Variational Quasi-Harmonic Maps for Computing Diffeomorphisms

YU WANG, MINGHAO GUO, and JUSTIN SOLOMON, Massachusetts Institute of Technology, USA

Computation of injective (or inversion-free) maps is a key task in geometry processing, physical simulation, and shape optimization. Despite being a longstanding problem, it remains challenging due to its highly nonconvex and combinatoric nature. We propose computation of *variational quasi-harmonic maps* to obtain smooth inversion-free maps. Our work is built on a key observation about inversion-free maps: A planar map is a diffeomorphism if and only if it is quasi-harmonic and satisfies a special Cauchy boundary condition. We hence equate the inversion-free mapping problem to an optimal control problem derived from our theoretical result, in which we search in the space of parameters that define an elliptic PDE. We show that this problem can be solved by minimizing within a family of functionals. Similarly, our discretized functionals admit exactly injective maps as the minimizers, empirically producing inversion-free discrete maps of triangle meshes. We design efficient numerical procedures for our problem that prioritize robust convergence paths. Experiments show that on challenging examples our methods can achieve up to orders of magnitude improvement over state-of-the-art, in terms of speed or quality. Moreover, we demonstrate how to optimize a generic energy in our framework while restricting to quasi-harmonic maps.

CCS Concepts: • **Mathematics of computing** → *Partial differential equations.*

Additional Key Words and Phrases: geometric graph theory, diffeomorphism, inverse problems, optimal control, partial differential equations (PDEs), PDE-constrained optimization, σ-harmonic functions, quasi-conformal maps, geometric variational problem, Dirichlet-to-Neumann.

ACM Reference Format:
Yu Wang, Minghao Guo, and Justin Solomon. 2023. Variational Quasi-Harmonic Maps for Computing Diffeomorphisms. *ACM Trans. Graph.* 42, 4, Article 130 (August 2023), 26 pages. https://doi.org/10.1145/3592105

1 INTRODUCTION

Inversion-free mapping is a fundamental task in geometry processing, scientific computing, computer graphics, and vision [Fu et al. 2021]. For example, in physical simulation and shape optimization, maps with inverted elements—and hence negative element volumes—cannot be realized or manufactured; in texture mapping, surface parameterizations must be injective to avoid content blending. Other applicable scenarios requiring maps to be inversion-free include shape deformation, correspondence, collision avoidance (§11.2), fabrication, design, and mesh quality improvement.

The inversion-free mapping problem goes by many names. The goal typically is to compute a *map*, *parameterization*, *shape*, or *deformation* that is *inversion-free*, *injective*, *bijective*, *foldover/flip-free*, *orientation-preserving*, and/or *diffeomorphic/homeomorphic*. Subtle nuances shade the differences between these nearly-identical instances, as discussed in §4.

Barrier-type methods and variants of Tutte's embedding algorithm comprise two major approaches to the inversion-free mapping problem. Barrier-type methods suffer from numerical instability and slow progress on extreme examples (§10), and Tutte's embedding is limited to mappings onto convex domains (§5).

As an alternative to these strategies, we present a fast and robust method for injective surface mapping by extending Tutte's embedding to nonconvex domains. Our approach relies on a global condition for map injectivity, eliminating the need for inversion-preventing line search, which yields larger step sizes that are prohibited in barrier and/or interior point methods. Our method can be understood as optimizing a differentiable variant of Tutte's embedding algorithm, searching for edge weights that will produce inversion-free mappings onto nonconvex domains. Our approach extends recent advances in computing skinning weights [Wang and Solomon 2021], built on the strong interplay between map topology, partial differential equations, global/conformal geometry, and geometric graph theory. Empirically, our method passes a challenging dataset for inversion-free maps [Du et al. 2020], joining relatively few works [Du et al. 2020; Garanzha et al. 2021] and achieving significant improvements on challenging examples.

Contributions. Contributions of our paper include:

- We propose a necessary and sufficient condition for a smooth map to be diffeomorphic, which equates mapping to the optimal control of PDEs to satisfy a Cauchy condition (§5).
- We discretize our model in a fashion that yields a necessary and sufficient characterization of discrete injectivity (§6).
- We propose efficient numerical schemes for solving inverse problems with the Cauchy condition (§7) using a tensor parameterization prioritizing robust convergence paths (§8).
- We demonstrate *quasi-harmonization* of deformation energies, enabling the optimization of generic energies within the space of quasi-harmonics (§7,§11).
- We propose injectivity-promoting functionals such as a parametric family of Poisson functionals and Dirichlet energies (§7).

2 RELATED WORK

Inversion-free mapping is an active field drawing interdisciplinary attention in physical simulation, geometry processing, and scientific computing.

Barrier-type methods. Barrier methods (and interior point methods) traverse the interior of the feasible region. Barrier-type methods are popular in graphics for injective mapping. For example, Schüller et al. [2013] use a log determinant barrier term for this task. Liu et al. [2016] solve a sequence of convex programs that minimize a quadratic elastic energy while staying in the feasible region. Smith and Schaefer [2015] design a line search method, deriving the maximal step size such that no flip will be introduced. Their line search

Authors' address: Yu Wang (corresponding author), wangyu9@csail.mit.edu; Minghao Guo, guomh2014@gmail.com; Justin Solomon, jsolomon@mit.edu, Massachusetts Institute of Technology, 77 Mass. Ave., Cambridge, MA, 02139, USA..

0730-0301/2023/8-ART130
https://doi.org/10.1145/3592105

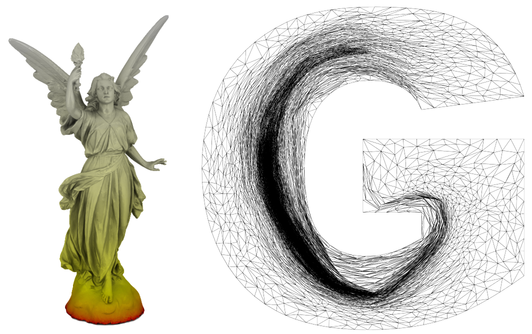


Fig. 1. Mapping the "Lucy" shape, with boundary marked in the red color, onto the nonconvex domain of letter G. Our method (tanh) minimizing a $\mathbf{A}$-Dirichlet energy using Adam finds an inversion-free map within tens of gradient steps. Despite being naïvely implemented in MATLAB, ours finds an injective map in merely 3.1 seconds at ~10% CPU rate. In contrast, the state-of-the-art Garanzha et al. [2021] using C++ and OpenMP, even at 100% CPU rate, is 300× slower in wallclock time than ours on this example. See also Figure 5.

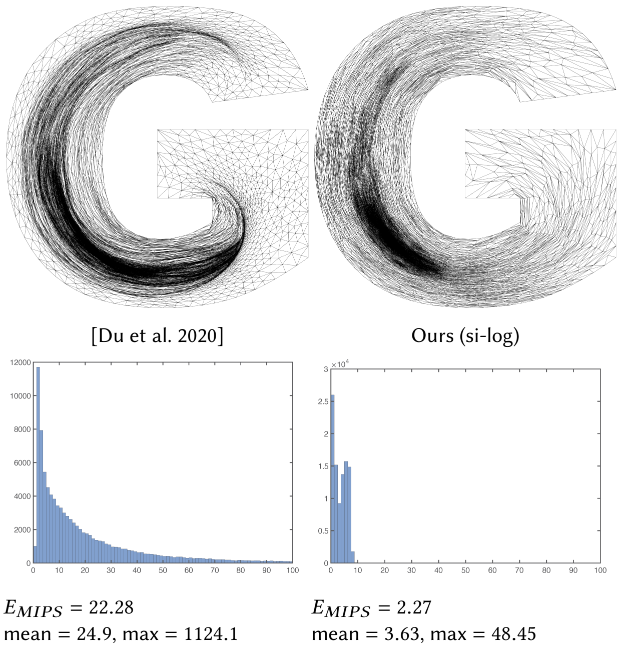


Fig. 2. Our method (si-log) finds an injective map whose MIPS energy is an order of magnitude smaller than [Du et al. 2020]. Our method (si-log), even without explicitly using any conformal energy, leads to smaller conformal distortion by any metric: Plots compare distributions of the ratio of singular values $\sigma_1/\sigma_2 \geq 1$. Our method finds an injective map with very low energy in 190 Adam-gradient steps, while even after 28000 iterations TLC+PN has an energy much larger than ours. See also Figure 5.

has been widely adopted, e.g. in [Claici et al. 2017; Fang et al. 2021; Shtengel et al. 2017].

Some methods allow infeasible points either as the initialization or as an intermediate state. As a recent example, Garanzha et al. [2021] introduce a progressive method to optimize their energy with a barrier scheduling procedure. See §10.2 for detailed discussion and comparison.

Many energies prevent inversions by exploding when the Jacobian approaches zero. Some examples include the symmetric Dirichlet energy [Schreiner et al. 2004; Smith and Schaefer 2015] and (many variants of) the Neo-Hookean energy. The MIPS energy [Hormann and Greiner 2000], often known as the Winslow functional outside graphics, standalone may not penalize inversions but can be used in barrier methods [Garanzha et al. 2021].

Fast solvers. Energy-based methods directly benefit from the advancement of general-purpose fast solvers and preconditioners for mapping and deformation [Claici et al. 2017; Kovalsky et al. 2016; Liu et al. 2017; Shtengel et al. 2017; Zhu et al. 2018].

Some solvers leverage specific structures in the energy. Scalable Locally Injective Mapping (SLIM) [Rabinovich et al. 2017] generalizes the local-global as-rigid-as-possible (ARAP) solver [Liu et al. 2008; Sorkine and Alexa 2007] to work on the symmetric Dirichlet energy by employing spatially-varying reweighting. Their weighting is a scalar field (isotropic), less general than our anisotropic weights. Brown and Narain [2021] advance in this direction by dynamically reweighting a Laplacian matrix using weights obtained from singular values of per-element Hessian; Stein et al. [2022] propose a splitting scheme based on alternating direction method of multipliers (ADMM). Fu et al. [2015] propose an inexact block coordinate descent method.

Harmonic maps. Tutte's embedding [Floater 2003a; Tutte 1963] finds interior node positions by solving a sparse Laplacian system, with a guarantee of injectivity when mapping into a planar domain with a convex boundary. It appears in the pipelines of many methods, often as an initializer for a nonlinear mapping procedure, e.g., [Rabinovich et al. 2017; Shen et al. 2019]. Outside the limited setting of mapping onto convex domains, harmonic maps can be non-injective. Quasi-harmonic maps have been considered for surface parameterization [Yoshizawa et al. 2004; Zayer et al. 2005]; these works do not consider the boundary conditions articulated in our paper and hence can fail injectivity similarly to generic harmonic maps.

Variational harmonic maps (VHM) [Ben-Chen et al. 2009] operate in the (linear) space of harmonic functions using a boundary integral representation. Deformations in this space are not necessarily injective, unless additional conditions are enforced [Chen and Weber 2017; Fargion and Weber 2022; Liao et al. 2021]. Regardless, optimizing in the space of harmonic maps excludes many viable options for the injective mapping problem.

As a natural idea, the source and target shapes can be mapped onto the same domain via harmonic maps ϕ, ψ, respectively, constructing the final map as the composition $\psi^{-1} \circ \phi$ [Weber and Zorin 2014]. Inverse harmonic maps also appear as the minimizer of the MIPS energy [Garanzha et al. 2021], when restricted to injective maps.

Conformal and extremal quasi-conformal maps. Conformal maps—those preserve angles—provide another source of injective maps widely used in parameterization and texture mapping [Ben-Chen et al. 2008; Lévy et al. 2002; Mullen et al. 2008; Sawhney and Crane 2017; Springborn et al. 2008]. Conformal maps, however, no longer exist under generic positional constraints, suggesting the use of

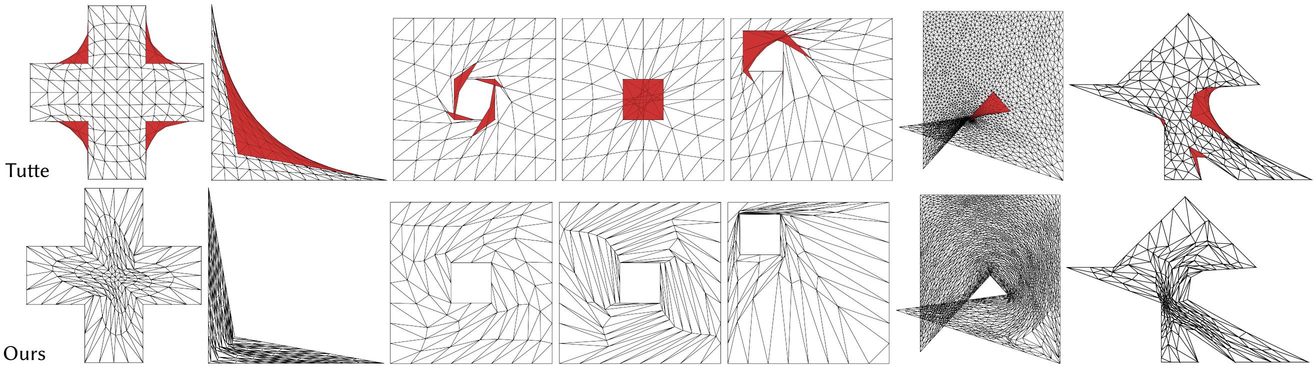


Fig. 3. Discrete harmonic mappings with cotangent weights, a.k.a. Tutte's embeddings, fail to produce injective maps onto nonconvex boundaries. Inverted triangles are marked in red. Shapes are from [Du et al. 2020; Weber and Zorin 2014]. Note (quasi) harmonic maps generally admit inversions at the concave corners of the boundary. Our method (si-log) searches for an anisotropic Laplacian under which quasi-harmonic mapping becomes injective. Similar to the case of harmonic maps, enforcing quasi-harmonicity also "pushes" inversions onto the boundary, so that we only need to detect and resolve inversions at the the boundary.

an extremal quasi-conformal map—the map with least global angle distortion [Weber et al. 2012]. Despite being bijective in the smooth setting, without remeshing these maps can lack discrete injectivity.

Lipman et al. [2012] derive closed-form formulas for extremal quasi-conformal planar deformations determined by four control points. Weber et al. [2012] compute extremal quasi-conformal maps by minimizing violation of the Beltrami equation via least-squares, alternatively optimizing the map, Beltrami coefficients, and a dilatation scalar that measures the global angular distortion. Following this direction, Teichmüller-mapping surface registration [Lui et al. 2014] and an ADMM-based splitting method [Lui and Ng 2015] have been proposed for similar formulations.

Total unsigned area. With a fixed boundary map, the total unsigned area (TUA) provides another energy term for promoting injectivity [Xu et al. 2011]. Recently, Du et al. [2020] propose Total Lifted Content (TLC) to regularize TUA, making it more robust to optimize with a customized Newton-type solver. Du et al. [2022] propose another variant, the Isometric TLC (IsoTLC), for recovering an injective map that minimizes isometric distortion rather than conformal distortion.

Global injectivity. Achieving global injectivity is often regarded as a related but somewhat different problem, since overlap or collision detection between each pair of boundary elements becomes necessary. Smooth Excess Area (SEA) extends TLC to free-boundary optimization for global injectivity, minimizing a smooth version of overlapping area [Du et al. 2021]. Tessellating the space between objects prevents inversions of an "air-mesh" [Jiang et al. 2017; Müller et al. 2015], providing a means of linking local and global injectivity (§11.2). Fang et al. [2021] put barrier energies on distances to prevent intersections.

Other ideas. Aigerman and Lipman [2013]; Kovalsky et al. [2014]; Lipman [2012] optimize in the space of orientation-preserving maps with bounded distortion. Levi and Weber [2016] show a convex characterization of the space of locally injective harmonic mappings with bounded distortion. Rather than working with vertex positions, Chen et al. [2013]; Chien et al. [2016] optimize in the space of metrics (edge lengths).

Another condition for map injectivity is that the Gaussian curvature of the target surface should always be zero, i.e. the image of the map should be *developable.* Discretely, this condition requires the adjacent angles to each interior vertex to sum to 2π, motivating Angle-Based Flattening (ABF) [Sheffer and de Sturler 2001; Sheffer et al. 2005].

Liu et al. [2018] compute a progressive sequence of bounded distortion maps using distortion energies defined on intermediate reference triangles.

Addressing a problem in a different setting, there are also methods remeshing the domain for injectivity [Campen et al. 2021; Gillespie et al. 2021; Gu et al. 2018; Shen et al. 2019; Weber and Zorin 2014].

Inverse problems and optimal control. Inverse problems find a PDE that reproduces given solution(s) [Isakov 2017; Kirsch 2011], including the distributed parameter identification problem [Richter 1981], with applications to electrical impedance tomography (EIT). Optimal control is a more general setup that measures the solution using some functional instead [Tröltzsch 2010; Wang and Solomon 2021]. Methods relevant to ours include the output least square method (OLS) directly measuring errors in the space of solutions [Frind and Pinder 1973] and the equation error method penalizing the violation of constraints [Acar 1993]. The Sobolev norms have been used to measure the residual [Hinze and Quyen 2016; Ito and Kunisch 1990].

3 KEY IDEAS

Our methods rely on a particular characterization of the space of inversion-free maps. Generalizing the notion of harmonic maps and following Wang and Solomon [2021], we optimize in the space of *quasi-harmonic maps*, the family of maps wherein each interior point is positioned at a weighted average of its neighbors' positions. The weighting factors together with the boundary determine positions of the interior points.

The family of quasi-harmonic maps provides a rich space of maps including *all* inversion-free maps. In particular, as we will derive in §5.2, a map is diffeomorphic/injective if and only if it is quasi-harmonic and satisfies a particular *Cauchy boundary condition*, which is a coupled pair of simultaneous Dirichlet and Neumann boundary conditions (BCs). Schematically, we show

Injectivity = {Quasi-harmonic} + {Dirichlet & Neumann BCs}.

Based on this theoretical result, we formulate mapping as an optimal control (or inverse PDE) problem, searching in the space of elliptic operators for one whose associated map satisfies both Dirichlet and Neumann conditions. Our variable is a spatially-varying anisotropy field, and the Neumann boundary condition involves the anisotropic co-normal derivative. Our algorithm maintains quasi-harmonicity with Dirichlet boundary conditions exactly, while minimizing violation of the Neumann boundary conditions. This strategy "pushes" inversions onto the boundary: Detecting and resolving inversions at the boundary then yields inversion-free maps.

We enforce boundary conditions via a few functionals, such as the anisotropic Dirichlet energy and the Poisson functional. Their minimizers are provably inversion-free assuming an inversion-free map exists, both in the continuous and discrete cases (see §5,§6). Through a careful parameterization, we optimize these functionals via an *unconstrained* problem on which simple gradient descent methods like Adam or L-BFGS are very efficient.

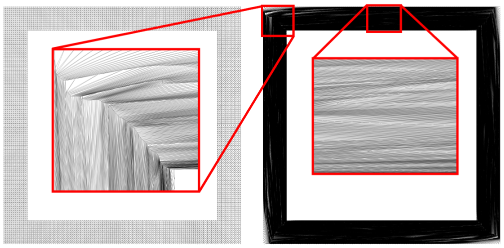


Fig. 4. A stress test that our method passes. We rotate the inner rectangle by π, setting the inner and outer rectangles as positional constraints. A bijective mapping will inevitably undergo an extreme amount of distortion. The zoom-in views show that the triangles become very thin: Successful maps must avoid "geometric locking" that possibly prevent any solution, which is not a concern in the smooth setting. This example suggests that the injective mapping problem can have more of a combinatoric nature rather than mimicking behaviors of smooth PDEs.

A key challenge is to propose injectivity-promoting functionals that, after discretization, still yield exact injectivity. For example, Figures 3 and 4 illustrate that under coarse meshing or extreme distortions, injective mapping becomes *combinatoric*; in this regime, methods that hold asymptotically in the smooth limit may not work in practice. In §6, we show that our model also works in the discrete setting.

4 MATHEMATICAL PRELIMINARIES

4.1 Basic Definitions

In this paper, we consider mapping from a source domain Ω, a two dimensional manifold with boundary $\partial\Omega$, to a target $\Gamma \subset \mathbb{R}^2$, i.e. $\phi : \Omega \to \Gamma : \mathbf{x} = (x, y) \to \phi(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))$. The Jacobian (or, deformation gradient) of ϕ is

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} = \begin{bmatrix} \nabla u(\mathbf{x}) & \nabla v(\mathbf{x}) \end{bmatrix}^\top .$$

The map is called "injective" if distinct points on the source domain are mapped to distinct points on the target. The map is called inversion-free, flip-free, or orientation-preserving if $\det \mathbf{J}(\mathbf{x}) > 0$ everywhere. These concepts are closely related and often become equivalent.

We consider continuous maps, which prevent fracture or tearing. Discretely, like many existing works, we choose to keep the mesh triangulation unchanged.

In the smooth setting, we work with *diffeomorphisms*—or differentiable maps with differentiable inverses—as a result of our PDE-based characterization. In the discrete case, despite often being referred to as *bijective* mapping in graphics, in reality we are computing *homeomorphisms*, which relax differentiability to continuity, thanks to the use of triangulations.

Restricting to diffeomorphisms does not practically limit the capacity of our model. Indeed, our discrete model can represent *any* inversion-free map of a triangle mesh (Theorem 6.3). Moreover, just like a continuous function can be approximated by smooth functions, Iwaniec et al. [2012] prove that any planar homeomorphism can be arbitrarily closely approximated by diffeomorphisms.

4.2 Elliptic PDEs and Quasi-harmonic Functions

We briefly review notions from elliptic PDE. In our discussion, the domain Ω can be a curved surface or even a Riemannian manifold equipped with a non-Euclidean metric, in which case the gradient ∇ will be understood as the intrinsic gradient and tensor field $\mathbf{A}(\cdot)$ will operate on the tangent space. See [Evans 1998; Gilbarg and Trudinger 2015; Taylor 2010] for general discussion of elliptic PDE and [Wang and Solomon 2019] for an accessible introduction including discussion of discretization.

We begin by defining our basic object of study:

Definition 4.1 (Quasi-harmonic map). We say a map (u, v) is quasi-harmonic, or, $\mathbf{A}(\mathbf{x})$-harmonic if $u(\mathbf{x}), v(\mathbf{x})$ satisfy the PDEs

$$\nabla \cdot [\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})] = 0 \quad \text{and} \quad \nabla \cdot [\mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})] = 0 \qquad (1)$$

for some positive definite tensor field $\mathbf{A}(\cdot) : \Omega \to \mathbb{R}^{2\times 2}$ and $K > 0$:

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} a_{11}(\mathbf{x}) & a_{12}(\mathbf{x}) \\ a_{12}(\mathbf{x}) & a_{22}(\mathbf{x}) \end{bmatrix} \quad \text{and} \quad \frac{1}{K}\mathbf{I} \le \mathbf{A}(\mathbf{x}) \le K\mathbf{I}. \qquad (2)$$

In this definition, the PDE is both inhomogeneous ($\mathbf{A}(\mathbf{x})$ can differ from point to point) and anisotropic (directionally-dependent).

On a Riemannian manifold with metric g, the Laplace–Beltrami (Laplacian) operator is defined as:

$$\Delta_{\mathcal{M}} = \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} = \frac{1}{\sqrt{|\det g|}} \sum_{ij} \partial_i \left(\sqrt{|\det g|} g^{ij} \partial_j \right), \qquad (3)$$

where $g_{ij} \in \mathbb{R}^{2\times 2}$ is the metric tensor and $(g^{ij}) = (g^{-1})_{ij}$ is the inverse metric. In this formula, $\sqrt{|\det g|} g^{-1}$ has determinant 1, hinting to restrict $\det \mathbf{A}(\mathbf{x}) = 1$ in our model later.

