Accelerating particle-in-cell kinetic plasma simulations via reduced-order modeling of space-charge dynamics using dynamic mode decomposition

Indranil Nayak [*] and Fernando L. Teixeira

ElectroScience Laboratory and Department of Electrical and Computer Engineering, The Ohio State University, Columbus, Ohio 43212, USA

Dong-Yeop Na

Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

Mrinal Kumar

LADDCS, Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, Ohio 43210, USA

Yuri A. Omelchenko

Trinum Research Inc., San Diego, California 92126, USA

We present a data-driven reduced-order modeling of the space-charge dynamics for electromagnetic particle-in-cell (EMPIC) plasma simulations based on dynamic mode decomposition (DMD). The dynamics of the charged particles in kinetic plasma simulations such as EMPIC is manifested through the plasma current density defined along the edges of the spatial mesh. We showcase the efficacy of DMD in modeling the time evolution of current density through a low-dimensional feature space. Not only do such DMD-based predictive reduced-order models help accelerate EMPIC simulations, they also have the potential to facilitate investigative analysis and control applications. We demonstrate the proposed DMD-EMPIC scheme for reduced-order modeling of current density and speedup in EMPIC simulations involving electron beam under the influence of magnetic field, virtual cathode oscillations, and backward wave oscillator.

I. INTRODUCTION

Kinetic plasma simulations enjoy a broad range of applications ranging from the modeling of high-power microwave sources, directed energy devices, particle accelerators, terahertz devices, etc. They are also used to improve our understanding of ionospheric phenomena, magnetosphere regions, and astrophysical events [1–4]. Historically, electromagnetic particle-in-cell (EMPIC) algorithms have been a popular choice for simulating collisionless kinetic plasmas. Compared to magnetohydrodynamics (MHD) simulations, particle-in-cell (PIC) algorithms can better capture intricate wave-particle interactions including electron bunching, kinetic instabilities, Landau damping, microscopic turbulence, space-charge effects, etc., [5–10]. However, higher physical accuracy of the PIC simulations comes at the cost of large computational resources. A very large number of computational (super)particles are needed for an adequate sampling of the phase space of the electrons and ions. This is one of the primary challenges preventing speeding up of EMPIC simulations [5]. In an EMPIC simulation, the position and velocity of each superparticle must be updated individually at each time step, covering the entire time-window of interest.

The high dimensionality and large computational cost of EMPIC simulations serve as motivation for developing reduced-order models capable of capturing the underly-ing plasma dynamics through a low-dimensional feature space. Such small sets of coherent spatiotemporal features have been shown to be effective in emulating the underlying physics in several plasma simulations [11–16]. The projection-based reduced-order methodologies such as proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD), in particular, have been recently employed for modeling the field evolution in EMPIC simulations with success [16–19]. However, these methods do not address the primary computational bottleneck due to the large number of particles. The authors in Ref. [20] presented a discrete empirical interpolation (DEIM) method for reducing the computational burden of particle to mesh projections in geometric particle-in-cell (GEMPIC) [21] simulations. In Ref. [22], the authors took a different approach based on sparse regression (SR) to discover the underlying partial differential equations (PDEs) from PIC simulation data. While Ref. [20] showed the effectiveness of the DMD-DEIM method for 1*D*-1*V* Vlasov-Poisson problems in a parametric setting, it did not address how to provide an explicit analytical time-update scheme for particle-dependent quantities such as current density, which can be crucial for prediction and diagnostic purposes. The SR approach in Ref. [22] strived to infer the corresponding MHD equations in integral form from the PIC time-series data, but was limited by the choice of PDE candidate terms and the cost of integration over a large volume. In this paper, the computational bottleneck of a large number of particles is addressed by time-domain reduced-order model-
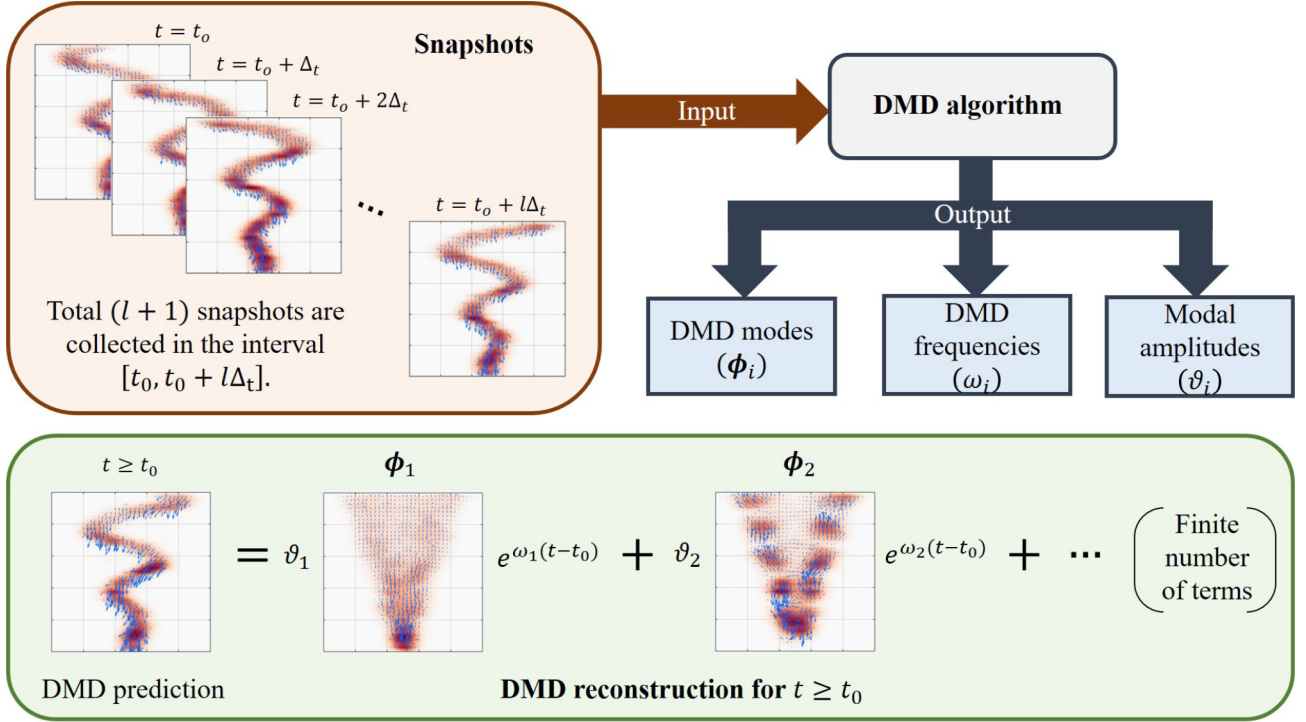
---

*[*]nayak.77@osu.edu

FIG. 1. Schematic illustration of the DMD algorithm for current density modeling of an electron beam. Since $\boldsymbol{\phi}_m$ are typically complex, for illustration purposes, the plots corresponding to $(\boldsymbol{\phi}_m + \overline{\boldsymbol{\phi}}_m)$ are shown as $\boldsymbol{\phi}_m$.

ing of the current density using dynamic mode decomposition (DMD). In a typical EMPIC setting, the charged particle dynamics manifests through temporal variation of charge and current densities defined on the spatial mesh. In the EMPIC algorithm, current density is employed for the time update of electromagnetic fields according to the Maxwell's equations. Updated fields, in turn, influence the motion of the charged particles and dictate the time variation of current density. These steps are executed in a cyclic fashion (bottom left of Fig. 2) for each time step of EMPIC. In essence, if the time evolution of current density can be modeled independently of the particles, the EMPIC steps involving the particles could be completely avoided, which would result in a drastic reduction in computation time.

Initially proposed in Ref. [23], DMD is a data-driven method for low-dimensional surrogate modeling of high-dimensional complex dynamical systems. Since its inception, DMD has been applied successfully to a vast array of problems, including fluid-based nonlinear plasma models and magnetized plasma experiments [15,24,25]. The ability of DMD to extract underlying dominant spatiotemporal features from self-field data of EMPIC simulation, and the effect of DMD extrapolated fields on the particle dynamics were recently investigated in Ref. [16]. In Ref. [20], DMD was used for approximating the electric field potential. However, DMD modeling of space-charge dynamics, especially the time evolution of *plasma current density* in EMPIC simulations, have not yet been explored. Since there is no analytical model available for the time evolution of current density in EMPIC simulations, a data-driven approach becomes crucial. Data-driven modeling of current density is particularly challenging due to the associated nonlinearity, nonsmooth time variation pertain-

ing to particle noise, and high dimensionality (large number of mesh elements). The effectiveness of DMD in modeling nonlinear dynamics can be attributed to its close relation to Koopman operator theory [26]. In particular, initial findings have suggested that Koopman autoencoders have good potential for reduced order modeling of currents as well [27,28]. However, Koopman autoencoders display the usual limitations of neural network models, namely, lack of interpretability (black-box nature) and intricate training process. The use of interpretable Koopman-based data-driven reduced-order models such as DMD helps to overcome these challenges.

In this paper, we demonstrate the application of time-series prediction in current density forecasting to accelerate EMPIC simulations. The main contributions of the present paper can be summarized as follows:

(1) This paper constructs an interpretable, reduced-order model for the space-charge dynamics in EMPIC simulations. This is achieved by DMD modeling of the current density, which is essentially the manifestation of charged particle dynamics in the EMPIC simulations. While our main goal is to accelerate the EMPIC simulations, such reduced-order modeling of space-charge dynamics also helps analyze and diagnose the problem at hand. Note that using DMD-based linear reduced-order models (ROMs) of the inherently nonlinear plasma dynamics also opens the door for leveraging control of theoretic tools that already exist for linear systems. The DMD modes and frequencies can help in analyzing the efficiency loss due to harmonic generation in high-power microwave devices. By analyzing the spatial patterns and growth rates of the modes, DMD can help predict areas that are prone to decay or damage due to high energy densities, heating, or other factors. DMD can also identify unstable
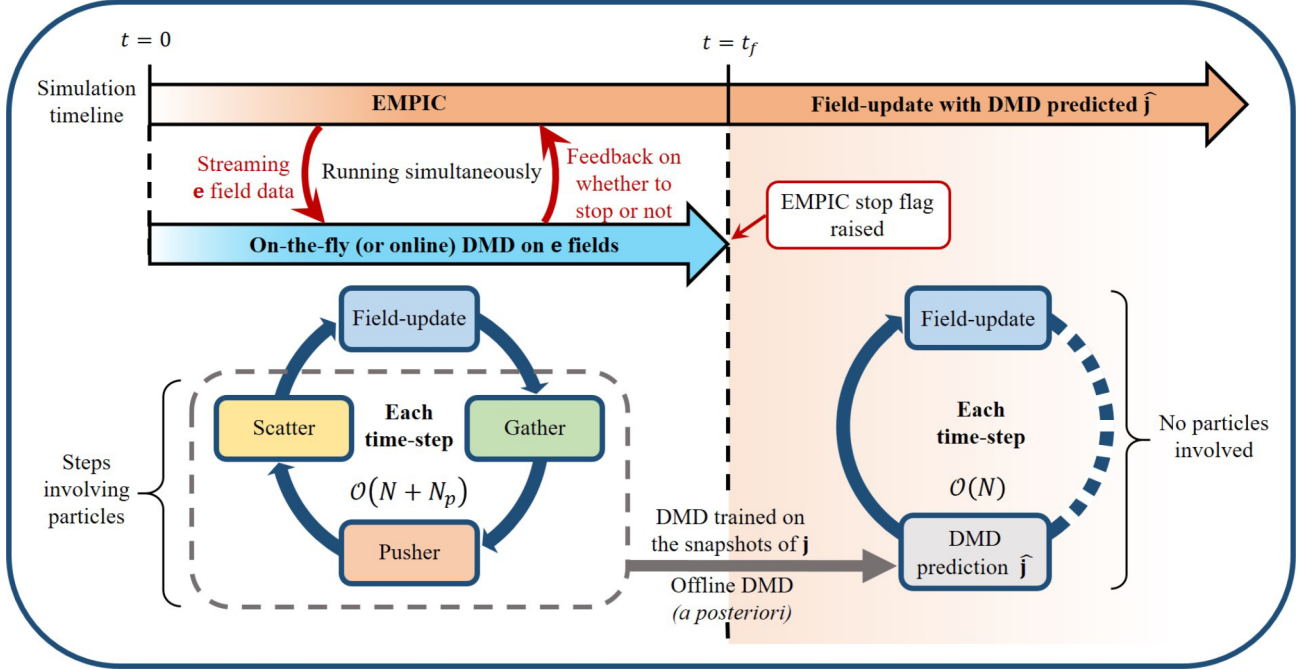
FIG. 2. DMD-EMPIC algorithm for accelerating EMPIC simulations. Note that for $t > t_f$, the stages are not exactly cyclic since fields no longer have any effect on the DMD-predicted current density (illustrated by the broken line).

modes and flaws in device design or the numerical solver itself. Furthermore, understanding the dominant modes and their characteristics can help optimize the design parameters of such devices.

