RigidFormer: Learning Rigid Dynamics and Beyond with Transformers

Zhiyang Dou[1], Minghao Guo[1], Haixu Wu[1], Doug Roble[2], Tuur Stuyck[2], Wojciech Matusik[1]
[1]Massachusetts Institute of Technology [2]Meta

***Abstract*—Learning dynamics begins with a question of *representation*: dense per-vertex meshes, or coarser units that capture how parts move and interact? We study RigidFormer, an object-centric Transformer that learns mesh-free dynamics from point sets, the form in which a depth sensor delivers a scene. Each object (or articulated part) is one token; a Transformer reasons over inter-unit interaction, a few anchors per unit carry the state, and a differentiable Kabsch projection recovers each unit's $SE(3)$ transform, so rigid structure holds by construction rather than by training. The same token-and-anchor model spans rigid, articulated, and deformable dynamics: it matches mesh-based simulators on MOVi from points alone, stays stable under 25% point masking, scales to 216 bodies, and rolls out command-conditioned motion for an ASE humanoid and a Unitree G1: the prediction a planner queries from a world model. Replacing the rigid projection with a blending module extends the same model to thin shells, soft bodies, and fragmentation. The object-level token is thus a scalable, mesh-free alternative to dense message passing.**

I. Dynamics Starts with Representation

A learned dynamics model faces a choice about what unit it reasons over. Mesh- and vertex-based simulators predict per-node forces over a connectivity graph [14, 1, 17, 22], which is expressive but ties the model to known topology and scales as $O(N_V^2)$ in interacting vertices. Robot perception rarely delivers that topology: a depth sensor returns an unordered, partial point cloud, and an articulated body is observed as a collection of moving surfaces whose joints and exact geometry are not given. Point-based models are a known route to robustness under such partial, noisy observation [13]. Most of what such a model must track, meanwhile, moves rigidly, as an object or a part on $SE(3)$, structure a network should not have to rediscover from data. A useful representation should therefore be mesh-free, tolerant of partial observation, organized around the granularity at which parts actually interact, and geometric where the motion is.

RigidFormer adopts these constraints directly: each object, or each part of an articulated body, is a single token, cutting interaction cost from vertex-level $O(N_V^2)$ to object-level $O(N_O^2)$ (Fig. 1); N_a=4 sparse anchors per unit carry the state, and a differentiable Kabsch step projects them back to a rigid transform, keeping every unit on $SE(3)$ over long rollouts. The representation is not specific to rigidity: the same architecture predicts command-conditioned articulated motion, serving as a world model a planner can query, and relaxing the rigid projection extends it to genuinely deformable scenes (Fig. 2).

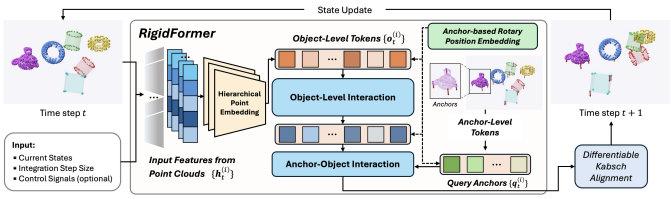


Fig. 1: RigidFormer pipeline. From two point states, a step size, and optional control signals, each object becomes a token; object- and anchor-level interactions advance a compact anchor set, integrated by Verlet and projected by differentiable Kabsch to $t{+}1$. Anchor-based RoPE aids generalization.

II. Related Work

Mesh-based dynamics. Graph-network simulators (MeshGraphNets [14], FIGNet [1], SDF-Sim [17], HopNet [22], HCMT [24]) pass messages over mesh or contact graphs: accurate, but tied to connectivity and scaling with vertex count, ill-suited to raw points and unknown joints. **Point/particle dynamics.** Particle simulators [23, 18] drop mesh assumptions but lack an explicit object abstraction. **Object-centric models.** Slot/object-token methods [8] compress scenes into entities; we instantiate this for dynamics and tie each token to a recoverable $SE(3)$ state. **Articulated motion.** Character controllers [11, 10] *generate* articulated behavior from commands; we instead *predict* it from part point sets, reusing the rigid-body machinery. Classical engines provide the reference simulations [20, 9, 3].

