Volume Rendering of Neural Implicit Surfaces (VolSDF)

NeurIPS 2021, Oral

Lior Yariv, Jiatao Gu, Yoni Kasten, Yaron Lipman

Weizmann Institute of Science,
Facebook AI Research

Outline

- `Introduction`
  - Related work
  - Motivation
  - Overview of VolSDF and its contribution
- `Method`
  - Method overview
  - Volume rendering
  - Volume density as transformed SDF
  - Bound on the opacity approximation error
  - Sampling algorithm
- `Evaluation`
  - Result

Related work

- IDR (Multi-view Neural Surface Reconstruction by Disentangling Geometry and Appearance) L. (L. Yariv et al, Neural IPS 2020 Spotlight)
  - Represent the geometry as the zero level set of a neural network $S_\theta = \{x \in R^3 \mid f(x;\theta) = 0\}$
  - Opt for f to model a signed distance function (SDF) to its zero level set S_θ
  - Estimate camera and 3D geometry/appearance jointly
  - 2D supervision require per-pixel masks

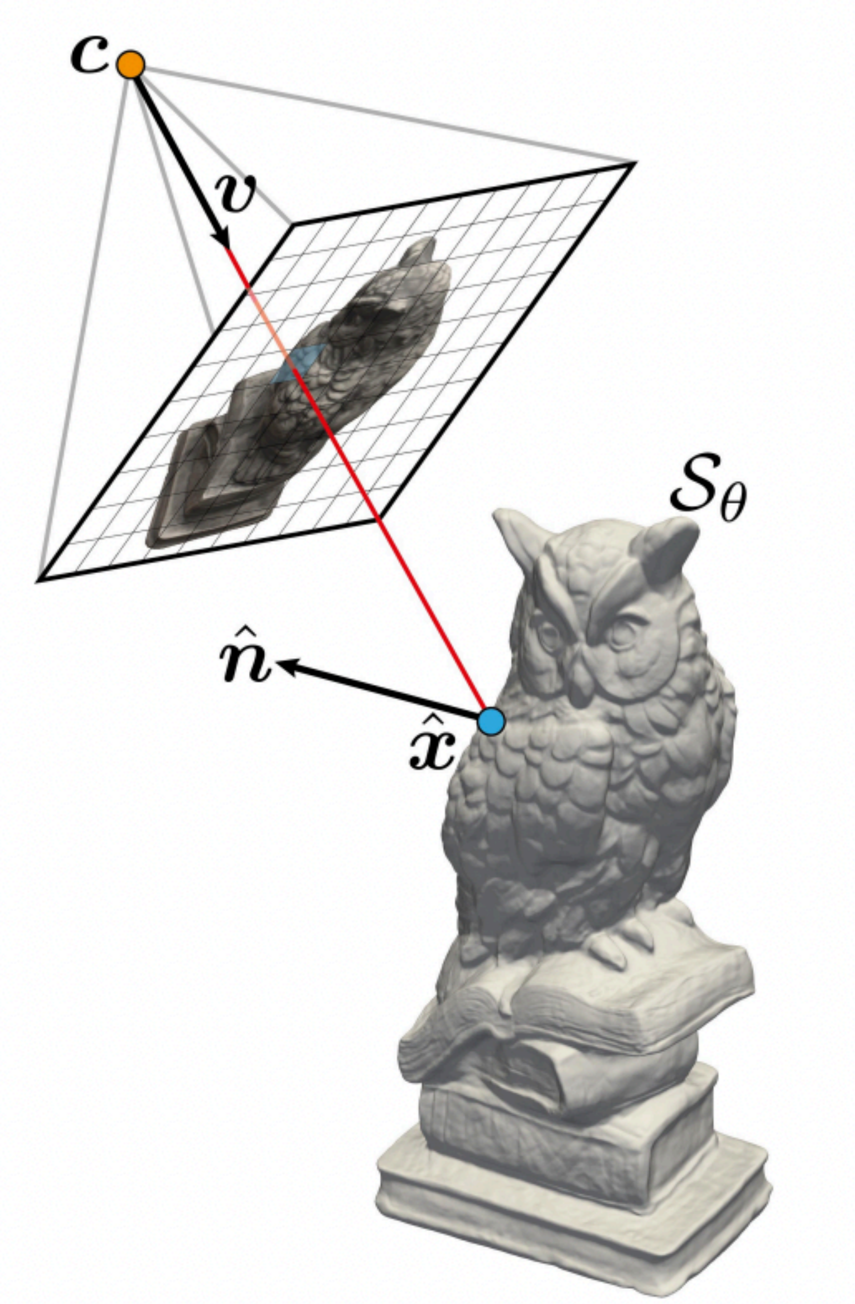
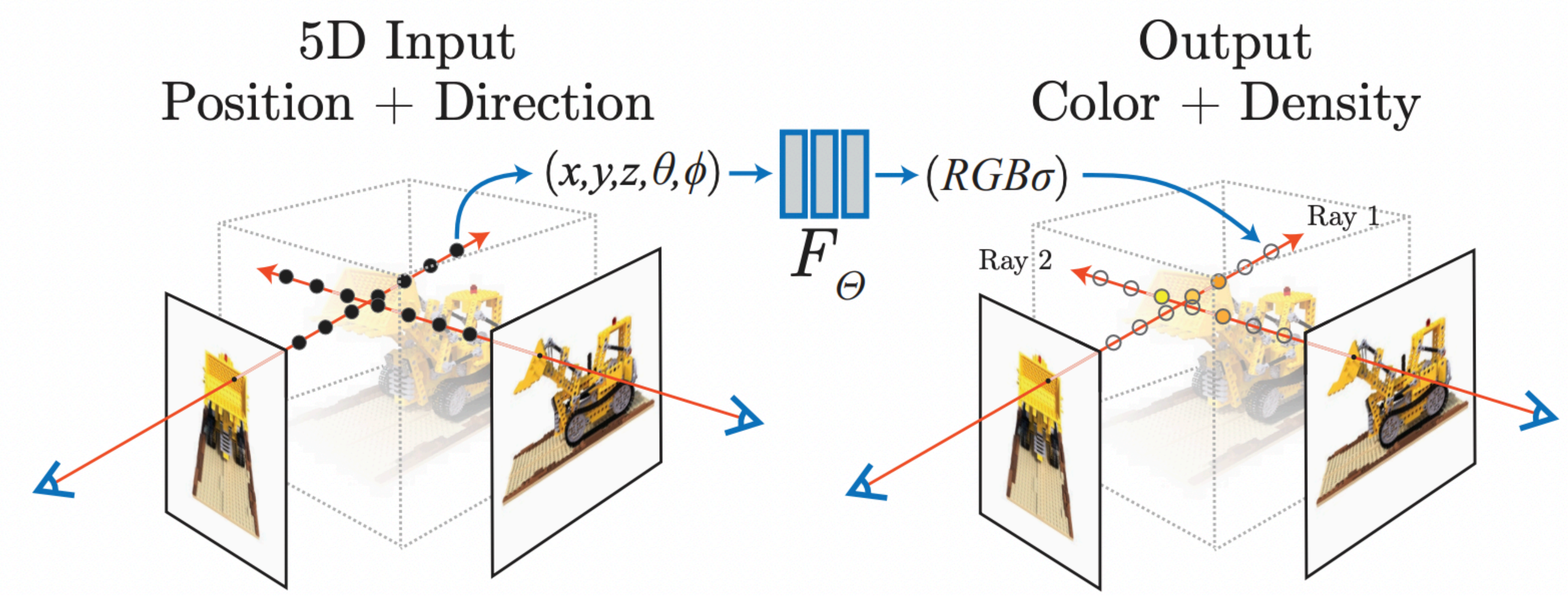


