DOI: 10.1111/cgf.70196
Eurographics Symposium on Geometry Processing 2025
M. Attene and S. Sellán
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 5

Volume Preserving Neural Shape Morphing

Camille Buonomo[1] and Julie Digne [1] and Raphaëlle Chaine[1]

[1]CNRS, Université Claude Bernard Lyon 1, INSA Lyon, LIRIS UMR5205

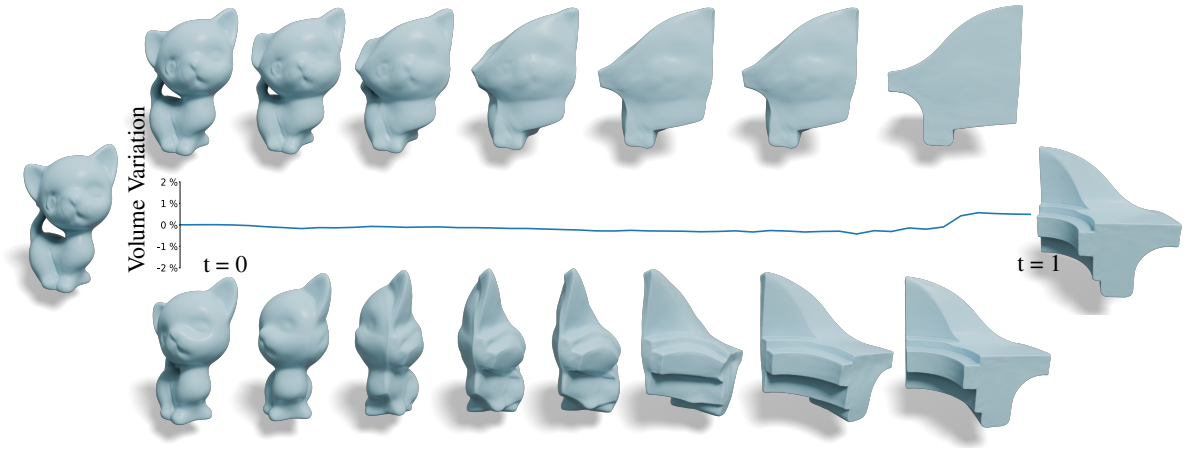


Figure 1: *Our method interpolates between two shapes by preserving the volume when no correspondences are available. Here the kitten is morphed into a fandisk and the morphing is only dependent on the relative positions of the source and target shapes. The bottom row shows the same morphing combined with a rotation for better visualization.*

Abstract

Shape interpolation is a long standing challenge of geometry processing. As it is ill-posed, shape interpolation methods always work under some hypothesis such as semantic part matching or least displacement. Among such constraints, volume preservation is one of the traditional animation principles. In this paper we propose a method to interpolate between shapes in arbitrary poses favoring volume and topology preservation. To do so, we rely on a level set representation of the shape and its advection by a velocity field through the level set equation, both shape representation and velocity fields being parameterized as neural networks. While divergence free velocity fields ensure volume and topology preservation, they are incompatible with the Eikonal constraint of signed distance functions. This leads us to introduce the notion of adaptive divergence velocity field, a construction compatible with the Eikonal equation with theoretical guarantee on the shape volume preservation. In the non constant volume setting, our method is still helpful to provide a natural morphing, by combining it with a parameterization of the volume change over time. We show experimentally that our method exhibits better volume preservation than other recent approaches, limits topological changes and preserves the structures of shapes better without landmark correspondences.

CCS Concepts

• Mathematics of computing → Partial differential equations; • Computing methodologies → Parametric curve and surface models; Neural networks;

1. Introduction

Morphing between geometric shapes, when no correspondences are available is an ill-posed problem. As such, all methods have introduced some hypothesis on the deformation, by constraining it to be as rigid as possible or by imposing least displacement. Among all these hypotheses, the volume preserving constraint is rooted in the celebrated *12 principles of animation* [Tho95], where the first principle, squash and stretch, recommends keeping the volume con-

stant. Another interesting constraint is to avoid unnecessary topological changes, e.g. avoiding a shape to disappear and reappear out of the blue elsewhere.

In this paper, we tackle the problem of interpolating between arbitrary shapes in arbitrary poses by relying on a neural implicit shape representation and leveraging the level set equation [BCOS01] which links the spatio-temporal implicit representation with an explicit velocity field. We first illustrate that enforcing the whole velocity field to be divergence-free is incompatible with obtaining a signed distance field. Indeed a divergence-free velocity field preserves the volume of all level sets when we are only interested in preserving the volume of the level set corresponding to the deforming object. Hence we introduce the notion of adaptive divergence velocity field, which preserves the 0 level set volume, and show much better volume and topolgical preservation than competing methods experimentally. As a side product, we also adapt our method to non-constant volume deformations by combining the constant volume evolution with a progressive rescaling.

To summarize we make the following contributions:

- The definition of *adaptative divergence vector fields*, a relaxation of divergence free vector field for which we prove the shape volume preservation.
- A neural framework combining a learned adaptive divergence vector field and a time-varying approximate signed distance field.
- A framework that strongly favors topology preservation while still allowing some topology change if required or if it offers a better compromise.

2. Related Work

Shape Morphing using explicit methods In Computer Graphics, shapes are commonly described by meshes, and several approaches have been developed to deform one mesh into another. This often requires the source and target meshes to share the same topology. Alexa et al. [ACOL00] first build a common triangulation of the interior of the shapes putting tetrahedra in one-to-one correspondence and computing an as-rigid-as possible deformation between them. But this method requires boundary vertex correspondences. Interpolating between several meshes as also been tackled by minimizing the elastic deformation to each input mesh [VTSSH15], with the constraint that the meshes need to be embeddings of the same topological mesh, hence giving the vertex correspondences implicitly. This approach can be applied to volume or surface meshes. Volume preservation requires the use of volume tetrahedral meshes. In this case, Aharon et al [ACZW19] exploits the fact that the space of metrics with bounded isometric and angular distortion is convex to interpolate between two or more shapes while preserving features. When vertex correspondences are not given, they must be estimated. Huang et al. [HAWG08], for example, recover correspondences between meshes under the isometric deformation hypothesis. Correspondences can also be recovered as a result of a divergence-free shape deformation. Given two compatible input shapes, Eisenberger et al. [ELC18] progressively construct a volume preserving deformation field by imposing a zero-divergence constraint. This method alternates between optimizing the deformation field and calculating correspondences for a small subset of keypoints based on feature description.

When correspondences are hard or impossible to compute due to shapes being unrelated or having different topological genus, it is still possible to perform interpolation in the sense of Optimal Transport (OT), considering the shapes as mass distributions (e.g. [Lév15], [FCVP17]). A drawback of these approaches is that the trajectories of the points will be solely guided by the minimization of their lengths, hence producing tearings in the shape. A dynamic formulation of OT was proposed by Benamou and Brenier [BB00] to preserve the mass when morphing from one density distribution to another, by optimizing the displacement vector field and not only the transport map.

Shape Morphing using implicit methods Implicit functions are also popular to describe shapes as level sets of some scalar field. They are interesting to design shapes that break or merge, or change topology, without requiring to maintain a mesh structure. However, directly interpolating between two existing implicit functions for shape morphing gives no control over the intermediate shapes and their volumes. It can be improved by defining a relevant velocity field at each surface point and some surface tension [DC98]. If the scalar field corresponds to a signed distance, Cohen-Or et al [COSL98] guide the interpolation with a warp function controlled by a set of corresponding anchor points, including a rigid rotational part and a minimal elastic part. Finally, Turk and O'Brien [TO99] directly compute a transformation between two 3D shapes by casting the problem as a scattered data interpolation problem in spacetime (4D) and using radial basis functions for the implicit surface representation. Tao et al. [TSB16] seek to learn the most rigid deformation between two shapes. To do so, they perform explicit level set tracking by computing a discrete solution to the level set equation under the constraint that the velocity field must be an approximate Killing vector field. In a broader context, Osher and Sethian [OF04] developped the level set method in order to track the motion of an interface under a velocity field, following the level set equation (LSE). The interface is defined as the zero level set of an implicit function with Lipschitz properties, usually a signed distance function. Bertalmio et al. [BCOS01] also introduced the use of partial differential equations to solve variational problems on implicit surfaces.

Implicit Neural Fields and shape deformation Recently, the idea has emerged to use a neural network to represent the implicit function encoding the shape (see e.g. [XTS*22] for a survey of this trend). This permits to be invariant of the surface discretization [YBHK21]. During training, an Eikonal regularization term brings this function as close as possible to a signed distance function [GYH*20]. Categories of shapes can also be taken into account by using latent code-conditioned decoder networks [PFS*19, MON*19]. Depending on the architecture of the network and activation functions (eg. SIREN [SMB*20]), there may be more or less fine-grained surface details and possible guarantees on the Lipschitz properties of the implicit function [CB24, WWY*25]. Neural Implicits have also proven useful where the input information is limited to a set of geometric design requirements. In that setting, Geometry-Informed Neural Networks [BRV*24] can be used

to solve a topology optimization problem using neural fields that approximate a signed-distance function.