Boundary conditions must be specified for elliptic PDEs to determine a solution, such as the Dirichlet conditions:

Definition 4.2 (Dirichlet condition). For $b : \partial\Omega \to \mathbb{R}$, we say $w(\cdot)$ is quasi-harmonic with the Dirichlet boundary condition if

$$\begin{aligned} \nabla \cdot [\mathbf{A}(\mathbf{x})\nabla w(\mathbf{x})] &= 0, && \forall \mathbf{x} \in \Omega \backslash \partial\Omega \\ w(\mathbf{x}) &= b(\mathbf{x}), && \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{4}$$

We say $w(\cdot)$ is the $\mathbf{A}(\mathbf{x})$-harmonic (quasi-harmonic) extension of b into the interior. The PDE (strong form) in (4) is equivalent to the variational problem (weak form) in (5)

$$\begin{aligned} \min_{w(\cdot)} \quad & \int_\Omega \nabla w(\mathbf{x})^\top \mathbf{A}(\mathbf{x}) \nabla w(\mathbf{x}) \\ \text{s.t.} \quad & w(\mathbf{x}) = b(\mathbf{x}) \quad \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{5}$$

The theory of linear elliptic PDEs implies that the tensor function $\mathbf{A}(\cdot)$ and Dirichlet boundary data $b(\cdot)$ uniquely determine the solution $w(\cdot)$. Thus, we can define the *coefficient-to-solution operator*:

$$\mathcal{F} : \mathbf{A}(\cdot); b(\cdot) \mapsto w(\cdot).$$

Another common boundary condition is the Neumann one:

Definition 4.3 (Neumann condition). For $g : \partial\Omega \to \mathbb{R} : \int_{\partial\Omega} g = 0$, We say $w(\cdot)$ is quasi-harmonic with the Neumann condition if

$$\begin{aligned} \nabla \cdot [\mathbf{A}(\mathbf{x})\nabla w(\mathbf{x})] &= 0, && \forall \mathbf{x} \in \Omega \backslash \partial\Omega \\ \mathbf{n}(\mathbf{x})^\top \mathbf{A}(\mathbf{x}) \nabla w(\mathbf{x}) &= g(\mathbf{x}), && \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{6}$$

The Neumann boundary condition/operator is in the form of the co-normal derivative $\mathbf{n}(\mathbf{x})^\top \mathbf{A}(\mathbf{x})\nabla$, which corresponds to the *natural boundary condition* resulted from minimizing the anisotropic Dirichlet energy in (5) without boundary conditions.

If the $w(\cdot)$ in (5) and (6) are the same, the map from $b(\cdot)$ to $g(\cdot)$ is the *Dirichlet-to-Neumann operator* (DtN) or *Poincaré–Steklov operator*; see [Gao et al. 2014; Sawhney and Crane 2017; Wang et al. 2018] for applications. For a prescribed tensor field $\mathbf{A}(\cdot)$, one has to choose *either* the Dirichlet or Neumann boundary condition, but not both simultaneously, since the Dirichlet data determine the Neumann data uniquely via the DtN operator. The reverse is also true: the Neumann data uniquely determine the Dirichlet data up to a constant via the *Neumann-to-Dirichlet operator* (NtD).

5 QUASI-HARMONIC MAPPING

We first introduce our forward model based on the notion of quasi-harmonic maps, which, intuitively, describe the equilibrium state of an elastic membrane with anisotropic elasticity specified by a tensor field $\mathbf{A}(\cdot)$. As a result, each interior point $\mathbf{x}$ is positioned at the $\mathbf{A}(\mathbf{x})$-weighted average of its neighbors' positions. This model is versatile: Any inversion-free map is quasi-harmonic, i.e., the equilibrium state for *some* tensor field $\mathbf{A}(\cdot)$. In addition, existence of $\mathbf{A}(\mathbf{x})$ conveniently provides a global condition to certify injectivity.

5.1 Forward Model with Fixed Boundary Maps

Let us consider a map whose entire boundary is subject to some positional constraints. Precisely speaking, we propose to represent a map (u, v) as the solution to an elliptic PDE with the Dirichlet boundary condition, parameterized by a tensor field $\mathbf{A}(\cdot)$:

Definition 5.1 (Quasi-harmonic map with fixed boundary). Given a positive field $\mathbf{A}(\mathbf{x}) : \Omega \to \mathbb{R}^{2\times2}$ and boundary coordinate functions $b_u(\mathbf{x}), b_v(\mathbf{x}) : \partial\Omega \to \mathbb{R}^2$, we consider the map (u, v) satisfying:

$$\begin{aligned} \nabla \cdot [\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})] = 0, \quad \nabla \cdot [\mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})] = 0, \quad & \forall \mathbf{x} \in \Omega \backslash \partial\Omega \\ u(\mathbf{x}) = b_u(\mathbf{x}), \quad v(\mathbf{x}) = b_v(\mathbf{x}), \quad & \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{7}$$

Quasi-harmonic maps provide a rich parametric family of maps: As we will see, the quasi-harmonic family of maps includes *all* inversion-free maps. The tensor field $\mathbf{A}(\cdot)$ together with the boundary map $b_u(\cdot), b_v(\cdot)$ can be thought of as a surrogate representation of the map (u, v).

The celebrated Radó-Kneser-Choquet (RKC) Theorem [Choquet 1945; Kneser 1926; Rado 1926] proves that harmonic maps onto convex regions are diffeomorphic; see [Duren 2004] for discussion. That is, for $\mathbf{A}(\mathbf{x}) \equiv \mathbf{I}_{2\times2}$, the map (u, v) in (7) is diffeomorphic. In fact, the map (u, v) is diffeomorphic even if $\mathbf{A}(\mathbf{x}) \neq \mathbf{I}_{2\times2}$, as long as the target domain (b_u, b_v) is convex [Alessandrini and Nesi 2001].

In practice, convexity of the target domain is undesirable, as it often yields large distortions in the computed map. In addition, wherever the user imposes positional constraints in the interior also becomes part of the boundary, easily breaking the convex boundary requirement. Hence, we derive a condition characterizing when maps onto potentially *nonconvex* regions are inversion-free.

5.2 Main Theory on Diffeomorphic Maps

Now we state the main theory, a necessary and sufficient condition determining when a map is a diffeomorphism onto a nonconvex domain. We only present key features and conclusions of the theory; details are provided in the Appendix §D. While we rely on recent progress in the PDE theory of maps [Alessandrini and Nesi 2021], our contribution is to identify the "if-and-only-if" nature of the condition and to convert it to a form involving natural boundary conditions for the elliptic PDE; this form suggests the optimal control problem proposed in §5.3.

With extra conditions on the boundary, map injectivity onto a nonconvex domain can be established:

Theorem 5.2 (Main result: Continuous version). *Suppose Ω is a two-dimensional Riemannian manifold with disk topology, and consider a planar domain $\Gamma \subset \mathbb{R}^2$ whose boundary $\partial\Gamma$ is a simple closed curve. Assume $\phi = (u, v) : \Omega \to \Gamma$ diffeomorphically maps $\partial\Omega$ onto $\partial\Gamma$. Denote the (given) boundary map as $[b_u, b_v](\cdot) : \partial\Omega \to \mathbb{R}^2$, and denote the outward normal to $\partial\Gamma$ as $\hat{\mathbf{n}}(\cdot) : \partial\Omega \to S^1$.*

Then, $\phi = (u, v)$ is a diffeomorphism if and only if there exist (1) a positive definite tensor field $\mathbf{A}(\cdot)$ satisfying $\frac{1}{K}\mathbf{I} \leq \mathbf{A}(\mathbf{x}) \leq K\mathbf{I}$ for some $K > 0$ and (2) a positive function $s : \partial\Omega \to \mathbb{R}$ that $s(\mathbf{x}) \geq S$ for some $S > 0$, such that ϕ is $\mathbf{A}(\mathbf{x})$-harmonic with a special Cauchy boundary condition, i.e.:

$$\begin{aligned} &\nabla \cdot [\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})] = 0 && \forall \mathbf{x} \in \Omega \backslash \partial\Omega \\ &\nabla \cdot [\mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})] = 0 && \forall \mathbf{x} \in \Omega \backslash \partial\Omega \\ &u(\mathbf{x}) = b_u(\mathbf{x}) && \forall \mathbf{x} \in \partial\Omega \\ &v(\mathbf{x}) = b_v(\mathbf{x}) && \forall \mathbf{x} \in \partial\Omega \\ &\mathbf{n}(\mathbf{x})^\top \begin{bmatrix} \mathbf{A}(\mathbf{x})\nabla u(\mathbf{x}) & \mathbf{A}(\mathbf{x})\nabla v(\mathbf{x}) \end{bmatrix} = s(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x})^\top && \forall \mathbf{x} \in \partial\Omega. \end{aligned} \tag{8}$$

Optionally, we can additionally assume

$$\begin{cases} \det \mathbf{A}(\mathbf{x}) = 1 & \forall \mathbf{x} \in \Omega \\ s(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x})^\top = \begin{bmatrix} \frac{\partial}{\partial t}\mathbf{b}_v(\mathbf{x}) & -\frac{\partial}{\partial t}\mathbf{b}_u(\mathbf{x}) \end{bmatrix} & \forall \mathbf{x} \in \partial\Omega. \end{cases} \tag{9}$$

Appendix §D provides proof and discussion of Theorem 5.2.

Intuitively, the theorem arises from two premises:

(1) the tensor field $\mathbf{A}(\cdot)$ allows us to "reproduce" the Laplace-Beltrami operator of any target domain induced by a diffeomorphism;
(2) in the reverse direction, our boundary conditions ensure the map is orientation-preserving at the boundary.

The extra degrees of freedom in choosing a tensor field $\mathbf{A}(\cdot)$ make it possible to specify *Cauchy boundary conditions*, namely enforcing *both* Dirichlet and Neumann boundary conditions. This is not possible for fixed $\mathbf{A}(\cdot)$. Cauchy boundary conditions should not be confused with Zaremba (or Robin) boundary conditions, which specify either Dirichlet or Neumann conditions (or their linear combination) at each point, but not both.

The extra condition (9) does not limit the representation capacity of our model. In this case, the formula (3) suggests that $\mathbf{A}(\cdot)$ can serve the role of the term $\sqrt{|\det g|}g^{-1}$: Then $\mathbf{A}(\mathbf{x})$ encodes $\sqrt{|\det g|}g^{-1}$, the pull-back metric induced by the map up to scaling, or, the *conformal structure*. But note this is only the case for our method *upon convergence*.

5.3 Mapping via Optimal Control

System (8) provides a PDE-based characterization of the space of inversion-free maps. It suggests using an inverse problem formulation to optimize for a high-quality diffeomorphism. As the starting point, we copy-paste (8) as the constraint, arriving at an optimal control problem:

$$\begin{array}{lll} \min_{\mathbf{A},u,v} & R(u,v,\mathbf{A}) & \\ \text{s.t.} & \nabla\cdot[\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})]=0 & \forall\mathbf{x}\in\Omega\backslash\partial\Omega \\ & \nabla\cdot[\mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})]=0 & \forall\mathbf{x}\in\Omega\backslash\partial\Omega \\ & u(\mathbf{x})=b_u(\mathbf{x}) & \forall\mathbf{x}\in\partial\Omega \\ & v(\mathbf{x})=b_v(\mathbf{x}) & \forall\mathbf{x}\in\partial\Omega \\ & \mathbf{n}(\mathbf{x})^\top\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})=g_u(\mathbf{x}) & \forall\mathbf{x}\in\partial\Omega \\ & \mathbf{n}(\mathbf{x})^\top\mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})=g_v(\mathbf{x}) & \forall\mathbf{x}\in\partial\Omega \\ & \mathbf{A}(\mathbf{x})\succeq 0 & \forall\mathbf{x}\in\Omega \\ & \det\mathbf{A}(\mathbf{x})=1 & \forall\mathbf{x}\in\Omega. \end{array} \tag{10}$$

Here, the objective $R(\cdot)$ is some energy or regularizer used to select among different non-inverting maps. As we are primarily interested in the feasibility problem, $R(\cdot)$ can even be 0, which is the default choice throughout the paper unless specified otherwise. More broadly, $R(\cdot)$ can be any of the following

- *Zero* as the default choice, which solves the feasibility problem and will accept any inversion-free map (§7).
- A *deformation energy* $R = E(u,v)$ based on physics or geometry, e.g., ARAP (§11.1).
- A *regularization term*, e.g., what measures $\textsc{cond}(\mathbf{A})$ and (upon convergence) corresponds to a conformal energy like MIPS (§11.1.4).

6 DISCRETIZATION WITH EXACT INJECTIVITY

To convert our theory to a practical algorithm, we introduce a discretization such that an analog of Theorem 5.2 holds, namely Theorem 6.4 below.

6.1 Motivation and Challenge

$\mathbf{A}(\mathbf{x})$-harmonic functions enjoy the *mean value property*: the value at each point $\mathbf{x}$ equals to the $\mathbf{A}(\mathbf{x})$-weighted average of its neighbors' values. This notion naturally transfers to a graph/mesh with edge weights w, by viewing the triangular mesh as a graph: We consider functions such that the value at each vertex/node i equals the w-weighted average of its adjacent vertices' values.

Analogous results hold for this discrete notion of harmonicity: Akin to the RKC Theorem, Tutte's Embedding Theorem asserts that a discrete mapping computed using this procedure is injective, when the target is a convex polygon and the edge weights are positive [Floater 2003a; Gortler et al. 2006; Tutte 1963].

In Tutte's embedding, ensuring symmetric positive edge weights greatly limits the representation capacity of the model [Wardetzky et al. 2007, Figure 1]. A discrete maximum principle is even harder to ensure for triangular meshes with arbitrary anisotropy, so negative weights are inevitable. Instead of relying on Tutte's embedding theorem, we put constraints on edge weights—possibly negative—to ensure injectivity, including the case of a nonconvex boundary where Tutte's embedding fails even with positive weights.

Remark. *In general, Tutte's embedding does allow for asymmetric edge weights, but we leave development of a variant of our approach with asymmetric but positive weights for future work. For example, in discrete computational geometry, the* geodesic triangulation *[Luo 2022] uses directed graph Laplacian with asymmetric weights, such as the mean-value Laplacian [Floater 2003b]. Gortler et al. [2006] consider when Tutte's embeddings with asymmetric weights onto nonconvex domains are injective, and Kovalsky et al. [2020] derive a conic condition for that.*

6.2 Discretization using Piecewise Linear FEM

Our approach to discretize the anisotropic Laplacian $\nabla\cdot[\mathbf{A}(\mathbf{x})\nabla]$ is a straightforward application of the piecewise linear finite element method (FEM), an intrinsic generalization of the extrinsic case [Wang and Solomon 2021]; experienced readers may skip this section and refer to Table 1 for notation.

Throughout the paper, we consider a triangular mesh $\mathcal{T}$ with f faces and n vertices, as a discrete approximation of the domain Ω. Applying piecewise linear FEM, the tensor field becomes a piecewise constant 2×2 matrix per triangle, and the anisotropic Laplacian $\nabla\cdot[\mathbf{A}(\mathbf{x})\nabla]$ is discretized by the matrix multiplication $\mathbf{G}^\top\mathbf{A}\mathbf{G}$. Here $\mathbf{G}\in\mathbb{R}^{2f\times n}$ is the discrete (intrinsic) gradient operator, and the anisotropy tensor field becomes the sparse matrix $\mathbf{A}\in\mathbb{R}^{2f\times 2f}$

$$\mathbf{G}=\begin{bmatrix}\mathbf{G}_x\\ \mathbf{G}_y\end{bmatrix}\in\mathbb{R}^{2f\times n},\quad \mathbf{A}=\begin{bmatrix}\textsc{diag}(\mathbf{a}_{11}) & \textsc{diag}(\mathbf{a}_{12})\\ \textsc{diag}(\mathbf{a}_{12}) & \textsc{diag}(\mathbf{a}_{22})\end{bmatrix},$$

where $\mathbf{a}_{11},\mathbf{a}_{12},\mathbf{a}_{22}\in\mathbb{R}^{f\times 1}$, and $\textsc{diag}(\cdot)$ expands a vector into a diagonal matrix.

To ease derivations, we stack the nonzero entries of $\mathbf{A}$ into the vector $\mathbf{a}$ such that:

$$\mathbf{a}:=\textsc{flatten}(\mathbf{A})=\begin{bmatrix}\mathbf{a}_{11}\\ \mathbf{a}_{12}\\ \mathbf{a}_{22}\end{bmatrix}\in\mathbb{R}^{3f\times 1}.$$

Throughout the paper, $\mathbf{A}$ and $\mathbf{a}$ will be used interchangeably.

Following Wang and Solomon [2021], we define an operator $\textsc{sp}(\cdot)$ that satisfies the expression $\mathbf{Ab} = \textsc{sp}(\mathbf{b})\textsc{flatten}(\mathbf{A})$:

Definition 6.1 (Span). For $\mathbf{b} = \begin{bmatrix}\mathbf{b}_0\\ \mathbf{b}_1\end{bmatrix} \in \mathbb{R}^{2f\times 1}$ with $\mathbf{b}_0, \mathbf{b}_1 \in \mathbb{R}^{f\times 1}$, define the sparse matrix $\textsc{sp}(\mathbf{b})$ as

$$\textsc{sp}(\mathbf{b}) := \begin{bmatrix}\textsc{diag}(\mathbf{b}_0) & \textsc{diag}(\mathbf{b}_1) & \\ & \textsc{diag}(\mathbf{b}_0) & \textsc{diag}(\mathbf{b}_1)\end{bmatrix} \in \mathbb{R}^{2f\times 3f}.$$

Denote $\mathbf{m} \in \mathbb{R}^{f\times 1}$ such that $\mathbf{m}_j$ stores the area of triangle j. We use $\mathbf{A}_{(j)}$ and $\mathbf{G}_{(j)}$ to denote the anisotropy tensor and intrinsic gradient at the j-th triangle:

$$\mathbf{A}_{(j)} = \begin{bmatrix}(\mathbf{a}_{00})_j & (\mathbf{a}_{01})_j\\ (\mathbf{a}_{01})_j & (\mathbf{a}_{11})_j\end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \mathbf{G}_{(j)} := \begin{bmatrix}(\mathbf{G}_x)_{j:}\\ (\mathbf{G}_y)_{j:}\end{bmatrix} \in \mathbb{R}^{2\times n}.$$

Note $\textsc{diag}(\mathbf{m})$, $\mathbf{G}$, and $\mathbf{A}$ are all intrinsic discrete operators: $\mathbf{G}_{(j)}$ yields the intrinsic gradient measured in a local coordinate system for triangle j. The per-triangle local coordinate can be arbitrarily chosen but $\mathbf{A}_{(j)}$ will need to operate consistently in the same coordinate system (a discrete tangent space).

$\mathbf{G}, \mathbf{m}$ reproduce the familiar cotangent Laplacian $\mathbf{L} \in \mathbb{R}^{n\times n}$:

$$\mathbf{L} = \mathbf{G}^\top \begin{bmatrix}\textsc{diag}(\mathbf{m}) & \\ & \textsc{diag}(\mathbf{m})\end{bmatrix} \mathbf{G}.$$

Denote $\mathbf{w} \in \mathbb{R}^n$ as the discrete solution to a Laplacian system (11). Let k be the number of vertices on the boundary of the mesh. Denote by $\mathbf{R} \in \mathbb{R}^{n\times k}$, $\mathbf{S} \in \mathbb{R}^{n\times(n-k)}$ the binary selection sparse matrices such that $\mathbf{R}^\top\mathbf{w}$, $\mathbf{S}^\top\mathbf{w}$ yield rows corresponding to vertices on the boundary and in the interior, respectively. For the discrete Laplacian $\mathbf{G}^\top\mathbf{AG}$, the output $\mathbf{R}^\top\mathbf{G}^\top\mathbf{AGw} \in \mathbb{R}^k$ measures the discrete Neumann boundary data (in weak form, i.e., without dividing by the average edge length associated with each boundary vertex).

The variational problem (5) can be discretized as

$$\begin{array}{ll}\min_{\mathbf{w}} & \frac{1}{2}\textsc{tr}(\mathbf{w}^\top\mathbf{G}^\top\mathbf{AGw})\\ \text{s.t.} & \mathbf{R}^\top\mathbf{w} = \mathbf{b},\end{array} \tag{11}$$

where $\mathbf{b} \in \mathbb{R}^k$ discretizes the per-vertex Dirichlet boundary data.

By standard linear algebra, the unknown rows $\mathbf{z} := \mathbf{S}^\top\mathbf{w}$ are uniquely determined by $\mathbf{A}, \mathbf{b}$ via

$$\mathbf{z} = -[\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGRb}. \tag{12}$$

Hence, the whole solution becomes $\mathbf{w} \equiv \mathbf{Sz} + \mathbf{Rb}$. Our derivation suggests the following definition:

Definition 6.2 (Coefficient-to-solution, Dirichlet-to-Neumann, Neumann-to-Dirichlet). The discrete *coefficient-to-solution* operator $\mathcal{F} : \mathbb{R}^{2f\times 2f}; \mathbb{R}^{n\times 1} \mapsto \mathbb{R}^{n\times 1}$ is the map $\mathbf{A}; \mathbf{b} \mapsto \mathbf{w}$ via Eq. 12, namely

$$\mathcal{F} : \mathbf{A}; \mathbf{b} \mapsto \mathbf{w} = \left(\mathbf{R} - \mathbf{S}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGR}\right)\mathbf{b}. \tag{13}$$

Closely related, we have the discrete *Dirichlet-to-Neumann* operator

$$\mathcal{S}(\mathbf{A}) := \mathbf{R}^\top\mathbf{G}^\top\mathbf{AGR} - \mathbf{R}^\top\mathbf{G}^\top\mathbf{AGS}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGR}. \tag{14}$$

from the Schur complement of $\begin{bmatrix}\mathbf{R}^\top\mathbf{G}^\top\mathbf{AGR} & \mathbf{R}^\top\mathbf{G}^\top\mathbf{AGS}\\ \mathbf{S}^\top\mathbf{G}^\top\mathbf{AGR} & \mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}\end{bmatrix}$. The discrete *Neumann-to-Dirichlet* operator is $\mathbf{R}^\top(\mathbf{G}^\top\mathbf{AG})^\dagger\mathbf{R} \equiv \mathcal{S}(\mathbf{A})^\dagger$.

Table 1. A summary of notations that we will introduce and use throughout the paper. An analogy between the discrete and continuous cases is given.