III. Object-Centric Model and the Articulated Mapping

RigidFormer predicts the next state from two consecutive point-level observations and an integration step size, $\mathbf{x}_{t+1} = f_\theta(\mathbf{x}_{t-1}, \mathbf{x}_t, \Delta t)$, using no mesh connectivity. Each vertex i carries a 12-D input feature ϕ_i: nearest-neighbor displacement to the closest point on another unit or the ground (contact geometry), per-step position increment (velocity surrogate), reference offset to the first frame, and physics parameters $[\text{mass}, \text{friction}, \text{restitution}]$ supplied by the simulator, an assumption we return to in Sec. V.

Object tokens. A shared hierarchical PointNet encoder [15] $\mathcal{E}$ maps each unit's points to one 768-D token, and a 4-layer Transformer [21] $\mathcal{T}$ with gated attention [16] models inter-unit interaction,

$$z_o = \mathcal{E}\big(\{\phi_i\}_{i\in\mathcal{U}_o}\big) \in \mathbb{R}^{768}, \quad \{\hat{z}_o\}_{o=1}^{N_O} = \mathcal{T}\big(\{z_o\}_{o=1}^{N_O}\big), \quad (1)$$

with $\mathcal{U}_o$ the vertex set of unit o. This is where the $O(N_V^2) \to O(N_O^2)$ reduction occurs: rigid bodies and articulated parts

both move as coherent units, so the token is the natural causal granularity.

Anchor advance and rigidity by construction. Rigid motion is 6-DoF, so each unit is advanced through $N_a{=}4$ farthest-point anchors $\{a_{o,k}\}_{k=1}^{N_a}$: three non-collinear points already determine a rigid frame, and a fourth makes the fit overdetermined, so one noisy anchor cannot corrupt the pose. Each anchor cross-attends (CA) to the interacted tokens and pulls in contact-local geometry through Anchor–Vertex Pooling (AVP), and an MLP predicts its acceleration, which Verlet integration turns into the advanced anchor position,

$$\ddot{a}_{o,k} = \mathrm{MLP}\big(\mathrm{CA}(a_{o,k}, \{\hat{z}_o\}),\, \mathrm{AVP}(a_{o,k}, \mathcal{U}_o)\big), \quad (2)$$

$$a_{o,k}^{t+1} = 2\, a_{o,k}^{t} - a_{o,k}^{t-1} + \ddot{a}_{o,k}\, \Delta t^2, \quad (3)$$

where AVP weights nearby vertices by a learnable isotropic kernel $\kappa_\sigma(d) = e^{-d/\sigma}$. A *differentiable Kabsch* step [7, 2] then fits the single $SE(3)$ motion that carries the unit's *reference* anchors onto the advanced anchors,

$$(R_o, \tau_o) = \arg\min_{R \in SO(3),\, \tau \in \mathbb{R}^3} \sum_{k=1}^{N_a} \big\| R\, a_{o,k}^{\mathrm{ref}} + \tau - a_{o,k}^{t+1} \big\|_2^2, \quad (4)$$

solved in closed form by the SVD of the centered anchor cross-covariance and broadcast to every vertex from the reference frame, $x_i^{t+1} = R_o\, x_i^{\mathrm{ref}} + \tau_o$ for $i \in \mathcal{U}_o$, so no shape drift can accumulate and long rollouts stay stable. For a free rigid body the projection enforces global rigidity; for an articulated part it enforces *part*-level rigidity while joint behavior emerges from cross-part attention.

Geometry and conditioning. Anchor-based RoPE (ARoPE) encodes spatial extent from the anchors, mean-pooled to be invariant to anchor reindexing [19, 6]; with no sequence-index embedding the decoder is permutation-equivariant over units. We do not impose global $SE(3)$-equivariance, since gravity and ground contact break that symmetry. Step size enters through a FiLM [12] modulation $\mathrm{FiLM}(h; c) = \gamma(c) \odot h + \beta(c)$ of the decoder features.

