Journal of Micro and Bio Robotics (2025) 21:17
https://doi.org/10.1007/s12213-025-00197-0

RESEARCH

Vision-based learning of magnetic microrobotic swarm using fast dimensionality reduction and feature extraction

Seyyed Shaho Alaviani[1] · Jaideep Katuri[2,3] · Jamel Ali[2,3] · Olugbenga Moses Anubi[4]

Received: 5 November 2024 / Revised: 23 October 2025 / Accepted: 25 October 2025 / Published online: 22 November 2025

Abstract

This paper aims at predicting the behavior of untethered magnetic microrobotic swarm (MMS), which also include emergent behaviors, from raw grayscale time-series image data. A fast dimensionality reduction algorithm for compressing image data set is proposed. Then based on the algorithm, an *interpretable* machine learning method which is based on Proper Orthogonal Decomposition (POD) and polynomial regression is proposed for feature extraction of compressed image data set and for further dimensionality reduction. The proposed method is reduced-order and purely data-driven not assuming any knowledge of the physics and/or interacting dynamics of MMS, which are extremely challenging to ascertain for systems of this nature and dimension. Finally, experimental results on several kinds of synthetic MMs show the effectiveness of the proposed method with less prediction error while using more dimensionality reduction than existing approaches in the literature.

Keywords Micro vision · Vision-based learning · Dimensionality reduction · Magnetic microrobotic swarm

1 Introduction

Microrobots are defined as an autonomous or semi-autonomous systems [1] with dimension range from micrometers to millimeters [2]. Such properties of microrobots make them very effective in reaching inaccessible areas, specially in the human body, but very difficult to visualize and track. Recently, microrobots have attracted a lot of attention due to their *in-vitro* and *in-vivo* applications such as in industry [3] and biomedical applications [4, 7]. Minimally invasive surgery and micro-object assembly can be exemplified for the biomedical and industrial applications, respectively.

Due to very small scale, microrobots cannot be actuated by a built-in power supply, and they need to be actuated from an external drive method such as magnetic, electric, optical, acoustic, chemical, and biological actuations (see Survey [7] for details). Hybrid actuation mechanism is also possible [7]. Among them, magnetic actuation has attracted much attention, specially in biomedical applications, due to deep tissue penetration and biological harmlessness [8] as well as invulnerability to environmental fluid [9].

Magnetic fields are typically generated by two ways: by *permanent magnets* and *electromagnetic coils*. Electromagnetic coils have the major advantage that they can deliver varying fields without moving parts. However, permanent magnets can provide large fields without the use of large electrical currents, but they cannot be turned off without moving the external magnets far from the workspace. Moreover, permanent magnets need additional mechanical devices to control the rotation and movement, that is

✉ Seyyed Shaho Alaviani
salavian@umn.edu

Jaideep Katuri
jkaturi@fsu.edu

Jamel Ali
jali@eng.famu.fsu.edu

Olugbenga Moses Anubi
oanubi@fsu.edu

[1] Department of Mechanical Engineering, University of Minnesota, 111 Church Street SE, Minneapolis 55455, MN, USA

[2] Department of Chemical and Biomedical Engineering, FAMU-FSU College of Engineering, 2525 Pottsdammer Street, Tallahassee 32310, FL, USA

[3] National High Magnetic Field Laboratory, Organization, 1800 E Paul Dirac Dr., Tallahassee 32310, FL, USA

[4] Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering, 2525 Pottsdammer Street, Tallahassee 32310, FL, USA

complex and may need larger spaces for operation. Thus, permanent magnets are not suitable for microrobot locomotion in practice, whereas electromagnetic coils are practically suitable to do such task.

Electromagnetic coils can generate *gradient*, *oscillating*, or *rotational* magnetic fields to actuate magnetic microrobots (MMs) [10]. Gradient magnetic field can make static force to a MM [10], while it usually needs large magnetic field strength to make enough actuation force. On the other hand, rotational and oscillating magnetic fields are more efficient in actuating MMs by generating torque with low magnetic force strength [10]. In general, the actuated MM can be classified into two categories: *tethered* devices such as intravascular microcatheters and *untethered* devices such as helical swimmers [2]. Since an untethered MM must be powered and controlled in a wireless manner, it poses more challenges. Microrobots can also be classified as *synthetic* and *bio-hybrid*. In the former case, the microrobot is made of fully synthetic materials such as magnetic materials, silicon oxide, and metals, while in the latter case it is made of both biological and synthetic materials.

Microrobotics research is particularly interested in understanding and controlling swarm behavior [11]. From a practical point of view, simultaneously navigating a large swarm of micro/nanorobots is often of great importance, for example, in the delivery of therapeutic agents at appropriate levels and then tracking them inside the body [12]. During locomotion, one MM may attach to others to form an unstructured larger cluster called a *growing magnetic microrobot (GMM)* whose size and dynamics are changing [13][1]. Such behavior is called *emergent behavior* [1, 10, 14]. In this work we define *behavior* as the translational and rotational motion of individual magnetically actuated particles and particle swarms, whereas *emergent behavior* refers to the structure of colloidal clusters that appear due to collective, coordinated interactions between particles and their surrounding environment. Hence, modeling and prediction of spatio-temporal behavior of magnetic microrobotic swarm (MMS) in complex environments are still challenging [12, 15], while the development of automated algorithms for microrobots is still an open and under-explored research topic [16].

An approach to predict the future behavior of microrobotic swarm (or *microswarm*) is to discover the underlying interacting dynamics of all microrobots by using *model-based* approaches. Nevertheless, since GMM dynamic behavior changes a lot frequently [13], such methods will provide prediction with more errors than what expected. Thus, *data-driven* methods are useful and practical methods to do so. Several investigators have probed data-driven [17, 18] and hybrid (i.e., combined model-based and data-driven approaches) [19, 22] modeling of generic (i.e., macro or micro) multi-agent swarm where in their proposed method measuring position, velocity, and/or orientation of *all* agents is performed. The authors in [1] and [23] have considered data-driven modeling of emergent behavior of microrobots, where in their proposed method measuring position, velocity, and/or orientation of *all* microrobots is performed (note that in [23], velocities of MMs are calculated from a predefined probability distribution).

However, it is still challenging to directly detect motion of one microrobot (or one GMM) since there is no onboard sensing, and the microrobot's movement is tracked and controlled by image-based, eye-to-hand (or better, eye-to-microrobot) visual survey (called *micro-vision*) [13, 24–26]. Furthermore, dimension of such data increases dramatically due to very large number of microrobots in a small area. Recently, the authors developed a data-driven interpretable machine learning (ML) method in [27] to overcome such challenges where dimensionality reduction is performed to predict future time-series images. The method in [27] also crosses the reality gap (see [27] for details). The result in [27] needs each two-dimensional matrix datum to be converted to *one-dimensional vector* datum (by concatenating all rows or columns of an image), and then Proper Orthogonal Decomposition (POD) [28, 30] is applied to the data matrix. The vector space model used in [27] computes Singular Value Decomposition (SVD) of the data matrix. *However, the SVD computation restricts its applicability to matrices of small size* [31].

Most existing ML methods take *vector* data for learning. If the data are in the other formats such as matrices, traditional ML methods need to convert them into vectors and then apply learning processes. However, this conversion can make serious issues: firstly, the converted vectorization data make vector-based methods suffer from curse of dimensionality for high-dimensional data [32]; secondly, it is well-known that fast increase of number of free variables results in over-fitting of the learning model; thirdly, vectorization fails to utilize the spatial structure information embedded in pixels of the image [33]; fourthly, vectorization of high-dimensional data increases computational complexity of a proposed learning algorithm; lastly (but not the least), the high-dimensional data usually contain many redundant or irrelevant features that can cause low efficiency and poor learning performance [34]. *Upon review, since the result in* [27] *uses vectorization, it suffers from aforementioned issues as well as applicability to matrices of small size.*

[1] We require to mention that an indeterministic state-observer based on micro-vision, where little information about GMM dynamic behavior is needed, has been developed in [13] to track *one* GMM; however, we study MMS behavior which can include many or no GMMs *without* any knowledge about GMM dynamic behavior.

Contributions In this paper, we address the aforementioned issues that arise in the result in our previous work in [27]. Contributions of this work are summarized as follows:

- We propose *Lanczos-based Non-Iterative Generalized Low Rank Approximations of Matrices (Lanczos+NIGLRAM)* by applying Lanczos algorithm [35]-[36, Subsec. 2.2] for obtaining most dominant eigenvectors to NIGLRAM [37] faster (related literature review is given in Subsection I.A below). This results in fast compression and feature extraction of high-dimensional image data set under which each datum is represented as a matrix and the collection of data is represented as a collection of matrices and can be applied to matrices of high size. Note that compression of original image data is *not* performed in [27].
- We apply POD [28, 30] on the compressed image data sets to extract most dominant *spatial* basis functions from even smaller number (compared to dimension of compressed images) of data. As mentioned in [31], the reconstruction (or training) error of Generalized Low Rank Approximations of Matrices+SVD (GLRAM+SVD) or similarly GLRAM+POD (see Subsection II for details) is close to that of SVD, especially when the intermediate dimension in the GLRAM stage is large, while it has a smaller computation cost than SVD. Consequently, the training error of the result in this paper *may* be close to that of [27] while it has smaller computation cost. Since we use Lanczos+NIGLRAM with POD on original images, the computation cost reduces more than that of GLRAM+SVD alone.
- We combine polynomial regression (in time) and Lanczos+NIGLRAM with POD to predict the MMS behavior (which includes emergent behaviors) in a totally data-driven way without any knowledge about the physics and/or interacting dynamics of MMs.
- We calculate both time complexity and space complexity of the proposed algorithm here and those of the algorithm in [27] and show that space complexity is significantly reduced in the proposed algorithm.
- We provide experimental results of three different synthetic MMs (namely, ellipsoidal, cubic, and rod-shaped MMs) where the results present that the prediction error is less than that of our previous work in [27], while in some cases the training error is also less than that of [27] (see Examples 2–3).

1.1 Related work

There has been a dramatic increase in the availability and use of high-dimensional data set such as images, videos etc. Since direct operation on high-dimensional data set is computationally inefficient and in some cases impossible, *dimensionality reduction (DR)* has received much attention in areas such as ML and computer vision. DR has been an important technique to compress data representation. DR reduces the number of variables or features so that the underlying model requires fewer parameters to be represented.

Literature Review on DR of Two-Dimensional (2D) Data The two-dimensional Principle Component Analysis (2DPCA) was proposed in [38] as an efficient technique to deal with 2D images. A disadvantage of 2DPCA is that it can only work on one-sided linear transform which results in low compression ratio. $(2D)^2$PCA method was developed in [39] which makes higher compression ration than 2DPCA. A disadvantage of $(2D)^2$PCA is that it is susceptible to lighting intensity and shooting angles. GLRAM [31] performs double-sided dimensionality reduction and has higher compression ration than 2DPCA and $(2D)^2$PCA. A disadvantage of GLRAM is that it needs large computation due to iterative procedure to compute the two-sided transformation. In this regard, several algorithms, briefly introduced next, have been proposed to reduce the large computation:

1) *NIGLRAM* [37] where the two-sided transformation in GLRAM is obtained via *non-iterative* way by calculating some dominant eigenvectors. The authors of [37] did not suggest any methods to calculate eigenvectors. The authors of [42] noted that SVD is used in GLRAM to calculate eigenvectors.
2) *Bilinear Lanczos Components (BLC)* [36] where Lanczos algorithm [35] to compute some dominant eigenvectors is used in GLRAM, *where still the two-sided transformation is calculated in iterative way.*
3) *Thick-Restarted Bilinear Lanczos Components (TRBLC)* [40] uses Ritz vectors to compute some dominant eigenvectors to reduce outer iteration less than that of BLC. *Thus TRBLC still calculates the two-sided transformation iteratively.*
4) *Simplified GLRAM (SGLRAM)* [41] uses *non-iterative* algorithm to obtain the two-sided transformation in GLRAM by using SVD.
5) *Randomized GLRAM* [42] applies randomized SVD (RSVD) [43] to take place in the SVD of GLRAM, SGLRAM, and NIGLRAM. Since RSVD produces a good approximation when singular values of a matrix decay sufficiently rapidly, randomized GLRAM performs well on such matrices. However, *it still calculates the two-sided transformation in iterative way.*

Upon review, only NIGLRAM and SGLRAM are non-iterative. Since they use SVD (or Eigen-Value Decomposition (EVD)) for computing eigenvectors, the time complexity of

them is cubic, namely $O(r^3)$ where *O* represents big o, and r is the size of a square matrix. *To reduce the time complexity, we apply Lanczos algorithm* [35]-[36, Subsec. 2.2] *to NIGLRAM, that has not been proposed in the literature, to reduce the time complexity more than those of NIGLRAM and SGLRAM.*

This paper is organized as follows. In Section 2, brief introduction of NIGLRAM, the Lanczos algorithm, and POD are provided. In Section 3, architecture of the micro-vision platform used in this study and formulation of the problem considered are given. The Lanczos+NIGLRAM and the proposed method are presented in Seubsection 4.1 and 4.2, respectively. Finally, the experimental results are provided in Section 5, followed by conclusions in Section 6.

Notations $\mathbb{R}$ denotes the set of all real numbers. $\mathbb{N}$ denotes the set of all natural numbers. $\mathbb{R}^{r\times c}$ represents the set of all real $r \in \mathbb{N}$ by $c \in \mathbb{N}$ matrices. $(.)^\top$ represents transpose of a vector or a matrix. $|.|$ represents absolute value. $span\{a_1, a_2, \ldots, a_m\}$ denotes the set of all linear combinations of $a_1, a_2, \ldots, a_m$. 0_m represents the vector of dimension *m* whose elements are all zero. For any vector $z \in \mathbb{R}^n$, $\|z\|_2 = \sqrt{z^\top z}$. Frobenius norm of a matrix $A = [a_{ij}] \in \mathbb{R}^{n\times m}$ is $\|A\|_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}^2}$. I_n denotes Identity matrix of size $n \in \mathbb{N}$.