Shape interpolation can be done in latent space, by interpolating the latent codes [CZ19]. It is also possible to add constraints to get better interpolations. For example, Atzmon et al. [ANVL21] add multiple deformation priors allowing to interpolate plausibly between shapes. However this interpolation requires a database to build the latent space. Liu et al. [LWJ*22] propose to control the variations of the implicit function with respect to a varying parameter such as a latent variable in the latent space setting. This can be also used for shape morphing by using the time as the varying parameter. This is done by constraining the Lipschitz constant w.r.t. the parameter to be as low as possible. However, this approach does not offer any control over the volume.

Dynamic implicit functions have also been tackled using the Level Set Equation framework. Given a set of points sampled on a shape and a velocity field, Mehta et al. [MCR22] alternate between advecting sample points, learning the updated signed distance field using the advected points and resampling the surface for the next time step. Another approach [NDSS*23] is to directly learn the solution of the level set equation under the form of an implicit neural function that is fully continuous, both in time and space (NISE). This method can perform shape deformation driven by any given vector field, including an adhoc field proposed for interpolating between two different shapes, without any control on the properties and the volume of the intermediate shapes. In particular, the solution is not constrained to be a signed distance field for intermediate times.

Very recently, Sang et al. introduced an additional neural network to learn the velocity field to be used in the level set equation [SCC*25]. This method adds a penalization on the divergence of the velocity field to favor volume preservation, as well as a smoothness regularization. Through a modified LSE, it builds a time-varying signed distance field with lower volume variation than any method before. Dense correspondences between the two shapes are also handled through the flow of a dense set of landmarks. However, this approach offers no strong guarantee on the divergence of the velocity field and the preservation of the volume. We call this method INSD (Implicit Neural Surface Deformation) and compare to it. For comparison purposes, we also introduce a landmark-free version of INSD, denoted as LF-INSD in our experiments.

Finally working on implicit vector fields using Neural Networks has been explored in the Optimal Transport setting. Richter-Powell et al. [RPLC22] propose to constrain a neural network to build a divergence free vector field, an idea that is also exploited in our work.

3. Neural Shape Morphing

Our method builds on the level set framework to describe shapes and their evolution. Let us assume that we have two shapes in $\mathbb{R}^d$ (with $d = 2$ or 3): a source shape $\mathcal{S}_0$ at $t = 0$, and a target shape $\mathcal{S}_1$ at $t = 1$, with g_0 and g_1 their respective signed distance functions (SDF). We always assume that the time interval is $[0,1]$, without loss of generality. Let $x(t)$ be the trajectory of a point from $\mathcal{S}_0$ to $\mathcal{S}_1$. A scalar field f is an implicit representation of the intermediate shape $\mathcal{S}_t$ if for all t, $f(x(t),t) = 0$. Taking the time derivative of this equality gives us the *Level Set Equation*:

$$\frac{df(x(t),t)}{dt} = \frac{\partial f}{\partial t}(x(t),t) + \langle \nabla f(x(t),t), \frac{\partial x}{\partial t}(t) \rangle = 0$$

The quantity $\frac{\partial x}{\partial t}(t)$ can be viewed as a vector field $\mathbf{V}(x(t),t)$. Thus, we aim at interpolating between $\mathcal{S}_0$ and $\mathcal{S}_1$ by determining a scalar field f and a vector field $\mathbf{V}$ satisfying:

$$\begin{cases} \frac{\partial f}{\partial t} + \langle \nabla f, \mathbf{V} \rangle & = 0 \text{ on } \mathbb{R}^d \times [0,1] \\ f(x,0) & = g_0(x)\ \forall x \in \mathbb{R}^d \\ f(x,1) & = g_1(x)\ \forall x \in \mathbb{R}^d \end{cases} \tag{1}$$

In the above equation, and in the remainder of the paper, the differential operators ∇, div and curl are defined with respect to the spatial coordinates only, omitting the time.

Similarly to the NISE approach proposed by Novello et al. [NDSS*23], we propose to learn f as a neural network f_θ with parameters θ. In our case, we also penalize $f_\theta(\cdot,t)$ to approximate the signed distance field to the shape at time t. Indeed, Signed distance functions are particularly useful for sphere tracing or other geometric queries. Hence, $f(\cdot,t)$ should verify the Eikonal equation for each t, and not only at $t = 0$ and $t = 1$ as in NISE. We recall the Eikonal equation :

$$|| \nabla f(x,\cdot) || = 1, \forall x \in \mathbb{R}^d \text{ where the gradient is defined.} \tag{2}$$

Our approach differs from NISE in the design the velocity field $\mathbf{V}$. In NISE, $\mathbf{V}$ is handcrafted by an ad-hoc combination of f_θ and g_1 ensuring that the function transitions between the two implicit fields. $\mathbf{V}$ aligns with the iso-surface normal field $\nabla f_\theta(x,t)$ as:

$$\mathbf{V}(p,t) = -(g_1(p) - f_\theta(p,t)) \frac{\nabla f_\theta(t,p)}{\|\nabla f_\theta(t,p)\|} \tag{3}$$

However, performing surface morphing using this vector field is incompatible with an Eikonal constraint at any time. Furthermore it does not allow for volume and topology preservation constraints: experimental results show that the enclosed volume is not entirely preserved during the transformation and can even disappear in some places and reappear in others (Figure 2).

Instead of using the NISE handcrafted velocity field, we can learn $\mathbf{V}$ and f jointly as neural networks f_θ, $\mathbf{V}_\theta$ with an Eikonal constraint over time but without any volume or topology constraint. As illustrated in Figure 2, for both this method and NISE, the surface is not advected but partly destroyed and regrown at the location of the target surface. This phenomenon is due to the existence of uncontrolled sources and sinks in $\mathbf{V}_\theta$, which can collapse the surface at the initial location and regrow it at the target location without really displacing it.

The aim of our work is to add consistent constraints when jointly estimating $\mathbf{V}$ and f, to provide constant-volume intermediate shapes and a reasonable approximation of the signed distance to the shape.

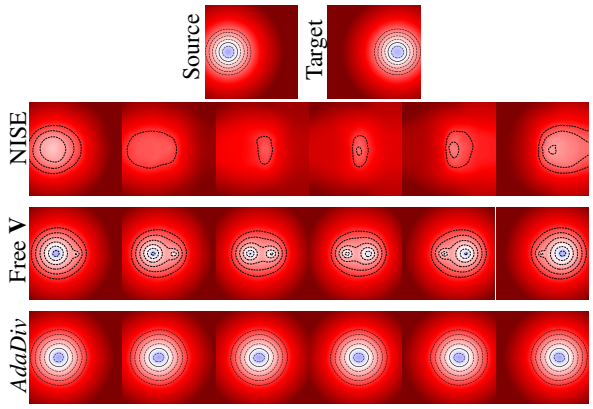


Figure 2: *2D translation of a disk. With NISE the shape disappears and reappears during the morphing (top). This problem also occurs when learning* $\mathbf{V}$ *and* f *jointly with an Eikonal constraint over time but without additional constraint on volume and topology preservation (middle). Our method (bottom) succeeds in preserving the shape through the translation due to the volume preserving property of our vector field. Top row: source* ($t=0$) *and target* ($t=1$)*, interpolation times:* $t = 0.14, 0.28, 0.42, 0.57, 0.71, 0.85$

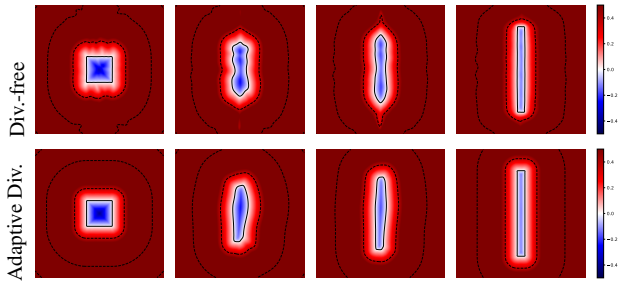


Figure 3: *An example of constant volume morphing which cannot satisfy the divergence-free condition for* $\mathbf{V}$ *in the level set framework. This condition applies for all level sets and collides with the Eikonal condition* ($t = 0, 0.25, 0.75, 1$)*.*

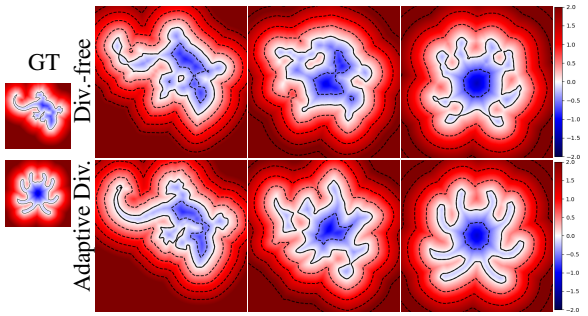


Figure 4: *Enforcing zero-divergence on* $\mathbf{V}$ *yields very low detail surfaces, a consequence of the volume preservation of every level set. Using adaptative divergence, the surfaces details are better recovered* ($t = 0, t = 0.5, t = 1$)*.*

4. Divergence free neural vector field

The divergence of a vector field measures the volume density of the outward flux from an infinitesimal volume. It is defined for a vector field $\mathbf{V}$ as:

$$\operatorname{div}(\mathbf{V}) = \sum_i \partial_i \mathbf{V}_i$$

$\mathbf{V}$ is said to be divergence free (or equivalently to have zero divergence) when $\operatorname{div}\mathbf{V} = 0$. Zero divergence implies volume and topology preservation of any volume advected by $\mathbf{V}$. Indeed a divergence free velocity field cannot involve neither source nor sink. Divergence-free vector field is important in many fields: incompressible fluids dynamics, animation, virtual sculpture. In Computer Graphics, volume preserving shape editing tools often rely on divergence-free vector fields [ACWK06, vFTS06]. For example, Von Funck et al. builds a deformation vector field as the cross-product of two gradient fields. By construction this vector field is divergence-free. Here we propose a different construction.

