

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository: <https://orca.cardiff.ac.uk/id/eprint/190144/>

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Keskin, Esila, Yang, Xintong , Wei, Minglun and Ji, Ze 2026. Event-based physics parameter inference from GPU-accelerated synthetic simulations using convolutional neural networks. Presented at: 2026 31st International Conference on Automation and Computing (ICAC), High Wycombe, UK, 26 - 28 August 2026. 2026 31st International Conference on Automation and Computing (ICAC). IEEE, 10.1109/icac70325.2026.11712415

Publishers page: <https://doi.org/10.1109/icac70325.2026.11712415>

Please note:

Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See <http://orca.cf.ac.uk/policies.html> for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Event-Based Physics Parameter Inference from GPU-Accelerated Synthetic Simulations Using Convolutional Neural Networks

Esila Keskin

School of Computing and Creative Technologies
University of the West of England
Bristol, UK
esila2.keskin@live.uwe.ac.uk

Xintong Yang, Minglun Wei

School of Engineering
Cardiff University
Cardiff, UK
{yangx66, weim9}@cardiff.ac.uk

Ze Ji

College of Mechanical and
Electrical Engineering
Hohai University, China
z.ji@hhu.edu.cn

Abstract—Identifying physical properties of objects from visual observations is a fundamental challenge in robotics. This paper presents a GPU-accelerated synthetic physics simulation pipeline using NVIDIA Warp, generating large-scale labelled datasets with domain-randomised physics parameters including gravity, restitution, and Coulomb friction. We investigate four neural network architectures for physics parameter inference from visual input: a single-frame Convolutional Neural Network (CNN), a two-frame CNN, a single event-frame CNN inspired by neuromorphic sensing, and a 5-frame sequence event CNN. Our primary contribution is demonstrating that environment-level physics parameters can be inferred directly from visual observations using a temporal event-based representation. Event-based encoding of frame differences outperforms raw frame input while using significantly fewer input channels, and temporal context across multiple frames further improves inference. We evaluate noise robustness, RGB+Event fusion, dedicated single-parameter inference heads, recurrent and structural temporal baselines, differentiable simulation, and an extended rolling friction setting with angular velocity prediction. Results confirm that sparse temporal motion signals are more informative than static or short-range RGB input for physics parameter inference across three datasets of increasing physical complexity.

Index Terms—event-based vision, physics parameter estimation, system identification, physics simulation, differentiable simulation

I. INTRODUCTION

Understanding the physical properties of objects from visual observations is essential for robotic interaction with the real world [1]. Physics engines provide a scalable way to generate synthetic training data with ground-truth labels that are difficult or impossible to obtain in real-world settings. Closing the sim-to-real gap requires identifying not only the structure of a physics model, but the specific parameter values that govern it. Inferring properties such as gravity, friction, and restitution from image sequences has direct applications in robotic manipulation, where identifying the physical model governing an object is a prerequisite for accurate interaction planning [2], [3].

A key open question is whether neural networks can identify not just object trajectories, but the underlying physics pa-

rameters that govern them, directly from image observations. Recovering parameters such as gravity, friction, and restitution from visual input alone would enable robots to estimate physical properties of the environment in real time and support more accurate manipulation planning. A secondary question concerns which forms of visual information are most informative for this task. Raw RGB frames encode appearance; event-based encodings retain only pixel-level brightness changes, mimicking neuromorphic sensors. Understanding which representation best supports physics parameter inference has direct implications for sensor selection and data pipeline design in robotic perception systems.