2 Preliminaries

2.1 NIGLRAM [37]

Let $\{A_i\}_{i=1}^m$, where $A_i \in \mathbb{R}^{r\times c}$, $m \in \mathbb{N}$, $r \in \mathbb{N}$, and $c \in \mathbb{N}$, be the training data set. NIGLRAM aims to calculate in *non-iterative way* two matrices $L \in \mathbb{R}^{r\times l_1}$ and $R \in \mathbb{R}^{c\times l_2}, l_1 \in \mathbb{N}, l_2 \in \mathbb{N}$, with orthonormal columns and *m* matrices $M_i \in \mathbb{R}^{l_1\times l_2}, i = 1, \ldots, m$, such that $LM_iR^\top$ approximates A_i for all *i*, where l_1 and l_2 are two pre-specified parameters. In fact, *L*, *R*, and M_i solve the following optimization problem [31]:

$$\begin{aligned} \min_{\{M_i\}_{i=1}^m, L, R} \quad & \sum_{i=1}^{m} \|A_i - LM_iR^\top\|_F^2 \\ \text{subject to} \quad & L^\top L = I_{l_1} \\ ; \quad & ; R^\top R = I_{l_2} \end{aligned} \tag{1}$$

Require: $\{A_i\}_{i=1}^m$, l_1, and l_2
Ensure: *L* and *R*
1: Form matrix $N_L \leftarrow \sum_{i=1}^m A_i A_i^\top$.
2: Compute *L* which is the l_1 eigenvectors of N_L corresponding to the first largest l_1 eigenvectors.
3: Form matrix $N_R \leftarrow \sum_{i=1}^m A_i^\top L L^\top A_i$.
4: Compute *R* which is the l_2 eigenvectors of N_R corresponding to the first largest l_2 eigenvalues.

Algorithm 1 NIGLRAM [37].

Remark 1 The time complexity of NIGLRAM has been mentioned in [37] to be $O(m(r\times c)^{\frac{3}{2}})$ without justification/proof. But we believe that it is $O(r^3 + c^3 + mc^2l_1 + mcrl_1 + mr^2c)$ that is still cubic with respect to *r* and *c*. *Justification:* it takes $O(mr^2c)$ time for computing N_L and $O(m(crl_1 + c^2l_1))$ time for computing N_R (see also [31, Subsec. 3.3]). Since SVD is used in NIGLRAM for calculating eigenvectors (see [42, Sec. 1]) and time complexity of one SVD is cubic, it takes $O(r^3)$ time for computing eigenvectors of N_R and $O(c^3)$ time for computing eigenvectors of N_L. Therefore, the exact time complexity of NIGLRAM, with respect to *r*, *c*, and l_1, is $O(r^3 + c^3 + mc^2l_1 + mcrl_1 + mr^2c)$.

2.2 Lanczos algorithm [35]-[36, Subsec. 2.2]

Given a *symmetric* matrix $D_i \in \mathbb{R}^{n\times n}, n \in \mathbb{N}$, and an initial unit vector $q_1 \in \mathbb{R}^n$, the Lanczos algorithm builds an orthonormal basis of the order-*k* subspace, i.e., $span\{q_1, Mq_1, \ldots, M^{k-1}q_1\}$. The time complexity of the Lanczos algorithm is $O(kn^2)$.

Require: M, q_1, and k
Require: **(Initialization)** $\beta_1 = 0$ and $q_0 = \mathbf{0}_n$
Ensure: $\{q_i\}_{i=1}^{k+1}$
1: **for** $i = 1, 2, \ldots, k$ **do**
2: $w_i \leftarrow Mq_i - \beta_i q_{i-1}$
3: $\alpha_i \leftarrow w_i^T q_i$
4: $w_i \leftarrow w_i - \alpha_i q_i$
5: $w_i \leftarrow w_i - \sum_{j=1}^{i-1}(w_i^\top q_j)q_j$
6: $\beta_{i+1} \leftarrow \|w_i\|_2$
7: **if** $\beta_{i+1} = 0$, **then**
8: Stop
9: **end if**
10: $q_{i+1} \leftarrow \frac{w_i}{\beta_{i+1}}$
11: **end for**

Algorithm 2 The Lanczos Algorithm [36, Subsec. 2.2].

2.3 Proper orthogonal decomposition (POD) [28, 30]

POD which is a powerful and elegant method of data analysis aimed at obtaining low dimensional approximate descriptions of high-dimensional processes. POD is also known as

Principle Component Analysis, Karhunen-Loéve Decomposition, Hotelling Transformation, Empirical Orthogonal Functions, and *Singular Value Decomposition*.

POD aims to decompose a function $u(x, t)$ into a set of deterministic spatial functions $\psi_k(x)$ modulated by time coefficients $a_k(t)$ so that

$$u(x,t) \approx \sum_{k=1}^{\bar{M}} a_k(t)\psi_k(x), \tag{2}$$

where $\bar{M} \in \mathbb{N}$, with the reasonable expectation that the approximation becomes exact in the limit as $\bar{M}$ approaches infinity. The $\psi_k(x)$ are the *POD basis vectors* or *POD modes*, and the $a_k(t)$ are their time coefficients. Let $Y := [u_1, \ldots, u_m] \in \mathbb{R}^{n\times m}$ be a real-valued matrix, called *data matrix*, whose columns $u_j \in \mathbb{R}^n, 1 \le j \le m$, are called snapshots.

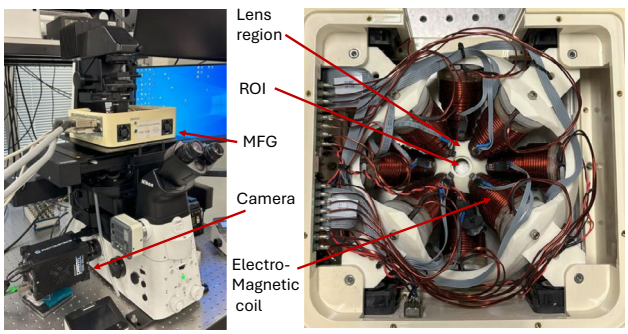
Lemma 1 *[29, Theorem 3.3.1]. (POD basis) Let* $Y := [u_1, \ldots, u_m] \in \mathbb{R}^{n\times m}$ *with rank* $d \le min\{m, n\}$ *and* $n, m \in \mathbb{N}$. *Moreover, let* $Y = \Psi\Sigma\Phi^\top$ *be the singular value decomposition (SVD) of Y. Then for* $1 \le r \le d, r \in \mathbb{N}$, *the optimization problem*

$$\begin{aligned} &\max_{\tilde{\psi}_1,\ldots,\tilde{\psi}_r\in\mathbb{R}^n} && \sum_{i=1}^{r}\sum_{j=1}^{m} |u_j^\top \tilde{\psi}_i|^2 \\ &subject to && \tilde{\psi}_i^\top \tilde{\psi}_j = \delta_{ij}, \forall 1 \le i, j \le r \end{aligned} \tag{3}$$

is being solved by the left singular vectors $\{\psi_i\}_{i=1}^r$ *where*

$$\delta_{ij} := \begin{cases} 1, & if \ \ i \neq j. \\ 0, & otherwise. \end{cases}$$

Remark 2 Two common methods to calculate POD modes are via 1) SVD of *Y* and 2) (sorted) eigen decomposition of Y^TY. A computationally efficient technique to obtain the POD modes accurately has been developed in [44].



(a) Micro-vision platform (b) Magnetic Field Generator (MFG)

Fig. 1 (a) Micro-vision platform, and (b) magnetic field generator used in this study

3 Architecture of micro-vision platform and problem formulation

3.1 Architecture of micro-vision platform

Similar to our previous work in [27], a micro-vision platform shown in Fig. 1 (a) is developed to create MMS under fixed rotational magnetic field. The micro-vision platform contains the following parts: an inverted microscope with a camera, a magnetic field generator (MFG), and one PC performing real-time observation. The MFG system was controlled using custom MATLAB scripts.

The MFG system is utilized for generating rotational magnetic field and is composed of 8 asymmetrically arranged electromagnetic coils as shown in Fig. 1 (b). Similar to our previous study, the region of interest (ROI) (100 milli-meter in diameter) is chosen at the center of the MFG system.

3.2 Problem formulation

Our formulation for vision-based learning of swarm behavior of untethered MMs is the same as that of our previous work in [27]. We have a grayscale video (or time-series image) data set obtained from the camera for untethered MMs in fluid actuated by rotational magnetic field from the MFG system.

Precisely, we have time-series grayscale images $\mathcal{I}_{t_i} \in \mathbb{R}^{q\times w}$, where $q \in \mathbb{N}$ and $w \in \mathbb{N}$, captured by the camera at time $t_i, i = 1, \ldots, m$, uniformly in the time interval $[0, \mathrm{T}], \mathrm{T} \in \mathbb{R}$, namely $\Delta t_i := t_{i+1} - t_i, t_1 = 0, t_m = \mathrm{T}$, where $t_{i+1} \ge t_i$ and $\Delta t_i = \frac{\mathrm{T}}{m-1}, i = 1, \ldots, m-1$.

The key difference between formulation of the problem here and that of our previous work in [27] is that image data are *vectorized* in [27], whereas they are directly used and are *not* vectorized in this paper. Under above consideration, the problem is to predict future images $\{\mathcal{I}_{t_i}\}_{i=m+1}^{m+P}$ for a fixed $P \in \mathbb{N}$ from available images $\{\mathcal{I}_{t_i}\}_{i=1}^{m}$.

4 Main results

In this section, we propose Lanczos+NIGLRAM in Subsection 4.1, then based on the results in Subsection 4.1, we propose our method to predict MMS behaviors of untethered MMs from time-series image data in Subsection 4.2.

4.1 Lanczos+NIGLRAM

Require: $\{\mathcal{I}_{t_i}\}_{i=1}^m$, $\tilde{l}_1$, and $\tilde{l}_2$
Ensure: $\tilde{L}$ and $\tilde{R}$
1: Form matrix $\tilde{N}_L \leftarrow \sum_{i=1}^m \mathcal{I}_{t_i}\mathcal{I}_{t_i}^\top$.
2: Select unit vector $q_1 \in \mathbb{R}^q$.
3: $M \leftarrow \tilde{N}_L$
4: $k \leftarrow \tilde{l}_1$
5: Apply Algorithm 2.
6: $\tilde{L} \leftarrow [q_1, q_2, \ldots, q_{\tilde{l}_1}] \in \mathbb{R}^{q\times\tilde{l}_1}$.
7: Form matrix $\tilde{N}_R \leftarrow \sum_{i=1}^m \mathcal{I}_{t_i}^\top \tilde{L}\tilde{L}^\top \mathcal{I}_{t_i}$.
8: Select unit vector $q_1 \in \mathbb{R}^w$.
9: $M \leftarrow \tilde{N}_R$
10: $k \leftarrow \tilde{l}_2$
11: Apply Algorithm 2.
12: $\tilde{R} \leftarrow [q_1, q_2, \ldots, q_{\tilde{l}_2}] \in \mathbb{R}^{w\times\tilde{l}_2}$.

Algorithm 3 Lanczos+NIGLRAM

Lanczos+NIGLRAM is simply applying Lanczos algorithm for computing eigenvectors in NIGLRAM. Since NIGLRAM is to compress the original image data set $\{\mathcal{I}_{t_i}\}_{i=1}^m \in \mathbb{R}^{q\times w}$ to the lower dimensional space $\mathbb{R}^{\tilde{l}_1\times\tilde{l}_2}$, we need to pre-specify $\tilde{l}_1 \le q, \tilde{l}_1 \in \mathbb{N}$, and $\tilde{l}_2 \le w, \tilde{l}_2 \in \mathbb{N}$. Lanczos+NIGLRAM aims to find two matrices $\tilde{L} \in \mathbb{R}^{q\times\tilde{l}_1}$ and $\tilde{R} \in \mathbb{R}^{w\times\tilde{l}_2}$ with orthogonal columns and m matrices $\tilde{M}_{t_i} \in \mathbb{N}^{\tilde{l}_1\times\tilde{l}_2}$ such that $\tilde{L}\tilde{M}_{t_i}\tilde{R}^\top$ approximate $\mathcal{I}_{t_i}$. Note that $\tilde{M}_{t_i} = \tilde{L}^\top\mathcal{I}_{t_i}\tilde{R}$ is the compressed version of $\mathcal{I}_{t_i}$ (see [31, Thm. 3.1] for proof). Therefore, we have Algorithm 3.

Time complexity of Lanczos+NIGLRAM Based on Remark 1 and Subsection II.B, time complexity of Lanczos+NIGLRAM is $O(\tilde{l}_1 q^2 + \tilde{l}_2 w^2 + mw^2\tilde{l}_1 + mwq\tilde{l}_1 + mq^2 w)$ which is in order of quadratic with respect to q and w, yet that of NIGLRAM is in order of cubic, i.e., $O(q^3 + w^3 + mw^2\tilde{l}_1 + mwq\tilde{l}_1 + mq^2 w)$.

Space complexity of Lanczos+NIGLRAM It is the same as those of GLRAM and NIGLRAM, i.e., $O(qw)$ (see also [31, Subsec. 3.3]).

4.2 The proposed method for predicting MMS behaviors

Our proposed method has three steps explained next.