Name	Continuous	Discrete
gradient	∇	$\mathbf{G}$
tensor field	$\mathbf{A}(\cdot), \mathbf{A}(\mathbf{x})$	$\mathbf{A}$ or $\mathbf{a}$, $\mathbf{A}_{(j)}$
scalar function	$w(\cdot), w(\mathbf{x})$	$\mathbf{w}, \mathbf{w}_i$
restrict to interior	$w\vert_{\Omega\backslash\partial\Omega}$	$\mathbf{S}^\top\mathbf{w}$
restrict to boundary	$w\vert_{\partial\Omega}$	$\mathbf{R}^\top\mathbf{w}$
Dirichlet energy	$\frac{1}{2}\int_\Omega \nabla w(\mathbf{x})^\top\mathbf{A}(\mathbf{x})\nabla w(\mathbf{x})$	$\frac{1}{2}\mathbf{w}^\top\mathbf{G}^\top\mathbf{AGw}$
Laplace equation	$\nabla\cdot[\mathbf{A}(\mathbf{x})\nabla w(\mathbf{x})]\vert_{\Omega\backslash\partial\Omega} = 0$	$\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGw} = \mathbf{0}$
Dirichlet condition	$w\vert_{\partial\Omega} = \mathrm{b}$	$\mathbf{R}^\top\mathbf{w} = \mathbf{b}$
Neumann condition	$\mathbf{n}(\mathbf{x})^\top\mathbf{A}(\mathbf{x})\nabla w(\mathbf{x}) = g(\mathbf{x})$	$\mathbf{R}^\top\mathbf{G}^\top\mathbf{AGw} = \mathbf{g}$

The last equality can be established by the Woodbury matrix identity applied to the Schur block. Later we will see that our key functionals for injectivity are weak forms of $\mathcal{S}(\mathbf{A})$ and its inverse.

It is convenient to derive boundary conditions by introducing the circular gradient operator:

Definition 6.3 (Circular boundary gradient, or discrete area form). Define $\mathbf{D} \in \mathbb{R}^{n\times n}$ as the sparse matrix computing the circular gradient of a function along the boundary:

$$\mathbf{D}_{ij} = \begin{cases}-\frac{1}{2} & \text{if } i \to j \text{ is a boundary edge}\\ \frac{1}{2} & \text{if } j \to i \text{ is a boundary edge}\\ 0 & \text{otherwise}\end{cases} \tag{15}$$

$\mathbf{D}$ can be viewed as a discrete 1-dimensional gradient via central differencing. It makes sense to define its restriction to the boundary vertices for simplicity:

$$\mathbf{C} := \mathbf{R}^\top\mathbf{DR}. \tag{16}$$

We have $\mathbf{D}^\top = -\mathbf{D}$. Due to Green's Theorem, $\mathbf{u}^\top\mathbf{Dv}$ is also the *signed area* of the domain [Mullen et al. 2008]; see details in Appendix §A.1.

6.3 Main Theory on Discrete Homeomorphic Maps

In the discrete case, we work with a variant requiring $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1$, corresponding to the weak version of Theorem 5.2 that additionally requires $\det\mathbf{A}(\mathbf{x}) = 1$. This version of quasi-harmonic maps also does not limit the modeling capacity discretely. This choice leads to a variant with an elementary proof. It also removes the requirement that the domain should have disk topology and does not rely on a discrete maximum principle. Specifically, we have

Theorem 6.4 (Main result: Discrete version). *Consider a discrete map $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{n\times 2}$. Assume the boundary map $\mathbf{b}_u, \mathbf{b}_v$ is bijective onto a polygon, with the (weak form) weighted normal $\hat{\mathbf{n}} = [\mathbf{g}_u \quad \mathbf{g}_v]$, where $\mathbf{g}_u, \mathbf{g}_v \in \mathbb{R}^{k\times 1}$ are constant vectors determined by the target boundary $\mathbf{b}_u, \mathbf{b}_v$ via $\mathbf{g}_u = \mathbf{C}\mathbf{b}_v$, $\mathbf{g}_v = -\mathbf{C}\mathbf{b}_u$.*

Then, the discrete map $(\mathbf{u}, \mathbf{v})$ is locally injective, i.e., $\det\mathbf{J}_{(j)} > 0, \forall j$, if and only if there exists a block-diagonal positive definite matrix $\mathbf{A}$ such that $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1\ \forall j$ and the following bilinear system *holds:*

$$\begin{cases} \mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}[\mathbf{u} \quad \mathbf{v}] = [\mathbf{0} \quad \mathbf{0}] \\ \mathbf{R}^\top[\mathbf{u} \quad \mathbf{v}] = [\mathbf{b}_u \quad \mathbf{b}_v] \\ \mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}[\mathbf{u} \quad \mathbf{v}] = [\mathbf{g}_u \quad \mathbf{g}_v]. \end{cases} \tag{17}$$

In addition, the condition $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1, \forall j$ *can be replaced by the convex relaxation* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) \geq 1, \forall j$.

See Appendix §B,C for a proof and detailed discussion. When the boundary polygon or mesh connectivity does not allow an injective map to exist, Theorem 6.4 still holds in the sense that there will not exist such an $\mathbf{A}$ satisfying all conditions.

System (17) can also be compactly written as a constraint on the associated Dirichlet-to-Neumann operator:

$$\mathcal{S}(\mathbf{A})[\mathbf{b}_u \quad \mathbf{b}_v] = [\mathbf{g}_u \quad \mathbf{g}_v]$$

Remark. *Future work could derive a version of the theorem allowing the determinant to be non-constant as in Theorem 5.2, possibly using nonlinear FEM schemes compatible with maximum principle [Droniou and Potier 2011; Liska and Shashkov 2008; Lu et al. 2014]. It would also be interesting to see if similar results hold using a DEC discretization [de Goes et al. 2014].*

7 INVERSE CAUCHY SOLVER: SIMULTANEOUS DIRICHLET & NEUMANN CONDITIONS

So far, we have equated the injective mapping to the PDE-constrained optimization problem (10). Next, we develop a solver for optimally controlling elliptic PDEs so that our Cauchy boundary condition holds. While they are designed for the mapping problem, in the future our techniques in this section may be applied to other inverse problems involving pairs of Neumann and Dirichlet data.

Problem (10) can be "ill-posed" if, for example, we take $R \equiv 0$, since injective maps are not unique. This aspect can make the problem even more difficult, calling for a careful numerical scheme.

7.1 Insights from Some Failed Attempts

The optimal control problem (10) is challenging: Before we arrive at our proposed algorithm, we tried several less successful methods. Some failed attempts providing key insights are listed here:

1. *Alternating descent.* Alternatively updating $\mathbf{A}$ and map (u, v) works only for trivial examples and easily gets stuck at local minima.
2. *Equation error method.* Directly measuring equation errors without appropriate norm gets stuck easily; failure patterns include ignoring the Neumann boundary conditions.
3. *Hard constraints for both Dirichlet & Neumann BCs.* While the boundary's one-ring-neighbor triangles become orientation-preserving, the flips appear at the two-ring neighbors as a local optimum.

These observations indicate that the boundary condition in (8) should be be viewed as *globally constraining* the tensor $\mathbf{A}_{(j)}$, including those in the interior of the domain, rather than only $\mathbf{A}_{(j)}$ adjacent to the boundary. As we will see, our method propagates and attributes errors in the Neumann data to tensors in the interior.

7.2 Our Solution: Strict Dirichlet & Soft Neumann

As an alternative to the approaches explored in §7.1, we propose to enforce the Dirichlet boundary conditions via hard constraints while minimizing the violation of the Neumann boundary conditions. Prioritizing Dirichlet also makes sense practically, since it is the visible positional constraint prescribed by users.

To do so, we eliminate $\mathbf{u}, \mathbf{v}$ via the coefficient-to-solution operator $\mathcal{F}$ in (13), yielding the following reformulation of (10):

$$\begin{aligned} \min\nolimits_{\mathbf{A}} \quad & Q\left(\mathcal{F}(\mathbf{A}; \mathbf{b}_u), \mathcal{F}(\mathbf{A}; \mathbf{b}_v), \mathbf{A}\right) \\ \text{s.t.} \quad & \text{some constraints on } \mathbf{A}_{(j)}, \forall j. \end{aligned} \tag{18}$$

where

$$Q(\mathbf{u}, \mathbf{v}, \mathbf{A}) := R(\mathbf{u}, \mathbf{v}, \mathbf{A}) + V\left(\mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}[\mathbf{u}, \mathbf{v}], \mathbf{A}; \mathbf{g}_u, \mathbf{g}_v\right) \tag{19}$$

$$V\left(\mathbf{n}_u, \mathbf{n}_v, \mathbf{A}; \mathbf{g}_u, \mathbf{g}_v\right) \text{ measures } \left([\mathbf{n}_u, \mathbf{n}_v] - [\mathbf{g}_u, \mathbf{g}_v]\right). \tag{20}$$

V takes the actual Neumann value $\mathbf{n}_u, \mathbf{n}_v$ and measures its violation from the the prescribed one $[\mathbf{g}_u, \mathbf{g}_v]$. We will discuss many choices of V. We ignore for now the constraint on $\mathbf{A}_{(j)}$, which can be enforced through a careful reparameterization $\mathbf{A} = p(\xi)$ in §8.

With R, V chosen, completely eliminating intermediate variables $\mathbf{u}, \mathbf{v}$ so the Neumann data $\mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}[\mathbf{u}, \mathbf{v}]$ becomes $[\mathcal{S}(\mathbf{A})\mathbf{b}_u, \mathcal{S}(\mathbf{A})\mathbf{b}_v]$, the objective Q can be written down:

$$\begin{aligned} Q\left(\mathcal{F}(\mathbf{A}; \mathbf{b}_u), \mathcal{F}(\mathbf{A}; \mathbf{b}_v), \mathbf{A}\right) = R\left(\mathcal{F}(\mathbf{A}; \mathbf{b}_u), \mathcal{F}(\mathbf{A}; \mathbf{b}_v), \mathbf{A}\right) \\ + V\left(\mathcal{S}(\mathbf{A})\mathbf{b}_u, \mathcal{S}(\mathbf{A})\mathbf{b}_v, \mathbf{A}; \mathbf{g}_u, \mathbf{g}_v\right). \end{aligned} \tag{21}$$

Differentiating into $\mathcal{F}$ or $\mathcal{S}$, the gradient can be obtained via

$$\frac{\mathrm{d}Q}{\mathrm{d}\mathbf{a}} = \left[\frac{\partial Q}{\partial \mathbf{a}} + \frac{\partial Q}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{a}} + \frac{\partial Q}{\partial \mathbf{v}}\frac{\partial \mathbf{v}}{\partial \mathbf{a}}\right]^\top \in \mathbb{R}^{3f\times 1}, \tag{22}$$

where $\frac{\partial \mathbf{u}}{\partial \mathbf{a}}$ is the Jacobian

$$\frac{\partial \mathbf{u}}{\partial \mathbf{a}} = -\mathbf{S}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{S}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\textsc{sp}(\mathbf{G}\mathbf{u}) \in \mathbb{R}^{n\times 3f}. \tag{23}$$

A similar equation holds for $\mathbf{v}$; in this section, we frequently omit derivations for the term $v, \mathbf{v}$; one can substitute $u, \mathbf{u}$ with $v, \mathbf{v}$.

7.3 Injectivity-promoting Functionals: Many Possibilities

Now we derive objective functionals V that promote injectivity by penalizing the deviation of the Neumann data from prescribed values. We propose a few such functionals, and each functional can be used standalone. We compare these functionals in §7.5.

7.3.1 Neumann Residual. The most natural choice is to measure the Neumann error using an L_2 norm, for u, v, respectively:

$$V_1 = V_1^u + V_1^v,$$

where

$$\begin{aligned} V_1^u &= \frac{1}{2}\|\mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} - \mathbf{g}_u\|^2 \\ &= \frac{1}{2}\|\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} - \mathbf{R}\mathbf{g}_u\|^2 \quad \text{since } \mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} = \mathbf{0} \\ &= \frac{1}{2}\|\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathcal{F}(\mathbf{A}; \mathbf{b}_u) - \mathbf{R}\mathbf{g}_u\|^2. \end{aligned} \tag{24}$$

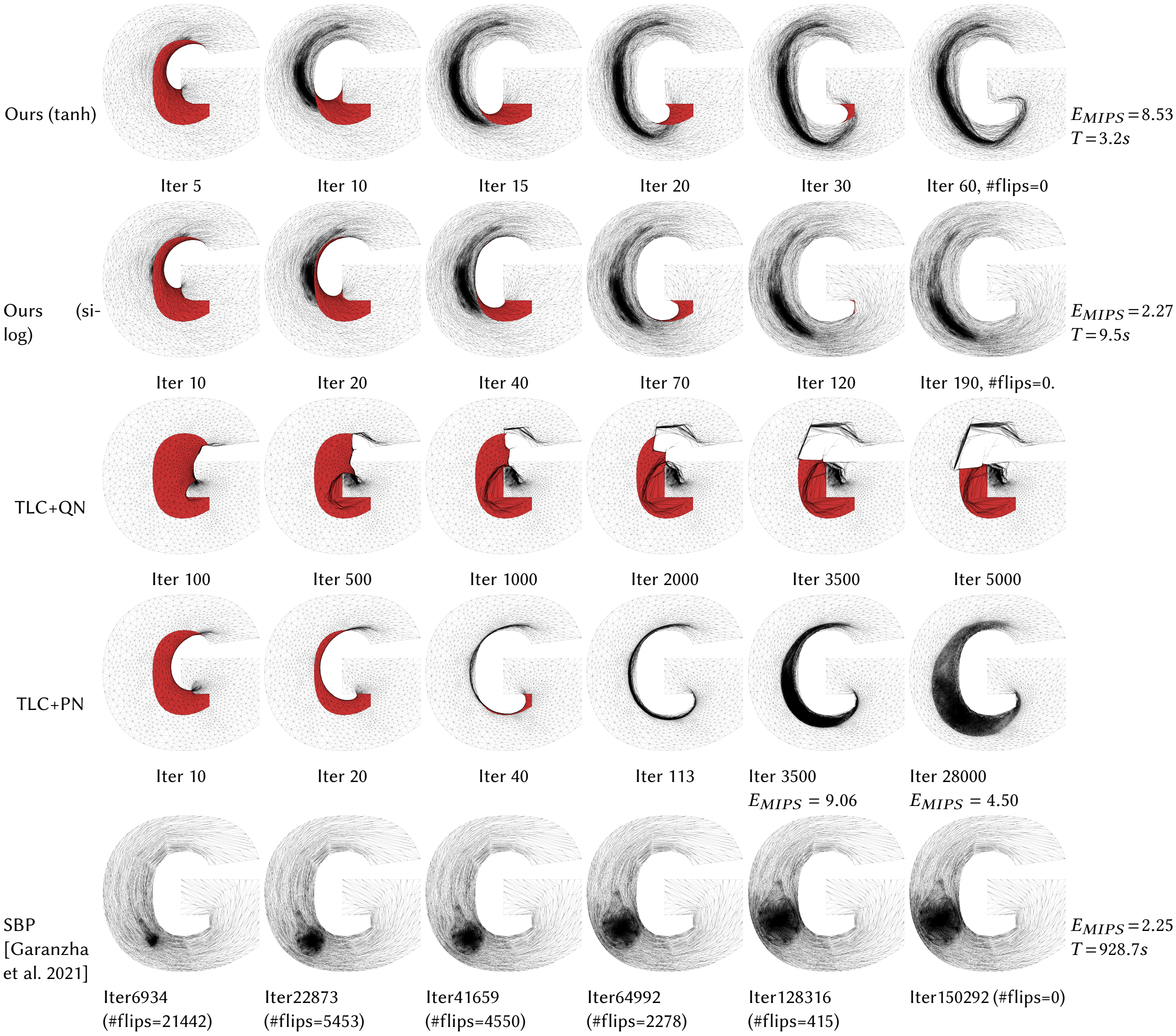


Fig. 5. Convergence paths of our methods and related works. We optimize the **A**-Dirichlet energy using Adam [Ling et al. 2022]. Ours (tanh) converges in fewer than 60 iterations; ours (si-log) converges to a map with lower conformal distortions. TLC [Du et al. 2020] using a quasi Newton (QN) solver produces a non-smooth trajectory, with much larger numbers of iterations. TLC with a projected Newton (PN) solver finds an injection with large distortions, taking many iterations to improve it. Soft barrier progression (SBP) [Garanzha et al. 2021] also requires many orders-of-magnitude more iterations than ours: Note their flipped triangles are very tiny, not visible from the figures. For all methods, we call it one iteration if the gradient oracle is called once: oracles of TLC+QN and SBP do not require linear solve; ours requires a Laplacian solve; TLC+PN requires solving a Hessian system (4× larger than Laplacian).

Intuitively, this functional aims to push the mesh toward the inside of the boundary. The gradient is:

$$\begin{aligned}\frac{\mathrm{d}V_1^u}{\mathrm{d}\mathbf{a}} &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}\mathbf{a}}\|\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u}-\mathbf{R}\mathbf{g}_u\|^2 \in \mathbb{R}^{3f\times 1}\\ &= \left[\mathbf{G}^\top \textsc{sp}(\mathbf{G}\mathbf{u}) + \mathbf{G}^\top\mathbf{A}\mathbf{G}\frac{\partial \mathbf{u}}{\partial \mathbf{a}}\right]^\top (\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u}-\mathbf{R}\mathbf{g}_u)\\ &= \textsc{sp}(\mathbf{G}\mathbf{u})^\top\left(\mathbf{I}_{2f} - \mathbf{G}\mathbf{S}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{S}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\right)\mathbf{G}\,\overbrace{(\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u}-\mathbf{R}\mathbf{g}_u)}^{\text{Neumann error}}.\end{aligned} \tag{25}$$

Remark. $\mathbf{G}\mathbf{S}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{S}]^{-1}\mathbf{S}^\top\mathbf{G}^\top$ *frequently appears in inverse problems, spreading the Neumann data error at the boundary into the interior through the* **A***-harmonic extension. The gradient formula reflects the philosophy that errors in the Neumann boundary data should be propagated and attributed to tensors in the interior.*

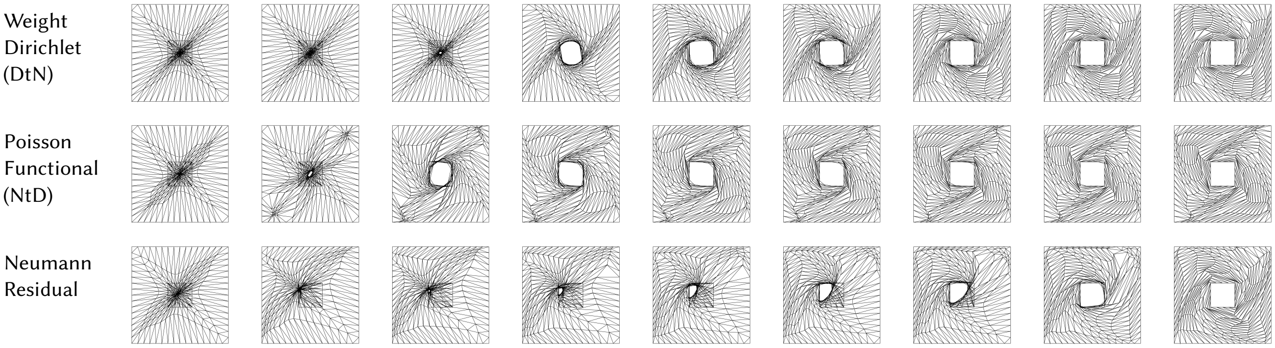


Fig. 6. Our methods (si-log) using different functionals consistently produce bijective maps. Different functionals in our framework produce diversed convergence paths on the example of rotated inner square by π: (a) Weighted Dirichlet: blank-area expansion; (b) Poisson functional: global spinning; (c) Neumann residual: normal alignment.

7.3.2 *Sobolev Neumann Residual.*

Replacing the L_2 norm in V_1, written in the equivalent form $\|\mathbf{G}^\top\mathbf{AGu}-\mathbf{Rg}_u\|^2$, with the Sobolev-type norm $\|\cdot\|^2_{(\mathbf{G}^\top\mathbf{AG})^\dagger}$, yields a second functional $V_2=V_2^u+V_2^v$:

$$\begin{aligned}V_2^u =&\frac{1}{2}\|\mathbf{G}^\top\mathbf{AGu}-\mathbf{Rg}_u\|^2_{(\mathbf{G}^\top\mathbf{AG})^\dagger}\\ =&\underbrace{\frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{AGu}-\frac{1}{2}\mathbf{g}_u^\top\mathbf{b}_u}_{\text{Weighted Dirichlet } V_3^u}+\underbrace{\frac{1}{2}\mathbf{g}_u^\top\mathbf{R}^\top(\mathbf{G}^\top\mathbf{AG})^\dagger\mathbf{Rg}_u-\frac{1}{2}\mathbf{g}_u^\top\mathbf{b}_u}_{\text{Poisson functional } V_4^u}.\end{aligned} \tag{26}$$

Here, the pseudo-inverses $(\cdot)^\dagger$ are well-defined via solvability of the Poisson problem:

$$\mathbf{g}_u^\top\mathbf{R}^\top\mathbf{1}=\mathbf{g}_u^\top\mathbf{1}=0,\quad \text{with}\quad \mathbf{S}^\top\mathbf{G}^\top\mathbf{AG1}=\mathbf{0}.$$

V_2 splits into two terms: the weighted Dirichlet energy and the Poisson functional, which are geometrically-meaningful functionals when extra determinant condition holds as we will discuss.

7.3.3 *Weighted Dirichlet Energy.*

(Applicable when determinant ≥ 1.) This energy is the default choice in our paper. We can stack the $\mathbf{u},\mathbf{v}$ components in a complex vector and measure its norm, for our particular Neumann data $\mathbf{R}[\mathbf{g}_u,\mathbf{g}_v]=[\mathbf{Dv},-\mathbf{Du}]$:

$$(\mathbf{G}^\top\mathbf{AGu}-\mathbf{Rg}_u)+\mathrm{i}(\mathbf{G}^\top\mathbf{AGv}-\mathbf{Rg}_v)=(\mathbf{G}^\top\mathbf{AG}+\mathrm{i}\mathbf{D})(\mathbf{u}+\mathrm{i}\mathbf{v})$$

By choosing another Sobolev-type norm $\|\cdot\|_{(\mathbf{G}^\top\mathbf{AG}+\mathrm{i}\mathbf{D})^\dagger}$—again the pseudo-inverse is well defined provided $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right)\geq 1$ holds (§A)—we arrive at the weighted Dirichlet energy

$$\begin{aligned}V_3=&\frac{1}{2}\|(\mathbf{G}^\top\mathbf{AG}+\mathrm{i}\mathbf{D})(\mathbf{u}+\mathrm{i}\mathbf{v})\|_{(\mathbf{G}^\top\mathbf{AG}+\mathrm{i}\mathbf{D})^\dagger}\geq 0\\ =&\underbrace{\frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{AGu}-\frac{1}{2}\mathbf{g}_u^\top\mathbf{b}_u}_{V_3^u\ngeq 0}+\underbrace{\frac{1}{2}\mathbf{v}^\top\mathbf{G}^\top\mathbf{AGv}-\frac{1}{2}\mathbf{g}_v^\top\mathbf{b}_v}_{V_3^v\ngeq 0}.\end{aligned} \tag{27}$$

The two terms V_3^u, V_3^v are not necessarily positive, but their sum is. The energy we obtain, $\frac{1}{2}(\mathbf{u}-\mathrm{i}\mathbf{v})^\top(\mathbf{G}^\top\mathbf{AG}+\mathrm{i}\mathbf{D})(\mathbf{u}+\mathrm{i}\mathbf{v})$, is a generalization of the LSCM energy [Lévy et al. 2002], and further falls back to our weighted Dirichlet energy with boundary fixed. In §F.3, we provide a simpler interpretation of the weighted Dirichlet energy V_3 as a tight upper-bound of an area functional.