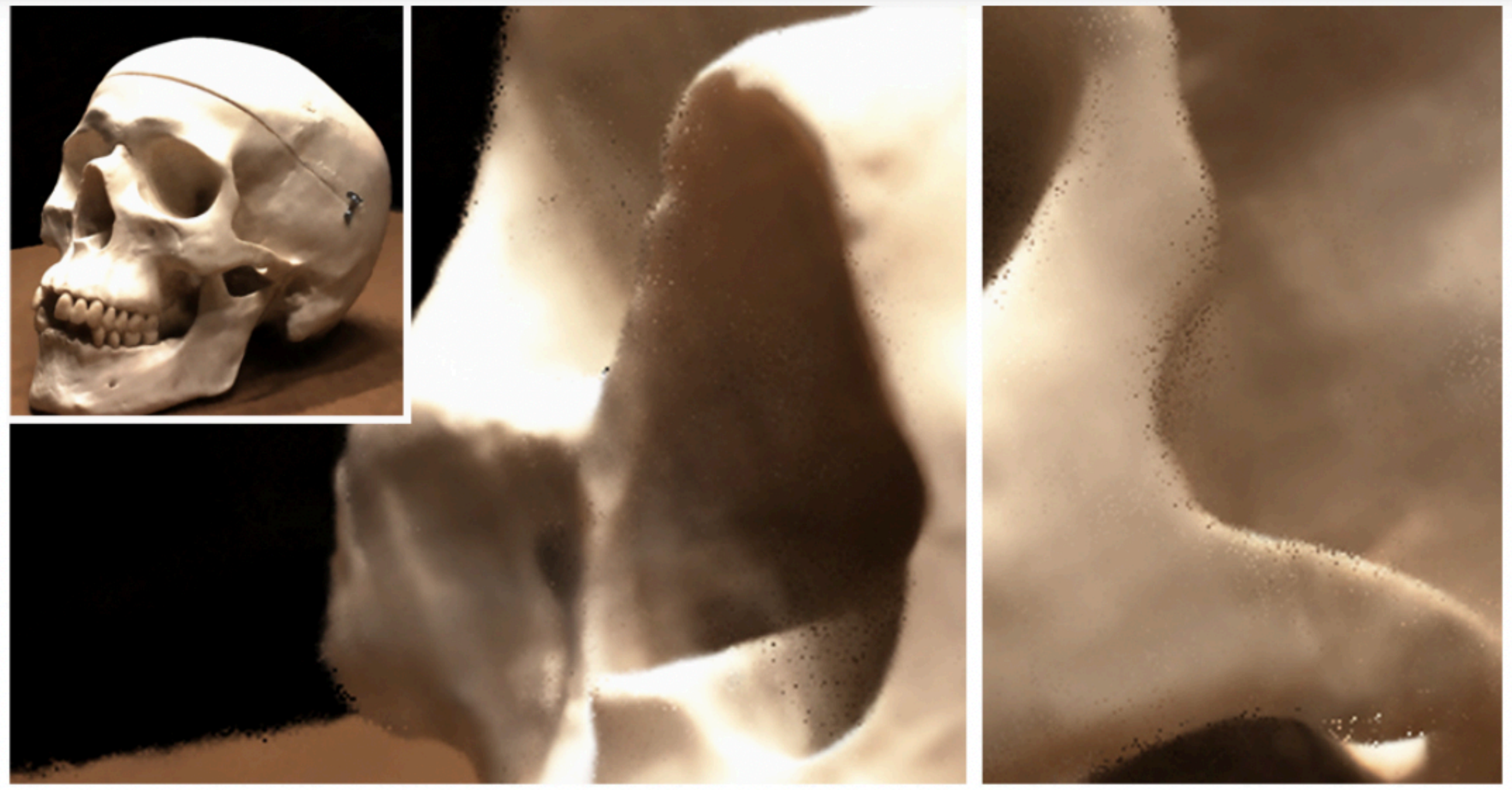
Figure 2: Notations.

- NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis (B. Mildenhall ECCV 2020 Best Paper)
  - Represent the geometry as volumetric density through MLP
  - Using volume rendering techniques to render novel views

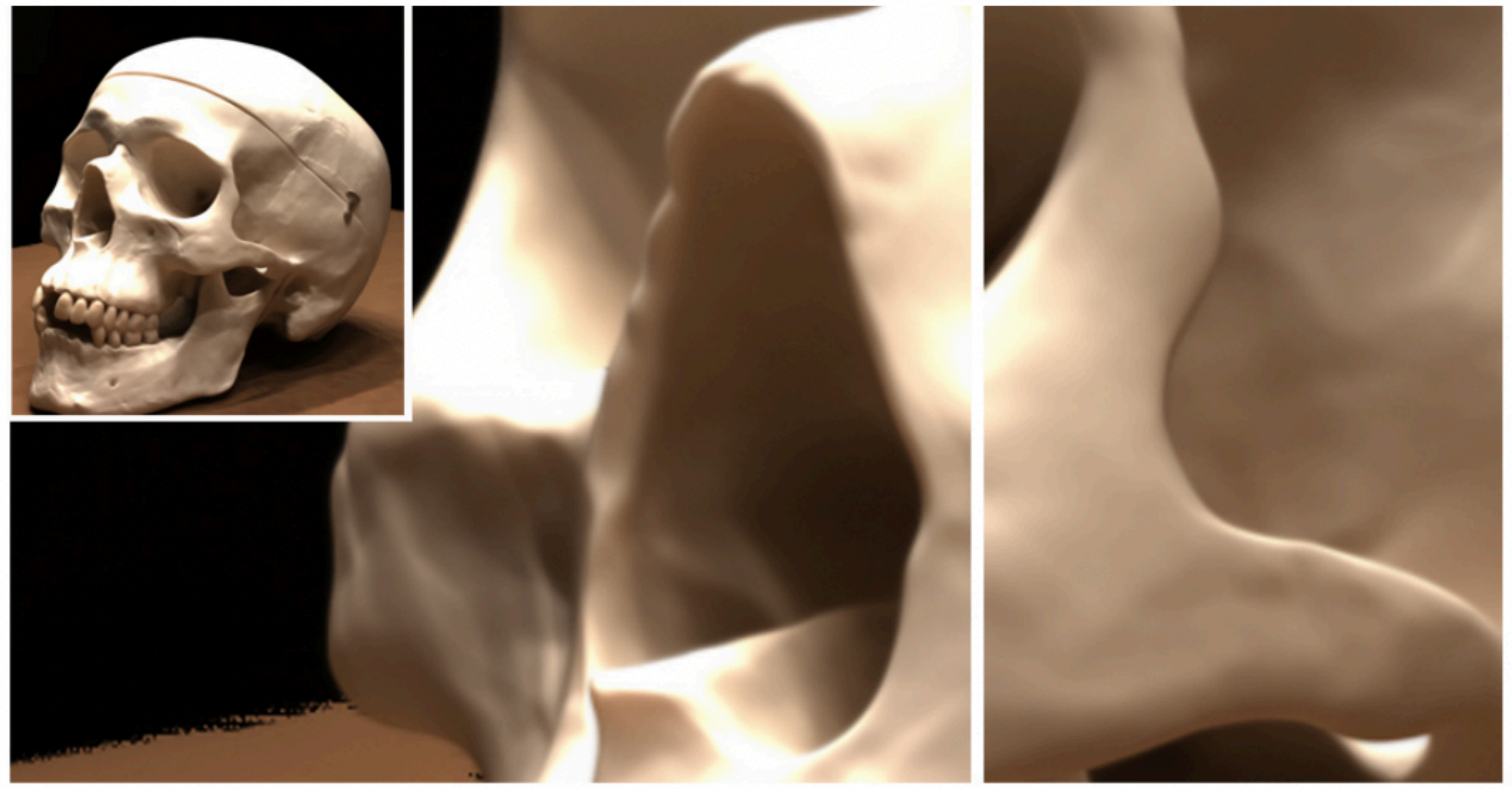


Motivation

- Neural volume rendering gains popular due to its recent success in synthesizing novel views
- Geometry learned by neural volume rendering techniques was modeled with a generic density function (MLP_θ in NeRF)
  - → Geometry is extracted using an arbitrary level set of the density function resulting in noisy, and low fidelity reconstruction
- Adaptive approximation of opacity adopted by NeRF lead to a sub-optimal sampling