Articulated parts as object components. Articulated bodies require no new module: each body part is one object-level component, joint coupling is captured by the same cross-part attention used for inter-object contact, and a control command (target speed, move direction, facing direction) is injected through its own FiLM modulation, mirroring the step-size pathway. An ASE humanoid [11] becomes 17 interacting components driven by a 5-D command; a Unitree G1 becomes 31 components driven by a 3-D command. Part-level Kabsch keeps each segment rigid while the network learns how commands reshape the inter-part configuration over time.

IV. Experiments

We evaluate on MOVi-A (3 primitives), MOVi-B (11 categories, 51–1142 vertices), and MOVi-Sphere [4], each with 1200 scenes of 480 frames, following the HopNet protocol (2 warmup frames, then autoregressive rollout). Metrics are center-of-mass translation RMSE (m) and quaternion-geodesic orientation RMSE (deg). RigidFormer (174.8M parameters) trains with AdamW (lr 10^{-4}) for 300 epochs, step sizes sampled from $\{1, 5, 10\}$.

TABLE I: Performance on MOVi-A/B/Sphere: position RMSE (m) / orientation RMSE (°) at 50/100 frames (step size 1). Top-two per column shaded (1st, 2nd); RigidFormer uses point inputs only.

Model	MOVi-A		MOVi-B		MOVi-Sphere	
	50	100	50	100	50	100
MGN^{+}	0.705/31.21	N/A	0.538/26.91	N/A	N/A	N/A
MGN-LargeRadius^{+}	0.119/15.07	N/A	0.460/26.34	N/A	N/A	N/A
FIGNet	0.115/14.84	N/A	0.127/13.99	N/A	N/A	N/A
FIGNet$_{\mathrm{reimpl}}$	0.132/7.10	0.492/23.30	0.141/7.39	0.516/24.96	N/A	N/A
HCMT$^{*}_{\mathrm{reimpl}}$	0.239/5.70	0.951/18.40	0.237/4.72	0.932/17.43	0.243/4.19	0.956/13.81
VPD$^{\dagger}_{\mathrm{reimpl}}$	0.235/5.10	0.827/20.37	0.275/4.65	0.987/16.99	0.244/4.47	0.855/17.65
HopNet	0.054/5.64	0.196/18.83	0.047/4.91	0.176/17.91	0.034/4.05	0.124/13.68
RigidFormer†,*	0.049/5.06	0.177/18.32	0.050/3.97	0.161/15.33	0.026/3.00	0.099/11.19

* Transformer-based; † point inputs; $_{\mathrm{reimpl}}$ reimpl.; $^{+}$ prior work.

Rigid main results. Tab. I shows that the object-token representation, despite consuming only points, matches or surpasses mesh-based simulators: on MOVi-B at 100 frames it improves the strongest baseline from $0.176/17.91°$ to $0.161/15.33°$, and on MOVi-Sphere it leads at every horizon ($0.099/11.19°$ at 100 frames). A coarse object-level representation is thus not a fidelity sacrifice relative to dense mesh message passing.

Variable step size. One trained RigidFormer rolls out at step sizes $\{1, 5, 10\}$ through FiLM conditioning, trading temporal resolution for horizon. Larger steps issue fewer autoregressive updates and lower the 50-frame error: on MOVi-B, RigidFormer improves from $0.050/3.97°$ at step 1 to $0.029/1.51°$ at step 10. A planner can thus query coarse rollouts for action search and fine rollouts near contact from one model, with no per-Δt retraining.

