Neural Networks as PRIOR-BASED INFERENCE ALGORITHMS

6.S980 | 2022-05-19

Project Proposal Details (due Oct 18th)

- 1-2 letter pages in CVPR 2022 layout (find on Canvas).
- Sections:
  - **Abstract / Problem definition:** What are you trying to do?
  - **Brief related work:** How is it done today, and what are the limits of current practice?
  - **Your proposed method:** What is new in your approach and why do you think it will be successful?
  - **Why is it interesting:** Who cares? If you are successful, what difference will it make?
  - Description of the first, simplest experiment that will show you if your idea has merit
  - Description of final experiment - best case, what will you be able to show?

Project Details

- Project proposals due in ~2 weeks (Oct 18th).
- **Vincent's project office hours:**
  - Wednesday, October 5th, 10 - 11 pm.
  - Friday, October 7th, 2 - 3 pm (**weekly**)
  - 10-minute time slots per team.
  - Sign up here.
- My advice: Try preparing the answers to the questions on the previous slide - that's the fastest, most effective pitch you can do!

Example Paper Discussion

Decomposing NeRF for Editing via Feature Field Distillation

Sosuke Kobayashi
Preferred Networks, Inc.
sosk@preferred.jp

Eiichi Matsumoto
Preferred Networks, Inc.
matsumoto@preferred.jp

Vincent Sitzmann
Massachusetts Institute of Technology
sitzmann@mit.edu

pfnet-research.github.io/distilled-feature-fields/

Abstract

Emerging neural radiance fields (NeRF) are a promising scene representation for computer graphics, enabling high-quality 3D reconstruction and novel view synthesis from image observations. However, editing a scene represented by a NeRF is challenging, as the underlying connectionist representations such as MLPs or voxel grids are not object-centric or compositional. In particular, it has been difficult to selectively edit specific regions or objects. In this work, we tackle the problem of semantic scene decomposition of NeRFs to enable query-based local editing of the represented 3D scenes. We propose to distill the knowledge of off-the-shelf, self-supervised 2D image feature extractors such as CLIP-LSeg or DINO into a 3D feature field optimized in parallel to the radiance field. Given a user-specified query of various modalities such as text, an image patch, or a

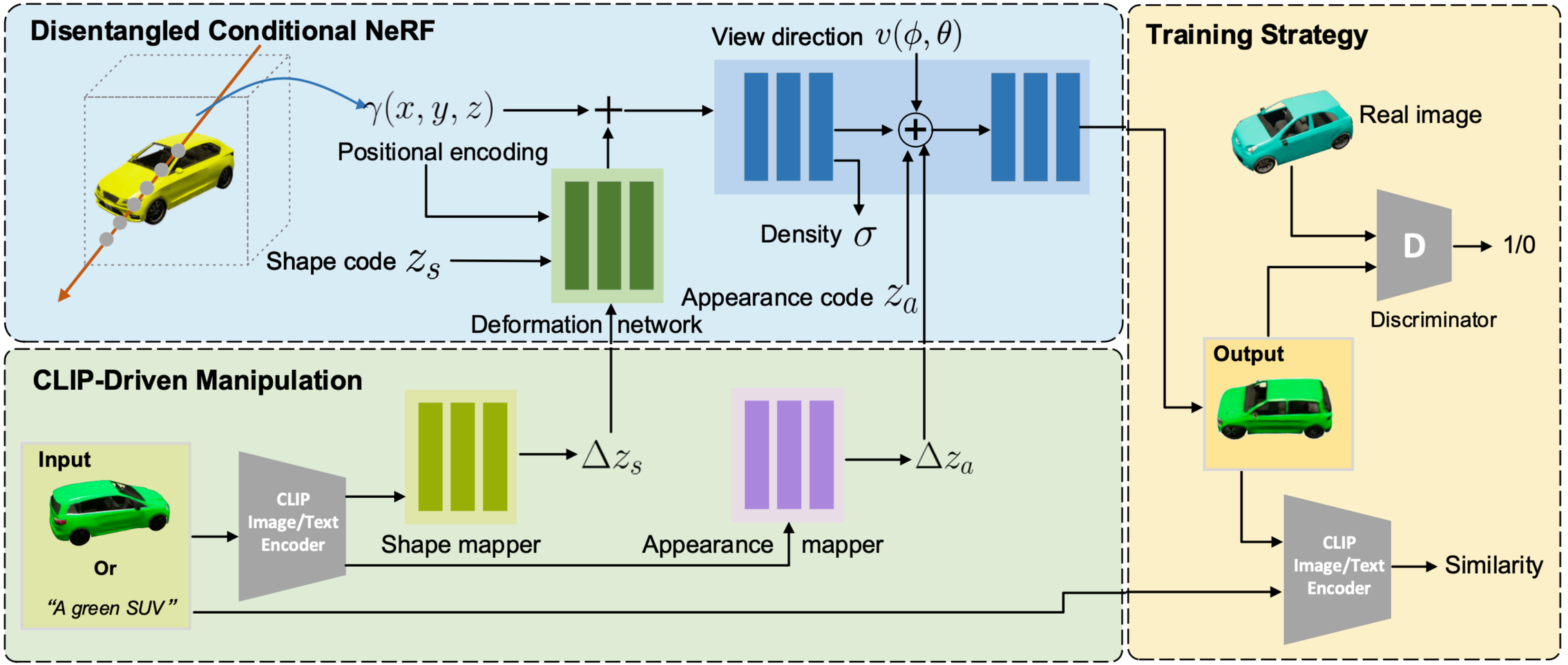
Problem Statement: What is it trying to do?

- Given: **Many** 2D images + cameras of **single** 3D scene
- Today: can reconstruct 3D scenes via diff. rendering, but not easily editable!
- Goal: Reconstruct, then **edit** 3D scene representation
- Edits are color, deletion, copy-paste, moving of scene parts, …

Related Work: CLIPNeRF

> Wang et al. [84] and Jain et al. [32] use CLIP to edit or generate a single-object NeRF with a text prompt query by optimizing the NeRF parameters to generate images matched with the text. While such methods are promising, they do not enable selective editing of specific scene regions. For example, the prompt “yellow flowers” may affect unintended scene regions, such as the leaves of a plant.

Related Work: CLIPNeRF



Related Work: Object-Centric Representations

> Other studies also explored reconstruction with more structured hybrid representations via pipelines specialized to a domain (e.g., traffic scene) [58, 22, 36] or situation (e.g., each object data is independently accessible) [25, 24, 92]. Note that this line of work defines and constrains domains or the types of segmentation during or before training, and thus limits the degrees of freedom for editable scenes and objects.

Background: Dino Feature Extractor

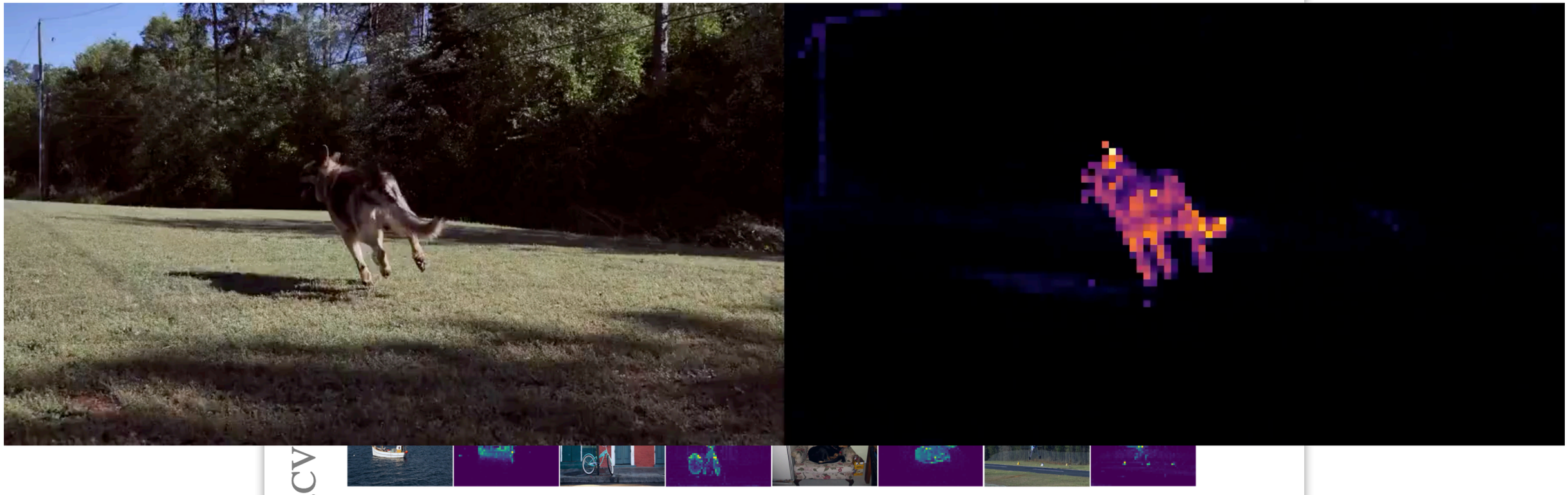
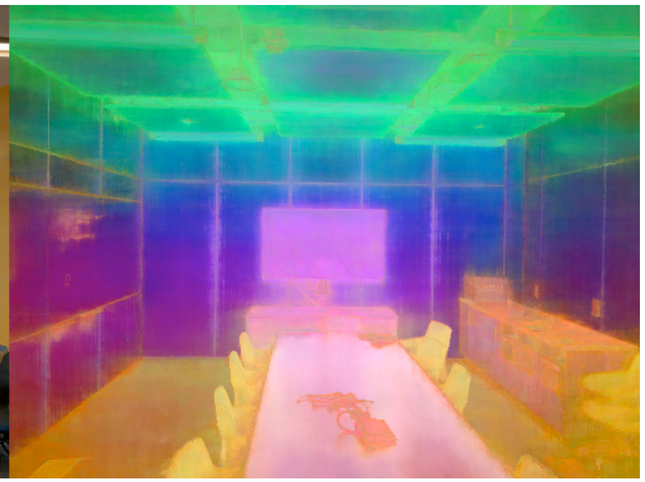
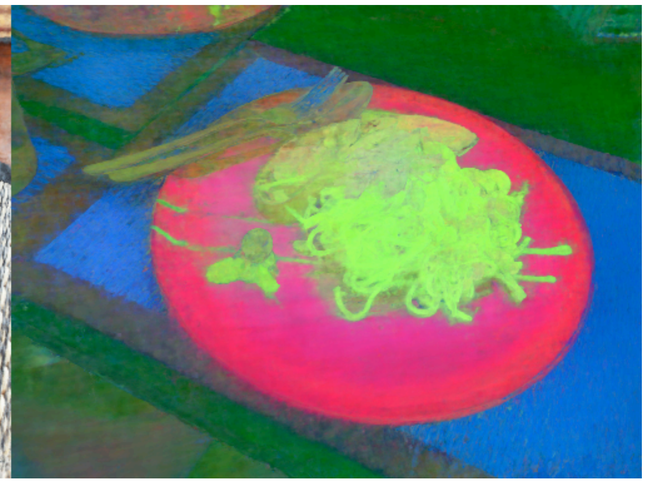


Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the `[CLS]` token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