(2) A DMD-EMPIC algorithm (Fig. 2) is presented to accelerate the EMPIC simulations showing post-transient behavior, i.e., either steady-state, equilibrium, or any type of periodic behavior. The time-domain DMD model of the current density implements rapid prediction of the current density at any time instant, and thus eliminates the need for EM-PIC stages involving particles. The DMD-EMPIC strategy utilizes the on-the-fly algorithm developed in Ref. [29] to detect the end of transience in real time. It then replaces the computationally expensive *gather, pusher,* and *scatter* with DMD predicted current beyond that point in time. It is important to highlight that the key distinction between this paper and previous research, as outlined in Ref. [16], lies in the application of the DMD model to current density for replacing EMPIC stages involving particles. This approach not only facilitates the prediction of future current density but also enables fast and accurate forecasting of self-field values, adhering precisely to the discrete Maxwell's equations. Moreover, this study employs a more versatile on-the-fly algorithm, distinguishing it from the method utilized in Ref. [16]. DMD-EMPIC has the potential to significantly expedite the EMPIC simulations for plasma systems showing long oscillations (e.g., limit-cycle behavior).

This paper is organized as follows. Section II provides a quick overview of the EMPIC algorithm used to generate the high-fidelity data and the DMD algorithm which is trained on that high-fidelity data. Section III introduces the DMD-EMPIC algorithm for accelerating EMPIC simulations. Section IV provides a series of results showcasing the effectiveness of DMD-EMPIC for accelerating EMPIC

simulations. An analysis of the computational gain for the examples considered, and of the computational complexity in general, is provided in Sec. V. Finally, Sec. VI summarizes the main takeaways.

II. THEORETICAL BACKGROUND

A. EMPIC algorithm

The EMPIC algorithm [21,30–34] primarily consists of four steps: (i) *field-update*, (ii) *gather*, (iii) *pusher*, and (iv) *scatter* (bottom left of Fig. 2), which are executed in a cyclic fashion at each time step. For completeness, we will briefly discuss each step in the context of a fully kinetic finite-element-based charge-conserving EMPIC algorithm [30,31,35]. We will also discuss the computational hurdle that a large number of particles poses in EMPIC simulations.

B. Field update

The field-update stage deals with the time update of the electric and magnetic fields by solving Maxwell's equations. Here, the field update is obtained on a finite element mesh where the fields are represented as (discrete) differential forms [36–41]. The electric field one-form $\mathcal{E}(t, \mathbf{r})$ and magnetic flux density two-form $\mathcal{B}(t, \mathbf{r})$ are expanded as

$$\mathcal{E}(t, \mathbf{r}) = \sum_{i=1}^{N_1} e_i(t) w_i^{(1)}(\mathbf{r}), \tag{1a}$$

$$\mathcal{B}(t, \mathbf{r}) = \sum_{i=1}^{N_2} b_i(t) w_i^{(2)}(\mathbf{r}), \tag{1b}$$

where $e_i(t)$ and $b_i(t)$ are their temporal degrees of freedom (DoF), and $w_i^{(1)}(\mathbf{r})$ and $w_j^{(2)}(\mathbf{r})$ represent Whitney one-forms (edge-based) and Whitney two-forms (facet based),

respectively [30,35,42]. N_1 and N_2, respectively, denote the number of edges and elements (faces) in the mesh. From (1a) and (1b) and using a Galerkin-type finite element discretization, Maxwell's curl equations in discrete form suitable for time stepping can be written as [30,35]

$$\mathbf{b}^{(n+\frac{1}{2})} = \mathbf{b}^{(n-\frac{1}{2})} - \Delta t \mathbf{C} \cdot \mathbf{e}^{(n)}, \tag{2a}$$

$$\mathbf{e}^{(n+1)} = \mathbf{e}^{(n)} + \Delta t [\star_\epsilon]^{-1} \cdot \mathbf{C}^T \cdot [\star_{\mu^{-1}}] \cdot \mathbf{b}^{(n+\frac{1}{2})}$$
$$- \Delta t [\star_\epsilon]^{-1} \cdot \mathbf{j}^{(n+\frac{1}{2})}, \tag{2b}$$

where Δt denotes the time step $\mathbf{e} = [e_1 \, e_2 \, \ldots \, e_{N_1}]^T$ is defined on integer time steps, and $\mathbf{b} = [b_1 \, b_2 \, \ldots \, b_{N_2}]^T$ and current density $\mathbf{j} = [j_1 \, j_2 \, \ldots \, j_{N_1}]^T$ are defined on half-integer time steps. More details on $\mathbf{j}$ are provided in the description of the scatter stage ahead. The incidence matrices $\mathbf{C}$ and $\mathbf{C}^T$ encode the exterior derivative (or, equivalently, the curl operator distilled from the metric structure) on the primal and dual meshes, respectively [35,42]. The elements of $\mathbf{C}$ are in the set $\{-1, 0, 1\}$ as they contain information about adjacency and relative orientation among mesh elements. The Hodge matrices $[\star_\epsilon]$ and $[\star_{\mu^{-1}}]$ represent generalized constitutive relations incorporating the metric information from the finite element mesh. Further details on the discrete field update equations can be found in Refs. [30,31,35].

C. Gather

In the gather step, the fields are calculated at the particle position $\mathbf{r}_p$ using the interpolatory Whitney functions:

$$\mathcal{E}(n\Delta t, \mathbf{r}_p) = \mathcal{E}_p^{(n)} = \sum_{i=1}^{N_1} e_i^{(n)} w_i^{(1)}(\mathbf{r}_p), \tag{3a}$$

$$\mathcal{B}\left(\left(n + \frac{1}{2}\right)\Delta t, \mathbf{r}_p\right) = \mathcal{B}_p^{(n+\frac{1}{2})} = \sum_{i=1}^{N_2} b_i^{(n+\frac{1}{2})} w_i^{(2)}(\mathbf{r}_p). \tag{3b}$$

D. Pusher

Once the fields are interpolated to each charged particle location, the particle position and velocity are updated using Newton's laws of motion and Lorentz force equations. The particle position $\mathbf{r}_p$ and (nonrelativistic) velocity $\mathbf{v}_p$ can be obtained as follows [30]:

$$\mathbf{r}_p^{n+1} = \mathbf{r}_p^n + \Delta t \mathbf{v}_p^{n+\frac{1}{2}}, \tag{4a}$$

$$\mathbf{v}_p^{n+\frac{1}{2}} = N^{-1} \cdot N^T \cdot \mathbf{v}_p^{n-\frac{1}{2}} + \frac{q_p \Delta t}{m_p} N^{-1} \cdot \mathcal{E}_p^{(n)}, \tag{4b}$$

where

$$N = \begin{bmatrix} 1 & -\frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,z}^{(n)} & \frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,y}^{(n)} \\ \frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,z}^{(n)} & 1 & -\frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,x}^{(n)} \\ -\frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,y}^{(n)} & \frac{q_p \Delta t}{2m_p} \mathcal{B}_{p,x}^{(n)} & 1 \end{bmatrix}, \tag{5}$$

$\mathcal{B}_p^{(n)} = \frac{1}{2}(\mathcal{B}_p^{(n+\frac{1}{2})} + \mathcal{B}_p^{(n-\frac{1}{2})})$, and m_p and q_p are, respectively, the mass and charge of the pth particle. Relativistic motion can also be incorporated, if required.

E. Scatter

The scatter step is essentially a particle-to-mesh mapping where the motion of each charged-particle during time step

Δt is mapped to the edges of the mesh to obtain each element of the current density vector $\mathbf{j}$

$$j_i = \sum_{p=1}^{N_p} \frac{q_p}{\Delta t} \int_{\mathbf{r}_{p1}}^{\mathbf{r}_{p2}} w_i^{(1)}(\mathbf{r}_p) \cdot d\mathbf{r}_p, \tag{6}$$

where the integration is performed along the straight line trajectory of the pth particle during one time step, from $\mathbf{r}_{p1}$ to $\mathbf{r}_{p2}$ [43]. The particular mapping ensures charge conservation at the discrete level, as shown in Ref. [30]. Mathematically, the above equation can be understood as a Galerkin projection of the two-form $\mathcal{J}$ (resulting from the collective movement of the charged particles) onto the Whitney one-form $w_i^{(1)}$ associated to edge i of the mesh. The edge current density vector $\mathbf{j}$ thus obtained is subsequently used to update the field values in the next time step according to (2).

The *gather, pusher,* and *scatter* steps need to be performed individually for each superparticle in the solution domain. As mentioned earlier, to accurately capture the phase-space evolution, a large number of particles is often necessary, a typical number being on the order of 10^6 or so.

F. DMD algorithm

Let us consider a discrete-time dynamical system consisting of the state $\mathbf{x} \in \mathcal{M}$, where $\mathcal{M}$ is an N-dimensional manifold and F is a sequential flow map $F : \mathcal{M} \mapsto \mathcal{M}$ such that

$$\mathbf{x}^{(n+1)} = F(\mathbf{x}^{(n)}). \tag{7}$$

The superscript n represents the discrete time index. In practical scenarios, $\mathcal{M}$ is typically represented by $\mathbb{R}^N$, N being the number of mesh elements (nodes, edges, faces, etc.) over which $\mathbf{x}$ is defined. It is important to note that in our context, $\mathbf{x}^{(n)}$ may represent $\mathbf{e}^{(n)}$, $\mathbf{j}^{(n+\frac{1}{2})}$, or any other relevant dynamic variable as introduced in the previous section. DMD collects time snapshots of the state $\mathbf{x}$ and generates a set of spatial DMD modes $\boldsymbol{\phi}_i(\mathbf{r})$, where $\mathbf{r}$ is the position vector, with corresponding DMD frequencies ω_i (in general, complex valued) and modal amplitudes ϑ_i ($i = 1, 2, \ldots$). These extracted DMD features are used to (approximately) reconstruct the spatiotemporal behavior of the original dynamics [23,44,45].

