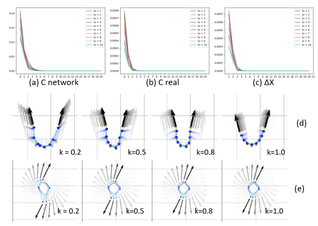
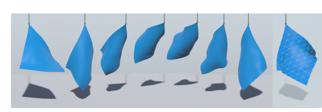
We thank all the reviewers for their constructive suggestions and insightful comments. We will address all the suggested expositional changes such as the algorithm complexity, details on training, and adding all the suggested references. We will also release our codebase with clear comments and naming conventions to ensure reproducibility.

Role of C and Quantitative Evaulations (R1,R2,R3,R4) The network output C corresponds to pure constraint satisfaction. No transition component was involved in the final output, owing to the projection nature of our iterative algorithm. In this sense, the constraint satisfaction functioned as the key criterion for our evaluations to measure the effectiveness of the learning model (e.g., see Fig 9). This criterion was fundamentally different from the previous work and we believed it to be a vital point to quantitatively reveal the nature of a constraint dynamic system. This fact is further evidenced by our latest experiment shown in the inset figures. We plotted the learned constraint C_{net} (network output), the real, observed constraint C_{real} (analytical expression), and the magnitude of the positional correction $\Delta \mathbf{x}$ against iteration steps. We observed that all three quantities converge to zero after 5 iterations (the same iteration number we set in training). Further, we calculated the Pearson correlation coefficient of our learned C and real C, using 1000 random frames, each with 5 iterations, and got a statistically significant correlation of 0.914 (for linear relations). This observation indicates a clear physical meaning of the network C that is almost identical to its analytical counterpart. To show this, we further operate on the learned C by relaxing its value and observe different levels of constraint satisfaction. For instance, we can predict rope / rigid behaviors with different stiffness by relaxing the network C to different extents (see figure (d)(e) above).



Comparison with IN and Other Approaches (R1,R2,R3) We chose to make direct comparisons with IN because we believe it is the family of approaches most relevant to ours, which aims to uncover *unknown* dynamics from limited observation. We did not compare our method with differentiable physics solvers, such as ChainQueen, which assume *known* governing equations. One of the main reasons that our model can outperform IN is due to its implicit nature, realized by time-independent correction, which is inherently suited for tackling stiff systems such as rigid and articulated bodies. Such systems are challenging for explicit transitional methods due to the timestep restriction (imagine the difficulty on simulating a rigid body using springs with infinite stiffness). Currently, we reported the timestep size ($\Delta t = .1$ for all examples) in Supplementary. We will highlight this in Results and add further discussions endorsed by a new experiment we conducted to demonstrate the different timestep sensitivities between the two models. We are also happy to incorporate comparisons with other models, but to the best of our knowledge, our projection paradigm is the only approach that can uncover constraints in such a simple and end-to-end fashion.

Real-World 3D Applications (R1,R2,R3,R4) There was no technical barrier that prevents our approach from being used in predicting 3D physical systems. Here we show a 3D cloth example, as asked by R4, to showcase its capability in predicting more complicated 3D physics. We are happy to extend all our four examples to 3D to better demonstrate its scalability. On the other hand, we also want to argue that the main difficulty on reasoning a real-world physical system lies in the system's range of stiffness rather than its number of DoFs. E.g., a rigid body has 6 effective DoFs only, yet its dynamics is challenging to obtain using a transitional learning model which does not take a rotational prior and has the same parameter size as ours (0.3M). The four examples we showed in our manuscript covered dynamic systems exhibiting a broad range of stiffness and different types of constraints, which we believe can characterize the main portion of real-world solid systems (rigid, soft, articulation, and collision). Last, as mentioned in Limitation, we acknowledge that our algorithm can process solids only (rigid, soft, or any system with a fixed material space). This model cannot predict Eulerian systems with temporally varying local relations (e.g., fluid). R1 made insightful suggestions on tackling such challenges by incorporating GNNs into neural projection. We will discuss this direction in Future Work.



R1 Individual Comments *Collision*: All the collisions in the dataset are currently inelastic; *Alg 2*: Yes, the outer loop is for averaging corrections among groups and the projection function is the same as Alg 1.; *Does Δx converge to zero?* Yes! *More/fewer iterations at test time?* More is fine (because of projection) but fewer does not work; *Are sample points the same in Fig 6?*: Yes; *Error accumulation*: The constraint errors do not accumulates but the trajectory errors do; *IN and MLP*: We used a single layer IN for comparison. MLP outperforms IN because it predicts correction only.

R2 Individual Comments *Size of benchmark*: See Supplementary B1; *Parameter variation*: We randomized initial conditions for position, orientation, and external forces, all ranging from $[-5, 5]$; *Problem structure*: We used the analytical expressions to measure position, length, and angle constraints; *Simulation*: Our model is not sensitive to simulation algorithms as far as the underlying constraints can be observed from data; *Single example*: The plots for Fig 3-6 were used specifically to accommodate the animated examples. We had obtained and will incorporate statistical data with more parameter variations; *Grouping*: Yes, the grouping information was set as a prior input.

R3 Individual Comments *Gradient of projection*: The gradient was calculated using the standard auto-differentiation.