Trust-Region Eigenvalue Filtering for Projected Newton

HONGLIN CHEN, Columbia University, United States of America

HSUEH-TI DEREK LIU, Roblox & University of British Columbia, Canada

ALEC JACOBSON, University of Toronto & Adobe Research, Canada

DAVID I.W. LEVIN, University of Toronto & NVIDIA, Canada

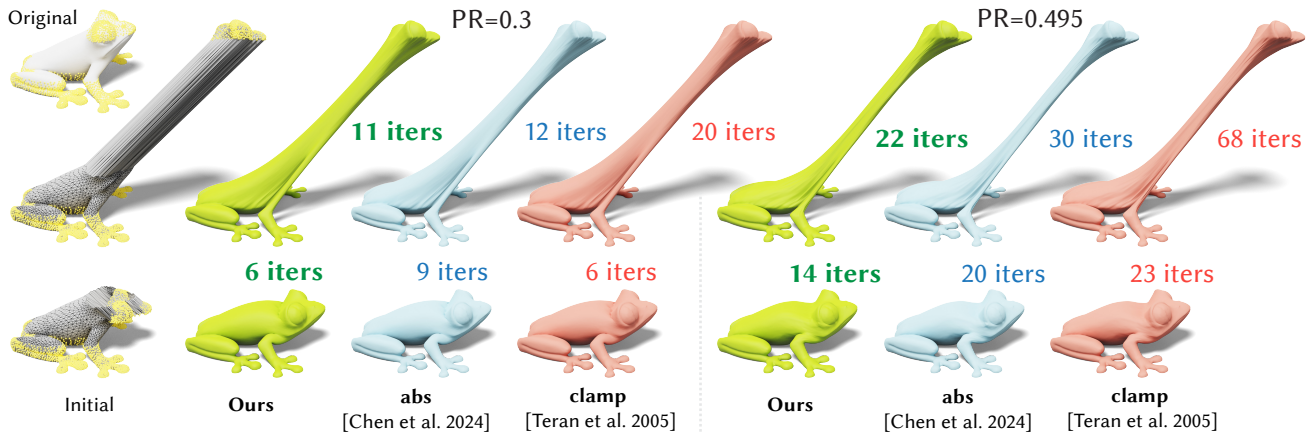CHANGXI ZHENG, Columbia University, United States of America

Fig. 1. Our adaptive eigenvalue projection scheme stabilizes the projected Newton optimization of stable Neo-Hookean energy under high Poisson's ratios (PR) and large initial volume change while maintaining fast convergence in other cases. Here the fixed vertices are colored in yellow. Our method consistently outperforms the other approaches.

We introduce a novel adaptive eigenvalue filtering strategy to stabilize and accelerate the optimization of Neo-Hookean energy and its variants under the Projected Newton framework. For the first time, we show that Newton's method, Projected Newton with eigenvalue clamping and Projected Newton with absolute eigenvalue filtering can be unified using ideas from the generalized trust region method. Based on the trust-region fit, our model adaptively chooses the correct eigenvalue filtering strategy to apply during the optimization. Our method is simple but effective, requiring only two lines of code change in the existing Projected Newton framework. We validate our model outperforms stand-alone variants across a number of experiments on quasistatic simulation of deformable solids over a large dataset.

ACM Reference Format:

Honglin Chen, Hsueh-Ti Derek Liu, Alec Jacobson, David I.W. Levin, and Changxi Zheng. 2024. Trust-Region Eigenvalue Filtering for Projected Newton. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687650

1 INTRODUCTION

Hyperelastic simulations are indispensable in capturing the rich visual behaviors of deformable objects – from highly compressible foam to nearly incompressible rubber and human tissues. However, the numerical optimization of these energies can be notoriously challenging due to the non-convexity of the volume-preserving term. This high non-convexity can lead to unstable and slow convergence in commonly applied Projected Newton optimizers, especially under high Poisson's ratios (near 0.5) and large initial volume change. This necessitates the use of eigenvalue filtering strategies to enforce the positive definiteness of the Hessian matrix and guarantee convergence of the optimization algorithm.

In this work we depart from the Projected Newton perspective and show that (somewhat surprisingly), we can unify existing eigenvalue clamping and filtering strategies under the *generalized trust region* formalism. This motivates us to rethink the eigenvalue filtering step. Instead of *a priori* choosing eigenvalue filtering strategy ahead of time, we show that a significant reduction in Newton iterations can be obtained by adaptively choosing the correct strategy based on how well a quadratic model fits the underlying nonlinear energy in a particular trust-region. This leads to an extremely *elegant* and *effective* solution that requires only two lines of code change in an existing Projected Newton solver to implement our more performant, trust-region-based approach (see Sec.4.3).

Authors' addresses: Honglin Chen, Columbia University, United States of America, honglin.chen@columbia.edu; Hsueh-Ti Derek Liu, Roblox & University of British Columbia, Canada, hsuehtil@gmail.com; Alec Jacobson, University of Toronto & Adobe Research, Canada, jacobson@cs.toronto.edu; David I.W. Levin, University of Toronto & NVIDIA, Canada, diwlevin@cs.toronto.edu; Changxi Zheng, Columbia University, United States of America, cxz@cs.columbia.edu.

We demonstrate the effectiveness and efficiency of our method through a wide range of challenging examples, including different deformations, physical parameters, geometries, mesh resolutions, and elastic energies. Through extensive experiments, we show that on average, our adaptive method outperforms all other stand-alone variants in terms of stability and convergence speed in quasistatic simulations. Our method is robust across different mesh resolutions and Poisson's ratios and highly distorted and inverted elements, while maintaining the same per-iteration computational cost and simplicity of implementation.

To summarize, our contributions are as follows:

(1) We offer a theoretical analysis of nonconvex optimization of Neo-Hookean energy based on the generalized trust region framework. For the first time, we show that Newton's method, Projected Newton with eigenvalue clamping and Projected Newton with absolute eigenvalue filtering can be unified based on insights from the generalized trust region method.

(2) Based on our analysis, we propose a novel adaptive eigenvalue filtering strategy which correctly chooses the strategy to apply based on the trust-region fit. Our method is still formally a (regularized) Projected Newton, where the regularization parameter mimics the Lagrange multiplier arising in trust-region subproblems. Our approach ensures stability under high Poisson's ratios and large initial volume change, while still maintaining fast convergence in all other regimes.

2 RELATED WORK

Projected Newton's method. Projected Newton's methods rely on projection strategies to restore the positive definiteness of the Hessian matrix (to guarantee convergence to a local minimum). As an example, Paternain et al. [2019] proposed projecting eigenvalues of the *global* Hessian to a small positive value when its magnitude is small, and to its absolute value otherwise. Due to the computational cost of a global eigendecomposition, in a finite element setting, these projection often operate on the element-wise Hessian submatrices. Choices of projection range from clamping negative eigenvalues to a small positive number (or zero) [Teran et al. 2005], adding a diagonal [Fu and Liu 2016] or setting negative eigenvalues to their absolute values [Chen et al. 2024; Gill et al. 1981; Nocedal and Wright 2006]. Some strategies eschewed eigenvalue filtering which allows for modifications to the global hessian. For instance, Longva et al. [2023] either suggested using the original Hessian matrix whenever it is positive definite or adding multiples of the mass matrix until Cholesky Factorization succeeds.