The first stage of the DMD process is snapshot collection, i.e., sampling the state $\mathbf{x}$ at different time instants to build the training data set. The snapshots are typically collected with a uniform sampling interval from either high-fidelity simulations (i.e., EMPIC) or experimental data. The corresponding time window is referred to as the training window (region), or DMD harvesting window (region) or simply DMD window. Let the DMD harvesting window consist of $(l + 1)$ snapshots, spanning from $t_0 = n_0\Delta t$ to $t_l = (n_0 + l\Delta_n)\Delta t$, Δ_n, and Δt being the number of time steps between two consecutive snapshots, and the time step interval, respectively. The snapshot matrix X and the shifted snapshot matrix X' are formed as

$$X = [\mathbf{x}^{(n_0)} \quad \mathbf{x}^{(n_0+\Delta_n)} \ldots \mathbf{x}^{(n_0+(l-1)\Delta_n)}], \tag{8a}$$

$$X' = [\mathbf{x}^{(n_0+\Delta_n)} \quad \mathbf{x}^{(n_0+2\Delta_n)} \ldots \mathbf{x}^{(n_0+l\Delta_n)}]. \tag{8b}$$

Assuming a linear predictive relationship $X' \approx A \cdot X$, DMD proceeds to extract the eigenvalues and eigenvectors of A in an efficient fashion. Singular value decomposition (SVD) of the snapshot matrix X, results in the U, Σ, and V matrices

such that

$$X = U\Sigma V^*, \tag{9}$$

where the superscript '∗' denotes complex conjugate transpose. The columns of U are essentially the POD modes [46] which capture the dominant spatial pattern. The columns of V represent the corresponding temporal pattern whereas the singular values (elements of the diagonal matrix Σ) indicate the weight (importance) of corresponding modes. Typically, the singular values show an exponentially decay pattern hinting at the underlying low-dimensional structure. We only retain the first r $(< l)$ singular values corresponding to the first r columns of U, V, leading to reduced SVD matrices U_r, Σ_r, and V_r such that $X \approx U_r\Sigma_r V_r^*$. The value of r can be chosen based on a hard energy threshold or optimal hard thresholding [47,48]. Consequently, A can be approximated as

$$A \approx X'V_r\Sigma_r^{-1}U_r^*. \tag{10}$$

Next, A is projected onto the columns of U_r, leading to

$$\tilde{A} = U_r^*AU_r = U_r^*X'V_r\Sigma_r^{-1}. \tag{11}$$

The lower dimensionality of $\tilde{A}$ makes the eigendecomposition operation $\tilde{A}W = W\Lambda$ computationally efficient, where the diagonal matrix Λ contains the eigenvalues λ_i, $i = 1, 2, \ldots, r$ that approximate the eigenvalues of A. The extracted spatial DMD modes $\boldsymbol{\phi}_i(\mathbf{r})$ are given by the columns of the matrix $\Phi = X'V_r\Sigma_r^{-1}W$ [44], resulting in the DMD reconstruction $\hat{\mathbf{x}}$ of the dynamic state for $t \geqslant t_0$, i.e.,

$$\mathbf{x}(\mathbf{r}, t) \approx \hat{\mathbf{x}}(\mathbf{r}, t) = \sum_{i=1}^{r} \vartheta_i\boldsymbol{\phi}_i(\mathbf{r})e^{\omega_i(t-t_0)}, \tag{12}$$

where $\omega_i = \ln(\lambda_i)/\Delta_t$, Δ_t being the time interval between two consecutive DMD snapshots i.e., $\Delta_t = \Delta_n\Delta t$. The modal amplitudes ϑ_i can be calculated by solving an optimization problem as described in Ref. [49]. Further details can be found in Refs. [23,44,45,50]. The width of the DMD data harvesting window is generally taken such that it can capture the entire dynamics. For systems showing periodic behavior, the DMD window should ideally cover multiple periods. The basis for selection of Δ_t is the Nyquist criterion and noise frequency. In practice, since we generally deal with *real* data, the DMD modes ($\boldsymbol{\phi}_i$), frequencies (ω_i), and modal amplitudes (ϑ_i) as in (12) appear in complex-conjugate pairs. Equation (12) can be written in terms of M complex-conjugate (overbar) pairs as

$$\hat{\mathbf{x}}(\mathbf{r}, t) = \sum_{m=1}^{M} (\vartheta_m\boldsymbol{\phi}_m e^{\omega_m(t-t_0)} + \overline{\vartheta}_m\overline{\boldsymbol{\phi}}_m e^{\overline{\omega}_m(t-t_0)}). \tag{13}$$

For purely real modes [direct current (dc) modes], two terms in (13) can be combined into a single term with $2M \geqslant r$. From this point onward, by the mth DMD mode or frequency, we will refer to the mth complex-conjugate pair. Also, for conciseness, we will address both ω_m and f_m as the mth DMD frequency where $f_m = |\frac{\text{Im}\{\omega_m\}}{2\pi}|$ (Im$\{\cdot\}$ represents the imaginary part). A schematic representation of the DMD process for current density modeling is provided in Fig. 1.

DMD's efficacy in modeling nonlinear dynamics can be attributed to its close relation to the Koopman operator [26]. The Koopman operator theory tells us that with a suitable transformation $g(\cdot)$, with $g : \mathcal{M} \mapsto \mathcal{C}$ residing in the infinite-dimensional Hilbert space, a dynamical system, nonlinear in the original state space $\mathbf{x}$, can be represented by a linear dynamical system in the transformed feature or observable space $g(\mathbf{x})$ [51]. If we can find a finite (p)-dimensional subspace invariant under the Koopman operator, the infinite-dimensional Koopman operator can be represented by a $p \times p$ matrix $\mathbf{K}$, such that the time evolution of a vector-valued observable $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}); g_2(\mathbf{x}); \ldots; g_p(\mathbf{x})]$ belonging to that subspace is given by $\mathbf{g}(\mathbf{x}^{(n+1)}) = \mathbf{K} \cdot \mathbf{g}(\mathbf{x}^{(n)})$, where [;] denotes the vertical stacking. DMD assumes an identity transformation, i.e., $\mathbf{g}(\mathbf{x}) = \mathbf{x}$, which works well for variety of high-dimensional nonlinear dynamical systems [25,26,44,45,52,53]. However, the assumption of $\mathbf{g}(\mathbf{x}) = \mathbf{x}$ might not be sufficient for highly nonlinear systems, including kinetic plasmas. To adequately capture nonlinearities, the Hankel variant of DMD makes use of the time-delay coordinates where the vector valued observable at the nth time step is given by

$$\mathbf{g}(\mathbf{x}^{(n)}) = [\mathbf{x}^{(n)}; \mathbf{x}^{(n-\Delta_n)}; \ldots; \mathbf{x}^{(n-(d-1)\Delta_n)}], \tag{14}$$

where ";" indicates vertical stacking, d is the number of Hankel stacks [29,50,54,55], and Δ_n is the DMD sampling interval in terms of timesteps. The success of delay embeddings in forming a suitable vector-valued Koopman observable is rooted in Takens' embedding theorem [55,56], which states that under certain conditions, the attractor of a dynamical system in delayed coordinates is *diffeomorphic* to the attractor of original system. Broadly speaking, incorporation of time-delayed embeddings helps better model the nonlinearities through a linear model such as DMD. Here, we apply the Hankel DMD which follows the same steps as the regular DMD, but with two notable exceptions. First, instead of $\mathbf{x}$, we use $\mathbf{g}(\mathbf{x})$ (14) to form the snapshot matrices (8) and, second, we retain the first N elements of the DMD modes ($\boldsymbol{\phi}$) to reconstruct the state as in (12).

III. DMD-EMPIC ALGORITHM

As described in Sec. II A, the primary computational bottleneck comes from the *gather*, *pusher*, and *scatter* stages as they involve each and every particle in the solution domain. However, these steps are necessary as they dictate the time evolution of the current density, which subsequently helps update the electromagnetic fields. We try to address this issue by directly modeling the time evolution of current density $\mathbf{j}$ using DMD. An analytical equation of the type (13) for time variation of current density in EMPIC serves the following purpose:

(1) DMD helps in rapid prediction of $\mathbf{j}$ with negligible computation cost compared to the particle operations. Also, a linear time-evolution model for current density can facilitate control theory applications.

(2) The *gather*, *pusher*, and *scatter* stages can be replaced by the DMD-predicted $\mathbf{j}$, which is then used to update the electric and magnetic fields in the *field-update* stage (Sec. II B) within the finite-element time-domain (FETD) setting. This can drastically reduce the computation cost of EMPIC.

The DMD-EMPIC algorithm is illustrated in Fig. 2. It consists of two main phases, the transient ($t \leqslant t_f$) and post-transient phase ($t > t_f$).

(1) Transient phase ($t \leqslant t_f$): The high-fidelity EMPIC simulation is run until the transient phase ends, denoted by the final time t_f. The on-the-fly DMD algorithm [29] is run

simultaneously with the ongoing EMPIC simulation to identify t_f on the fly. The electric field data (**e**) is fed to the Hankel DMD algorithm and provides feedback in real time regarding whether to stop the EMPIC simulation or not. To maximize computational gains, it is desirable to terminate the time-consuming EMPIC simulation at the earliest opportunity. At the same time, if the simulation is terminated too early (i.e., before the transient ends), DMD will not be able to make accurate time extrapolation due to the lack of quality training data [16]. Thus, a real-time algorithm for timely termination of high-fidelity EMPIC simulation is necessary. In the DMD-EMPIC algorithm, this is achieved by the sliding-window on-the-fly DMD algorithm developed in Ref. [29] which analyzes the time evolution of **e** to detect end of transience at $t = t_f$. However, to handle the repetitive execution of DMD for large data sets, we make modifications to the algorithm described in Ref. [29]. Instead of using the standard version of DMD, we perform randomized DMD to reduce the computational load. The on-the-fly algorithm is described in the Appendix.

(2) *Post-transient phase* ($t > t_f$): Following the detection of end-of-transience (t_f), Hankel DMD is performed in offline or *a posteriori* fashion on the snapshots of **j** collected inside the final DMD window. The purpose of the offline DMD is to predict the current density beyond t_f, denoted by **ĵ**. Typically, the *gather*, *pusher,* and *scatter* steps are required for the time update of **j**. Here, since an analytical expression [similar to (12)] for the time evolution of **j** is available from the offline DMD, we avoid these steps by using the DMD-predicted **ĵ**. The predicted current **ĵ** is then used in consecutive time steps to update the self-electric and magnetic fields in the *field-update* stage (Sec. II B). Note that for $t > t_f$, the relation between the self-fields and current density is not exactly cyclic. Beyond $t = t_f$, the fields do not have any effect on the time evolution of the current density **ĵ**.

As shown in Fig. 2, before the end of transience is detected ($t < t_f$), the computation cost for each time step of the simulation is the same as the cost of typical EMPIC [$O(N + N_p)$] with the added cost of on-the-fly DMD, where N is aggregate mesh dimension and N_p is the total number of charged particles. However, beyond t_f the computation cost per time step reduces to $O(N)$, which is a significant reduction in computation time given that $N_p \gg N$ in typical EMPIC settings [57]. The computational gain of the proposed DMD-EMPIC is discussed in detail in Sec. V.

IV. RESULTS

A. Oscillating electron beam

A 2D electron beam propagating along the positive y direction and oscillating under the influence of an external transverse magnetic flux is shown in Fig. 3(a). The solution domain (xy plane) is a square cavity of dimension 1 cm × 1 cm, which is discretized via an unstructured mesh composed of triangular elements. The mesh consists of $N_0 = 844$ nodes, $N_1 = 2447$ edges, and $N_2 = 1604$ elements (triangles). Superparticles [blue dots in Fig. 3(a)], are injected at the bottom of the cavity in the $+ve$ y direction with a velocity of 5×10^6 m/s. The superparticles are injected at the rate of 50 per time step in a random fashion uniformly in the range [0.45 cm, 0.55 cm]. The superparticles discretize the phase space of the electrons assuming a delta distribution in both position and velocity space. Superparticles are treated as point charges with mass $m_{sp} = r_{sp} m_e$ and charge $q_{sp} = r_{sp} q_e$, where m_e and q_e are, respectively, the mass and charge of an electron, and $r_{sp} = 5000$ is the number of actual electrons represented by each superparticle (superparticle ratio). An external oscillating magnetic flux $\mathcal{B}_{ext} = \mathcal{B}_0 \sin(2\pi / T_{osc}) \hat{z}$ is applied in the z direction, where $\mathcal{B}_0 = 2.5 \times 10^{-2}$ T, and $T_{osc} = 0.8$ ns. The simulation is run until $n = 320\,000$ time steps or $t = 64$ ns with the time-step interval $\Delta t = 0.2$ ps.