A key motivation for adopting event-based representations is that environment-level physics parameters, such as gravity and friction, do not manifest in any single image frame but rather through the temporal character of motion accumulated across frames. Event-based encoding, which computes pixel-wise frame differences and thresholds them into sparse binary maps, discards static background appearance and retains only the motion-relevant signal. This is particularly well-suited to physics inference, where what matters is how objects are moving, not what they look like. Accordingly, this paper makes the following contributions:

- A comparative evaluation of four CNN architectures of increasing temporal capacity, from raw RGB frames to event-based encoding to multi-frame sequence modelling for physics parameter inference.
- A GPU-parallel 2D physics simulator with domain-randomised gravity, restitution, Coulomb friction, and rolling friction, implemented in NVIDIA Warp.
- A synthetic labelled dataset generator producing RGB frames, segmentation masks, bounding boxes, velocity labels, and environment-level physics annotations across three datasets of increasing complexity.
- Empirical analysis of noise robustness, RGB+Event fusion, dedicated single-parameter inference heads, a comparison against recurrent and structural temporal baselines and differentiable simulation, and an extension to rolling friction with angular velocity prediction.

Minglun Wei was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Doctoral Training Partnership (No. EP/W524682/1). This work was also partially supported by the UK EPSRC grant No. EP/X018962/1. Corresponding author: Ze Ji.

II. RELATED WORK

Physics simulation has become the centre of robot learning research. Todorov et al. [4] introduced MuJoCo, which has been widely used for contact-rich manipulation tasks. PyBullet [5] offers an open-source alternative with broad adoption in the community. More recently, differentiable physics engines such as Brax [6] enable gradient-based optimisation through simulation, opening new directions for physics-aware learning [7]. Yang et al. [8] used differentiable physics to identify elastoplastic material properties from depth observations in robotic manipulation. Similar simulation frameworks have also been applied to granular materials [9]–[11].

The idea of inferring physical properties from visual observations has been explored in the context of intuitive physics [2], where neural networks are trained to predict object trajectories and physical outcomes. Lerer et al. demonstrated that convolutional networks can learn to predict the stability of block towers from images, suggesting that visual appearance carries recoverable physical information. Li et al. [12] introduced graph-based particle dynamics models for manipulating rigid and deformable objects. Jaques et al. [13] further showed that physical parameters can be recovered from video by treating physics as an inverse graphics problem, while Wu et al. [14] demonstrated physical property recovery via visual de-animation.

Domain randomisation [15] has proven effective for sim-to-real transfer by exposing models to diverse simulated conditions; Peng et al. [16] further demonstrated its effectiveness for dynamics randomisation. Our work applies domain randomisation over physics parameters to train models that generalise across physical conditions.

Event cameras, which respond to pixel-level brightness changes asynchronously rather than capturing full frames at fixed intervals, have shown promise for high-speed motion estimation [17] due to their sparse and temporally precise output. Gallego et al. provide a comprehensive survey of event-based vision methods and applications in robotics. However, existing approaches typically require known physics parameters, hand-crafted features, or access to full state information, limiting their applicability to real-world settings where only visual observations are available. Our work bridges these directions by applying event-based frame-difference encoding to synthetic physics simulations, enabling efficient extraction of motion signals for physics parameter inference.

III. METHOD

A. Physics Simulation

We implement a 2D multi-object physics simulator using NVIDIA Warp [18], running on a Tesla T4 GPU. The simulator supports massively parallel execution across 100 independent environments simultaneously. Each environment contains four coloured balls subject to gravity, boundary collisions, inter-object collisions, and Coulomb friction.

The dynamics update per timestep Δt follows Euler integration:

$$v_y \leftarrow v_y + g \cdot \Delta t \quad \mathbf{p} \leftarrow \mathbf{p} + \mathbf{v} \cdot \Delta t \quad (1)$$

Upon boundary contact, the normal velocity component is reversed with restitution coefficient e , and Coulomb friction is applied to the tangential velocity component:

$$v_n \leftarrow -e \cdot v_n, \quad \Delta v_t = \mu \cdot |v_n| \quad (2)$$

where $\mu \in [0, 1]$ is the friction coefficient. Inter-object collisions are resolved using impulse-based elastic collision with restitution and tangential friction, with penetration correction to prevent overlap.