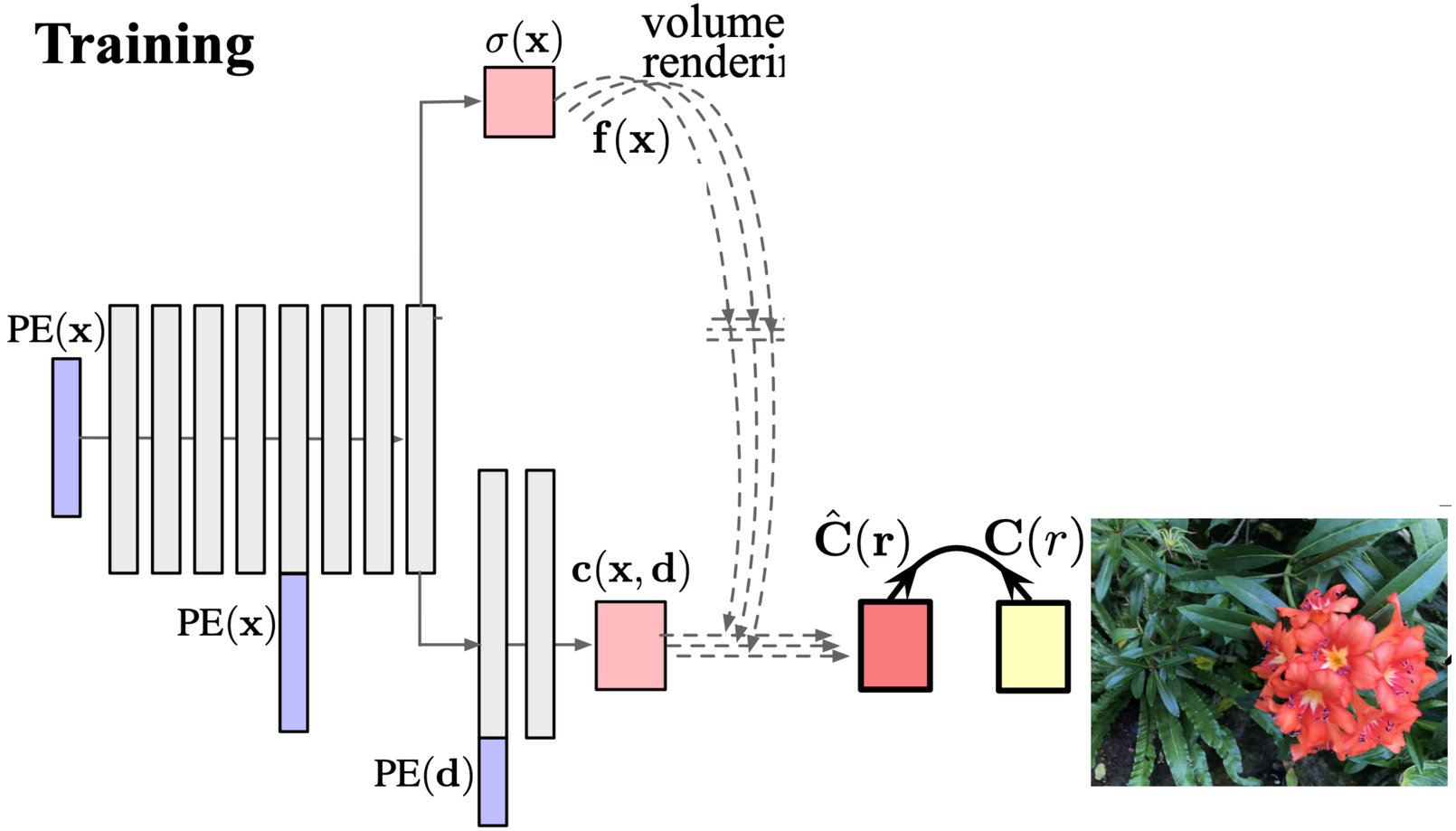
Background: Dino Feature Extractor



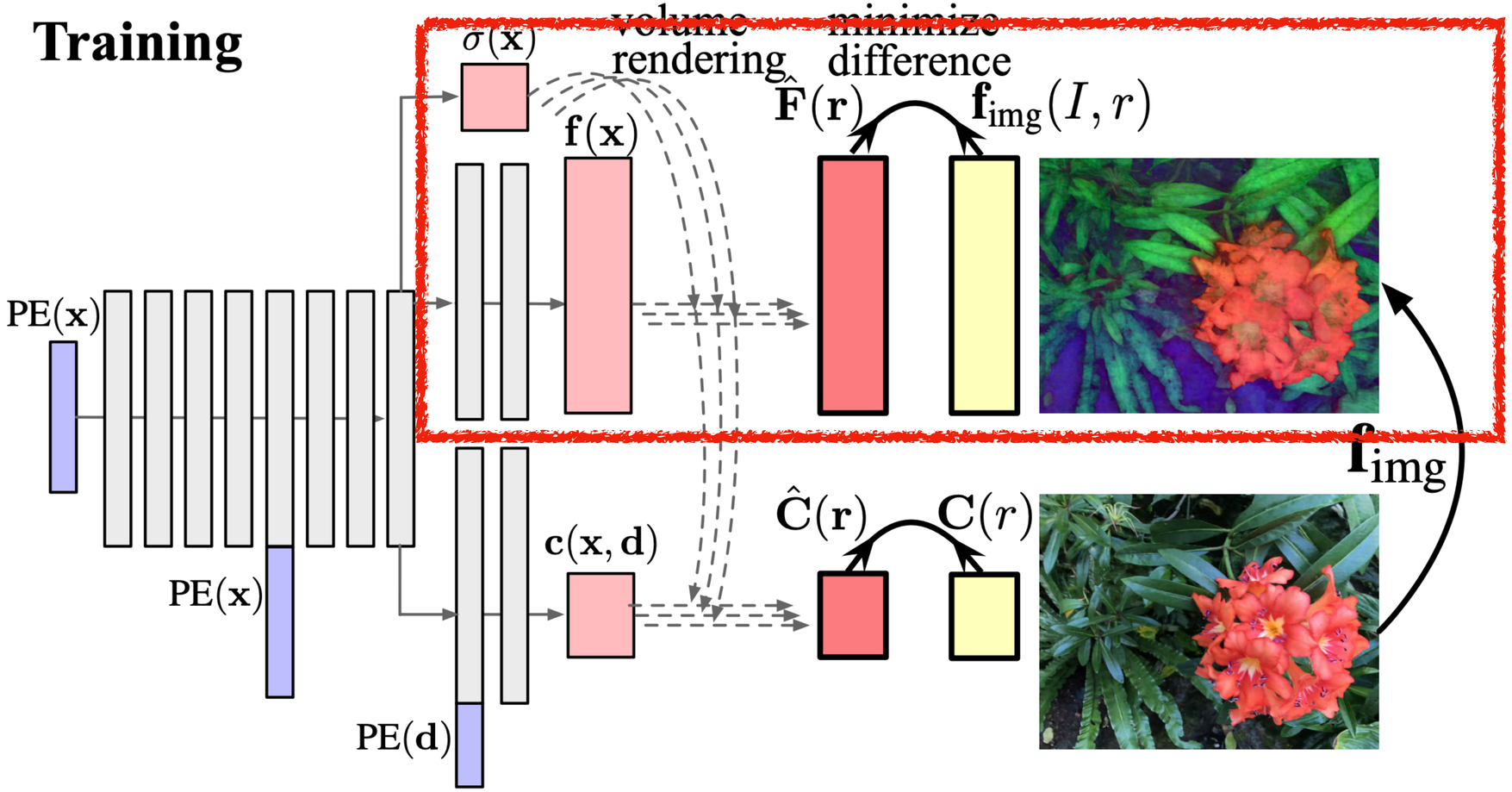





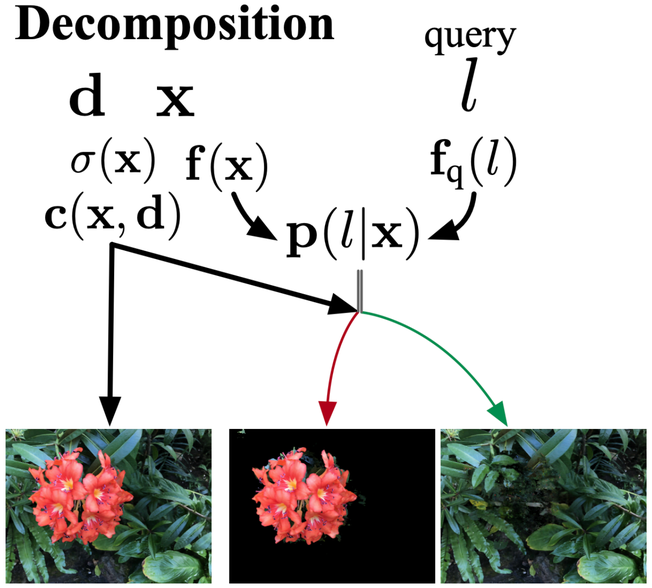
Method: Training Feature Field



Method: Training Feature Field



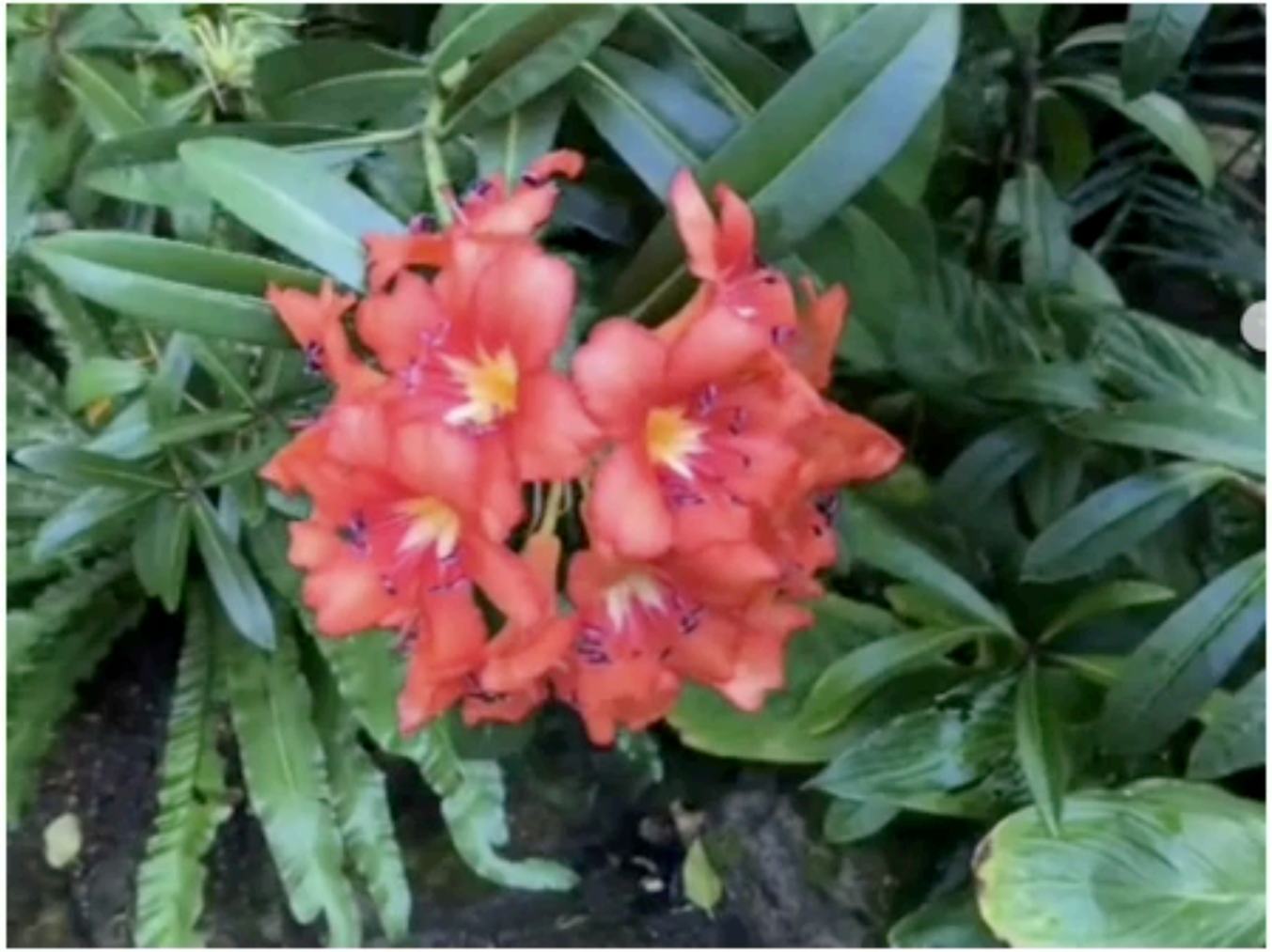
Method: Selecting Regions



A trained DFF model can perform 3D zero-shot segmentation by its feature field $\mathbf{f}$ and a query encoder $\mathbf{f}_q$. Probability of a label l of a point $\mathbf{x}$ in the 3D space, $\mathbf{p}(l|\mathbf{x})$, is calculated by dot product of the 3D feature $\mathbf{f}(\mathbf{x})$ and text label feature $\mathbf{f}_q(l)$ followed by a softmax:

$$\mathbf{p}(l|\mathbf{x}) = \frac{\exp(\mathbf{f}(\mathbf{x})\mathbf{f}_q(l)^T)}{\sum_{l' \in \mathcal{L}} \exp(\mathbf{f}(\mathbf{x})\mathbf{f}_q(l')^T)} \quad . \tag{5}$$

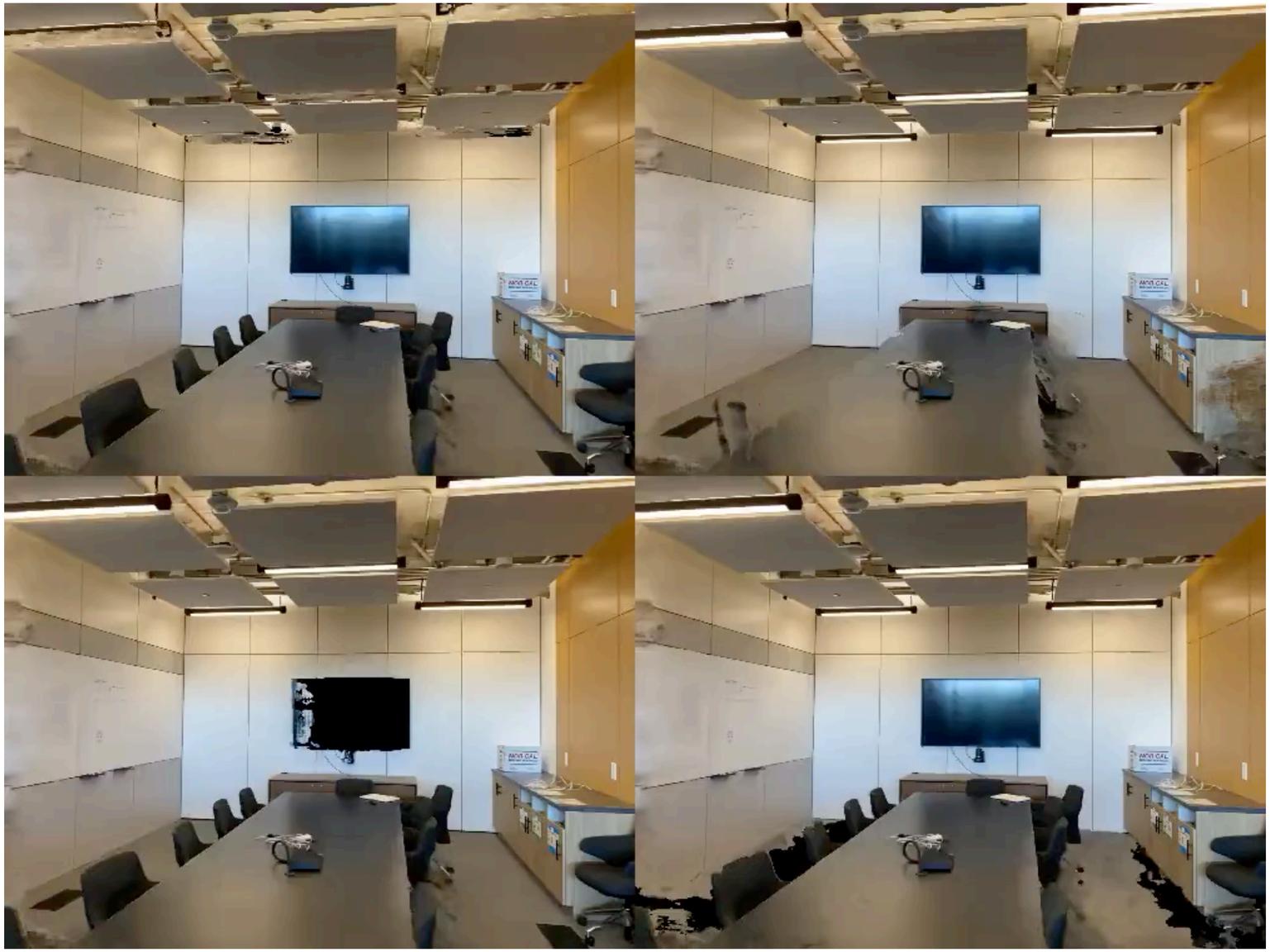
Results



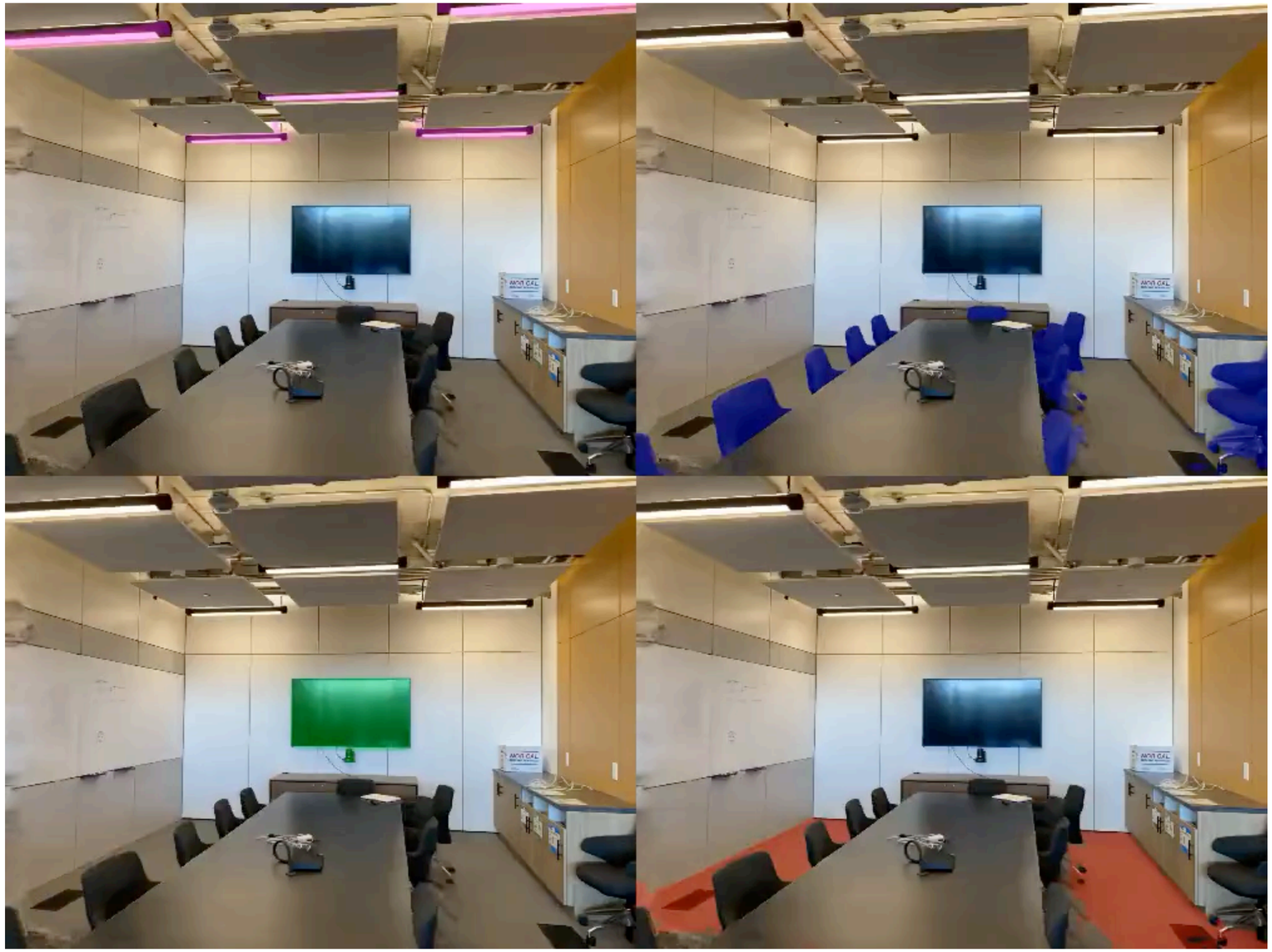
Baseline



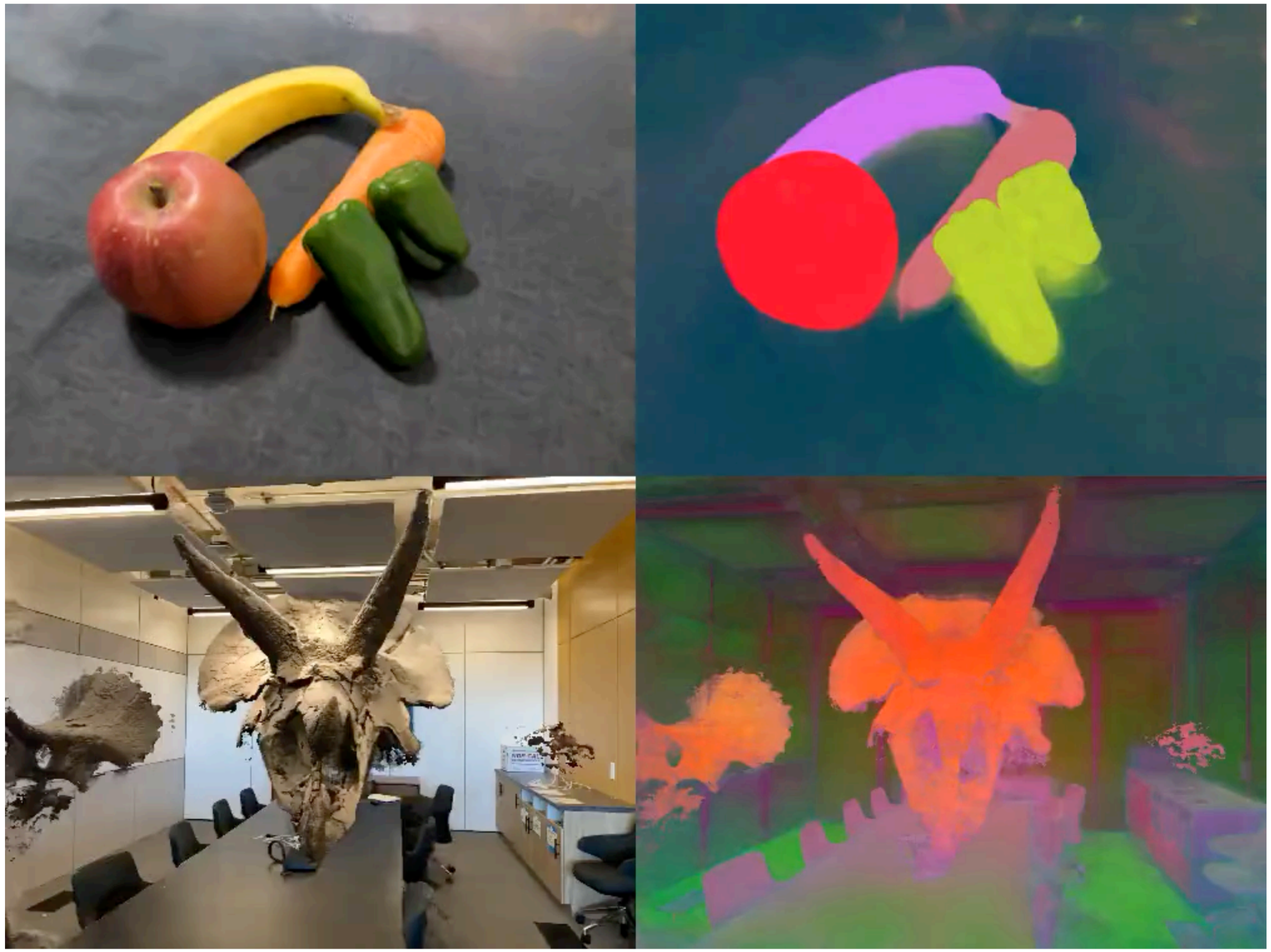

Deleting objects!



Re-coloring objects!