The post-transient snapshot of the current density **j** is shown in Fig. 3(b) at $t = 16$ ns ($n = 80\,000$). The goal is to model the time evolution of such snapshots inside the cavity using DMD. Unlike electromagnetic fields, **j** is restricted to

(a) Electron beam snapshot at $t = 64$ ns.

(b) Current density snapshot at $t = 64$ ns.
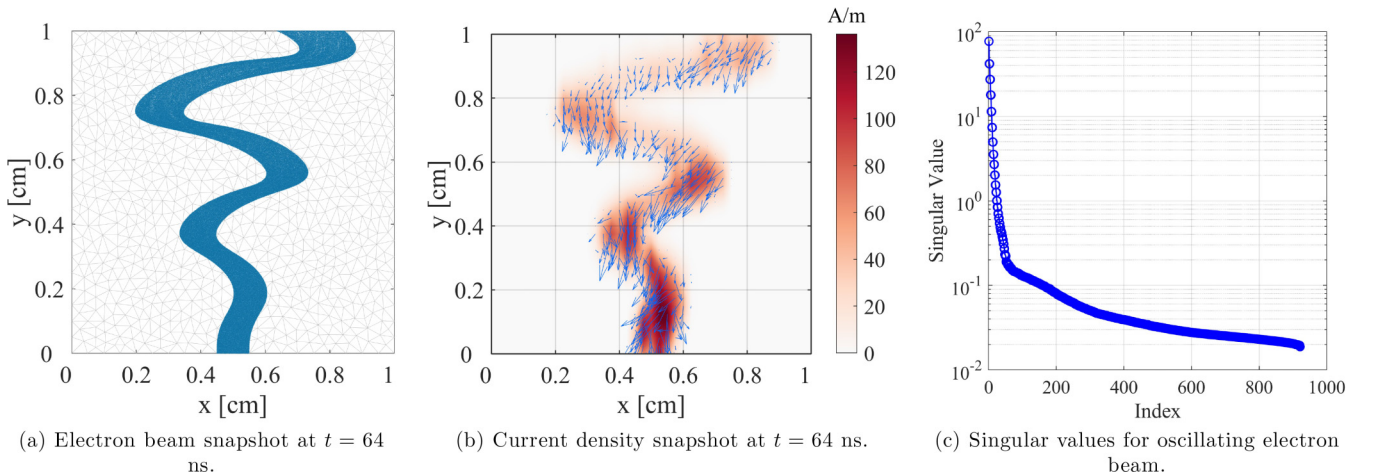
(c) Singular values for oscillating electron beam.

FIG. 3. (a) Snapshot of the electron beam at $t = 64$ ns. The beam is propagating along the $+ve$ y direction, and oscillating under the influence of a z-directed transverse magnetic flux. The blue dots represent superparticles and grey lines show the finite element mesh edges. (b) Snapshot of the current density at $t = 64$ ns. The magnitude is shown by a color map [59], whereas the direction is denoted by the blue arrows. Note that in all the current density plots, the magnitude color map is smoothed for visualization purpose. (c) Singular values after performing SVD on the snapshots of current density.

TABLE I. DMD parameters for modeling current density.

Parameters	Oscillating beam	Vircator	BWO
t_{st}	15.60 ns	22.40 ns	91.20 ns
t_{en}	23.60 ns	30.40 ns	107.20 ns
Δt_w	8 ns	8 ns	16 ns
Δ_t	8 ps	40 ps	2 ps
d	80	50	20
r	302	42	1999
M	152	22	1006

only the mesh elements interacting with the particles. In other words, the number of active edges N_{1a} over which $\mathbf{j}$ is nonzero (within the DMD window span) is less than total number of mesh edges N_1. We only consider those active edges for DMD modeling with $\mathbf{x} = \mathbf{j}_a$, where $\mathbf{j}_a$ is the current density vector with active edges. After performing the DMD, we revert back to the original state space with zero padding.

1. On-the-fly DMD on fields

The on-the-fly DMD is carried out on the electric field data to detect the end of transience or onset of the periodic behavior as described in the Appendix [29]. Approximate prior knowledge about the timescale is required to choose the DMD window width Δt_w accordingly, ensuring that it covers multiple oscillation cycles. We select $\Delta t_w = 8$ ns. Once the EMPIC simulation reaches $t = \Delta t_w$, fast Fourier transform (FFT) is

performed on a randomly chosen set of 20 points in space. Averaged FFT allows us to select the DMD sampling interval $\Delta_t = 8$ ps. We select the target rank $r = 200$ and number of Hankel stacks $d = 10$. The shift in consecutive sliding DMD windows is $\delta t_w = 0.4$ ns. The on-the-fly DMD parameters are mentioned in detail in Table IV. The onset of equilibrium (end of transience) is detected at $t = t_f = 23.06$ ns.

2. Offline DMD on current density

The primary contribution of this paper is to accelerate EMPIC simulations by DMD modeling of the plasma current density. EMPIC stops at $t = t_f$ (detected end of transience), and offline DMD on current density $\mathbf{j}$ is performed in the window $t \in [t_f - \Delta t_w, t_f]$ for time extrapolation. We first identify the active edges to construct $\mathbf{j}_a$, perform DMD on the snapshots of $\mathbf{j}_a$ to get the predictions $\hat{\mathbf{j}}_a$, and then revert back to $\hat{\mathbf{j}}$. The DMD parameters are summarized in Table I. The t_{st}, t_{en}, i.e., the location of the DMD window for current density, is already determined from the on-the-fly DMD with $t_{st} = t_f - \Delta t_w$ and $t_{en} = t_f$. FFT is performed in $[t_{st}, t_{en}]$ to decide the DMD sampling interval Δ_t. As a rule of thumb, we choose DMD sampling frequency to be four times the Nyquist frequency.

The sharp decay in the singular values [Fig. 3(c)] indicates the existence of a low-dimensional structure in the plasma current dynamics. Figure 4 shows the first four most energetic DMD modes and corresponding DMD eigenvalues, as well as the spatial cross-correlation matrix. The DMD modes
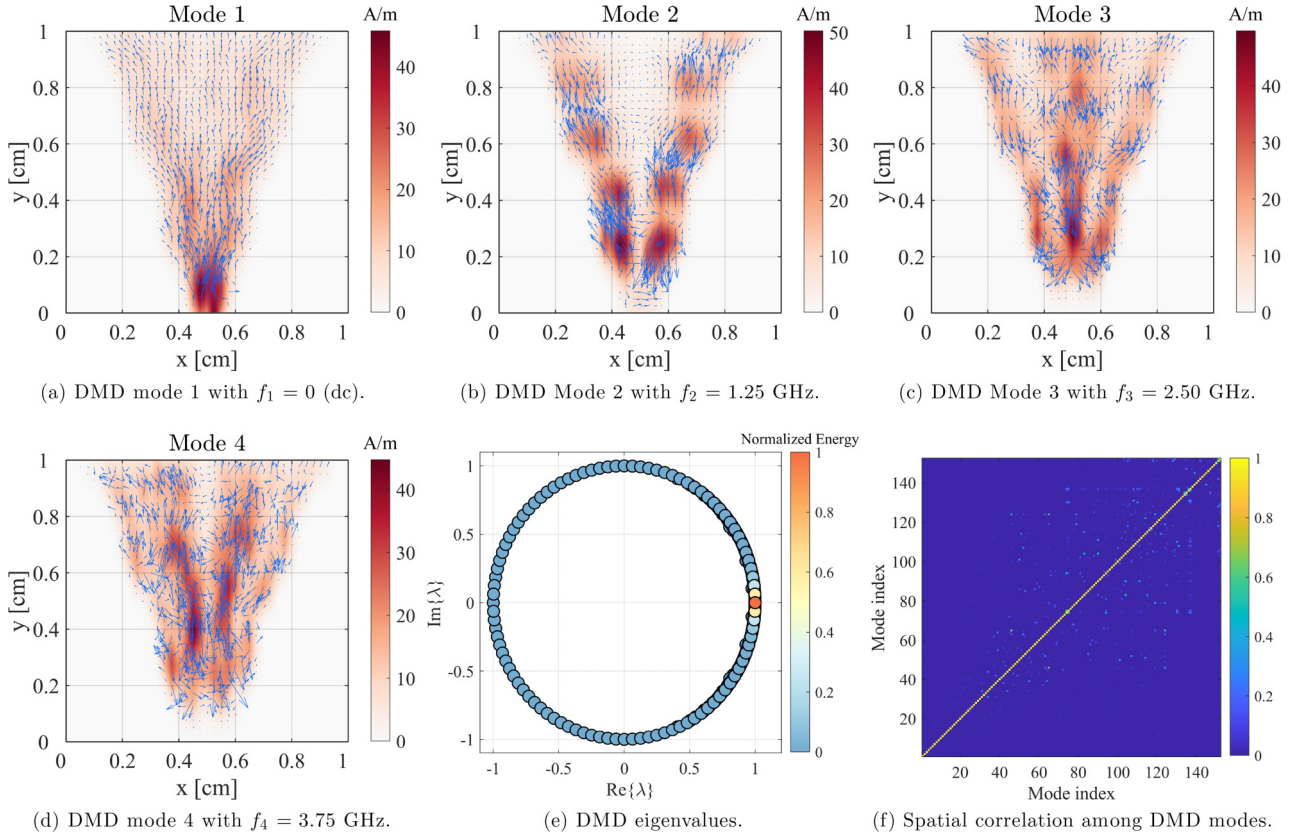

(a) DMD mode 1 with $f_1 = 0$ (dc).

(b) DMD Mode 2 with $f_2 = 1.25$ GHz.

(c) DMD Mode 3 with $f_3 = 2.50$ GHz.

(d) DMD mode 4 with $f_4 = 3.75$ GHz.

(e) DMD eigenvalues.

(f) Spatial correlation among DMD modes.

FIG. 4. Extracted DMD features for oscillating electron beam. (a)–(d) The DMD modes ($\boldsymbol{\phi}_m + \bar{\boldsymbol{\phi}}_m$) and their corresponding frequencies (f_m) for the current density modeling. (e) The DMD eigenvalues on the complex plane with respect to the unit circle. The DMD eigenvalues are color mapped according to their normalized energy. (f) Spatial correlation (ρ) among different DMD modes.

(a) EMPIC simulation of current density at $t = 64$ ns.

(b) DMD prediction of current density at $t = 64$ ns.