Trust-region method. Trust-region methods have been extensively studied in optimization literature [Conn et al. 2000; Nocedal and Wright 2006] for its stability and robustness. In some sense it can be viewed as the dual of line-search method: it first chooses a step size and then picks a step direction within it. Sorensen [1982] first proposed the trust region method by making a trust region modification to Newton's method. Moré and Sorensen [1983] studied the strategy to pick the trust region step. Moré [1993] extended it to the generalized trust region method, which has been actively studied until today [Pong and Wolkowicz 2014; Wang and Kilinc-Karzan 2020]. Several works combined the idea of trust-region and line search
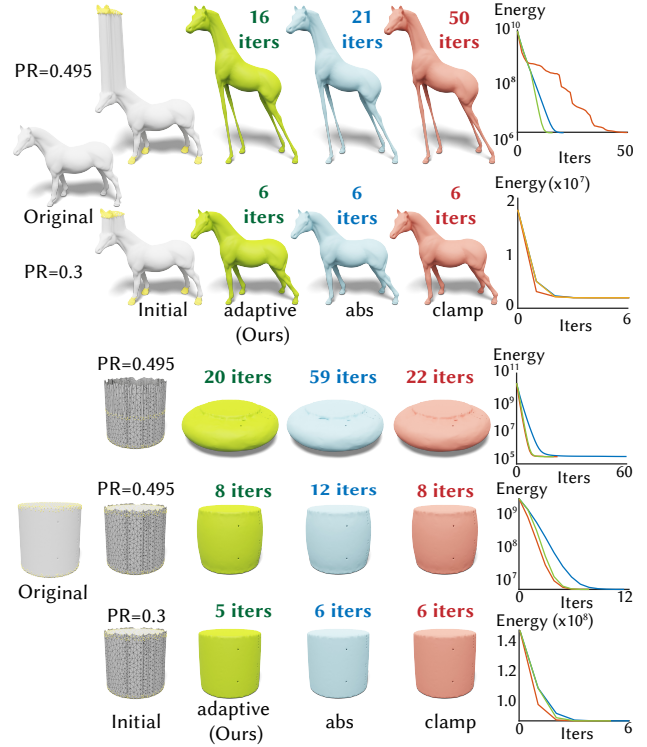
Fig. 2. Our adaptive strategy stabilizes the optimization under large volume changes and high Poisson's ratios (PR), while maintaining fast convergence in other cases. The three columns in the middle visualize the deformed meshes after the energy minimization converges in our method (green), absolute eigenvalue projection (blue), and eigenvalue clamping (red). The number of iterations for each method until convergence is labeled on the top. The last column shows the energy convergence curves with respect to the number of iterations.

method and propose the regularized Newton's method [Steck and Kanzow 2023; Ueda 2014; Zhang and Ni 2015]. In machine learning, Dauphin et al. [2014] introduced the saddle-free Newton method, which is a generalized trust region method with a first-order model and a quadratic constraint.

Neo-Hookean Energy. Several recent works have focused on improving the stability of Neo-Hookean energy [Ogden 1997]. To improve robustness to mesh inversion and rest-state stability, Smith et al. [2018] proposed the stable Neo-Hookean energy and applied their analysis to stabilize other hyperelastic variants [Kim and Eberle 2022; Lin et al. 2022]. Chen et al. [2024] further showed the high non-convexity of the stable Neo-Hookean energy stems from high Poisson's ratio and large volume change, and proposed an eigenvalue filtering scheme to stabilize the optimization.

3 BACKGROUND

In many computer graphics applications, such as quasistatic hyperelastic simulation, an optimization problem arises:

$$\min_{\mathbf{x}} f(\mathbf{x}) , \tag{1}$$

where f denotes a (twice differentiable) energy with parameters $\mathbf{x}$. We start by briefly reviewing two seemingly separate methods for solving this optimization problem—namely, the *projected Newton's method* and *trust-region method*—to lay out the foundation for our contributions, which revolve around a formal connection between these two methods (which we will develop in Sec. 4). Readers familiar with these methods may directly jump to Sec. 4 instead.

Newton's method. In many cases, one can solve Eq. (1) with Newton's method by taking the second-order Taylor expansion of $f(\mathbf{x})$:

$$\tilde{f}(\mathbf{x} + \mathbf{u}) \approx f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u}, \qquad (2)$$

where $\mathbf{g} = \nabla f(\mathbf{x})$ and $\mathbf{H} = \nabla^2 f(\mathbf{x})$ are the gradient and the Hessian of f, respectively. Then, the optimal update $\mathbf{u}$ based on the quadratic approximation $\tilde{f}$ can be obtained by solving a linear system

$$\mathbf{u} = -\mathbf{H}^{-1}\mathbf{g}. \qquad (3)$$

It is assumed that $\mathbf{u}$ is an update direction that leads to a decrease of the energy f, and thus iterations between the above two steps will converge to a local minimum. More precisely, this assumption requires that **1)** the Hessian $\mathbf{H}$ to be *positive definite* and **2)** the second-order expansion $\tilde{f}$ being a *close* quadratic approximation of the energy function f. Unfortunately, these requirements do not always hold in practice.

3.1 Projected Newton's Method

Regarding the requirement 1), if $\mathbf{H}$ is indefinite or negative definite, even if $\tilde{f}$ is a close approximation of f, the update $\mathbf{u}$ may still be an *ascent* direction, because their *critical points* $\partial \tilde{f}/\partial \mathbf{u} = 0$ are not at the minimum of $\tilde{f}$.

As a response to this situation, the *Projected Newton's method* approximates $\mathbf{H}$ in Eq. (2) with a positive definite matrix. A straightforward approach performs eigenvalue analysis on $\mathbf{H}$, modifies the negative eigenvalues to positive ones, and then reconstructs a positive definite matrix using the updated eigenvalues. Yet, for large matrices, this process is computationally intractable.

Eigenvalue clamping. In finite element simulations, a more efficient approach exists: Notice that the (global) Hessian $\mathbf{H}$ is assembled by summing up the Hessian matrix $\mathbf{H}_i$ from each finite element i, that is,

$$\mathbf{H} = \sum_i \mathbf{P}_i^\top \mathbf{H}_i \mathbf{P}_i, \qquad (4)$$

where $\mathbf{P}_i$ denotes the selection matrix that maps the per-element indices to the global indices. Leveraging this property, Teran et al. [2005] proposed to project each per-element Hessian $\mathbf{H}_i$ to a positive definite one $\mathbf{H}_i^+$. Their approach clamps the eigenvalues λ_k of each $\mathbf{H}_i$ to zero (or a small positive value ϵ) using

$$\lambda_k^+ = \begin{cases} \epsilon & \text{if } \lambda_k \le \epsilon, \\ \lambda_k & \text{otherwise,} \end{cases} \qquad (5)$$

and computes $\mathbf{H}_i^+$ with the clamped eigenvalues. This approach requires only a tiny eigenvalue analysis for each finite element,
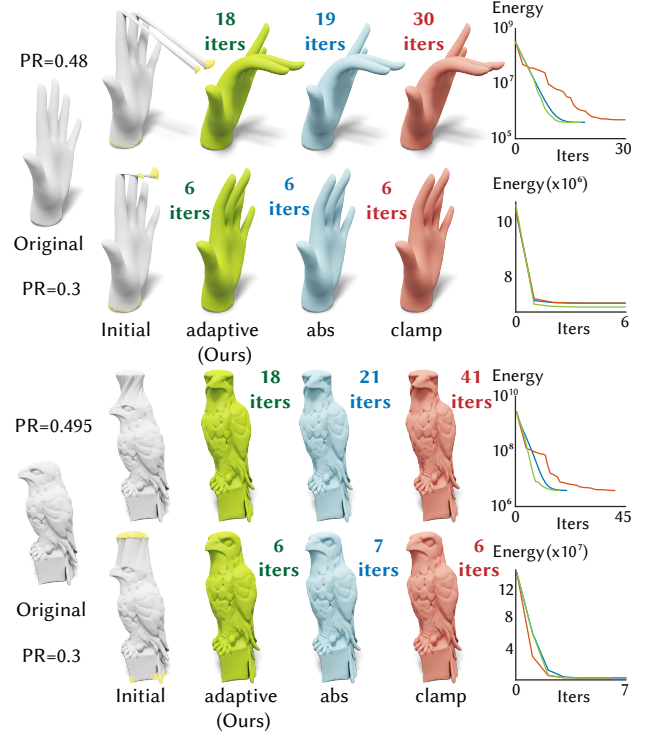

Fig. 3. Two other examples in experiments similar to Fig. 2. Here we compare our method with others under bending (top) and twisting (bottom) deformations. In all cases, our method requires the least number of iterations to converge.

all of which can be computed in parallel, and it guarantees the reconstructed global Hessian $\mathbf{H}^+$ from

$$\mathbf{H}^+ = \sum_i \mathbf{P}_i^\top \mathbf{H}_i^+ \mathbf{P}_i, \qquad (6)$$

to be positive definite [Rockafellar 1970], leading to a more stable Newton's solver for minimizing f. Even though the negative eigenvalues of per-element Hessian are clamped to zero, the additive contributions of neighboring elements and boundary conditions always lead to a positive-definite global Hessian [Teran et al. 2005].

