepsilon_\alpha(\%)$	$\hat{c}$	$\varepsilon_c(\%)$
512	1.9786	1.07	1.5115	0.77
1 024	1.9760	1.20	1.5158	1.05
2 048	1.9722	1.39	1.5222	1.48
4 096	1.9701	1.49	1.5240	1.60
8 192	1.9706	1.47	1.5220	1.46

Fig. 5 and Table I show that the final estimates remain of the same order when N_c varies from 512 to 8192. The $N_c = 512$ case gives the best estimate in this experiment, but the gap with the other settings is small. It is therefore more appropriate to conclude that the method exhibits limited sensitivity to N_c over the tested range than to claim the existence of a clear optimum.

C. Influence of the architecture

Table II
RANKING OF THE 10 BEST ARCHITECTURES AMONG 16 TESTED ONES. SCORE $s = \varepsilon_\alpha + \varepsilon_c$.

Rank	L	W	$\hat{\alpha}$	ε_α	$\hat{c}$	ε_c
1	5	32	1.9683	1.59	1.5220	1.47
2	5	96	1.9666	1.67	1.5224	1.49
3	4	64	1.9666	1.67	1.5224	1.50
4	5	128	1.9658	1.71	1.5219	1.46
5	4	96	1.9663	1.69	1.5226	1.50
6	4	128	1.9656	1.72	1.5223	1.48
7	5	64	1.9658	1.71	1.5231	1.54
8	3	128	1.9652	1.74	1.5230	1.53
9	4	32	1.9649	1.76	1.5240	1.60
10	3	64	1.9649	1.75	1.5242	1.62
13	2	96	1.9455	2.72	1.5323	2.15
16	2	32	1.9285	3.57	1.5436	2.90

Table II highlights two trends. First, architectures with two hidden layers are consistently less accurate than those with at least three hidden layers. Second, among architectures of depth three or larger, the differences remain modest. In this context, the 3×64 architecture is not the best in absolute terms, but it provides a reasonable balance between accuracy and model size.

D. Influence of the optimizer

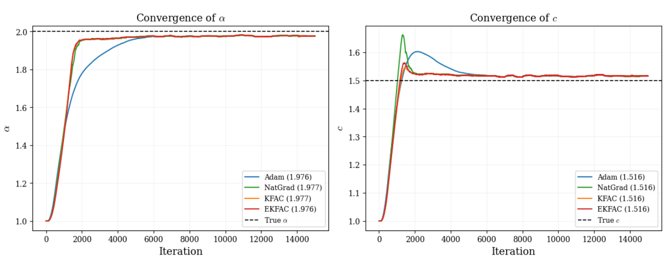

Figure 6. Convergence of $\hat{\alpha}$ and $\hat{c}$ for four optimization strategies. Second-order methods slightly accelerate the early stage on α, while K-FAC and EK-FAC yield smoother trajectories than NatGrad on c.

Table III
COMPARISON OF OPTIMIZATION STRATEGIES FOR $N_c = 1024$ AND ARCHITECTURE 3×64.

Method	$\hat{\alpha}$	$\varepsilon_\alpha(\%)$	$\hat{c}$	$\varepsilon_c(\%)$
Adam	1.9760	1.20	1.5158	1.05
NatGrad	1.9767	1.17	1.5161	1.07
K-FAC	1.9767	1.17	1.5157	1.05
EK-FAC	1.9758	1.21	1.5157	1.05

Fig. 6 and Table III show that all four methods reach very similar final values. The main differences concern the transient regime. On α, NatGrad, K-FAC, and EK-FAC converge slightly faster at the beginning of training. On c, NatGrad displays the largest overshoot, whereas K-FAC and EK-FAC lead to somewhat smoother trajectories. Final discrepancies nevertheless remain small.

E. Sensitivity to noisy observations

To evaluate the sensitivity of the identification to measurement noise, the observation data are perturbed before training by additive Gaussian noise in the reference configuration:

$$z^\eta = z + \eta\, \sigma_z\, \xi, \qquad \xi \sim \mathcal{N}(0, 1), \tag{15}$$

where $z \in \{\rho, v_x, v_y\}$, σ_z is the empirical standard deviation of z, and η is the prescribed noise level. The noisy data are used in the data-mismatch term, while the PDE residuals remain evaluated through the neural-network approximation.

Table IV
SENSITIVITY TO NOISY OBSERVATIONS IN THE REFERENCE CONFIGURATION.

Noise level	$\hat{\alpha}$	$\varepsilon_\alpha(\%)$	$\hat{c}$	$\varepsilon_c(\%)$
0%	1.9716	1.42	1.5185	1.23
1%	1.9745	1.28	1.5187	1.25
5%	1.9738	1.31	1.5188	1.25
10%	1.9728	1.36	1.5188	1.26
20%	1.9707	1.46	1.5187	1.24

In the present synthetic setting, the parameter estimates remain stable when the noise level increases up to $20\,\%$. The relative errors remain below $1.5\,\%$ for both parameters, suggesting that the physics residual contributes to regularizing the inverse problem. This conclusion should nevertheless be interpreted within the limits of the chosen additive noise model.