(c) Relative error between EMPIC and DMD predicted current density according to (**??**).
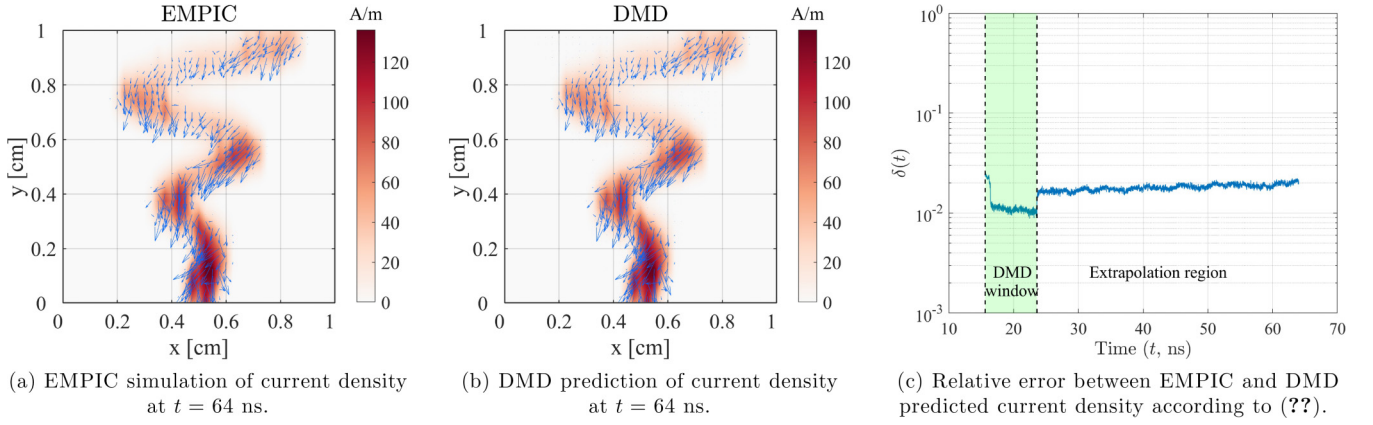
FIG. 5. Comparison between EMPIC and DMD-predicted current density. The shaded green region in (c) denotes the DMD training window. Note that the gap at the end of DMD window is due to the time-delayed stacking.

are indexed according to their energy [60], with mode 1 [Fig. 4(a)] being the most energetic one which is essentially a dc mode. Mode 2 [Fig. 4(b)] captures the oscillation frequency of the external magnetic flux with DMD frequency of $f_2 = 1.25$ GHz ($f_m = |\frac{\text{Im}\{\omega_m\}}{2\pi}|$). Mode 3 [Fig. 5(c)] indicates the first harmonic. Together these two modes capture $>90\%$ of the total energy [Fig. 4(e)]. As the mode index increases, the spatial pattern becomes less structured due to the effect of numerical noise. The frequencies associated with mode 3 [Fig. 5(c)] and mode 4 [Fig. 4(d)] indicate that those are essentially the harmonics of mode 2, generated due to the nonlinear wave-particle interaction. The correlation ρ between different spatial patterns of DMD modes indicates their extent of orthogonality. We use the absolute value of modal assurance criterion (MAC) [16,61] to compute the spatial correlation among DMD modes:

$$\rho(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j) = |\text{MAC}(\boldsymbol{\psi}_i, \boldsymbol{\psi}_j)| = |\frac{\left|\boldsymbol{\psi}_i^\text{T} \overline{\boldsymbol{\psi}}_j\right|^2}{(\boldsymbol{\psi}_i^\text{T} \boldsymbol{\psi}_i) \cdot (\boldsymbol{\psi}_j^\text{T} \boldsymbol{\psi}_j)}|. \quad (15)$$

We reiterate that by the mth DMD mode, we refer to the $\{\boldsymbol{\phi}_m, \overline{\boldsymbol{\phi}}_m\}$ pair in (13). While plotting the modes in Fig. 4 and throughout this paper, as well as calculating ρ, we use $\boldsymbol{\psi}_m = (\boldsymbol{\phi}_m + \overline{\boldsymbol{\phi}}_m) = 2\text{Re}\{\boldsymbol{\phi}_m\}$, where $\text{Re}\{\cdot\}$ represents the real

part. The spatial correlation matrix shows that the dominant DMD modes are orthogonal to each other with off-diagonal elements close to zero. However, it is important to note that unlike POD, DMD does not ensure orthogonality in space, but guarantee orthogonality in time.

The DMD predicted current density [Fig. 5(b)] deep into the prediction (extrapolation) region is plotted against the current density from high-fidelity EMPIC simulation [Fig. 5(a)] for side-to-side comparison. The relative error given by (16) is also plotted in Fig. 5(c),

$$\delta(t) = \frac{||\hat{\mathbf{j}}(t) - \mathbf{j}(t)||_2}{||\mathbf{j}(t)||_2}, \quad (16)$$

where $|| \cdot ||_2$ indicates the two-norm. The average relative error in the extrapolation region is 1.80%. The *gather, pusher,* and *scatter* stages are replaced by the DMD prediction $\hat{\mathbf{j}}$ for $t > t_f$ ($\equiv n > n_f$) in the EMPIC simulation, as illustrated in Fig. 2. The self-fields $\mathbf{e}$ and $\mathbf{b}$ generated beyond t_f are compared against the self-fields generated from EMPIC simulation. The relative error is calculated in a similar manner as in (16). The electric field patterns from EMPIC and DMD-EMPIC in the extrapolation region at $t = 64$ ns are shown in Figs. 6(a) and 6(b), showing good agreement. The relative errors in the self-electric and magnetic fields are 0.94% and
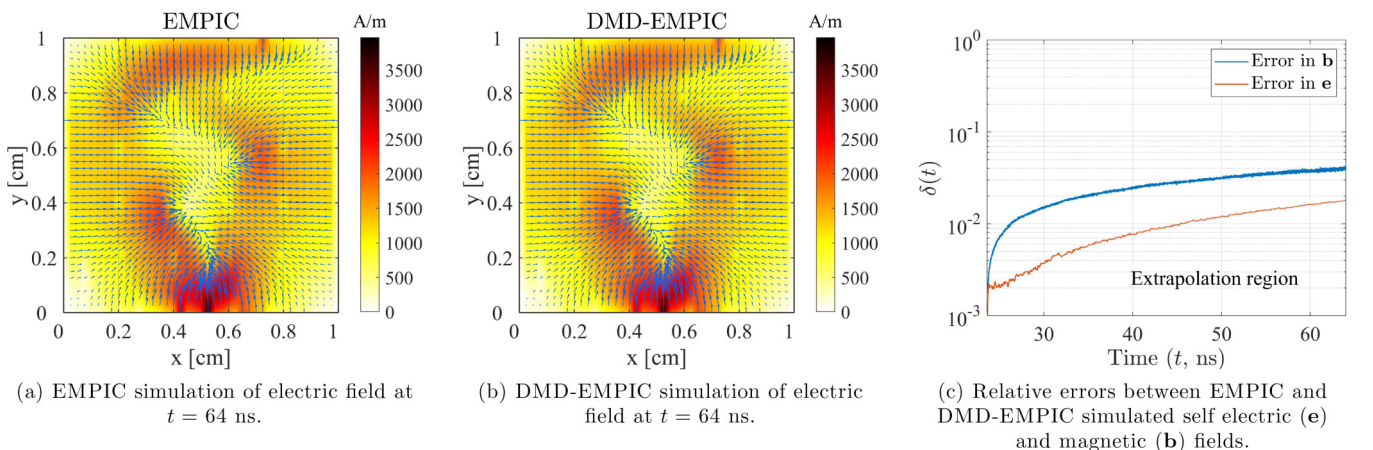

(a) EMPIC simulation of electric field at $t = 64$ ns.

(b) DMD-EMPIC simulation of electric field at $t = 64$ ns.

(c) Relative errors between EMPIC and DMD-EMPIC simulated self electric (**e**) and magnetic (**b**) fields.

FIG. 6. Comparison between EMPIC and DMD-EMPIC simulated self-fields in the extrapolation region.

(a) Virtual cathode snapshot at $t = 64$ ns.

(b) Current density snapshot at $t = 64$ ns.

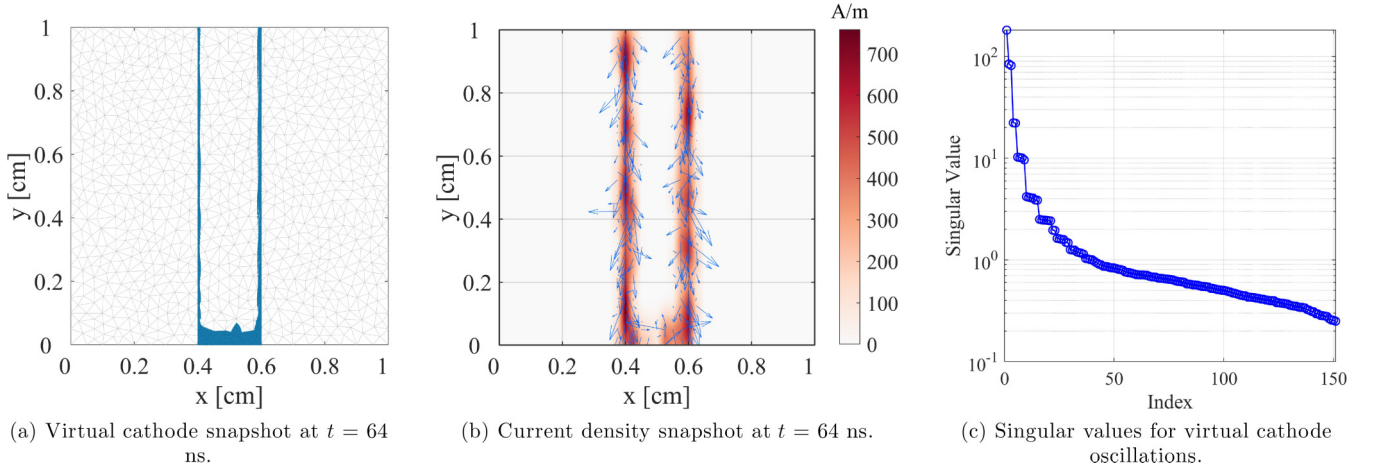(c) Singular values for virtual cathode oscillations.

FIG. 7. (a) Snapshot of the virtual cathode formation at $t = 64$ ns. (b) Snapshot of the current density at $t = 64$ ns. (c) The singular values after performing SVD on the snapshots of current density.

2.60%, respectively [Fig. 6(c)]. The gain in run time is discussed in Sec. V.

B. Virtual cathode oscillations

Next we consider a more challenging example of virtual cathode oscillation. We use the same setup as in Sec. IV A, however, with the following modifications. We increase the current injection by both increasing the superparticle ratio to $r_{sp} = 75\,000$ and the superparticle injection rate to 200 per time step. The superparticles are injected at the bottom in the region [0.4 cm, 0.6 cm]. Instead of a transverse oscillating magnetic flux, we apply a strong confining magnetic field, $\mathbf{B} = B_y \hat{y}$ along the y direction, with $B_y = 100$ T. The simulation is run until $n = 320\,000$ time steps or $t = 64$ ns with time step $\Delta t = 0.2$ ps. The snapshot of the beam after virtual cathode formation at $t = 16$ ns, and the corresponding current density plot are shown in Figs. 7(a) and 7(b), respectively. Modeling the current density for virtual cathode oscillations is particularly challenging because there are no external forces dictating a clear oscillation pattern of the electrons. The majority of oscillations are limited to a small region (near the bottom) causing possible rank deficiency, and the leakage

from the sides makes variation of $\mathbf{j}$ more prone to the particle noise.

1. On-the-fly DMD on fields

Similar to the oscillating beam case, the on-the-fly DMD is carried out to detect the end of transience indicating the onset of virtual cathode oscillations. We select $\Delta t_w = 8$ ns, $\Delta_t = 8$ ps, $\delta t_w = 0.4$ ns, $r = 200$, and number of Hankel stacks $d = 10$. The onset of the virtual cathode oscillations is detected at $t_f = 30.40$ ns. The on-the-fly DMD parameters are provided in detail in Table IV.

2. Offline DMD on current density

The training parameters for the Offline DMD is summarized in Table I. Fast decay of the singular values in Fig. 7(c) indicates that most of the energy is concentrated in a small number of modes. The first two most energetic DMD modes carry more than 95% of the total energy and are plotted in Fig. 8. Mode 1 [Fig. 8(a)] represents the dc leakage current from the sides of the virtual cathode, whereas mode 2 [Fig. 8(b)] captures the oscillations near the root of the
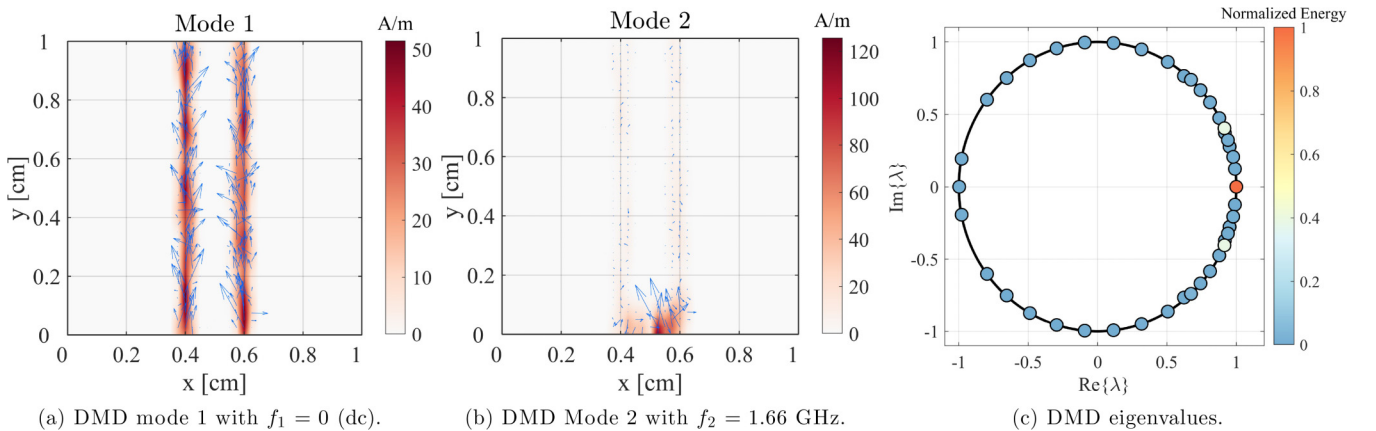

(a) DMD mode 1 with $f_1 = 0$ (dc).