Step 1: Compressing the data set

Before we compress the image data set $\{\mathcal{I}_{t_i}\}_{i=1}^m$, we require to pre-specify the dimension of the compressed domain $\mathbb{R}^{\tilde{l}_1\times\tilde{l}_2}$, namely we need to select $\tilde{l}_1 \in \mathbb{N}, \tilde{l}_1 \le q, \tilde{l}_2 \in \mathbb{N}, \tilde{l}_2 \le w$ a priori. Then we apply Algorithm 3 to give the matrices $\tilde{L}$ and $\tilde{R}$. Consequently, $\{\tilde{M}_{t_i}\}_{i=1}^m$ where $\tilde{M}_{t_i} = \tilde{L}^\top\mathcal{I}_{t_i}\tilde{R}$ are the compressed version of $\{\mathcal{I}_{t_i}\}_{i=1}^m$, respectively.

Step 2: Predicting future compressed data in the compressed domain

This step is similar to the method proposed in [27, Subsec. 4.1] with one difference. *The difference is that we utilize the Lanczos algorithm to find leading eigenvectors here instead of EVD (or SVD) used in Algorithm 1 in* [27].

Due to computational capacity of a processor, we are not able to store a compressed data matrix for any $m \in \mathbb{N}$. Thus, we utilize *moving window* compressed data matrix for the prediction problem and consider small P, say $1 \le P \le 5$. Let the processor be able to store a compressed data matrix of size $n := \tilde{l}_1\tilde{l}_2$ by $h, h \le m, h \in \mathbb{N}$. So our *time-varying* compressed data matrix at time t_i is defined as

$$X_{t_i} := [\mathcal{V}_{t_{i-h+1}}, \ldots, \mathcal{V}_{t_i}] \in \mathbb{R}^{n\times h}, \quad i \ge h, \tag{4}$$

where each compressed image $\tilde{M}_{t_i}$ is vectorized as $\mathcal{V}_{t_i} = [\tilde{M}_{t_i,1}^\top, \ldots, \tilde{M}_{t_i,\tilde{l}_2}^\top]^\top \in \mathbb{R}^{\tilde{l}_1\tilde{l}_2}$ where $\tilde{M}_{t_i,j} \in \mathbb{R}^{\tilde{l}_1}$ is the $j^{th}, j = 1, \ldots, \tilde{l}_2$, column of the matrix $\tilde{M}_{t_i}, i = 1, \ldots, m$.

Now we need to extract the most dominant basis (spatial) functions of the time-varying compressed data matrix X_{t_i} (defined in (4)) at time t_i. Hence, according to Lemma 1, we require to calculate r left singular vectors of X_{t_i} for a pre-selected r. To pre-select r, we select a threshold $\mathcal{T} \ge 0, \mathcal{T} \in \mathbb{R}$, on singular values of X_{t_i}, i.e., we only select those left singular vectors of X_{t_i} whose corresponding singular values are greater than $\mathcal{T}$. Since singular values of both matrices X_{t_i} and $X_{t_i}^\top$ are the same, computing the singular values can be efficiently performed by calculating square root of eigenvalues of $X_{t_i}^\top X_{t_i} \in \mathbb{R}^{h\times h}$. One way to select $\mathcal{T}$ is $\mathcal{T} := \eta\sigma_{t_i,1}$ for some pre-selected $\eta \in [0, 1)$ where $\sigma_{t_i,1}$ is the largest singular value of X_{t_i}.

Once r is calculated, then we can compute the r left singular vectors of the time-varying compressed data matrix X_{t_i} from Remark 2 in Subsection II.C. Precisely, the r left singular vectors of X_{t_i} provide the most dominant spatial basis functions of the time-series image data in the time interval $[t_{i-h+1}, \ldots, t_i]$ in a totally data-driven way. To calculate the r left singular vectors of X_{t_i}, we apply Algorithm 2.

The SVD of X_{t_i} is represented as $X_{t_i} = U_{t_i}\Sigma_{t_i}V_{t_i}^\top$. Let $U_{t_i}^r \in \mathbb{R}^{n\times r}, \Sigma_{t_i}^r \in \mathbb{R}^{r\times r}$, and $V_{t_i}^r \in \mathbb{R}^{h\times r}$ be left $r-$truncated of the matrices U_{t_i}, Σ_{t_i}, and V_{t_i}, respectively. Thus, $U_{t_i}^r\Sigma_{t_i}^r V_{t_i}^{r\top}$ approximates X_{t_i}. By making hypothesis that the solution of the underlying unknown PDE of the process can be written in the form of (2), we have that the j^{th} row of $\Sigma_{t_i}^r V_{t_i}^{r\top}, 1 \le j \le r$, is samples of coefficient of the j^{th} spatial basis (or j^{th} column of $U_{t_i}^r$) at time t_i.

It remains to predict the coefficients of the r most dominant spatial basis functions. One way is to utilize

polynomial regression where we choose polynomial functions up to degree $N_j, 1 \leq j \leq r$, (i.e., $1, t, t^2, \ldots, t^{N_j}$) for the j^{th} column of the matrix $U_{t_i}^r$. Let $Y_{t_i}^j \in \mathbb{R}^{1\times h}$ be the j^{th} row of $\Sigma_{t_i}^r {V_{t_i}^r}^\top, 1 \leq j \leq r$. Then the polynomial regression problem for each $Y_{t_i}^j$ is given by solving the following linear algebraic equations

$$ {Y_{t_i}^j}^\top = \begin{pmatrix} 1 & t_{i-h+1} & t_{i-h+1}^2 & \ldots & t_{i-h+1}^{N_j} \\ 1 & t_{i-h+2} & t_{i-h+2}^2 & \ldots & t_{i-h+2}^{N_j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & t_i & t_i^2 & \ldots & t_i^{N_j} \end{pmatrix} \begin{pmatrix} a_1^j \\ \vdots \\ a_{N_j+1}^j \end{pmatrix} \tag{5} $$

for $1 \leq j \leq r$ where each $a_l^j, l = 1, \ldots, N_j + 1$, is the unknown coefficient of the polynomial t^{l-1} at time t_i. Once (5) is solved, we can predict the coefficient of j^{th} column of $U_{t_i}^r$ as

$$ b_{t_{i+z}}^j := \sum_{l=1}^{N_j+1} {a_l^j}^* t_{t_{i+z}}^{l-1}, \quad 1 \leq z \leq P, 1 \leq j \leq r, \tag{6} $$

where ${a_l^j}^*, l = 1, \ldots, N_j + 1$, are solutions of (5). Hence, the predicted compressed images at time $t_{i+z}, 1 \leq z \leq P$, are calculated as

$$ \tilde{\mathcal{V}}_{t_{i+z}} := \sum_{j=1}^{r} {U_{t_i}^r}^j b_{t+z}^j \in \mathbb{R}^{qw}, \quad 1 \leq z \leq P \tag{7} $$

where ${U_{t_i}^r}^j$ is the j^{th} column of $U_{t_i}^r$, and $\tilde{\mathcal{V}}_{t_{i+z}}$ need to be put in matrices $\mathcal{M}_{t_{i+z}} \in \mathbb{R}^{\tilde{l}_1 \times \tilde{l}_2}, 1 \leq z \leq P$ (we use notation $\mathcal{M}$ for predicted compressed images $\tilde{M}_{t_i+z}$).

Require: $\{\mathcal{I}_{t_j}\}_{j=i-h+1}^{i}, i \geq h, \tilde{l}_1, \tilde{l}_2, h, \eta$, and P
Require: (Initialization) $\theta = 0$
Ensure: Predicted images $\tilde{\mathcal{I}}_{t_{i+z}} \in \mathbb{R}^{q\times w}, 1 \leq z \leq P$
$m \leftarrow h$
Apply Algorithm 3
for $j = i - h + 1, \ldots, i$ **do**
  $\tilde{M}_{t_j} \leftarrow \tilde{L}^\top \mathcal{I}_{t_j} \tilde{R}$
end for
for $j = i - h + 1, \ldots, i$ **do**
  $\mathcal{V}_{t_j} \leftarrow [\tilde{M}_{t_j,1}^\top, \ldots, \tilde{M}_{t_j,\tilde{l}_2}^\top]^\top$ where $\tilde{M}_{t_i,l} \in \mathbb{R}^{\tilde{l}_1}, \{l\}_{l=1}^{\tilde{l}_2}$
end for
Compute the matrix X_{t_i} defined in (4)
Calculate eigenvalues of $X_{t_i}^\top X_{t_i}$ in decreasing order
Calculate singular values of X_{t_i}, i.e.,$\sigma_{t_i,v}, 1 \leq v \leq h$
$\mathcal{T} \leftarrow \eta\sigma_{t_i,1}$
for $v = 1, \ldots, h$ **do**
  if $\sigma_{t_i,v} > \mathcal{T}$ **then**
    $\theta \leftarrow \theta + 1$
  end if
end for
$k \leftarrow r$
Select unit vector $q_1 \in \mathbb{R}^h$
$M \leftarrow X_{t_i}^\top X_{t_i}$
Apply Algorithm 2
$V_{t_i}^r \leftarrow [q_1, q_2, \ldots, q_r] \in \mathbb{R}^{h\times r}$
Put $r-$leading singular values of X_{t_i} in diagonal entries of $\Sigma_{t_i}^r$
$U_{t_i}^r \leftarrow X_{t_i} V_{t_i}^r {\Sigma_{t_i}^r}^{-1}$
for $j = 1, \ldots, r$ **do**
  Put each row of $\Sigma_{t_i}^r {V_{t_i}^r}^\top$ in $Y_{t_i}^j \in \mathbb{R}^{1\times h}$
  Select $N_j \in \mathbb{N}$
end for
for $j = 1, \ldots, r$ **do**
  Solve (5)
end for
for $j = 1, \ldots, r$ and $z = 1, \ldots, P$, **do**
  Calculate $b_{t_{i+z}}^j$ in (6)
end for
for $z = 1, \ldots, P$, **do**
  Calculate predicted compressed image $\tilde{\mathcal{V}}_{t_{i+z}}$ in (7).
  Put the vector $\tilde{\mathcal{V}}_{t_{i+z}} \in \mathbb{R}^{\tilde{l}_1\tilde{l}_2}$ in matrix $\mathcal{M}_{t_{i+z}} \in \mathbb{R}^{\tilde{l}_1\times\tilde{l}_2}$
end for
for $z = 1, \ldots, P$ **do**
  $\tilde{\mathcal{I}}_{t_{i+z}} \leftarrow \tilde{L}\mathcal{M}_{t_{i+z}}\tilde{R}^\top$
end for

Algorithm 4 Proposed algorithm for predicting MMS behaviors

Step 3: Recovering the predicted future actual images from corresponding predicted compressed images

So far, we have been able to predict the future compressed images, i.e., $\{\mathcal{M}_{t_{i+z}}\}_{z=1}^P$ in *Step 2*. $\tilde{\mathcal{I}}_{t_{i+z}} := \tilde{L}\mathcal{M}_{t_{i+z}}\tilde{R}^\top, 1 \leq z \leq P$ approximate $\mathcal{I}_{t_{i+z}}$. The pseudo-code of the above method is given in Algorithm 4.

The schematic of our proposed method is similar to that of [27] (shown in Fig. 2 in [27]) by substituting Algorithm 4 for Algorithm 1 in [27, Fig. 2].

Space complexity of Algorithm 4: Space complexity of Algorithm 4 is $O(max\{\tilde{l}_1\tilde{l}_2 h, qw\})$. The space complexity of Algorithm 1 in [27] is $O(qwh)$ which is larger than that of Algorithm 4.

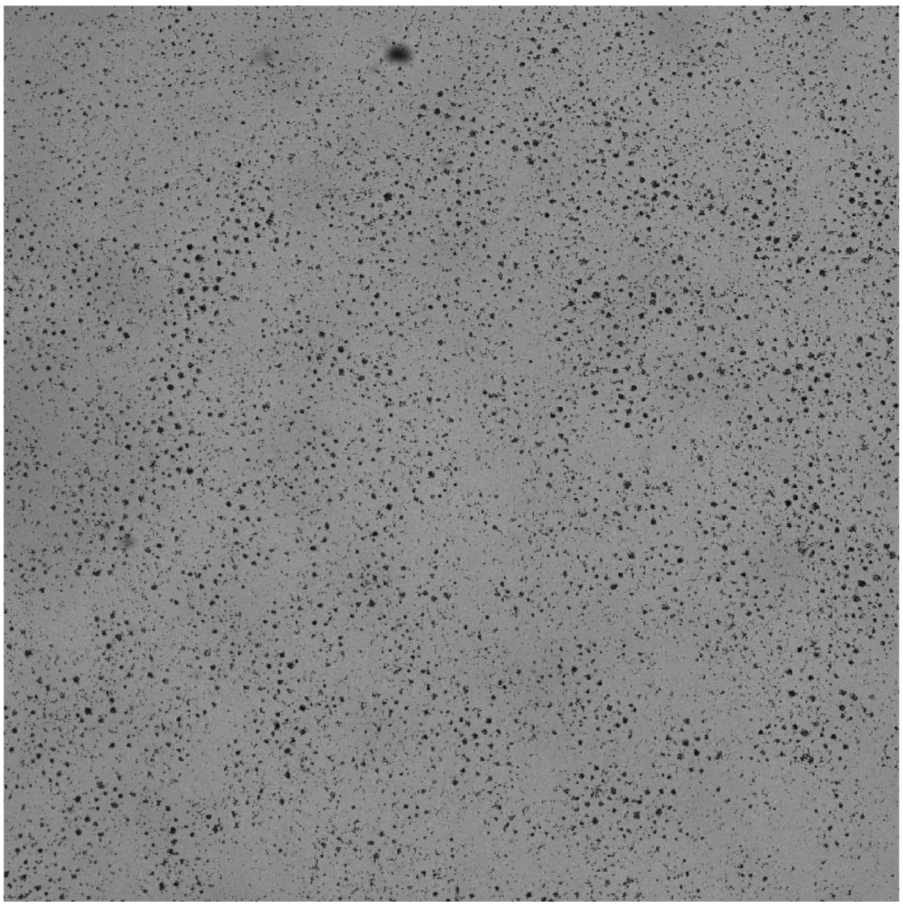


Fig. 2 Ground truth of first prediction image for Trial 1 in Example 1 (i.e., 301^{st} image)

