Dear Reviewers: Thank you for the comments. We address the main issues and clarify some confusions below.

Comparison to optimization methods (e.g., Wang et al.) using finite differences (Reviewers #1, #3). To obtain the ground-truth stretching and bending parameters, Wang et al. designed a number of controlled real-world environments. With known external forces and labeled data, they used L-BFGS to optimize the parameters to fit the observed data. They used finite differences to estimate the gradient.

For comparison, we run their optimization method in our environments, as requested. We used the PyTorch L-BFGS implementation and set the learning rate ranging from 0.1 to 0.2 depending on the convergence speed. Using the best parameters that we can obtain, we report runtime and accuracy in Table 1. (See Sec. 4.2 for definitions of metrics.) We report the runtime per simulation step of each iteration for both methods. The error metric of the material parameters is the Frobenius norm of the difference normalized by the Frobenius norm of the target. The error of the simulated result is defined by the average pairwise vertex distance normalized by the size of the cloth. The numbers in each cell are mean values with the standard deviation across 10 sample materials. Our method achieves better results and runs faster.

The mathematical derivation and notations (Reviewers #1, #2): We will revise them as suggested. The detailed derivation will be provided in the supplementary document.

Characterization of control task (Reviewer #1). The initial control force is set to zero. The control network consists of two FC layers, where the input (size $81 * 2 * 3$) is the position and velocity of each vertex, the hidden layer is of size 200, and the output is the control force (size $4 * 3$). The learning rate is 10^{-4} and the momentum is 0.5. The reported result is the best among 10 trials. We will provide these and other implementation details in the supplement.

Cloth-body interaction (Reviewer #1). As mentioned in Sec. 3.2 and 3.4, the cloth-body interaction is achieved by continuous collision detection using a bounding volume hierarchy (BVH), and collision response using impact zone optimization. It can be integrated into other simulations as long as the corresponding mesh BVH is used, which is often the case.

Comparison to regular simulators (Reviewer #1). Our contribution to the efficiency is mostly in the backward propagation phase, which regular simulators do not have. Our simulator is designed to be embedded in deep networks. When gradients are needed, our simulator shows significant improvement over finite difference methods, as discussed above. Regular simulators need to run one simulation for each input variable to compute the gradient, while our method only needs to run once for all gradients to be computed. Therefore, the more input variables there are during learning, the greater the performance gain that can be achieved by our method over finite difference methods.

Relationship between Algorithm 1 and QP (Reviewer #2). Algorithm 1 is a general flow of physical simulations. During the collision response phase, a set of linear constraints needs to be satisfied to avoid collision. In order to introduce minimum change to the original mesh state, we develop a QP problem to solve for the constraints. We will provide a general tutorial of physical simulation as a QP problem in the supplement.

Collision-rich applications (Reviewer #3). We will provide additional examples of collision-rich motion control in the final version.

Discussion on frictional contacts (Reviewer #3). Frictional contacts are modeled using frictional forces when two objects are in close proximity and have relative motions. The description was omitted since the formulation for adding frictional forces is standard. We are happy to add the description in the final version.

Method	Runtime (sec/step/iter)	Density Error (%)	Non-Ln Streching Stiffness Error (%)	Ln Streching Stiffness Error (%)	Bending Stiffness Error (%)	Simulation Error (%)
Wang *et al.*	2.89 ± 0.02	4.2 ± 5.6	64 ± 34	72 ± 90	$\mathbf{70 \pm 43}$	4.9 ± 3.3
Ours	$\mathbf{2.03 \pm 0.06}$	$\mathbf{1.8 \pm 2.0}$	$\mathbf{57 \pm 29}$	$\mathbf{45 \pm 41}$	77 ± 36	$\mathbf{1.6 \pm 1.4}$

Table 1: Results on the material parameter estimation task. Results with lower values have higher accuracy. 'Ln' stands for 'linear' in the table. Our method achieves higher accuracy with faster runtime in comparison to Wang *et al.*, which uses finite differences for gradient computation.