For the rolling friction extension, we additionally model rotational dynamics. Each object carries an angular velocity ω . The moment of inertia for a solid sphere of unit mass and radius r is $I = 0.4r^2$. At each wall contact, the tangential impulse J_t is computed from the contact slip velocity $v_{\text{slip}} = v_{\text{contact}} - \omega r$ and clamped to the Coulomb limit $|J_t| \leq \mu |J_n|$. Angular velocity is updated as $\omega \leftarrow \omega + J_t r / I$ and clamped to $[-30, +30]$ rad/s for numerical stability.

B. Domain Randomisation

Physics parameters are independently randomised per environment at initialisation. This ensures the dataset contains diverse physical behaviours, preventing models from learning environment-specific shortcuts [15], [16]. Table I summarises the parameter ranges used.

TABLE I
DOMAIN RANDOMISATION PARAMETER RANGES

Parameter	Symbol	Range	Unit
Gravity	g	$[-12.0, -5.0]$	sim/step ²
Restitution	e	$[0.70, 0.95]$	dimensionless
Friction	μ	$[0.0, 1.0]$	dimensionless
Velocity	$v_{x,y}$	$[-1.5, 1.5]$	sim/step
Radius	r	$0.05 \text{ (v1)}; [0.03, 0.08] \text{ (v2/v3)}$	sim units
Angular vel.	ω	$[-30, +30] \text{ (v3 only)}$	rad/s

The restitution range $[0.70, 0.95]$ is chosen to reflect physically realistic rigid-body materials: values below 0.70 correspond to highly inelastic collisions (e.g., clay or foam) that are uncommon in the rigid-body manipulation scenarios this work targets, while values above 0.95 approach idealised elastic behaviour. The narrow range is therefore intentional and physically motivated, though it does inflate percentage error metrics relative to the domain range, as discussed in Section IV.

C. Dataset Generation

For each simulation timestep and environment, we render a 64×64 RGB image with colour-coded objects and a pixel-level segmentation mask. Labels include per-object position, velocity (v_x, v_y) , bounding boxes, object radius, and the environment physics parameters (g, e, μ) .

We generate three datasets of increasing complexity. The primary dataset (v1) contains 30,000 images and 120,000 object annotations across 100 environments and 300 timesteps, with uniform object radius $r = 0.05$. Dataset v2 introduces per-object randomised radii $r \sim \mathcal{U}(0.03, 0.08)$, increasing physical diversity. Dataset v3 further adds rolling friction, recording angular velocity ω as an additional label. Figure 1 shows example rendered frames from the simulator, and Figure 2 shows frames from dataset v2. All models are evaluated using an *environment-level* train/validation split (80 train / 20 val environments), ensuring that the physics parameters of validation environments are never seen during training.

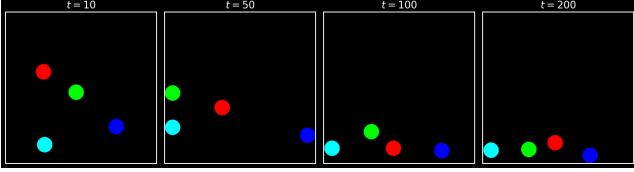


Fig. 1. Example rendered frames from the synthetic physics simulator (dataset v1). Each frame shows four colour-coded objects in a single environment at timesteps $t = 10, 50, 100, 200$.

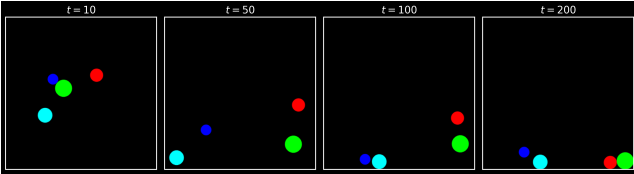


Fig. 2. Example rendered frames from dataset v2 at timesteps $t = 10, 50, 100, 200$. Objects have randomised radii.

D. Neural Network Architectures

We evaluate four architectures of increasing temporal capacity, all sharing the backbone illustrated in Fig. 3.