Time complexity of algorithm 4 From Subsection IV.A, time complexity of Algorithm 3 is $O(\tilde{l}_1 q^2 + \tilde{l}_2 w^2 + hw^2\tilde{l}_1 + hwq\tilde{l}_1 + hq^2 w)$. Since $\tilde{L}^\top(\mathcal{I}_{t_j}\tilde{R})$ has time complexity $O(\tilde{l}_1 q\tilde{l}_2 + qw\tilde{l}_2)$, the first For loop's time complexity is $O(h\tilde{l}_1 q\tilde{l}_2 + hqw\tilde{l}_2)$. The second For loop's time complexity is h. Time complexity of calculating $X_{t_i}^\top X_{t_i}$ is $O(h^2\tilde{l}_1\tilde{l}_2)$. Since we need to compute all eigenvalues of $X_{t_i}^\top X_{t_i}$, the time complexity is $O(h^3)$. Calculating the singular value of X_{t_i} (which are the square root of eigenvalues of $X_{t_i}^\top X_{t_i}$) as well as the third For loop have time complexity of $O(h)$. Time complexity of Algorithm 2 is $O(rh^2)$. Time complexity of $X_{t_i}(V_{t_i}^r \Sigma_{t_i}^{r^{-1}})$ is $O(hr^2 + \tilde{l}_1\tilde{l}_2 hr)$. The fourth loop has time complexity r. Assuming $N_j \leq h, j = 1, \ldots, r$, (which is realistic assumption, see the examples), time complexity of the fifth For loop is $O(r \quad max\{N_j + 1\}^2(h - \frac{max\{N_j+1\}}{3}))$ (see Table C.2 in [45] for QR factorization). Time complexity of the sixth For loop is $O(max\{N_j + 1\}rP)$. Time complexity of seventh For loop is $O(\tilde{l}_1\tilde{l}_2 rP)$. Time complexity of the last For loop (where $\tilde{L}(\mathcal{M}_{t_{i+z}}\tilde{R}^\top)$) is $O(P(qw\tilde{l}_1 + \tilde{l}_1\tilde{l}_2 w))$. Therefore, the time complexity of Algorithm 4 is $O(q^2(\widetilde{l}_1 + hw) + w^2(\widetilde{l}_2 + h\widetilde{l}_1) + qw(h\widetilde{l}_1 + h\widetilde{l}_2 + P\widetilde{l}_1) + \widetilde{l}_1\widetilde{l}_2\,(h^2 + qh + hr + rP) + h^3 + r\,max\{N_j + 1\}^2\,(h - \frac{max\{N_j+1\}}{3}) + max\{N_j + 1\}rP)$.

Time complexity of Algorithm 1 in [27] is $O(h^3 + qwh^2 + qwhr + max\{N_j + 1\}^2(h - \frac{max\{N_j+1\}}{3})\,r + max\{N_j + 1\}rP + qwrP)$ which seems to be smaller than that of Algorithm 4 where quadratic terms of q and w exist. This is the price that we need to pay for dimensionality reduction in *Step 1*.

5 Experimental and simulation results

In this section, we apply Algorithm 4 to different kinds of untethered synthetic MMS in fluid to show the effectiveness of the method to predict their swarm behaviors, specially emergent behaviors, with less prediction error than the method in [27], and in some cases less training error than that of [27] (see Examples 2–3).

Example 1 *(Ellipsoidal MMs)* In this example, we use the same data utilized in Example 1 in [27]. A droplet of 10 μL ($\mu = 10^{-6}$, and L represents liter) of an ellipsoidal iron

oxide Fe_2O_3 (or *hematite*), where the particles were suspended in water, was sandwiched between two glass slides separated by a spacer (height=120 micro meter) at the temperature of 71 degree Fahrenheit (or 21.66 degree Celsius). The ellipsoidal hematite particles have long axis and short axis lengths of 1.4 and 1 micro meter, respectively. The particles sediment to the surface of experimental chamber as they are density mismatched ($\rho_{hematite}$ = 5.15 $grams/centimeter^3$ =5.15 $\times 10^3 \frac{kg}{m^3}$) and the experimental observations are carried out at the sedimentation plane. A circularly polarized external rotating magnetic field is applied in the $x-y$ plane with magnitude B_0=10 milli-Tesla and frequency $f = 20$ Hertz, i.e., $\mathrm{B} = B_0(cos(40\pi t)\overrightarrow{x} + sin(40\pi t)\overrightarrow{y})$ where t represents time. The hematite colloids experience a torque in the rotating magnetic field causing them to spin on the surface. The interactions between the spinning colloids are mediated by the fluid and the hydrodynamic interactions lead to emergent behavior, including the formation of large circulating clusters.

Dynamics of the active system are captured at 20 frame per second using a CMOS camera, i.e., $\Delta t_i = 0.05, i = 1, \ldots, 300,$ second. The size of images are $q = 1608$ by $w = 1608$ pixels, and the intensity of each pixel is between 0 and 1. We use $h = 300$ images for training and $P = 5$. We do not show the $1^{st}, 100^{th}$, and 200^{th} training images since they appear in [27, Figs. 3-4]. Note that $\|\mathcal{I}_{t_i}\|_F = \|vectorized(\mathcal{I}_{t_i})\|_2$. Therefore, *error*(*z*) here and in [27] are the same.

We consider two same trails as in [27], i.e., Trial 1 and Trial 2, where the results are simulated by MATLAB R2023b. Training errors (i.e., $\mathcal{E} := \frac{1}{300}\sum_{i=1}^{300} \|\mathcal{I}_{t_i} - \tilde{L}\tilde{M}_{t_i}\tilde{R}^\top\|_F^2$) are shown in Tables 1 and 2 for Trails 1 and 2, respectively. As seen in Tables 1 and 2, the training errors of the method in [27] are smaller than those of here. We use average (per pixel) Frobenius norm for calculating the error between ground truth image and the predicted one, i.e., $error(z) := \frac{\|\mathcal{I}_{t_{300+z}} - \tilde{\mathcal{I}}_{t_{300+z}}\|_F}{1608\times 1608}, z = 1, \ldots, 5.$ $errors(z), z = 1, \ldots, P,$ for Trails 1 and 2 are presented in Tables 3 and 4, respectively. We provide mean and standard deviation of the error $\{e_i := \|\mathcal{I}_{t_i} - \tilde{\mathcal{I}}_{t_i}\|_F\}_{i=301}^{305}$ in Table 5. We also provide prediction error $\mathcal{P} := \frac{1}{5}\sum_{i=301}^{305} \|\mathcal{I}_{t_i} - \tilde{L}\tilde{\mathcal{I}}_{t_i}\tilde{R}^\top\|_F^2$ in Table 6. As seen from Tables 3-6, Algorithm 4 provides less prediction error than the method in [27] while space complexity is significantly reduced. We show the predicted images in Trails 1 and 2 where $z = 1$ in Figs. 2-4 and 5-7, respectively. One reason to justify the results in Tables 1-6 is that vectorization of

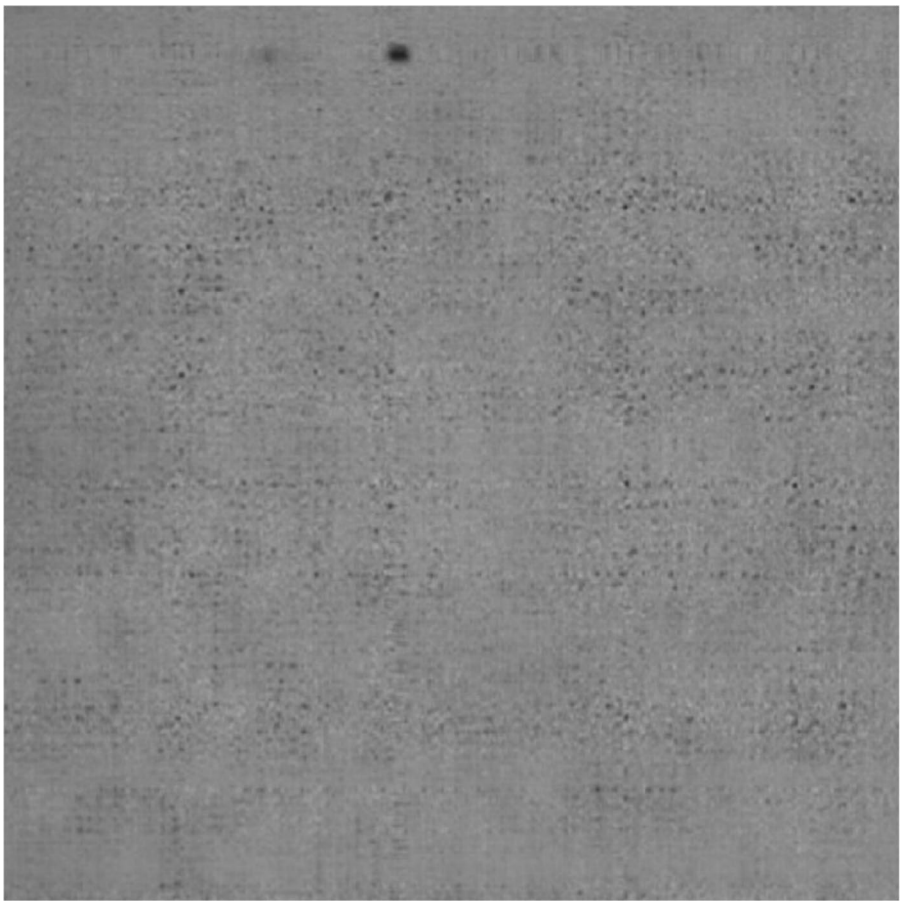


Fig. 3 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 50$ for Trial 1 in Example 1. Ground truth image is shown in Fig. 2

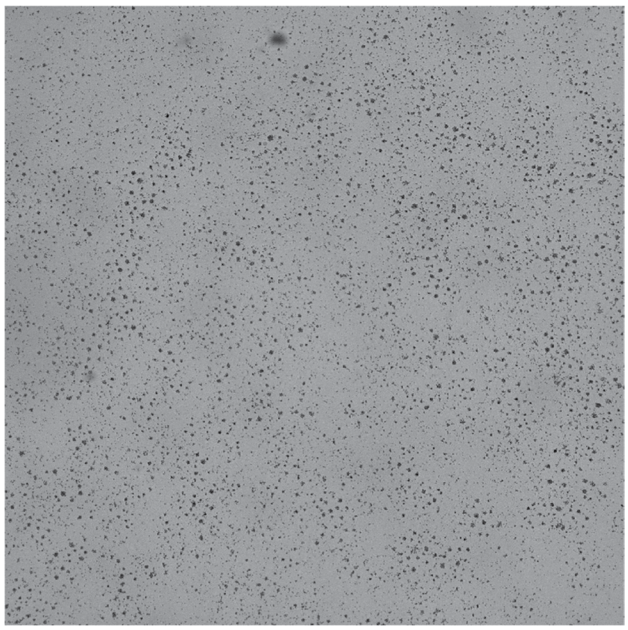


Fig. 4 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 100$ for Trial 1 in Example 1 (see Table 3 for details). Ground truth image is shown in Fig. 2

Table 1 Training Error of Trial 1 in Example 1

$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 50$	$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 100$	[27] (see [31, Thm. 2.1])
1.9271×10^4	1.7343×10^4	2.7687×10^3

Table 2 Training Error of Trial 2 in Example 1

$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 50$	$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 100$	[27] (see [31, Thm. 2.1])
2.3502×10^3	1.6413×10^3	1.3452×10^3

Table 3 error(z) of Trial 1 in Example 1 for $\eta = \frac{1}{200}$ and $N_j = 10, j = 1, \ldots, r$

z	$r = 4, \tilde{l}_1 = \tilde{l}_2 = 50$	$r = 7, \tilde{l}_1 = \tilde{l}_2 = 100$	112 in [27]
1	0.052×10^{-3}	0.049×10^{-3}	0.063×10^{-3}
2	0.057×10^{-3}	0.055×10^{-3}	0.083×10^{-3}
3	0.069×10^{-3}	0.068×10^{-3}	0.109×10^{-3}
4	0.065×10^{-3}	0.065×10^{-3}	0.129×10^{-3}
5	0.095×10^{-3}	0.096×10^{-3}	0.168×10^{-3}

Table 4 error(z) of Trial 2 in Example 1 for $\eta = \frac{1}{100}$ and $N_j = 8, j = 1, \ldots, r$

z	$r = 7, \tilde{l}_1 = \tilde{l}_2 = 50$	$r = 10, \tilde{l}_1 = \tilde{l}_2 = 100$	23 in [27]
1	0.2100×10^{-4}	0.2162×10^{-4}	0.2448×10^{-4}
2	0.2154×10^{-4}	0.2352×10^{-4}	0.2858×10^{-4}
3	0.2171×10^{-4}	0.2492×10^{-4}	0.3222×10^{-4}
4	0.2237×10^{-4}	0.2724×10^{-4}	0.3680×10^{-4}
5	0.2317×10^{-4}	0.2988×10^{-4}	0.4164×10^{-4}

Table 5 Mean and Standard Deviation of the Error $\{e_i\}_{i=301}^{305}$ in Example 1

$\{e_i\}_{i=301}^{305}$	$\tilde{l}_1 = \tilde{l}_2 = 50$	$\tilde{l}_1 = \tilde{l}_2 = 100$	[27]
Trial 1	176.09 ± 43.99	173.18 ± 46.25	287.08 ± 105.20
Trial 2	56.77 ± 2.16	65.76 ± 8.33	84.66 ± 17.41

Table 6 Prediction Error $\mathcal{P}$ in Example 1

$\mathcal{P}$	$\tilde{l}_1 = \tilde{l}_2 = 50$	$\tilde{l}_1 = \tilde{l}_2 = 100$	[27]
Trial 1	3.2559×10^4	3.1704×10^4	9.1271×10^4
Trial 2	3.2267×10^3	4.3811×10^3	7.4110×10^3

matrices makes over-fitting (see Section 1) and provides more error for prediction.