Viewing $\mathbf{u},\mathbf{v}$ as functions of $\mathbf{A}$, V_3 becomes a nonlinear function of $\mathbf{A}$, involving the *weak-form of the Dirichlet-to-Neumann (DtN)*:

$$V_3^u=\frac{1}{2}\mathbf{b}_u^\top\mathcal{S}(\mathbf{A})\mathbf{b}_u-\frac{1}{2}\mathbf{g}_u^\top\mathbf{b}_u, \tag{28}$$

using which the derivative of V_3 can be computed as follows:

$$\begin{aligned}\frac{\mathrm{d}V_3^u}{\mathrm{d}\mathbf{a}}&=\frac{1}{2}\left[\mathbf{u}^\top\mathbf{G}^\top\textsc{sp}(\mathbf{Gu})+2\mathbf{u}^\top\mathbf{G}^\top\mathbf{AG}\frac{\partial\mathbf{u}}{\partial\mathbf{a}}\right]^\top\\ =&\frac{1}{2}\textsc{sp}(\mathbf{Gu})^\top\left(\mathbf{I}_{2f}-\mathbf{GS}[\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}]^{-1}\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\right)\mathbf{Gu}=\frac{1}{2}\textsc{sp}(\mathbf{Gu})^\top\mathbf{Gu}.\end{aligned}$$

7.3.4 *Poisson Functional.*

(Applicable when determinant = 1.) As the "dual" of the weak-form Dirichlet-to-Neumann V_3, we have the Poisson functional V_4:

$$V_4=\underbrace{\frac{1}{2}\mathbf{g}_u^\top\mathbf{R}^\top(\mathbf{G}^\top\mathbf{AG})^\dagger\mathbf{Rg}_u-\frac{1}{2}\mathbf{g}_u^\top\mathbf{b}_u}_{V_4^u\ngeq 0}+\underbrace{\frac{1}{2}\mathbf{g}_v^\top\mathbf{R}^\top(\mathbf{G}^\top\mathbf{AG})^\dagger\mathbf{Rg}_v-\frac{1}{2}\mathbf{g}_v^\top\mathbf{b}_v}_{V_4^v\ngeq 0} \tag{29}$$

where $\mathbf{R}^\top(\mathbf{G}^\top\mathbf{AG})^\dagger\mathbf{R}$ is the *weak-form Neumann-to-Dirichlet (NtD).*

The proof that $V_4\geq 0$ when $\mathbf{A}\geq 0$, $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right)=1,\forall j$ holds and more justification can be found in Appendix §E. We also derive its gradient formula in Appendix §E, where we differentiate into the inverse of system $\mathbf{G}^\top\mathbf{AG}$ instead of $[\mathbf{S}^\top\mathbf{G}^\top\mathbf{AGS}]$. Note V_4 does not involve $\mathbf{u},\mathbf{v}$ in any way, unlike previous functionals.

7.4 Discrete Injectivity at Functional Minimizers

The many options for functionals reflect the versatility and relevance to map injectivity of our framework, and optimizing any of the (nonconvex) functionals to the global minimum leads to an injective map when it exists:

Proposition 7.1. *When the boundary positions and mesh connectivity allow a locally injective map to exist, global minimizers of any functional in §7.3.1 to §7.3.4 in the space of* $\mathbf{A}\geq 0$ *satisfying* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right)=1,\forall j$ *are injective maps.*

Note these functionals can still be nonconvex in the variable $\mathbf{A}$, so there is no theoretical guarantee that the gradient descent will find the global minimum; in practice, however, we found our methods always yield injective maps. The nonconvexity is expected since the problem we studied is intrinsically nonconvex: For instance, in Figure 6 there are two injective maps that correspond to either $+\pi$ or $-\pi$ rotation. They are disconnected globally optima, indicating the injective mapping problem must be nonconvex.

The determinant condition can be relaxed, but the current form is sufficient for our purpose. For the regular and Sobolev Neumann residual functionals and the weighted Dirichlet energy, this result is a corollary of Theorem 6.4; for the Poisson functional, it can be shown by using a Lagrangian multiplier argument.

Proposition 7.2. *The functionals in §7.3.1 to §7.3.4 are differentiable as functions of* $\mathbf{A}$.

Proposition 7.2 holds since in this section we have written down their derivatives. Thus, we successfully bypass the vanishing gradient and non-smoothness issues of the area functional, with an approach different from the regularization method [Du et al. 2020].

7.5 Comparing Injectivity Functionals

In this paper, we choose $V = V_3$, weighted Dirichlet energy, as the default choice, unless specified otherwise. But these functionals optimize for injectivity by different means. Figure 6 visualizes the convergence path when optimizing different functionals from §7.3; details of the algorithm are postponed to §9. Adapting an example from Du et al. [2020], we recover a bijective map under positional constraints that rotate the inner square by 180°, while fixing the outer square, illustrating the preference of different functionals for attaining injectivity.

For the weighted Dirichlet energy, the map improves slowly at the beginning and starts to improve dramatically when a small hole appears at the center, after which the weighted Dirichlet energy "realizes" that its value can be greatly decreased by emptying the inner square.

For the Neumann residual, symmetric pattern is broken by moving the singularity in the center to the left-up corner of the inner boundary. Then the map orientation starts to align with the normal to the boundary. The region where the map aligns with the outward normal keeps growing, and eventually the normal alignment propagates to the whole inner boundary.

For the Poisson functional, progress is made across the entire domain: Unlike the other functionals in which unfolding gradually spreads out, in this example the entire domain adjusts to conform to the boundary normal orientation. It converges in the least iterations.

8 ELLIPTICITY PARAMETERIZED VIA BELTRAMI

We still have to handle the triangle-wise constraint:

$$\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1, \mathbf{A}_{(j)} \succeq 0. \tag{30}$$

Instead of introducing explicit constraints on $\det \mathbf{A}_{(j)}$, we parameterize $\mathbf{A}_{(j)}$ so that the constraints hold automatically. As a point of comparison, Wang and Solomon [2021] parameterize a positive tensor using its Cholesky factorization without any guarantee on its determinant. Note (30) implies strict inequality $\mathbf{A}_{(j)} \succ 0$.

8.1 Elliptic PDEs and Complex Beltrami Equation

Our parameterization of the tensor field is motivated by the equivalence between 2D elliptic systems and the complex Beltrami equation. In this section, we recall relevant concepts; see [Astala et al. 2008] for details. An important concept is the conjugate quasi-harmonic map:

Definition 8.1 (Conjugate quasi-harmonic map). For positive definite $\mathbf{A}(\cdot) \in \mathbb{R}^{2\times 2}$, map (u, v) is conjugate quasi-harmonic if

$$\nabla \cdot [\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x})] = 0, \quad \nabla \cdot [\bar{\mathbf{A}}(\mathbf{x})\nabla v(\mathbf{x})] = 0 \tag{31}$$

where $\bar{\mathbf{A}}(\cdot)$ is the (matrix) conjugate of $\mathbf{A}(\cdot)$: $\bar{\mathbf{A}}(\mathbf{x}) = *^{\top}\mathbf{A}(\mathbf{x})^{-1}*$, in which $*$ is the matrix $* := \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, the planar Hodge star operator.

This definition differs from Definition 4.1 in that we use the conjugate $\bar{\mathbf{A}}$ to compute $v(\cdot)$. That said, since $\bar{\mathbf{A}} = \mathbf{A}$ if $\det(\mathbf{A}) = 1$, our choice in (30) to take $\det(\mathbf{A}) = 1$ also makes the map conjugate quasi-harmonic; again, the choice does not limit our ability to represent any smooth bijective map.

A map (u, v) can be compactly written as a complex-valued function $f : \Omega \to \mathbb{C}$, $f(z) = u(z) + iv(z)$. Via a change of variables to $\mu, \nu : \Omega \to \mathbb{C}$ that $|\mu| + |\nu| < 1$, the complex Beltrami equation [Ahlfors 2006; Daripa 1993; Gardiner and Lakic 2000] is equivalent to the system of PDEs (31), via the reparameterization

$$\mathbf{A} = \frac{1}{|1+\nu|^2 - |\mu|^2} \begin{bmatrix} |1-\mu|^2 - |\nu|^2 & -2\Im(\mu) \\ -2\Im(\mu) & |1+\mu|^2 - |\nu|^2 \end{bmatrix}. \tag{32}$$

Our condition $\det \mathbf{A}(\mathbf{x}) \equiv 1$ becomes $\nu(z) \equiv 0$ and $|\mu| < 1$.

8.2 Tensor Parameterization: A Prior on Conformality

Our change of variable $\mathbf{A} \mapsto \mu$, however, still requires the (convex) constraint $|\mu| \le 1$ to ensure $\mathbf{A} \succ 0$. Aiming at a fully unconstrained optimization, we apply an additional change of variables.

We consider a bijection $\mu \leftrightarrow \xi$ between unit disk $\mathbb{D}$ and the complex plane $\mathbb{C} = \mathbb{R}^2$ given by

$$\arg \mu = \arg \xi, \quad |\mu| = r(|\xi|),$$

where $[0, +\infty) \underset{r^{-1}}{\overset{r}{\rightleftharpoons}} [0, 1)$; we use the notation $\mu = f[r](\xi)$.

In this paper, r is one of the two sigmoid functions.

- shifted-inverse-log (si-log): $r(x) = 1 - 1/(1 + \log(1 + x))$
- hyperbolic tangent: $r(x) = \tanh(x)$

We choose r to be the hyperbolic tangent function by default, and the shifted-inverse-log function if close-to-conformal maps are desired.

As Figure 7 shows, shifted-inverse-log (si-log) gives a strong prior that discourages large angle distortions: The closer it approaches $|\mu| \to 1$, the exponentially longer a distance $|\xi|$ has to travel for improving $|\mu|$. Our μ can grow arbitrarily close to unit length if needed without a bounding parameter on $\textsc{cond}(\mathbf{A}_{(j)})$, unlike approaches that constrain conformal distortion [Aigerman and Lipman 2013; Lipman 2012].

We use $\tanh(\cdot)$ in our model whenever a well-behaving regularizer $R(\cdot)$ has been chosen, or the mesh is subject to distortion to the extreme level of Figure 4. Our use of $\tanh(\cdot)$ was initially motivated by possible equivalence to optimization in the hyperbolic space under the Poincaré disk model, which we leave for future work.

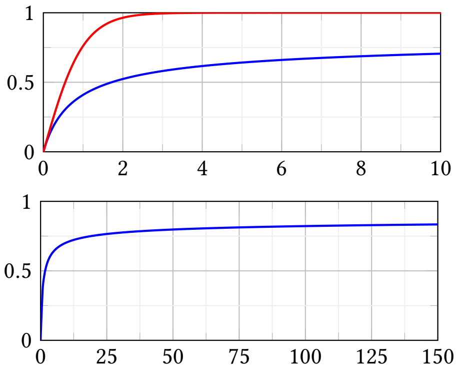
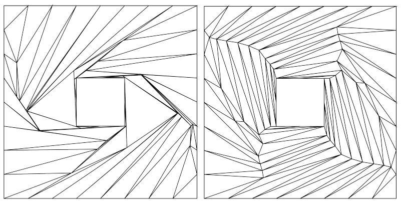


Fig. 7. The curve $|\xi| \to |\mu|$ via $r(x) = \text{si-log}(x)$, or $r(x) = \tanh(x)$.



tanh si-log

Fig. 8. The weighted Dirichlet is minimized using either parameterization, and the first inversion-free map each parameterization encounters is shown: si-log yields smaller angle distortion. Remarkably, the first injection si-log encounters (b), is already similar to running TLC until convergence [Du et al. 2020, Figure 5], while the first injection tanh (a) encounters is similar to that of TLC. Recall that TLC explicitly minimizes a conformal energy, while our weighted Dirichlet does not.

8.3 Tensor Parameterization: A Summary

In summary, in this section we advocate the per-triangle change of variable compositing two bijections: $\mathbf{A} \to \mu \to \xi$, with $\mathbf{m}_j^{-1}\mathbf{A}_{(j)} = \mathcal{A}(r(\xi_j))$, using

$$\mathcal{A}: \mathbb{D} \to \mathbb{R}^{2\times2}, \mathcal{A}(\mu) = \frac{1}{1-|\mu|^2}\begin{bmatrix} |1-\mu|^2 & -2\Im(\mu) \\ -2\Im(\mu) & |1+\mu|^2 \end{bmatrix} \tag{33}$$

which we compactly summarize via

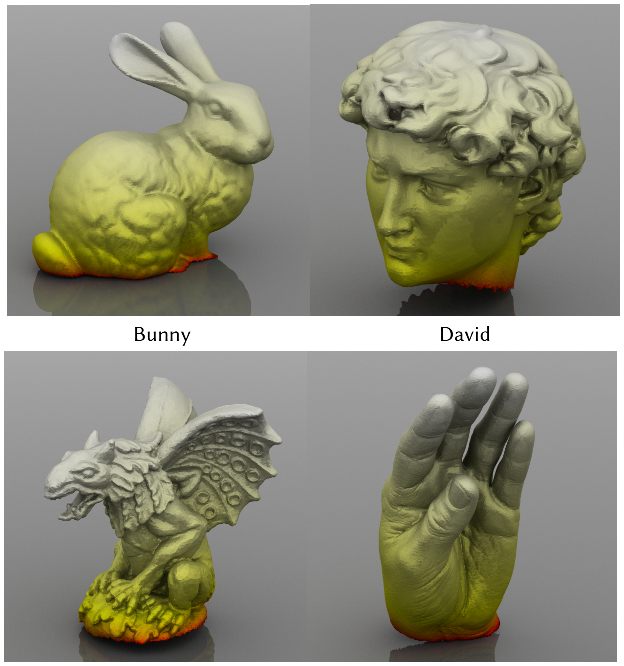
$$\mathbf{a} = p(\xi; \mathbf{m}), \quad \text{where } \mathbf{a} \in \mathbb{R}^{3f}, \xi \in \mathbb{C}^f.$$

Our optimization variable now becomes $\xi \in \mathbb{C}^f$, a complex number per triangle.

There are multiple advantages of optimizing for ξ rather than $\mathbf{A}$, in addition to getting rid of explicit constraints on $\det \mathbf{A}_{(j)}$ or $\textsc{cond}(\mathbf{A}_{(j)})$. Without explicitly modifying the objective, our parameterization makes the *numerical procedure* prioritize robust convergence paths and favor conformality by choosing the si-log sigmoid function.

9 OVERALL ALGORITHM AND VALIDATIONS

In this section, we state the overall algorithm and provide validations. Implementing our base algorithm amounts to translating matrix expressions to, e.g., MATLAB.



Bunny David

Gargoyle Hand

Fig. 9. Rest poses of meshes to be mapped to a planar domain. Red colors indicate where the boundary is.

9.1 The Base Algorithm: Unconstrained Optimization

After the change of variable $\mathbf{u}, \mathbf{v} \to \mathbf{A} \to \xi$, Problem (18) becomes a totally *unconstrained* optimization problem:

$$\min_{\xi} \quad Q\Big(\mathcal{F}(\mathbf{A}(\xi), b_u), \mathcal{F}(\mathbf{A}(\xi), b_v), \mathbf{A}(\xi)\Big). \tag{34}$$

With $\frac{\mathrm{d}Q}{\mathrm{d}\xi} = \frac{\mathrm{d}\mathbf{a}}{\mathrm{d}\xi}\frac{\mathrm{d}Q}{\mathrm{d}\mathbf{a}}$, the gradient w.r.t. the solo variable ξ, we optimize the energy Q using off-the-shelf gradient-based methods.

Visualization of results. As shown in Figure 10, using L-BFGS [Wright et al. 1999] to minimize any of the functionals—weighted Dirichlet, Poisson functional, and Neumann residual—leads to injective maps on the 30 shape-to-letter examples provided in [Du et al. 2020].

We focus on validating the weighted Dirichlet energy, and all results use this default functional unless specified otherwise. We do not find a case that only weighted Dirichlet succeeds and the other functionals cannot, but we leave it for future work to further explore the other functionals or even combine them.

9.2 Auxiliary Solvers for Post-Processing

In the vast majority of the cases, our gradient solver in §9.1 already converges to exact injections. There are cases, especially in conformal maps involving extreme numerics, where first-order methods can slow down when they almost reach convergence. Hence, in this section we describe a Newton-type solver and an alternating solver we can optionally use to refine our solution.

A Newton solver in the joint space. It is difficult to directly apply a second-order method to problem (34): The optimization variable ξ or $\mathbf{A}$ has global impact, so its Hessian is dense. As a simple workaround,

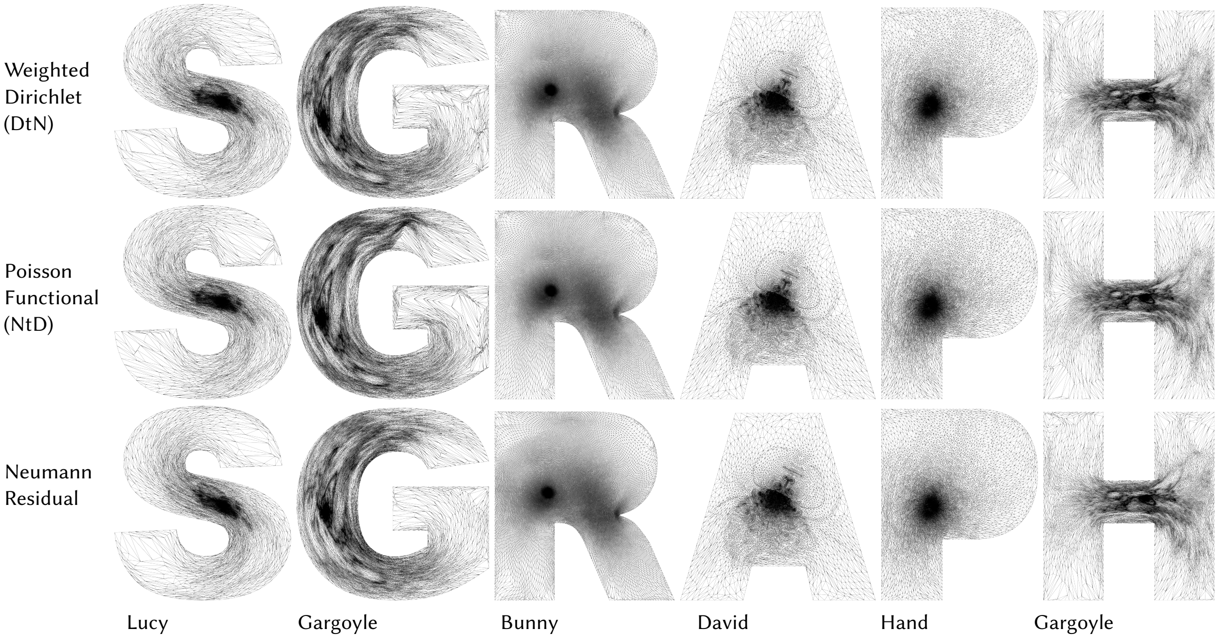


Fig. 10. Inversion-free maps produced by different functionals in our method (si-log).

we design a Newton solver operating in the joint space $(\mathbf{u}, \mathbf{v}, \xi)$ to optimize the $\mathbf{A}(\xi)$-Dirichlet energy of $\mathbf{u}, \mathbf{v}$. See details in §F.

Although our Newton solver as a standalone algorithm succeeds on less challenging examples, we only apply it as a post-processing step: We find in early phases searching for injective maps, algorithms that progress over $\mathbf{u}, \mathbf{v}, \mathbf{A}$ jointly can follow worse paths on challenging examples than our base method operating in the space of $\mathbf{A}$ only.

A greedy alternating solver. Alternatively updating either $[\mathbf{u}, \mathbf{v}]$ or $\mathbf{A}$ to optimality while fixing the other variable leads to a greedy local-global style solver, which falls back to the post-processing algorithm suggested by Xu et al. [2011]. Even in the close-to-injection regime, alternating descent often but not always removes left-over tiny flips, but when it works it can be effective in removing flipped triangles under extreme numerics. In addition, this alternating solver is cheap to apply, so we incorporate it in our pipeline.

Final pipeline. We find the following strategy is most efficient in our framework: First, apply the first-order gradient descent, usually all flips are removed quickly. Otherwise, the common pattern is that there are $1 \sim 10$ leftover flips with tiny areas that are time-consuming to remove: Then we apply the post-processing step that alternatively attempts the cheaper routine [Xu et al. 2011] and invokes the second-order Newton solver; when [Xu et al. 2011] fail we discard its iterations. This strategy passes the entire benchmark [Du et al. 2020] (§10).

The post-processing step is not critical for practical purposes: Typical changes made at this stage are local adjustments around tiny flipped triangles with area smaller than $10^{-6} \sim 10^{-20}$, which are not directly visible.

9.3 Non-Euclidean Reference Poses

Given the positional constraints, whether or not an injective map exists only depends on the mesh topology, not the geometry of rest pose triangles—which only matters for measuring distortion. Thus, we have some degrees of freedom in choosing the reference triangle, the one in which $\mathbf{G}$ and $\mathbf{m}$ are computed. Such degrees of freedom in choosing a reference/source triangle also appear in [Du et al. 2020; Liu et al. 2018]. The choice of reference triangle will not affect our capacity to represent any injection. By default, we choose the reference triangle as its rest pose, so our algorithm initializes from the generalized Tutte's embedding with cotangent weights.

9.4 A Further Improvement with Cheaper Gradient Steps

Since we carefully formulate the problem to be well-conditioned and unconstrained, our problem allows applying an even simpler and cheaper Adam-type solver [Ling et al. 2022]. With a very large step size, the Adam-type solver converges faster than L-BFGS.

[Ling et al. 2022] is a simple first-order method, advancing in the gradient direction without any step-size search. It is much cheaper than L-BFGS, whose line search and gradient history add extra overhead. Remarkably, Problem (34) remains stable even if we use relatively large learning rates (step sizes), like 0.1 or larger in Adam. Adam is rarely a choice in geometry processing, but we manage to apply Adam to minimizing generic energies (§11) with competitive performance, made possible by the fact that our variable $\mathbf{A}$ or ξ has a global impact over $\mathbf{u}, \mathbf{v}$, following [Wang and Solomon 2021].

We primarily use Adam: Testings with L-BFGS are to demonstrate that our method is agnostic to the choice of solver, unlike the method of Wang and Solomon [2021].

10 COMPARISONS AND DISCUSSIONS

Despite the large number of works on injective maps, the vast majority cannot handle generic positional constraints, often due to the requirement of injective intermediate maps for applying interior point methods.

That said, we are in the exactly same setting of Du et al. [2020], searching injective mapping with the entire boundary constrained. Thus we refer to their work for discussion and experiments that show prior works can fail to produce injections under positional constraints. An exception is [Garanzha et al. 2021], to which we also compare. All evaluation and experiments are done with our core variant $R = 0, V = V_3$ (weighted Dirichlet) unless specified.

Evaluation metric. Du et al. [2020] provide a dataset including 10743 meshes with given boundary maps. The dataset contains challenges of two kinds:

(1) The vast majority, including examples taken from [Liu et al. 2018] with boundary obtained by [Jiang et al. 2017], requires relatively localized adjustment to the mesh: The Tutte initialization usually is not far from injection, requiring adjustment around concave corners.

(2) Examples requiring substantially global deformations, including many shape-to-nonconvex-letter examples in the benchmark, especially the letters "S, G, H."

For < 8% meshes in category (1), the input contains very thin triangles that may cause slow progress due to numerical issues: For our method, we modify these reference triangles to have angles larger than 25°, affecting only our map initialization. See details in Appendix §F.