Re-sizing and copy-pasting



Ground truth
Picture
Wall
Lamp
Table
Cushion
Sofa
Floor
coarse MLP
fine MLP

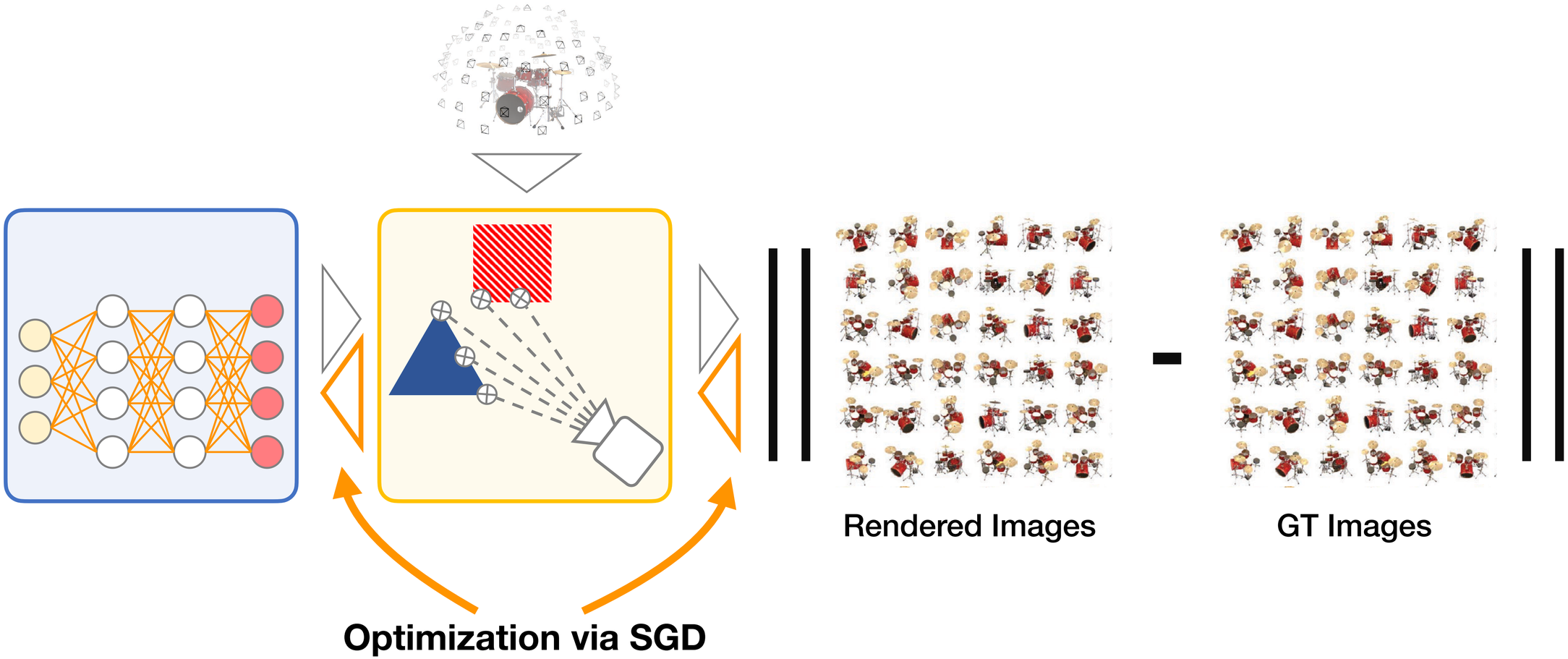
Limitations and Future Work

• • •

Example Paper Discussion

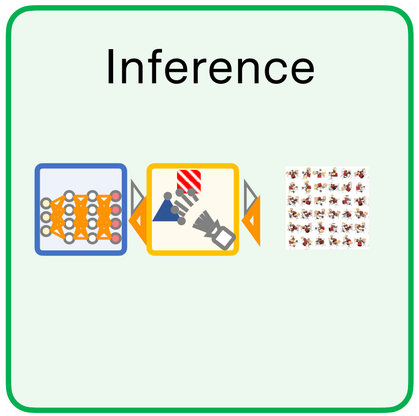
—END—

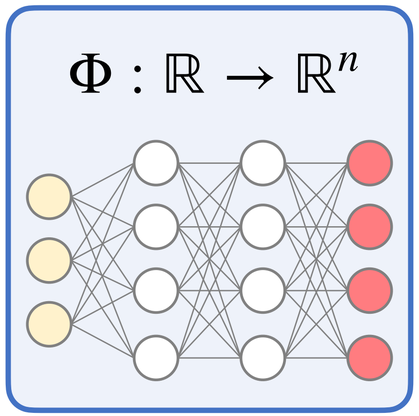
Recap: Diff. Rendering

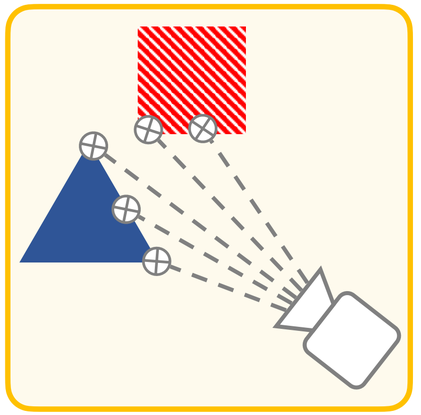


Recap: Diff. Rendering is inference

Observations



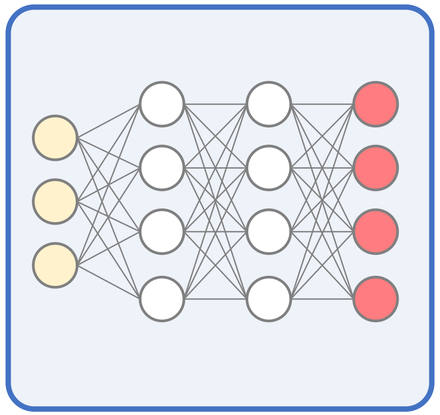




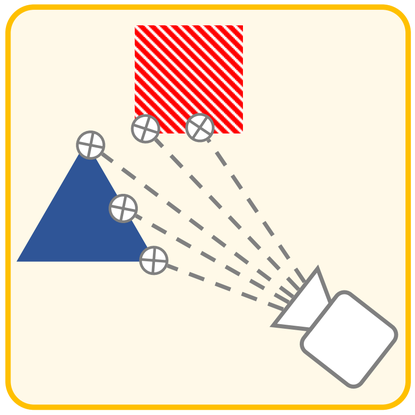
Renderings

We inferred “hidden/latent” variables 3D appearance and geometry, given “observable” variables RGB images and camera poses.

Recap: What if observations don't constrain scene representation?



Srn_{ϕ}



Render_{θ}

$+$

$(\mathcal{I}, \xi)$

No 3D inform.

$$\underset{\phi,\theta}{\text{argmin}} \left\| \text{Render}_{\theta}(\text{Srn}_{\phi}, \xi) - \mathcal{I} \right\|$$

Input

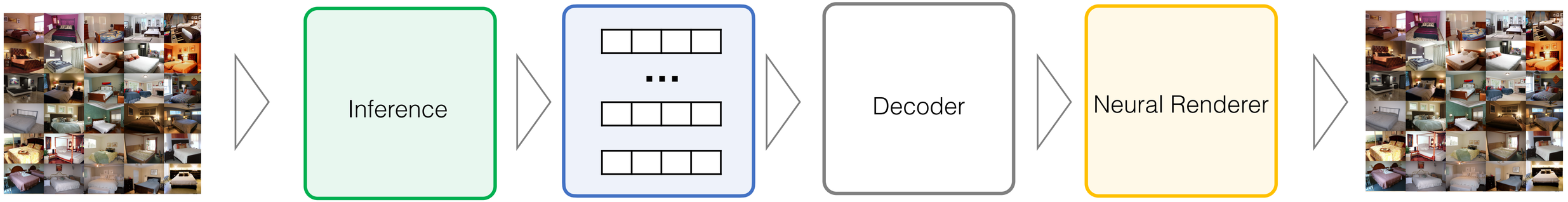
Normal map

RGB

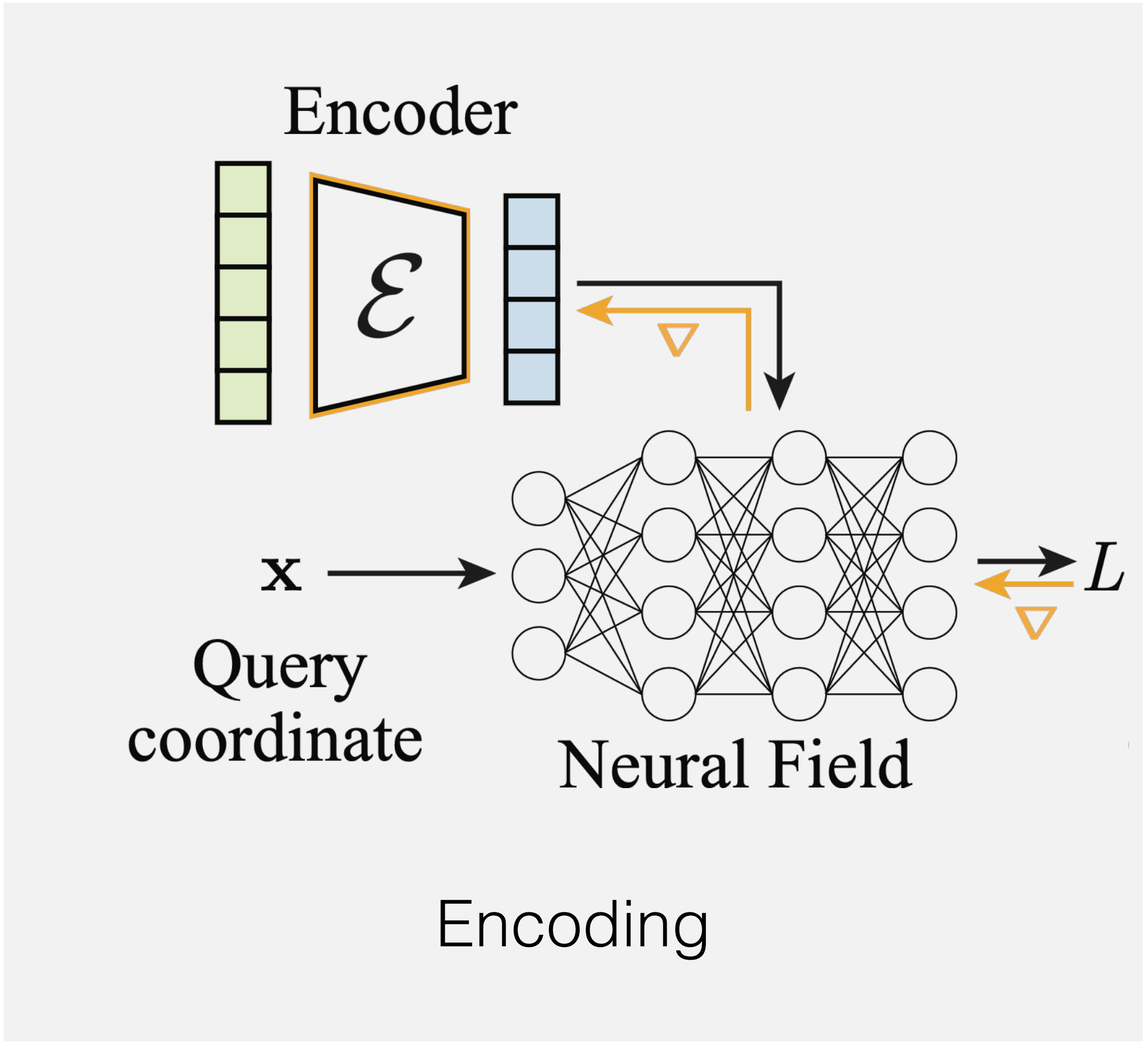
GT

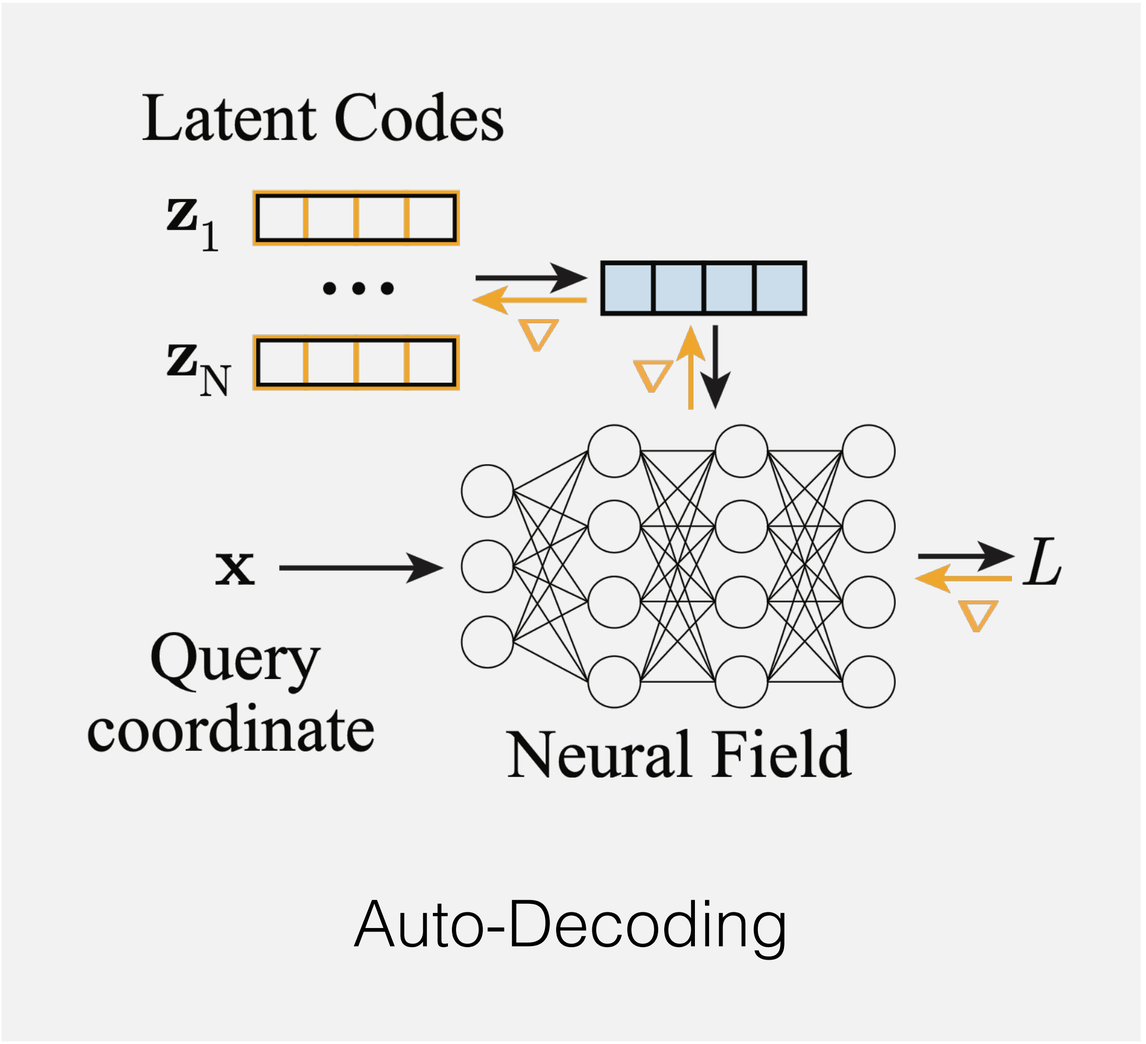
Recap: General Framework for auto-encoding based Scene Representation Learning

(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$



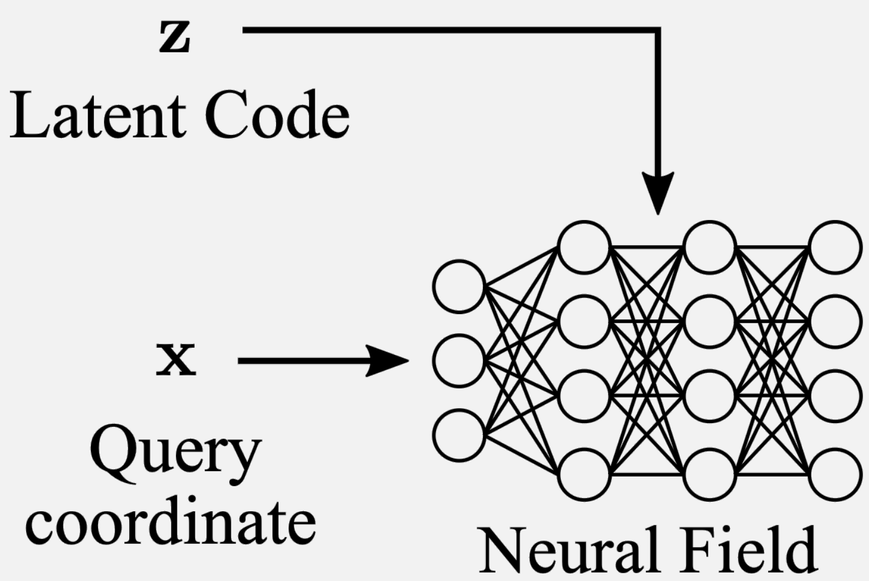
Recap: Encoding & Auto-Decoding



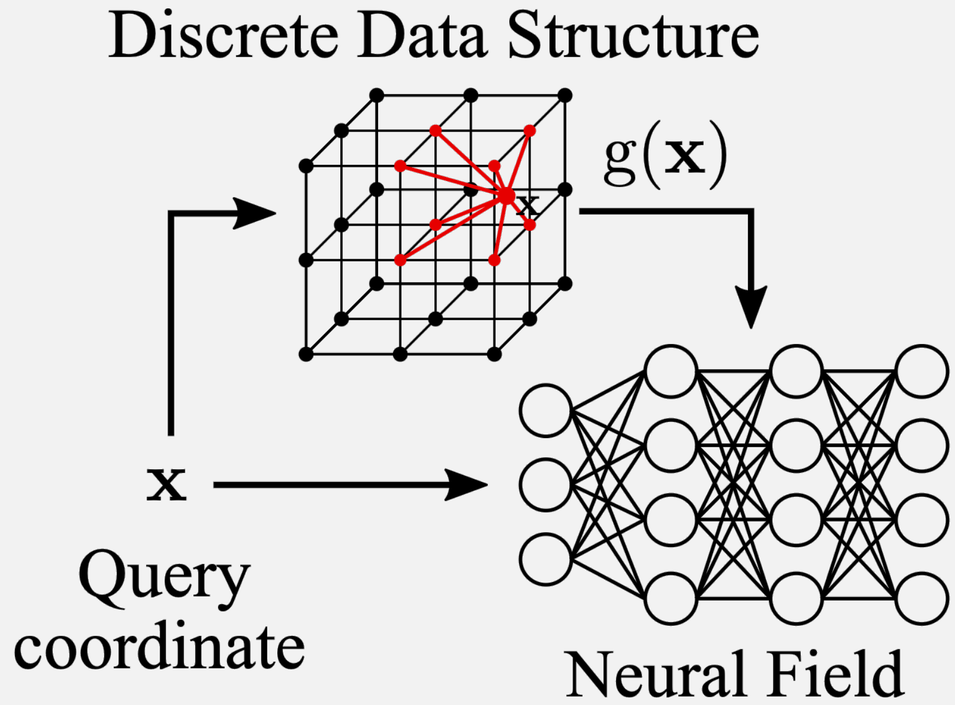


Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Recap: Global Conditioning and Local Conditioning