Absolute eigenvalue projection. For a highly nonconvex energy landscape—such as one that arises in simulation of nearly incompressible materials, eigenvalue clamping (Eq. (5)) could lead to unstable search directions and an excessively large number of iteration steps [Chen et al. 2024].

This issue stems from the violation of the requirement 2) stated above. For a highly nonconvex energy function f, the quadratic approximation $\tilde{f}$ tends to be a poor approximation of f. Chen et al. [2024] further proposed a remedy for this issue by projecting the negative eigenvalues of each $\mathbf{H}_i$ to their absolute values:

$$\lambda_k^+ = |\lambda_k|. \qquad (7)$$

They demonstrated that this absolute-value projection strategy leads to orders of magnitude speed-up in the best case. But in other nearly convex cases (e.g., materials involving minor volume changes), this

strategy can slightly damp the convergence along the negative curvature directions.

In short, how to choose the projection strategy for any given simulation scenario remains to be an open question.

3.2 (Generalized) Trust-Region Method

Separate from the projected Newton's method, *Trust-region methods* [Conn et al. 2000; Sorensen 1982] were proposed in response to the violation of the requirement 2) stated above. The key idea here is to identify a *radius* around the current solution $\mathbf{x}$ where the quadratic function $\tilde{f}$ is a sufficiently tight approximation of f. Then, the update $\mathbf{u}$ (including both the update direction and the step size) is computed to be the optimal solution within this "trust-worthy" region [Moré and Sorensen 1983]. Formally, the trust region method can be written as a constrained optimization problem:

$$\min_{\mathbf{u}} \quad \tilde{f}(\mathbf{x} + \mathbf{u}) = f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u}$$
$$\text{s.t.} \quad \|\mathbf{u}\|^2 = \mathbf{u}^\top \mathbf{u} \leq \Delta \tag{8}$$

where $\Delta \in \mathbb{R}$ denotes the squared ball radius of the trust region. A key component in trust region methods lies in dynamic adjustment of the trust region size Δ to ensure sufficiently small error coming from the quadratic approximation.

Vanilla trust region methods measure the radius based on the *Euclidean ball* distance $\|\mathbf{u}\|^2 = \mathbf{u}^\top \mathbf{u}$. But in practice, many problems favor measuring distance by an underlying (Riemannian or non-Riemannian) metric. In other words, the bound should adapt to the underlying energy landscape. This observation motivates the developement of the *generalized trust region method* [Moré 1993; Pong and Wolkowicz 2014] which replace the Euclidean distance $\|\mathbf{u}\|$ constraint by a quadratic measure:

$$\min_{\mathbf{u}} \quad \tilde{f}(\mathbf{x} + \mathbf{u}) = f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u}$$
$$\text{s.t.} \quad \Delta_{\text{lower}} \leq \mathbf{u}^\top \mathbf{A}\mathbf{u} + \mathbf{u}^\top \mathbf{b} \leq \Delta_{\text{upper}}. \tag{9}$$

where $\mathbf{A}$ is a symmetric matrix that could be non-positive definite and $\Delta_{\text{lower}}/\Delta_{\text{upper}}$ denote the lower/upper bound of the trust region.

4 OUR METHOD

The efficacy of eigenvalue filtering strategies discussed in Sec. 3.1 depends on specific simulation scenarios [Chen et al. 2024]. Interestingly, we discover that these strategies are fundamentally connected, and they can be viewed in an unified framework, which in turn enables automatic adaptation of eigenvalue filtering strategy during the Newton's iterations. We now describe this unified framework.

4.1 Per-element Generalized Trust-region Newton Method

The starting point of our development is the generalized trust-region method (Sec. 3.2). Consider a specific instance of the generalized trust-region problem (9), wherein the lower and upper bounds have the same value but with opposite signs (i.e., $\Delta_{\text{upper}} = -\Delta_{\text{lower}} = \Delta$). This trust-region problem has the following form:

$$\min_{\mathbf{u}} \quad f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \mathbf{u}^\top \mathbf{H}\mathbf{u}$$
$$\text{s.t.} \quad -\Delta \leq \mathbf{u}^\top \mathbf{A}\mathbf{u} + \mathbf{u}^\top \mathbf{b} \leq \Delta. \tag{10}$$
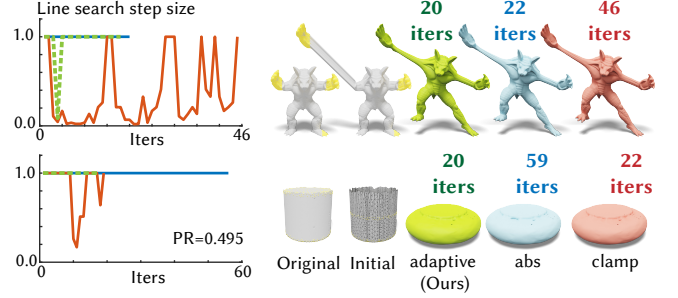
Fig. 4. We demonstrate the effectiveness of our descent direction through the line search step size during the optimization. Our trust-region based adaptive scheme often requires only few iterations of line search to find a suitable step size.

where the energy is the quadratic approximation of a simulation energy f (see Eq. (2)). We use $\mathbf{g}$ to denote the gradient and $\mathbf{H}$ to denote the Hessian. If we choose $\mathbf{b} = 0$, $\mathbf{A} = \mathbf{H}$, the two inequalities (lower and upper bounds) can be re-written as a single inequality with the absolute value. These specifications lead to the following instance of generalized trust-region problem:

$$\min_{\mathbf{u}} \quad f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \mathbf{u}^\top \mathbf{H}\mathbf{u}$$
$$\text{s.t.} \quad |\mathbf{u}^\top \mathbf{H}\mathbf{u}| \leq \Delta. \tag{11}$$

In finite element simulation, the Hessian $\mathbf{H}$ can often be constructed by summing up per-element Hessians $\mathbf{H}_i$ from each tetrahedral element i. Leveraging this property, we can re-write the constraint in (11) using the following lemma:

LEMMA 4.1. *Let* $|\mathbf{A}_e|$ *be the matrix obtained by performing per-element absolute eigenvalue projection of* $\mathbf{A}$, *i.e.,* $|\mathbf{A}_e| = \sum_i \mathbf{P}_i^\top |\mathbf{A}_i| \mathbf{P}_i$, *where* $|\mathbf{A}_i|$ *are the matrix obtained by taking the absolute value of each of the eigenvalues of* $\mathbf{A}_i$. *Then it holds that* $|\mathbf{x}^\top \mathbf{A}\mathbf{x}| \leq \mathbf{x}^\top |\mathbf{A}_e|\mathbf{x}$.

PROOF. See App. A. □

By applying Lemma A.1, we have the following relationship:

$$\mathbf{u}^\top |\mathbf{H}_e|\mathbf{u} \leq \Delta \implies |\mathbf{u}^\top \mathbf{H}\mathbf{u}| \leq \Delta, \tag{12}$$

from which Eq. (11) can be re-written as

$$\min_{\mathbf{u}} \quad f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} + \mathbf{u}^\top \mathbf{H}\mathbf{u}$$
$$\text{s.t.} \quad \mathbf{u}^\top |\mathbf{H}_e|\mathbf{u} \leq \Delta. \tag{13}$$

This reformulation is a more conservative version of Eq. (11); it "tightens" the size of trust region, but still ensures that the search radius imposed by the constraint satisfies the original problem Eq. (11).

During Newton's iterations, the trust-region size Δ may change. When Δ is sufficiently large, the minimum point of the unconstrained quadratic approximation is within the trust region, and this step is the same as the original Newton step. But more often, the unconstrained quadratic minimizer lies outside the trust region. Then, the constrained minimizer will occur on the trust-region boundary.

Suppose Δ is sufficiently small so that the constrained minimizer occurs on the trust-region boundary. In this situation, we can replace the inequality constraint in Eq. (13) with an equality constraint $\mathbf{u}^\top |\mathbf{H}_e|\mathbf{u} = \Delta$, and use Lagrangian multipliers to obtain its solution:

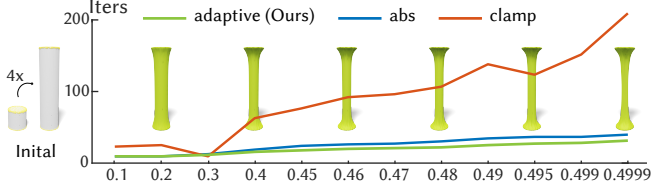$$\mathbf{g} + 2\mathbf{H}\mathbf{u} + 2\lambda|\mathbf{H}_e|\mathbf{u} = 0. \tag{14}$$

Fig. 5. Here, the cylinder is stretched to 4× of its rest length. We increases the cylinder's Poisson's ratio (x-axis) and test the number of iterations (y-axis) needed for convergence in our method and the comparing methods. Across the entire range of Poisson's ratios, out method consistently outperforms others.