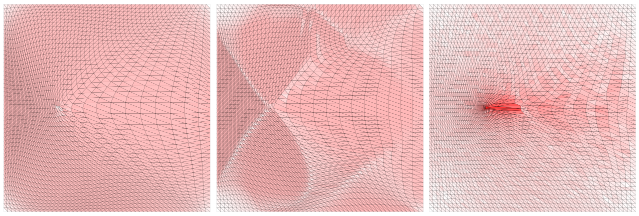
Following Du et al. [2020], we measure the conformal distortion using the MIPS energy [Hormann and Greiner 2000]:

$$E_{MIPS} = \frac{1}{\mathcal{A}} \int \left(\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_1} - 2 \right),$$

where $\sigma_1 \geq \sigma_2$ are the singular values of the Jacobian, and $\mathcal{A}$ is the area of the rest pose.

While we are reporting the MIPS energy in the paper, we emphasize that it may not be a fair metric; our base methods also do not explicitly minimize a conformal energy like MIPS. As shown in Figure 11, our method with si-log produces a map with small maximal conformal distortion. On this example, our map is very similar to Bounded Distortion Mapping (BDM) [Lipman 2012], but without their aliasing patterns that depend on the triangulation. Another injective map that minimizes the MIPS energy, produces a "spike" at the point constraint. This map is also found by our framework, using the method introduced in §11. Both ours and BDM produce maps that are smoother with distortion more evenly distributed.

This example suggests that MIPS may not always be a desired metric since it can compromise smoothness at positional constraints. This pattern of large shearing at positional constraints can also be observed in inverse harmonic maps, the injection minimizing MIPS energy, as shown in Figure 24.

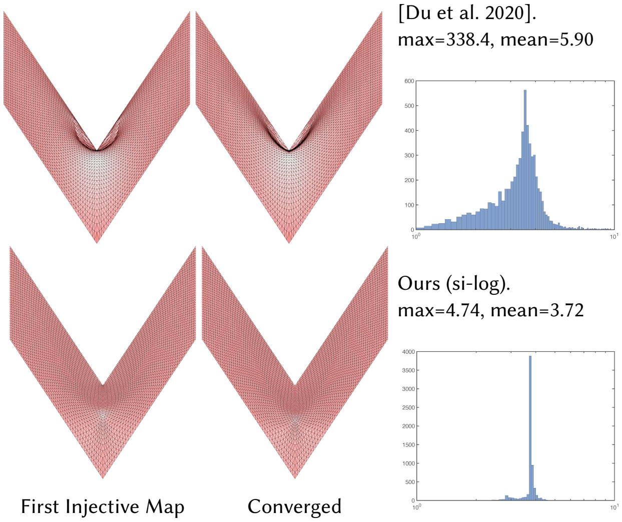


Ours (si-log). E_{MIPS}=1.01. max=3.11, mean=2.63.

[Lipman 2012]. E_{MIPS}=1.03. max=3.00, mean=2.63.

Ours (tanh+MIPS). E_{MIPS}=0.55. max=27.83, mean=2.00.

Fig. 11. In this example taken from Fu et al. [2015], the point in the center of the rectangle is moved to its left. We visualize $\sigma_1/\sigma_2 \geq 1$. Ours (si-log) reduces the maximal conformal distortion even without explicitly bounding it, producing a map similar to that of Bounded Distortion Mapping (BDM) [Lipman 2012]. However, BDM produces aliasing patterns that are sensitive to the triangulation, which is not the case for our method. Ours (tanh+MIPS) produces an injective map minimizing the MIPS energy, with distortion concentrated around the constraint.



[Du et al. 2020]. max=338.4, mean=5.90

Ours (si-log). max=4.74, mean=3.72

First Injective Map

Converged

Fig. 12. Distributions of $\sigma_1/\sigma_2 \geq 1$ over triangles. Our method (si-log) evenly distribute distortions into across the entire domain: Surprisingly, ratios are very concentrated even we do not explicitly encourage that. TLC+PN [Du et al. 2020], even when converges, produces injective maps with large distortion at the concave corner.

10.1 Comparison with Total Lifted Content [Du et al. 2020]

Total unsigned area (TUA), while being a simple and intuitive objective for injectivity, is challenging to optimize due to its combinatoric and non-smooth nature. Indeed, Xu et al. [2011] suggest their alternating algorithm for TUA as a post-processing step rather than a standalone algorithm. Improving on TUA, the key idea in total lifted content (TLC) is to regularize TUA into a smoother objective without suffering from vanishing gradients [Du et al. 2020].

Our methods share the same wisdom with TLC by considering a smoother alternative objective: TLC adds terms promoting MIPS energy, while (the core variant of) ours considers the **A**-parametric family of Dirichlet energies. Our methods spread out distortions broader than TLC, and Figure 12 provides a minimal example: The

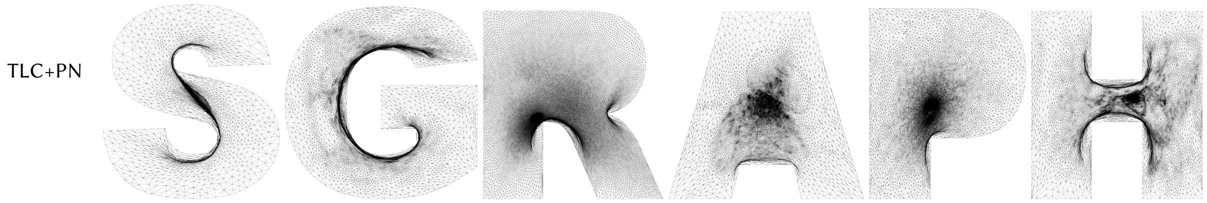


Fig. 13. Examples produced by TLC [Du et al. 2020]. We see recurring patterns in TLC that triangles are clustered along a few curves where distortions are concentrated, leading to a large conformal energy.

boundary of a rectangle is subject to vertical displacement proportional to the $|x|$-coordinate. TLC produces a concentration of distortion at the concave corner, while our method (si-log) produces a map with distortions that are more evenly distributed.

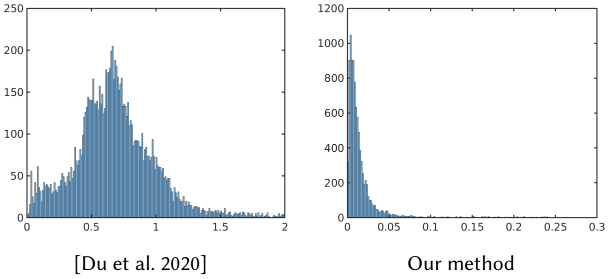
Our methods consistently outperform TLC in terms of speed and quality, by a large margin on challenging examples. On the Lucy example, Figure 2 compares the map provided by Du et al. [2020] with our method, plotting histograms of ratios of singular values. The distortion of the first injection that our method (si-log) found is an order-of-magnitude smaller, by any metric, such as the MIPS energy, or the mean/max of $\sigma_1/\sigma_2 \geq 1$.

Figure 5 compares the convergence paths. Both the si-log and tanh variants of our methods converge quickly. The tanh variant finds an injection in merely 60 Adam-gradient steps. In contrast, TLC with quasi-Newton (TLC+QN) fails to converge after 10^4 iterations, suggesting that it can be a challenging objective to optimize. TLC with projected Newton (TLC+PN) finds an injective map in 113 Newton steps but progresses slowly: After 28000 extra Newton iterations, the MIPS energy is still much larger than ours. The pattern of slow progress is similar to the case of Figure 12, in which TLC finds a suboptimal injection without spreading out distortion. Subsequent quality improvement over their initial injections can be extremely slow, or not possible [Du et al. 2022, Figure 7]. In contrast, our si-log variant finds an injective map whose MIPS is an-order-of-magnitude smaller in 190 iterations. We consistently observe similar patterns on individual examples.

When evaluating our methods on the benchmark [Du et al. 2020], we design the following strategy that efficiently finds injections with low conformal distortion: Since our injection functional does not explicitly optimize any conformal energy, first we run our si-log variant for at most 1000 iterations, and in $> 90\%$ of the cases, an injection is found in the given iterations. If not, rather than keep running gradient descent, we run the other variant (tanh), which often finds injections in much fewer iterations. Usually this indicates that some extreme distortion level is necessary for obtaining an injective map, and si-log has not necessarily slowed down the convergence. We always stop when the first injection is found.

Using the strategy above, we are able to quickly find injections with low distortion over the entire benchmark of [Du et al. 2020]. Figure 14 compares the distribution of MIPS energies over injective maps produced by TLC and our mixed strategy. MIPS energies by our method are significantly lower.

In sum, our method consistently yields better results than TLC on all examples we conduct in the paper: Appendix §F.3 provides a perspective, by equating injections as minimal surfaces.



[Du et al. 2020]

Our method

Fig. 14. The distribution of MIPS energy over the dataset [Du et al. 2020].

As a minor difference, TLC energy admits injective maps in its kernel only when the regularization weighting diminishes to zero. In contrast, our functional accepts all injective maps as its minimizers.

10.2 Comparison with Barrier-type Methods

Barrier-type methods, e.g., [Schüller et al. 2013; Smith and Schaefer 2015] are frequently adopted in practice due to their simplicity and generality. However, the per-element barrier term introduces artificial stiffness and ill-conditioning. Most importantly, in our setting with (hard) position constraints, barrier methods cannot be applied since there is no feasible/injective initial map.

Table 2. A comparison of quality measured by the MIPS energy on the shape-to-letter examples in Figure 10. Ours (si-log), without using any energy explicitly, can have a larger MIPS energy than SBP [Garanzha et al. 2021]; but surprisingly, there are exceptions (Lucy,Hand). In addition, our framework allows explicitly introducing a small amount of MIPS to reduce the energy value, as shown in the last column.

Example	SBP	Ours (si-log)	Ours (tanh+MIPS)
Lucy-to-G	2.24	2.24	0.60
Lucy-to-S	2.37	1.82	0.511
Gargoyle-to-G	1.18	3.69	2.98
Bunny-to-R	0.350	1.45	0.372
David-to-A	0.088	0.512	0.0843
Hand-to-P	0.781	0.391	0.0910
Gargoyle-to-H	0.717	2.25	1.548

Methods using a soft barrier provide an important exception avoiding many of these limitations: Coincidentally, the state-of-the-art barrier method [Garanzha et al. 2021] employs a soft barrier progression (SBP). SBP progressively sharpens the soft barrier to eventually resume infinity at 0, with a careful barrier scheduling. SBP has to use a particular form of energy, $(1-\theta)E_{MIPS} + \theta E_a$ that blends the Winslow functional E_{MIPS} and an area-preserving one E_a; they choose $\theta = 1/128$ in the released code. The scale-invariant part, Winslow functional, yields inverse harmonic maps with extreme boundary scaling, so some amount of (scale-dependent) E_a must be introduced to suppress the scaling in SBP.

Table 3. A comparison of timing (in seconds) on the shape-to-letter examples in Figure 10. Our methods are orders of magnitude faster on these examples than SBP [Garanzha et al. 2021].

Example	Ours (tanh)	Ours (si-log)	SBP
Lucy-to-G	3.13	9.48	928.7
Lucy-to-S	3.43	9.53	402.6
Gargoyle-to-G	2.34	6.91	219.5
Bunny-to-R	5.22	10.1	348.5
David-to-A	1.65	3.68	63.1
Hand-to-P	1.19	3.59	302.1
Gargoyle-to-H	1.42	5.85	336.3

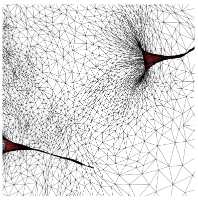


Fig. 15

Figure 5 compares the typical convergence path of SBP with ours: In general, SBP can take orders-of-magnitude more gradient steps than our method, and its convergence path can be different from ours. The Lucy example shows a typical convergence pattern of SBP: The majority of the mesh is quickly restricted to the interior side of the boundary, and a cluster starts to develop; there is a region with triangles densely packed in the center. A large number of triangles—mainly those in the center of clusters—are flipped, but they are too tiny to be visible. See the zoom-in view in Figure 15 for a typical pattern of inversions in SBP. Two "spikes" remain in the cluster at center, which only become visible after zooming in. The cluster of triangles moves upwards, and the flips are gradually resolved. The differences in paths reflect distinct objectives: TLC and ours would favor conformal/harmonic maps, while SBP favors inverse harmonic maps.

Indeed, SBP works in a regime complementary to ours: SBP must use energies with infinite barriers, while we optimize energies without infinite barriers, handling a distinct set of objectives. See details in §11, where we show that our framework successfully produces purely (quasi-)conformal bijections, or bijections with diverse objectives including area-preservation (Figures 10,23,21).

The timings of our method and [Garanzha et al. 2021] are reported in Table 3 for the shape-to-letter examples. Figure 16 provides a comparison of timing on individual meshes over the entire dataset [Du et al. 2020]. In both cases, our methods can achieve one to two orders of magnitude speedup over [Garanzha et al. 2021] in wallclock time.

Our reduction of computational cost may be even more significant than the reported improvement in wallclock timing. Our methods are implemented in MATLAB, running at only ~ 10% CPU rate; Garanzha et al. [2021] use an efficient C++ implementation with OpenMP, running at 100% CPU rate. So, when running large-scale tests on a single workstation, our implementation can run multiple instances simultaneously with little interference, allowing a further ~ 10× speedup in the amortized sense. Future work may also implement our method in more efficient languages or explore cheaper inexact linear solvers for further acceleration.

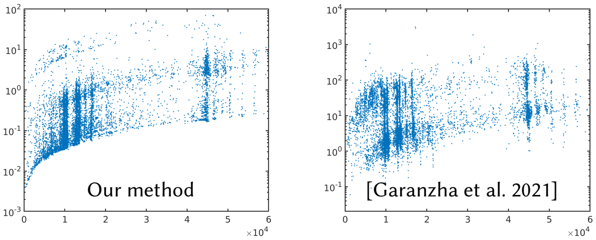


Fig. 16. The timing comparison between our method and Garanzha et al. [2021]. Each dot corresponds to one mesh in the dataset. For our method, most meshes are in the range of $0.01 \sim 5$ seconds, while for Garanzha et al. [2021] the typical range is $1 \sim 100$ seconds. The scale of the figure is set identical to that of [Du et al. 2022, Figure 6] for comparison purpose: Our method is still more than an order of magnitude faster than [Du et al. 2022].

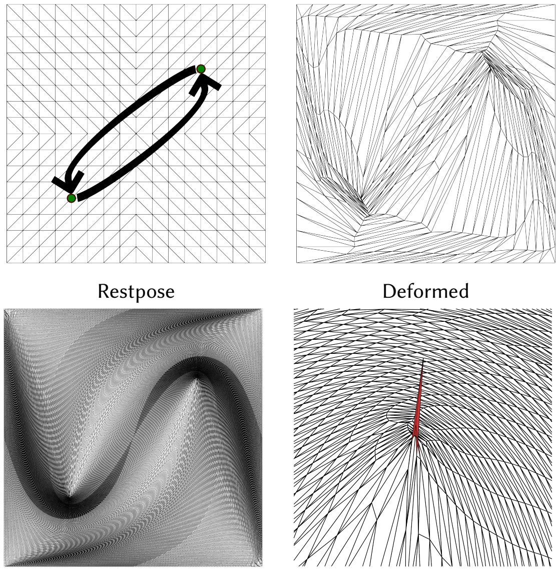
10.3 Extra Validations

Point-type constraints. Our methods can handle interior point-type constraints, by simply considering constrained interior vertices as part of the boundary. Figure 11 gives an example, in which the center point in a rectangular domain is displaced with the entire boundary fixed. Note incorporating interior constraints inevitably makes the domain nonconvex, a scenario for which our methods are designed over Tutte's embedding. In Figure 17, the positions of two points are swapped and our method (si-log) succeeds in producing an injective map.

However, when upsampling the mesh multiple times to a denser one, the spikes around point constraints become more extreme, and our method does slow down when it almost converges. Our method (si-log) still finds an injective map when the mesh is as large as 140 × 140, but fails to remove the inversions around the point within given iterations, for the mesh grid 160 × 160. This is not surprising, since elliptic PDEs are only well-defined for regional boundaries: Likely bi-quasi-harmonic systems should be considered. Alternatively, one can use small regional handles in lieu of point-type constraints.

Boundaries preventing injectivity. Since there are cases when injectivity inevitably fails, e.g., due to self-intersecting boundaries [Weber and Zorin 2014], an advantage of our barrier-free approach is the robustness under these cases. As shown in Figure 18, when the fixed boundary does not allow an injective map to exist, ours remains stable, while the barrier methods including Garanzha et al. [2021] explode. In this cases, ours yields *minimal surfaces.*

A Comparison with IsoTLC [Du et al. 2022]. The Isometric TLC (IsoTLC) [Du et al. 2022] supports different objectives from our methods. As shown in Table 4, our method (si-log) always yields smaller MIPS energies; however, ours can have much larger isometric energies, measured by $\max(\sigma_1, 1/\sigma_2)$ as considered in Du



Restpose Deformed

Upsampled to 160x160 Zoomed-in view at the spike.

Fig. 17. Another example of point constraints. The positions of two vertices on the rectangular domain are swapped, leading to an extreme level of distortion. The injective map produced by our method is shown. However, when upsampling the mesh to as large as 160 × 160, there are two extremely distorted spikes that our method fails to remove in given iterations.

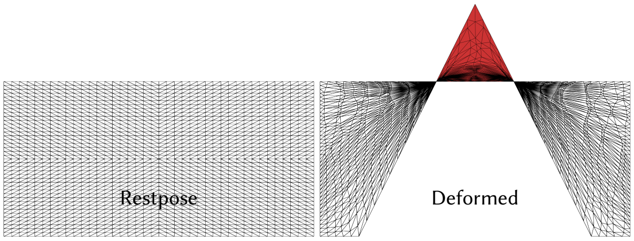
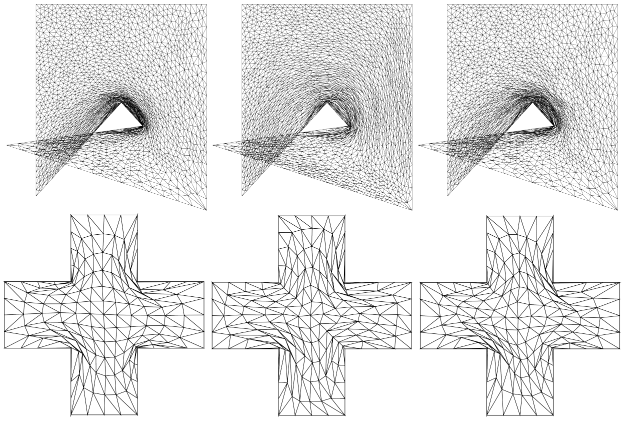


Fig. 18. Under positional constraints when an injective map does not exist, our methods remain stable. Inverted triangles are marked in red.

Table 4. A comparison with IsoTLC [Du et al. 2022]

	MIPS		Isometric	
Examples	Ours	IsoTLC	Ours	IsoTLC
Lucy-to-G	2.21	5.67	2.30e+08	6.74
Lucy-to-S	1.85	5.56	2.63e+08	6.31
Gargoyle-to-G	3.65	3.08	36.37	3.71
Bunny-to-R	1.48	2.76	139.1	1.91
David-to-A	0.433	1.62	5.71	0.853
Hand-to-P	0.387	4.27	797.5	2.77
Gargoyle-to-H	2.27	2.81	69.5	3.12

et al. [2022] (assuming $\sigma_1 \geq \sigma_2$). This is expected since ours only penalizes deviations from conformal maps and does allow scaling.



ARAP Area-Preserving Mass Spring

Fig. 19. Inversion-free maps via minimizing variant energies—blended with A-weighted Dirichlet—in the space of quasi-harmonic maps.

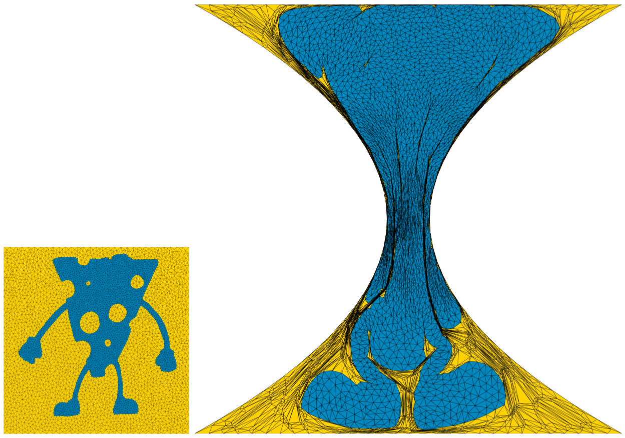


Fig. 20. Collision avoidance by way of bijectivity: Cheeseman with multiple holes is put in an hourglass. We optimize the ARAP energy, only over the shape mesh (in blue). Our method, successfully ensuring bijective meshes for the shape and air, allows a unified treatment of contact detection and response. Note that the air mesh layer (in yellow) is subject to extreme distortion, demanding exact and discrete injectivity.

11 APPLICATION: QUASI-HARMONIZATION FOR OPTIMIZING WITHIN DIFFEOMORPHISMS

Next, we will show how to optimize a conventional energy while restricting the shape within the space of diffeomorphisms. In the optimal control problem (10), we can choose $R = E(u, v)$ as an energy derived in physics or geometry, providing a generic alternative to constraining $\det(\mathbf{J}_{(j)}) > 0, \forall j$.

Similar to variational harmonic maps [Ben-Chen et al. 2009], which restrict to a linear space of harmonic maps, we restrict to the *nonlinear* space of diffeomorphisms. Unlike harmonic maps, our theoretical results show that this formulation is optimizing in the *full* space of diffeomorphisms.

Precisely speaking, in this section we optimize $Q = V_3 + \eta E$, where V_3 is the weighted Dirichlet, $\eta = 10^{-4}$, and E is a distortion energy.