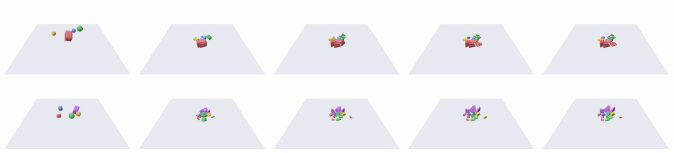
Articulated dynamics. Treating each part as an object-level component and feeding commands through FiLM, RigidFormer reaches $0.062/14.47°$ at 100 frames for the ASE humanoid and $0.072/16.26°$ for the Unitree G1, the same error scale as the multi-object rigid task, with no architectural change. The same machinery covers a manipulator acting on a free object: Fig. 4 shows simulated Franka push-cube rollouts where the arm's links are object-level components and the cube a free body, co-evolving under contact, with no joint-aware loss or per-embodiment tuning.

Partial point clouds. Robots see incomplete surfaces. Masking 25% of each unit's points at test time, reusing the *same* full-cloud model with no retraining or completion, still yields stable, low-drift rollouts: the token-plus-anchor representation degrades gracefully, summarizing object extent rather than reconstructing it vertex by vertex.

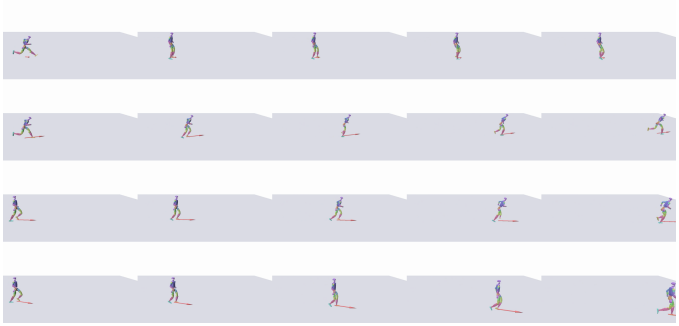
Resolution generalization. Trained with randomly sampled point counts, RigidFormer stays stable at an unseen test resolution of 768 points per object: the token summary does not overfit to sampling density, which matters when one model ingests clouds of varying size across sensors and scales.

Scalability and controllability. On WreckingBall scenes with 64, 125, and 216 cubes (Fig. 6, left), RigidFormer stays stable at $\sim$20 FPS, using $\sim 58\times$ fewer attention pairs than vertex-level at 217 objects (216 cubes plus the ball), and follows heading and speed commands for articulated embodiments (Fig. 6, right): the payoff of choosing the object as the reasoning unit when many parts interact.

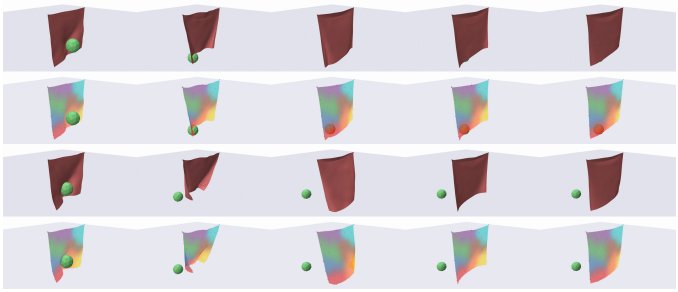
(a) Rigid to soft: a learnable skinning module under physics-informed supervision; FEM reference simulated with libuipc.



(b) Object fragmentation: a body cracking into many free pieces under impact.



(c) Controllable Unitree G1, each row a different initial state and command (Isaac Gym).



(d) Thin-shell drape: mesh visualization (rows 1, 3) and per-vertex skinning weights (rows 2, 4).

Fig. 2: RigidFormer also models non-rigid and controllable dynamics: soft bodies, object fragmentation, and deformable thin shells relax the rigid projection to a blending module, while the command-conditioned humanoid keeps per-part Kabsch, all from the same object-token backbone. Meshes are shown only for visualization; the model operates on points.

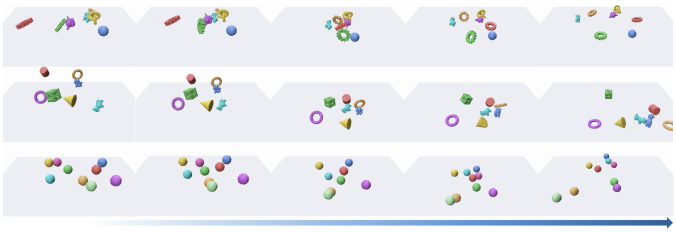


Fig. 3: Qualitative RigidFormer rollouts on held-out MOVi-A, MOVi-B, and MOVi-Sphere scenes, capturing multi-object bounces, stacking, and resting contacts over long horizons. Meshes are shown only for visualization; the model operates on point inputs.