Single-Frame CNN: Takes a single 64×64 RGB image (3 channels). This serves as the baseline, as physics parameters are environment-level quantities whose effects accumulate over time and are not directly observable from a static image.

Two-Frame CNN: Takes two consecutive frames concatenated along the channel dimension (6 channels total), providing explicit motion information across one timestep interval.

Event-Based CNN: Computes the pixel-wise grayscale difference between consecutive frames and thresholds at $\tau = 10$ intensity units to produce sparse binary event maps: positive events where brightness increased and negative events where it decreased. The resulting 2-channel representation mimics neuromorphic event cameras [17] and discards redundant static background information while retaining motion-relevant signals.

5-Frame Sequence Event CNN: Stacks event frames from five consecutive timesteps (10 channels total), providing a longer temporal window. This is particularly motivated by environment-level parameters such as gravity and friction,

whose effects are subtle in a single frame difference but become apparent over multiple timesteps of accumulated motion.

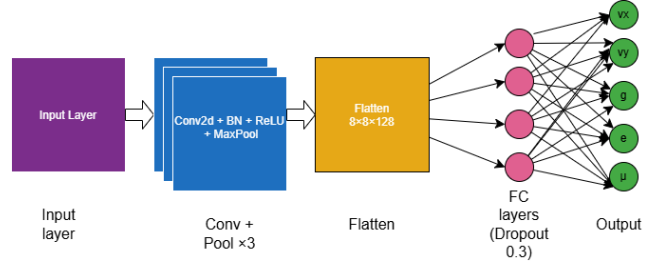


Fig. 3. Shared convolutional backbone used across all four architectures. Input channels C_{in} : 3 (single-frame RGB), 6 (two-frame RGB), 2 (event-based), 10 (5-frame sequence). Output size N : 2–7 depending on the experiment.

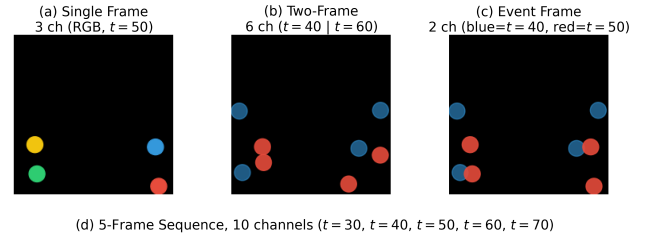


Fig. 4. Input representations. Top: (a) single-frame RGB (3 ch, $t = 50$); (b) two-frame (6 ch, $t = 40 - t = 60$); (c) event frame (2 ch, blue= $t = 40$, red= $t = 60$). Bottom: (d) 5-frame sequence (10 ch, $t = 30, 40, 50, 60, 70$).

All models use the same convolutional backbone: three convolutional layers [19], [20] (32, 64, 128 filters) each followed by batch normalisation, ReLU activation, and 2×2 max pooling. The fully connected head uses $FC(2048 \rightarrow 256) + \text{ReLU} + \text{Dropout}(0.3)$, then $FC(256 \rightarrow 64) + \text{ReLU}$, then $FC(64 \rightarrow N)$. All outputs are z-score normalised. Models are trained with Adam optimiser, MSE loss, batch size 128, and a ReduceLROnPlateau scheduler (patience 3). Training runs for 20 epochs; convergence is confirmed by the validation loss curves shown in Section IV, all of which plateau well before epoch 20.

IV. EXPERIMENTS

Our primary goal is inferring environment-level physics parameters – gravity (g), restitution (e), and friction (μ) – from visual observations. We structure the experiments in three subsections.

Subsection IV-A directly addresses the central research question: *which input representation performs best for physics parameter inference?* All five input configurations are compared under identical conditions, varying only the input representation while keeping the output head and training procedure fixed. Results are presented across three output tasks: (A1)

physics parameter inference, (A2) velocity prediction as a complementary kinematic task, and (A3) joint prediction of all five quantities simultaneously.