(b) DMD Mode 2 with $f_2 = 1.66$ GHz.

(c) DMD eigenvalues.

FIG. 8. Extracted DMD modes and eigenvalues for current density modeling in virtual cathode oscillations.

(a) Relative error in DMD predicted current density.

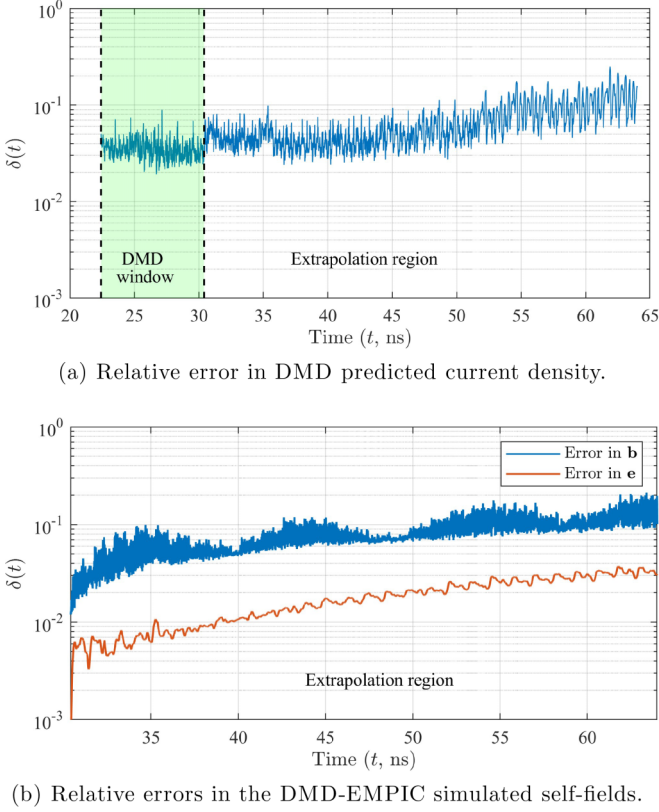(b) Relative errors in the DMD-EMPIC simulated self-fields.

FIG. 9. Relative errors for the virtual cathode oscillations.

beam, having a frequency of 1.65 GHz. The DMD eigenvalues [Fig. 8(c)] also indicate the dominance of the dc mode in terms of energy.

With the help of extracted DMD modes, frequencies, and modal amplitudes, current density is extrapolated for $t > t_f$ using DMD. The relative error in predicted current is shown in Fig. 9(a). The average error in the extrapolation region around 6.65%, which is higher compared to the oscillating beam case (1.80%). This is expected because the leakage from both sides of the virtual cathode results in nonsmooth current variation due to high-particle noise. Also, the localized nature of the oscillation contributes to a possible rank-deficiency resulting in higher error. The relative error in self-electric field **e** and magnetic flux **b** is shown in Fig. 9(b). The average relative error in **e** is around 1.81% whereas in **b** it is around 8.10%. The rror in **b** is typically higher for both test cases since the self-magnetic field is generally very low in magnitude, and more susceptible to particle and numerical noise.

C. Backward wave oscillator

Now we consider a more challenging and practical 2.5 D case of a backward wave oscillator (BWO) [62]. The finite element discretization of the longitudinal cross section of a sinusoidally corrugated slow-wave structure (SCSWS) is depicted Fig. 10(a). The SCSWS has boundary profile $R(z) = \frac{1}{2}(A+B) + \frac{1}{2}(A-B)\cos(\frac{2\pi}{C}z)$. Based on an eigenmode analysis, the SCSWS is designed to have $A = 19.5$ mm, $B = 10.5$ mm, $C = 16.7$ mm, and $N_{crg} = 8.5$. Each superparticle in the EMPIC model represents $r_{sp} = 1.25 \times 10^8$ electrons, with

an injection rate of 20 superparticles per time step ($\Delta_t = 0.5$ ps). The superparticles are injected in a random fashion with uniform distribution within the region centered around $y = 0.008$ m with beam width 0.0018 m. Superparticles are injected with a velocity of 2.5×10^8 m/s in the x direction. The simulation is run for 440,001 time steps or $t = 220.005$ ns. The fundamental frequency of the BWO is 8.31 GHz.

The on-the-fly algorithm detects the end of transience at $t_f = 107.20$ ns with $\Delta t_w = 16$ ns, $\Delta_t = 2$ ps, and $\delta t_w = 1.6$ ns (see Table IV for details). The training parameters for offline DMD on current density are provided in Table I. The average extrapolation error in the current density is 13.52% (Fig. 11). The higher error can be attributed to particle noise, which plays a significant role in the absence of a strong external force dictating the oscillations (oscillating beam). The most energetic mode is the dc mode [Fig. 10(b)] followed by the mode [Fig. 10(c)] oscillating with fundamental frequency at $f_2 = 8.32$ GHz. As expected, the oscillations are concentrated towards the end of BWO where the bunching of electrons (superparticles) occurs. The error in the self-electric field from the DMD-EMPIC framework is 5.58% (Fig. 11).

V. COMPUTATIONAL GAIN

A. Computational complexity

The time complexity of typical EMPIC algorithm for each time-step with *explicit* field solver is given by $O(N + N_p)$, where N is the aggregate mesh dimension and N_p is the total number of (super)particles. In a traditional EMPIC setting $N_p \gg N$, thus the number of particles remains the primary bottleneck for expediting EMPIC simulations. If the simulation is run until n_q time-steps, the total time complexity is given by $O(Nn_q + N_pn_q)$.

The computation cost of repeated calculation of the DMD features for the on-the-fly application also adds to the typical EMPIC cost for $n \leqslant n_f$ ($\equiv t \leqslant t_f$) in the DMD-EMPIC algorithm. The computation complexity of DMD is dominated by the SVD step, resulting in a time complexity of $O(l_d^2 Nd)$ [63], where l_d is the number of DMD snapshots after d number of Hankel stacking. Time complexity for randomized DMD with target rank $r(< l_d)$, is typically $O(rl_dNd)$. Note that as mentioned in Ref. [64], SVD need not be recalculated every time we shift the DMD window. The resulting DMD features can be calculated in an incremental manner. In the worst case scenario, let us consider that the SVD is performed for each sliding window. Also, in the worst case scenario, the DMD window is shifted by only one snapshot, i.e., the amount of shift in terms of time steps $\Delta_{ns} = \Delta_n$. With these considerations, the cost of the sliding-window on-the-fly DMD (randomized) is approximately $O(\frac{rl_dNdn_f}{\Delta_n})$, where n_f is the time step at which EMPIC is stopped, and Δ_n is the number of time steps between two consecutive DMD snapshots.

The time complexity of the DMD-EMPIC algorithm for $n < n_f$ ($\equiv t < t_f$) is $O(Nn_f + N_pn_f + \frac{rl_dNdn_f}{\Delta_n})$, whereas for $n > n_f$ ($\equiv t > t_f$) is $\mathcal{O}(N(n_q - n_f) + l_d^2Nd)$ considering one-time cost of offline DMD with d Hankel stacks. Since there are no particles involved in case of DMD-EMPIC for $n > n_f$, N_p does not appear in the time complexity. For a simulation run until $n = n_q$, the time-complexity of the DMD-EMPIC is

(a) Snapshot of electron beam inside the BWO at $t = 140$ ns.

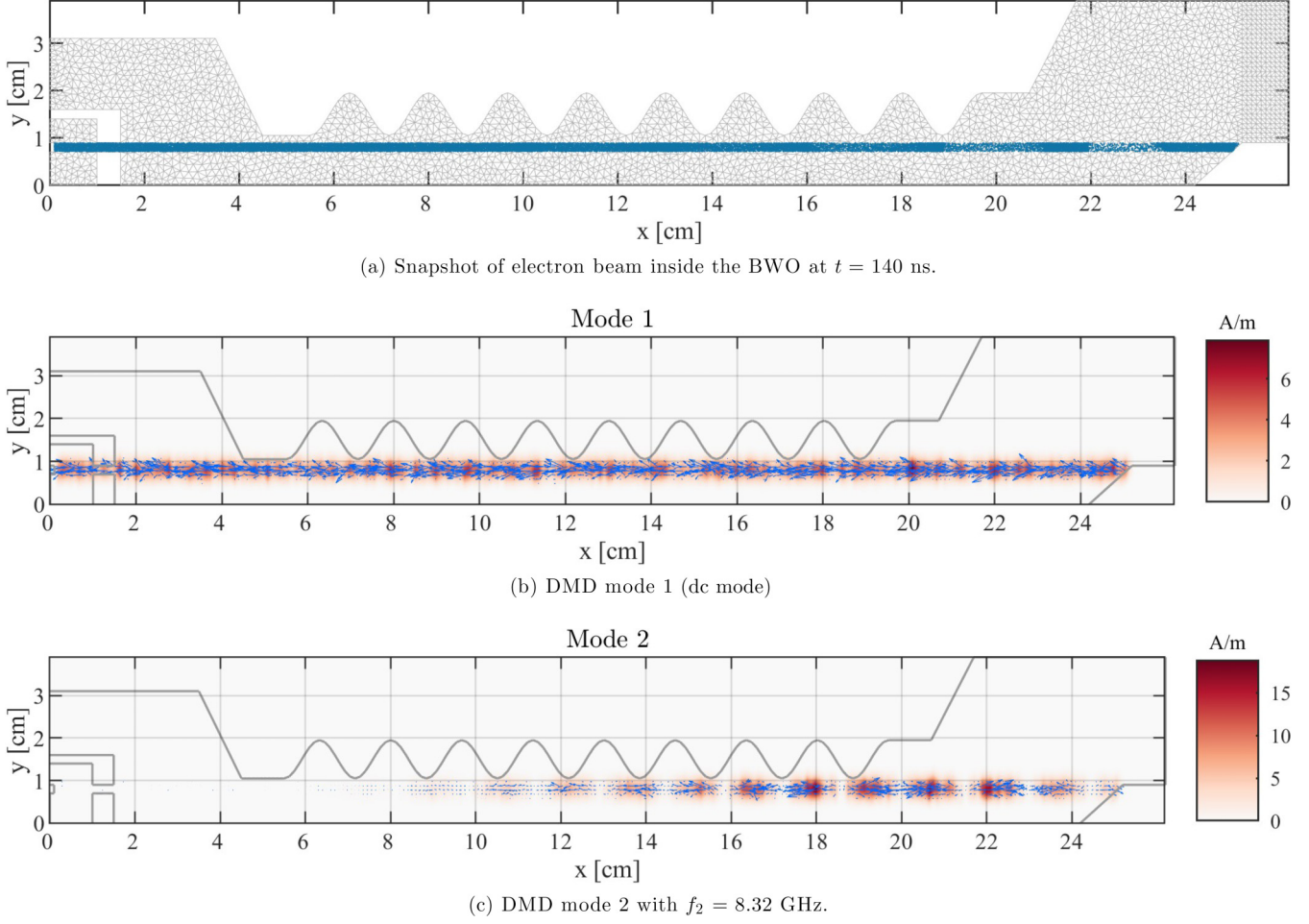(b) DMD mode 1 (dc mode)

(c) DMD mode 2 with $f_2 = 8.32$ GHz.