Fig. 4: Simulated Franka push-cube (three trajectories, six time-spaced frames each). The 7-DoF arm is modeled as interacting object-level components and the cube as a free rigid body; both co-evolve under contact. This manipulation regime, with an articulated arm acting on a free object, lies beyond the MOVi free-fall setting and motivates action-conditioned rollout.

Beyond rigidity: thin shells, soft bodies, fragmentation. The interaction layer is agnostic to how a unit advances; only the final projection encodes rigidity. Replacing the per-component Kabsch step with a *blending* module turns RigidFormer non-rigid: each vertex follows an RBF-skinned, residual-corrected blend of the anchor transforms instead of one rigid motion (shells are split into 8 patch tokens of 4 anchors each), so a shell drapes over a sphere, soft bodies deform, and objects fragment (Fig. 2). The same rollouts also drive rendering, coupling RigidFormer with Diffusion-as-Shader [5]

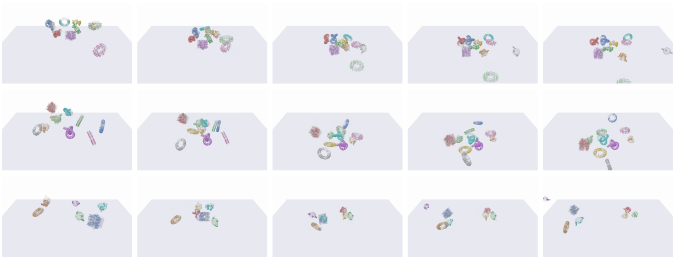


Fig. 5: Partial point-cloud rollouts. Three example sequences (rows) with 25% of each object's points masked at test time; RigidFormer reuses the same model with no retraining and stays stable, with accurate inter-object contacts.

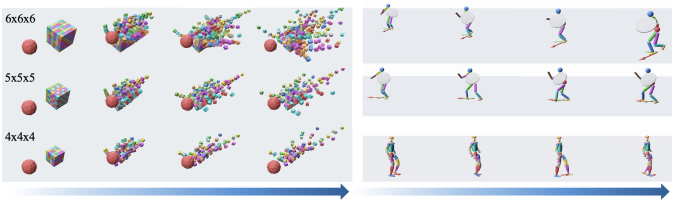


Fig. 6: Scalability and controllability. Left: 216/125/64 cubes; RigidFormer stays stable as object count grows. Right: direction-controlled articulated dynamics (humanoid: green/red = facing/move arrows; G1: arrow = move direction).

to restyle a trajectory under varied prompts (Fig. 7).

Efficiency. On MOVi-B 50-step rollouts, RigidFormer runs at 41.9 ms/step (23.9 FPS) at 1.80 GB peak memory, 8× faster than FIGNet (336.0 ms) and 101× faster than HopNet (4228.7 ms); for closed-loop manipulation, where many futures are rolled out per control cycle, this margin favors object-level tokenization. Tab. II summarizes these trade-offs.

V. Discussion, Limitations, and Future Work

RigidFormer's contribution is *representational*: one object-level token plus sparse anchors per unit is a mesh-free,

Fig. 7: Controllable video generation: coupling RigidFormer with Diffusion-as-Shader renders one predicted trajectory under varied style prompts. Left two: the tracking rollout and rendered mesh used as controls; right four: stylizations driven by the same predicted motion.

TABLE II: Method comparison. Mesh-free: no connectivity; Var. Δt: multiple step sizes; Preproc.-free: no offline geometry; #Warmup: frames to start a rollout.