Example 2 *(Cubic MMs)* In order to demonstrate the generality of our modelling to different geometries of MMs, we fabricate cuboidal iron oxide colloidal particles (with length of one edge of the cube to be 1 micro meter) using a similar synthesis process as ellipsoidal particles [46]. An identical experimental set-up as Example 1 is used and a circularly polarized rotating magnetic field is applied in the x-y plane with magnitude B_0 = 10 milli-Tesla and frequency f = 20 Hertz. Similar to ellipsoidal particles, cuboidal particles form circulating clusters over time, but the formed clusters are smaller in size compared to ellipsoidal particles due to the difference in hydrodynamic interactions. Dynamics of the active system are captured at 10 frame per second using a CMOS camera, i.e., $\Delta t_i = 0.1, i = 1, \ldots, 300,$ second. The size of images are $q = 1608$ by $w = 1608$ pixels, and the intensity of each pixel is normalized to be between 0 and 1.

Similar to Example 1, we use $h = 300$ images for training and $P = 5$. We show the $1^{st}, 150^{th}$, and 301^{th} images in Figs. 8-10 (note that due to slow evolution, we do not provide 300^{th} image since it is very similar to the 300^{th} image). Training error $\mathcal{E}$, *error*(*z*), mean and standard deviation of the Error $\{e_i\}_{i=301}^{305}$, and prediction error $\mathcal{P}$ are shown in Tables 7, 8, 9, and 10, respectively. As seen from Tables 8-10, Algorithm 4 provides less prediction error than the method in [27] while space complexity is significantly reduced. For the case $\tilde{l}_1 = \tilde{l}_2 = 500$, the results show that both training and prediction errors are less than those of the results in [27]. We show the predicted images where $z = 1$ in Figs. 11-13.

Example 3 *(Rod-shaped MMs)* As we have seen from Example 1 and 2, the collective dynamics of rotating magnetic colloidal systems is dependent on the geometry of the constituent particles. Using a variation of the synthesis process used to fabricate ellipsoidal and cuboidal particles, we

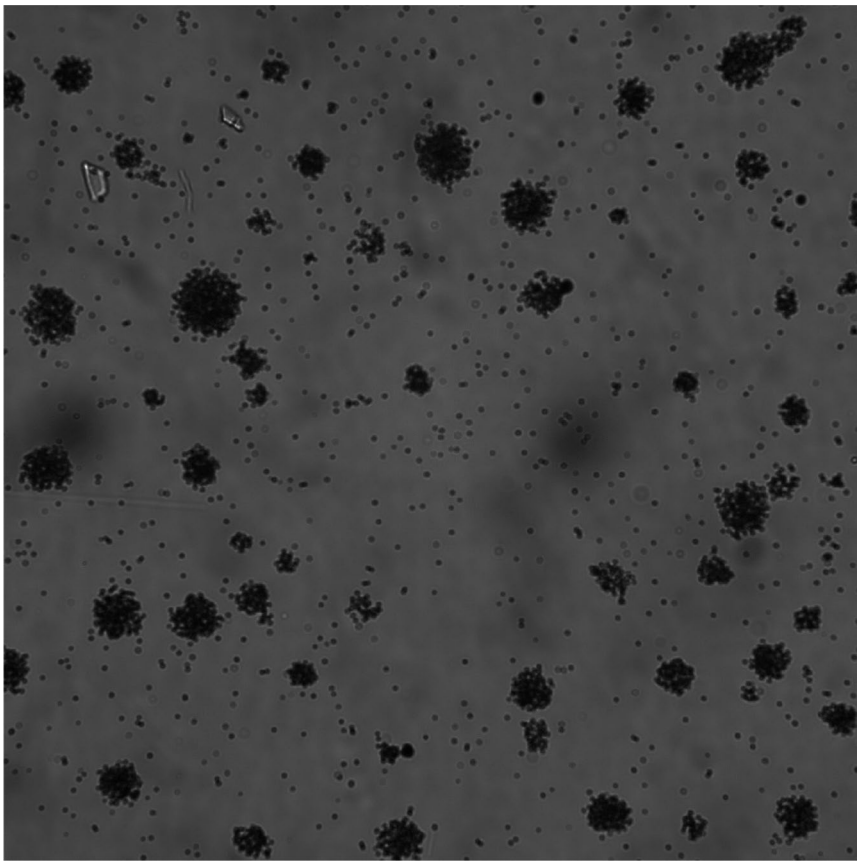


Fig. 5 Ground truth of first prediction image for Trial 2 in Example 1 (i.e., 301^{st} image)

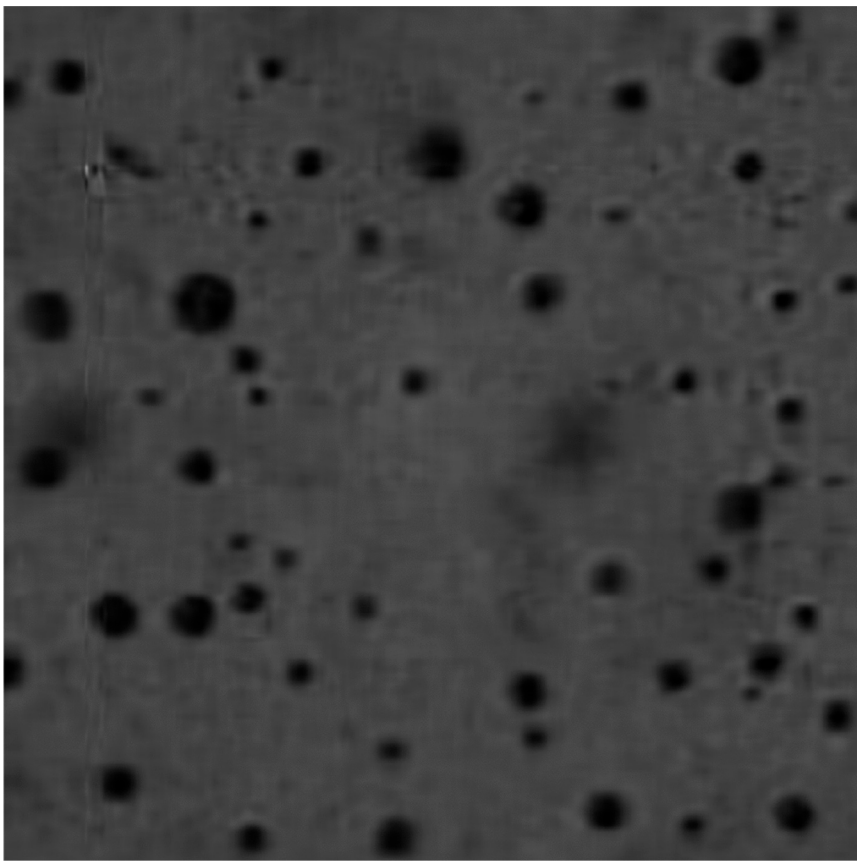


Fig. 6 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 50$ for Trial 2 in Example 1. Ground truth image is shown in Fig. 5

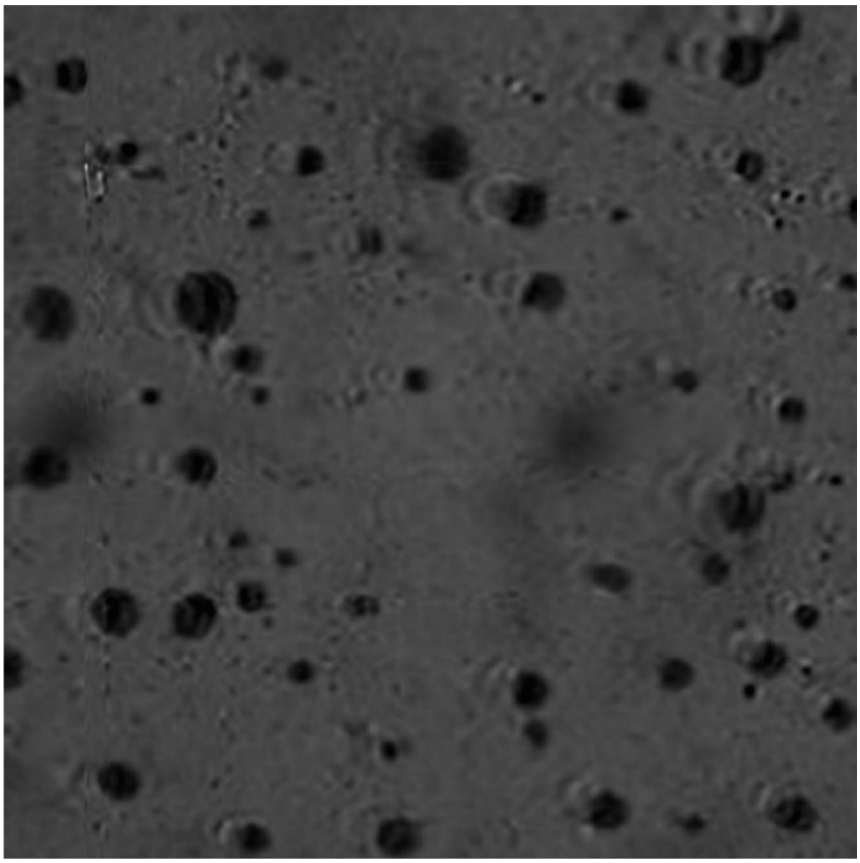


Fig. 7 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 100$ for Trial 2 in Example 1 (see Table 4 for details). Ground truth image is shown in Fig. 5