Here, the Lagrangian multiplier λ must satisfy $\lambda \geq 0$. It can be shown that $\lambda = 0$ corresponds to the original Newton step Eq. (3) after a line search (see more details in Sec. 4.2.1). In other words, although we arrive Eq. (14) by assuming the constrained minimizer occurs on the trust-region boundary, Eq. (14) itself also covers the case where the unconstrained minimizer is within the trust region.

Solving for Eq. (14) gives a Newton step of the following form (up to a scalar folded into line search):

$$\begin{aligned}
\mathbf{u} &= -(\mathbf{H} + \lambda |\mathbf{H}_e|)^{-1} \mathbf{g} \\
&= -\left((1 + \lambda) \left(\frac{1}{1 + \lambda} \mathbf{H} + \frac{\lambda}{1 + \lambda} |\mathbf{H}_e| \right) \right)^{-1} \mathbf{g}
\end{aligned} \tag{15}$$

With a change of variable $w = \frac{\lambda}{1+\lambda}$ (and folding $(1 + \lambda)$ into the line search), we can see that the step is a linear combination of the unprojected Newton step and the per-element absolute eigenvalue projected Newton step:

$$\mathbf{u} = -((1 - w)\mathbf{H} + w|\mathbf{H}_e|)^{-1} \mathbf{g} \tag{16}$$

As the Lagrangian multiplier λ must satisfy $\lambda \geq 0$, the weight w must satisfy $0 \leq w \leq 1$. To ensure the resulting Newton step is a descent direction, we also require $(1 - w)\mathbf{H} + w|\mathbf{H}_e|$ to be positive definite.

Our derivation above can be interpreted as a generalized trust-region Newton method [Boyd and Vandenberghe 2004; Nocedal and Wright 2006] with the per-element extension, equality constraint enforcement and an additional line search, or a regularized Newton's method [Steck and Kanzow 2023; Ueda 2014; Zhang and Ni 2015] regularized by a generalized trust region. These extensions retain the same computational cost as the original Newton iteration while benefiting from the stability of the trust-region based method.

4.2 Generalized Trust-region Newton Step

Perhaps surprisingly, different choices of w in Eq. (16) lead to different pre-existing eigenvalue filtering strategies. Here, we enumerate three typical choices of w and their corresponding eigenvalue filtering strategies:

4.2.1 Unprojected Newton step.
When $w = 0$, i.e., $\lambda = 0$, and the positive-definiteness constraint is satisfied, the step reduces to the traditional (unprojected) Newton step:

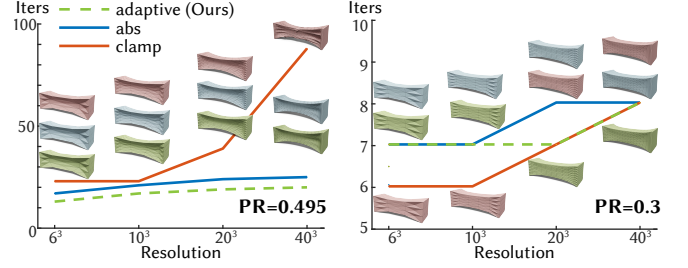$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{g} \tag{17}$$

Fig. 6. We plot the number of iterations needed for convergences with respect to different mesh resolutions. Note that in the simple case (bottom plot), our method may takes one more Newton iteration than eigenvalue clamping, simply because our method always starts from the absolute eigenvalue projection in its first Newton iteration.

4.2.2 Projected Newton with eigenvalue clamping [Teran et al. 2005].

LEMMA 4.2. *Let $\mathbf{A}_e^+$ be the matrix obtained by performing eigenvalue clamping (with threshold 0) of each $\mathbf{A}_i$, i.e., $\mathbf{A}_e^+ = \sum_i \mathbf{P}_i^\top \mathbf{A}_i^+ \mathbf{P}_i$, where $\mathbf{A}_i^+$ are the matrix obtained by clamping each eigenvalue of $\mathbf{A}_i$ to 0. Then it holds that $\mathbf{u}^\top \mathbf{A} \mathbf{u} + \mathbf{u}^\top |\mathbf{A}_e| \mathbf{u} = 2\mathbf{u}^\top \mathbf{A}_e^+ \mathbf{u}$.*

PROOF. See App. B. □

When $w = 0.5$, i.e., $\lambda = 1$, using Lemma B.1, the step reduces to the projected Newton step with eigenvalue clamping:

$$\begin{aligned}
\mathbf{u} &= -(\frac{1}{2}\mathbf{H} + \frac{1}{2}|\mathbf{H}_e|)^{-1} \mathbf{g} \\
&= -(\mathbf{H}_e^+)^{-1} \mathbf{g}
\end{aligned} \tag{18}$$

4.2.3 Projected Newton with absolute eigenvalue projection [Chen et al. 2024].
When $w = 1$, i.e., $\lambda \to \infty$, the step reduces to the projected Newton step with absolute eigenvalue projection:

$$\mathbf{u} = -|\mathbf{H}_e|^{-1} \mathbf{g} \tag{19}$$

Note that is also equivalent to using the first-order model and quadratic constraint with Hessian distance measure:

$$\begin{aligned}
\min_{\mathbf{u}} \quad & f(\mathbf{x}) + \mathbf{g}^\top \mathbf{u} \\
\text{s. t.} \quad & \mathbf{u}^\top |\mathbf{H}_e| \mathbf{u} \leq \Delta.
\end{aligned} \tag{20}$$

which, following a derivation similar to [Dauphin et al. 2014], leads to a step in the form Eq. (19).

4.3 Adaptive Per-element Projection

Now, how should we choose the weight w in Eq. (16) to achieve stable energy descent? To address this question, we seek inspiration from how the trust-region size is adjusted in trust-region method literature [Nocedal and Wright 2006]. A typical strategy is to increase the trust region size when original function is well-approximated by the quadratic form, and decrease the trust region size otherwise.

In light of this, we choose w based on how well the quadratic form (in Eq. (13)) approximates the original energy function f:

(1) If the quadratic form approximates the original energy well, then it is safe to allow a large trust-region size. In the most extreme case where an infinite trust-region size is used, we can just remove the trust-region constraint in Eq. (13), which corresponds to the unprojected Netwon step in Sec. 4.2.1 (i.e.,

$w = 0$). However, to apply $w = 0$, we need to ensure the global Hessian to be PSD. Therefore, we slightly relax it and apply $w = 0.5$, which corresponds to eigenvalue clamping—the strategy that ensures a PSD Hessian.

(2) If the quadratic fitting quality is inadequate, then a small trust region size should be used. In the most extreme case where a near-zero trust region size is used, we will have $\lambda \to \infty$ and thus the approximation model falls back to the first-order model, which corresponds to the absolute eigenvalue projection strategy. This ensures sufficient regularization along the negative eigenvalue directions when the quadratic approximation is far from the original function.

Therefore, we look for a ratio to quantify the agreement between the quadratic model and the original function, and adaptively adjust the weight w based on this ratio. In the trust region literature [Conn et al. 2000; Sorensen 1982], a classic choice to measure this agreement is to use the ratio

$$\rho = \frac{f(\mathbf{x}) - f(\mathbf{x} + \mathbf{u})}{\tilde{f}(\mathbf{x}) - \tilde{f}(\mathbf{x} + \mathbf{u})} \qquad (21)$$

where $\tilde{f}(\mathbf{x}) = f(\mathbf{x}) + \mathbf{g}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x}$ is the quadratic approximation of $f(\mathbf{x})$. When ρ is close to 1, the quadratic model is a good approximation of the original function, and a large trust region size can be allowed. When ρ is far from 1 (e.g., close to zero or even negative), the quadratic model cannot accurately estimate the local landscape of the original function, and a small trust region size should be used.