Method	Mesh-free	Var. Δt	Preproc.-free	#Warmup ↓	FPS ↑
MGN [14]	✗	✗	✓	2	5.7
FIGNet [1]	✗	✗	✓	3	3.0
SDF-Sim [17]	✗	✗	✗†	3	–
HopNet [22]	✗	✗	✗‡	3	0.2
RigidFormer (Ours)	✓	✓	✓	2	**23.9**

† SDF pre-learning $\sim$5 h; ‡ simplicial-complex build $\sim$15 days (MOVi-B).

perception-friendly state that matches dense mesh simulators on rigid contact, tolerates partial observation, scales to hundreds of bodies, and covers command-conditioned articulated bodies with no architectural change. The blending variant shows the same backbone spanning the rigid-to-soft range: the token carries the interaction prior, and the per-unit projection picks the deformation model, so shells, soft bodies, and fragmentation reuse the rigid contact machinery.

Limitations and future work. The current results come with four caveats. The input features include per-vertex physics parameters (mass, friction, restitution): setting them is how material behavior is specified, but they come from the simulator, and a real system rarely knows them; inferring them from observed motion is open. Beyond the articulated command channel, the model is a passive predictor of scene evolution, not yet conditioned on applied robot actions. The projection guarantees per-unit rigidity, so no prediction can deform an object, but inter-unit penetration is not excluded and should be measured. And every evaluation here is in simulation: we have not run the model on real captures, nor compared against analytic engines such as MuJoCo [20], so how far it carries across the sim-to-real gap is untested. Beyond these lie a joint-aware objective, a quantitative deformable benchmark, robustness to unknown part segmentation (per-unit membership is currently given), and action-conditioned rollouts scored by control success rather than open-loop RMSE.

References

[1] Kelsey R Allen, Yulia Rubanova, Tatiana Lopez-Guevara, William Whitney, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Learning rigid dynamics with face interaction graph networks. *arXiv preprint arXiv:2212.03574*, 2022.

[2] Romain Brégier. Deep regression on manifolds: a 3d rotation case study. In *2021 International Conference on 3D Vision (3DV)*, pages 166–174. IEEE, 2021.

[3] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[4] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3749–3761, June 2022.

[5] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–12, 2025.

[6] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

[7] Wolfgang Kabsch. A solution for the best rotation to relate two sets of vectors. *Foundations of Crystallography*, 32(5):922–923, 1976.

[8] Chanho Kim and Li Fuxin. Object dynamics modeling with hierarchical point cloud-based representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20977–20986, 2024.

[9] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[10] Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

[11] Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[12] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[13] Skand Peri, Iain Lee, Chanho Kim, Li Fuxin, Tucker Hermans, and Stefan Lee. Point cloud models improve visual robustness in robotic learners. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17529–17536. IEEE, 2024.

[14] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

[15] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[16] Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, et al. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *arXiv preprint arXiv:2505.06708*, 2025.

[17] Yulia Rubanova, Tatiana Lopez-Guevara, Kelsey R Allen, William F Whitney, Kimberly Stachenfeld, and Tobias Pfaff. Learning rigid-body simulators over implicit shapes for large-scale scenes and vision. *Advances in Neural Information Processing Systems*, 37:125809–125838, 2024.

[18] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, pages 8459–8468. PMLR, 2020.

[19] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[20] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[22] Amaury Wei and Olga Fink. Integrating physics and topology in neural networks for learning rigid body dynamics. *Nature Communications*, 16(1):6867, 2025.

[23] William F Whitney, Jacob Varley, Deepali Jain, Krzysztof Choromanski, Sumeet Singh, and Vikas Sindhwani. Modeling the real world with high-density visual particle dynamics. *arXiv preprint arXiv:2406.19800*, 2024.

[24] Youn-Yeol Yu, Jeongwhan Choi, Woojin Cho, Kookjin Lee, Nayong Kim, Kiseok Chang, Chang-Seung Woo, Ilho Kim, Seok-Woo Lee, Joon-Young Yang, et al. Learning flexible body collision dynamics with hierarchical contact mesh transformer. *arXiv preprint arXiv:2312.12467*, 2023.