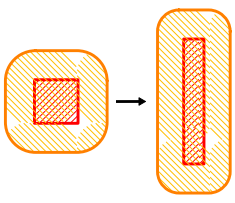
When jointly learning a neural vector field $\mathbf{V}_\theta$ and neural scalar field f_θ so that they satisfy the level set equation 1, control over the divergence of $\mathbf{V}_\theta$ can be obtained by adding various terms to the loss, such as a penalisation of the divergence as in INSD [SCC*25]:

$$\int_{\mathbb{R}^d \times (a,b)} |\operatorname{div}(\mathbf{V})| \, \mathrm{d}p\mathrm{d}t \tag{4}$$

Doing so only favors a low divergence but does not guarantee it to be zero. Instead, we propose to learn a neural vector field with a strongly imposed divergence free condition.

In our neural implicit setting, rather than directly learning a volume preserving velocity field $\mathbf{V}$ in $\mathbb{R}^3$, we propose to learn an auxiliary vector field $\mathbf{D}$ whose curl is the divergence-free field that will guide the morphing: $\mathbf{V} = \operatorname{curl}(\mathbf{D})$. The curl operator can be computed automatically through auto-differentiation of the neural network $\mathbf{D}_\theta$ producing the auxiliary vector field $\mathbf{D}$. Note that it is also possible to build a divergence-free vector field in $\mathbb{R}^2$ by using a scalar potential $\mathcal{H}$ and a skew-symmetric matrix A_{sks}: the vector field given by $A_{sks}\nabla\mathcal{H}$ is divergence-free (See appendix A). We call *NullDiv* the network architecture (in 2D or 3D) producing such guaranteed divergence-free vector field. By the Helmotz theorem, any divergence free vector field of $\mathbb{R}^3$ can be obtained as the curl of another vector field. Hence, this architecture is a universal approximator of divergence free vector fields as long as the network learning $\mathbf{D}$ is a universal approximator. A similar approach for constraining a neural network to produce divergence-free vector fields in any dimension has been proposed by Richter-Powell et al. [RPLC22], in the context of continuity equations.

Unfortunately, none of the fields produced by *NullDiv* networks are satisfactory for solving the level set equation (1) under the Eikonal constraint enforced over time (equation 2). Indeed $\mathbf{V}_\theta$ being divergence-free, it preserves the volume of *every* level set of the potential f_θ which is incompatible with the Eikonal constraint. To illus-



trate this, let us consider a 1×1 square morphing into a $0.3 \times \frac{1}{0.3}$ rectangle. For their signed distance fields, both 0-level sets (red stripes in the inset figure) have area 1. Yet their 1-level sets (orange stripes in the inset figure) have different areas ($5+\pi \approx 8.14$ for the square, $1+2(0.3+\frac{1}{0.3})+\pi \approx 11.41$ for the rectangle) To make the areas of their 1-level sets equal, the distance between the 0-level set and the 1-level set should either grow for the square or diminish for the rectangle, breaking therefore the Eikonal constraint. Figure 3 shows the numerical result of this 2D example morphing, yielding defect-laden level sets throughout the deformation. Figure 4 shows that with more detailed shapes, the network degrades the initial shape when trying to satisfy both LSE and Eikonal constraint, and even so it cannot satisfy them.

Therefore, to satisfy the Eikonal equation everywhere (where the gradient is defined), an LSE-based method should preserve the volume of the 0-level set only.

Our approach is thus different from INSD [SCC*25] which penalizes a nonzero divergence everywhere but relaxes the Eikonal property away from the surface. This approach works better with the introduction of a dense set of correspondences between the two surfaces.

5. Adaptive Divergence

5.1. Definition and Properties

Let us consider a divergence-free velocity field defined in the ambient space, which morphs $\mathcal{S}_0$ into $\mathcal{S}_1$. We propose to preserve this vector field only on the surface of the evolving shape, and to modify it elsewhere to better comply with the Eikonal equation, while keeping a zero divergence on the surface. Therefore, we augment the divergence-free component by an additional scalar potential u which is added through its gradient modulated by a function of f:

$$\mathbf{V} = \mathrm{curl}(\mathbf{D}) + \beta(f)\nabla u \tag{5}$$

with β a continuous non negative scalar function such that $\beta(0) = 0$ and $\beta'(0) = 0$. In our implementation, we used $\beta(s) = 1 - \exp(-\alpha s^2)$, with $\alpha = 0.1$, which we found to be experimentally efficient.

By construction, $\mathbf{V}$ is the sum of a divergence-free vector field and a gradient vector field which is modulated to be 0 and have 0 divergence on the surface. We call *AdaDiv* our estimation of such a vector field. One can notice that it is similar in spirit to the Helmotz-Hodge decomposition with a geometric modulation. Note however that after modulation by $\beta(f)$, the gradient of u may lose its irrotationality, depending on the angle between ∇f and ∇u. Indeed, we have:

$$\mathrm{div}\,\mathbf{V} = \beta(f)\Delta u + \beta'(f)\nabla f \cdot \nabla u \tag{6}$$

We show that this constructive way of relaxing the divergence of a velocity field preserves the shape volume advected by $\mathbf{V}$. More precisely, for any divergence-free vector field $\mathbf{W}$, if we add a vector field $\mathbf{F}$ whose value and divergence are both vanishing on the surface, the induced morphing is volume-preserving.

Before defining formally the notion of adaptive divergence, we first recall the definition of a flow $\phi_{\mathbf{V}}$ associated with a vector field $\mathbf{V}(x,t)$. The advection $y(t)$ of a point y_0 through $\mathbf{V}$ writes as an Ordinary Differential Equation (ODE):

$$\begin{cases} \frac{\partial y}{\partial t} & = \mathbf{V}(y(t),t) \\ y(0) & = y_0 \end{cases} \tag{7}$$

If $\mathbf{V}$ satisfies the Cauchy-Lipschitz conditions, this ODE has a unique solution $y(t)$. Then the flow $\phi_{\mathbf{V}}$, defined as:

$$\forall y_0, t,\ \phi_{\mathbf{V}}(y_0,t) = y(t)$$

By definition, $\phi_{\mathbf{V}}$ is a diffeomorphism, with ${\phi_{\mathbf{V}}}^{-1}$ the inverse flow: for all $(x,t) \in \mathbb{R}^d \times [0,1]$, ${\phi_{\mathbf{V}}}^{-1}(\phi_{\mathbf{V}}(x,t),t) = x$.

Definition 1 (Adaptive divergence) We say that a time varying vector field $\mathbf{V} \in \mathbb{R}^d$ has adaptive divergence with respect to an initial shape $\mathcal{S}$ if there exists a divergence-free vector field $\mathbf{W}$ with associated flow $\phi_{\mathbf{W}}$ and a vector field $\mathbf{F}$ such that:

$$\begin{aligned}
&\forall x \in \mathbb{R}^d, t \in [0,1], \mathbf{V}(x,t) = \mathbf{W}(x,t) + \mathbf{F}(x,t) \\
&\forall x \in \partial\mathcal{S}, t \in [0,1], \mathbf{F}(\phi_{\mathbf{W}}(x,t),t) = 0 \\
&\forall x \in \partial\mathcal{S}, t \in [0,1], \mathrm{div}(\mathbf{F})(\phi_{\mathbf{W}}(x,t),t) = 0 \text{ so } \mathrm{div}(\mathbf{V})(\phi_{\mathbf{W}}(x,t),t) = 0
\end{aligned}$$

where $\partial\mathcal{S}$ is the boundary surface of $\mathcal{S}$. Importantly enough, for given $\mathcal{S}$ and $\mathbf{W}$ there exist multiple adaptive divergence fields, and $\mathbf{W}$ is one of them (with $\mathbf{F} = 0$).