Inspired by this, we propose to use the ratio ρ to adaptively adjust the weight w in Eq. (16) based on the quality of the quadratic approximation:

$$w = \begin{cases} 0.5, & |\rho - 1| \le \epsilon \\ 1, & |\rho - 1| > \epsilon \end{cases}, \qquad (22)$$

where ϵ is a small positive constant between 0.01 and 0.1.

Our discrete w is a practical design choice for computational efficiency. Computing the optimal w from Eq. (13) would require solving a trust-region subproblem per Newton iteration, which involves an expensive iterative solve (see Sec 4.3 of [Nocedal and Wright 2006]). Therefore, instead of solving the exact trust-region subproblems, we opt to focus on a discrete set of classical w and pick one based on the trust region ratio.

In other words, let Λ be the eigenvalues of the per-element Hessian, we perform the following element-wise projection:

$$\Lambda^+ = \begin{cases} \max(\Lambda, 0), & |\rho - 1| \le \epsilon \\ |\Lambda|, & |\rho - 1| > \epsilon \end{cases}, \qquad (23)$$

Our method adaptively chooses between the absolute eigenvalue projection and eigenvalue clamping strategy based on the current energy landscape, ensuring sufficient regularization in the highly nonconvex regime while maintaining fast convergence in the (nearly) convex scenario.

For simplicity, we always start from the clamping strategy $|\Lambda|$ for the first Newton iteration, and then use the trust region ratio ρ_k computed using $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ to pick the projection strategy for the k-th Newton iteration.

Implementation. Our method requires only two lines of code change in the existing Projected Newton framework with line search.
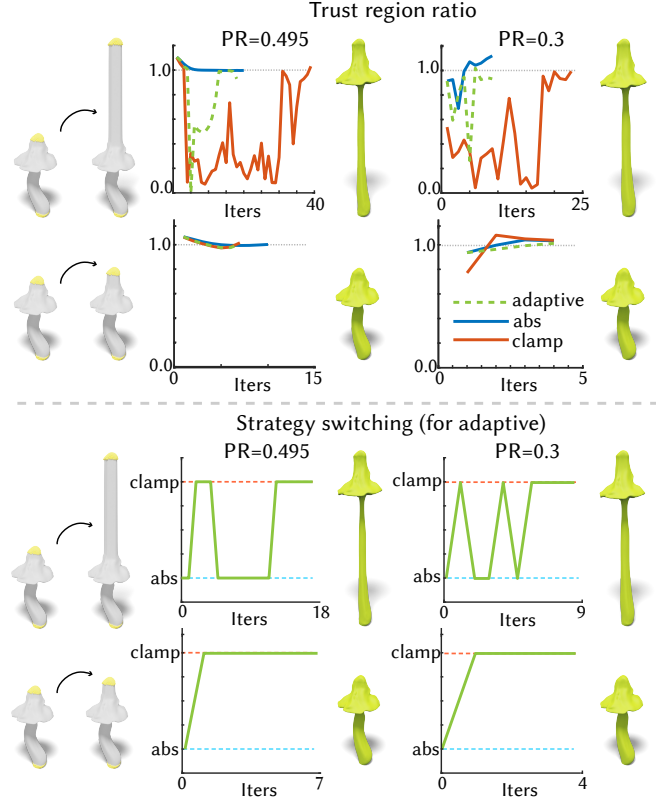
Fig. 7. We visualize the trust-region ratio ρ (for all strategies) and strategy switching trend (for adaptive) during the optimization under different deformations and Poisson's ratios. While the trust region ratio can change drastically in the early stage of optimization for the highly nonconvex energy landscape—e.g., large deformation and high Poisson's ratio, it usually approaches 1 near convergence. Thus the adaptive strategy often starts from abs strategy for the first few iterations, alternates between abs and clamp during the optimization, and falls back to clamp near convergence. On the contrary, solely using the abs strategy may excessively damp the descent direction when the quadratic approximation is still acceptable (e.g., small PR or volume change) or later becomes good (e.g., near convergence).

For completeness, we provide the full algorithm of our adaptive eigenvalue projection as below:

```
1   initialize H_proj to empty sparse matrix
2   # compute the trust region ratio
3   rho = (f(x_prev)-f(x))/(f_tilde(x_prev)-f_tilde(x))
4   foreach element i:
5       # Pi*x selects element i's local variables from x
6       either:
7           Hi = constructLocalHessian(i,Pi*x)
8           Λ,U = eig(Hi)
9       or:
10          Λ,U = analyticDecomposition(i,Pi*x)
11      # our adaptive eigenvalue projection
12      Λ = (|rho - 1|<epsilon) ? max(Λ,0) : abs(Λ)
13      # compute the projected local Hessian
14      Hi_proj = U*Λ*U'
15      # accumulate into global Hessian
16      H_proj += Pi*Hi_proj*Pi'
```
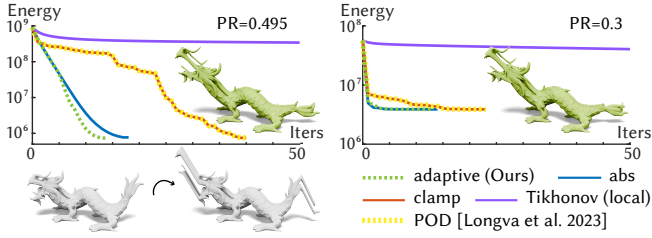
Fig. 8. We compare the energy convergence curve in our approach, the Projection-On-Demand (POD) [Longva et al. 2023] and per-element Tikhonov regularization. We also note that, separate from the optimization convergence performance, POD may require additional Cholesky decompositions to check the positive definiteness of the Hessian.
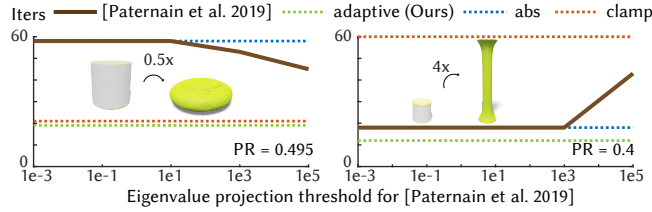

Fig. 9. We compare our method to a per-element version of the eigenvalue projection scheme in [Paternain et al. 2019]. Due to the large variation across local elements' eigenvalue magnitudes (under different deformations and Poisson's ratios), it is unclear how to set an eigenvalue projection threshold for [Paternain et al. 2019]. Here, we sweep through a range of threshold values (x-axis) and plot the needed number of iterations in each setting. Our method is consistently more efficient.

5 RESULTS

We evaluate our method on a diverse set of examples, including different deformations, physical parameters, geometries, mesh resolutions, and elastic energies. Furthermore, we perform extensive experiments on the TetWild Thingi10k dataset [Hu et al. 2018; Zhou and Jacobson 2016] to demonstrate our speedup over existing Hessian projection schemes (Fig. 13 and Fig. 14).

We implement our method in C++ using the TinyAD library [Schmidt et al. 2022] for automatic differentiation, and perform our experiments using a MacBook Pro with an Apple M2 processor and 24GB of RAM.

In our hyperelastic simulation, we use the stable Neo-Hookean energy [Kim and Eberle 2022; Smith et al. 2018] with a Young's Modulus of 10^8 and different Poisson's ratios. We run the Projected Newton's method for a maximum of 200 iterations with a convergence criteria of 10^{-5} on the Newton decrement $0.5\mathbf{u}^\top\mathbf{g}$. Since the mesh's initial large deformation may invert some finite elements, we perform a backtracking line search without inversion check to find a suitable step size, iteratively shrinking the step size by 0.8 (starting from step size 1) until Armijo condition is satisfied. We set the trust-region ratio threshold ϵ in Eq. (23) to be 0.01 for all the examples (except compression for which ϵ is set to be 0.1).