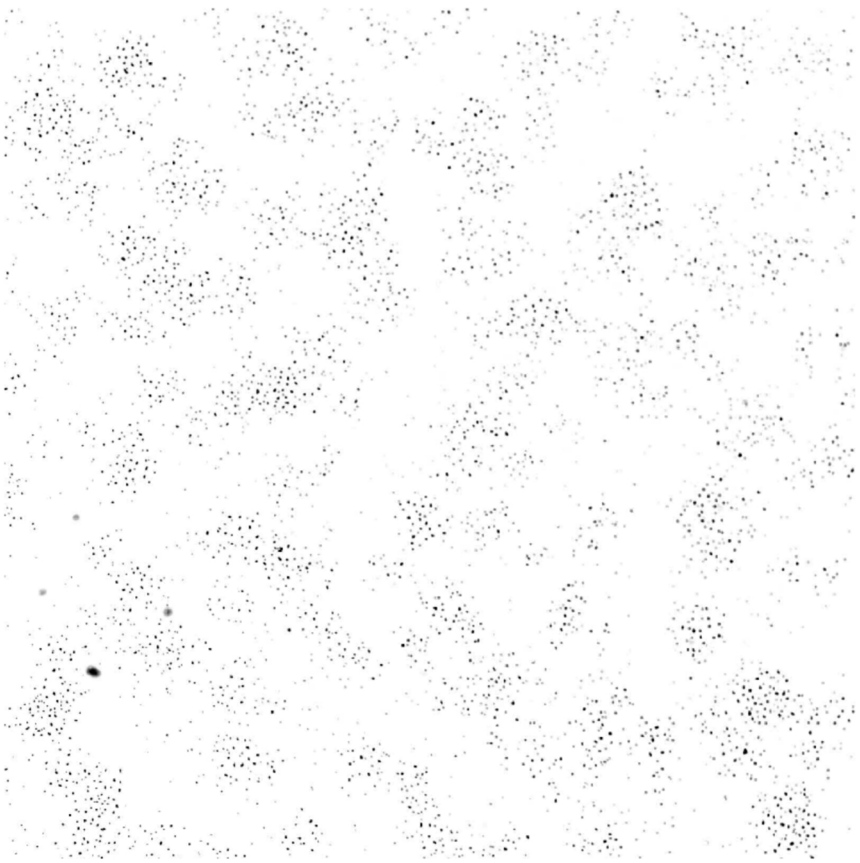


Fig. 8 Ground truth of first training image in Example 2

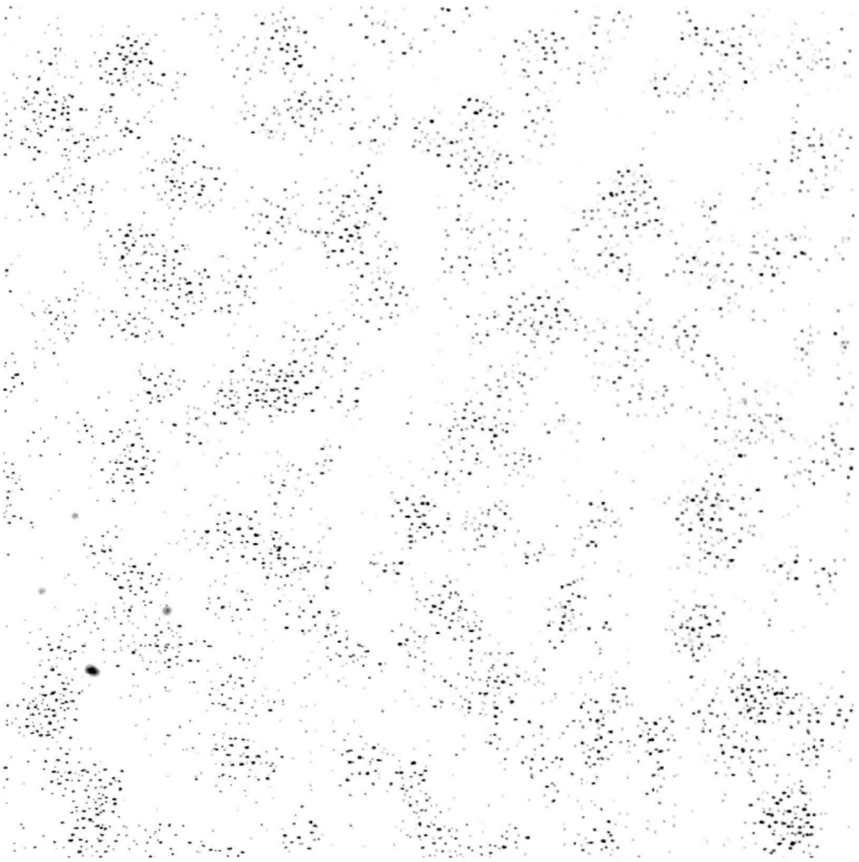


Fig. 9 Ground truth of 150^{th} training image in Example 2

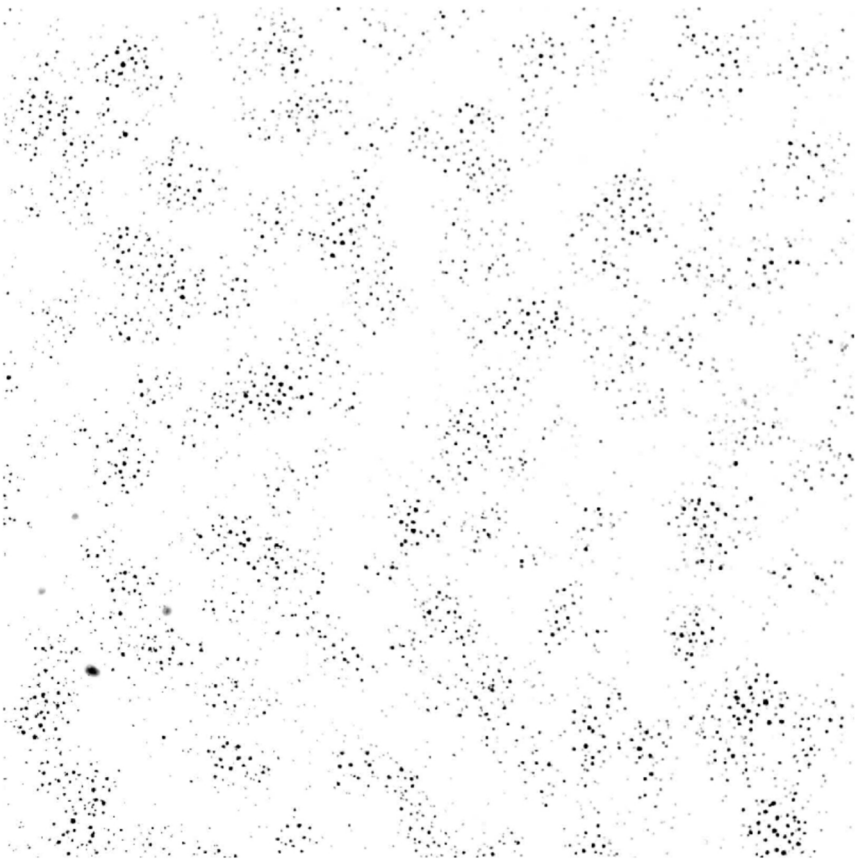


Fig. 10 Ground truth of 301^{st} image in Example 2

Table 7 Training Error in Example 2

$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 300$	$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 500$	[27] ([31, Thm. 2.1])
5.0118×10^3	1.3088×10^3	1.4695×10^3

Table 8 error(z) in Example 2 for $\eta = \frac{1}{150}$ and $N_j = 5, j = 1, \ldots, r$

z	$r = 9, \tilde{l}_1 = \tilde{l}_2 = 300$	$r = 11, \tilde{l}_1 = \tilde{l}_2 = 500$	$r = 13$ in [27]
1	0.5315×10^{-4}	0.0517×10^{-3}	0.0518×10^{-3}
2	0.5692×10^{-4}	0.0570×10^{-3}	0.0575×10^{-3}
3	0.6101×10^{-4}	0.0625×10^{-3}	0.0634×10^{-3}
4	0.6533×10^{-4}	0.0683×10^{-3}	0.0694×10^{-3}
5	0.7000×10^{-4}	0.0744×10^{-3}	0.0759×10^{-3}

Table 9 Mean and Standard Deviation of the Error $\{e_i\}_{i=301}^{305}$ in Example 2

	$\tilde{l}_1 = \tilde{l}_2 = 300$	$\tilde{l}_1 = \tilde{l}_2 = 500$	[27]
$\{e_i\}_{i=301}^{305}$	158.45 ± 17.23	162.33 ± 23.19	164.40 ± 24.61

Table 10 Prediction Error $\mathcal{P}$ in Example 2

	$\tilde{l}_1 = \tilde{l}_2 = 300$	$\tilde{l}_1 = \tilde{l}_2 = 500$	[27]
$\mathcal{P}$	2.5345×10^4	2.6782×10^4	2.7515×10^4

fabricate rod shaped particles with long axis and short axis lengths of 1 micro meter and 0.25 micro meter, respectively. Again, we use an identical experimental set-up to Examples 1 and 2, and apply the same rotating magnetic field in the x-y plane with magnitude B_0 = 10 milli-Tesla and frequency f = 20 Hertz. However, unlike the case of ellipsoidal (Example 1) and cuboidal (Example 2) particles, we do not observe any cluster formation even after t = 300s. Closer observation reveals that while rotating rod shaped particles interact with each other hydrodynamically the two-particle coupling is much weaker than in the case of ellipsoidal and cube-shaped particles leading to sparse system without circulating clusters.

Dynamics of the active system are captured at 30 frame per second using a CMOS camera, i.e., $\Delta t_i = \frac{1}{30}, i = 1, \ldots, 300$, second. The size of images are $q = 1608$ by $w = 1608$ pixels, and the intensity of each pixel is between 0 and 1. Similar to Examples 1 and 2, we use $h = 300$ images for training and $P = 5$. We show the $1^{st}, 150^{th}$, and 301^{th} images in Figs. 14-16 (note that due to slow evolution, we do not provide 300^{th} image since it is very similar to the 301th 300^{th} image).

Similar to Examples 1 and 2, training error $\mathcal{E}$, *error*(*z*), mean and standard deviation of the Error $\{e_i\}_{i=301}^{305}$, and prediction error $\mathcal{P}$ are shown in Tables 11, 12, 13, and 14, respectively. As seen from Tables 11-14, Algorithm 4 provides less prediction error than the method in [27] while

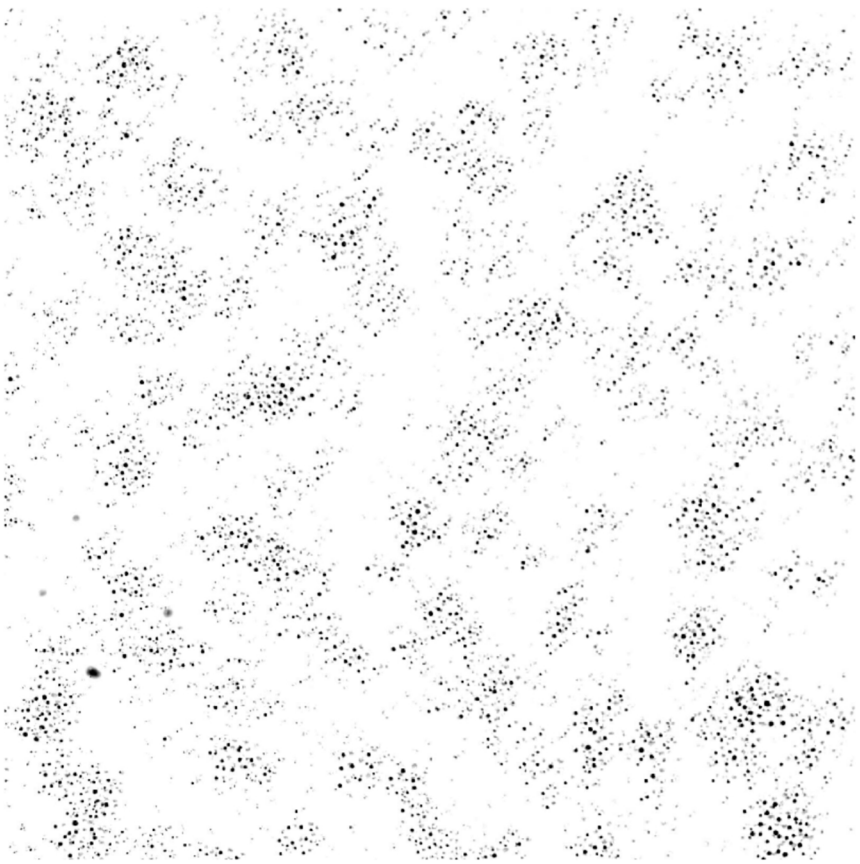


Fig. 11 Predicted image at $z = 1$ by the proposed method in [27] in Example 2. Note that the ground truth image is shown in Fig. 10

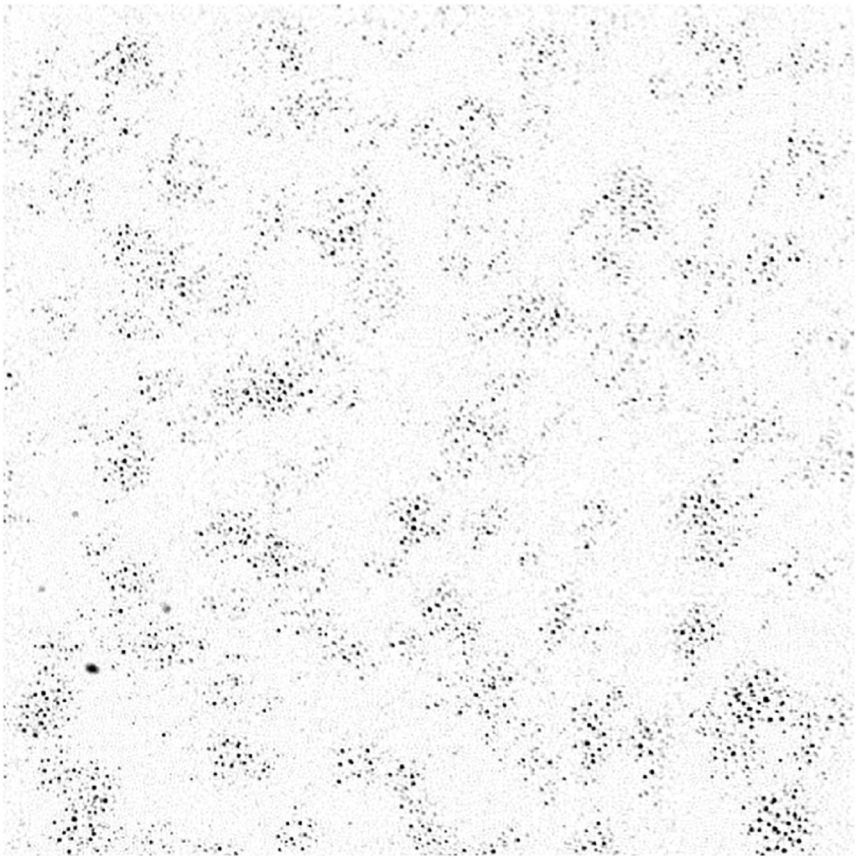


Fig. 12 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 300$. in Example 2. Note that the ground truth image is shown in Fig. 10

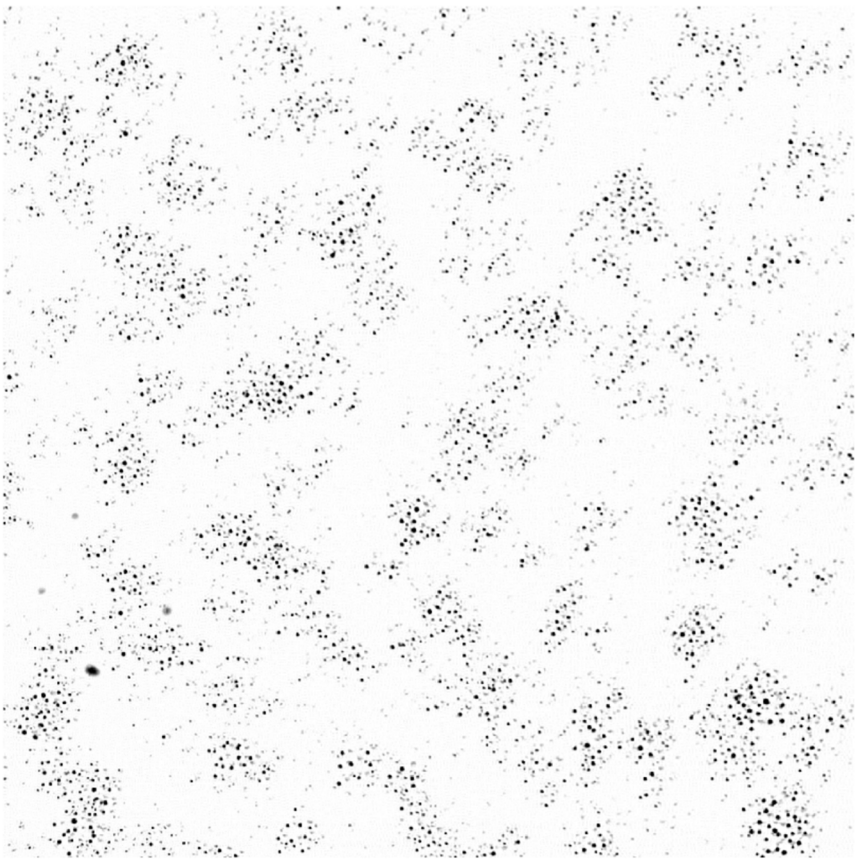


Fig. 13 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 500$ in Example 2. Note that the ground truth image is shown in Fig. 10