FIG. 10. Offline DMD on current density of a backward wave oscillator (BWO).

$O(Nn_q + N_p n_f + \frac{rl_d Nd n_f}{\Delta_n} + l_d^2 Nd)$. The time complexities for EMPIC and DMD-EMPIC are summarized in Table II.

B. Runtime comparison

As mentioned earlier, in a typical EMPIC setting, $N_p \gg N$. As long as n_q is moderately larger than n_f, there will be sig-

FIG. 11. Relative two-norm error in current density **j** (DMD extrapolation) and self electric field **e** (DMD-EMPIC framework) for the backward wave oscillator.

nificant gain in the run time. For late-time queries ($n_q \gg n_f$), the ratio of run times for EMPIC (T_q) and DMD-EMPIC ($\widehat{T}_q$) can be roughly given by $\widehat{T}_q/T_q \approx n_f/n_q$, given the field solver takes negligible time compared to the entire simulation. Note that for large scale problems such as BWO, this approximation does not necessarily hold, as evident from Table III.

The simulation run time depends on several factors, including the specific computational platform and hardware, and code optimization. In this work, the numerical experiments were run on Intel Xeon E5-2680 v4 (Broadwell) compute CPU with 2.4 GHz and 14 cores per processor. Each node had 128 GB of memory. The interconnect used was Mellanox EDR Infiniband Networking (100Gbps). Each simulation job was allocated one node and five cores. The node run times for all test cases are listed in Table III. Note that the CPU run time
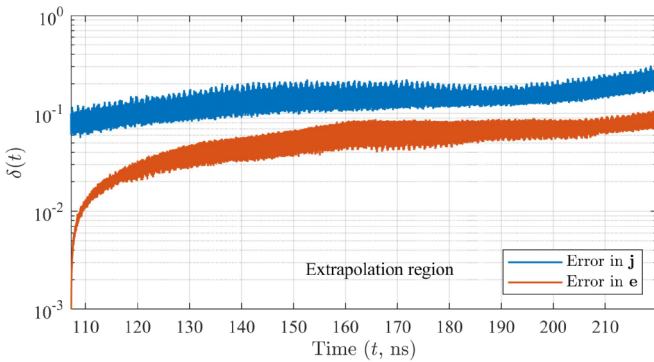
TABLE II. EMPIC and DMD-EMPIC complexities with explicit field solver.

EMPIC	$O(Nn_q + N_p n_q)$
DMD-EMPIC	$O(Nn_q + N_p n_f + \frac{rl_d Nd n_f}{\Delta_n} + l_d^2 Nd)$

TABLE III. Node run time in days.

	EMPIC	DMD-EMPIC
Oscillating beam	4.43	1.66
Vircator	5.96	2.93
BWO	7.04	4.08

is approximately 5 times the node run time, exhibiting good shared-memory parallelization across cores.

C. Effect of parallelization

It is important to note that particle-in-cell (PIC) algorithms are highly parallelizable and an appropriate parallel computing architecture can be employed to accelerate EMPIC simulations [65]. Single nodes with multiple CPUs (shared memory) or multiple nodes (each with one or more CPUs and distributed memory) can reduce the simulation time from days to several hours. Fortunately, the DMD-EMPIC algorithm should also achieve comparable acceleration from parallelization for the following reason: let the computation time for performing DMD (online + offline) be T_{DMD}. Let the run time for the original EMPIC simulation up to the desired query time be T_q. Suppose the on-the-fly (online) DMD raises the flag to stop the EMPIC simulation at $T_f (< T_q)$. The relative gain in computation time G_T is

$$G_T = \frac{T_q}{T_f + T_{\text{DMD}} + T_{\text{FS}}} \approx \frac{T_q}{T_f}, \qquad (17)$$

where T_{FS} is the time taken by the field solver beyond T_f. This approximation holds if T_{DMD} and T_{FS} are negligible compared to T_f. Depending on the scale of the problem, type of the field solver, and parallelization, T_{FS} can be negligible compared to T_f. T_{DMD} is usually much less than T_f even with parallelization, because parallelization not only helps accelerate EMPIC but also the DMD computation. For example, the authors in Ref. [66] utilized a parallel Tall and Skinny QR (TSQR) algorithm for parallelizing the SVD computation. The construction of the Koopman operator, eigendecomposition, and the construction of DMD modes are also done in an embarrassingly parallel fashion. Recently, the authors in Ref. [67] used a communication-optimal parallel TSQR algorithm for reduced-communication parallel DMD. These parallel DMD methods are reported to scale well. A detailed study of how parallel EMPIC scales compared to parallel DMD in the presence of parallel computing architecture is beyond the scope of this paper. Since both DMD and EMPIC can benefit from parallel computation, the overall acceleration by DMD-EMPIC depends on how early the system reaches equilibrium or pseudoequilibrium, i.e., how small T_f is compared to T_q. Consequently, Eq. (17) is independent of the hardware platform or the type of PIC algorithms used, as both T_f and T_q are similarly affected.

VI. CONCLUSION AND LOOK AHEAD

In this paper, we have demonstrated a reduced-order framework for modeling space-charge dynamics from EMPIC simulations by performing DMD on the current density. Extracted features such as DMD spatial modes and frequencies help to discover and analyze the dominant physics. DMD-EMPIC shows great promise in reducing overall run time of EMPIC simulations via time extrapolation of the current density data to future time. However, similar to any other data-driven method, the extrapolation accuracy depends on the quality of the training data. While our ultimate goal is to extrapolate from very early (transient) time-series data, current time-extrapolation methods like DMD and recurrent neural networks require further development to realize this fully. In the meantime, a more viable and effective approach is to interpolate time-series data of current density across various simulation parameters. This technique, rooted in reduced-order modeling of current density, offers a promising solution to the longstanding challenge of accelerating EMPIC plasma simulations. Another important aspect is ensuring conservation laws in the extrapolated evolution. Currently, the DMD-EMPIC algorithm does not enforce energy conservation explicitly but only approximately as the predicted current density and fields are close to the original EMPIC quantities. As a future work, physical constraints such as conservation laws may be explicitly included in the training and extrapolation process. Another possible venue of future work is to make the DMD parameter selection fully automatic.

The code and data for the oscillating electron beam case are available from GitHub [68].

APPENDIX: ON-THE-FLY DMD ALGORITHM

The on-the-fly DMD algorithm [29] for identifying the end of transient is based on the invariance of the DMD extracted features irrespective of the location of the DMD window in equilibrium, as long as the DMD window is wide enough to capture the dynamics. Invariance in DMD extracted features leads to invariance in the DMD predicted solution at a fixed time window. However, when the system is in transience, no such claim regarding the invariance of extracted DMD features can be made. It is important to note that the on-the-fly DMD necessitates frequent recalculations of DMD algorithm, which can become computationally intensive for large data sets. To address this challenge, we employ the randomized DMD algorithm as proposed in Ref. [69]. While this approach offers greater efficiency, it does so at the cost of some accuracy. However, this trade-off is acceptable in our context, as the primary focus during the on-the-fly stage is to capture the system's overall behavior rather than precise reconstruction.

The hyperparameters in Algorithm 1 include δn_w, η, q, and δ_0. The shift between consecutive DMD windows, denoted by

ALGORITHM 1. On-the-fly sliding-window DMD algorithm for detecting t_f for real-time termination of EMPIC.

Input: (i) DMD window width Δt_w or, equivalently, Δn_w based on the approximate idea of the timescale of the problem.
(ii) Electric field data from the ongoing EMPIC simulation.
Output: EMPIC termination flag.
1: After the EMPIC simulation reaches $n = \Delta n_w$, shift the DMD window by δn_w time steps (represented by increasing window index k) as the simulation progresses. Let us denote the starting and end of the kth window by $t_{st,k}(\equiv n_{st,k})$ and $t_{en,k}(\equiv n_{en,k})$, respectively.
2: **for** Current (kth) DMD window
3: Perform randomized DMD for the kth window and obtain the DMD prediction from $n_k = (n_{en,k} + \eta \Delta n_w)$ to $(n_k + \Delta n_w)$. Let us denote this DMD prediction corresponding to the kth DMD window as $\widehat{\mathbf{x}}_k^{(n)}$.
4: Get the average relative two-norm error between the overlapping region of $\widehat{\mathbf{x}}_k^{(n)}$ and $\widehat{\mathbf{x}}_{k-1}^{(n)}$ denoted by $\langle \delta^{(n)} \rangle_w$,

$$\langle \delta^{(n)} \rangle_k = \frac{1}{(n_k - \delta n_w + 1)} \sum_{n=n_k}^{n_k + \Delta n_w - \delta n_w} \frac{||\widehat{\mathbf{x}}_w^{(n)} - \widehat{\mathbf{x}}_{w-1}^{(n)}||_2}{||\widehat{\mathbf{x}}_{w-1}^{(n)}||_2} \qquad (A1)$$

5: **if** $(\langle \delta^{(n)} \rangle_k + \langle \delta^{(n)} \rangle_{k-1} + \ldots + \langle \delta^{(n)} \rangle_{k-q+1})/q < \delta_0$ **then** Raise the EMPIC termination flag.
6: Raise the EMPIC termination flag.
7: **return**
8: **else**
9: Move to the next DMD window, $w = w + 1$.
10: **enf if**
11: **end for**

TABLE IV. On-the-fly DMD parameters.

Parameters	Oscillating beam	Vircator	BWO
Δt_w	8 ns	8 ns	16 ns
δt_w	0.4 ns	0.4 ns	1.6 ns
η	10	10	10
q	5	5	5
δ_0	0.05	0.05	0.05
r	200	200	200
Δ_t	8 ps	8 ps	2 ps
d	10	10	5

δn_w (equivalently, $\delta t_w = \delta n_w \Delta t$), balances precision and computational cost. Specifically, a smaller δn_w enhances the accuracy in identifying t_f but increases the number of required DMD computations. Conversely, a larger δn_w reduces computation at the expense of precision. For large-scale problems, a larger δn_w is advisable. The factor η determines the extent of time for comparing DMD predictions, reflecting our prediction goals for equilibrium behavior. The decision to halt the EMPIC simulation is based on whether the relative error between DMD predictions remains below δ_0 across q consecutive windows, indicating the end of the transient state. The choice of δ_0 reflects the desired precision in DMD extrapolation. Opting for a higher value of q increases robustness against noise in data, but may delay the detection of t_f. The hyperparameters for Algorithm 1, including parameters for randomized DMD [69] (r, Δ_t, d), are provided in Table IV. Note that r for randomized DMD corresponds to the target rank of randomized SVD operation.

[1] S. H. Gold and G. S. Nusinovich, Review of high-power microwave source research, Rev. Sci. Instrum. **68**, 3945 (1997).

[2] J. H. Booske, Plasma physics and related challenges of millimeter-wave-to-terahertz and high power microwave generation, Phys. Plasmas **15**, 055502 (2008).

[3] J. Benford, J. A. Swegle, and E. Schamiloglu, *High Power Microwaves* (CRC Press, New York, 2015).

[4] M. Wan, W. H. Matthaeus, V. Roytershteyn, H. Karimabadi, T. Parashar, P. Wu, and M. Shay, Intermittent dissipation and heating in 3D kinetic plasma turbulence, Phys. Rev. Lett. **114**, 175002 (2015).

[5] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (CRC Press, New York, 2004).

[6] K. Eppley, The use of electromagnetic particle-in-cell codes in accelerator applications, Technical Report No: SLAC-PUB-4812; CONF-881049-46; ON: DE89004968, Stanford Linear Accelerator Center, Menlo Park, CA, 1988.