Convergence and Stability. Our method achieves consistent speedup over existing eigenvalue projection schemes across a wide range of deformations (Fig. 2 and Fig. 3), Poisson's ratios (Fig. 5),

Table 1. We report the **average** line search iterations per Newton iteration of the teaser (Fig. 1) and the horse (Fig. 2) example. Our adaptive method maintains low line search iterations across different Poisson's ratios and deformation sizes.

	large deformation					
	teaser: Fig. 1 (top)			horse: Fig. 2 (top)		
	adaptive	abs	clamp	adaptive	abs	clamp
PR = 0.495	1.0	1.0	7.5	1.8	1.0	7.4
PR = 0.3	1.0	1.0	4.0	1.5	1.0	5.0
	small deformation					
	teaser: Fig. 1 (bottom)			horse: Fig. 2 (bottom)		
	adaptive	abs	clamp	adaptive	abs	clamp
PR = 0.495	1.4	1.0	4.1	1.0	1.0	1.0
PR = 0.3	1.0	1.0	1.0	1.0	1.0	1.0

Table 2. We measure the average per-iteration cost (including all PR and deformation combinations) in wall-clock time on the teaser example ($12k$ vertices, $44k$ tetrahedrons). Here we provide the statistics of the average time per Newton iteration, and the percentage of time spent on computing the Newton direction (including the linear solve), line search and (optionally) computing the trust-region (TR) ratio for one Newton iteration.

	adaptive	abs	clamp
Newton direction	88.6% (solve: 76.2%)	94.6% (solve: 81.7%)	80.5% (solve: 69.4%)
line search	5.4%	5.4%	19.5%
TR ratio	6.0%	/	/
avg time per Newton iter	0.117 sec	0.107 sec	0.126 sec

mesh resolutions (Fig. 6), tetrahedralizations (Fig. 11), and hyperelastic models (Fig. 10). Our approach adaptively chooses between absolute eigenvalue projection and clamping at each iteration. In the case where a specific eigenvalue projection strategy performs dominantly better than the other throughout the optimization, our adaptive approach performs at least comparable to the best-performing eigenvalue projection strategy. Note that in the case where eigenvalue clamping performs consistently better than absolute eigenvalue projection, our adaptive approach can takes one more Newton iteration than eigenvalue clamping (Fig. 6 bottom). This is just because we always start from the absolute eigenvalue projection in the first Newton iteration. Our trust-region based strategy also enables us to obtain more effective descent directions that require much fewer line search iterations to find a suitable step size (Fig. 4). We provide the statistics of the trust-region ratio trend, line search iterations, and timings in Fig. 7, Table 1 and Table 2, respectively.

Generality. In Fig. 13 and Fig. 14, we provide the statistics of our speedup over [Teran et al. 2005] and [Chen et al. 2024] under diverse deformations on 254 high-resolution, closed, genus-0 tetrahedral meshes with more than 5000 vertices. These meshes are randomly chosen from the TetWild Thingi10k dataset [Zhou and Jacobson 2016]. Our method achieves significant speedup under high Poisson's ratios and large deformations, while still being comparable to the existing eigenvalue projection schemes under small deformations and low Poisson's ratios.
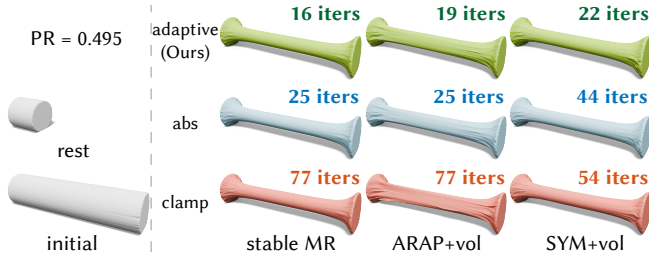
Fig. 10. Our adaptive strategy generalizes across different Neo-Hookean variants. Here in the last three columns, we experiment with different strain energies, including Mooney-Rivlin, ARAP, and Symmetric Dirichlet energy, each augmented with a volume-preservation term $(J-1)^2$.
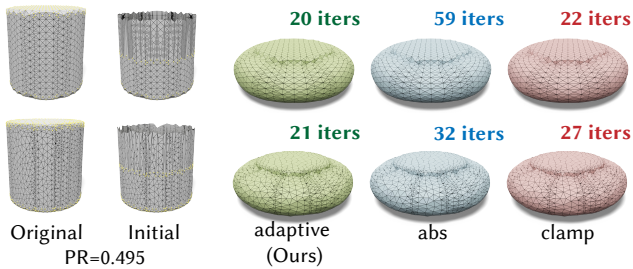

Fig. 11. Our method is robust to different tetrahedralizations. Here we tetrahedralize the same cylinder to two different tetrahedral meshes. The meshes on the top and bottom row have 6.1k and 6.6k vertices, respectively. In both settings, our method outperforms existing strategies.

Comparison. Last but not least, we compare our method with another two Hessian projection strategies, namely, the Projection-On-Demand (POD) strategy [Longva et al. 2023] in Fig. 8 and the per-element Tikhonov regularization [Paternain et al. 2019] in Fig. 9. As the POD strategy is mostly designed for dynamic simulation where the Hessian can be close to positive-definite given a small timestep, its convergence is similar to eigenvalue clamping in the quasistatic setting, but it suffers from additional Cholesky decomposition cost for the positive-definiteness check. Moreover, Tikhonov regularization [Paternain et al. 2019] requires the eigendecomposition on the global Hessian matrix, which is computationally intractable for large meshes. Therefore, we compare our approach to the per-element version of [Paternain et al. 2019] (see Fig. 9), where the local element's eigenvalues are projected to their absolute values when large than a threshold and clamped otherwise. As the eigenvalues can have drastically different magnitudes under different deformations and Poisson's ratios, it is challenging to set a universal projection threshold for local eigenvalue projection in [Paternain et al. 2019].

6 CONCLUSION & FUTURE WORK

We introduce a novel adaptive eigenvalue filtering strategy for Projected Newton's method to stabilize and accelerate the minimization of Neo-Hookean energy. Our method is simple to implement and requires only two lines of code change in the Projected Newton framework, making it easy to integrate into existing simulation pipelines. Our trust-region based framework opens up the possibility of analyzing different eigenvalue projection schemes while taking the quality of the quadratic approximation into account.
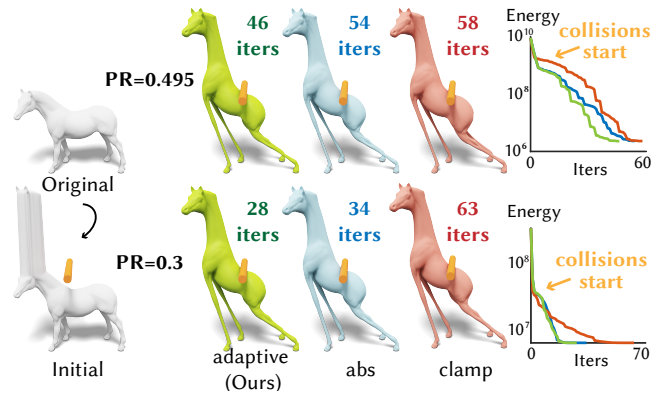
Fig. 12. We perform a collision experiment using Incremental Potential Contact (IPC) [Li et al. 2020] by placing a cylinder (orange) above the back of a horse. Note that IPC's intersection-aware line search dominates convergence and step size after collisions happen. Extending our framework to accelerate the convergence of barrier functions could be an interesting future direction.

We primarily experiment on the quasistatic simulation of stable Neo-Hookean energy in this work. Extending our framework to other hyperelastic models, collisions (Fig. 12) and dynamic simulation, such as combining our work with [Longva et al. 2023], could be a promising direction. Our choice of w eliminates the need of checking the positive-definiteness of the global Hessian, and is well-suited for quasistatic simulation where the energy is highly nonconvex. For other more convex scenarios (e.g., dynamic simulation with a small timestep), the global Hessian can sometimes be *serendipitously* positive definite even if some local Hessians are indefinite, in which case setting $w \in [0, 0.5]$ could potentially lead to faster convergence [Longva et al. 2023].