Subsection IV-B then evaluates the best-performing configuration on more challenging physical settings to assess *generalisation* (variable object sizes, rolling friction) and *robustness* (sensor noise). Subsection IV-C compares against temporal and physics-aware baselines. All MAE values are reported in mean absolute error (MAE) using the simulator’s internal units, where gravity is measured in sim/step^2 , velocities in sim/step , and restitution and friction are dimensionless.

Figure 5 shows validation loss curves for all four architectures on datasets v1 and v2. All models converge within 20 epochs; the 5-frame sequential model achieves the lowest final validation loss on both datasets.

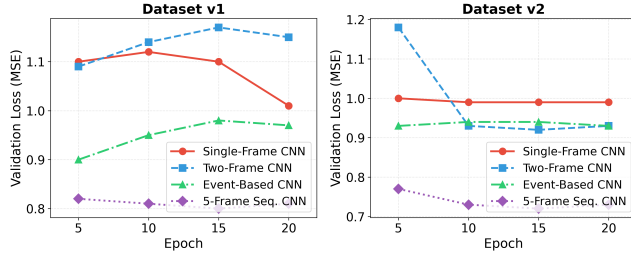


Fig. 5. Validation loss curves for all four architectures on datasets v1 (left) and v2 (right). All models plateau well before epoch 20; the 5-frame sequential model achieves the lowest final loss on both datasets.

A. Input Representation Comparison on Dataset v1

1) *Physics Parameter Inference*: Table II reports MAE for all five input configurations on a 3-output regression head predicting (g, e, μ) on dataset v1. The only variable across rows is the input representation.

TABLE II
PHYSICS PARAMETER INFERENCE MAE ON DATASET V1 – ALL INPUT CONFIGURATIONS, SAME 3-OUTPUT HEAD

Input Representation	g MAE	e MAE	μ MAE
Single-Frame CNN	1.1683	1.0075	0.9616
Two-Frame CNN	1.2068	0.9581	0.9647
Event-Based CNN	1.0993	0.8645	0.7844
5-Frame Seq. CNN	1.1273	0.8354	0.7532
RGB+Event (5-ch)	1.1824	0.9198	1.1156

Event-based encoding consistently outperforms raw RGB input across all three physics parameters. The 5-frame sequential model achieves the lowest MAE on restitution and friction, confirming that a longer temporal window improves inference of environment-level parameters whose effects accumulate over multiple frames. The Event-Based CNN outperforms both raw-frame baselines using only 2 input channels. RGB+Event fusion underperforms event-only on restitution and friction.

2) *Velocity Prediction*: Object velocity (v_x, v_y) is a per-object kinematic quantity distinct from the environment-level physics parameters. We evaluate it as a complementary task using a 2-output head on dataset v1. Table III reports results.

TABLE III
VELOCITY PREDICTION MAE ON DATASET V1 – ALL INPUT CONFIGURATIONS, SAME 2-OUTPUT HEAD

Input Representation	Ch.	v_x MAE	v_y MAE
Single-Frame CNN	3	0.6110	0.6394
Two-Frame CNN	6	0.4948	0.5343
Event-Based CNN	2	0.4242	0.5263
5-Frame Seq. CNN	10	0.3850	0.5097
RGB+Event (5-ch)	5	0.4784	0.5289

The architecture ordering is identical to Table II. The single-frame configuration shows substantially higher error, confirming velocity cannot be reliably inferred from a static image. RGB+Event performs worse than event-only on both axes.

3) *Joint Physics and Velocity Prediction*: To investigate whether predicting physics parameters and velocities jointly affects performance, we trained all five input configurations with a shared 5-output head predicting (v_x, v_y, g, e, μ) simultaneously. Table IV reports results on dataset v1.