Let us consider g_0 an implicit representation of an input shape $\mathcal{S}_0$, and $\mathbf{W}$ a time varying divergence-free vector field. If we further assume that $\mathbf{W}$ satisfies the Cauchy-Lipschitz conditions (we will show later that this is the case in our construction), then the flow $\phi_{\mathbf{W}}$ is a diffeomorphism with inverse $\phi_{\mathbf{W}}^{-1}$. $f(x,t)$ the implicit representation of the evolving surface under this flow is such that $f(x,t) = g_0({\phi_{\mathbf{W}}}^{-1}(x,t))$. Let us show that modifying $\mathbf{W}$ by adding a time varying vector field $\mathbf{U}$ modulated by $\beta(f)$ (as in Equation 5) builds an adaptive divergence vector field $\mathbf{V}$ with respect to $\mathcal{S}$.

For any smooth time-varying vector field $\mathbf{U}$, let us define for all $(x,t) \in \mathbb{R}^d \times [0,1]$, $\mathbf{F}(x,t) = \beta(f(x,t))\mathbf{U}(x,t)$, and $\mathbf{V}(x,t) = \mathbf{W}(x,t) + \mathbf{F}(x,t)$.

Then for any $x \in \partial\mathcal{S} = \{x | g_0(x) = 0\}$:

$$\begin{aligned}
\mathbf{F}(\phi_{\mathbf{W}}(x,t),t) &= \beta(f(\phi_{\mathbf{W}}(x,t),t))\mathbf{U}(\phi_{\mathbf{W}}(x,t),t) \\
&= \beta(g_0({\phi_{\mathbf{W}}}^{-1}(\phi_{\mathbf{W}}(x,t),t)))\mathbf{U}(\phi_{\mathbf{W}}(x,t),t) \\
&= \beta(0)\mathbf{U}(\phi_{\mathbf{W}}(x,t),t) = 0,
\end{aligned} \tag{8}$$

$$\begin{aligned}
\mathrm{div}(\mathbf{F}) &= \mathrm{div}(\beta(f)\mathbf{U}) \\
&= \beta(f)\,\mathrm{div}(\mathbf{U}) + \langle \nabla\beta(f), \mathbf{U} \rangle \\
&= \beta(f)\,\mathrm{div}(\mathbf{U}) + \langle \beta'(f)\nabla f, \mathbf{U} \rangle.
\end{aligned} \tag{9}$$

Hence for any $x \in \partial\mathcal{S} = \{x | g_0(x) = 0\}$, $\mathrm{div}(F)(\phi_{\mathbf{W}}(x,t),t)$ is zero.

By taking $\mathbf{U} = \nabla u$ (with u a smooth function), we see that a velocity field $\mathbf{V}$ defined as in Equation 5 has adaptive divergence

with respect to $\mathcal{S}$. An advantage of taking the gradient form ∇u is that, far away from the surface, $\beta(f(x,t)) \approx 1$ so that $\mathbf{V} \approx \text{curl}(\mathbf{D}) + \nabla u$ where the second term is irrotational, hence corresponding to a Helmotz-Hodge decomposition.

Theorem 1 (Volume Preservation) Let $\mathbf{V}$ be an adaptive divergence vector field with respect to $\mathcal{S}$ and $\phi_{\mathbf{V}}$ the associated flow. Then, the advection of $\mathcal{S}$ by $\mathbf{V}$ preserves its volume :

$$d_t Vol(\phi_{\mathbf{V}}(\mathcal{S},t)) = 0$$

The proof relies on the fact that the adaptive divergence field $\mathbf{V}$ is constructed over a divergence-free field $\mathbf{W}$ deforming $\mathcal{S}_0$ into $\mathcal{S}_1$ and the fact that the flows $\phi_{\mathbf{V}}$ and $\phi_{\mathbf{W}}$ coincide on $\partial\mathcal{S}$. In addition, the points initially away from the surface cannot cross it, since the trajectories under an Ordinary Differential Equation cannot cross. The full proof is given in appendix B.

To predict a vector field $\mathbf{V}$ with adaptive divergence, we use a single neural network which produces $\mathbf{D}_\theta$, u_θ for computing both the divergence free component and u. The outputs are then combined with $\beta(f)$ modulation. This *AdaDiv* network takes as input the spatial coordinates of a point x and a time t. Since we use an infinitely smooth architecture with sinusoidal activations, $\text{curl}(\mathbf{D}_\theta)$ and $\beta(f)\nabla u$ are Lipschitz vector fields (of unknown constant) thus the Cauchy-Lipschitz conditions hold and so does our volume preservation theorem.

5.2. Neural morphing using adaptive divergence

Given a source and a target shape of identical volumes, even with the improvements introduced earlier there is no guarantee that there exists a velocity vector field $\mathbf{V}$ with adaptive divergence and a potential field f satisfying both Equation 1 (LSE) and the Eikonal constraint for f (Equation 2). However, in our setting, we learn the velocity field $\mathbf{V}$ and scalar field f through two different neural networks with the link between the estimations f_θ and $\mathbf{V}_\theta$ enforced weakly through a loss. Therefore, such fields will always be produced, sometimes at the cost of a higher LSE and Eikonal errors. In practice, our definition offers enough degrees of freedom for the fields produced by our neural networks to be well behaved, i.e. the local error with respect to the LSE and the Eikonal equation remains small everywhere and it is concentrated on the medial axis of the shape where the gradient of the distance function is not defined (See supplementary material). Of course, different adaptive divergence velocity fields may be designed for morphing the source to the target in the LSE framework. We observe in practice that our neural approximation tends to favor the generation of a direct morphing between parts that are close in the source and target. More precisely, the resulting fields tend to deform the input shape by favoring tangential sliding over tearing, thus avoiding the tearing effects generally observed with Optimal Transport.

5.3. Topological changes

The advection of a shape by a divergence-free field is fold-over free, i.e. it does not allow topological genus changes. By construction, this property remains valid for the advection of a shape by a field with adaptive divergence. However, since our networks yield a field with adaptive divergence and an implicit representation of the evolving shape that are weakly coupled, it may happen that the implicit representation undergoes topological changes. In this case, the local LSE and Eikonal errors will be high at the place and time of the topological changes. This is shown experimentally in the supplementary. Even in cases where the morphing operates between constant genus shapes, it may happen that topological noise appear as a result of a local minimum with higher LSE loss value.

6. Network architecture and training

Network Architecture and Optimization. Our method relies on two networks to produce the velocity field $\mathbf{V}_\theta$ (through a combination of $\mathbf{D}_\theta$ and u_θ) and the scalar potential f_θ jointly. For their training, two auxiliary networks are necessary to provide the SDF of the source and target shapes respectively. All our networks are fully connected networks with 6 hidden layers, 128 neurons per layer, and Sine activations. The two auxiliary networks are initialized as prescribed in the SIREN paper [SMB*20] and the velocity and time-varying SDF networks use the same initialization as NISE. We use ADAM as an optimizer with a learning rate of $5 \cdot 10^{-5}$.

Losses The auxiliary models are trained offline to produce the SDFs g_0 and g_1 of the source and target shapes respectively, with losses reflecting Dirichlet and Neumann boundary conditions and satisfaction of the Eikonal equation [GYH*20, SMB*20]. In addition to g_0 and g_1 we also take as input point sampled on the surfaces $\partial\mathcal{S}_0$ and $\partial\mathcal{S}_1$

The main networks $\mathbf{V}_\theta$ and f_θ are independent, but they are linked through the LSE which measures the compliance with the LSE equation for all t. It is the only loss part that involves the parameters of $\mathbf{V}_\theta$:

The LSE loss is similar as in NISE

$$l_{LSE} = \int_{\mathbb{R}^d \times [0,1]} \left| \frac{\partial f_\theta}{\partial t} - \langle \nabla f_\theta, \mathbf{V}_\theta \rangle \right| \, dp dt \tag{10}$$

The minimization of the error w.r.t. the Eikonal equation drives f_θ to be as close as possible to a signed distance. While in NISE, this loss is only enforced for times 0 and 1, we extend it to any $t \in [0,1]$ through a time-integral:

$$l_{Eik} = \int_{\mathbb{R}^d \times [0,1]} |\, 1 - |\, \nabla f_\theta(p,t) \,|\, | \, dp dt \tag{11}$$

Finally, the target and source surfaces attachment terms strive to make f mimic g_0 and g_1 at times 0 and 1 and match the input points of the shapes (Dirichlet loss), and align its spatial gradient with the source and target normals (Neumann loss). In our method, we have two terms for the Dirichlet loss: one that favors the points in ambient space at times 0 and 1 to have the correct SDF value, and one that focuses on the surface by sampling points on $\mathcal{S}_0$ and $\mathcal{S}_1$ and checking that their SDF is indeed 0. This adds more weight to the surface (similar to [NDSS*23]).

$$l_{Dirichlet} = \sum_{i=0,1} \int_{\mathbb{R}^d} \mid g_i - f_\theta(.,i) \mid \mathrm{d}p + \sum_{i=0,1} \int_{x \in \partial \mathcal{S}_i} |f_\theta(x,i)| dx \tag{12}$$

$$l_{Neumann} = \sum_{i=0,1} \int_{\mathcal{S}_i} \mid 1 - \langle \nabla f_\theta(.,i), \vec{N}_{\mathcal{S}_i} \rangle \mid \mathrm{d}p \tag{13}$$

All the losses integrals are computed using Monte-Carlo approximation by, at each gradient step, sampling points, either on the source and target surfaces ($l_{Dirichlet}, l_{Neumann}$) or uniformly in the ambient space (l_{LSE}, l_{Eik}). Then theses losses are combined as:

$$\mathcal{L} = \lambda_{LSE} l_{LSE} + \lambda_{Eik} l_{Eik} + \lambda_{Dirichlet} l_{Dirichlet} + \lambda_{Neumann} l_{Neumann} \tag{14}$$

In our experiments we always use the following weights: $\lambda_{LSE} = 10e4$, $\lambda_{Dirichlet} = 10e4$, $\lambda_{Neumann} = 10e2$, $\lambda_{Eik} = 10e1$. See the supplementary material for an ablation study on these terms. Both networks weights are updated simultaneously at each gradient step.