Exploration of having w as a continuous function in $[0, 1]$ for more fine-grained control could be another interesting future direction. Computing the trust-region ratio and projection strategy independently *for each element* could potentially further improve the convergence. Our method requires picking a threshold ϵ for the trust region ratio. Further investigation on the choice of this threshold could be beneficial, especially for the case of small compression (see Fig. 14). Our approach always starts from the absolute eigenvalue projection for the first Newton iteration. Deriving a better strategy for the initial eigenvalue projection could potentially further improve the performance.
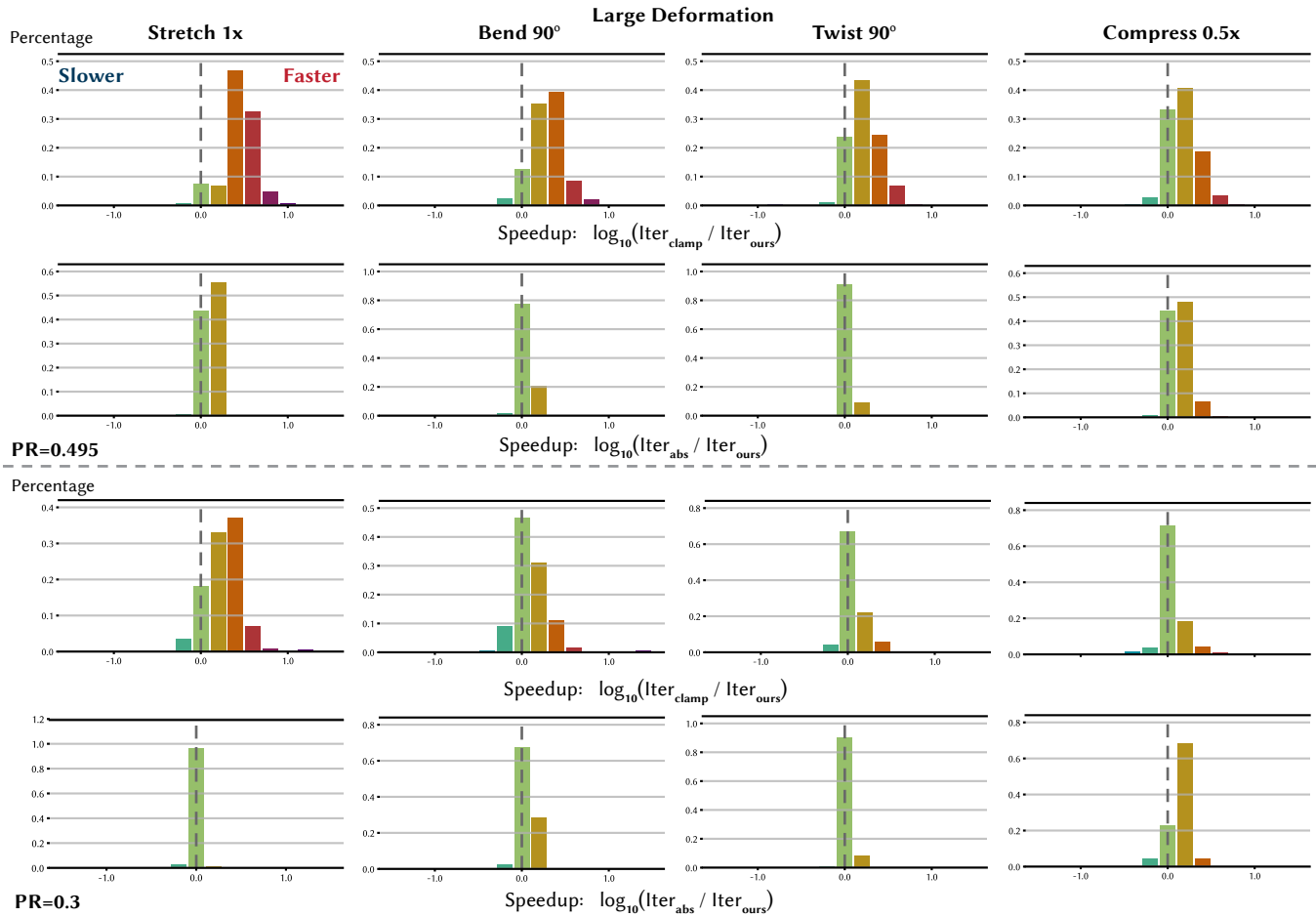
ACKNOWLEDGMENTS

Fig. 13. Histogram of the speedup of our adaptive method over the absolute eigenvalue projection [Chen et al. 2024] and eigenvalue clamping [Teran et al. 2005] on TetWild Thingi10k dataset. Here, we test with a diverse set of *large* deformations (shown in each column). First row: speedups over eigenvalue clamping with a Poisson's ratio 0.495. Second row: speedups over absolute eigenvalue projection with a Poisson's ratio 0.495. Third row: speedups over eigenvalue clamping with a Poisson's ratio 0.3. Fourth row: speedups over absolute eigenvalue projection with a Poisson's ratio 0.3.

REFERENCES

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Honglin Chen, Hsueh-Ti Derek Liu, David I.W. Levin, Changxi Zheng, and Alec Jacobson. 2024. Stabler Neo-Hookean Simulation: Absolute Eigenvalue Filtering for Projected Newton. In *ACM SIGGRAPH 2024 Conference Proceedings*.

Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. 2000. *Trust Region Methods*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898719857

Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. MIT Press, Cambridge, MA, USA, 2933–2941.

Xiao-Ming Fu and Yang Liu. 2016. Computing inversion-free mappings by simplex assembly. *ACM Trans. Graph.* 35, 6 (2016), 216:1–216:12.

P.E. Gill, W. Murray, and M.H. Wright. 1981. *Practical Optimization*. Academic Press. https://books.google.com/books?id=xUzvAAAAMAAJ

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages. https://doi.org/10.1145/3197517.3201353

Theodore Kim and David Eberle. 2022. Dynamic Deformables: Implementation and Production Practicalities (Now with Code!). In *ACM SIGGRAPH 2022 Courses* (Vancouver,

British Columbia, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 259 pages. https://doi.org/10.1145/3532720.3535628

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. 2020. Incremental Potential Contact: Intersection- and Inversion-free Large Deformation Dynamics. *ACM Trans. Graph. (SIGGRAPH)* 39, 4, Article 49 (2020).

Huancheng Lin, Floyd M. Chitalu, and Taku Komura. 2022. Isotropic ARAP Energy Using Cauchy-Green Invariants. *ACM Trans. Graph.* 41, 6, Article 275 (nov 2022), 14 pages. https://doi.org/10.1145/3550454.3555507

Andreas Longva, Fabian Löschner, José Antonio Fernández-Fernández, Egor Larionov, Uri M. Ascher, and Jan Bender. 2023. Pitfalls of Projection: A study of Newton-type solvers for incremental potentials. arXiv:2311.14526

Jorge J Moré. 1993. Generalizations of the trust region problem. *Optimization methods and Software* 2, 3-4 (1993), 189–209.

Jorge J. Moré and D. C. Sorensen. 1983. Computing a Trust Region Step. *SIAM J. Sci. Stat. Comput.* 4, 3 (sep 1983), 553–572. https://doi.org/10.1137/0904038

Jorge Nocedal and Stephen J. Wright. 2006. Numerical Optimization. (2006).

Raymond W Ogden. 1997. *Non-linear elastic deformations*. Courier Corporation.

Santiago Paternain, Aryan Mokhtari, and Alejandro Ribeiro. 2019. A Newton-Based Method for Nonconvex Optimization with Fast Evasion of Saddle Points. *SIAM Journal on Optimization* 29, 1 (2019), 343–368. https://doi.org/10.1137/17M1150116