TABLE IV
JOINT PREDICTION MAE ON DATASET V1 – ALL INPUT CONFIGURATIONS, SAME 5-OUTPUT HEAD

Input Representation	v_x	v_y	g	e	μ
Single-Frame CNN	0.6075	0.6454	1.3067	1.0080	0.9250
Two-Frame CNN	0.5391	0.5901	1.1569	0.9673	0.9836
Event-Based CNN	0.4460	0.5283	1.1217	0.8624	0.7956
5-Frame Seq. CNN	0.4076	0.5147	1.1413	0.8125	0.7399
RGB+Event (5-ch)	0.5473	0.5539	1.0873	0.9926	0.9971

The 5-frame sequential model achieves the best overall average MAE across all five quantities. The ordering is entirely consistent with Tables II and III, confirming the input representation ranking is stable across all output tasks.

4) *Dedicated Single-Parameter Inference*: To assess the observability of each physics parameter independently, we trained dedicated single-output models using the 5-frame sequence backbone. Table V compares dedicated MAE against the 3-output baseline.

TABLE V
DEDICATED SINGLE-OUTPUT INFERENCE MAE ON DATASET V1 – 5-FRAME SEQ. CNN

Parameter	Dedicated MAE	3-Output MAE
Gravity g	1.6279	1.1273
Restitution e	0.8347	0.8354
Friction μ	0.7608	0.7532

Gravity is substantially worse when trained in isolation (1.628 vs. 1.127), indicating its visual signature is entangled with the other physics outputs and benefits from shared training context. Restitution shows negligible change; friction is marginally better under joint training. These results confirm joint training is generally preferable.

5) *Percentage Estimation Error*: The three physics parameters span very different range widths (gravity: 7.0 units;

friction: 1.0; restitution: 0.25). Table VI reports MAE as a percentage of the randomisation range to enable fair cross-parameter comparison.

TABLE VI
PERCENTAGE ESTIMATION ERROR – v1, 5-FRAME SEQ. CNN

Parameter	MAE	Range	% Error	Unit
Gravity g	1.127	7.00	16.1%	sim/step ²
Restitution e	0.835	0.25	334.2% [†]	dimensionless
Friction μ	0.753	1.00	75.3%	dimensionless

[†]Range only 0.25 units; small absolute error inflates the percentage.

Gravity is estimated with 16.1% error because it exerts a continuous, sustained downward acceleration visible throughout the observation window. Friction (75.3%) is harder to infer as its effect is conditional on contact events and confounded with initial velocity. Restitution’s high percentage error reflects its narrow 0.25-unit range [0.70, 0.95], not poor absolute inference; it is observable only during brief collision events, and a wider randomisation range or longer observation horizon would reduce this error.

B. Generalisation and Robustness

1) *Effect of Variable Object Size*: We retrained the 5-frame sequential model on dataset v2 with per-object randomised radii $r \sim \mathcal{U}(0.03, 0.08)$. Table VII reports physics and velocity results.

TABLE VII
PHYSICS AND VELOCITY INFERENCE MAE ON DATASET v2 – 5-FRAME SEQ. CNN

g MAE	e MAE	μ MAE	v_x MAE	v_y MAE
0.8952	0.8439	0.9171	0.4213	0.5275

Physics inference on v2 remains comparable to v1, with gravity actually lower (0.895 vs. 1.127), confirming the model captures motion patterns rather than appearance. Velocity also remains stable.

2) *Noise Robustness*: We retrained the Event-Based CNN on dataset v1 with Gaussian noise $\mathcal{N}(0, \sigma^2)$ added to event frames before thresholding. Table VIII reports results across four noise levels.

TABLE VIII
NOISE ROBUSTNESS – EVENT-BASED CNN, DATASET v1

Noise σ	v_x	v_y	g	e	μ
0.00 (clean)	0.4264	0.5205	1.1028	0.8708	0.7934
0.05	0.4237	0.5205	1.0991	0.8648	0.7912
0.10	0.4228	0.5230	1.1028	0.8648	0.7912
0.20	0.4194	0.5242	1.0874	0.8538	0.7871
Max. degradation	<2% across all parameters				

Physics parameters remain stable even at $\sigma = 0.20$, with less than 2% MAE degradation. This arises naturally from event thresholding: low-amplitude noise falls below threshold $\tau = 10$ and is suppressed before reaching the CNN, a practical advantage for real neuromorphic sensors [17].