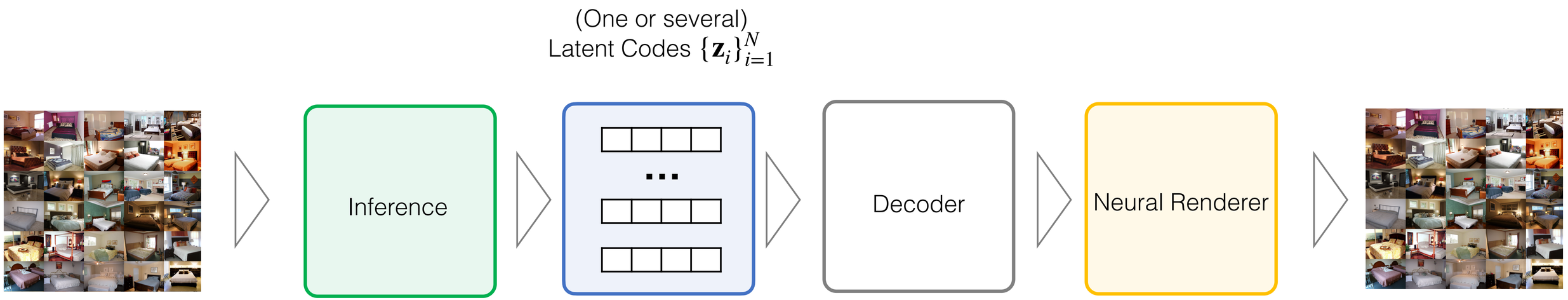
Global Conditioning



Local Conditioning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Today: Advanced Inference Topics



Amortized Neural Rendering (=Light Fields)

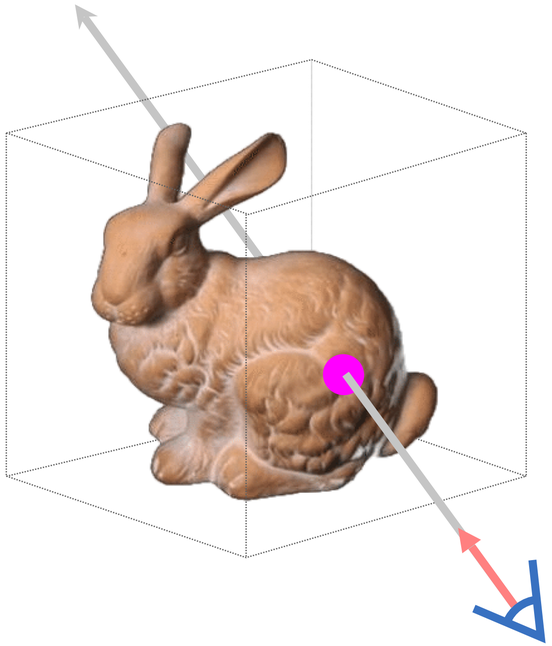
Conditioning with Transformers

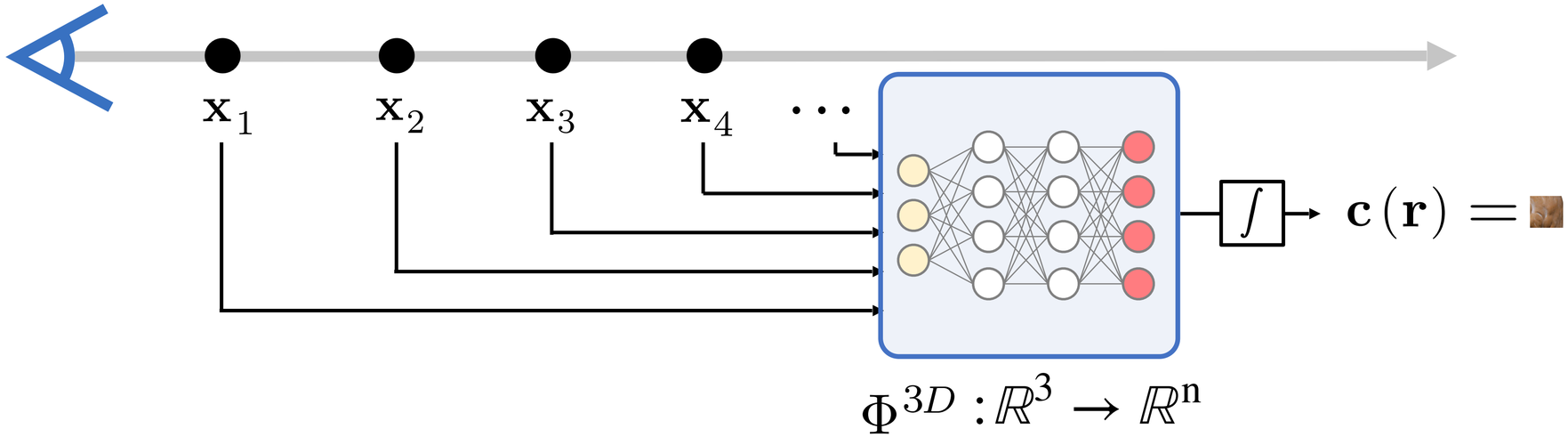
Operating on Neural Scene Representations

Geometric Deep Learning

…

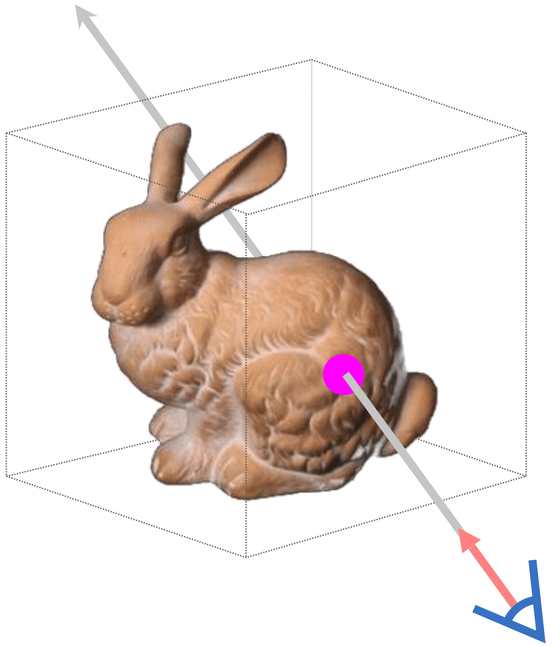
General structure of Neural Renderers for 3D-structured Representations





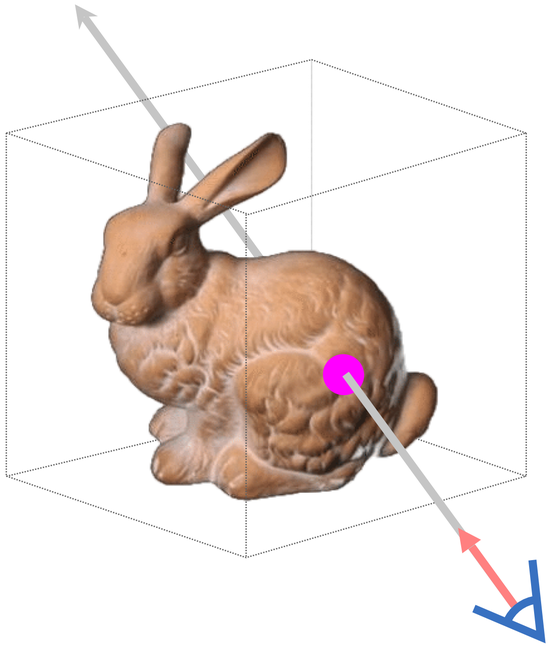
Hundreds of samples per ray.

Time- and memory-intensive.



Light Field

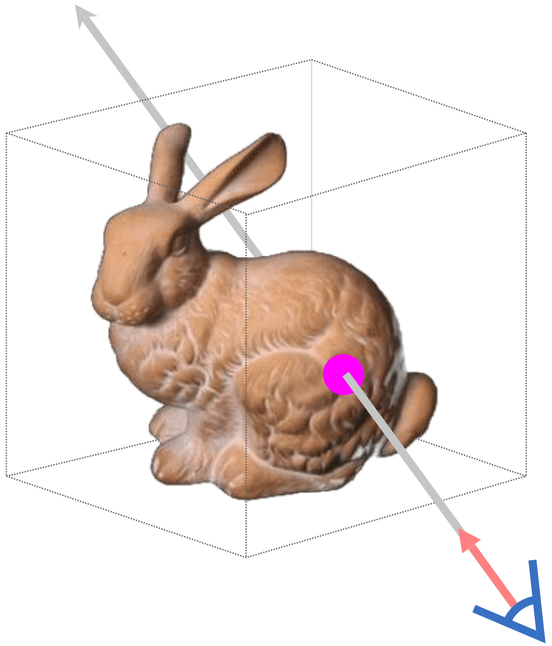
[Adelson et al. 1991, Levoy et al. 1996, Gortler et al. 1996]

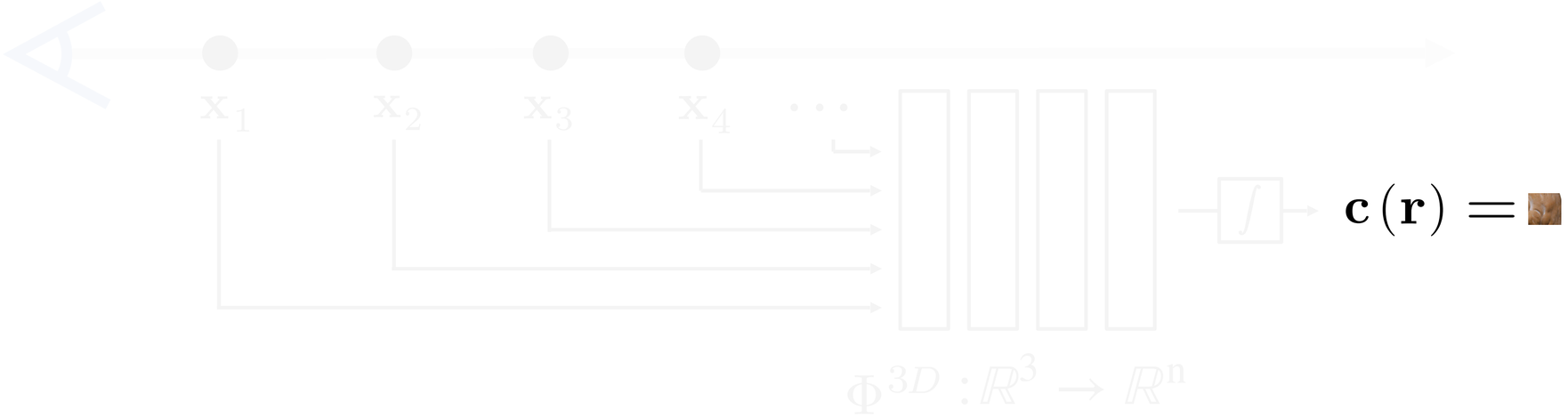


Light Field Networks

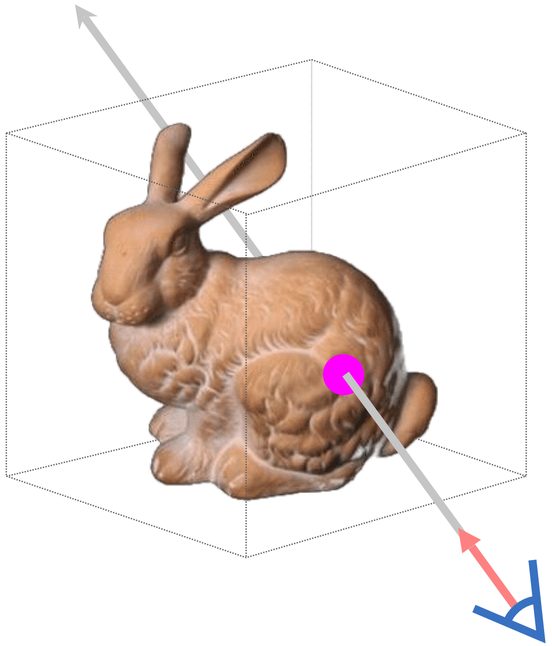
$\mathbf{c}(\mathbf{r}) =$

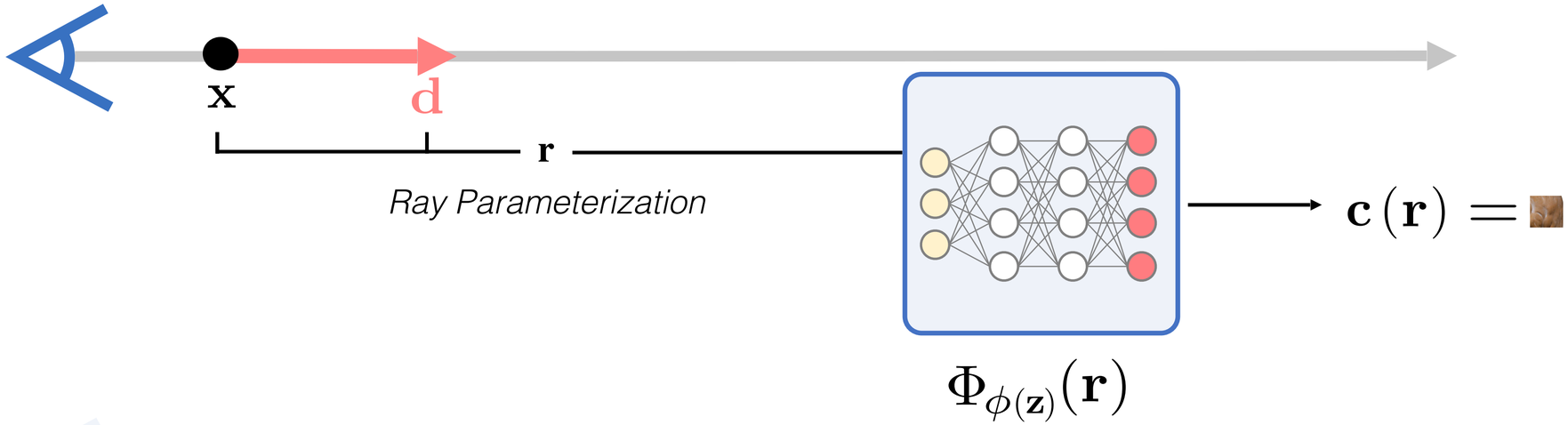
Light Field Networks





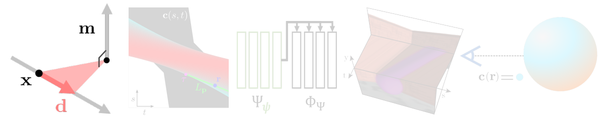
Light Field Networks



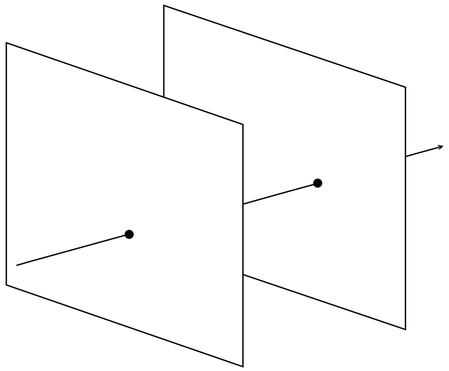


An Alternative Scene Representation

Conventional Light Field Parameterizations

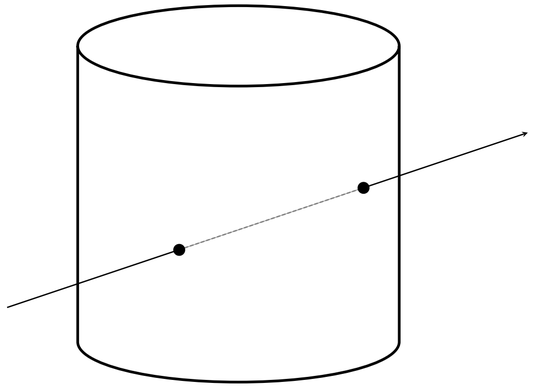


Two-Plane



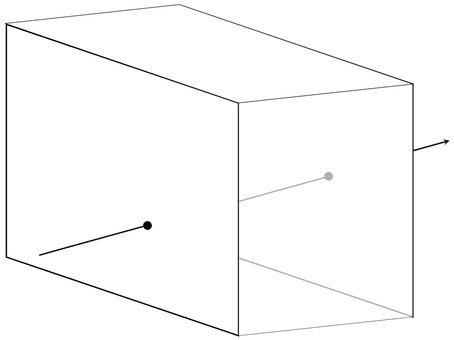
Not 360°

Cylindrical



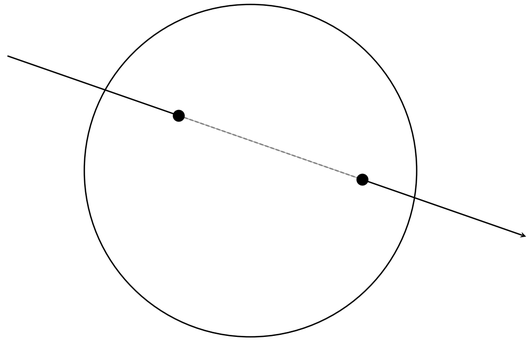
Not 360°

Lumigraph



Not Continuous

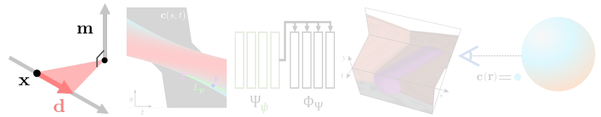
Two-Sphere



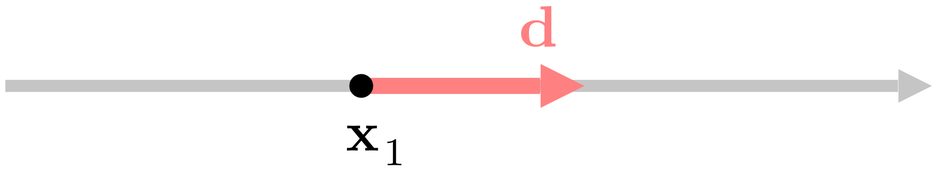
Bounded Scenes

Difficult to use as a complete scene representation

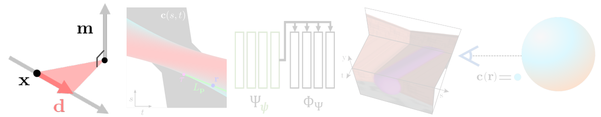
“Point-direction” coordinates



$$\mathbf{r} = (\mathbf{x}_1, \mathbf{d})$$



“Point-direction” coordinates



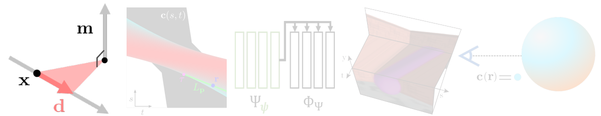
$$\mathbf{r} = (\mathbf{x}_1, \mathbf{d})$$