Ting Kei Pong and Henry Wolkowicz. 2014. The generalized trust region subproblem. *Comput. Optim. Appl.* 58, 2 (jun 2014), 50 pages. https://doi.org/10.1007/s10589-013-9635-7
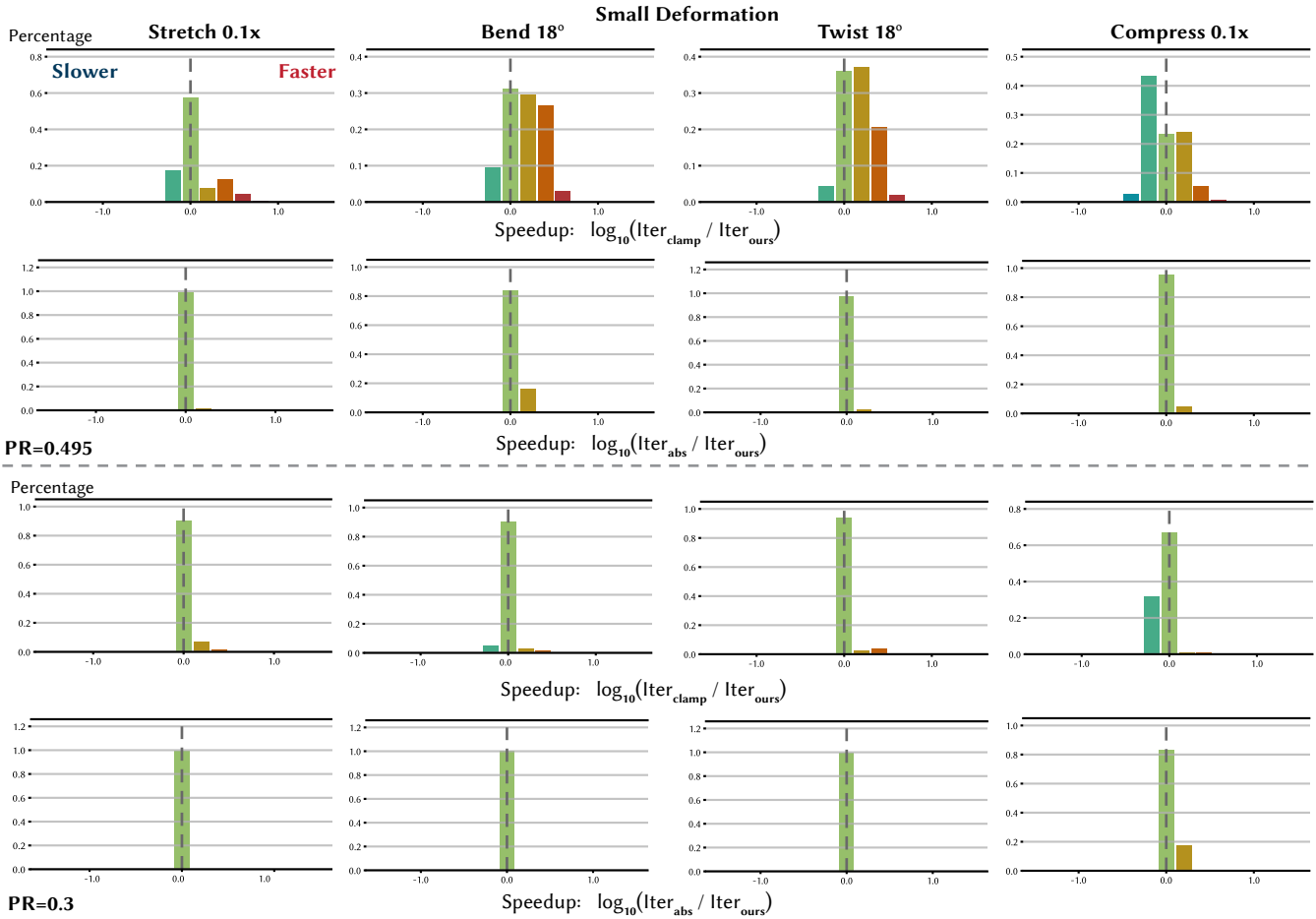
Fig. 14. Histogram of the speedup of our adaptive method in settings similar to Fig. 13 but with much *smaller* deformations. Even in these simple scenarios, our method outperforms eigenvalue clamping under a large Poisson's ratio, and performs comparably in other cases.

Ralph Tyrell Rockafellar. 1970. *Convex Analysis*. Princeton University Press, Princeton. https://doi.org/doi:10.1515/9781400873173

Patrick Schmidt, Janis Born, David Bommes, Marcel Campen, and Leif Kobbelt. 2022. TinyAD: Automatic Differentiation in Geometry Processing Made Simple. *Computer Graphics Forum* 41, 5 (2022).

Breannan Smith, Fernando De Goes, and Theodore Kim. 2018. Stable Neo-Hookean Flesh Simulation. *ACM Trans. Graph.* 37, 2, Article 12 (mar 2018), 15 pages. https://doi.org/10.1145/3180491

D. C. Sorensen. 1982. Newton's Method with a Model Trust Region Modification. *SIAM J. Numer. Anal.* 19, 2 (1982), 409–426. https://doi.org/10.1137/0719026

Daniel Steck and Christian Kanzow. 2023. Regularization of Limited Memory Quasi-Newton Methods for Large-Scale Nonconvex Minimization.

Joseph Teran, Eftychios Sifakis, Geoffrey Irving, and Ronald Fedkiw. 2005. Robust Quasistatic Finite Elements and Flesh Simulation. In *ACM/Eurographics Symposium on Computer Animation (SCA)*, K. Anjyo and P. Faloutsos (Eds.). 181–190. http://graphics.cs.wisc.edu/Papers/2005/TSIF05

Kenji Ueda. 2014. A Regularized Newton Method without Line Search for Unconstrained Optimization. *Computational Optimization and Applications* 59 (10 2014). https://doi.org/10.1007/s10589-014-9656-x

Alex Wang and Fatma Kilinc-Karzan. 2020. The generalized trust region subproblem: solution complexity and convex hull results. *Mathematical Programming* 191 (10 2020), 1–42. https://doi.org/10.1007/s10107-020-01560-8

Hao Zhang and Qin Ni. 2015. A new regularized quasi-Newton algorithm for unconstrained optimization. *Appl. Math. Comput.* 259 (2015), 460–469. https://doi.org/10.1016/j.amc.2015.02.032

Qingnan Zhou and Alec Jacobson. 2016. Thingi10K: A Dataset of 10,000 3D-Printing Models. *arXiv preprint arXiv:1605.04797* (2016).

A PROOF OF LEMMA 4.1

LEMMA A.1. *Let $|\mathbf{A}_e|$ be the matrix obtained by performing per-element absolute eigenvalue projection of $\mathbf{A}$, i.e., $|\mathbf{A}_e| = \sum_i \mathbf{P}_i^\top |\mathbf{A}_i| \mathbf{P}_i$, where $|\mathbf{A}_i|$ are the matrix obtained by taking the absolute value of each of the eigenvalues of $\mathbf{A}_i$. Then it holds that $\left| \mathbf{x}^\top \mathbf{A} \mathbf{x} \right| \le \mathbf{x}^\top |\mathbf{A}_e| \mathbf{x}$.*

$$
\left| \mathbf{x}^\top \mathbf{A} \mathbf{x} \right| = \left| \mathbf{x}^\top \left(\sum_i \mathbf{P}_i^\top \mathbf{A}_i \mathbf{P}_i \right) \mathbf{x} \right|
$$

$$
= \left| \sum_i (\mathbf{P}_i \mathbf{x})^\top \mathbf{A}_i (\mathbf{P}_i \mathbf{x}) \right|
$$

$$
= \left| \sum_i \sum_k \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right) \lambda_{ik} \left(\mathbf{e}_{ik}^\top (\mathbf{P}_i \mathbf{x}) \right) \right|
$$

$$
= \left| \sum_i \sum_k \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right) \lambda_{ik} \left(\mathbf{e}_{ik}^\top (\mathbf{P}_i \mathbf{x}) \right) \right|
$$

$$
= \left| \sum_i \sum_k \lambda_{ik} \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right)^2 \right|
$$

$$
\le \sum_i \sum_k |\lambda_{ik}| \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right)^2
$$

$$
= \sum_i \sum_k \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right) |\lambda_{ik}| \left(\mathbf{e}_{ik}^\top (\mathbf{P}_i \mathbf{x}) \right)
$$

$$
= \mathbf{x}^\top \left(\sum_i \mathbf{P}_i^\top |\mathbf{A}_i| \mathbf{P}_i \right) \mathbf{x}
$$

$$
= \mathbf{x}^\top |\mathbf{A}_e| \mathbf{x}
$$

□

B PROOF OF LEMMA 4.2

LEMMA B.1. $\mathbf{u}^\top \mathbf{H} \mathbf{u} + \mathbf{u}^\top |\mathbf{H}_e| \mathbf{u} = 2 \mathbf{u}^\top \mathbf{H}_e^+ \mathbf{u}$.

Let λ_{ik}^+ and λ_{ik}^- be the positive and negative eigenvalues of each $\mathbf{A}_i$. Then we have:

$$
\mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{x}^\top |\mathbf{A}_e| \mathbf{x}
$$

$$
= \sum_i \sum_k \lambda_{ik} \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right)^2 + \sum_i \sum_k |\lambda_{ik}| \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right)^2
$$

$$
= 2 \sum_i \sum_k \lambda_{ik}^+ \left((\mathbf{P}_i \mathbf{x})^\top \mathbf{e}_{ik} \right)^2
$$

$$
= 2 \mathbf{x}^\top \mathbf{A}_e^+ \mathbf{x}.
$$

□