7. Results

Our method is implemented in pytorch and the code is available at `https://github.com/camillebnm/neural_shape_morphing`. We use shapes from the NISE github repository, the Stanford repository and Thingi10K [ZJ16]. Shapes are supplied as a set of points sampled on a surface with their normals, on which the auxiliary networks are trained offline. For vizualisation purposes, meshes are reconstructed using Marching Cubes with a 256^3 grid (0.6s per reconstruction), and rendered using blender. Additional details, results, comparisons and analyses are provided in the supplementary and the accompanying video.

Our method starts by normalizing the shapes so that they have the same volume. Volumes are estimated using a fixed grid of size 256 by counting the inside/outside voxels given by the SDF, and used to compute a scaling factor. The normalized shape SDF is then obtained by dividing the first layer weight matrix of the first SDF auxiliary network by this scaling factor and multiplying the last layer weight matrix by the same scaling factor .

Our method infers smooth transitions between shapes in pre-aligned poses (Figs 5, 1) or in arbitrary poses (Fig. 6). Even if our velocity field tends to preserve the topology of the shape, we sometimes observe small blobs detaching from the main shape and reattaching later in the interpolation (Feline-Horse in Fig. 6). This is due to a local minimum of the total loss corresponding to a nonzero LSE error.

7.1. Comparisons

We compare our method with other recent interpolation methods that do not require given correspondences between the two shapes: NISE [NDSS*23] and Lipschitz MLP [LWJ*22]. We also compare our approach with an OT method, called GeomLoss [FRTG19], since OT computes a transformation without needing any landmarks between two shapes with equal volume. Let us stress out that this approach was not designed to produce morphings, but to establish correspondences between two shapes. We also compare with the INSD approach, although we do not have dense correspondences between the shapes. For this reason, we introduce a landmark-free adaptation of INSD [SCC*25] that we denote as LF-INSD (see the supplementary, for a comparison with INSD in a case where a dense set of landmark correspondences is provided as input). Other results were obtained using the authors provided implementations with the parameters provided in their code (NISE, LipMLP, GeomLoss). For the OT result, GeomLoss working between discrete distributions, we sample points in the source and target volumes and use GeomLoss between these two distributions. Interpolated volumes are then recovered through kernel density estimation and density thresholding. For INSD, the authors provided code did not provide satisfactory results when removing the landmark constraint, therefore we used our re-implementation.

Figures 7 and 8 compare the various methods on the Kitten to Fandisk and Spot to Bob experiments, showing that OT still produces tearings, LipMLP and LF-INSD produce oversmoothed shapes, while NISE produces blobby intermediate shapes. See also the supplementary file and video for additional comparisons.

7.2. Quantitative evaluation

We assess our method in terms of volume preservation and compare it numerically to other methods. Figure 10 and Tab. 9 shows the mass variation across time for various experiments.

As pointed out by our experiments, our method is more efficient to preserve the volume compared to the other methods. The metrics used are the variance of the volume over time (first row) and the maximal volume variation (second row).

Since we perform an arbitrary thresholding to reconstruct the surface for the OT method, the volume of the reconstructed surface is biased by this threshold. Our kernel estimation and density thresholding used for volume computation, combined with the tendency of OT to tear the shape and thus increase the surface area, yields an increase in measured volume. It is also important to note that marginal probability density preservation of a sum of Dirac masses is not equivalent to volume preservation (as computed by kernel density estimation) throughout the deformation. For the methods with no divergence constraint, NISE and LipMLP, Fig. 10 and Tab. 9 show that the volume exhibits huge variations. This often translates visually into the disappearance of key parts of the intermediate shapes, such as the legs of an animal for example (feline-horse on Fig. 5). In some cases, the whole shape disappears in the middle of the interpolation (See the cylinders to torus experiment in the supplementary). Our method and LF-INSD perform better (Figure 10-left) than NISE or LipMLP. But INSD volume preservation constraint is not as strong as ours, and often fails to preserve the volume in more complex cases (Figure 10-right, Tab. 9). Our method on the other hand exhibits consistently almost constant volumes throughout the shape interpolation (Figure 9). In the supplementary, we further analyze the different loss terms and their impact in terms of gradient norm and LSE error.

Figure 11 shows the LSE and Eikonal errors at the time and place where a change of topology is occuring (two cylinders are

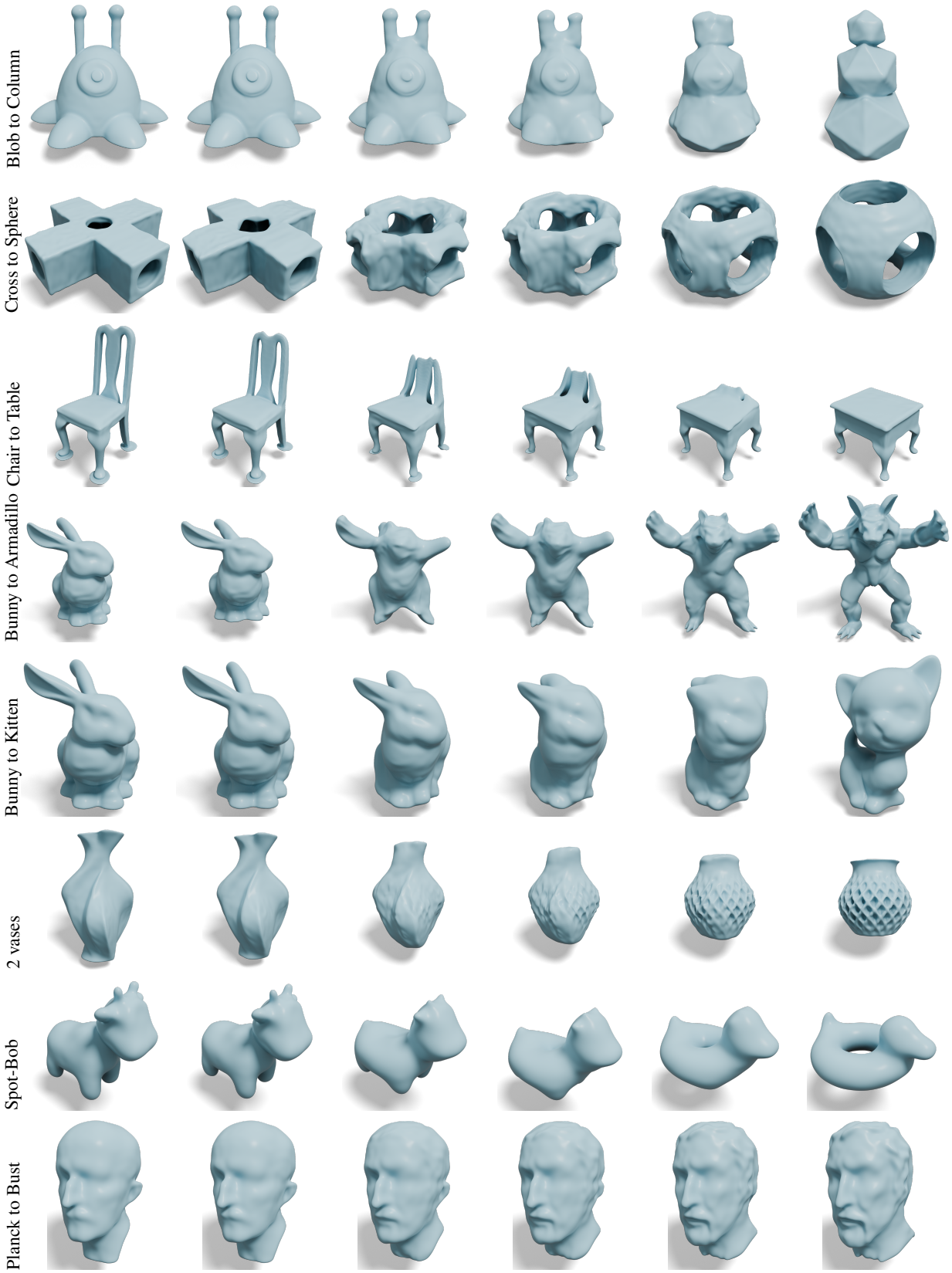


Figure 5: *Deformation results for some standard shapes.*

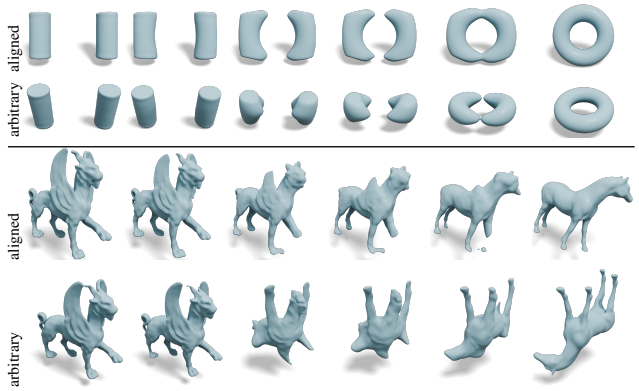


Figure 6: *Importance of the source and target relative poses. Morphings obtained from aligned poses or arbitrary poses for the cylinders to torus and feline to horse experiments.*