NeRF

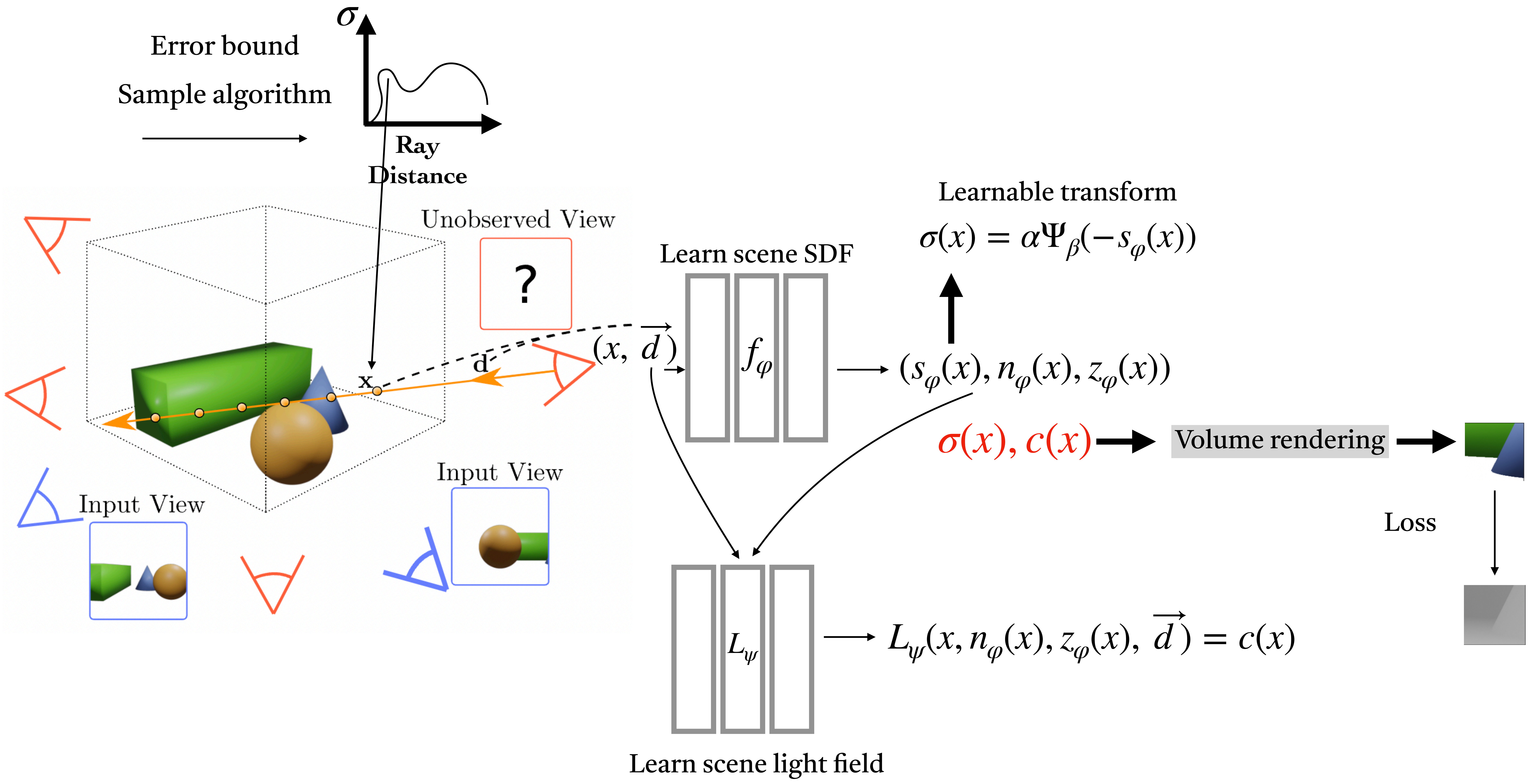


VolSDF

Overview of VolSDF

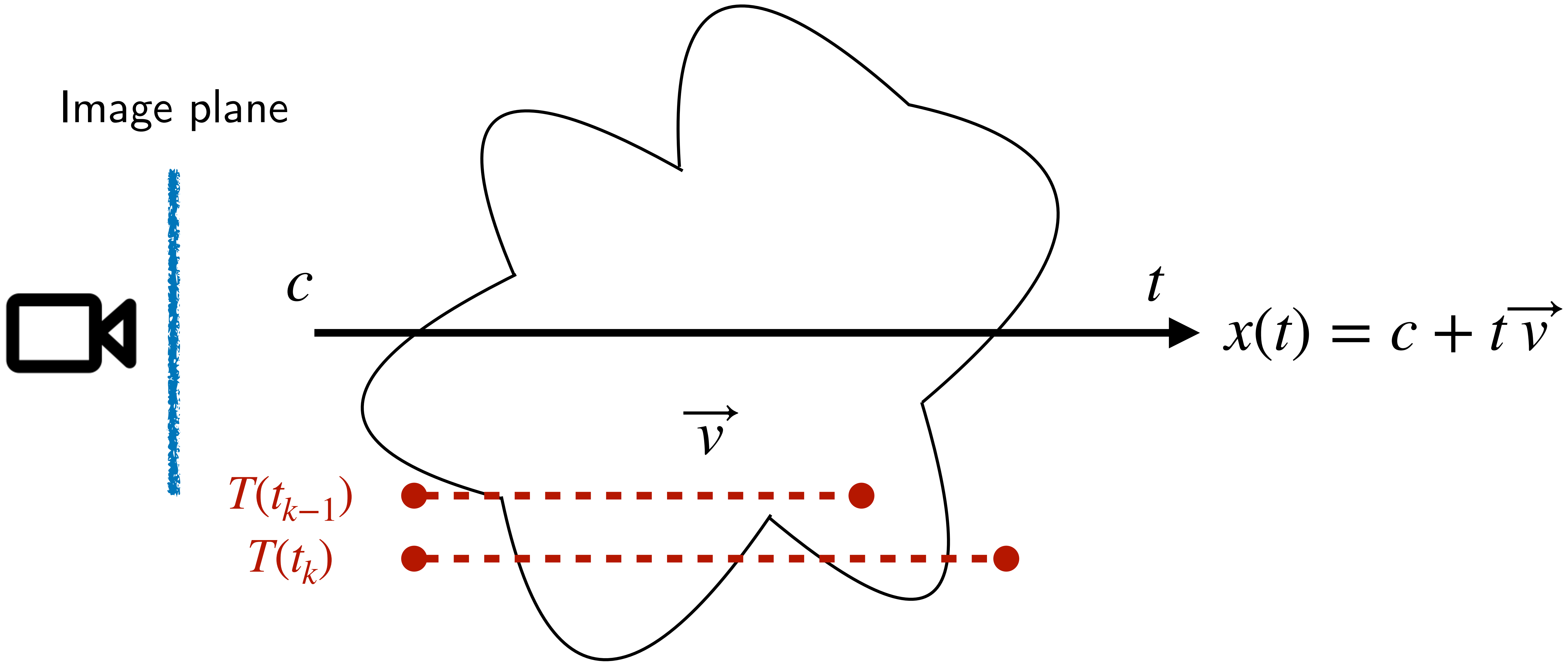
- **Overview**
  - A volume rendering framework for implicit neural surface
- **Contributions**
  - Bridging and get the best of two different fields: Volume rendering & Neural implicit surfaces
  - Propose a way to model the volume density as a function of the geometry
    - Contrast to previous works where the geometry was modeled as a function of the volume density (e.g NeRF)
    - Representing volume density as a CDF derived from the learned SDF which represents the geometry of the scene

Method overview



Volume rendering

- A "classic" technique(*) for rendering 2D image of 3D scene
  - Transmittance: represents the ratio of ray that is able to get through a medium over a certain distance
    - $T(t) = \exp\left(-\int_0^t \sigma(x(s))\mathrm{ds}\right)$
  - Opacity: complement probability of transmittance
    - $O(t) = 1 - T(t), O(0) = 0, O(\infty) = 1$(assume every ray is eventually occluded)
    - Can be seen as a CDF of some probability distribution!
  - Light field of the scene $L(x(t), n(t), \mathrm{v})$: color, lighting, reflection



$$I(\mathbf{c}, \mathbf{v}) = \int_0^\infty L(\mathbf{x}(t), \mathbf{n}(t), \mathbf{v})\tau(t)dt,$$