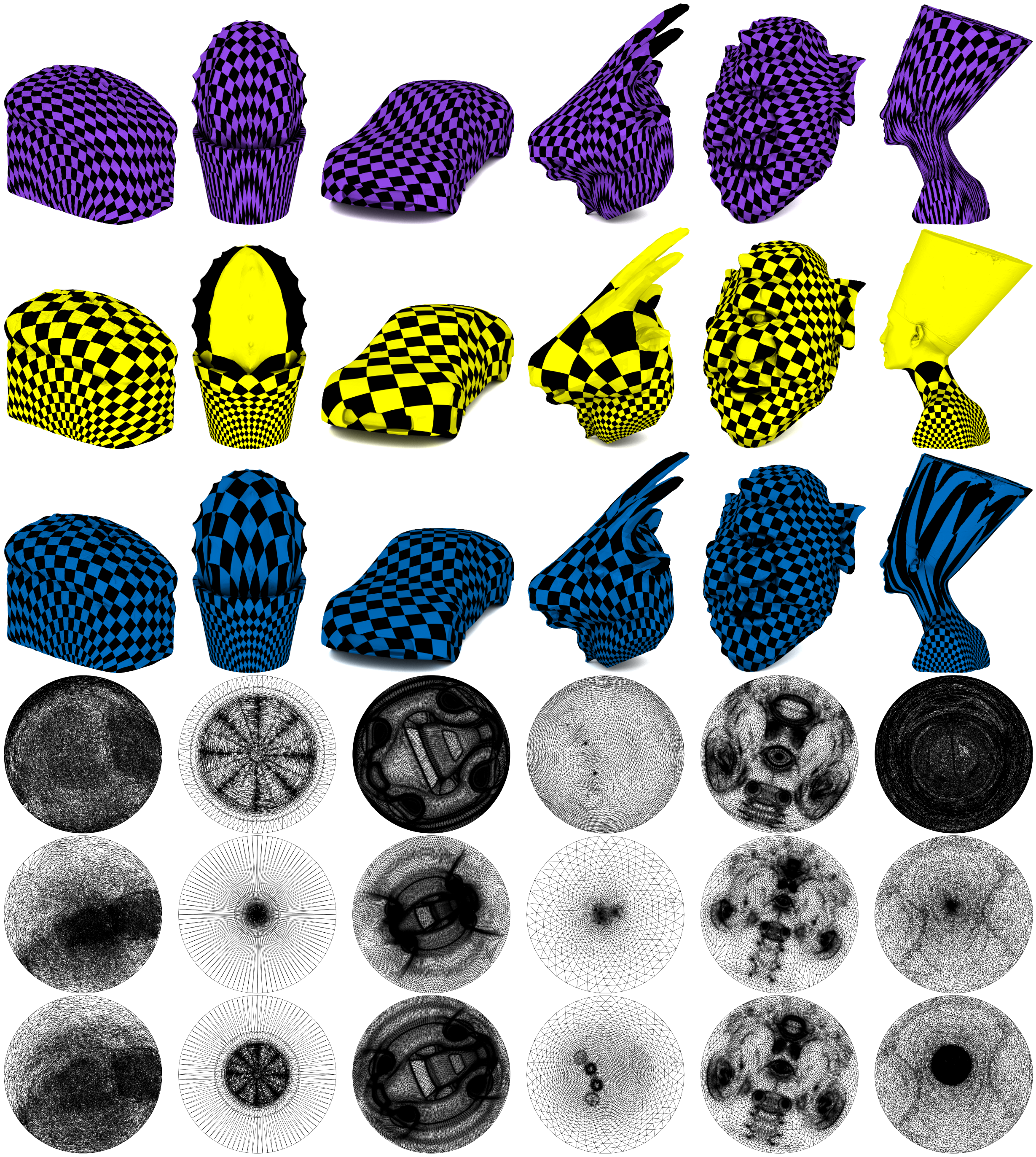


Fig. 21. Bijective parameterization produced by our methods with the boundary constrained to be a disk, though our method does support other nonconvex boundaries. Row 1,4: Area-preserving. Row 2,5: Our method with si-log. Row 3,6: ARAP.

We always use the tanh parameterization for optimizing a given energy, as the preference for conformality is no longer needed.

11.1 Inversion-Permitting Energies

Our framework is most valuable in optimizing an energy that does not penalize inversions, such as ARAP, removing all flips immediately. Specifically, we test with the following energies.

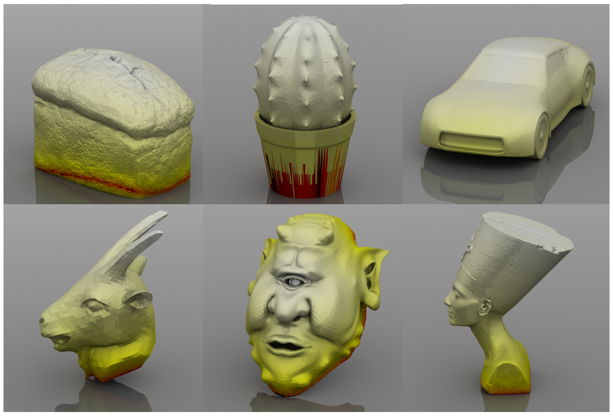


Fig. 22. Rest poses of meshes to be mapped to a planar domain. Red colors indicate where the boundary is. Note the Cactus model has a bad triangulation at the bottom that our method has no problem dealing with.

11.1.1 *As-Rigid-As-Possible (ARAP) [Sorkine and Alexa 2007].*

$$E_1 = \sum_j ((\sigma_1 - 1)^2 + (\sigma_2 - 1)^2)\mathbf{m}_j$$

Here and in our other energies, $\sigma_1 \geq \sigma_2 \geq 0$ are the singular values of the Jacobian.

11.1.2 *Area Preserving Energy.*

$$E_2 = \sum_j \mathbf{m}_j (\det \mathbf{J}_{(j)} - 1)^2 = \sum_j \mathbf{m}_j (\sigma_1 \sigma_2 - 1)^2$$

Blending of E_2 with the regular Dirichlet energy yields a compression-resistant Poisson membrane [Setaluri et al. 2015].

11.1.3 *Mass Spring Energy.*

$$E_3 = \sum_e (l_e - l_e^0)^2$$

The mass spring energy measures the change of lengths l_e for every edge e in the mesh, from its rest length l_e^0.

11.1.4 Conformal Energy. While we can simply set $E = E_{MIPS}$ to promote conformal maps, alternatively this can be achieved by using a regularization term that only depends on **A**. The following regularizer corresponds to the MIPS energy [Hormann and Greiner 2000] upon convergence:

$$R_1(\xi) = \sum_j \mathbf{m}_j \mathcal{R}(r(\xi_j)), \text{ where} \tag{35}$$

$$\mathcal{R}(\mu) := \left(\frac{1 - |\mu|}{1 + |\mu|} + \frac{1 + |\mu|}{1 - |\mu|} - 2 \right) = 4 \left(\frac{1}{1 - |\mu|^2} - 1 \right). \tag{36}$$

With the tanh variant that $|\mu| = \tanh |\xi|$, by applying the hyperbolic trigonometric identities we have that the $\mathcal{R}$ term becomes

$$\mathcal{R}(r(\xi)) = 4(\cosh^2 |\xi| - 1), \tag{37}$$

where cosh is the hyperbolic cosine function.

Results. Figures 21, 23 demonstrate the results of optimizing the energies above in our framework. In the experiment, the boundary is normalized to match the area of the input surface. We see the mass spring and ARAP usually produce similar results, while the area-preserving energy leads to patterns very different from other energies and/or quasi-conformal maps: The triangles are more evenly distributed, rather than clustered at the center.

Our framework conveniently minimizes many energies that are not supported by [Du et al. 2022; Garanzha et al. 2021]. While it seems plausible that in [Garanzha et al. 2021] setting their λ to be 1 should lead to purely area-preserving maps, we found doing so makes their code fail to converge; see Figure 24.

11.2 Application: Collision Avoidance with Air Meshes

Many practical applications can benefit from advances in bijective mapping under extreme distortion, such as collision avoidance. By choosing E as the elasticity of the shape with zero weighting at the "air mesh," we are able to compute globally bijective maps that avoid collision, following [Jiang et al. 2017; Müller et al. 2015].

As demonstrated in Figure 20, the "air mesh" marked in yellow can undergo extreme distortion: Only methods with discrete bijectivity can exactly prevent collision.

12 CONCLUSION AND FUTURE WORK

Contribution. In summary, we propose a framework to compute injective maps under positional constraints with entire boundary fixed. Our approach supports diverse objectives and can improve over state-of-the-art methods by orders of magnitude in terms of speed and map quality in challenging cases.

While existing methods are expensive to handle positional constraints, Tutte's embedding and its generalized family are extremely cheap to apply, with guaranteed bijections in many cases. Hence, our methods efficiently search in the space of all generalized Tutte's embeddings, yielding dramatic improvement over existing approaches.

The key to the success of our methods is to appropriately attribute local inversions to variables with global influence, by differentiating into the Dirichlet-to-Neumann operator and its inverse. By considering of a parametric family of Dirichlet energies, we successfully avoid the vanishing gradient and non-smooth issues of the area functional.

Limitations. Our approach—centered around a PDE-based characterization of planar map injectivity—is limited to 2D. While quasi-harmonicity with the Cauchy boundary condition is no longer sufficient to guarantee injectivity due to the many possibilities of saddle points in 3D, it is still a necessary condition, readily removing a major pattern of inversions. For future, we plan to generalize our approach to 3D by augmenting with an extra condition. We have to rely on nonconvex optimization routines; after all, the positional constraints might make the problem intrinsically nonconvex.

Future work. Our approach suggests many promising directions, as we have mentioned throughout the paper.

Moving forward, we plan to optimize for the boundary positions of the map together with the interior quasi-harmonicity, like the

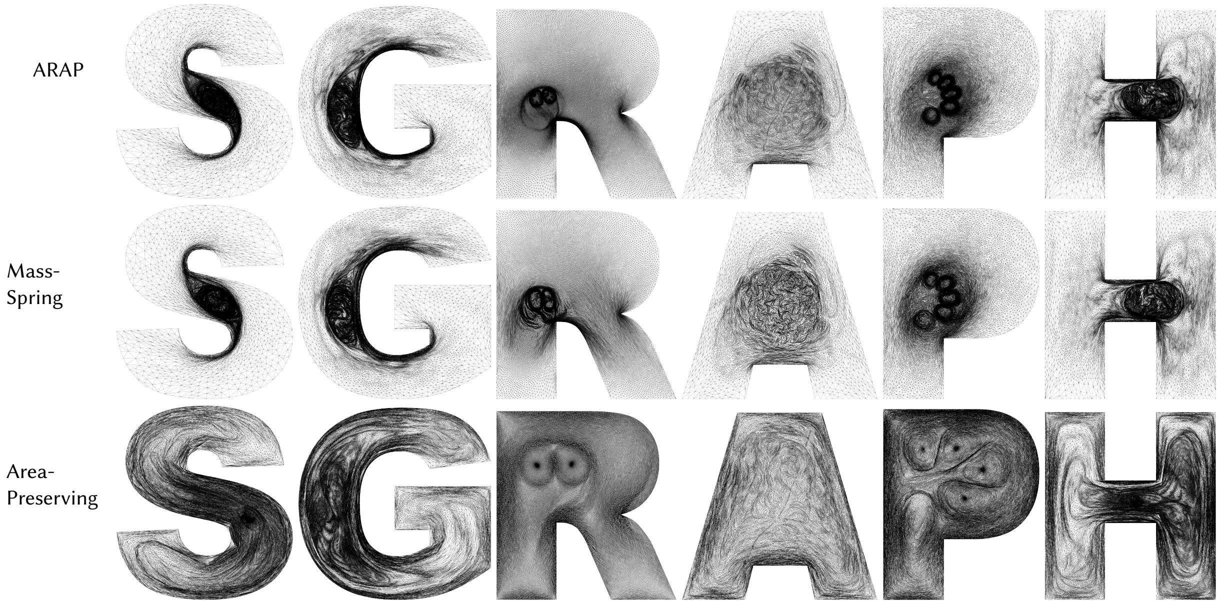


Fig. 23. Injective maps produced in our framework by quasi-harmonizing inversion-permitting energies augmented with the weighted Dirichlet.

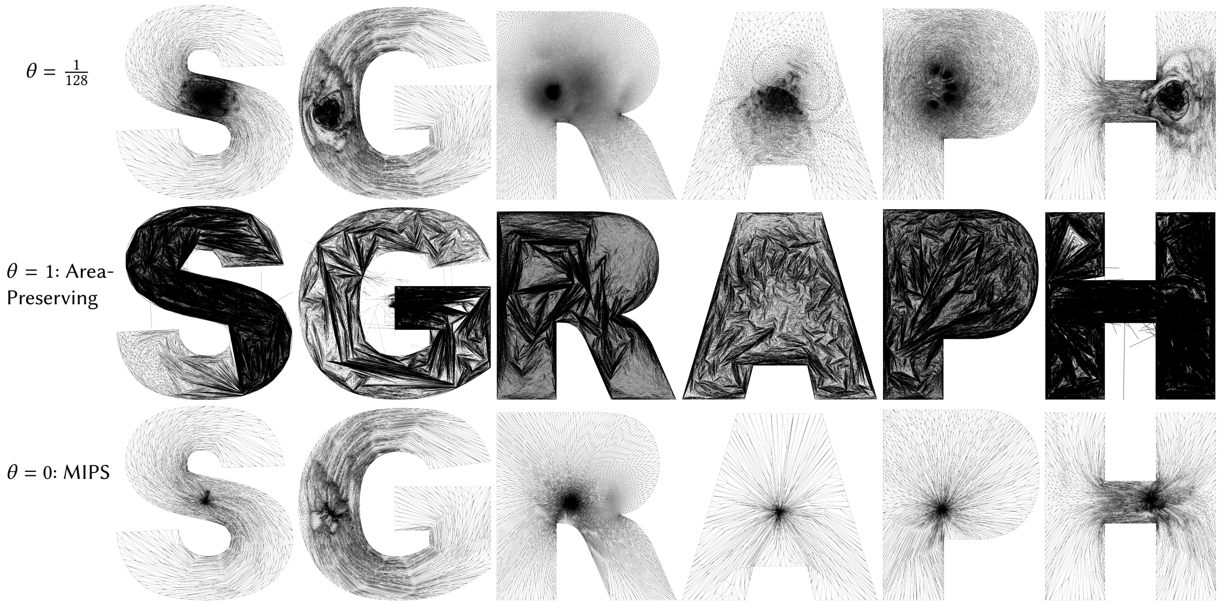


Fig. 24. It is not clear how to obtain area-preserving maps in SBP [Garanzha et al. 2021]. Only using area-preserving energy ($\theta = 1$) in their framework fails to produce injections except on the David-to-A example. Note the sharp triangles extruded from the mesh in SBP. Also, SBP has many sharp patterns, visually very different from our area-preserving functional. Without blending with their area-preserving term ($\theta = 0$), the inverse harmonic map produced by SBP can have extreme scaling and shearing at the boundary: It is not clear how SBP can obtain purely scale-invariant behaviors like harmonic/conformal maps.

free boundary setting [Du et al. 2021]. There are many works finding maps within harmonic subspaces: We can use theirs as fast initialization, and further enrich the map with quasi-harmonics.

Our continuous theorem without requiring $\det \mathbf{A}(\mathbf{x}) = 1$ is largely unexplored; we plan to fully unlock the capacity of our quasi-harmonic framework. The extra factor can add scaling control to quasi-conformal maps.

Incorporating ideas from computational mathematics and scientific computing, we might attempt to derive provable guarantees of our approach, including conditions when the program becomes convex and the optimality of the Sobolev-type norms.

Extending to other applications, by interpreting our $\mathbf{A}$ as stiffness or material property, we could possibly apply our solver to computational design and inverse mechanics.

ACKNOWLEDGMENTS

We thank Leticia Mattos Da Silva and Bohan Wang for proofreading, and David Palmer, Oded Stein, and Steven Gortler for illuminating discussions. Meshes are courtesy of Keenan Crane, Alec Jacobson, the Stanford 3D Scan Repository, and the AIM@Shape Repository. The MIT Geometric Data Processing group acknowledges the generous support of Army Research Office grants W911NF2010168 and W911NF2110293, of Air Force Office of Scientific Research award FA9550-19-1-031, of National Science Foundation grants IIS-1838071 and CHS-1955697, from the CSAIL Systems that Learn program, from the MIT–IBM Watson AI Laboratory, from the Toyota–CSAIL Joint Research Center, from a gift from Adobe Systems, and from a Google Research Scholar award.

REFERENCES

Robert Acar. 1993. Identification of the coefficient in elliptic equations. *SIAM Journal on Control and Optimization* 31, 5 (1993), 1221–1244.

Lars Valerian Ahlfors. 2006. *Lectures on quasiconformal mappings*. Vol. 38. American Mathematical Soc.

Noam Aigerman and Yaron Lipman. 2013. Injective and bounded distortion mappings in 3D. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–14.

Giovanni Alessandrini and Vincenzo Nesi. 2001. Univalent σ-harmonic mappings. *Archive for Rational Mechanics and Analysis* 158, 2 (2001), 155–171.

Giovanni Alessandrini and Vincenzo Nesi. 2021. Globally diffeomorphic σ-harmonic mappings. *Annali di Matematica Pura ed Applicata (1923-)* 200, 4 (2021), 1625–1635.

Kari Astala, Tadeusz Iwaniec, and Gaven Martin. 2008. Elliptic Partial Differential Equations and Quasiconformal Mappings in the Plane (PMS-48). In *Elliptic Partial Differential Equations and Quasiconformal Mappings in the Plane (PMS-48)*. Princeton University Press.

Mirela Ben-Chen, Craig Gotsman, and Guy Bunin. 2008. Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 449–458.

Mirela Ben-Chen, Ofir Weber, and Craig Gotsman. 2009. Variational harmonic maps for space deformation. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 34.

George E Brown and Rahul Narain. 2021. WRAPD: weighted rotation-aware ADMM for parameterization and deformation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–14.

Marcel Campen, Ryan Capouellez, Hanxiao Shen, Leyi Zhu, Daniele Panozzo, and Denis Zorin. 2021. Efficient and robust discrete conformal equivalence with boundary. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–16.

Renjie Chen and Ofir Weber. 2017. GPU-accelerated locally injective shape deformation. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–13.

Renjie Chen, Ofir Weber, Daniel Keren, and Mirela Ben-Chen. 2013. Planar shape interpolation with bounded distortion. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–12.

Edward Chien, Zohar Levi, and Ofir Weber. 2016. Bounded distortion parametrization in the space of metrics. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 1–16.

Gustave Choquet. 1945. Sur un type de transformation analytique généralisant la représentation conforme et définie au moyen de fonctions harmoniques. *Bull. Sci. Math.* 69, 2 (1945), 156–165.

Sebastian Claici, Mikhail Bessmeltsev, Scott Schaefer, and Justin Solomon. 2017. Isometry-Aware Preconditioning for Mesh Parameterization. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 37–47.

Tobias H Colding and William P Minicozzi. 2011. *A course in minimal surfaces*. Vol. 121. American Mathematical Soc.

Prabir Daripa. 1993. A fast algorithm to solve the Beltrami equation with applications to quasiconformal mappings. *J. Comput. Phys.* 106, 2 (1993), 355–365.

Timothy A Davis et al. 2015. SuiteSparse: A suite of sparse matrix software. *URL http://faculty. cse. tamu. edu/davis/suitesparse. html* (2015).

Fernando de Goes, Beibei Liu, Max Budninskiy, Yiying Tong, and Mathieu Desbrun. 2014. Discrete 2-tensor fields on triangulations. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 13–24.

Jesse Douglas. 1931. Solution of the problem of Plateau. *Trans. Amer. Math. Soc.* 33, 1 (1931), 263–321.

Jérôme Droniou and Christophe Le Potier. 2011. Construction and convergence study of schemes preserving the elliptic local maximum principle. *SIAM J. Numer. Anal.* 49, 2 (2011), 459–490.

Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z Kovalsky, Yajie Yan, Danny M Kaufman, and Tao Ju. 2020. Lifting simplices to find injectivity. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 120–1.

Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z Kovalsky, Yajie Yan, Danny M Kaufman, and Tao Ju. 2021. Optimizing Global Injectivity for Constrained Parameterization. *ACM Transactions on Graphics (TOG)* 40, 8 (2021), 1x.

Xingyi Du, Danny M Kaufman, Qingnan Zhou, Shahar Kovalsky, Yajie Yan, Noam Aigerman, and Tao Ju. 2022. Isometric Energies for Recovering Injectivity in Constrained Mapping. In *SIGGRAPH Asia 2022 Conference Papers*. 1–9.

Peter Duren. 2004. *Harmonic mappings in the plane*. Vol. 156. Cambridge University Press.

Lawrence C Evans. 1998. Partial differential equations. *Graduate studies in mathematics* 19, 4 (1998), 7.

Yu Fang, Minchen Li, Chenfanfu Jiang, and Danny M Kaufman. 2021. Guaranteed globally injective 3D deformation processing. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 10–1145.

Guy Fargion and Ofir Weber. 2022. Globally Injective Flattening via a Reduced Harmonic Subspace. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–17.

Michael Floater. 2003a. One-to-one piecewise linear mappings over triangulations. *Math. Comp.* 72, 242 (2003), 685–696.

Michael S Floater. 2003b. Mean value coordinates. *Computer Aided Geometric Design* 20, 1 (2003), 19–27.

Emil O Frind and George F Pinder. 1973. Galerkin solution of the inverse problem for aquifer transmissivity. *Water Resources Research* 9, 5 (1973), 1397–1410.

Xiao-Ming Fu, Yang Liu, and Baining Guo. 2015. Computing locally injective mappings by advanced MIPS. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Xiao-Ming Fu, Jian-Ping Su, Zheng-Yu Zhao, Qing Fang, Chunyang Ye, and Ligang Liu. 2021. Inversion-free geometric mapping construction: A survey. *Computational Visual Media* 7, 3 (2021), 289–318.

Ming Gao, Nathan Mitchell, and Eftychios Sifakis. 2014. Steklov-Poincaré skinning. In *Proc. Symposium on Computer Animation*. 139–148.

Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. 2021. Foldover-free maps in 50 lines of code. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.

Frederick P Gardiner and Nikola Lakic. 2000. *Quasiconformal Teichmuller theory*. Number 76. American Mathematical Soc.

David Gilbarg and Neil S Trudinger. 2015. *Elliptic partial differential equations of second order*. Springer.

Mark Gillespie, Boris Springborn, and Keenan Crane. 2021. Discrete conformal equivalence of polyhedral surfaces. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–20.

Steven J Gortler, Craig Gotsman, and Dylan Thurston. 2006. Discrete one-forms on meshes and applications to 3D mesh parameterization. *Computer Aided Geometric Design* 23, 2 (2006), 83–112.

Xianfeng David Gu, Feng Luo, Jian Sun, and Tianqi Wu. 2018. A discrete uniformization theorem for polyhedral surfaces. *Journal of Differential Geometry* 109, 2 (2018), 223–256.

Michael Hinze and Tran Nhan Tam Quyen. 2016. Matrix coefficient identification in an elliptic equation with the convex energy functional method. *Inverse Problems* 32, 8 (2016), 085007.

Kai Hormann and Giinther Greiner. 2000. MIPS: An Efficient Global Parametrization Method. *France on 1-7 July 1999. Proceedings, Volume 1. Curve and Surface Design. F61775-99-WF068* (2000), 153.

Victor Isakov. 2017. *Inverse Problems for Partial Differential Equations*. Vol. 127. Springer.

Kazufumi Ito and Karl Kunisch. 1990. The augmented Lagrangian method for parameter estimation in elliptic systems. *SIAM Journal on Control and Optimization* 28, 1 (1990), 113–136.

Tadeusz Iwaniec, Leonid V Kovalev, and Jani Onninen. 2012. Hopf differentials and smoothing Sobolev homeomorphisms. *International Mathematics Research Notices* 2012, 14 (2012), 3256–3277.

Zhongshi Jiang, Scott Schaefer, and Daniele Panozzo. 2017. Simplicial complex augmentation framework for bijective maps. *ACM Transactions on Graphics* 36, 6 (2017).

Andreas Kirsch. 2011. *An introduction to the mathematical theory of inverse problems*. Vol. 120. Springer Science & Business Media.

Hellmuth Kneser. 1926. Losung der aufgabe 41. *Jahresber. Deutsche Meth.* (1926), 123–124.

Shahar Z Kovalsky, Noam Aigerman, Ronen Basri, and Yaron Lipman. 2014. Controlling singular values with semidefinite programming. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 68–1.

Shahar Z Kovalsky, Noam Aigerman, Ingrid Daubechies, Michael Kazhdan, Jianfeng Lu, and Stefan Steinerberger. 2020. Non-convex planar harmonic maps. *arXiv preprint arXiv:2001.01322* (2020).

Shahar Z Kovalsky, Meirav Galun, and Yaron Lipman. 2016. Accelerated quadratic proxy for geometric optimization. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 134.