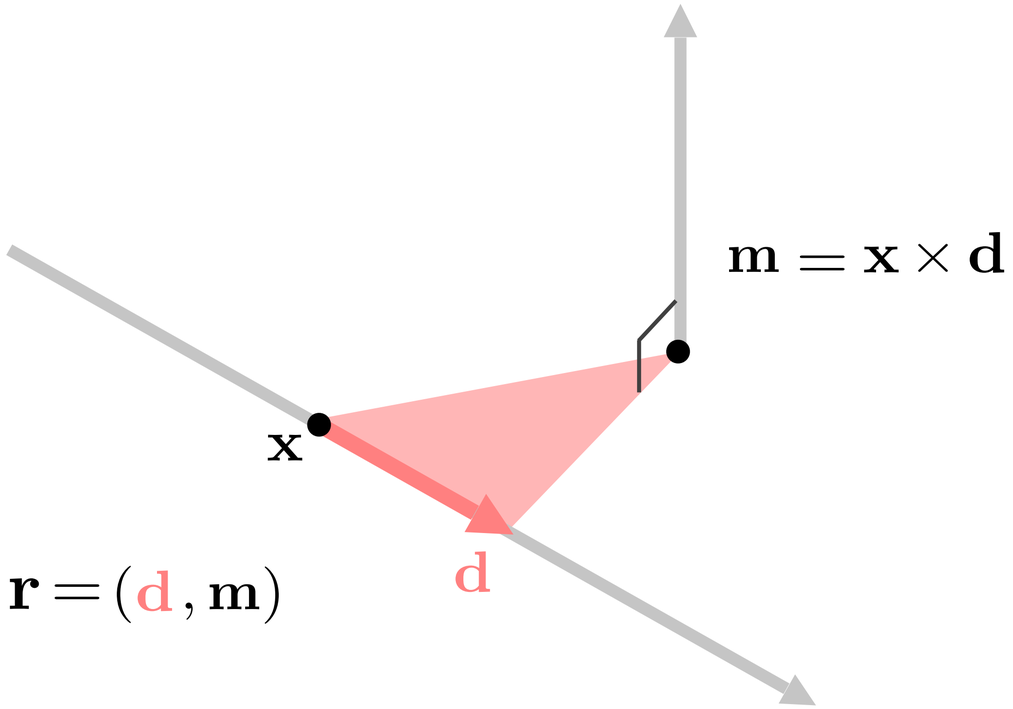
$$\neq$$

$$\mathbf{r} = (\mathbf{x}_2, \mathbf{d})$$

Not unique: Same ray, two different coordinates.

Plücker coordinates





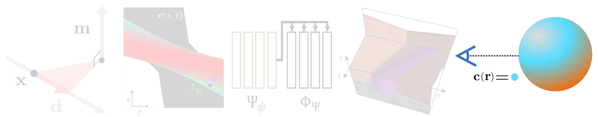
Invariant to choice of $\mathbf{x}$.

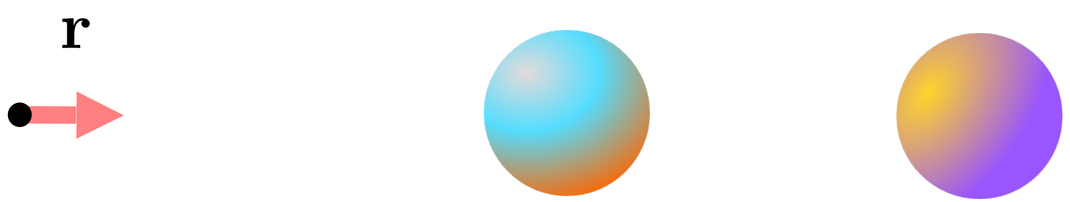
Parameterize all rays without special cases.

Impractical for discrete representations, since $\mathbf{r} \in \mathbb{R}^6$.

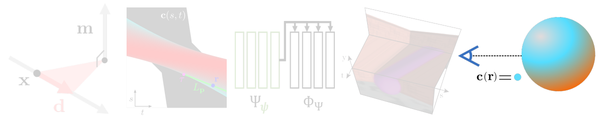
Invariant to choice of $\mathbf{x}$.

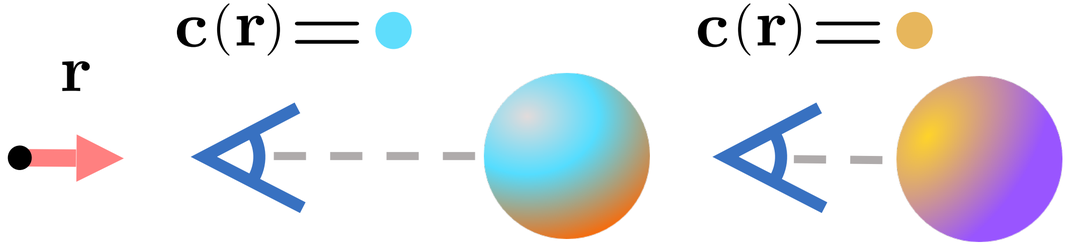
Impact of ray parameterization



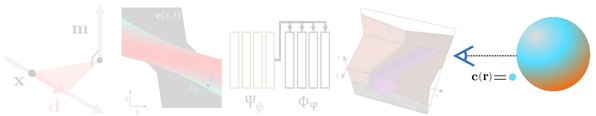


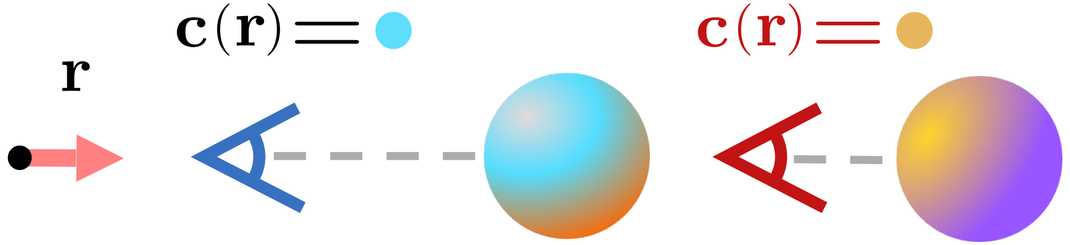
Limitations



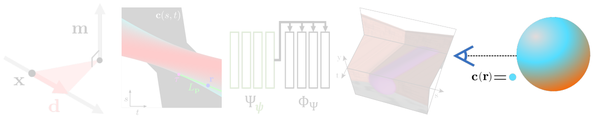


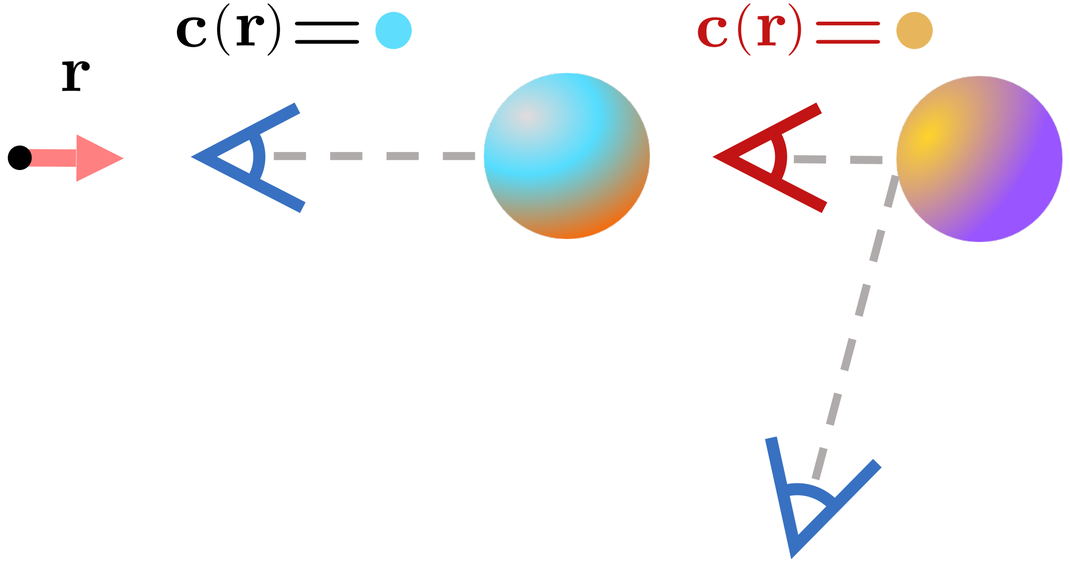
Limitations



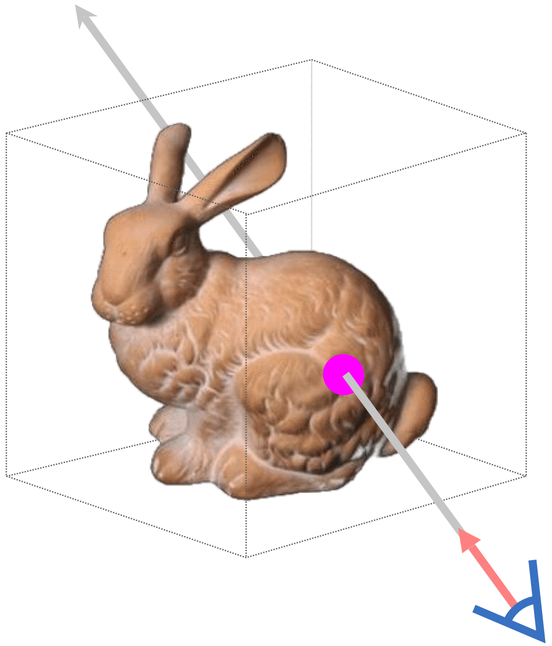


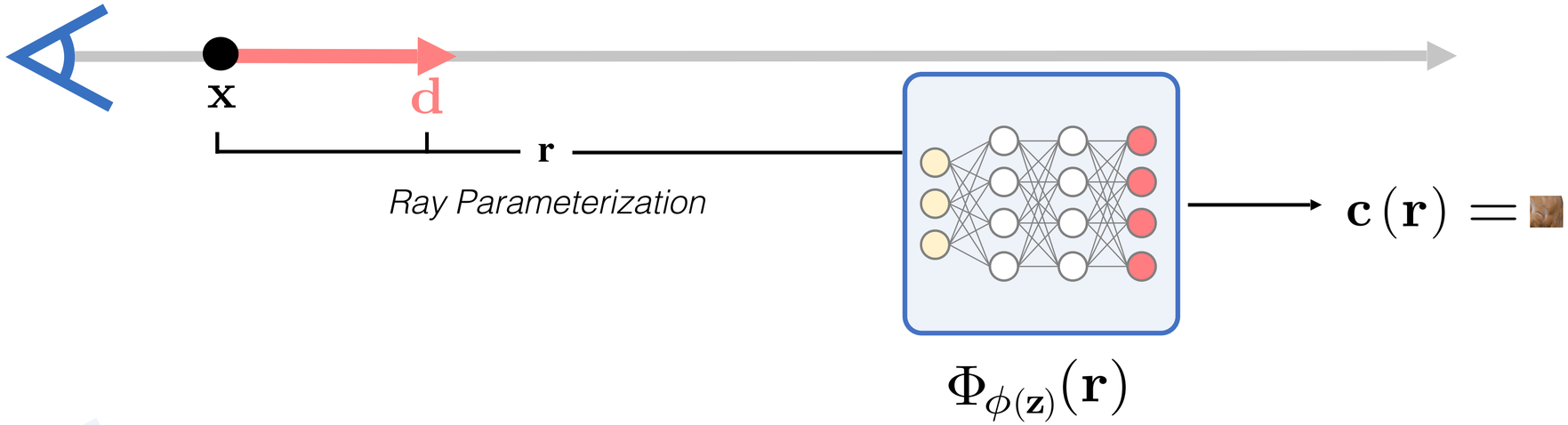
Limitations





Light Field Networks





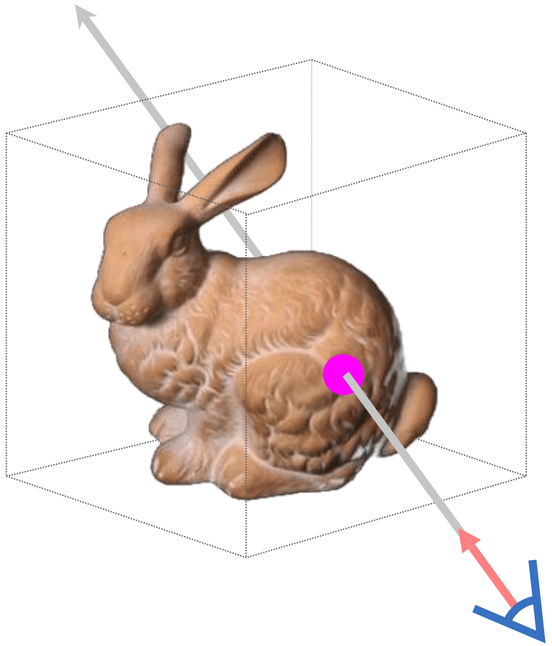
An Alternative Scene Representation

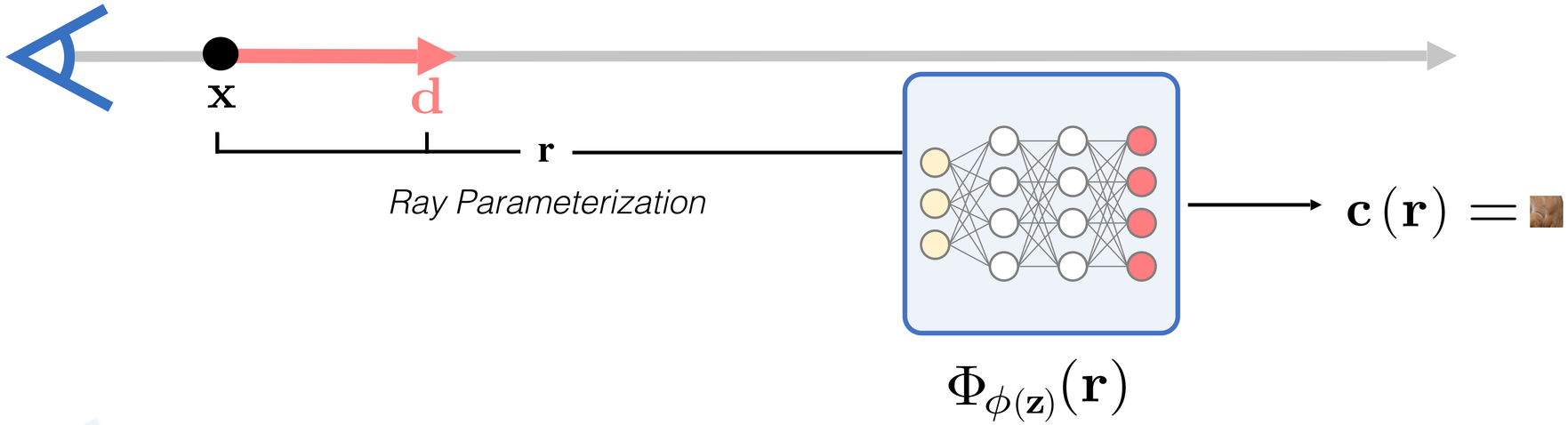
Overfitting doesn’t work out-of-the-box: No built-in multi-view consistency!

