# Supplementary Material for CoRL 2024 Submission: ”Learning Differentiable Tensegrity Dynamics using Graph Neural Networks”

## A Training details

**Training strategy** For models trained with simulation data, a curriculum learning strategy is employed, where the trajectory roll-out length is progressively increased. Initially, the model is trained to predict the immediate 1-step ahead state for 200 epochs. The model train loss and validation loss will start to flatten out and further training tends to see an increase in full trajectory error. Next, the model is trained to perform a 2-steps look ahead for 100 epochs, 4-steps ahead for 50 epochs, and finally, 8-steps ahead for 25 epochs. It is observed that further increase in roll-out length does not decrease error. This procedure allows the model to incrementally experience the errors it makes during roll-out and to adjust to them.

For the mesh and surface-based models, the models were trained based on the procedures described in prior work [1, 2, 3]. These models were only trained with 1-step roll-out lengths and with Gaussian random-walk noise of $5 \times 10^{-4}$ to node positions upon input to the model, for 800 epochs.

**Training hyperparameters** The models were trained with progressively decreasing learning rates of $10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}$. These learning rates correspond to the n-steps look ahead curriculum phases as detailed above. A mini-batch size of 128 was used during training. The Adam optimizer with weight-decay $10-2$ was used.

**Data Augmentation** At the start of each training step, the mini-batch is rotated by a random angle $[-\pi, \pi]$ about the z-axis.

## B Model details

**Network architecture** All the Multi-Layer Perceptron ($\mathbf{MLP}$) models have two layers, ReLU activations, and residual connections. All $\mathbf{MLP}$s, except the decoder, apply LayerNorm on their the outputs. For the surface-based vs multi-object experiments, the width and latent vector dimension sizes were 64. For all other experiments, the dimension sizes were 128. Lastly, models for the 3-bar tensegrity had 4 message-passing steps, while the models for the 6-bar tensegrity had 10 message-passing steps due to the larger graph. All inputs were normalized to zero-mean and unit variance.

**Cable model** The cable model is a linear model following the Hooke's law that allows for tension but not compression. It computes the cable force $\mathbf{F}^t_{cable}$ at time $t$, based on the stiffness component $F^t_K$ and the damping component $F^t_c$:

$$
\begin{aligned}
F^t_K &= \begin{cases} K(l^t_{rest} - \Delta x^t) & \text{if } l^t_{rest} \geq \Delta x^t \\ 0 & \text{otherwise} \end{cases} \\
F^t_c &= c(\mathbf{v}^t_{rel} \cdot \hat{\mathbf{x}}^t) \\
\mathbf{F}^t_{cable} &= (F^t_k - F^t_c)\hat{\mathbf{x}}^t
\end{aligned}
$$

where $K$ is the cable stiffness, $c$ is the cable damping, $l^t_{rest}$ is the rest length at time $t$, $\Delta x^t$ is the distance between the cable attachment points at time $t$, $\mathbf{v}^t_{rel}$ is the relative velocity between the cable attachment points, and $\hat{\mathbf{x}}^t$ is the unit direction pointing from one attachment point to the other.

**Actuation model** The actuation model consists of a linear model of a motor that receives an input control signal $u^t$ between $[-1, 1]$, and acts on a cable by changing the cable's rest length $l^t_{rest}$ at time t.

$$\omega^t = s\omega_{max}u^t$$
$$\Delta l_{rest} = 0.5 * (\omega^t + \omega^{t-1}) r_{winch} \Delta t$$
$$l_{rest}^t = l_{rest}^{t-1} - \Delta l_{rest}$$

where $\omega^t$ is the angular velocity of the motor at time $t$, $s$ is an input speed parameter $[0, 1]$, $\omega_{max}$ is the maximum angular velocity the motor can achieve, $\Delta l_{rest}$ is the change in rest length due to the motor, $r_{winch}$ is the winch radius of the motor, and $\Delta t$ is the time step size.

# C Tensegrity Robot Details

Table 4 provides the values for physical parameters of the real tensegrity robot. These measurements were used in order to set the corresponding parameters in simulation to the same value. The parameters are not learned by the differentiable physics engine.

| Attribute | Measurement |
|---|---|
| inner rod length | $0.325m$ |
| inner rod radius | $0.0016m$ |
| inner rod mass | $3.8g$ |
| end cap radius | $0.0175m$ |
| end cap mass | $10.5g$ |
| motor radius | $0.0175m$ |
| motor length | $0.045m$ |
| motor offset (center to center) | $0.1175m$ |
| motor mass | $35.3g$ |
| short cable stiffness | $10^6 N/kg$ |
| short cable damping ratio | $10^3 N \cdot s/m$ |
| long cable stiffness | $10^4 N/kg$ |
| long cable damping ratio | $10^3 N \cdot s/m$ |

Table 4: Physical tensegrity robot measurements used in simulation parameters