Zohar Levi and Ofir Weber. 2016. On the convexity and feasibility of the bounded distortion harmonic mapping problem. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–15.

Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. 2002. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 362–371.

Wentao Liao, Renjie Chen, Yuchen Hua, Ligang Liu, and Ofir Weber. 2021. Real-time locally injective volumetric deformation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16.

Selena Ling, Nicholas Sharp, and Alec Jacobson. 2022. VectorAdam for rotation equivariant geometry optimization. *Advances in Neural Information Processing Systems* (2022).

Yaron Lipman. 2012. Bounded distortion mapping spaces for triangular meshes. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 1–13.

Yaron Lipman. 2014. Bijective mappings of meshes with boundary and the degree in mesh processing. *SIAM Journal on Imaging Sciences* 7, 2 (2014), 1263–1283.

Yaron Lipman, Vladimir G Kim, and Thomas A Funkhouser. 2012. Simple formulas for quasiconformal plane deformations. *ACM Transactions on Graphics (TOG)* 31, 5 (2012), 1–13.

Richard Liska and Mikhail Shashkov. 2008. Enforcing the discrete maximum principle for linear finite element solutions of second-order elliptic problems. *Commun. Comput. Phys.* 3, 4 (2008), 852–877.

Ligang Liu, Chunyang Ye, Ruiqi Ni, and Xiao-Ming Fu. 2018. Progressive parameterizations. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Ligang Liu, Lei Zhang, Yin Xu, Craig Gotsman, and Steven J Gortler. 2008. A local/global approach to mesh parameterization. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1495–1504.

Tiantian Liu, Sofien Bouaziz, and Ladislav Kavan. 2017. Quasi-newton methods for real-time simulation of hyperelastic materials. *ACM Transactions on Graphics (TOG)* 36, 3 (2017), 1–16.

Tiantian Liu, Ming Gao, Lifeng Zhu, Eftychios Sifakis, and Ladislav Kavan. 2016. Fast and robust inversion-free shape manipulation. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 1–11.

Changna Lu, Weizhang Huang, and Jianxian Qiu. 2014. Maximum principle in linear finite element approximations of anisotropic diffusion–convection–reaction problems. *Numer. Math.* 127, 3 (2014), 515–537.

Lok Ming Lui, Ka Chun Lam, Shing-Tung Yau, and Xianfeng Gu. 2014. Teichmuller mapping (t-map) and its applications to landmark matching registration. *SIAM Journal on Imaging Sciences* 7, 1 (2014), 391–426.

Lok Ming Lui and Tsz Ching Ng. 2015. A splitting method for diffeomorphism optimization problem using Beltrami coefficients. *Journal of Scientific Computing* 63, 2 (2015), 573–611.

Yanwen Luo. 2022. Spaces of geodesic triangulations of surfaces. *Discrete & Computational Geometry* (2022), 1–19.

Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. 2008. Spectral conformal parameterization. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1487–1494.

Matthias Müller, Nuttapong Chentanez, Tae-Yong Kim, and Miles Macklin. 2015. Air meshes for robust collision handling. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable locally injective mappings. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1.

T. Rado. 1926. Aufgave 41. *Jber. Deutsch. Math.* 35 (1926).

Gerard R Richter. 1981. An inverse problem for the steady state diffusion equation. *SIAM J. Appl. Math.* 41, 2 (1981), 210–221.

Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Transactions on Graphics (TOG)* 37, 1 (2017), 5.

John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. 2004. Inter-surface mapping. In *ACM SIGGRAPH 2004 Papers*. 870–877.

Christian Schüller, Ladislav Kavan, Daniele Panozzo, and Olga Sorkine-Hornung. 2013. Locally injective mappings. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 125–135.

Rajsekhar Setaluri, Yu Wang, Nathan Mitchell, Ladislav Kavan, and Eftychios Sifakis. 2015. Fast grid-based nonlinear elasticity for 2D deformations. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 67–76.

Alla Sheffer and Eric de Sturler. 2001. Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with Computers* 17, 3 (2001), 326–337.

Alla Sheffer, Bruno Lévy, Maxim Mogilnitsky, and Alexander Bogomyakov. 2005. ABF++: fast and robust angle based flattening. *ACM Transactions on Graphics (TOG)* 24, 2 (2005), 311–330.

Hanxiao Shen, Zhongshi Jiang, Denis Zorin, and Daniele Panozzo. 2019. Progressive embedding. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 32.

Anna Shtengel, Roi Poranne, Olga Sorkine-Hornung, Shahar Z Kovalsky, and Yaron Lipman. 2017. Geometric optimization via composite majorization. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–11.

Jason Smith and Scott Schaefer. 2015. Bijective parameterization with free boundaries. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–9.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Proc. SGP*, Vol. 4.

Boris Springborn, Peter Schröder, and Ulrich Pinkall. 2008. Conformal equivalence of triangle meshes. In *ACM Transactions on Graphics (TOG)*. 1–11.

Oded Stein, Jiajin Li, and Justin Solomon. 2022. A splitting scheme for flip-free distortion energies. *SIAM Journal on Imaging Sciences* 15, 2 (2022), 925–959.

Michael E Taylor. 2010. Partial differential equations I. Basic theory. *Applied Mathematical Sciences* 115 (2010).

Fredi Tröltzsch. 2010. *Optimal control of partial differential equations: theory, methods, and applications*. Vol. 112. American Mathematical Soc.

William Thomas Tutte. 1963. How to draw a graph. *Proceedings of the London Mathematical Society* 3, 1 (1963), 743–767.

Yu Wang, Mirela Ben-Chen, Iosif Polterovich, and Justin Solomon. 2018. Steklov spectral geometry for extrinsic shape analysis. *ACM Transactions on Graphics (TOG)* 37, 7 (2018), 21.

Yu Wang and Justin Solomon. 2019. Intrinsic and extrinsic operators for shape analysis. In *Handbook of Numerical Analysis*. Vol. 20. Elsevier, 41–115.

Yu Wang and Justin Solomon. 2021. Fast quasi-harmonic weights for geometric data interpolation. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–15.

Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. 2007. Discrete Laplace operators: no free lunch. In *Symposium on Geometry Processing*. 33–37.

Ofir Weber, Ashish Myles, and Denis Zorin. 2012. Computing extremal quasiconformal maps. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 1679–1689.

Ofir Weber and Denis Zorin. 2014. Locally injective parametrization with arbitrary fixed boundaries. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 75.

Stephen Wright, Jorge Nocedal, et al. 1999. Numerical optimization. *Springer Science* 35, 67-68 (1999), 7.

Yin Xu, Renjie Chen, Craig Gotsman, and Ligang Liu. 2011. Embedding a triangular graph within a given boundary. *Computer Aided Geometric Design* 28, 6 (2011), 349–356.

Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. 2004. A fast and simple stretch-minimizing mesh parameterization. In *Proceedings Shape Modeling Applications, 2004*. IEEE, 200–208.

Rhaleb Zayer, Christian Rossl, and H-P Seidel. 2005. Discrete tensorial quasi-harmonic maps. In *International Conference on Shape Modeling and Applications 2005 (SMI'05)*. IEEE, 276–285.

Yufeng Zhu, Robert Bridson, and Danny M Kaufman. 2018. Blended cured quasi-newton for distortion optimization. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14.

A GENERALIZED LSCM

A.1 Discrete Area Form

Here we provide more details on Definition 15. The circular gradient is counterclockwise on the outer boundary, and clockwise on the inner boundary (if there is any). For $i \longrightarrow j$ being a boundary edge, we mean that $i \longrightarrow j$ follows right-hand rule in the (only) adjacent triangle: If boundary edge $i \longrightarrow j$ is on the outline (outer boundary) of the domain, it has to be counter-clockwise along the boundary; otherwise $i \longrightarrow j$ must be in a hole (inner boundary) of the domain and clockwise.

Note $\mathbf{u}^\top \mathbf{D}\mathbf{v}$ is also the *signed area* of the domain due to Green's Theorem, so is $\frac{1}{2}(\mathbf{u}+\mathrm{i}\mathbf{v})^{\mathrm{H}}(-\mathrm{i}\mathbf{D})(\mathbf{u}+\mathrm{i}\mathbf{v}) = \mathbf{u}^\top \mathbf{D}\mathbf{v}$, where $(\cdot)^{\mathrm{H}}$ denotes the Hermitian conjugate. The matrix $\mathbf{D}$ can also be equivalently written as:

$$\mathbf{D} = \frac{1}{2}(\mathbf{D}_h - \mathbf{D}_h^\top),$$

where

$$\mathbf{D}_h := \sum_{j \in \mathcal{T}} \mathbf{1}_{\{F_{j,1},F_{j,2}\}} + \mathbf{1}_{\{F_{j,2},F_{j,3}\}} + \mathbf{1}_{\{F_{j,3},F_{j,1}\}},$$

in which $\mathbf{1}_{\{i,j\}} \in \mathbb{R}^{n\times n}$ denotes the matrix that is 1 at the i, j-th entry and zero elsewhere, and $F_{j,1}, F_{j,2}, F_{j,3}$ are the indices of vertices in the j-th triangle, in counter-clockwise order. To see where the formula comes from, recall the area of the triangle $(x_1, y_1) \longrightarrow$

$(x_2, y_2) \longrightarrow (x_3, y_3)$ is:

$$\frac{1}{2}\det\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}^\top \begin{bmatrix} & +1 & -1 \\ -1 & & +1 \\ +1 & -1 & \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

The equivalence of two ways writing down $\mathbf{D}$ can be seen by cancelling out interior "half-edges" (directed edges). In the continuous case, the equivalence is due to the Green's Theorem.

A.2 Positive-definiteness of Generalized LSCM

The weighted Dirichlet subtracted from the complex-valued area form, $\mathbf{G}^\top\mathbf{A}\mathbf{G} + \mathrm{i}\mathbf{D}$, is still positive semi-definite:

Proposition A.1. *If* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) \geq 1$ *and* $\mathbf{m}_j^{-1}\mathbf{A}_{(j)} \geq 0$, *we have* $\mathbf{G}^\top\mathbf{A}\mathbf{G} + i\mathbf{D} \geq 0$.

This has been well-known in the continuous case [Astala et al. 2008]. It generalizes LSCM [Lévy et al. 2002; Mullen et al. 2008] to $\mathbf{m}_j^{-1}\mathbf{A}_{(j)} \neq \mathbf{I}$.

Proof. The signed area of j-th triangle is

$$\begin{aligned}\mathbf{m}_j\mathbf{G}_{(j)}\mathbf{u} \times \mathbf{G}_{(j)}\mathbf{v} =& \mathbf{u}^\top\mathbf{G}_{(j)}^\top\mathbf{m}_j\begin{bmatrix} & -1 \\ 1 & \end{bmatrix}\mathbf{G}_{(j)}\mathbf{v} \\ =& \frac{1}{2}(\mathbf{u}+\mathrm{i}\mathbf{v})^\mathsf{H}\mathbf{G}_{(j)}^\top\mathbf{m}_j\begin{bmatrix} & \mathrm{i} \\ -\mathrm{i} & \end{bmatrix}\mathbf{G}_{(j)}(\mathbf{u}+\mathrm{i}\mathbf{v})\end{aligned}$$

and the j-th triangle's contribution to the Dirichlet energy is

$$\begin{aligned}&\frac{1}{2}\mathbf{u}^\top\left(\mathbf{G}_{(j)}^\top\mathbf{A}_{(j)}\mathbf{G}_{(j)}\right)\mathbf{u} + \frac{1}{2}\mathbf{v}^\top\left(\mathbf{G}_{(j)}^\top\mathbf{A}_{(j)}\mathbf{G}_{(j)}\right)\mathbf{v} \\ =& \frac{1}{2}(\mathbf{u}+\mathrm{i}\mathbf{v})^\mathsf{H}\left(\mathbf{G}_{(j)}^\top\mathbf{A}_{(j)}\mathbf{G}_{(j)}\right)(\mathbf{u}+\mathrm{i}\mathbf{v}).\end{aligned}$$

It suffices to show that

$$\mathbf{A}_{(j)} - \mathbf{m}_j\begin{bmatrix} & \mathrm{i} \\ -\mathrm{i} & \end{bmatrix} = \mathbf{m}_j\begin{bmatrix} (\mathbf{a}_{00})_j & (\mathbf{a}_{01})_j - \mathrm{i} \\ (\mathbf{a}_{01})_j + \mathrm{i} & (\mathbf{a}_{11})_j \end{bmatrix} \geq 0,$$

which can be verified by checking the determinant

$$\det\begin{bmatrix} (\mathbf{a}_{00})_j & (\mathbf{a}_{01})_j - \mathrm{i} \\ (\mathbf{a}_{01})_j + \mathrm{i} & (\mathbf{a}_{11})_j \end{bmatrix} = \det(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}) - 1 \geq 0.$$

□

B BIJECTION ⇒ {QUASI-HARMONIC} + {CAUCHY BC}

Now we will argue the "necessary" direction of Theorem 6.4, certifying that our framework of quasi-harmonic maps does not limit the modeling capacity at the discrete level.

B.1 Review: Linear Precision of Cotangent Laplacian

The linear reproducing property of the planar discrete Laplacian is well known. See, e.g., [Wardetzky et al. 2007] and references therein.

Proposition B.1 (Laplacian: Linear Reproducing Property). *Consider a* planar *triangle mesh with the standard cotangent Laplacian* $\mathbf{L} = \mathbf{G}^\top\mathbf{M}\mathbf{G}$, *where the diagonal matrix* $\mathbf{M} := \mathit{diag}([\mathbf{m}; \mathbf{m}])$ *repeats the triangle areas twice. Denote the* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n\times 1}$ *as the* x, y *coordinates of the vertices. We have the linear reproducing property:*

$$\begin{aligned}\mathbf{S}^\top\mathbf{L}[\mathbf{x} \quad \mathbf{y}] &= [0 \quad 0] \\ \mathbf{R}^\top\mathbf{L}[\mathbf{x} \quad \mathbf{y}] &= [\mathbf{C}\mathbf{b}_v \quad -\mathbf{C}\mathbf{b}_u].\end{aligned} \tag{38}$$

Proposition B.1 is the origin of the idea of our proof: the rest pose of a planar mesh is in the kernel of its Laplacian. It is also where the Neumann boundary condition comes from: $\mathbf{g}_u = +\mathbf{C}\mathbf{b}_v, \mathbf{g}_v = -\mathbf{C}\mathbf{b}_u$.

B.2 Discrete Completeness

The following shows that our model is capable to reproduce any discrete map containing no foldover by pulling back and reconstructing the cotangent Laplacian associated to the planar target mesh, whose kernel admits the map $(\mathbf{u}, \mathbf{v})$.

Proposition B.2 (Realizability). *Consider a target pose of the same triangle mesh with the standard cotangent Laplacian* $\hat{\mathbf{L}} = \hat{\mathbf{G}}\hat{\mathbf{M}}\hat{\mathbf{G}}$ *where* $\hat{\mathbf{M}} := \mathit{diag}([\hat{\mathbf{m}}; \hat{\mathbf{m}}])$. *The anisotropic Laplacian operator* $\mathbf{G}^\top\mathbf{A}\mathbf{G}$ *on the source domain, under the constraint* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1, \forall j$, *is capable to reproduce any weak form Laplacian matrix* $\hat{\mathbf{L}}$, *as long as* $\hat{\mathbf{m}} > 0$; *namely, there exists a block-diagonal* $\hat{\mathbf{A}} \geq 0$ *such that* $\hat{\mathbf{L}} = \mathbf{G}^\top\hat{\mathbf{A}}\mathbf{G}$.

Proof. For each triangle j, denote the gradients of hat functions as $\nabla\mathbf{h}_{(j,1)}, \nabla\mathbf{h}_{(j,2)}, \nabla\mathbf{h}_{(j,3)} \in \mathbb{R}^2$, respectively; similarly, denote the gradients of hat functions for the deformed target triangle j as $\nabla\hat{\mathbf{h}}_{(j,1)}, \nabla\hat{\mathbf{h}}_{(j,2)}, \nabla\hat{\mathbf{h}}_{(j,3)} \in \mathbb{R}^2$, respectively.

It suffices to show that for each triangle j there exists $\hat{\mathbf{A}}_{(j)} \in \mathbb{R}^{2\times 2}$ such that

$$\begin{bmatrix} \nabla\mathbf{h}_{(j,1)}^\top \\ \nabla\mathbf{h}_{(j,2)}^\top \\ \nabla\mathbf{h}_{(j,3)}^\top \end{bmatrix}\hat{\mathbf{A}}_{(j)}\begin{bmatrix} \nabla\mathbf{h}_{(j,1)}^\top \\ \nabla\mathbf{h}_{(j,2)}^\top \\ \nabla\mathbf{h}_{(j,3)}^\top \end{bmatrix}^\top = \begin{bmatrix} \nabla\hat{\mathbf{h}}_{(j,1)}^\top \\ \nabla\hat{\mathbf{h}}_{(j,2)}^\top \\ \nabla\hat{\mathbf{h}}_{(j,3)}^\top \end{bmatrix}\hat{\mathbf{m}}_j\begin{bmatrix} \nabla\hat{\mathbf{h}}_{(j,1)}^\top \\ \nabla\hat{\mathbf{h}}_{(j,2)}^\top \\ \nabla\hat{\mathbf{h}}_{(j,3)}^\top \end{bmatrix}^\top.$$

Let

$$\hat{\mathbf{K}}_{(j)} = \hat{\mathbf{m}}_j^{1/2}[\nabla\hat{\mathbf{h}}_{(j,1)}\nabla\hat{\mathbf{h}}_{(j,2)}][\nabla\mathbf{h}_{(j,1)}\nabla\mathbf{h}_{(j,2)}]^{-1}.$$

By noticing $\nabla\mathbf{h}_{(j,1)} + \nabla\mathbf{h}_{(j,2)} + \nabla\mathbf{h}_{(j,3)} = \mathbf{0}$, as well as $\nabla\hat{\mathbf{h}}_{(j,1)} + \nabla\hat{\mathbf{h}}_{(j,2)} + \nabla\hat{\mathbf{h}}_{(j,3)} = \mathbf{0}$, by linearity we have

$$\hat{\mathbf{K}}_{(j)}[\nabla\mathbf{h}_{(j,1)}\nabla\mathbf{h}_{(j,2)}\nabla\mathbf{h}_{(j,3)}] = \hat{\mathbf{m}}_j^{1/2}[\nabla\hat{\mathbf{h}}_{(j,1)}\nabla\hat{\mathbf{h}}_{(j,2)}\nabla\hat{\mathbf{h}}_{(j,3)}].$$

So we have found such $\hat{\mathbf{A}}_{(j)} = \hat{\mathbf{K}}_{(j)}^\top\hat{\mathbf{K}}_{(j)} \geq 0$.

To derive $\hat{\mathbf{A}}_{(j)}$ which is unique, we observe the following: in triangle j, edge vectors $\mathbf{e}_{(j,1)}, \mathbf{e}_{(j,2)}, \mathbf{e}_{(j,3)}$ are transformed by $\mathbf{J}_{(j)}$, so their orthogonal vectors $\nabla\mathbf{h}_{(j,1)}, \nabla\mathbf{h}_{(j,2)}, \nabla\mathbf{h}_{(j,3)}$ with the reciprocal magnitude are transformed to $\nabla\hat{\mathbf{h}}_{(j,1)}\nabla\hat{\mathbf{h}}_{(j,2)}\nabla\hat{\mathbf{h}}_{(j,3)}$ by $\mathbf{J}_{(j)}^{-\top}$. So we have found the formula for $\hat{\mathbf{A}}_{(j)}$:

$$\mathbf{m}_j^{-1}\hat{\mathbf{A}}_{(j)} = \det(\mathbf{J}_{(j)})[\mathbf{J}_{(j)}^\top\mathbf{J}_{(j)}]^{-1}. \tag{39}$$

It is straightforward to verify

$$\det\left(\mathbf{m}_j^{-1}\hat{\mathbf{A}}_{(j)}\right) = 1.$$

Note we require $\hat{\mathbf{m}}_j \geq 0$ to construct our $\hat{\mathbf{A}}$. □

Finally, by Proposition B.1 the map $(\mathbf{u}, \mathbf{v})$ can be recovered from the kernel of $\hat{\mathbf{L}}$: Namely, it holds that

$$\begin{aligned}\mathbf{S}^\top\mathbf{G}^\top\hat{\mathbf{A}}\mathbf{G}[\mathbf{u} \quad \mathbf{v}] &= [0 \quad 0] \\ \mathbf{R}^\top\mathbf{G}^\top\hat{\mathbf{A}}\mathbf{G}[\mathbf{u} \quad \mathbf{v}] &= [\mathbf{g}_u \quad \mathbf{g}_v] \equiv [\mathbf{C}\mathbf{b}_v, -\mathbf{C}\mathbf{b}_u]\end{aligned}$$

So we have the "only-if" part of Theorem 6.4 holds.

Remark (Modeling Capacity). *Proposition B.1 also holds for planar triangle mesh with flips, if flipped triangles use negative areas. Our choice requiring $\mathbf{A}_{(j)} \succ 0$ desirably limits our model's ability to reproduce maps with inversions or zero-area triangles.*

C BIJECTION $\Leftarrow$ {QUASI-HARMONIC} + {CAUCHY BC}

Let us prove the "sufficient" direction of Theorem 6.4. We will show the Cauchy boundary condition assures the removal of flips that quasi-harmonics may still have.

Our proof is based on two facts: 1) weighted Dirichlet and total signed area tightly "sandwich" total unsigned area, and 2) the map is free from flips if the total unsigned area equals the signed area.

C.1 Weighted Dirichlet Upper Bounds Area

Definition C.1. The total signed area is

$$\begin{aligned}\mathcal{A}(\mathbf{u},\mathbf{v}) &:= \sum_{j\in\mathcal{T}} \mathbf{m}_j \mathbf{G}_{(j)}\mathbf{u} \times \mathbf{G}_{(j)}\mathbf{v} \\ &= \frac{1}{2}(\mathbf{u}+\mathrm{i}\mathbf{v})^{\mathsf{H}}(-\mathrm{i}\mathbf{D})(\mathbf{u}+\mathrm{i}\mathbf{v}) = \mathbf{u}^\top\mathbf{D}\mathbf{v} = \mathbf{b}_u^\top \mathbf{C}\mathbf{b}_v.\end{aligned} \tag{40}$$

The last equality has been discussed in §A.1.