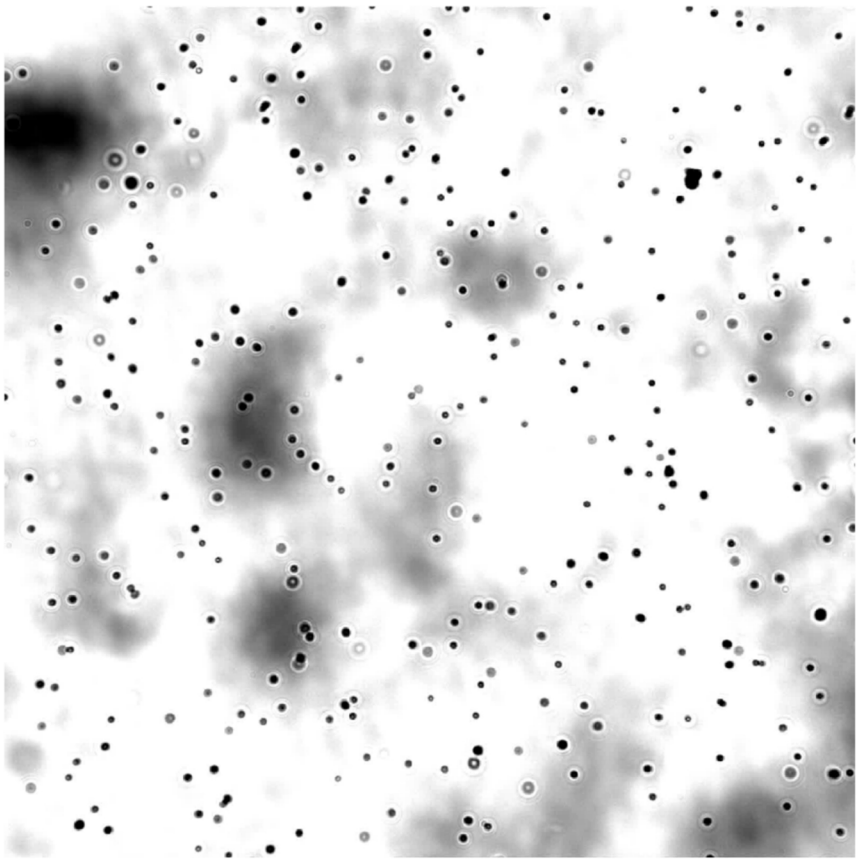


Fig. 14 Ground truth of first training image in Example 3

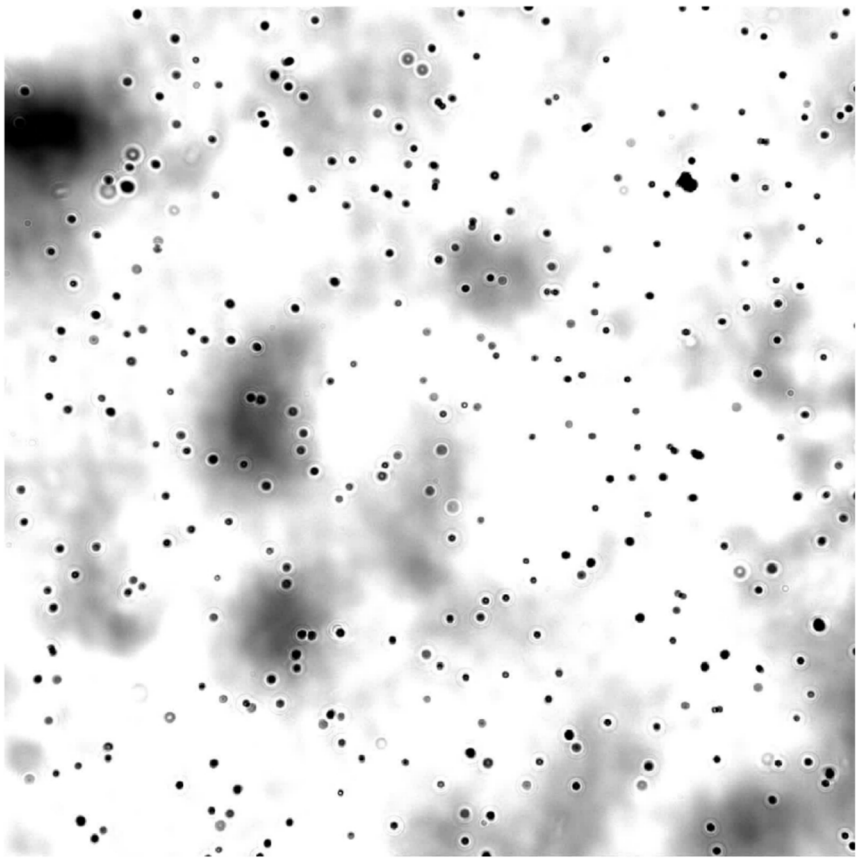


Fig. 15 Ground truth of 150^{th} training image in Example 3

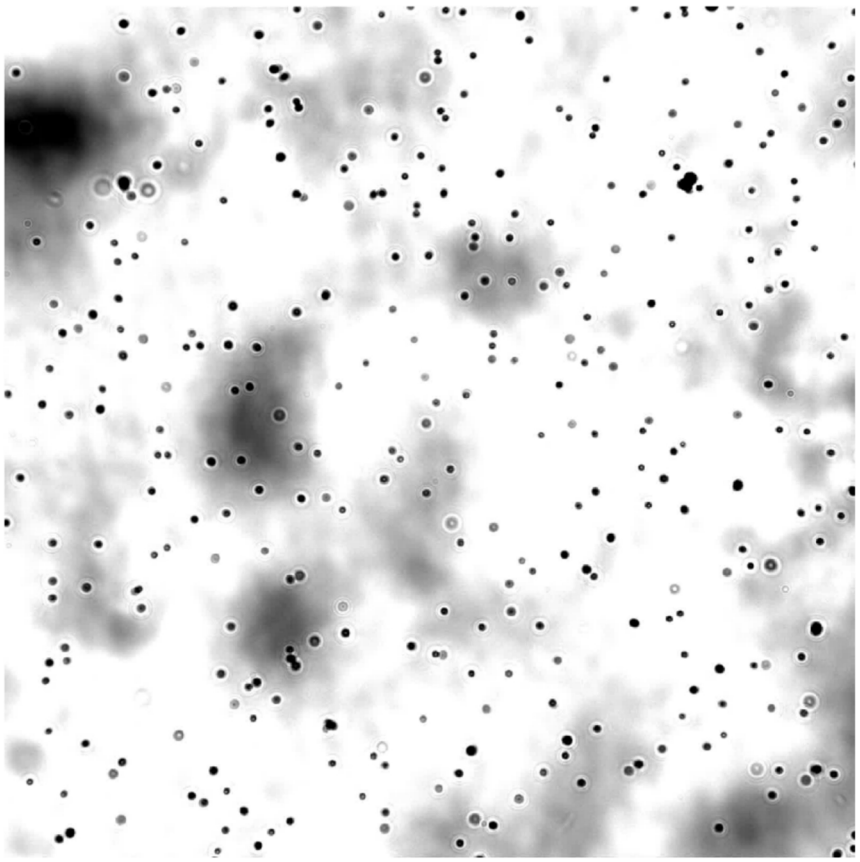


Fig. 16 Ground truth of 301^{st} image in Example 3

Table 11 Training Error in Example 3

$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 200$	$\mathcal{E}$ for $\tilde{l}_1 = \tilde{l}_2 = 400$	[27] ([31, Thm. 2.1])
2396.2	476.1	1659.0

Table 12 error(z) in Example 3 for $\eta = \frac{1}{300}$ and $N_j = 2, j = 1, \ldots, r$

z	$r = 59, \tilde{l}_1 = \tilde{l}_2 = 200$	$r = 69, \tilde{l}_1 = \tilde{l}_2 = 400$	$r = 73$ in [27]
1	0.5856×10^{-4}	0.5850×10^{-4}	0.5850×10^{-4}
2	0.5993×10^{-4}	0.6001×10^{-4}	0.6002×10^{-4}
3	0.6147×10^{-4}	0.6165×10^{-4}	0.6166×10^{-4}
4	0.6347×10^{-4}	0.6372×10^{-4}	0.6374×10^{-4}
5	0.6438×10^{-4}	0.6466×10^{-4}	0.6468×10^{-4}

Table 13 Mean and Standard Deviation of the Error $\{e_i\}_{i=301}^{305}$ in Example 3

	$\tilde{l}_1 = \tilde{l}_2 = 200$	$\tilde{l}_1 = \tilde{l}_2 = 400$	[27]
$\{e_i\}_{i=301}^{305}$	159.17 ± 6.22	159.55 ± 6.58	159.58 ± 6.60

Table 14 Prediction Error $\mathcal{P}$ in Example 3

	$\tilde{l}_1 = \tilde{l}_2 = 200$	$\tilde{l}_1 = \tilde{l}_2 = 400$	[27]
$\mathcal{P}$	2.5369×10^4	2.5492×10^4	2.5502×10^4

space complexity is significantly reduced. For the case $\tilde{l}_1 = \tilde{l}_2 = 400$, the results show that both training and prediction errors are less than those of the results in [27]. We show the predicted images where $z = 1$ in Figs. 17-19.

6 Conclusions

In this paper, we considered the problem of predicting magnetic microrobotic swarm's behaviors, including emergent behaviors of untethered magnetic microrobots, from time-series image data. We proposed Lanczos+NIGLRAM algorithm for dimensionality reduction with lower time complexity than existing methods in the literature. Then we used the Lanczos+NIGLRAM algorithm to compress the image data set. Next we applied POD on compressed image data to extract the most dominant spatial basis functions and predict the future compressed image data. The proposed approach is an interpretable ML method which is able to cross the reality gap. It is also reduced-order and totally data-driven without knowing any knowledge about the physics and/or interacting dynamics of MMs. Finally, several experimental results were provided in order to show the effectiveness of the result and its advantages.

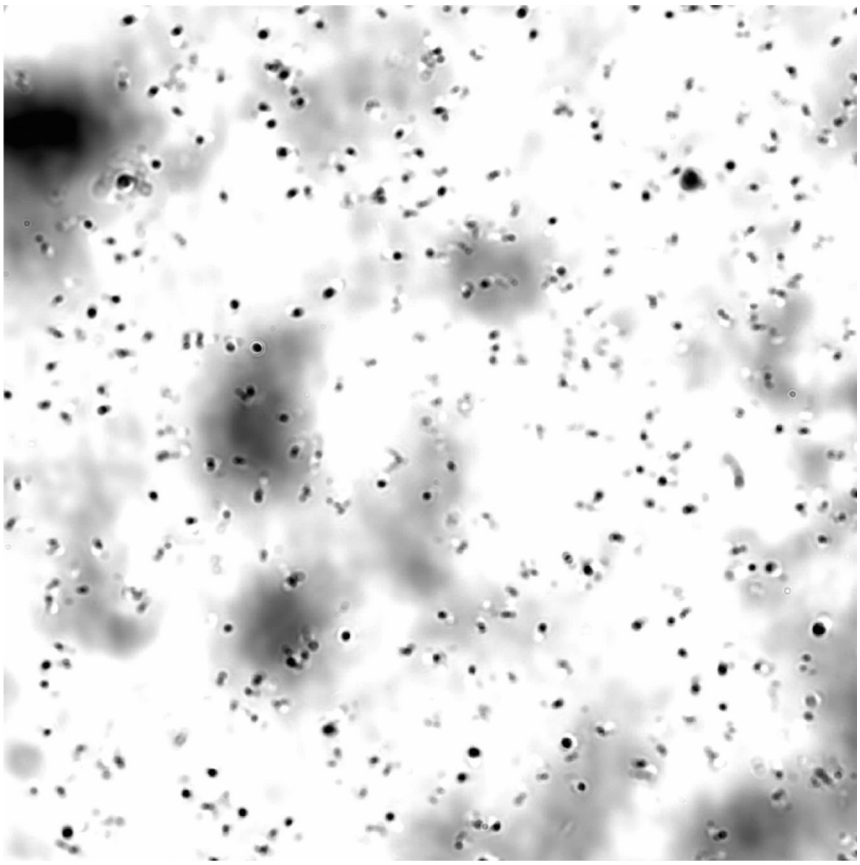


Fig. 17 Predicted image at $z = 1$ by the proposed method in [27] in Example 3. Note that the ground truth image is shown in Fig. 16.