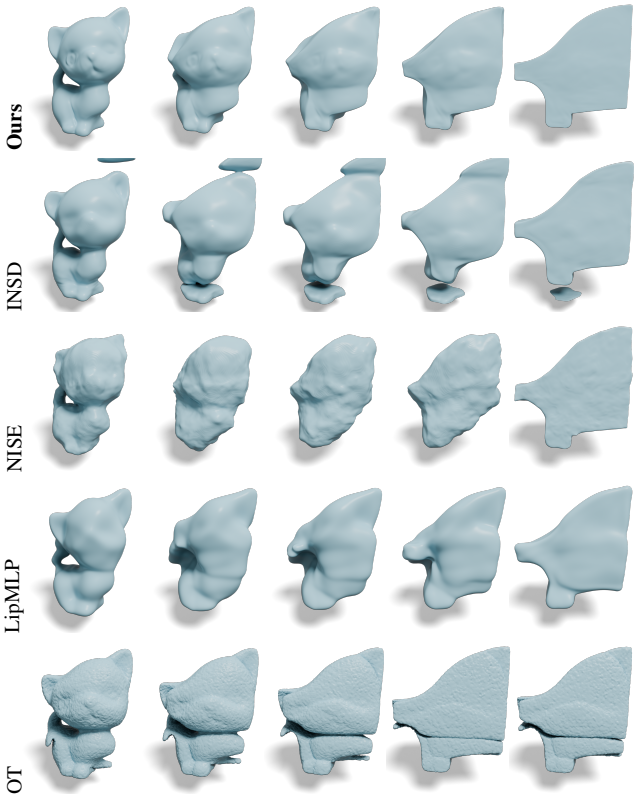


Figure 7: *Comparison of the Kitten to Fandisk morphing using various methods (Ours, INSD, NISE, LipMLP and OT). LF-INSD and LipMLP produce oversmoothed shapes, NISE produces blobby intermediate shapes, and OT tears some parts of the shape, something which is typical for this family of methods.* $t = 0.2, 0.4, 0.5, 0.6, 0.8$. *See Fig. 10 for the volume evolution for all these methods.*

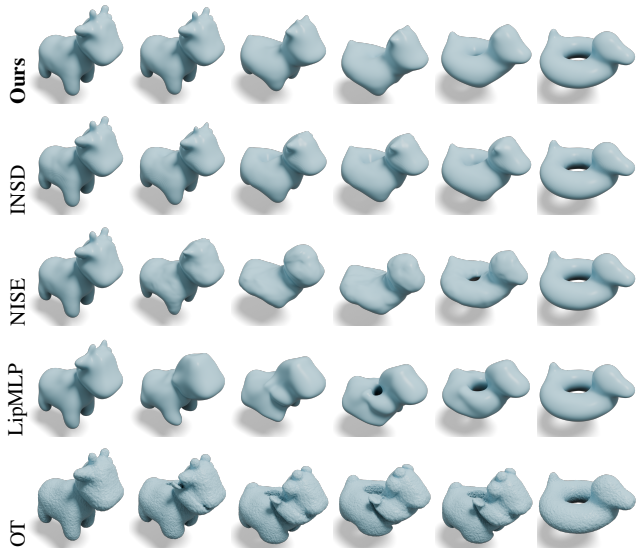


Figure 8: *Comparison on Spot-Bob. The tearings produced by OT are clearly visible. AdaDiv, LF-INSD, NISE and LipMLP all produce smooth and natural transitions. However the volume is not preserved similarly by all these methods (see Figure 10).*

		AdaDiv	NISE	LF-INSD	LipMLP	OT
Variance **x500**	feline-horse	**0.016**	0.080	0.063	0.040	0.038
	kitten-fandisk	**0.0041**	0.076	0.014	0.060	0.14
	spot-bob	**0.029**	0.11	0.034	0.48	0.093
	cylinders-tore	**0.0026**	0.098	0.040	0.073	0.042
Maximum	feline-horse	**0.018**	0.60	0.36	0.28	0.093
	kitten-fandisk	**0.0014**	0.45	0.013	0.24	1.1
	spot-bob	0.051	0.55	**0.034**	14	0.77
	cylinders-tore	**0.00026**	0.68	0.096	0.54	0.098

Figure 9: *Volume preservation in terms of variance and maximum volume variation over time (lowest value in bold).*

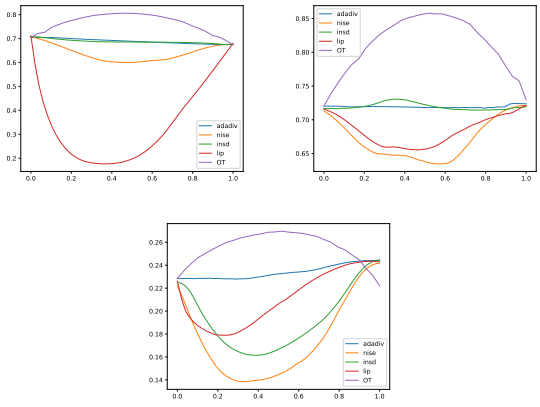


Figure 10: *Evolution of the volume of the surface w.r.t. time for spot to bob (left) and the kitten to fandisk (middle) and feline to horse (right) experiments.*

merging into one torus, see top row of Figure 6), to illustrate how local non-compliance with the LSE and non-compliance with the Eikonal equation enable this topological change, even though the velocity field has adaptive divergence w.r.t. the initial shape.

We can observe that the norm of the gradient of f_θ vanishes on the medial axis, this is due to our smooth approximation of the signed distance field and the change of gradient orientation when crossing the medial axis. The gradient also vanishes at the location of the topological change. This is consistent with the fact that the surface normal (and therefore the gradient of f_θ) locally switches from one direction to another where and when two parts of the shape are merging. We also observe that the isocontours of f_θ are tighter in this area, in contrast to the good properties imposed by the Eikonal equation away from the skeleton in other experiments without topology change.

In the spatial distribution of the LSE error, we can observe that the LSE equation is also not fulfilled at the location of the topological change, as expected. On the other hand, we can see that the error on the LSE is very small on the medial axis, which can be explained by the fact that the gradient of f_θ degenerates in this area.

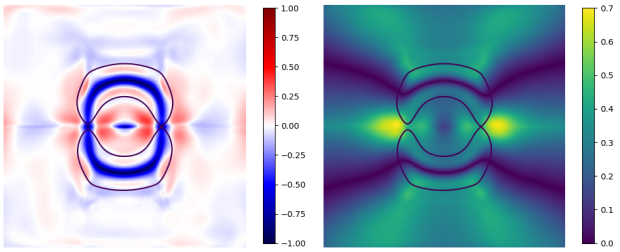


Figure 11: *Eikonal and LSE losses during a change of topological genus with our method (2D slice of a 3D morphing) with the contour of the shape superimposed in black.*

7.3. Computation time

Table 12 reports our computation times compared to other standard methods on a NVIDIA RTX 4070. The time scales with the size of the input point set as highlighted in Table 12 for the kitten-fandisk experiment (4M input points) compared to the cylinders-torus (73K input points). Our unoptimized implementation is more demanding than NISE for example but less than our implementation of LF-INSD. For the OT method, the size of the input point set has no effect since a fixed number of points (2M) is sampled inside the shape, leading to roughly the same computational time for all experiments. Note that the version of OT we used (GeomLoss) is extremely fast. However, we only count the matching time and not the implicit surface reconstruction time, which makes it less comparable to other methods. The computation time of LF-INSD is obtained using our own implementation, which produced much better results for the landmark-free morphing. Note that it is much higher than the computation time reported in the [SCC*25] paper (JAX implementation) for the landmark-based morphing. In the supplementary, we show that when stopping after 1h (the convergence time for nise) we already get a correct morphing, even if details are smoothed out.