[7] R. Lehe, M. Kirchen, B. B. Godfrey, A. R. Maier, and J.-L. Vay, Elimination of numerical Cherenkov instability in flowing-plasma particle-in-cell simulations by using Galilean coordinates, Phys. Rev. E **94**, 053305 (2016).

[8] F. Cruz, T. Grismayer, and L. O. Silva, Kinetic instability in inductively oscillatory plasma equilibrium, Phys. Rev. E **103**, L051201 (2021).

[9] A. Davidson, A. Tableman, W. An, F. Tsung, W. Lu, J. Vieira, R. Fonseca, L. Silva, and W. Mori, Implementation of a hybrid particle code with a PIC description in $r-z$ and a gridless description in ϕ into OSIRIS, J. Comput. Phys. **281**, 1063 (2015).

[10] O. Shapoval, R. Lehe, M. Thévenet, E. Zoni, Y. Zhao, and J.-L. Vay, Overcoming timestep limitations in boosted-frame particle-in-cell simulations of plasma-based acceleration, Phys. Rev. E **104**, 055311 (2021).

[11] M. Pandya, Low edge safety factor disruptions in the compact toroidal hybrid: Operation in the low-Q regime, passive disruption avoidance and the nature of MHD precursors, Ph.D. thesis, Auburn University, 2016.

[12] B. P. Van Milligen, E. Sánchez, A. Alonso, M. Pedrosa, C. Hidalgo, A. M. De Aguilera, and A. L. Fraguas, The use of the biorthogonal decomposition for the identification of zonal flows at TJ-II, Plasma Phys. Controlled Fusion **57**, 025005 (2015).

[13] P. J. Byrne, Study of external kink modes in shaped HBT-EP plasmas, Ph.D. thesis, Columbia University, 2017.

[14] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, Physics-constrained, low-dimensional models for magnetohydrodynamics: First-principles and data-driven approaches, Phys. Rev. E **104**, 015206 (2021).

[15] M. Sasaki, Y. Kawachi, R. Dendy, H. Arakawa, N. Kasuya, F. Kin, K. Yamasaki, and S. Inagaki, Using dynamical mode

decomposition to extract the limit cycle dynamics of modulated turbulence in a plasma simulation, Plasma Phys. Controlled Fusion **61**, 112001 (2019).

[16] I. Nayak, M. Kumar, and F. L. Teixeira, Detection and prediction of equilibrium states in kinetic plasma simulations via mode tracking using reduced-order dynamic mode decomposition, J. Comput. Phys. **447**, 110671 (2021).

[17] J. L. Nicolini, D. Na, and F. L. Teixeira, Model order reduction of electromagnetic particle-in-cell kinetic plasma simulations via proper orthogonal decomposition, IEEE Trans. Plasma Sci. **47**, 5239 (2019).

[18] I. Nayak and F. L. Teixeira, Dynamic mode decomposition reduced-order models for multiscale kinetic plasma analysis, in *2021 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)* (IEEE, Boulder, Colorado, 2021), pp. 261–262.

[19] I. Nayak and F. L. Teixeira, Dynamic mode decomposition for prediction of kinetic plasma behavior, in *2020 International Applied Computational Electromagnetics Society Symposium (ACES)* (IEEE, Monterey, CA, 2020), pp. 1–2.

[20] J. Hesthaven, C. Pagliantini, and N. Ripamonti, Adaptive symplectic model order reduction of parametric particle-based Vlasov–Poisson equation, Math. Comput. **93**, 1153 (2024).

[21] M. Kraus, K. Kormann, P. J. Morrison, and E. Sonnendrücker, GEMPIC: Geometric electromagnetic particle-in-cell methods, J. Plasma Phys. **83**, 905830401 (2017).

[22] E. P. Alves and F. Fiuza, Data-driven discovery of reduced plasma physics models from fully kinetic simulations, Phys. Rev. Res. **4**, 033192 (2022).

[23] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech. **656**, 5 (2010).

[24] R. Taylor, J. N. Kutz, K. Morgan, and B. A. Nelson, Dynamic mode decomposition for plasma diagnostics and validation, Rev. Sci. Instr. **89**, 053501 (2018).

[25] A. A. Kaptanoglu, K. D. Morgan, C. J. Hansen, and S. L. Brunton, Characterizing magnetized plasmas with dynamic mode decomposition, Phys. Plasmas **27**, 032108 (2020).

[26] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, Spectral analysis of nonlinear flows, J. Fluid Mech. **641**, 115 (2009).

[27] I. Nayak, F. L. Teixeira, and M. Kumar, Koopman autoencoder architecture for current density modeling in kinetic plasma simulations, in *2021 International Applied Computational Electromagnetics Society Symposium (ACES)* (IEEE, 2021), pp. 1–3.

[28] I. Nayak, M. Kumar, and F. L. Teixeira, Koopman autoencoders for reduced-order modeling of kinetic plasmas, in *Advances in Electromagnetics Empowered by Artificial Intelligence and Deep Learning* (Wiley Online Library, 2023), pp. 515–542.

[29] I. Nayak, F. L. Teixeira, and R. J. Burkholder, On-the-fly dynamic mode decomposition for rapid time-extrapolation and analysis of cavity resonances, IEEE Trans. Antennas Propag. **72**, 131 (2024).

[30] H. Moon, F. L. Teixeira, and Y. A. Omelchenko, Exact charge-conserving scatter-gather algorithm for particle-in-cell simulations on unstructured grids: A geometric perspective, Comput. Phys. Commun. **194**, 43 (2015).

[31] J. Kim and F. L. Teixeira, Parallel and explicit finite-element time-domain method for Maxwell's equations, IEEE Trans. Antennas Propag. **59**, 2350 (2011).

[32] E. Evstatiev and B. Shadwick, Variational formulation of particle algorithms for kinetic plasma simulations, J. Comput. Phys. **245**, 376 (2013).

[33] J. W. Burby, Finite-dimensional collisionless kinetic theory, Phys. Plasmas **24**, 032101 (2017).

[34] X. Jianyuan, Q. Hong, and L. Jian, Structure-preserving geometric particle-in-cell methods for Vlasov-Maxwell systems, Plasma Sci. Technol. **20**, 110501 (2018).

[35] D.-Y. Na, H. Moon, Y. A. Omelchenko, and F. L. Teixeira, Local, explicit, and charge-conserving electromagnetic particle-in-cell algorithm on unstructured grids, IEEE Trans. Plasma Sci. **44**, 1353 (2016).

[36] H. Flanders, *Differential Forms with Applications to the Physical Sciences* (Dover, New York, 1989).

[37] F. L. Teixeira and W. Chew, Lattice electromagnetic theory from a topological viewpoint, J. Math. Phys. **40**, 169 (1999).

[38] P. W. Gross and P. R. Kotiuga, *Electromagnetic Theory and Computation: A Topological Approach* (Cambridge University Press, Cambridge, 2004).

[39] B. He and F. L. Teixeira, Differential forms, Galerkin duality, and sparse inverse approximations in finite element solutions of Maxwell equations, IEEE Trans. Antennas Propag. **55**, 1359 (2007).

[40] G. A. Deschamps, Electromagnetics and differential forms, Proc. IEEE **69**, 676 (1981).

[41] F. L. Teixeira, Differential forms in lattice field theories: An overview, ISRN Math. Phys. **2013**, 487270 (2013).

[42] F. L. Teixeira, Lattice Maxwell's equations, Prog. Electromagn. Res. **148**, 113 (2014).

[43] This is a good approximation for small Δt.

[44] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: Theory and applications, J. Comput. Dyn. **1**, 391 (2014).

[45] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems* (SIAM, 2016).

[46] G. Kerschen and J.-C. Golinval, Physical interpretation of the proper orthogonal modes using the singular value decomposition, J. Sound Vib. **249**, 849 (2002).

[47] M. Gavish and D. L. Donoho, The optimal hard threshold for singular values is $4/\sqrt{3}$, IEEE Trans. Inf. Theory **60**, 5040 (2014).

[48] M. Gavish and D. L. Donoho, Matlab code: The optimal hard threshold for singular values is $4/\sqrt{3}$, https://exhibits.stanford.edu/data/catalog/vg705qn9070.

[49] M. R. Jovanović, P. J. Schmid, and J. W. Nichols, Sparsity-promoting dynamic mode decomposition, Phys. Fluids **26**, 024103 (2014).

[50] S. Pan and K. Duraisamy, On the structure of time-delay embedding in linear models of non-linear dynamical systems, Chaos **30**, 073135 (2020).

[51] B. O. Koopman, Hamiltonian systems and transformation in hilbert space, Proc. Natl. Acad. Sci. USA **17**, 315 (1931).

[52] C. W. Curtis and D. J. Alford-Lago, Dynamic-mode decomposition and optimal prediction, Phys. Rev. E **103**, 012201 (2021).

[53] D. Dylewsky, M. Tao, and J. N. Kutz, Dynamic mode decomposition for multiscale nonlinear physics, Phys. Rev. E **99**, 063311 (2019).

[54] M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, Time-delay observables for Koopman: Theory and applications, SIAM J. Appl. Dyn. Syst. **19**, 886 (2020).

[55] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Chaos as an intermittently forced linear system, Nat. Commun. **8**, 19 (2017).

[56] F. Takens, Detecting strange attractors in turbulence, in *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a symposium held at the University of Warwick 1979/80, Coventry, England* (Springer, Berlin, Heidelberg, 2006), pp. 366–381.

[57] Note that the contemporary EMPIC algorithms such as geometric electromagnetic particle-*n*-cell algorithm or GEM-PIC [21] and conformal EMPIC algorithms [58] involves the grid-to-particle and particle-to-grid interpolation stages along with the particle pusher stage. Our DMD-EMPIC framework can be applied to these algorithms as well as replacing the particle stages with the DMD model of current density.

[58] Y. Wang, J. Wang, Z. Chen, G. Cheng, and P. Wang, Three-dimensional simple conformal symplectic particle-in-cell methods for simulations of high power microwave devices, Comput. Phys. Commun. **205**, 1 (2016).

[59] V. Martínez-Cagigal, Custom color map (2023), MATLAB Central File Exchange. Retrieved February 22, 2023.

[60] The energy of a DMD mode is calculated by averaging the 2-norm square of that particular DMD mode over all the time-samples inside the DMD training window.

[61] C. Beaverstock, M. Friswell, S. Adhikari, T. Richardson, and J. Du Bois, Automatic mode tracking for flight dynamic analysis using a spanning algorithm, Aerosp. Sci. Technol. **47**, 54 (2015).

[62] D.-Y. Na, Y. A. Omelchenko, H. Moon, B.-H. V. Borges, and F. L. Teixeira, Axisymmetric charge-conservative electromagnetic particle simulation algorithm on unstructured grids: Application to microwave vacuum electronic devices, J. Comput. Phys. **346**, 295 (2017).

[63] Note that there is a typo in section "Computational complexity" of Ref. [16]. The time complexity of truncated SVD of N by l matrix should be $O(l^2 N)$, not $O(lN^2)$.

[64] M. S. Hemati, M. O. Williams, and C. W. Rowley, Dynamic mode decomposition for large and streaming datasets, Phys. Fluids **26**, 111701 (2014).

[65] V. K. Decyk and T. V. Singh, Particle-in-cell algorithms for emerging computer architectures, Comput. Phys. Commun. **185**, 708 (2014).

[66] T. Sayadi and P. J. Schmid, Parallel data-driven decomposition algorithm for large-scale datasets: With application to transitional boundary layers, Theor. Comput. Fluid Dyn. **30**, 415 (2016).

[67] K. R. Maryada and S. E. Norris, Reduced-communication parallel dynamic mode decomposition, J. Comput. Sci. **61**, 101599 (2022).

[68] https://github.com/nayak77/DMD-EMPIC/tree/master.

[69] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton, Randomized dynamic mode decomposition, SIAM J. Appl. Dyn. Syst. **18**, 1867 (2019).