Context Views

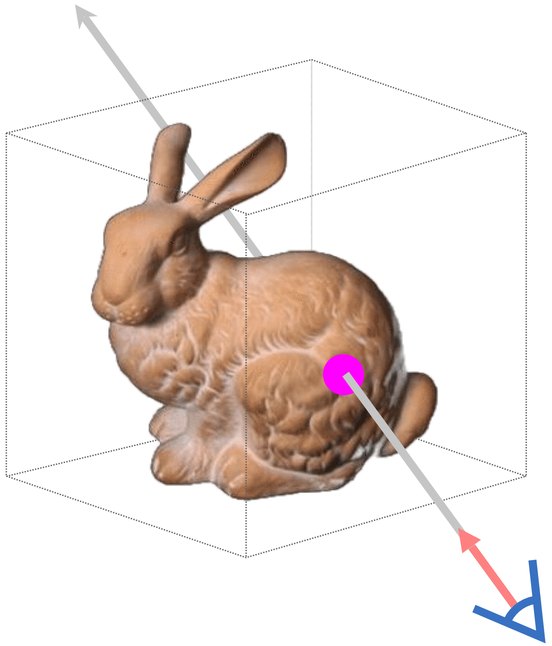
Intermediate Views

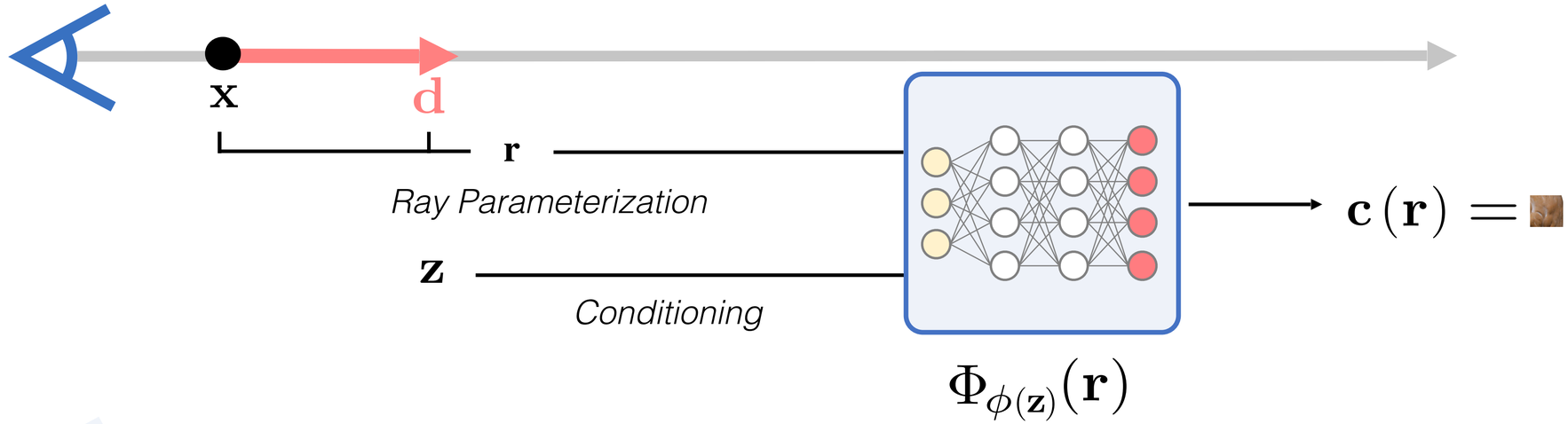
Light Field Networks





Now: *Learn Prior over Light Fields!*





Light Field Networks

(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$

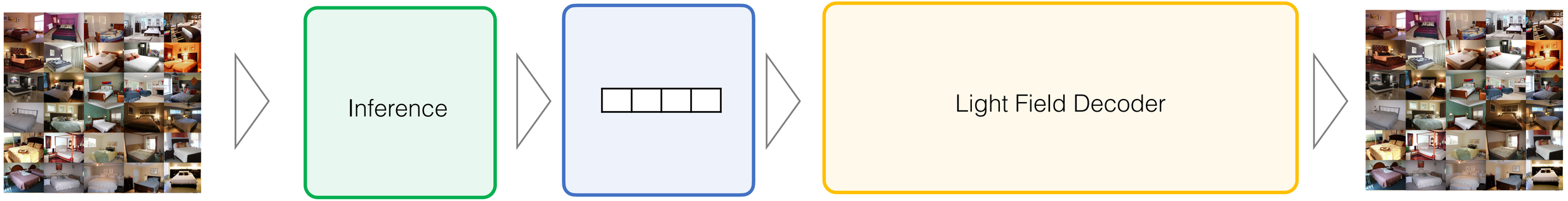
Inference

Decoder

Neural Renderer

Light Field Networks

(One or several)
Latent Codes $\{\mathbf{z}_i\}_{i=1}^{N}$



More difficult inference problem, but cheaper & maximally general renderer.

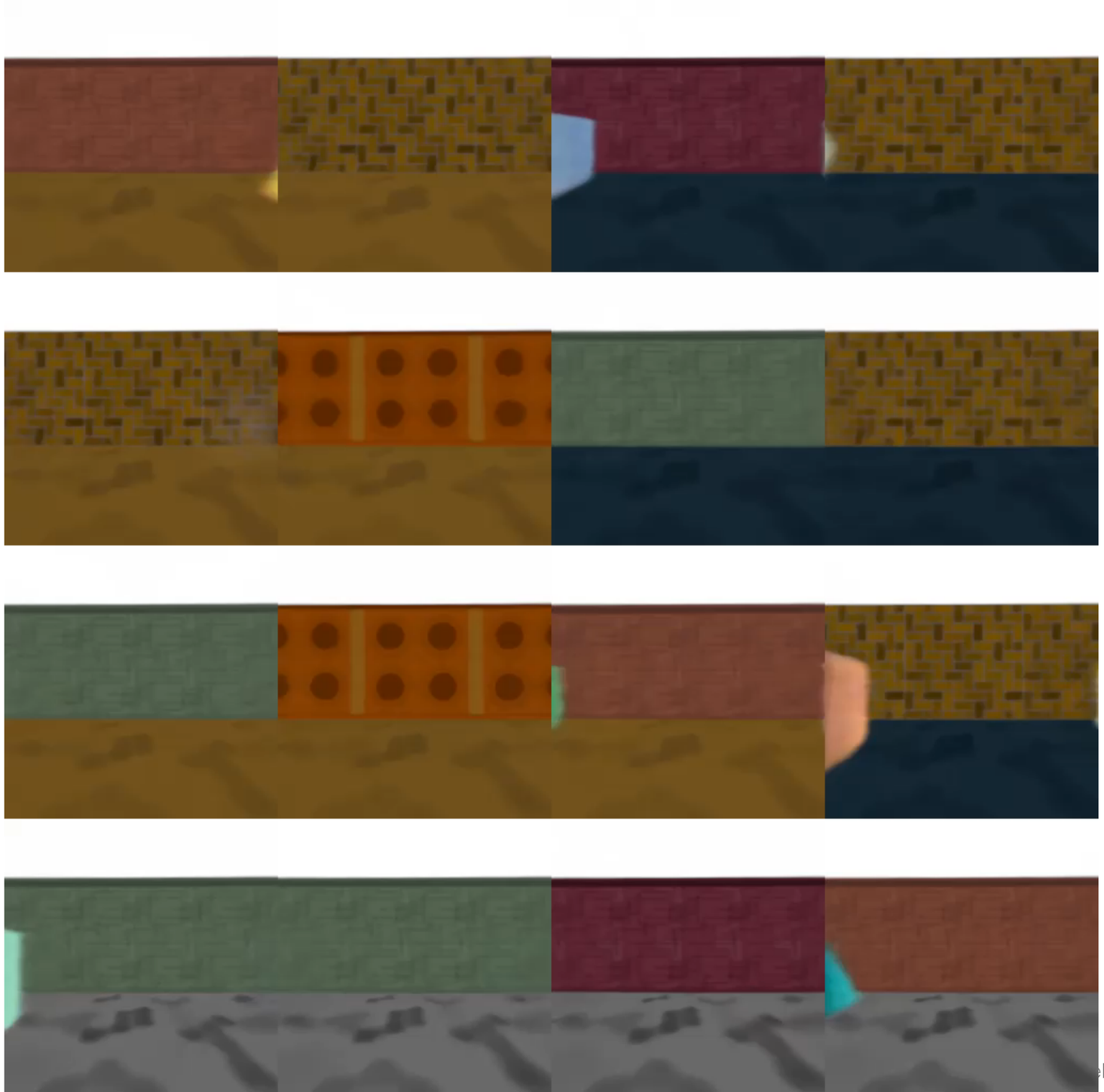
Light Field Networks
500 FPS
1 evaluation per ray

Volumetric Rendering (pixelNeRF)
0.033 FPS
196 evaluations per ray

100x speed

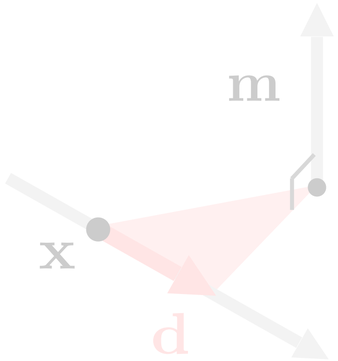
Real-time.
>100x reduction in memory: Can be trained on small GPUs!

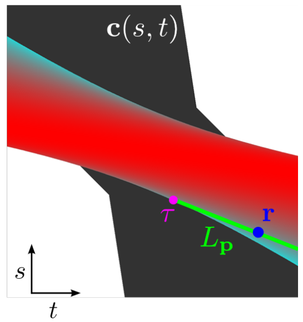
Light Field Networks
500 FPS
1 evaluation per ray

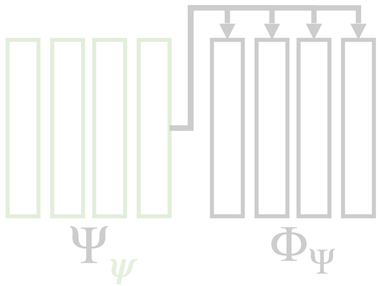


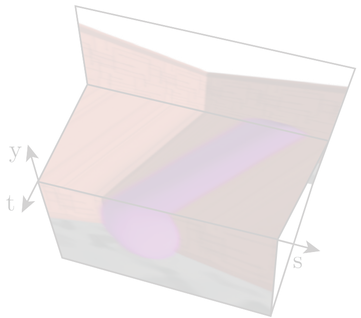
How LFNs encode 3D geometry

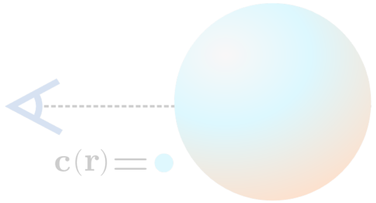
Parameterization | LFN Geometry | Meta-Learning | Results | Limitations