	AdaDiv	NISE	LF-INSD	LipMLP	OT
cylinders-torus	12min26s	7min29s	2h46min	4min12s	44.33s
kitten-fandisk	11h47min	7h10min	6d15h55min	4h17min	45.36s
kitten-fandisk(lr)	2h35min	43min19s	19h6min	24min4s	44.12s
spot-bob	4h52min	48min32s	1d6h30min	45min66s	42.78s
max-bust	2h35min	32min20s	22h47min	23min83s	46.51s

Figure 12: *Computation times for 2000 epochs on a NVIDIA RTX 4070. lr means* low resolution*: 200k points instead of 2M points for the full resolution.*

7.4. Adaptive divergence with volume change

While our adaptive divergence is natural when morphing between shapes with the exact same volume, we show in this section that it can also be useful in the non constant volume setting, by choosing a smooth parameterization of the volume change through time.

We start by normalizing the shapes by computing their volumes V_0 and V_1 based on their SDF representations. For simplicity assume that $\mathcal{S}_0$ is much smaller than $\mathcal{S}_1$. This yields a scaling factor α for $\mathcal{S}_1$ and the normalized SDF writes:

$$\tilde{g}_1(x) = \alpha g_1(\frac{1}{\alpha}x)\ \forall x \in \mathbb{R}^d \tag{15}$$

Then, both shapes having equal volume, we can apply our adaptive divergence method. The last step combines the resulting constant volume morphing with a volume change parameterized over time. Let us consider a volume change $\alpha(t)$ such that for $i \in \{0,1\}, Vol(\alpha(i) f(\frac{x}{\alpha(i)}, i)) = V_i$, then the combined morphing can be written as:

$$\tilde{f}(x,t) = \alpha(t) f(\frac{x}{\alpha(t)}, t) \tag{16}$$

Figure 13 shows an example of such a volume change and compares it with NISE and an ablated version of our method removing any constraint on $\mathbf{V}$.

8. Limitations

Our method has some limitations. First, while the shape structures are well deformed and represented, the details tend to be smoothed out. This is due to the many constraints on the implicit representation, which tend to prevail over the Dirichlet and Neumann attachment terms to the source and target surfaces. A workaround would be to combine it with implicit displacement fields [YRSh22], a multiresolution approach encoding the detail field explicitly. But, this detail layer would need to be deformed which is a nontrivial adaptation. Another limitation is the computation time (see the supplementary) of our unoptimized research code, which is due to the necessary first order differentiation of $\mathbf{V}_\theta$ at inference and the 2nd order differentiation when backpropagating.

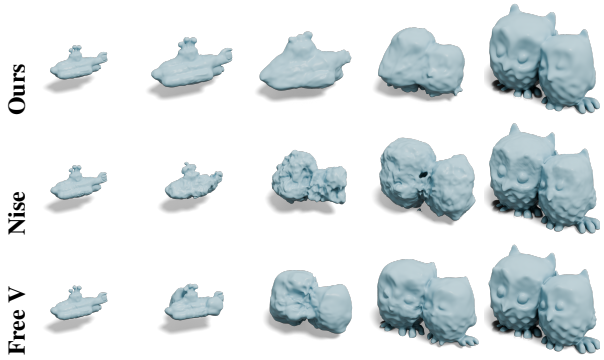


Figure 13: *Morphing between two shapes of different volumes. The bottom row shows an ablated version where* **V** *is not constrained.*

9. Conclusion

We introduced a new method for morphing shapes without any landmark correspondence, favoring volume and topology preservation. This is due to our new type of velocity field with interesting theoretical properties including the volume preservation property for a specific set, without requiring the velocity field to be divergence free everywhere. We showed on several examples that it produces smooth natural interpolations between unrelated shapes. One of the most promising future research direction is to study how the offset surfaces are preserved by our adaptive velocity field, an interesting property for advecting information around the surface, which could lead to detail transportation.

Acknowledgements This work was partially funded by ANR-23-PEIA-0004 (PDE-AI). Work under CC-BY 4.0 license https://creativecommons.org/licenses/by/4.0/.

References

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques* (USA, 2000), SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., p. 157–164. 2

[ACWK06] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-Sweepers: Constant Volume Modeling. *Graphical Models 68*, 4 (2006), 324–332. 4

[ACZW19] AHARON I., CHEN R., ZORIN D., WEBER O.: Bounded distortion tetrahedral metric interpolation. *ACM Transactions on Graphics 38* (11 2019), 1–17. 2

[ANVL21] ATZMON M., NOVOTNY D., VEDALDI A., LIPMAN Y.: Augmenting implicit neural shape representations with explicit deformation fields. *arXiv preprint arXiv:2108.08931* (2021). 3

[BB00] BENAMOU J.-D., BRENIER Y.: A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik 84*, 3 (2000), 375–393. 2

[BCOS01] BERTALMIO M., CHENG L.-T., OSHER S., SAPIRO G.: Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics 174*, 2 (2001), 759–780. 2

[BRV*24] BERZINS A., RADLER A., VOLKMANN E., SANOKOWSKI S., HOCHREITER S., BRANDSTETTER J.: Geometry-informed neural networks. *arXiv preprint arXiv:2402.14009* (2024). 2

[CB24] COIFFIER G., BETHUNE L.: 1-lipschitz neural distance fields, 2024. 2

[COSL98] COHEN-OR D., SOLOMOVIC A., LEVIN D.: Three-dimensional distance field metamorphosis. *ACM Trans. Graph. 17*, 2 (Apr. 1998), 116–141. 2

[CZ19] CHEN Z., ZHANG H.: Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 5939–5948. 3

[DC98] DESBRUN M., CANI M.-P.: Active Implicit Surface for Animation. In *Graphics Interface* (Vancouver, Canada, June 1998), the Canadian Human-Computer Communications Society, pp. 143–150. Published under the name Marie-Paule Cani-Gascuel. 2

[ELC18] EISENBERGER M., LÄHNER Z., CREMERS D.: Divergence-free shape interpolation and correspondence, 2018. 2

[FCVP17] FEYDY J., CHARLIER B., VIALARD F.-X., PEYRÉ G.: Optimal transport for diffeomorphic registration. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20* (2017), Springer, pp. 291–299. 2

[FRTG19] FEYDY J., ROUSSILLON P., TROUVÉ A., GORI P.: Fast and scalable optimal transport for brain tractograms. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22* (2019), Springer, pp. 636–644. 7

[GYH*20] GROPP A., YARIV L., HAIM N., ATZMON M., LIPMAN Y.: Implicit geometric regularization for learning shapes, 2020. 2, 6

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *Computer Graphics Forum 27*, 5 (2008), 1449–1457. 2

[Lév15] LÉVY B.: A numerical algorithm for *l_{2}* semi-discrete optimal transport in 3d. *ESAIM: Mathematical Modelling and Numerical Analysis 49*, 6 (2015), 1693–1715. 2

[LWJ*22] LIU H.-T. D., WILLIAMS F., JACOBSON A., FIDLER S., LITANY O.: Learning smooth neural functions via lipschitz regularization. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. 3, 7

[MCR22] MEHTA I., CHANDRAKER M., RAMAMOORTHI R.: A Level Set Theory for Neural Implicit Evolution under Explicit Flows, July 2022. arXiv:2204.07159 [cs]. 3

[MON*19] MESCHEDER L., OECHSLE M., NIEMEYER M., NOWOZIN S., GEIGER A.: Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 4460–4470. 2

[NDSS*23] NOVELLO T., DA SILVA V., SCHARDONG G., SCHIRMER L., LOPES H., VELHO L.: Neural implicit surface evolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 14279–14289. 3, 6, 7

[OF04] OSHER S., FEDKIW R.: *The Level Set Methods and Dynamic Implicit Surfaces*, vol. 57. 05 2004, pp. xiv+273. 2

[PFS*19] PARK J. J., FLORENCE P., STRAUB J., NEWCOMBE R., LOVEGROVE S.: Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 165–174. 2

[RPLC22] RICHTER-POWELL J., LIPMAN Y., CHEN R. T.: Neural conservation laws: A divergence-free perspective. *Advances in Neural Information Processing Systems 35* (2022), 38075–38088. 3, 4

[SCC*25] SANG L., CANFES Z., CAO D., BERNARD F., CREMERS D.: Implicit neural surface deformation with explicit velocity fields. In *The Thirteenth International Conference on Learning Representations* (2025). 3, 4, 5, 7, 10

[SMB*20] SITZMANN V., MARTEL J., BERGMAN A., LINDELL D., WETZSTEIN G.: Implicit neural representations with periodic activation functions. *Advances in neural information processing systems 33* (2020), 7462–7473. 2, 6

[Tho95] THOMAS F.: *The Illusion of Life: Disney Animation*. Disney Editions Deluxe (Film). Disney Editions, 1995. 1