$$\tau(t) = \frac{dO}{dt}(t) = \sigma(\mathbf{x}(t))T(t)\,.$$

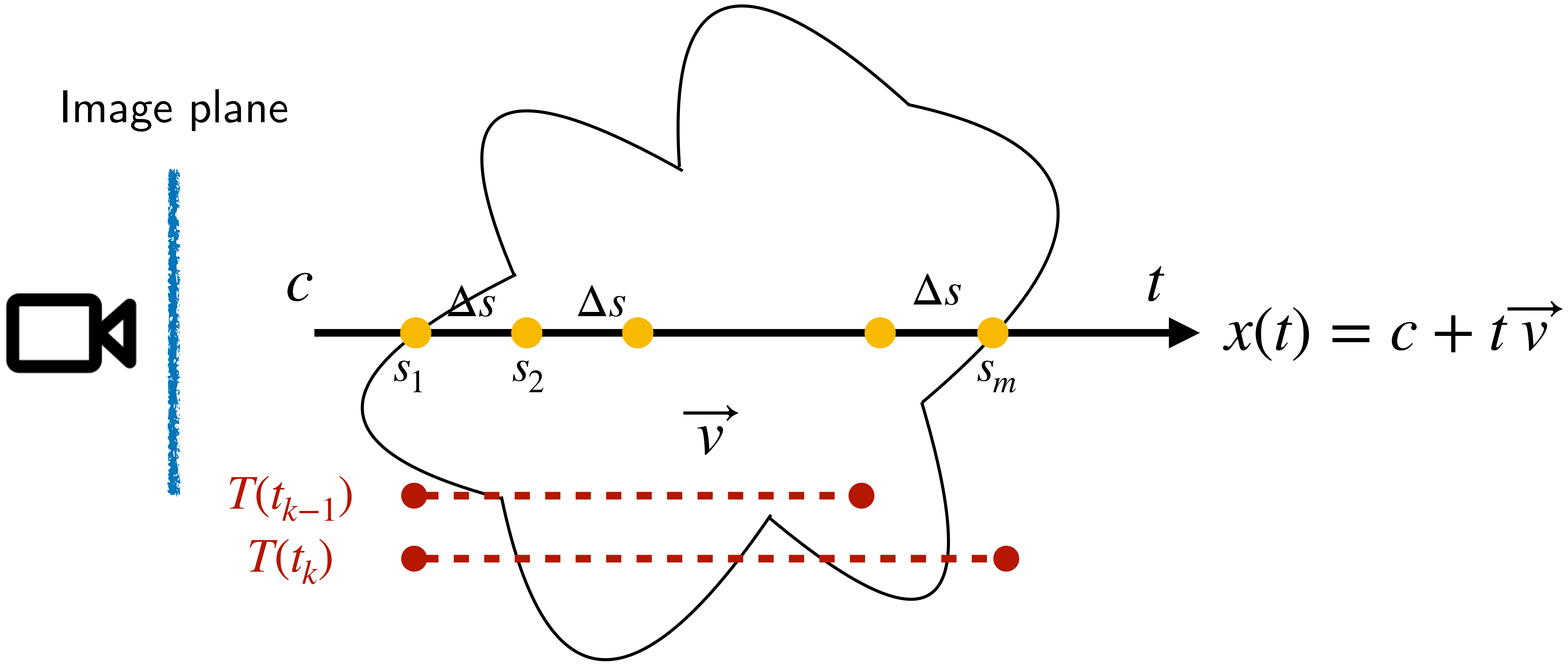
↑

probability density function (PDF) related to $O(t)$

* Ray Tracing Volume Density, Kajiya et al, SIGGRAPH 1984

Volume rendering(cont.)

- Approximated rendering equation using numerical quadrature
- Sample set $\mathcal{S} = \{s_i\}_{i=1}^{m}, 0 = s_1 < s_2 < \ldots < s_m = M$
- Approximated PDF $\tau(s_i)$
- Interval between sampled point Δs

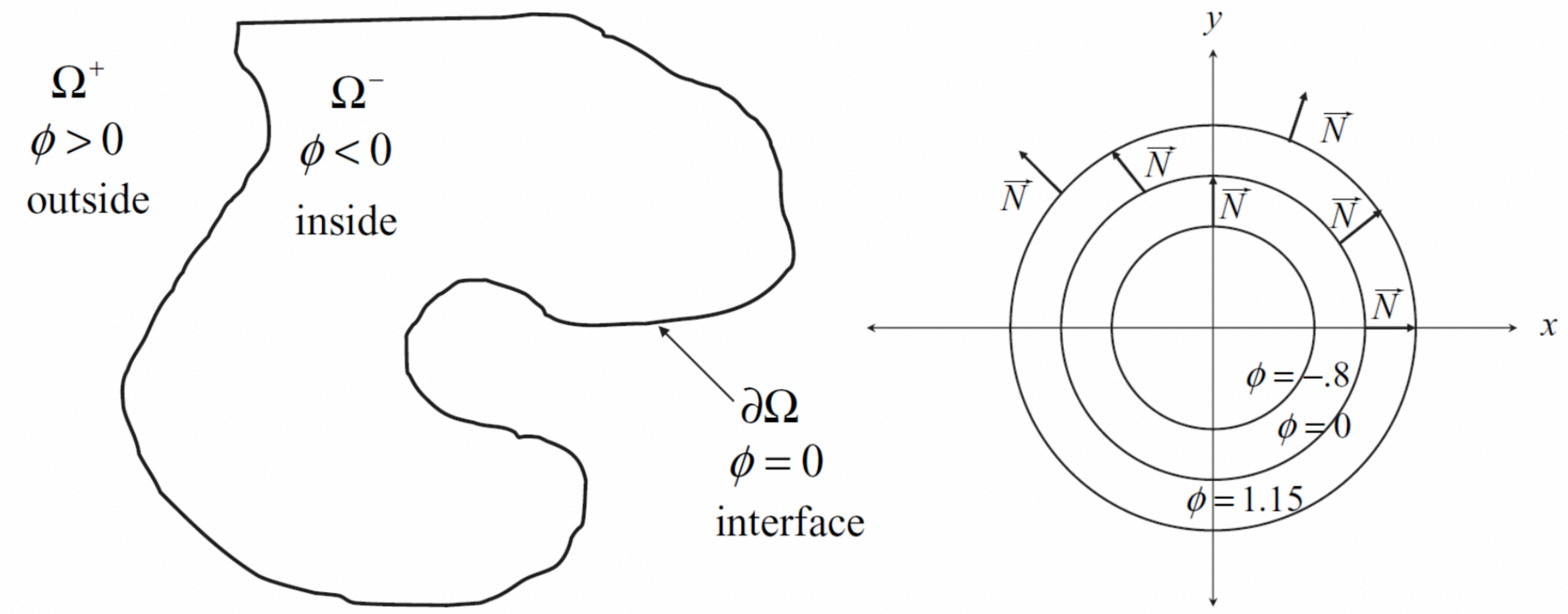


$$I(\mathbf{c}, \mathbf{v}) \approx \hat{I}_{\mathcal{S}}(\mathbf{c}, \mathbf{v}) = \sum_{i=1}^{m-1} \hat{\tau}_i L_i$$

$$\hat{\tau}_i \approx \tau(s_i)\Delta s$$

Volume density as transformed SDF

- Signed distance function (SDF)
  - positive-valued outside, negative-valued inside and zero-valued interface
  - SDF gradient $\nabla_x(\Phi(x))$, which is always orthogonal to the level sets
  - $||\nabla_x(\Phi(x)) = 1||, \forall x \in \partial\Omega$