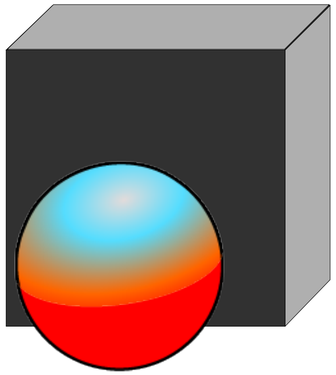




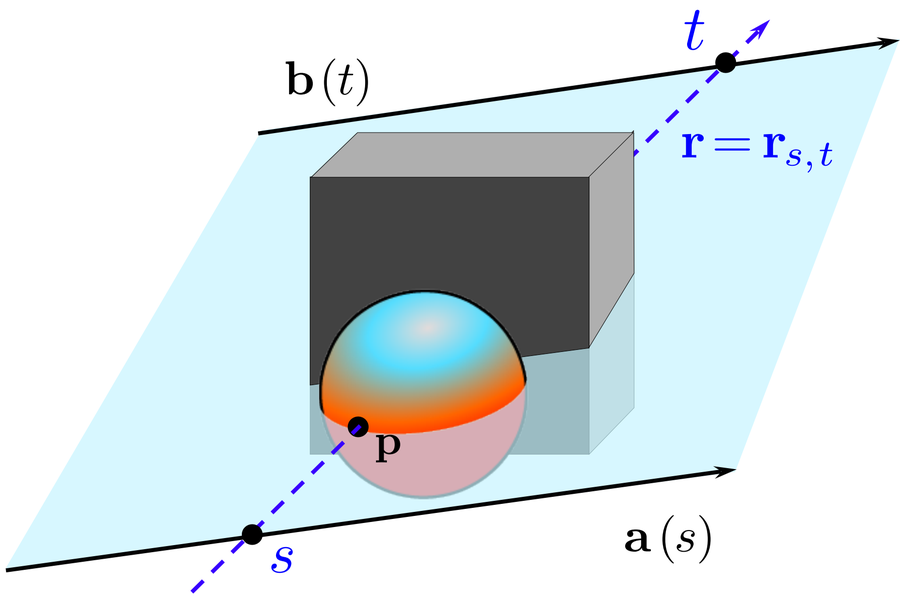




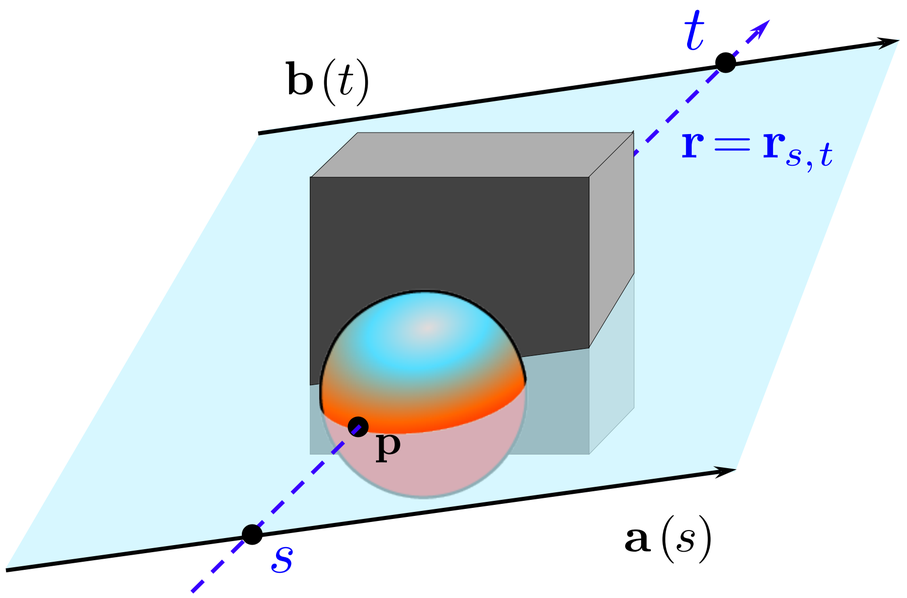
The geometry of LFNs

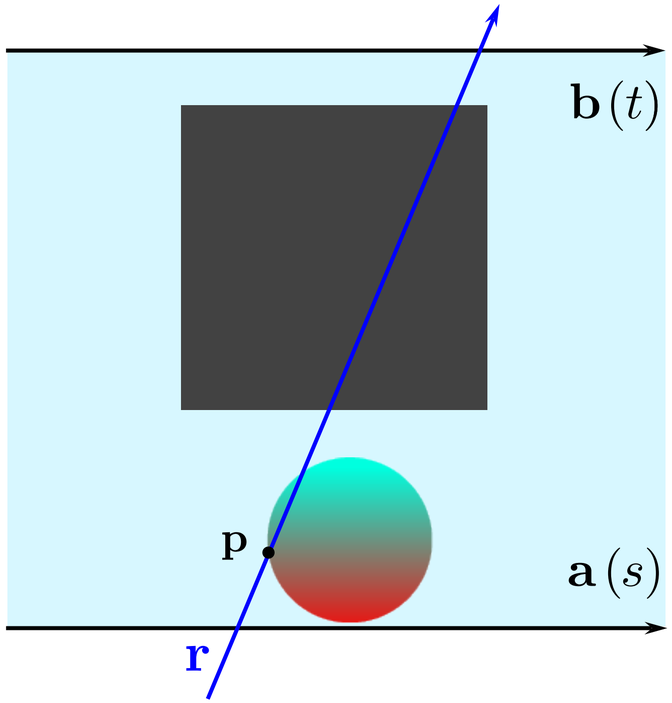


The geometry of LFNs

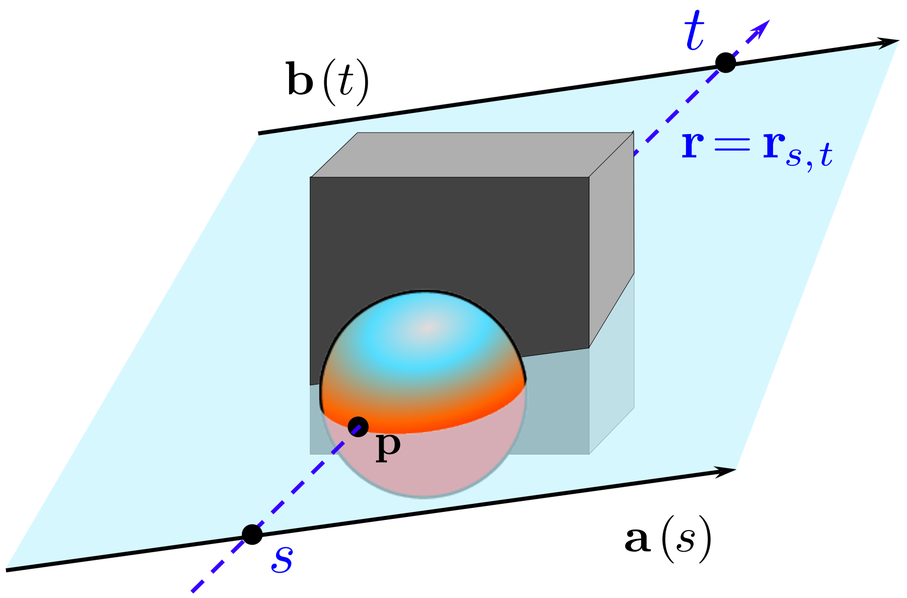


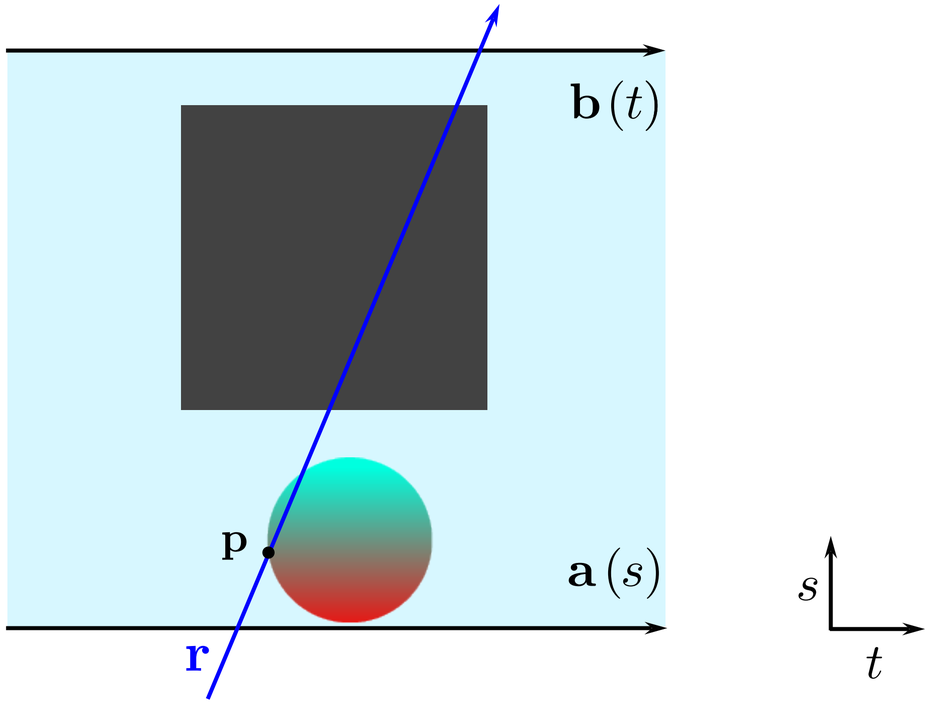
The geometry of LFNs



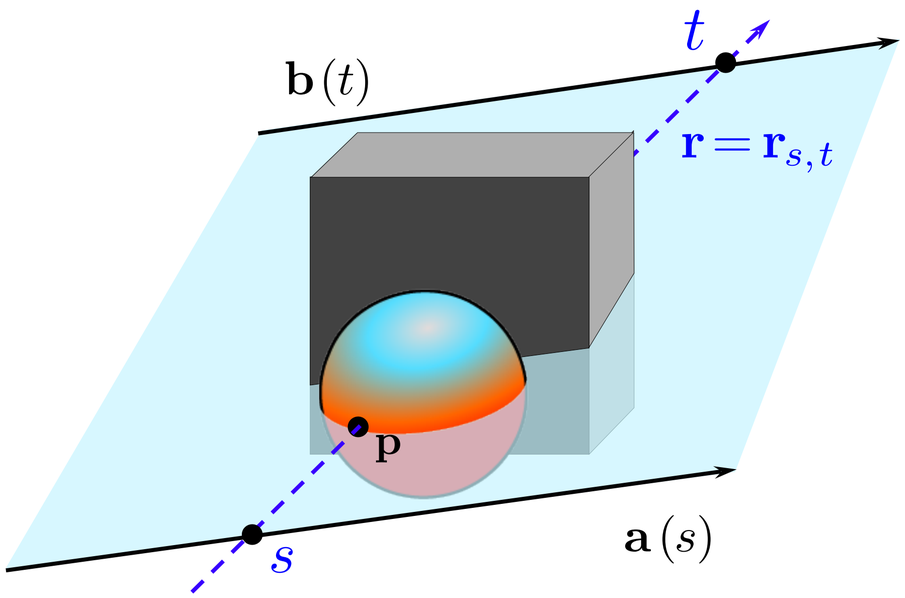


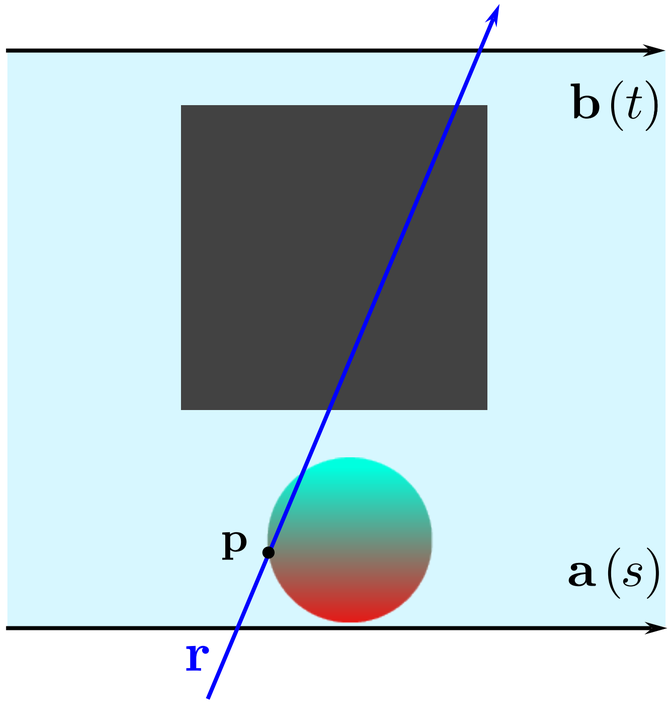
The geometry of LFNs

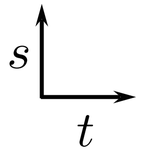




The geometry of LFNs

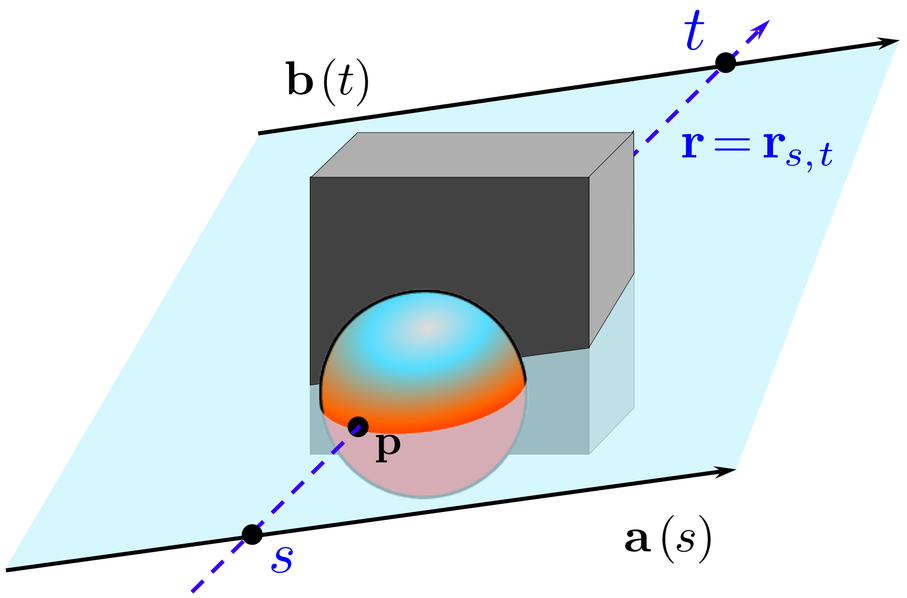


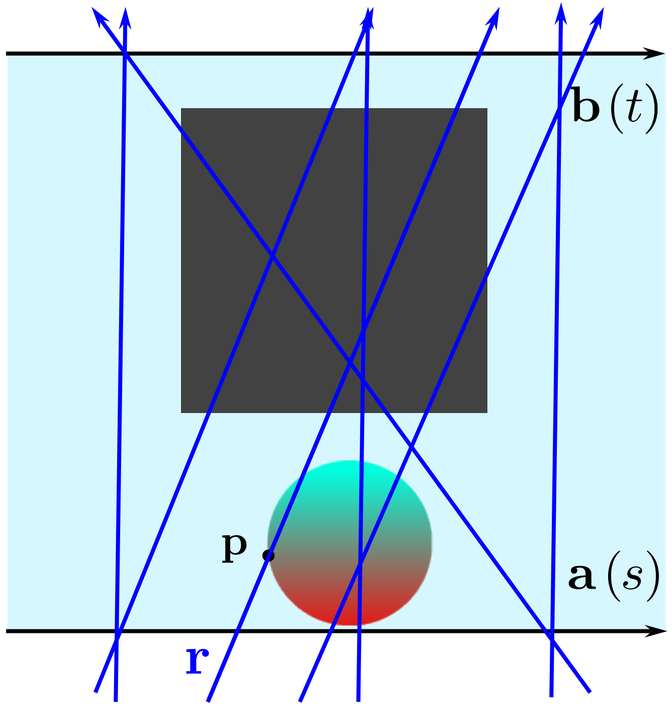


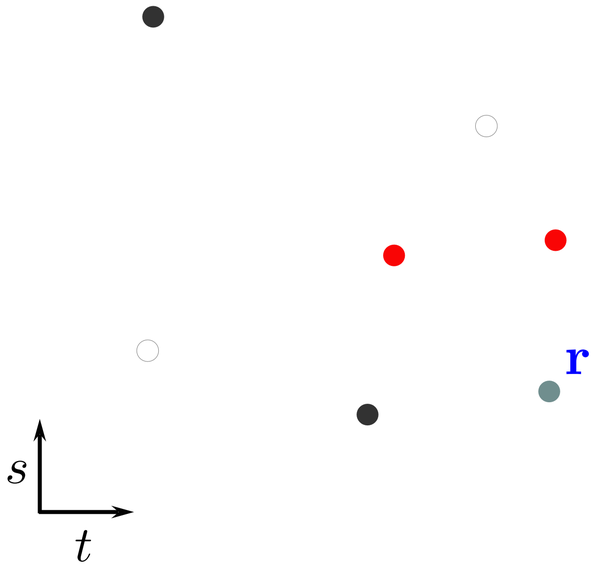


$\mathbf{r}$

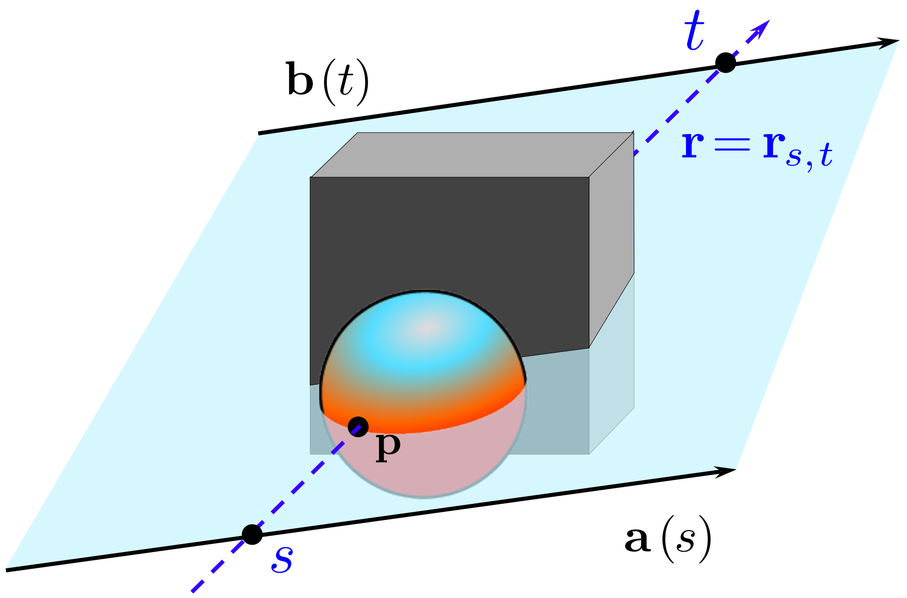
The geometry of LFNs

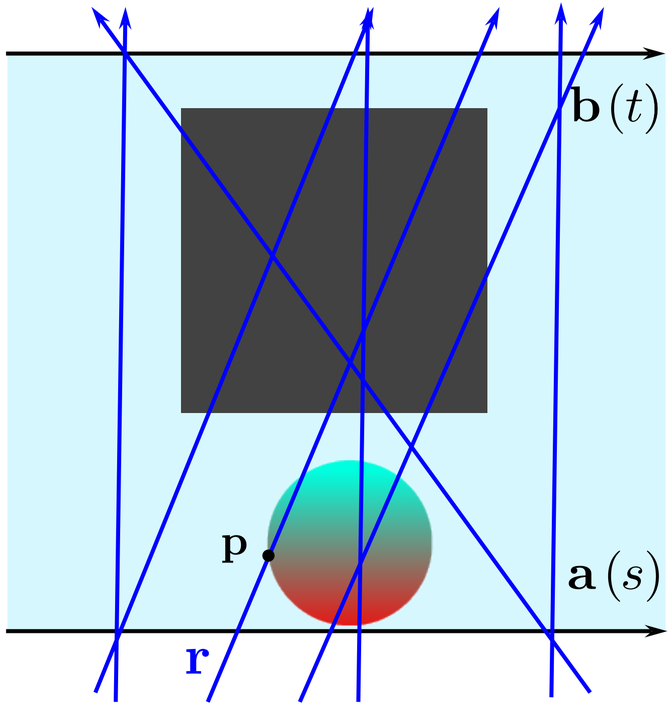




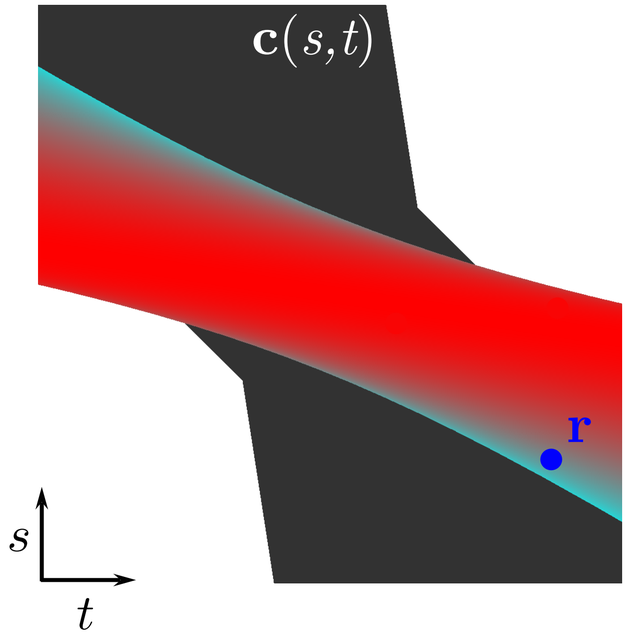


The geometry of LFNs

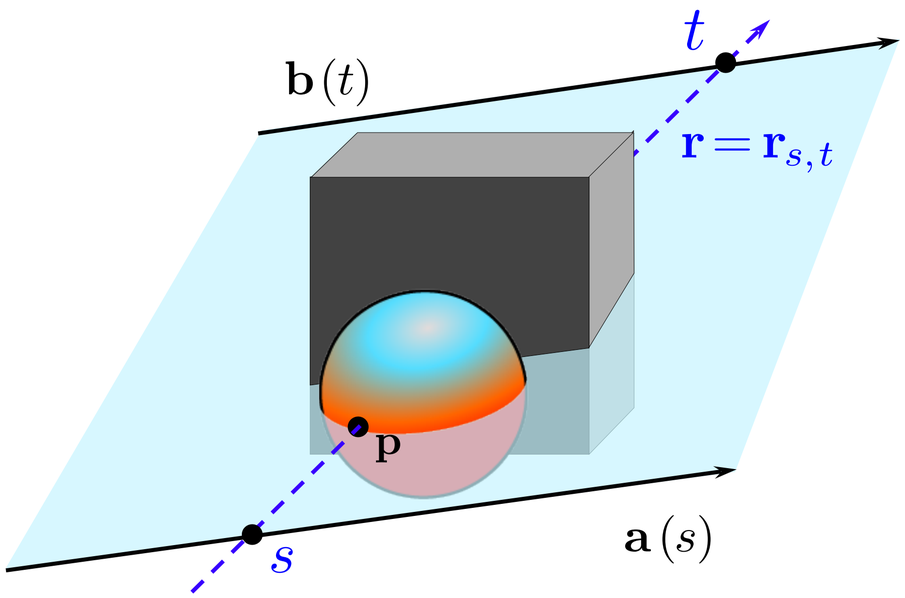


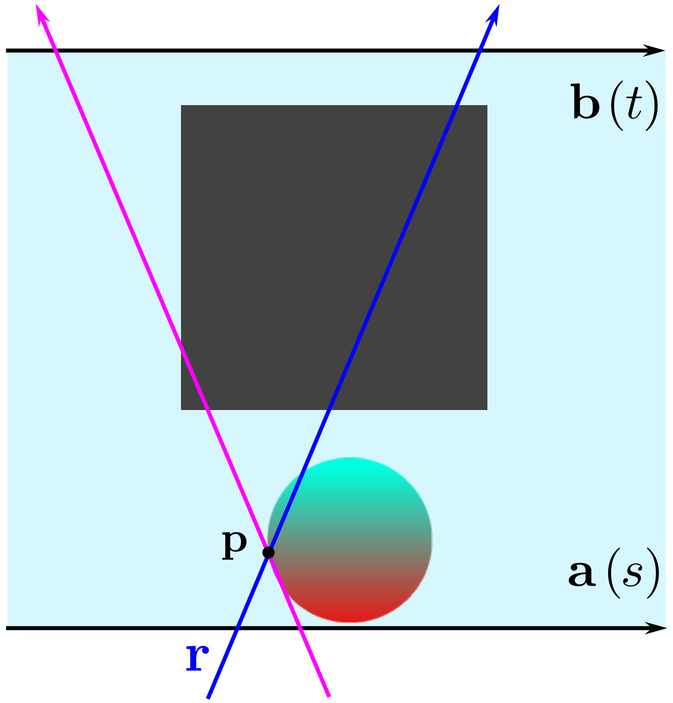