[TO99] TURK G., O'BRIEN J. F.: Shape transformation using variational implicit functions. In *Proceedings of ACM SIGGRAPH 1999* (Aug. 1999), pp. 335–342. 2

[TSB16] TAO M., SOLOMON J., BUTSCHER A.: Near-isometric level set tracking. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 65–77. 2

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Trans. Graph. 25*, 3 (July 2006), 1118–1125. 4

[VTSSH15] VON-TYCOWICZ C., SCHULZ C., SEIDEL H.-P., HILDEBRANDT K.: Real-time nonlinear shape interpolation. *ACM Trans. Graph. 34*, 3 (May 2015). 2

[WWY*25] WANG Z., WANG C., YOSHINO T., TAO S., FU Z., LI T.-M.: Hotspot: Signed distance function optimization with an asymptotically sufficient condition. In *CVPR* (2025). 2

[XTS*22] XIE Y., TAKIKAWA T., SAITO S., LITANY O., YAN S., KHAN N., TOMBARI F., TOMPKIN J., SITZMANN V., SRIDHAR S.: Neural fields in visual computing and beyond. *Computer Graphics Forum 41*, 2 (2022), 641–676. 2

[YBHK21] YANG G., BELONGIE S., HARIHARAN B., KOLTUN V.: Geometry processing with neural fields. In *Advances in Neural Information Processing Systems* (2021), Ranzato M., Beygelzimer A., Dauphin Y., Liang P., Vaughan J. W., (Eds.), vol. 34, Curran Associates, Inc., pp. 22483–22497. 2

[YRSh22] YIFAN W., RAHMANN L., SORKINE-HORNUNG O.: Geometry-consistent neural shape representation with implicit displacement fields. In *International Conference on Learning Representations* (2022). 10

[ZJ16] ZHOU Q., JACOBSON A.: Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797* (2016). 7

Appendix A: Proof of the divergence free construction in dimension 2d

Let $A_{sks} \in \mathbb{R}^{2d\times 2d}$ be a constant skew symmetric matrix and $\mathcal{H}$ a smooth scalar potential $\mathcal{H} : \mathbb{R}^{2d} \longrightarrow \mathbb{R}$. To prove that $A_{sks}\nabla\mathcal{H}$ is divergence free, we have to show that $Tr(\mathcal{J}_{A_{sks}\nabla\mathcal{H}}) = 0$, (with $\mathcal{J}$ the Jacobian operator). We will use the following properties:

1. $\mathcal{J}_{\nabla\mathcal{H}}$ the Hessian of $\mathcal{H}$ is a symmetric matrix if $\mathcal{H}$ is smooth: $\mathcal{J}_{\nabla\mathcal{H}} = \mathcal{J}^T_{\nabla\mathcal{H}}$
2. $Tr(A^T) = Tr(A)$, $Tr(AB) = Tr(BA)$, $Tr(-A) = -Tr(A)$

Using properties (1) and (2) we can write:

$$\begin{aligned} Tr(\mathcal{J}_{A_{sks}\nabla\mathcal{H}}) &= Tr(A_{sks}\mathcal{J}_{\nabla\mathcal{H}}) \\ &= Tr((\mathcal{J}^T_{\nabla\mathcal{H}}A^T_{sks})^T) \\ &\underset{\text{using (1)}}{=} Tr(-(\mathcal{J}_{\nabla\mathcal{H}}A_{sks})^T) \\ &\underset{\text{using (2)}}{=} -Tr(A_{sks}\mathcal{J}_{\nabla\mathcal{H}}) \\ &= -Tr(\mathcal{J}_{A_{sks}\nabla\mathcal{H}}) = 0 \end{aligned} \tag{17}$$

which proves that $A_{sks}\nabla\mathcal{H}$ is divergence free.

Appendix B: Proof of Theorem 1 - Volume Preservation

For this proof we need the following lemma:

Lemma 1 Let $\mathbf{V}$ be a vector field with adaptive divergence with respect to a set $\mathcal{S}$, and $\phi_{\mathbf{V}}$ its associated flow. Let $\mathbf{W}$ be the corresponding divergence free vector field as in the adaptive divergence definition, and $\phi_{\mathbf{W}}$ its associated flow. Then $\phi_{\mathbf{V}}$ coincides with $\phi_{\mathbf{W}}$ on $\partial\mathcal{S}$. Moreover, $\mathbf{V}$ coincides with $\mathbf{W}$ on the trajectories of $\partial\mathcal{S}$ by $\phi_{\mathbf{W}}$ (or equivalently $\phi_{\mathbf{V}}$)

Proof By definition of a flow, for $y_0 \in \mathbb{R}^d$, $\phi_{\mathbf{W}}(y_0,t)$ is a solution of the system:

$$\begin{cases} \frac{\partial y}{\partial t} = \mathbf{W}(y,t) \\ y(0) = y_0 \end{cases} \tag{18}$$

and $\phi_{\mathbf{V}}(y_0,t)$ is a solution of the system:

$$\begin{cases} \frac{\partial y}{\partial t} = \mathbf{V}(y,t) \\ y(0) = y_0 \end{cases} \tag{19}$$

By definition of the adaptive divergence, there exists a function $\mathbf{F}(x,t)$ such that $\mathbf{V}(x,t) = \mathbf{W}(x,t) + \mathbf{F}(x,t)$; and if $y_0 \in \partial\mathcal{S}$, then $F(\phi_{\mathbf{W}}(y_0,t),t) = 0$. Thus,

$$\partial_t\phi_{\mathbf{W}}(y_0,t) = \mathbf{W}(\phi_{\mathbf{W}}(y_0,t),t) = \mathbf{W}(\phi_{\mathbf{W}}(y_0,t),t) + \mathbf{F}(\phi_{\mathbf{W}}(y_0,t),t) \tag{20}$$

Thus, if $y_0 \in \partial\mathcal{S}$, $\phi_{\mathbf{W}}(y_0,t)$ is also a solution of Equation 19.
By unicity of the solution, the Cauchy-Lipschitz theorem ensures that $\phi_{\mathbf{W}}(y_0,t) = \phi_{\mathbf{V}}(y_0,t)\ \forall y_0 \in \partial\mathcal{S}$. It follows directly that $\mathbf{W}(\phi_{\mathbf{W}}(y_0,t),t) = \mathbf{V}(\phi_{\mathbf{W}}(y_0,t),t) \forall y_0 \in \partial\mathcal{S}$.
□

We now turn to the full proof of theorem 1.

Proof
Recalling that $\mathrm{d}_t \det(\mathcal{J}_{\phi_{\mathbf{W}}}) = \det(\mathcal{J}_{\phi_{\mathbf{W}}})\mathrm{div}(\mathbf{W})$, we can obtain the well known result linking the volume variation of $\phi_{\mathbf{W}}(\mathcal{S},t)$ and the divergence of $\mathbf{W}$:

$$\mathrm{d}_t Vol(\phi_{\mathbf{W}}(\mathcal{S},t)) = \int_{\phi_{\mathbf{W}}(\mathcal{S},t)} \mathrm{div}(\mathbf{W}(x,t))\mathrm{d}x \tag{21}$$

Moreover, since $\mathbf{W}$ is divergence-free, $\phi_{\mathbf{W}}$ is volume preserving:

$$0 = \mathrm{d}_t Vol(\phi_{\mathbf{W}}(\mathcal{S},t)) = \int_{\phi_{\mathbf{W}}(\mathcal{S},t)} \mathrm{div}(\mathbf{W}(x,t))\mathrm{d}x \tag{22}$$

We then use the divergence theorem on $\phi_{\mathbf{W}}(\mathcal{S},t)$ and the fact that $\mathbf{V}$ coincides with $\mathbf{W}$ on $\phi_{\mathbf{W}}(\mathcal{S},t)$:

$$\begin{aligned} \int_{\phi_{\mathbf{W}}(\mathcal{S})} \mathrm{div}(\mathbf{W})(x,t)\mathrm{d}x &= \int_{\phi_{\mathbf{W}}(\partial\mathcal{S},t)} \mathbf{W}(x,t)\cdot\overrightarrow{n}_{\partial\mathcal{S}} = \int_{\phi_{\mathbf{W}}(\partial\mathcal{S},t)} \mathbf{V}(x,t)\cdot\overrightarrow{n}_{\partial\mathcal{S}} \\ &= \int_{\phi_{\mathbf{W}}(\mathcal{S})} \mathrm{div}(\mathbf{V})(x,t)\mathrm{d}x \\ &= \int_{\phi_{\mathbf{V}}(\mathcal{S})} \mathrm{div}(\mathbf{V})(x,t)\mathrm{d}x = \mathrm{d}_t Vol(\phi_{\mathbf{V}}(\mathcal{S},t)) \end{aligned} \tag{23}$$

By combining 21 and 23, we get $\mathrm{d}_t Vol(\phi_{\mathbf{V}}(\mathcal{S},t)) = 0$, which proves that the volume of $\mathcal{S}$ is preserved through $\phi_{\mathbf{V}}$. □