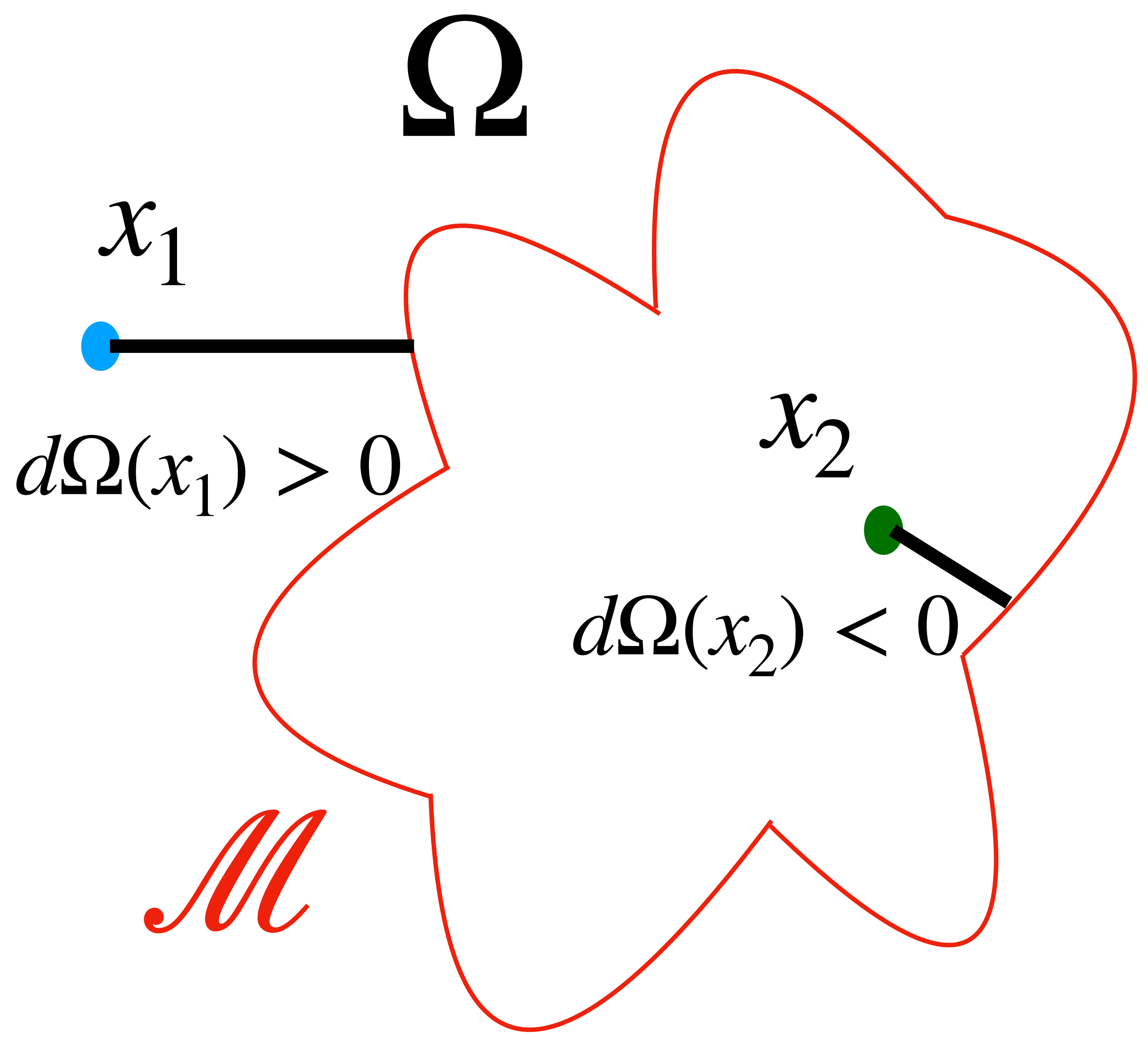


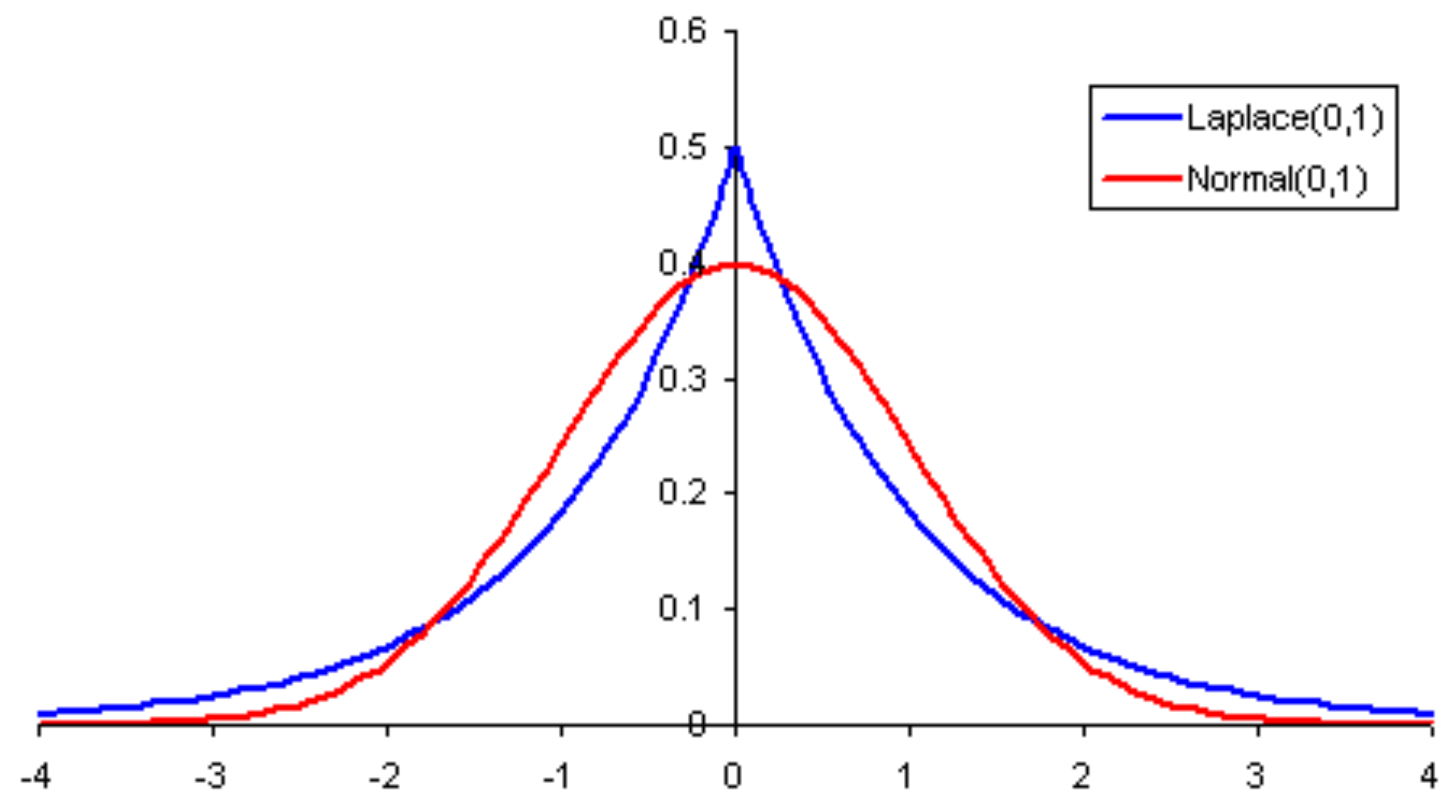
(a) Positive-, negative-, and zero-valued SDF regions.

(b) SDF gradient.

Volume density as transformed SDF(cont.)

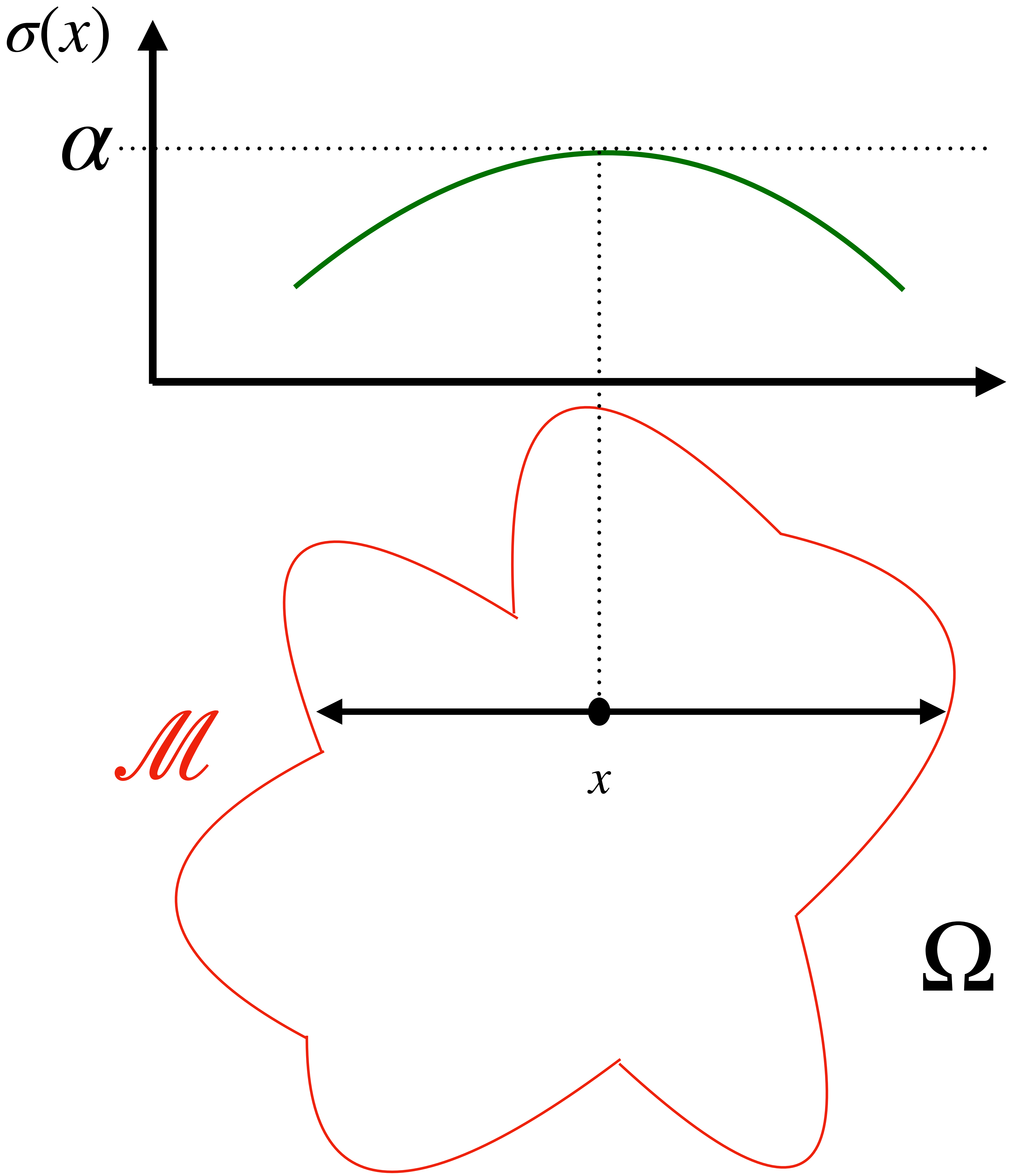
- Volume density $\sigma(x)$ as transformed SDF
  - space occupied by Ω, and $\mathscr{M} = \partial\Omega$ be its boundary surface
  - Let 1Ω be inside/outside indicator function
  - d_Ω be the minimum SDF value to its boundary $\mathscr{M}$
- Model volume density as learnable SDF
  - $\sigma(x) = \alpha\Psi_\beta(-\text{SDF})$
    - $$\Psi_\beta(s) = \begin{cases} \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s \le 0 \\ 1 - \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s > 0 \end{cases}$$
    - α, β are two learnable parameters