3) *RGB+Event Fusion*: A 5-channel input concatenates the 3-channel RGB frame with the 2-channel event frame. Table IX compares RGB+Event to event-only using the same 5-output joint head.

TABLE IX
RGB+EVENT FUSION VS. EVENT-ONLY ON DATASET v1

Encoding	v_x	v_y	g	e	μ
Event only (2-ch)	0.4460	0.5283	1.1217	0.8624	0.7956
RGB+Event (5-ch)	0.5473	0.5539	1.0873	0.9926	0.9971

RGB+Event achieves marginally lower gravity MAE but higher error on restitution and friction, confirming that static RGB appearance provides no complementary benefit when temporal event encoding is already available.

4) *Rolling Friction and Extended Parameter Set*: Table X reports results of the 5-frame sequential model on dataset v3 with 7 outputs.

TABLE X
FULL PHYSICS AND KINEMATIC PARAMETER INFERENCE ON DATASET v3 – 5-FRAME SEQ. CNN (7 OUTPUTS)

v_x	v_y	g	e	μ	ω	r
0.5837	0.5362	1.0164	0.9465	1.1471	0.7319	0.7395

The network successfully infers angular velocity (ω MAE = 0.732) and radius (r MAE = 0.740) from visual motion cues alone. Higher MAE values in v3 are expected given 7 coupled outputs under rolling dynamics. The gravity MAE (1.016) remains comparable to v1, confirming the 5-frame event representation continues to capture environment-level physics signals in the more complex setting.

Across all experiments, the architecture ranking Single-Frame < Two-Frame < Event-CNN < 5-Frame-Seq is stable regardless of output task, confirming the ranking reflects genuine differences in input representation quality.

C. Temporal and Physics-Aware Baseline Comparison

To address requests for stronger temporal baselines, we additionally trained CNN-LSTM and CNN-GRU (sequential processing of five per-frame CNN embeddings through a 256-unit recurrent layer), CNN-Transformer (two layers of 4-head self-attention, $d_{\text{model}} = 256$), and a simplified CNN-GNN (spatial 4×4 grid-patch nodes, one message-passing round, global pooling). We further evaluated a gradient-based differentiable simulation baseline that directly optimises (g, e, μ) by backpropagating through the Warp simulator [18] over 50 timesteps (300 gradient steps, $lr = 0.05$, initialised at domain midpoints). All baselines were evaluated on an independent 80/20 environment split from dataset v1; the 5-Frame Seq. CNN is retrained on the same split as a fair reference.

The 5-Frame Seq. CNN outperforms all learned baselines overall. CNN-Transformer overfits visibly. LSTM and GRU underperform as the five-frame window is too short for recurrent state accumulation. CNN-GNN is competitive on

TABLE XI
TEMPORAL AND PHYSICS-AWARE BASELINE COMPARISON ON DATASET
V1 (3-OUTPUT HEAD, INDEPENDENT SPLIT)

Model	g MAE	e MAE	μ MAE
5-Frame Seq. CNN	0.8299	0.8097	0.7376
CNN-LSTM	0.8922	1.2189	0.9106
CNN-GRU	1.0438	1.0823	0.9840
CNN-Transformer	0.9535	0.8965	0.7984
CNN-GNN	0.8946	0.8587	0.7279
Diff. Simulation	0.9393	0.4821	0.2570

friction, suggesting spatial modelling benefits contact-driven parameters.

The differentiable simulation baseline achieves substantially lower error on restitution ($e = 0.482$ vs. 0.810) and friction ($\mu = 0.257$ vs. 0.738), but higher gravity error ($g = 0.939$). Contact parameters are encoded directly in the simulator’s collision equations and are recoverable by gradient-based optimisation. Gravity produces a global trajectory bias difficult to isolate from initial velocity over 50 timesteps; the CNN learns to factor this out statistically across thousands of training environments. This complementarity between learned and differentiable approaches motivates future work combining event-based representations with differentiable physics for real-world material identification.

