Neural Lagrangian Mesh: A Generative Model for Manifold Meshes of Arbitrary Genus

Adrish Dey [1] Edward Chien [1]

[1]Boston University

BOSTON UNIVERSITY

Introduction

We explore the use of Lagrangian mesh representation to build a conditional generative model for manifold meshes of arbitrary genus, conditioned on shape descriptors (currently images).

We introduce a conditional flow-matching framework to generate the particles and a neural convex solver based on input features. Next, we treat the object as a collection of "volumetric parcels," defined as convex polytopes within a bounded continuous domain Ω, allowing us to leverage OT machinery. We also explore a Neural Convex solver for faster solutions. Finally, we employ skin-surface meshing (Edelsbrunner '99) to obtain the manifold mesh.

Flow Matching for Particle Generation

Given a conditional feature vector z_j, we aim to generate a set of samples $\{y_i\}_{[N]}$ of the target shape $\mathcal{O}_j$. To achieve this, we leverage Flow Matching to parameterize a conditional integration map $\phi(x|z_j) \overset{\text{def}}{:=} \int_0^1 v_\theta(x, z_j, t)\, dt$ acting as a pushforward between $\mathcal{N}(0,1) \to \nu$. Here, $v_\theta(x, z_j, t)$ is the neural network parameterizing the vector field, and $\mathcal{N}(0,1)$ denotes the isotropic Gaussian. Next, we design the conditional vector field $u(x|z_j)$. We follow OT-CFM by (Tong et al., 2023) to build an Optimal Transport gradient field as follows:

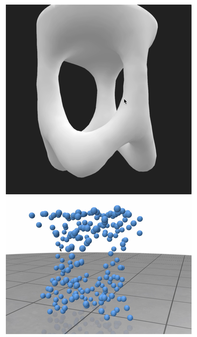


Figure 1. **Top:** Conditioning Image. **Bottom:** Generated Point Cloud

$$u(x|z_j) = x_1^* - x_0^*$$

where $(x_0^*, x_1^*) \sim \Pi(\mathbf{x}_0, \mathbf{x}_1)$ and $\mathbf{x}_0 \sim \mathcal{N}(0,1), \mathbf{x}_1 \sim P(\mathcal{O}_j)$. $\Pi(\mathbf{x}_0, \mathbf{x}_1)$ denotes the Optimal Transport coupling between the sample minibatches $\mathbf{x}_0$ and $\mathbf{x}_1$.

The final objective function is:

$$\mathcal{L}_\theta^{\text{CFM}} = \|v_\theta(x, z_j, t) - u(x|z_j)\|^2$$

Semi-Discrete Optimal Transport

Consider the continuous background $\Omega := [0,1]^3$. Let $\{y_i\}_{[N]}$ be the set of N generated particles. We equip the set of points $\{y_i\}_{[N]}$ with Dirac masses $\{\nu_i\}_{[N]}$ such that $\sum_i \nu_i < \|\Omega\|$. From the Lagrangian perspective, each point denotes the location of the volumetric parcels $\{V_i\}_{[N]}$ that constitute the mesh. Our goal is to find the volume parameter $\{\psi_i\}_{[N]}$ of the particles.

Let $\psi \overset{\text{def}}{:=} \{\psi_i\}_{[N]}$. Following (Levy, 2021), let us define V_i^ψ as the "volume parcel" with volume parameter ψ_i and the background "air" as a continuum volume of remaining mass $\|\Omega\| - \sum_i \nu_i$. **Note**: Lag_i^ψ denotes the i^{th} cell of the induced Laguerre tessellation parameterized by ψ. This gives rise to a **convex** optimization objective as follows:

$$\arg\max_{\psi \in \mathbb{R}_+^N} K(\psi) = \sum_{i\in[N]} \int_{V_i^\psi} \|x - y_i\|\, d\mu + \sum_{i\in[N]} y_i \nu_i$$

where,

$$V_i^\psi = Lag_i^\psi \cap B(y_i, \sqrt{\psi_i})$$

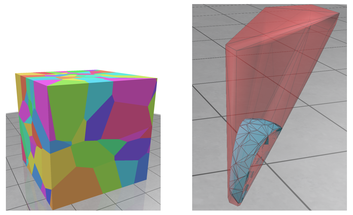


Figure 2. **Left**: Laguerre Tessellation of Domain. **Right**: (Blue) V_i^ψ cell. (Red) Laguerre Cell Lag_i^ψ

1: $\psi_i \leftarrow 0$
2: **while** $\left\|\frac{\partial K}{\partial \psi_i}\right\|^2 \geq \epsilon$ **do**
3: Compute $V_i := \text{Lag}_i^\psi \cap B\left(y_i, \sqrt{\psi_i}\right)$
4: $\frac{\partial K}{\partial \psi_i} = \nu_i - |V_i^\psi|$
5: $\psi_i \leftarrow \psi_i + \lambda \cdot \frac{\partial K}{\partial \psi_i}$
6: **end while**