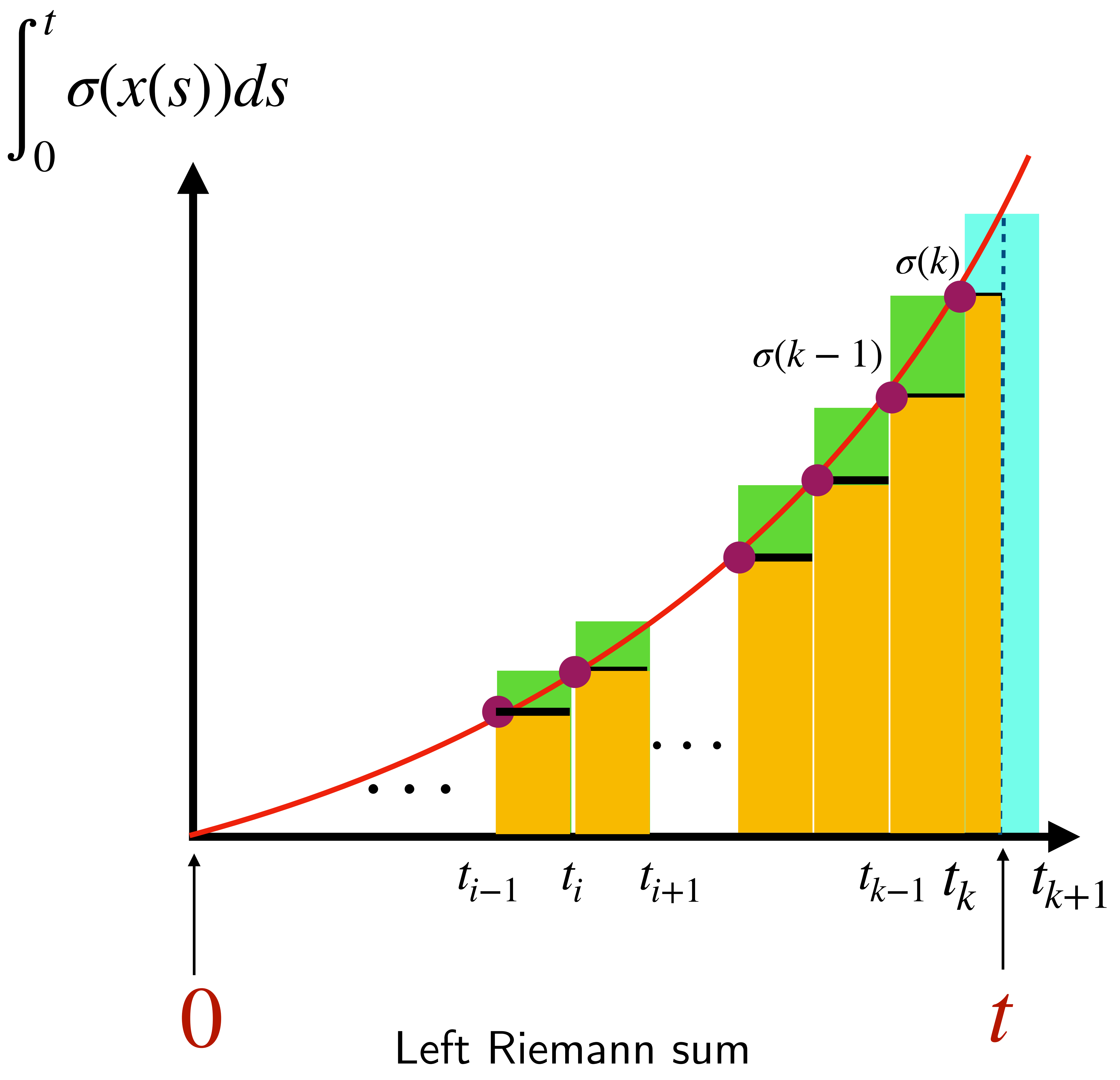


Volume density as transformed SDF(cont.)

- Model volume density as learnable SDF
  - $\sigma(x) = \alpha \Psi_\beta(-\text{SDF})$
    - $$\Psi_\beta(s) = \begin{cases} \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s \le 0 \\ 1 - \frac{1}{2}\exp\left(\frac{s}{\beta}\right) & \text{if } s > 0 \end{cases}$$
    - α, β are two learnable parameters
- Benefit ?
  - Some where inside Ω we will have $\Psi_\beta(-d\Omega(x)) \approx 1$, $\sigma(x) \approx \alpha$
  - Smoothly decrease when near object's controlled by factor β
- Using SDF makes it easy for us to extract surface by taking level set value zero
- Allows us to easily estimate the approximation error of opacity (and develop algorithms to reduce the error)