Definition C.2. The total unsigned area is

$$\mathcal{U}(\mathbf{u},\mathbf{v}) := \sum_{j\in\mathcal{T}} \mathbf{m}_j \|\mathbf{G}_{(j)}\mathbf{u} \times \mathbf{G}_{(j)}\mathbf{v}\|. \tag{41}$$

Proposition C.3. *Provided* $\mathbf{A} \succeq 0$ *such that* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) \geq 1, \forall j$, *we have*

$$\frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} + \frac{1}{2}\mathbf{v}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v} \geq \mathcal{U}(\mathbf{u},\mathbf{v}) \geq \mathcal{A}(\mathbf{u},\mathbf{v}). \tag{42}$$

Proof. We have the Jacobian for the j-th triangle denoted as:

$$\mathbf{J}_{(j)} = [\mathbf{G}_{(j)}\mathbf{u} \quad \mathbf{G}_{(j)}\mathbf{v}]^\top \in \mathbb{R}^{2\times 2}.$$

For any 2×2 matrix, $\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22}\end{bmatrix}$, we have

$$\det\mathbf{B} \leq |b_{11}b_{22} - b_{12}b_{21}| \leq \frac{1}{2}(b_{11}^2 + b_{12}^2 + b_{21}^2 + b_{22}^2) = \frac{1}{2}\|\mathbf{B}\|_F^2.$$

Now we are ready to show that the $\mathbf{A}$-weighted Dirichlet energy upper bounds the total unsigned area:

$$\begin{aligned}&\frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} + \frac{1}{2}\mathbf{v}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v} = \sum_{j\in\mathcal{T}} \frac{1}{2}\mathrm{TR}(\mathbf{J}_{(j)}\mathbf{A}_{(j)}\mathbf{J}_{(j)}^\top) \\ &= \sum_{j\in\mathcal{T}} \frac{1}{2}\mathbf{m}_j \|\sqrt{\mathbf{m}_j^{-1}\mathbf{A}_{(j)}}\mathbf{J}_{(j)}^\top\|_F^2 \geq \sum_{j\in\mathcal{T}} \mathbf{m}_j \sqrt{\det \mathbf{m}_j^{-1}\mathbf{A}_{(j)}}\,|\det\mathbf{J}_{(j)}| \\ &\geq \sum_{j\in\mathcal{T}} \mathbf{m}_j |\det\mathbf{J}_{(j)}| \equiv \mathcal{U}(\mathbf{u},\mathbf{v}) \geq \sum_{j\in\mathcal{T}} \mathbf{m}_j \det\mathbf{J}_{(j)} \equiv \mathcal{A}(\mathbf{u},\mathbf{v}).\end{aligned}$$

□

C.2 A Simple Characterization of Bijections

Under the same setting of fixed boundary in Theorem 6.4, we have

Proposition C.4. *The map* $(\mathbf{u},\mathbf{v})$ *is (locally) injective if and only if the total unsigned area equals to the total signed area, and that no triangle degenerates.*

The theorem is essentially a result of Xu et al. [2011]. The map becomes homeomorphic/bijective when the conditions in Lipman [2014] are met, including the boundary is free from self-intersection.

C.3 Completing the Proof of Discrete Theorem 6.4

Now we complete our proof of "sufficient" direction of Theorem 6.4, by showing that the boundary condition (17) magically enforces weighted Dirichlet arrives at its global minimum.

Proof. Provided $\mathbf{A} \succ 0$ such that $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) \geq 1, \forall j$, we have (42) hold. When the Cauchy boundary condition—all of the equations in (17)—holds, the Dirichlet energy reaches its lower bound—total signed area—by checking:

$$\mathbf{u}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} = \mathbf{u}^\top\mathbf{S}\cancelto{0}{[\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u}]} + \underbrace{\mathbf{u}^\top\mathbf{R}}_{\mathbf{b}_u^\top}\underbrace{\mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u}}_{\mathbf{g}_u} = \mathbf{b}_u^\top\mathbf{g}_u$$

$$\mathbf{v}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v} = \mathbf{v}^\top\mathbf{S}\cancelto{0}{[\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v}]} + \underbrace{\mathbf{v}^\top\mathbf{R}}_{\mathbf{b}_v^\top}\underbrace{\mathbf{R}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v}}_{\mathbf{g}_v} = \mathbf{b}_v^\top\mathbf{g}_v$$

$$\frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} + \frac{1}{2}\mathbf{v}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v} = \frac{1}{2}(\mathbf{b}_u^\top\mathbf{g}_u + \mathbf{b}_v^\top\mathbf{g}_v) \equiv \mathcal{A}(\mathbf{u},\mathbf{v}).$$

The weighted Dirichlet energy arriving at the global minimum forces the "sandwiched" total unsigned area equal to the minimum as well. By constraining $\mathbf{A}_{(j)} \succ 0$ no triangle degenerates. By Proposition C.4, we can conclude that the system (17) ensures that no triangle has negative area. □

D A CHARACTERIZATION OF DIFFEOMORPHISMS

We expand the discussion on Theorem 5.2. A detailed proof is beyond the scope of this paper, and we only sketch the argument.

The special case $\det\mathbf{A}(\mathbf{x}) = 1$ follows a similar argument of the discrete case using Dirichlet energies as a tight upper bound.

For the generic case that $\det\mathbf{A}(\mathbf{x}) = 1$ may not hold, we have the very recent result due to [Alessandrini and Nesi 2021].

Theorem D.1. *[Alessandrini and Nesi 2021, Theorem* 1.1*] Assume the source domain* $\Omega = B$ *is a unit ball in* $\mathbb{R}^2$, *and consider a planar domain* $\Gamma \subset \mathbb{R}^2$ *whose boundary* $\partial\Gamma$ *is a simple closed curve. Assume* $\phi = (u,v) : \Omega \to \Gamma$ *diffeomorphically maps* $\partial\Omega$ *onto* $\partial\Gamma$. *Given that the entries of* $\mathbf{A}(\mathbf{x})$ *are sufficiently smooth and* $\frac{1}{K}\mathbf{I} \preceq \mathbf{A}(\mathbf{x}) \preceq K\mathbf{I}$ *for some* $K > 0$, *the generated map* (u,v) *is a diffeomorphism of* $\bar{B}$ *onto* $\bar{\Gamma}$ *if and only if* $\det[\nabla u \; \nabla v] > 0$ *everywhere on* ∂B.

We notice the boundary condition can be substituted:

Proposition D.2. *Enforcing the Neumann boundary condition*

$$\mathbf{n}(\mathbf{x})^\top[\mathbf{A}(\mathbf{x})\nabla u(\mathbf{x}),\ \mathbf{A}(\mathbf{x})\nabla v(\mathbf{x})] = s(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x})^\top$$

where $s : \partial\Omega \to \mathbb{R}_+$, $s(\partial\Omega) > S$ *for some* $S > 0$, *is sufficient to guarantee* $\det[\nabla u(\mathbf{x}) \; \nabla v(\mathbf{x})] > 0$.

Proof. Denote $\mathbf{n}, \mathbf{t}$ as the normal, and tangent direction (by counterclockwise $\pi/2$-rotation of the normal) at $\mathbf{x}$. And $\hat{\mathbf{n}}, \hat{\mathbf{t}}$ are the normal and tangent at $u(\mathbf{x}), v(\mathbf{x})$.

$$\begin{bmatrix} \frac{\partial u}{\partial \mathbf{n}} & \frac{\partial u}{\partial \mathbf{t}} \\ \frac{\partial v}{\partial \mathbf{n}} & \frac{\partial v}{\partial \mathbf{t}} \end{bmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} [\mathbf{n} \quad \mathbf{t}] = \mathbf{J}[\mathbf{n} \quad \mathbf{t}]$$

The tangents $\mathbf{t}, \hat{\mathbf{t}}$ are related by the Jacobian $\mathbf{J}$: $\mathbf{J}\mathbf{t} = c\hat{\mathbf{t}}$ for $c > 0$. The Neumann condition

$$\mathbf{n}^\top [\mathbf{A}\nabla u\ \mathbf{A}\nabla v] = \mathbf{n}^\top \mathbf{A}\mathbf{J}^\top = f\hat{\mathbf{n}}^\top$$

can be written as

$$\mathbf{n}^\top [\mathbf{n} \quad \mathbf{t}]\tilde{\mathbf{A}}[\mathbf{n} \quad \mathbf{t}]^\top \mathbf{J}^\top = f\hat{\mathbf{n}}^\top,$$

where

$$\tilde{\mathbf{A}} := [\mathbf{n} \quad \mathbf{t}]^\top \mathbf{A}[\mathbf{n} \quad \mathbf{t}] \geq \frac{1}{k}\mathbf{I}.$$

So, we have

$$[1 \quad 0]\tilde{\mathbf{A}}[\mathbf{J}\mathbf{n} \quad c\hat{\mathbf{t}}]^\top = f\hat{\mathbf{n}}^\top.$$

Further multiplying both sides with $\hat{\mathbf{n}}$ from the right,

$$[1 \quad 0]\tilde{\mathbf{A}} \begin{bmatrix} (\mathbf{J}\mathbf{n})^\top \hat{\mathbf{n}} \\ 0 \end{bmatrix} = s(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x})^\top \hat{\mathbf{n}}(\mathbf{x}) = s(\mathbf{x}) > S > 0.$$

Since $\tilde{\mathbf{A}}_{11} > 0$, we have $(\mathbf{J}\mathbf{n})^\top \hat{\mathbf{n}} > 0$, and thus we conclude

$$\det \mathbf{J} = \det([\mathbf{J}\mathbf{n} \quad c\hat{\mathbf{t}}]) > 0.$$

□

Remark. *The condition on the source domain $\Omega = B$ to be an unit disk is not essential and can be relaxed. Uniformization allows conformally pulling back the PDE on manifold onto the unit disk B.*

E DETAILS ON THE POISSON FUNCTIONAL

In this section, we prove that minimizing the Poisson functional (§7.3.2) leads to an injective map (assume it exists). Lacking an intuitive interpretation, we conjecture that our Poisson functional measures a total (weighted) Gaussian curvature for developability.

E.1 Gradient Formula

We first derive the gradient formula for the Poisson functional (§7.3.2):

$$\frac{1}{2}\mathbf{g}_u^\top \mathbf{R}^\top (\mathbf{G}^\top \mathbf{A}\mathbf{G})^\dagger \mathbf{R}\mathbf{g}_u - \frac{1}{2}\mathbf{g}_u^\top \mathbf{b}_u.$$

Denote

$$\mathbf{p} = -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{R}\mathbf{g}_u$$

such that

$$\frac{\partial \mathbf{p}}{\partial \mathbf{a}} = -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{G}^\top \textsc{sp}(\mathbf{G}\mathbf{p}) \in \mathbb{R}^{n\times 3f} \tag{43}$$

in which $\frac{\partial \mathbf{p}}{\partial \mathbf{a}}$ can derived by

$$\begin{aligned} \textsc{const} =& [\mathbf{G}^\top \mathbf{A}\mathbf{G}]\mathbf{p} \\ 0 =& [\mathbf{G}^\top \mathbf{A}\mathbf{G}]\delta\mathbf{p} + [\mathbf{G}^\top \delta\mathbf{A}\mathbf{G}]\mathbf{p} \\ \delta\mathbf{p} =& -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{G}^\top \delta\mathbf{A}\mathbf{G}\mathbf{p} \\ \delta\mathbf{p} =& -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{G}^\top \textsc{sp}(\mathbf{G}\mathbf{p})\textsc{flat}(\delta\mathbf{A}). \end{aligned}$$

We have the gradient formula

$$\begin{aligned} \frac{\mathrm{d}V_4^u}{\mathrm{d}\mathbf{a}} =& \frac{1}{2}\left[\frac{\partial \mathbf{p}}{\partial \mathbf{a}}\right]^\top \mathbf{R}\mathbf{g}_u = -\frac{1}{2}\textsc{sp}(\mathbf{G}\mathbf{p})^\top \mathbf{G}[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{R}\mathbf{g}_u \\ =& -\frac{1}{2}\textsc{sp}(\mathbf{G}\mathbf{p})^\top \mathbf{G}\mathbf{p}. \end{aligned} \tag{44}$$

E.2 Minimizers of Poisson Functional are Injective Maps

Before the proof, we show a simple fact about $\mathbf{SL}(2, \mathbb{R}) \cap \mathbf{S}^2_+$, namely: For

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \geq 0, \quad \det \mathbf{A} = a_{11}a_{22} - a_{12}^2 \equiv 1,$$

we have $0 = \delta \det \mathbf{A} = \delta a_{11}a_{22} + a_{11}\delta a_{22} - 2a_{12}\delta a_{12}$, which implies, $\delta\mathbf{A}$, a local infinitesimal derivation from $\mathbf{A}$, must be orthogonal to $[*\mathbf{A}*^\top]$:

$$0 = \delta \det \mathbf{A} = [*\mathbf{A}*^\top] : \delta\mathbf{A}.$$

Proposition E.1. *Assume the boundary positions and mesh connectivity allow a locally injective map to exist. In the space of* $\mathbf{A} \geq 0$ *that* $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1, \forall j$, *$V_4$ has value* 0 *at its minimizers.*

Proof. We have

$$\mathbf{p}_u = -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{R}\mathbf{g}_u, \mathbf{p}_v = -[\mathbf{G}^\top \mathbf{A}\mathbf{G}]^\dagger \mathbf{R}\mathbf{g}_v.$$

Denote

$$\mathbf{P}_{(j)} = [\mathbf{G}_{(j)}\mathbf{p}_u, \mathbf{G}_{(j)}\mathbf{p}_v]^\top.$$

For $\mathbf{a}$ that minimizes V_4, the gradient $\frac{\mathrm{d}V_4^u}{\mathrm{d}\mathbf{a}} + \frac{\mathrm{d}V_4^v}{\mathrm{d}\mathbf{a}}$, when unfolded to a 2×2 symmetric matrix, $\mathbf{P}_{(j)}^\top \mathbf{P}_{(j)} \in \mathbb{R}^{2\times 2}$, must be orthogonal to the constraint $\det\left(\mathbf{m}_j^{-1}\mathbf{A}_{(j)}\right) = 1$. We have $*\mathbf{A}_{(j)}*^\top \propto \mathbf{P}_{(j)}^\top \mathbf{P}_{(j)}$. So we conclude that $\mathbf{m}_j^{-1}\mathbf{A}_{(j)} = |\det(\mathbf{P}_{(j)}|[\mathbf{P}_{(j)}^\top \mathbf{P}_{(j)}]^{-1}$, which indicates that $\mathbf{p}_u, \mathbf{p}_v, \mathbf{A}$ satisfy

$$\frac{1}{2}\mathbf{p}_u^\top \mathbf{G}^\top \mathbf{A}\mathbf{G}\mathbf{p}_u + \frac{1}{2}\mathbf{p}_v^\top \mathbf{G}^\top \mathbf{A}\mathbf{G}\mathbf{p}_v = \mathcal{U}(\mathbf{p}_u, \mathbf{p}_v) = \mathcal{A}(\mathbf{p}_u, \mathbf{p}_v).$$

So, V_4 is minimized at $V_4 = 0$. □

F OTHER DETAILS AND DISCUSSION

F.1 Performance and Implementation Details

Below, first we list a few common tricks to avoid pitfalls that might invoke unnecessary computations. Many of them are not even a concern if being implemented as per-element operations, but can add extra overhead if being implemented as matrix operations.

Computation re-ordering and operation merging. $\textsc{span}(\cdot)^\top$ followed by sparse-matrix-vector multiplication can be replaced by element-wise vector product. Thus, in our implementation we never assemble the sparse matrix $\textsc{span}(\cdot)$. Specifically, for vectors

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \in \mathbb{R}^{2f\times 1}$$

we have

$$\textsc{sp}(\mathbf{b})^\top \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \textsc{diag}(\mathbf{b}_1) & \\ \textsc{diag}(\mathbf{b}_2) & \textsc{diag}(\mathbf{b}_1) \\ & \textsc{diag}(\mathbf{b}_2) \end{bmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \odot \mathbf{c}_1 \\ \mathbf{b}_1 \odot \mathbf{c}_2 + \mathbf{b}_2 \odot \mathbf{c}_1 \\ \mathbf{b}_2 \odot \mathbf{c}_2 \end{bmatrix},$$

where $\textsc{sp}(\mathbf{b})^\top \mathbf{c} \equiv \textsc{sp}(\mathbf{c})^\top \mathbf{b}$ is the outer product of vector fields $\mathbf{b}$ and $\mathbf{c}$.

Matrix-matrix multiplication, except assembling the Laplacian $\mathbf{G}^\top \mathbf{A}\mathbf{G}$, can be completely avoided at runtime by *cascading matrix-vector multiplications*. For example, $\mathbf{S}\mathbf{G}\mathbf{u}$ should be evaluated as $\mathbf{S}(\mathbf{G}\mathbf{u})$, rather than $(\mathbf{S}\mathbf{G})\mathbf{u}$.

Symbolic pre-factorization. The sub-block of the discrete anisotropic Laplacian $\mathbf{S}^\top\mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{S}$ has a fixed sparsity pattern. We symbolically pre-factorize it once and reuse using SuiteSparse [Davis et al. 2015], as is done by Wang and Solomon [2021].

Tensor differentiability. Relying on the auto-differentiation feature of MATLAB, the parameterization $\xi \longrightarrow \mu$ may have some issues with differentiability at $(0,0)$ due to subtle numerical concerns. This is not an issue in practice, since exact equality for floating numbers are almost impossible. When initializing ξ as $(0,0)$, we add a tiny random number as a deviation to avoid the non-differentiable issue.

Thin triangle modification. When evaluating our method on the bulk of dataset [Du et al. 2020], which is taken from [Liu et al. 2018] and has many thin triangles, we apply the following modification: We set any angle less than $\pi/6$ to be $\pi/6$, and normalize so that angles in each triangle sums to π. So the smallest angle in the mesh would be larger than $\pi/8$. We then use this normalized reference triangle to compute our $\mathbf{G}, \mathbf{m}$. The scale of the modified triangle does not matter since it will cancel out.

Adam solver. For Adam, we use VectorAdam [Ling et al. 2022], instead of the ordinary Adam. The only place they differ in our 2D setting is that the factor to normalize a complex number comes from the norm taken over the complex number, instead of separating the real and imaginary parts and independently normalizing, which is also a common practice in training complex-valued neural networks. It makes our convergence trajectory exactly invariant to the choice of per-triangle tangent spaces. However, in practice we usually do not observe much difference, but we suspect that adversarially choosing tangent spaces might make ordinary Adam bad.

F.2 A Newton Solver for Post-processing

Back to the discussion of the Newton solver in §9.2: Let the objective P be the weighted Dirichlet energy: we use P to emphasize that now it is a function of the tuple $(\mathbf{u}, \mathbf{v}, \xi)$. Again, we use the same $\xi \longrightarrow \mathbf{A}$ in §8 and have an unconstrained optimization problem.

$$\begin{aligned}&\min_{\mathbf{u},\mathbf{v},\xi} P(\mathbf{u}, \mathbf{v}, \xi)\\ &P(\mathbf{u}, \mathbf{v}, \xi) := \frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{A}(\xi)\mathbf{G}\mathbf{u} + \frac{1}{2}\mathbf{v}^\top\mathbf{G}^\top\mathbf{A}(\xi)\mathbf{G}\mathbf{v}.\end{aligned} \tag{45}$$

The gradient and Hessian can be derived:

$$\frac{\partial P}{\partial(\mathbf{u}, \mathbf{v}, \xi)} = \begin{bmatrix} \mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{u} \\ \mathbf{G}^\top\mathbf{A}\mathbf{G}\mathbf{v} \\ \frac{\partial \mathbf{a}}{\partial \xi}\left[\frac{1}{2}\textsc{sp}(\mathbf{G}\mathbf{u})^\top\mathbf{G}\mathbf{u} + \frac{1}{2}\textsc{sp}(\mathbf{G}\mathbf{v})^\top\mathbf{G}\mathbf{v}\right]\end{bmatrix}$$

$$\frac{\partial^2 P}{\partial(\mathbf{u}, \mathbf{v}, \xi)^2} = \begin{bmatrix} \mathbf{G}^\top\mathbf{A}\mathbf{G} & \mathbf{0} & \mathbf{G}^\top\textsc{sp}(\mathbf{G}\mathbf{u})\frac{\partial \mathbf{a}}{\partial \xi}^\top \\ \mathbf{0} & \mathbf{G}^\top\mathbf{A}\mathbf{G} & \mathbf{G}^\top\textsc{sp}(\mathbf{G}\mathbf{v})\frac{\partial \mathbf{a}}{\partial \xi}^\top \\ \frac{\partial \mathbf{a}}{\partial \xi}\textsc{sp}(\mathbf{G}\mathbf{u})^\top\mathbf{G} & \frac{\partial \mathbf{a}}{\partial \xi}\textsc{sp}(\mathbf{G}\mathbf{v})^\top\mathbf{G} & \mathbf{H}\end{bmatrix}.$$

We have the following block diagonal matrix:

$$\mathbf{H} = \begin{bmatrix}\mathbf{H}_{uu} & \mathbf{H}_{uv} \\ \mathbf{H}_{uv} & \mathbf{H}_{vv}\end{bmatrix}$$

in which $\mathbf{H}_{uu}, \mathbf{H}_{uv}, \mathbf{H}_{vv}$ are diagonal, which can be computed using the chain rule to find some second order derivatives. The per-triangle computation for tensor parameterization can be done by auto-differentiation. For example, we have

$$\begin{aligned}\mathbf{H}_{uu} = &\frac{1}{2}\left[\textsc{diag}\left(\mathbf{G}_x\mathbf{u} \odot \mathbf{G}_x\mathbf{u} + \mathbf{G}_x\mathbf{v} \odot \mathbf{G}_x\mathbf{v}\right)\right]\textsc{diag}\left(\frac{\partial^2 \mathbf{a}_{11}}{\partial \xi_u^2}\right)\\ &+ \left[\textsc{diag}\left(\mathbf{G}_x\mathbf{u} \odot \mathbf{G}_y\mathbf{u} + \mathbf{G}_x\mathbf{v} \odot \mathbf{G}_y\mathbf{v}\right)\right]\textsc{diag}\left(\frac{\partial^2 \mathbf{a}_{12}}{\partial \xi_u^2}\right)\\ &+ \frac{1}{2}\left[\textsc{diag}\left(\mathbf{G}_y\mathbf{u} \odot \mathbf{G}_y\mathbf{u} + \mathbf{G}_y\mathbf{v} \odot \mathbf{G}_y\mathbf{v}\right)\right]\textsc{diag}\left(\frac{\partial^2 \mathbf{a}_{22}}{\partial \xi_u^2}\right)\end{aligned}$$

in which $\odot$ denotes elementwise multiplication of two vectors, and second derivatives $\frac{\partial^2 \mathbf{a}_{11}}{\partial \xi_u^2}, \frac{\partial^2 \mathbf{a}_{12}}{\partial \xi_u^2}, \frac{\partial^2 \mathbf{a}_{22}}{\partial \xi_u^2} \in \mathbb{R}^{f\times 1}$ can also be handled by auto-differentiation. Similar formulas hold for $\mathbf{H}_{uv}, \mathbf{H}_{vv}$. Finally we apply a standard trust region method.

F.3 Finding Injections as Minimal Surfaces

Modern geometry suggests advantages of our core functional, the weighted Dirichlet energy, over the area functional, i.e., total unsigned area. A simple yet remarkable observation in [Xu et al. 2011] can be stated equivalently: Injective maps are *minimal surfaces*. However, in the mathematical literature of minimal surfaces, as an object of study, (total unsigned) area became obsolete almost immediately, replaced by the Dirichlet energy, which has nicer analytical properties. This modern approach to minimal surfaces dates to the seminal work of Douglas and Radó [Douglas 1931]. It has been argued that the area functional is not sufficiently *coercive*, i.e., it does not offer enough control over the map by admitting a huge kernel [Colding and Minicozzi 2011]. The deficiency of nice analytical properties for area functionals directly transfers to practical difficulties when working with total unsigned area: the vanishing gradient and non-smoothness issues. Beyond the theoretical superiority for mathematical analysis, we have demonstrate that *computationally* (weighted) Dirichlet energy, is a better optimization objective: weighted Dirichlet provides a *tight* upper bound of the original objective—so the minimizers remain the same; it is *smooth* and/or *differentiable*, much easier to optimize.

Our approach and TLC represent two ways to introduce coercivity: TLC adds a regularizer, lifting to a higher dimension and measuring volume there, while we work within the $\mathbf{A}$-parametric family of Dirichlet energies.