Algorithm 1. Gradient Ascent to find ψ_i

Computing V_i and $|V_i|$

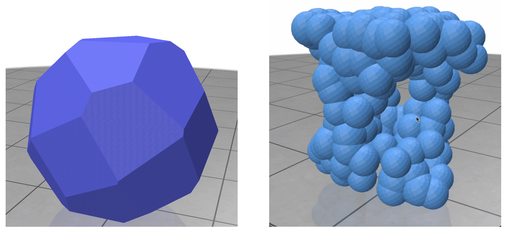


Figure 3. **Left**: Approximating $B(y_i, \sqrt{\psi_i})$ with a convex polytope of K-walls. **Right**: union of polytopes after optimization

For fast intersection of Lag_i^ψ with $B(y_i, \sqrt{\psi_i})$ we approximate each ball with a convex polytope $\hat{B}(y_i, \sqrt{\psi_i})$ with K walls. (NOTE: A large value of K is preferred for better approximation. We choose $K = 200$)

Polytope Intersection: To perform polytope intersection $V_i^\psi = Lag_i^\psi \cap \hat{B}(y, \sqrt{\psi_i})$, we first represent both Lag_i^ψ and $\hat{B}(y, \sqrt{\psi_i})$ using half spaces (i.e., H-Representation). Now, the desired polytope V_i^ψ is essentially the one which satisfies the conditions $\text{HalfSpace}(Lag_i^\psi) \cup \text{HalfSpace}(\hat{B})$. (Redundant walls are removed by using Chebychev's radius).

Polytope Volume Computation: We first convert the H-representation of V_i^ψ to V-representation. The volume of the polytope is then calculated by tetrahedralizing the convex hull of V_i^ψ.

Gradient Flows

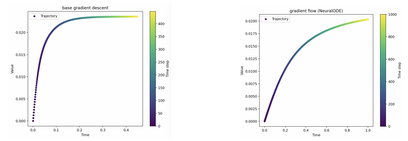


Figure 4. Gradient Flow Matching. **Left**: Value of potentials ψ_i at each discrete timestep t of gradient ascent. (Time taken: ~20min for 1000 steps) **Right**: Value of potentials obtained from learned continuous Gradient Flow.(Time taken: ~2 sec for the entire flow)

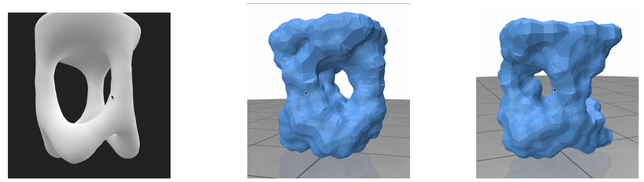
We are also interested in building a Neural Solver for the convex optimization in Algorithm 1, since solving this convex program (specifically computing the polytope V_i^ψ) is computationally expensive. To that end, we consider the gradient ascent algorithm as an Euler discretization of the gradient flow from $\psi_i = 0$ to ψ_i^{OPT}. Let T be the number of steps taken in the gradient ascent, and z_j the conditioning variable. We define a conditional integration map $\varphi_i(0|z_j) \overset{\text{def}}{:=} \int_0^T \frac{\partial K}{\partial \psi_i}[\psi_i(t)]\, dt$ that solves $\psi_i^{OPT} = \varphi(0|z_j)$ for a given shape with descriptors z_j. Now, given a parametric function $\varrho_\theta\left(\psi_i(t), z_j, t\right)$, we formulate a flow matching algorithm that approximates the solution $\{\psi_i^{OPT}\}_{[N]}$. The conditional vector field is given as follows:

$$u\left(\psi_i(t)|z_j\right) = \frac{\partial K}{\partial \psi_i}[\psi_i(t)]$$

And the flow matching loss is given as follows:

$$\mathcal{L}_\theta^{\text{CFM}} = \left\|\varrho_\theta\left(\psi_i(t), z_j, t\right) - u(\psi_i(t)|z_j)\right\|^2$$

Skinsurface Meshing (Edelsbrunner'99, Kruithofand et.al. '07)



(a) Ground Truth (b) Left View (Generated) (c) Right View (Generated)

Figure 5. Skin surface meshing provides an efficient way to extract manifold surface mesh of arbitrary topology from a union of balls system.

Looking for Input

• What are the advantages of a Lagrangian representation? Does the flow-based formulation allow for any correspondence between various generated geometries? • Can we combine these ideas with Gaussian splat representations? They are surface representations, while ours are volumetric. • Can we combine these ideas with diffusion models for other generative purposes? Perhaps with branching to allow for adaptive density for more/less geometric detail.

adrishd@bu.edu