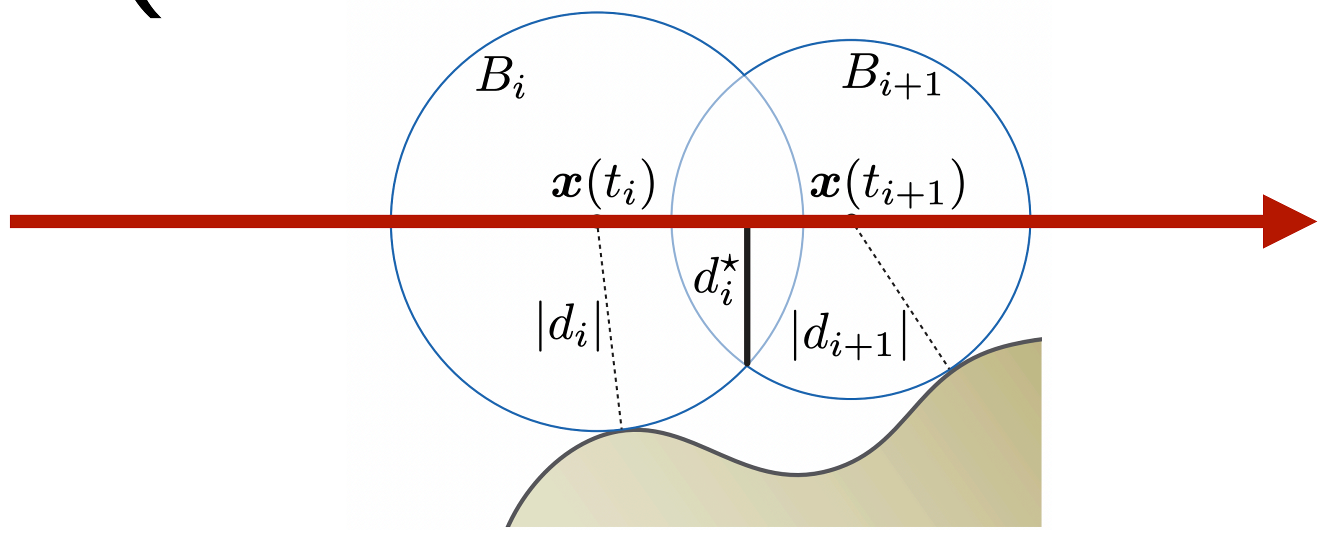


Bound on the opacity approximation error

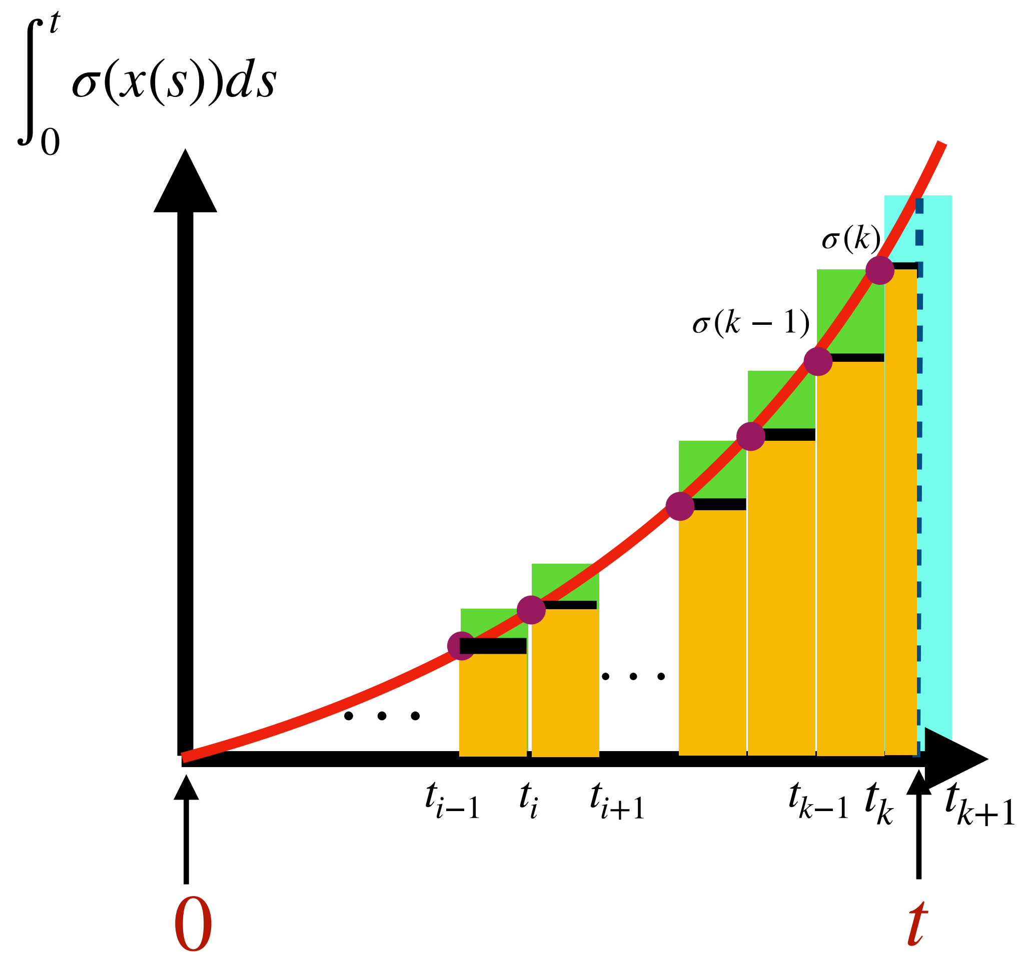


- $\int_0^t \sigma(x(s))ds = \hat{R}(t) + E(t)$
  - $\hat{R}(t) = \sum_{i=1}^{k-1} \delta_i \sigma_i + (t - t_k)\sigma_k$ approximated left Riemann sum
  - $E(t)$:opacity approximation error
- How to estimate opacity along the ray $O(t)$?
  - Since transparency $T(t) = \exp\left(-\int_0^t \sigma(x(s))\mathrm{ds}\right)$ and $O(t) = 1 - T(t)$
  - Estimate opacity : $O(t) \approx \hat{O}(t) = 1 - \exp(-\hat{R}(t))$
- How to lower approximation error $|O(t) - \hat{O}(t)|$ to get accurate opacity ?

Bound on the opacity approximation error (cont.)



- What is the upper bound of approximation error $|O(t) - \hat{O}(t)|$?
- Derive procedure (omit proof):
  1. $|\frac{d}{ds}\sigma(\mathrm{x}(s))| \leq \frac{\alpha}{2\beta}\exp(-\frac{d_i^\star}{\beta})$: derive the upper bound of derivative of volume density in arbitrary segment
  2. $|E(t)| \leq \hat{E}(t) = \frac{\alpha}{4\beta}(\sum_{i=1}^{k-1}\delta_i^2 e^{-\frac{d_i^\star}{\beta}} + (t-t_k)^2 e^{-\frac{d_k^\star}{\beta}})$ derive an error bound for the left Riemann sum approximation of the opacity
  3. $|O(t) - \hat{O}(t)| \leq \exp(-\hat{R}(t))(\exp(\hat{E}(t)) - 1)$ derive upper bound on the opacity approximation error
- $B_{\mathcal{T},\beta} = \max_{k\in[n-1]}\{\exp(-\hat{R}(t_k))(\exp(\hat{E}(t_{k+1})) - 1)\} \geq \max_{t\in[0,M]}|O(t) - \hat{O}(t)|$ maximum error bound over all intervals
- Useful lemmas for enhance sampling algorithm (omit proof):
  - Lemma1: fix $\beta > 0$. For any $\epsilon > 0$ a sufficiently dense sampling $\mathcal{T}$ will provide $B_{\mathcal{T},\beta} < \epsilon$
  - Lemma2: fix $n > 0$. For any $\epsilon > 0$ a sufficiently large $\beta \geq \frac{\alpha M^2}{4(n-1)\log(1+\epsilon)}$ will provide $B_{\mathcal{T},\beta} \leq \epsilon$