V. CONCLUSION

We presented a GPU-accelerated synthetic physics simulation pipeline that generates large-scale labelled datasets with domain-randomised physics parameters across three settings of increasing complexity, including variable object size and rolling friction. Building on this dataset, we introduced event-based temporal representations for physics parameter inference and conducted a comparative study of four CNN architectures of increasing temporal capacity. Event-based encoding consistently outperforms raw frame input: the Event-Based CNN outperforms both raw-frame baselines using only 2 input channels, and the 5-frame sequential model achieves the best overall performance. Comparison against recurrent, attention-based, graph-based, and differentiable simulation baselines confirms this complementarity and motivates hybrid future directions. Event-based representations show natural noise robustness via thresholding, with less than 2% degradation under heavy noise. An extension to rolling friction successfully recovers angular velocity and object radius from visual motion alone. These findings motivate the use of neuromorphic-inspired representations for robot perception of physical properties.

REFERENCES

- [1] G. Dulac-Arnold, D. J. Mankowitz, and T. Hester, “Challenges of real-world reinforcement learning,” *ArXiv*, vol. abs/1904.12901, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:140269588>
- [2] A. Lerer, S. Gross, and R. Fergus, “Learning physical intuition of block towers by example,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML’16. JMLR.org, 2016, p. 430–438.
- [3] P. W. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. Kavukcuoglu, “Interaction networks for learning about objects, relations and physics,” in *Neural Information Processing Systems*, 2016. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2200675>
- [4] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.
- [5] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” 2016.
- [6] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, “Brax - a differentiable physics engine for large scale rigid body simulation,” *ArXiv*, vol. abs/2106.13281, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:235420616>
- [7] F. de Avila Belbute-Peres, K. Smith, K. Allen, J. Tenenbaum, and J. Z. Kolter, “End-to-end differentiable physics for learning and control,” in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/842424a1d0595b76ec4fa03c46e8d755-Paper.pdf
- [8] X. Yang, Z. Ji, and Y.-K. Lai, “Differentiable physics-based system identification for robotic manipulation of elastoplastic materials,” *The International Journal of Robotics Research*, vol. 44, no. 13, pp. 2126–2155, 2025.
- [9] M. Wei, X. Yang, J. Yan, Y.-K. Lai, and Z. Ji, “Celebi’s choice: Causality-guided skill optimisation for granular manipulation via differentiable simulation,” in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 15 845–15 852.
- [10] M. Wei, X. Yang, Y.-K. Lai, S. Amir Taffirshi, and Z. Ji, “A physics-informed demonstration-guided learning framework for granular material manipulation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 37, no. 4, pp. 1590–1604, 2026.
- [11] X. Yang, M. Wei, Y.-K. Lai, and Z. Ji, “Ddbot: Differentiable physics-based digging robot for unknown granular materials,” *IEEE Transactions on Robotics*, vol. 42, pp. 152–169, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:282209753>
- [12] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, “Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids,” *ArXiv*, vol. abs/1810.01566, 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:52917627>
- [13] M. Jaques, M. Burke, and T. M. Hospedales, “Physics-as-inverse-graphics: Unsupervised physical parameter estimation from video,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:214002473>
- [14] J. Wu, E. Lu, P. Kohli, B. Freeman, and J. Tenenbaum, “Learning to see physics via visual de-animation,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/4c56ff4ce4aaf9573aa5dff913df997a-Paper.pdf
- [15] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:2413610>
- [16] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:3707478>
- [17] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 1, pp. 154–180, 2022.
- [18] M. Macklin, “Warp: A high-performance python framework for gpu simulation and graphics,” <https://github.com/NVIDIA/warp>, 2022.
- [19] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. [Online]. Available: <https://hal.science/hal-04206682>
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:206594692>