V. DISCUSSION

The results support the idea that joint optimization of state variables and physical parameters can be performed in a stable way for this synthetic problem. The reference case shows that both parameters can be recovered from an initialization deliberately chosen far from the exact values. The study with respect to N_c suggests that, over the explored range, the physical information injected through collocation points is already sufficient with a few hundred samples. The architecture study indicates that insufficient depth degrades estimation quality, whereas improvements become modest beyond three hidden layers. The optimizer comparison shows that most differences concern the transient regime rather than final accuracy.

From a practical standpoint, Adam remains competitive at the end of training, despite the use of methods that incorporate richer curvature information. This supports the use of a simple optimization pipeline when the main objective is the final estimate. Likewise, increasing N_c far beyond a few hundred collocation points does not produce a clear improvement in this experiment.

The scope of these conclusions should nevertheless remain clearly bounded. The observations are synthetic and generated by the same model used for inversion; hence the study does not yet address model mismatch. The noise analysis is limited to additive Gaussian perturbations of synthetic observations and does not replace validation on real measurements. These points do not undermine the internal consistency of the study, but they define what can be concluded from the reported experiments.

VI. CONCLUSION

We presented a PINN-based approach for the simultaneous identification of parameters (α, c) in a macroscopic crowd model governed by a hyperbolic PDE. The experiments performed on synthetic data show that accurate estimation can be achieved through joint optimization of the physical parameters and the network weights. The parametric analysis also indicates limited sensitivity to the number of collocation points, to the optimizer, and, beyond a certain threshold, to the network architecture. The additional noise-sensitivity experiment shows that the identification remains stable under additive Gaussian noise up to $20\,\%$ in the considered setting.

These conclusions must however be interpreted in light of the chosen experimental protocol: the validation remains synthetic and does not include real measurements or model discrepancy. Natural next steps therefore include partial measurements, stronger observation imperfections, and experiments closer to real-world conditions.

VII. FUTURE WORK

1. Adaptive weighting of the loss terms. The transient behavior observed for c suggests that adaptive balancing of the different terms in (6) could further improve stability during the early training phase, in line with the analyses reported in [15].

2. More targeted collocation sampling. While the present study indicates limited global sensitivity to N_c, it does not address whether non-uniform sampling would be beneficial in subregions with stronger spatial variations.

3. Partial and real-world observations. Although additive noise has been considered in the present work, an important next step is to assess the identification procedure under incomplete measurements, structured noise, and field data obtained from sensing or video-based systems.

4. Radial basis activation functions. Replacing or complementing $\tanh$ activations with radial basis functions is a promising direction, since RBFs may improve local approximation in regions with strong spatial gradients.

5. Extension of the physical model. The PINN framework considered here could be extended to enriched macroscopic dynamics, for instance by incorporating additional hyperbolic terms inspired by traffic or crowd models [6], provided that the experimental protocol is adapted accordingly.

6. Control and decision support. Once the parameters have been calibrated, the identified model could be embedded into a control or decision-support loop in order to optimize evacuation strategies, for instance through door management, dynamic flow orientation, or adaptive guidance policies.

7. Model reduction for real-time execution. For embedded or online supervision applications, coupling the present framework with model-reduction techniques appears to be an important direction for lowering the computational burden and moving the identification or prediction process closer to real-time execution.

REFERENCES

[1] G. Jradi, *Des EDPs pour le mouvement de foule*, Ph.D. thesis, Université de Limoges, 2023.

[2] N. Bellomo, C. Dogbé, "On the modeling of traffic and crowds: a survey of models, speculations, and perspectives," *SIAM Rev.*, vol. 53, no. 3, pp. 409–463, 2011.

[3] E. Cristiani, B. Piccoli, A. Tosin, *Multiscale Modeling of Pedestrian Dynamics*. Springer, 2014.

[4] D. Helbing, A. Johansson, "Pedestrian, crowd, and evacuation dynamics," in *Encyclopedia of Complexity and Systems Science*, Springer, 2012, pp. 6476–6495.

[5] D. Helbing, P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[6] N. Bellomo, C. Dogbé, "On the modelling crowd dynamics from scaling to hyperbolic macroscopic models," *Math. Models Methods Appl. Sci.*, vol. 18, pp. 1317–1345, 2008.

[7] R. L. Hughes, "A continuum theory for the flow of pedestrians," *Transp. Res. Part B*, vol. 36, no. 6, pp. 507–535, 2002.

[8] M. Raissi, P. Perdikaris, G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[9] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, pp. 422–440, 2021.

[10] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, "Automatic differentiation in machine learning: a survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, 2018.

[11] D. P. Kingma, J. Ba, "Adam: a method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, 2015.

[12] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.

[13] J. Martens, R. Grosse, "Optimizing neural networks with Kronecker-factored approximate curvature," in *Proc. Int. Conf. Machine Learning*, PMLR, vol. 37, pp. 2408–2417, 2015.

[14] T. George, C. Laurent, X. Bouthillier, N. Ballas, P. Vincent, "Fast approximate natural gradient descent in a Kronecker-factored eigenbasis," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[15] S. Wang, X. Yu, P. Perdikaris, "When and why PINNs fail to train: a neural tangent kernel perspective," *J. Comput. Phys.*, vol. 449, art. 110768, 2022.