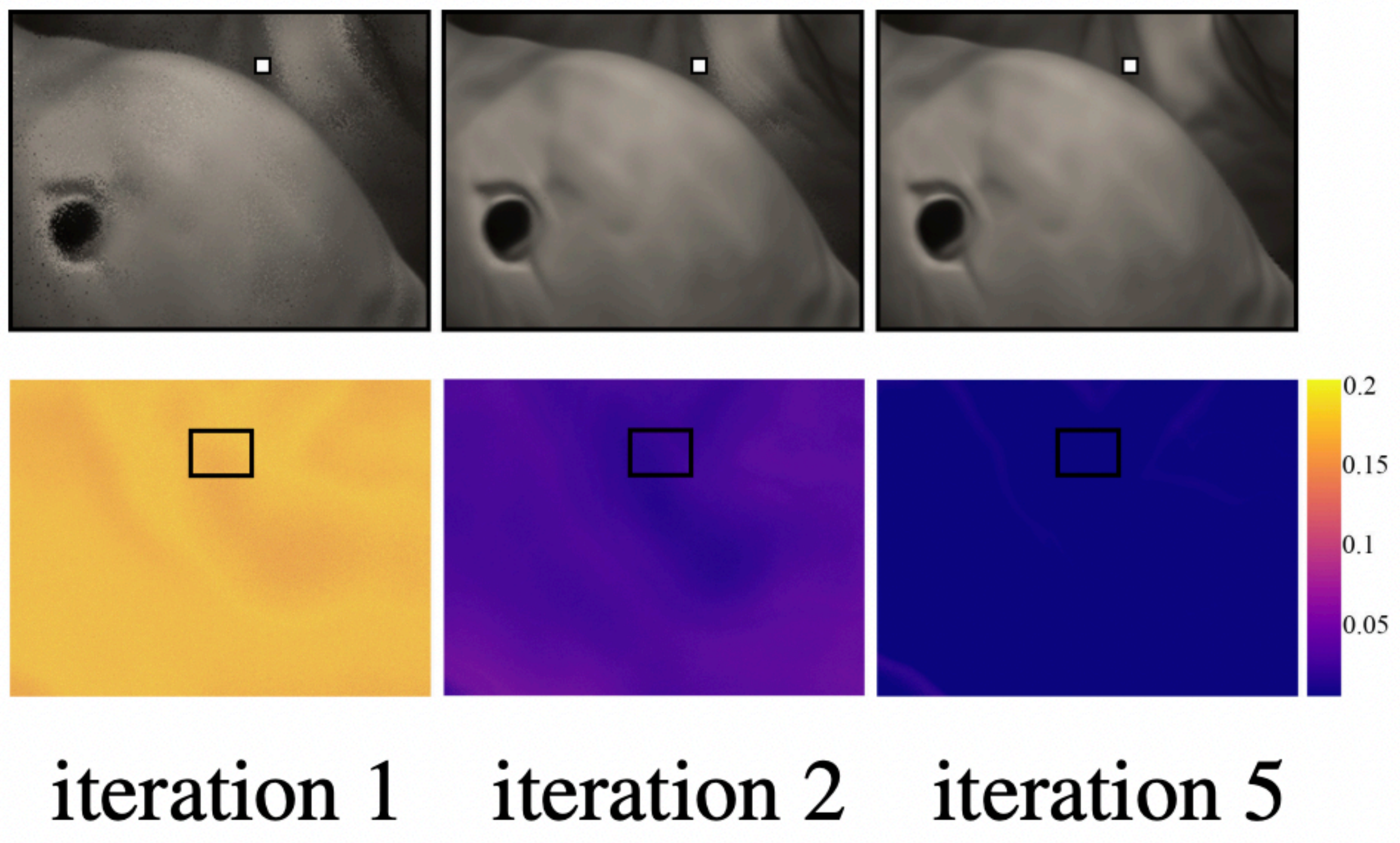


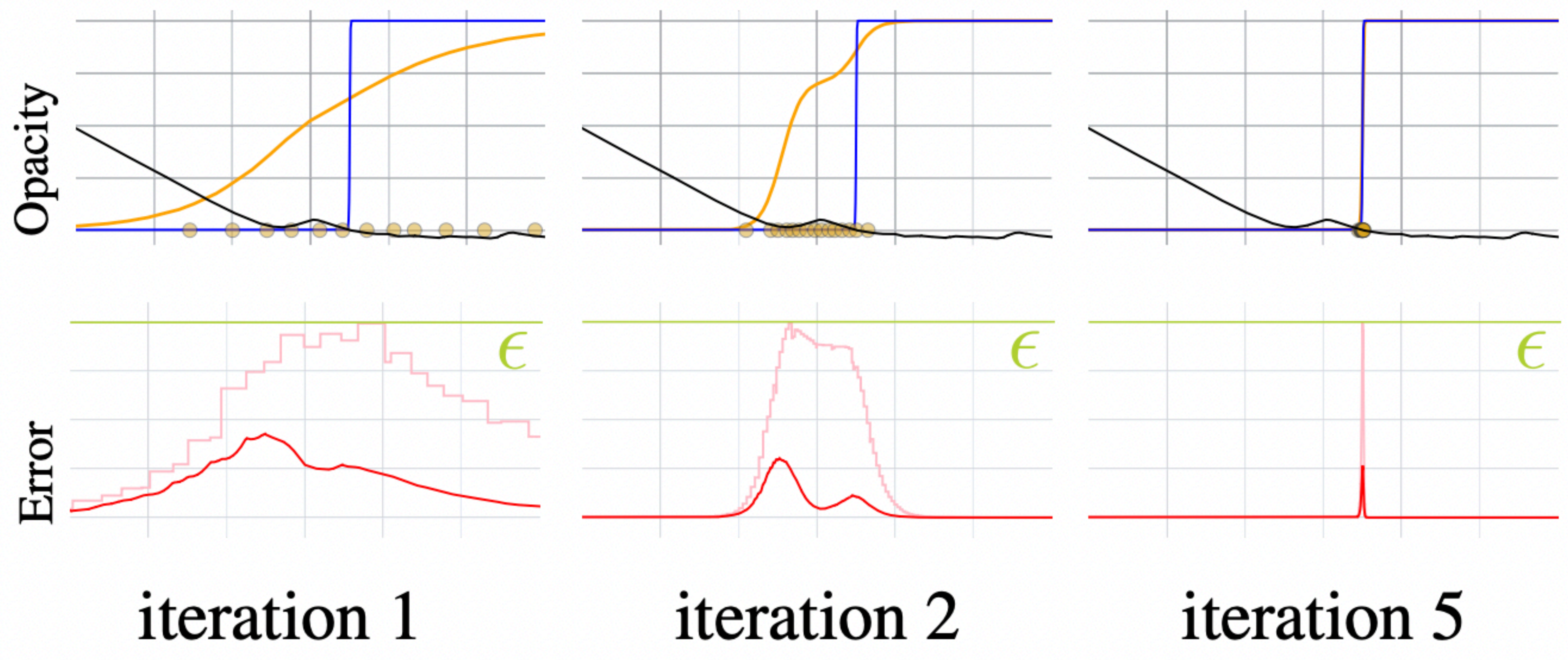
Sampling algorithm

Algorithm 1: Sampling algorithm.

Input: error threshold $\epsilon > 0$; β

Initialize $\mathcal{T} = \mathcal{T}_0$
Initialize β_+ such that $B_{\mathcal{T},\beta_+} \leq \epsilon$
while $B_{\mathcal{T},\beta} > \epsilon$ *and not max_iter* **do**
    upsample $\mathcal{T}$
    if $B_{\mathcal{T},\beta_+} < \epsilon$ **then**
        Find $\beta_\star \in (\beta, \beta_+)$ so that $B_{\mathcal{T},\beta_\star} = \epsilon$
        Update $\beta_+ \leftarrow \beta_\star$
    end
end
Estimate $\widehat{O}$ using $\mathcal{T}$ and β_+
$\mathcal{S} \leftarrow$ get fresh m samples using $\hat{O}^{-1}$
return $\mathcal{S}$





Evaluation

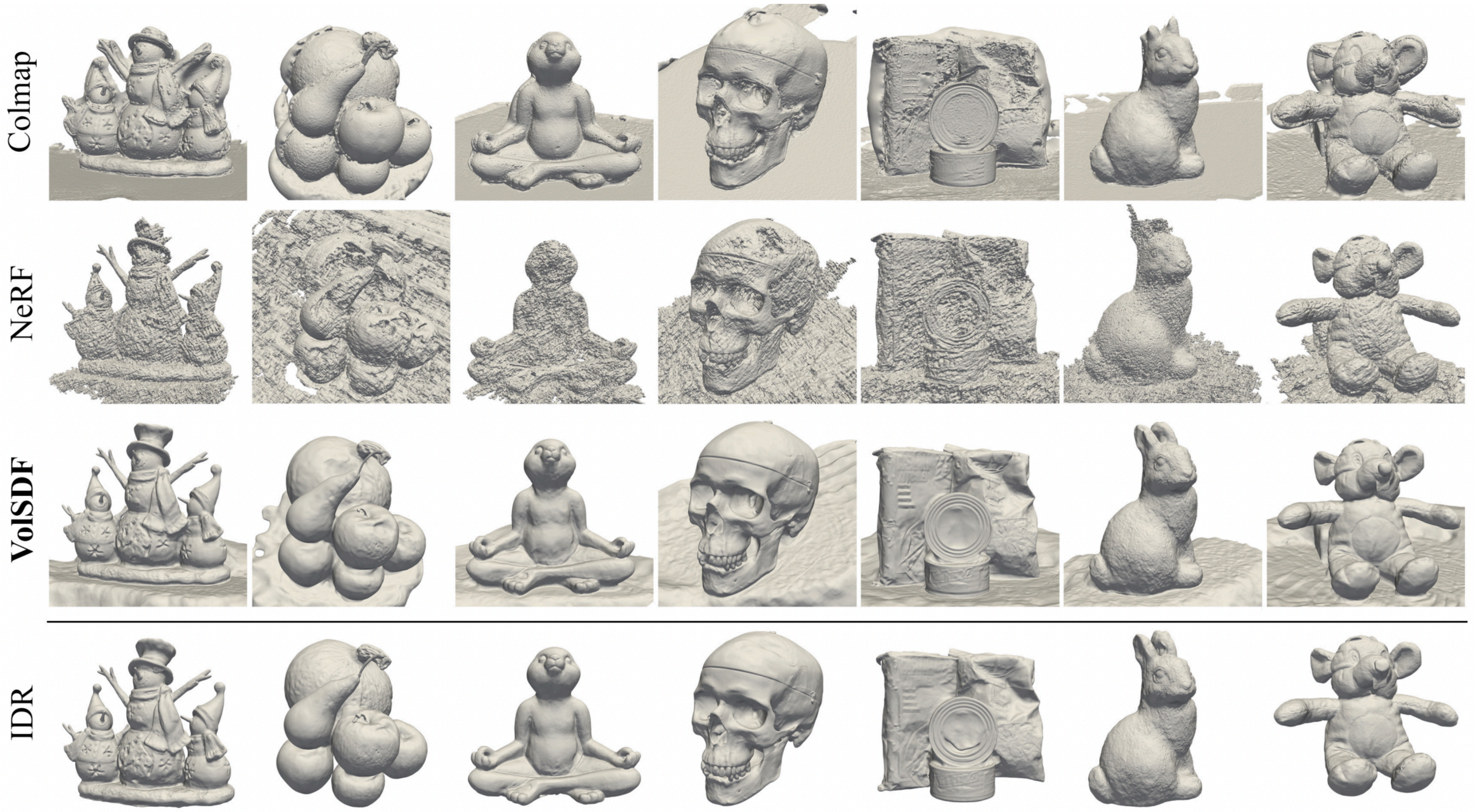


Figure 4: Qualitative results for reconstructed geometries of objects from the DTU dataset.