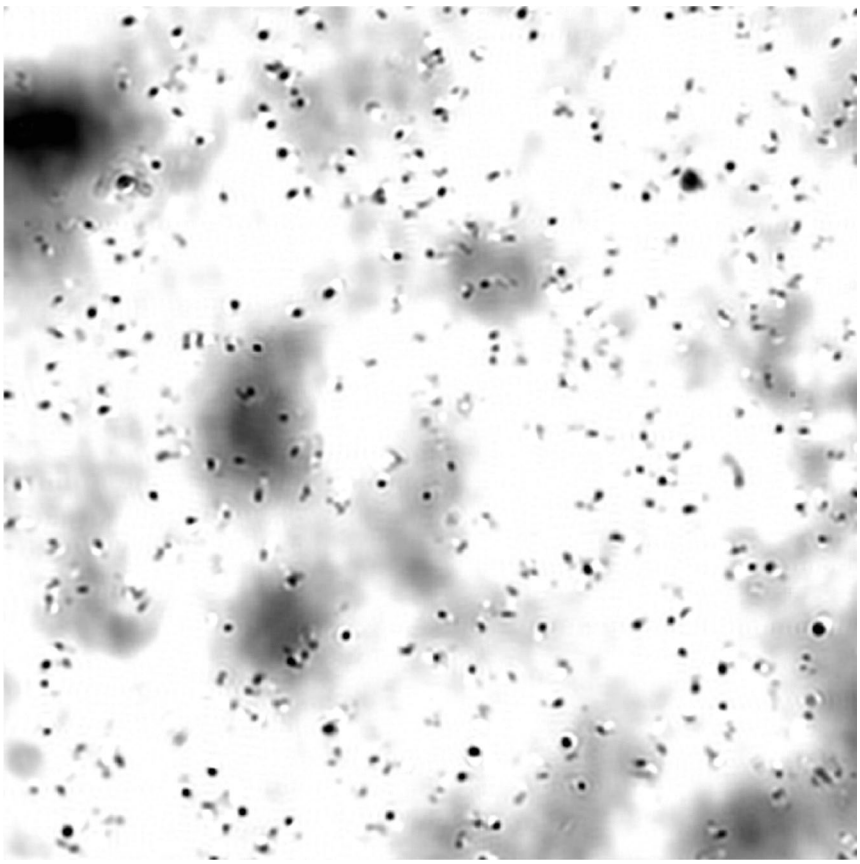


Fig. 18 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 200$ in Example 3. Note that the ground truth image is shown in Fig. 16

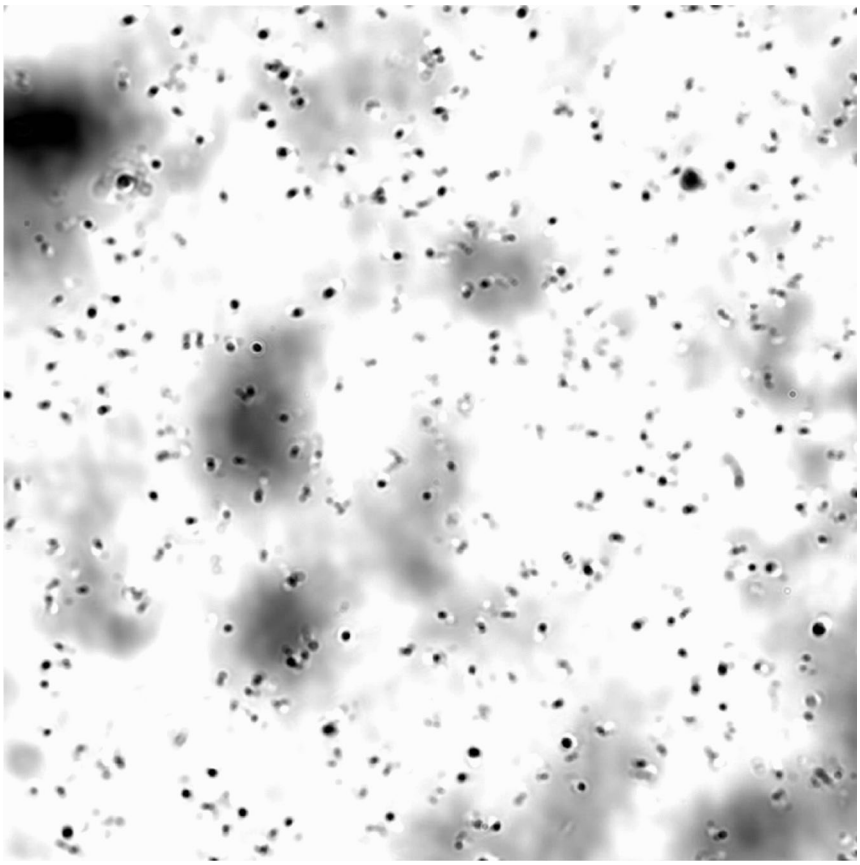


Fig. 19 Predicted image at $z = 1$ by the proposed method where $\tilde{l}_1 = \tilde{l}_2 = 400$ in Example 3. Note that the ground truth image is shown in Fig. 16

Acknowledgements This work was funded by the National Science Foundation(Grant Nos. EES-2000202, EES-2306449, EES-2219558) and supported by the NSF FAMU CREST Center award (Grant No. EES- 2514451). Acknowledgement is made to the donors of the American Chemical Society Petroleum Research Fund for support(or partial support) of this research. J.A. acknowledges partial support from NASA Award 80NSSC24M0175. Experimental work was performed at the National High Magnetic Field Laboratory, which is supported by National Science Foundation Cooperative Agreement No. DMR-2128556 and the State of Florida. This material is also based upon work supported by the Air Force Office of Scientific Research under award numbers FA9550-22-1-0247, FA9550-24-1-0353, FA8650-20-2-5853, and W911NF-24-1-0281. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation herein.

Author Contributions The authors confirm contribution to the paper as follows: study conception and design, analysis and interpretation of results, and draft manuscript preparation: Seyyed Shaho Alaviani; data collection: Jaideep Katuri; supervision: Jamel Ali, Olugbenga Moses Anubi. All authors reviewed the results and approved the final version of the manuscript.

Funding The funding was mentioned in Acknowledgments part above.

Data Availability Data sets generated during the current study are available from the authors on reasonable request.

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval and consent to participate N/A

Consent for publication None

Competing interests The authors declare no competing interests.

References

1. Leaman EJ, Sahari A, Traore MA, Geuther BQ, Morrow CM, Behkam B (2020) Data-driven statistical modeling of the emergent behavior of biohybrid microrobots. APL Bioeng 4:016104
2. Yang Z, Yang H, Cao Y, Cui Y, Zhang L (2023) Magnetically actuated continuum medical robots: a review. Adv Intell Syst 5:2200416
3. Bolopion A, Xie H, Haliyo DS, Régnier S (2012) Haptic teleoperation for 3-D microassembly of spherical objects. IEEE ASME Trans Mechatron 17:116–127
4. Wang B, Kostarelos K, Nelson BJ, Zhang L (2021) Trends in micro-/nanorobotics: materials development, actuation, localization, and system integration for biomedical applications. Adv Mater 33:2002047
5. Yan X, Zhou Q, Yu J, Xu T, Deng Y, Tang T, Feng Q, Bian L, Zhang Y, Ferreira A, Zhang L (2015) Magnetite nanostructured porous hollow helical microswimmers for targeted delivery. Adv Funct Mater 25(33):5333–5342

6. Lee JG, Raj RR, Day NB, Shields IV CW (2023) Microrobots for biomedicine: Unsolved challenges and opportunities for translation. ACS Nano 17, 14196?14204
7. Elnaggar A, Kang S, Tian M, Han B, Keshavarz M (2024) State of the art in actuation of micro/nanorobots for biomedical applications. Small Sci 4:2300211
8. Ongaro F, Pune S, Scheggi S, Misra S (2019) Design of an electromagnetic setup for independent three dimensional control of pairs of identical and nonidentical microrobots. IEEE Trans Robot 35:174–183
9. Liu X, Shi Q, Lin Y, Kojima M, Mae Y, Fukuda T, Huang Q, Arai T (2019) Multifunctional nanocontact micromanipulation using whirling flow generated by vibrating a single piezo actuator. Nano Micro Small 15:1804421
10. Xie H, Sun M, Fan X, Lin Z, Chen W, Wang L, Dong L, He Q (2019) Reconfigurable magnetic microrobot swarm: multimode transformation, locomotion, and manipulation. Sci Robot 4:eaav8006
11. Yang L, Yu J, Yang S, Wang B, Nelson BJ, Zhang L (2022) A survey on swarm microrobotics. IEEE Trans Robot 38:1531–1551
12. Yang L, Jiang J, Ji F, Li Y, Yung K-L, Ferreira A, Zhang L (2024) Machine learning for micro-and nanorobots. Nat Mach Intell 6(6):605–618
13. Sun Z, Cheng Y, Zhou C, Cheng E, Chen G, Dong L, Song B (2022) An indeterministic vision-based state-observer for grwoing magnetic microrobot motion status estimation. Proc of IEEE Int Conf Robot Autom (ICRA), May 23-27, Phildelphia, PA, USA, 9776–9782
14. Zöttl A, Stark H (2016) Emergent behavior in active colloids. J Phys Condens Matter 28:253001
15. Meng K, Jia Y, Yang H, Niu F, Wang Y, Sun D (2019) Motion planning and robust control for the endovascular navigation of a microrobot. IEEE Trans Ind Inform 16:4557–4566
16. Tang X, Li Y, Liu X, Liu D, Chen Z, Arai T (2022) Vision-based automated control of magnetic microrobots. Micromachines 13:337
17. Hansen E, Brunton SL, Song Z (2022) Swarm modeling with dynamic mode decomposition. IEEE Access 10:59508–59521
18. Hüttenrauch M, Šošić A, Neumann G (2019) Deep reinforcement learning for swarm systems. J Mach Learn Res 20:1–31
19. Zhong M, Miller J, Maggioni M (2020) Data-driven discovery of emergent behaviors in collective dynamics. Physica D Nonlinear Phenom 411:132542
20. Bongini M, Fornasier M, Hensen M, Maggioni M (2017) Inferring interaction rules from observations of evolutive systems I: the variational approach. Math Models Methods Appl Sci 27:909–951
21. Lu F, Maggioni M, Tang S (2021) Learning interaction kernels in heterogeneous systems of agents from multiple trajectories. J Mach Learn Res 22:1–67
22. Feng J, Maggioni M, Martin P, Zhong M (2022) Learning interaction variables and kernels from observations of agent-based systems. IFAC-PapersOnLine. https://doi.org/10.1016/j.ifacol.2022.11.046
23. Leaman EJ, Geuther BQ, Sahari A, Behkam B (2019) Statistical modeling of a distributed network of bacteria-propelled microrobots (bacteriaBots). International conference on manipulation, automation and robotics at small scales (MARSS). July 1-5, Helsinki, Finland
24. Hoed FMD, Carlotti M, Palagi S, Raffa P, Mattoli V (2024) Evolution of the microrobots: stimuli-responsive materials and additive manufacturing technologies turn small structures into microscale robots. Micromachines 15:Article 275
25. Yang L, Zhang Y, Vong C-I, Zhang L (2018) Automated control of multifunctional magnetic spores using fluorescence imaging for microrobotic cargo delivery. IEEE/RSJ International conference on intelligent robots and systems. Oct. 1-5, Madrid, Spain, pp 6180–6185
26. Yang L, Yu J, Zhang L (2020) Statistics-based automated control for a swarm of a paramagnetic nanoparticles in 2-D space. IEEE Trans Rob 36:254–270
27. Alaviani SS, Kturi J, Ali J, Anubi OM (2024) Vision-based learning of emergenet behavior of magnetic microrobots. Modeling, Estimation and Control Conference. Oct. 28-30, Chicago, Illinois, USA
28. Weiss J (2019) A tutorial on the proper orthogonal decomposition. Proc of AIAA Aviation Forum. June 17-21, Dallas, Texas, USA, pp 1–21
29. Roth J (2021) Proper orthogonal decomposition for fluid mechanics problems. Master thesis, Leibniz Universität Hannover, Hanover, Germany
30. Chatterjee A (2000) An introduction to the proper orthogonal decomposition. Curr Sci 78:808–817
31. Ye J (2005) Generalized low rank approximations of matrices. Mach Learn 61:167–191
32. Chen L, Zhou S, Ma J (2020) Stable sparse subspace embedding for dimensionality reduction. Knowledge-Based Systems 195:105639
33. Cai D, He X, Hu Y, Han J, Huang T (2007) Learning a spatially smooth subspace for face recognition. Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Minneapolis, MN, USA, pp 1–7
34. Pereira RB, Plastino A, Zadrozny B, Merschmann LHC (2018) Categorizing feature selection methods for multi-label classification. Artif Intell Rev 49:57–78
35. Lanczos C (1950) An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J Res Natl Bur Stand 45:255–282
36. Ren C-X, Dai D-Q (2010) Bilinear Lanczos components for fast dimensionality reduction and feature extraction. Pattern Recogn 43:3742–3752
37. Liu J, Chen S (2006) Non-iterative generalized low rank approximation of matrices. Pattern Recogn Lett 22:1002–1008
38. Yang J, Zhang D, Frangi A, Yang JY (2004) Two-dimensional PCA: a new appraoch to appearance based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26:131–137
39. Zhang D, Zhou Z (2005) $(2D)^2$ PCA: 2-directional 2-dimensional PCA for efficient face representation and recognition. Neurocomputing 69:224–231
40. Wu G, Xu W, Leng H (2015) Inexact and incremental bilinear Lanczos components algorithms for high dimensionality reduction and image reconstruction. Pattern Recogn 48:244–263
41. Lu C, Liu W, An S (2008) A simplified GLRAM algorithm for face recognition. Neurocomputing 72:212–217
42. Li K, Wu G (2021) A randomized generalized low rank approximations of matrices algorithm for high dimensionality reduction and image compression. Numer Linear Algebra Appl 28:e2338
43. Liberty E, Woolfe F, Martinsson P, Rokhlin V, Tygert M (2007) Randomized algorithms for low-rank approximation of matrices. Proceedings of the national academy of sciences of the US 104:20167–20172
44. Charumathi V, Ramakrishna M, Vasudevan V (2022) Fast and accurate proper orthogonal decomposition using efficient sampling and iterative techniques for singular value decomposition. ACM Trans Math Softw 48:16
45. Higham NJ (2008) Functions of Matrices: Theory and Computation. SIAM, Philadelphia, PA
46. Meijer JM, Rossi L (2021) Preparation, properties, and applications of magnetic hematite microparticles. Soft Matter 17:2354–2368