Epipolar Plane Image

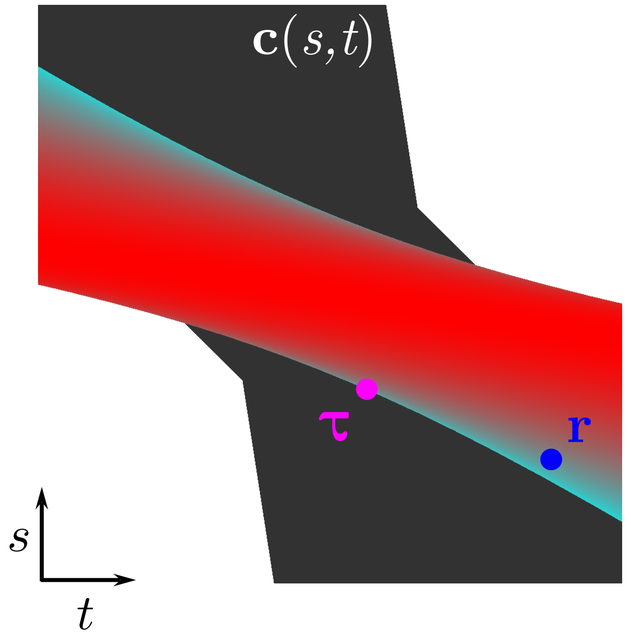


The geometry of LFNs

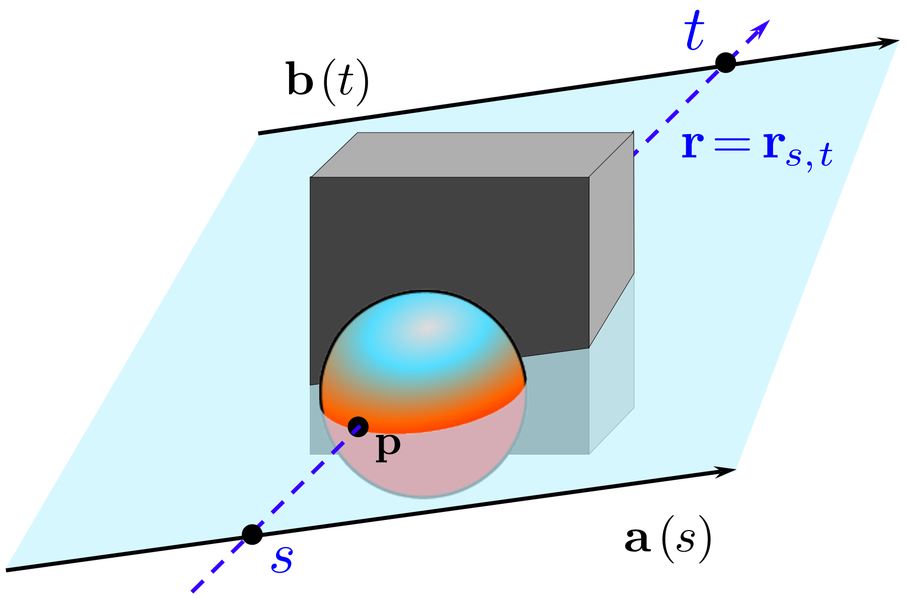


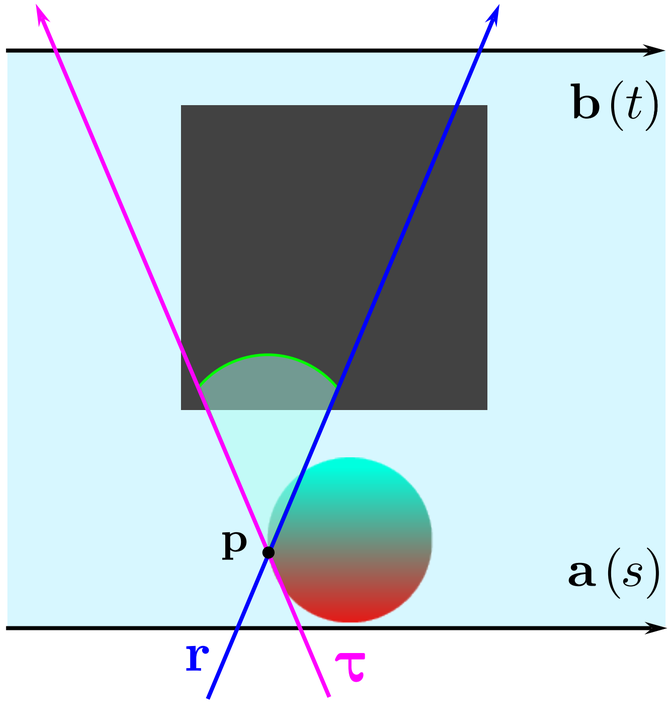


Epipolar Plane Image

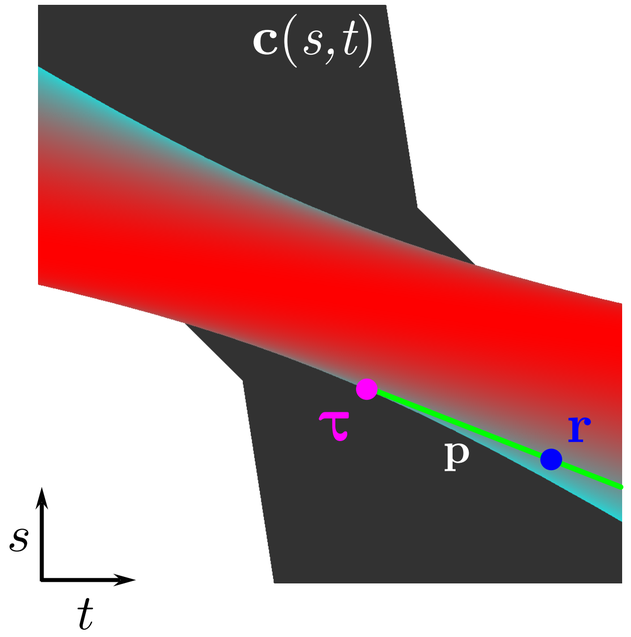


The geometry of LFNs

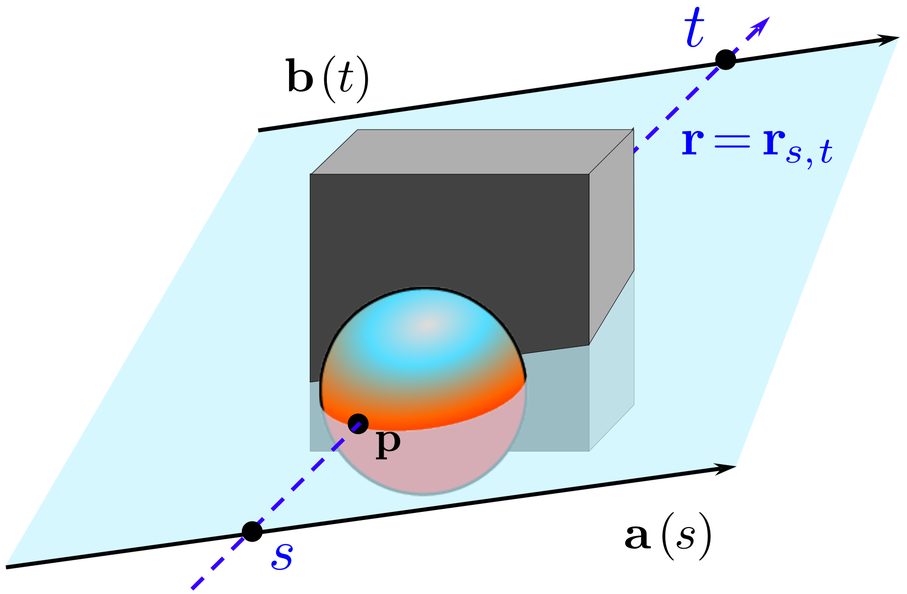


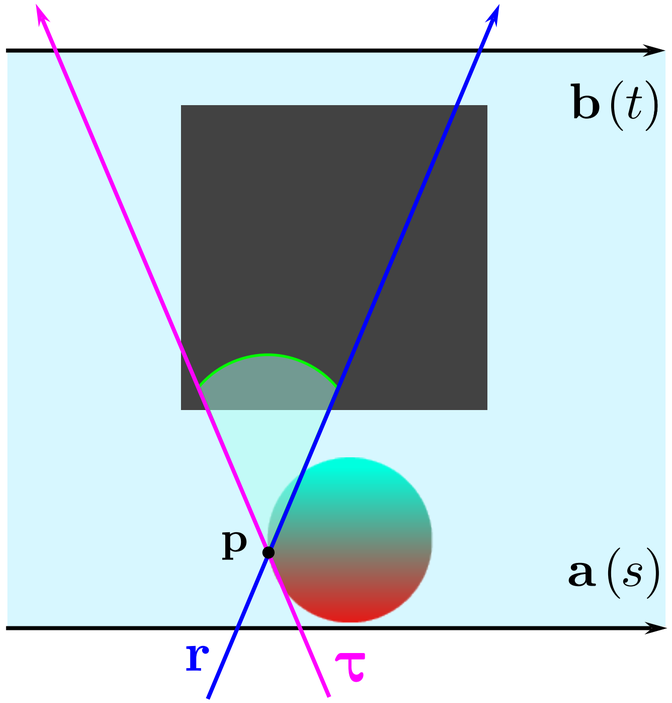


Epipolar Plane Image

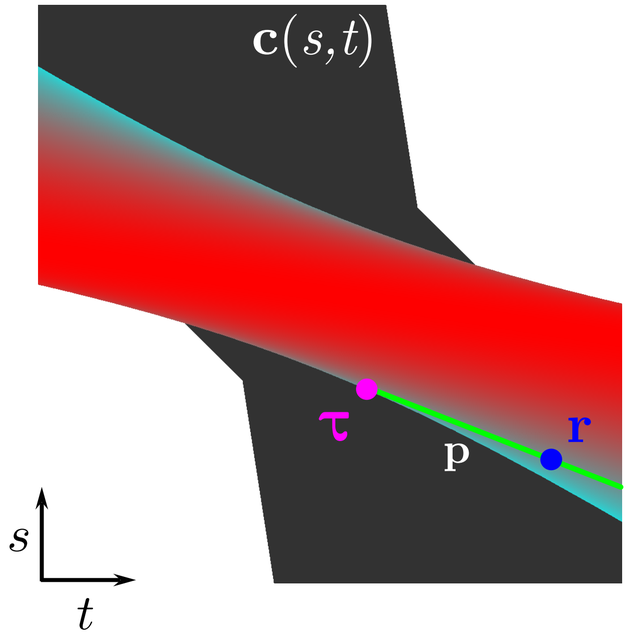


The geometry of LFNs



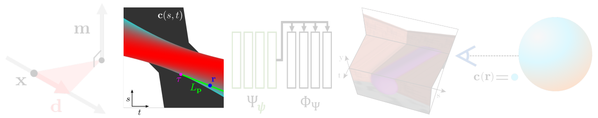


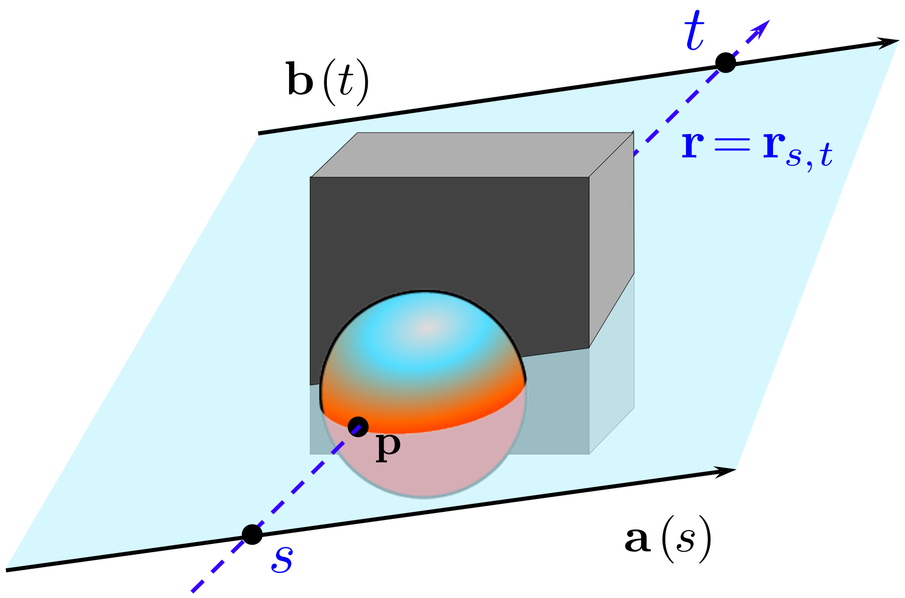
Epipolar Plane Image

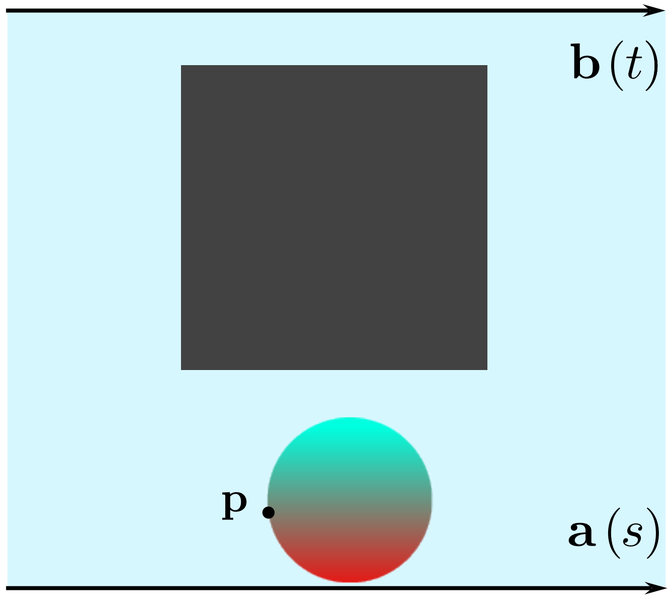


Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

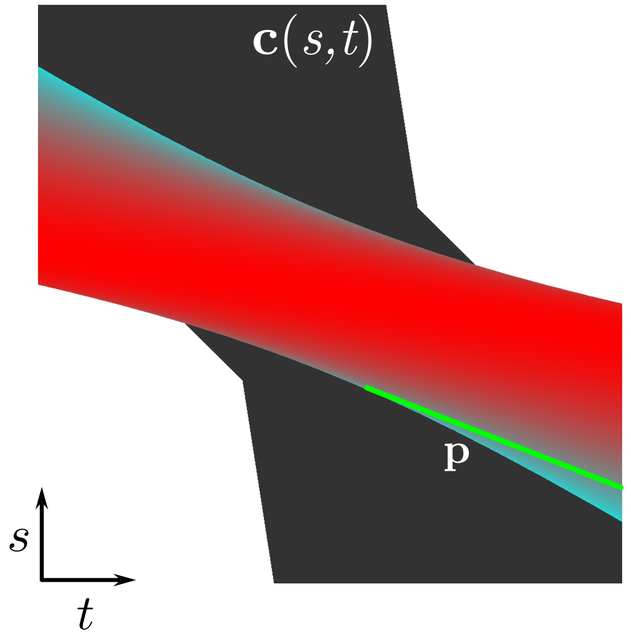
The geometry of LFNs







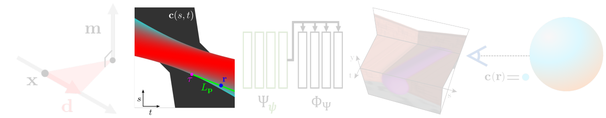
Epipolar Plane Image

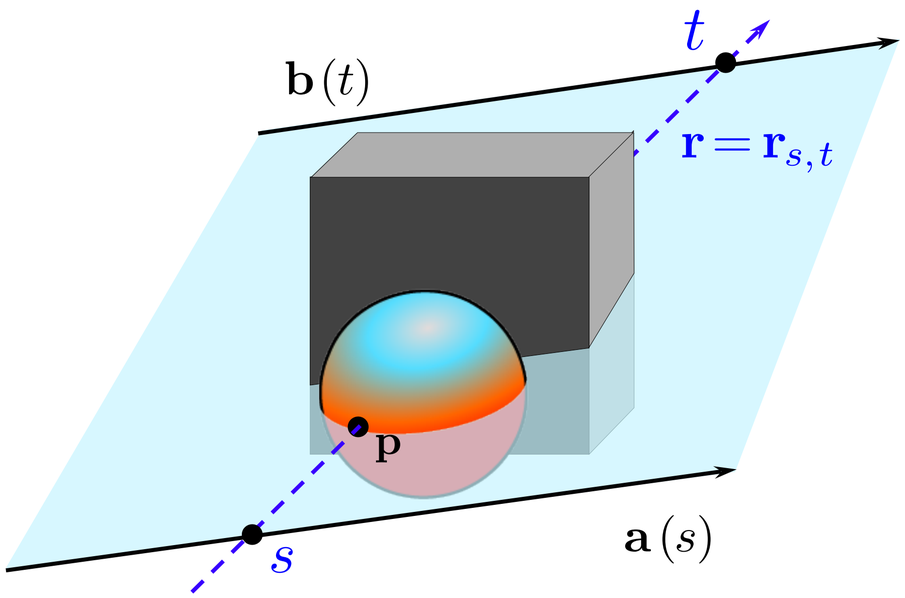


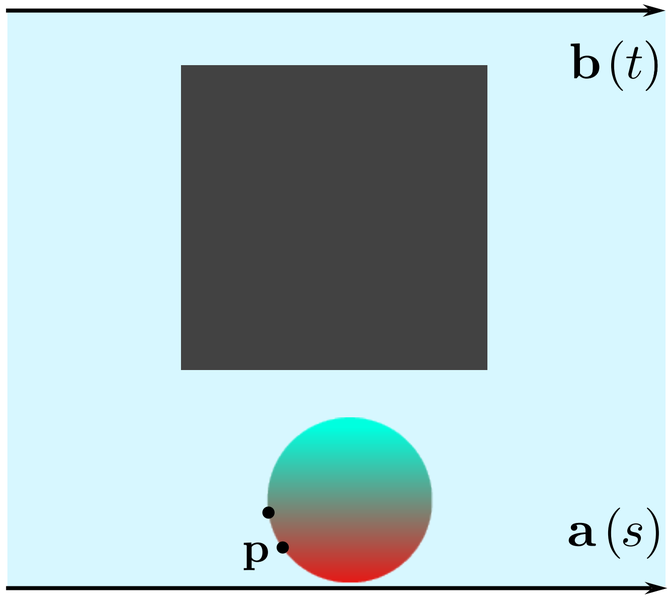
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

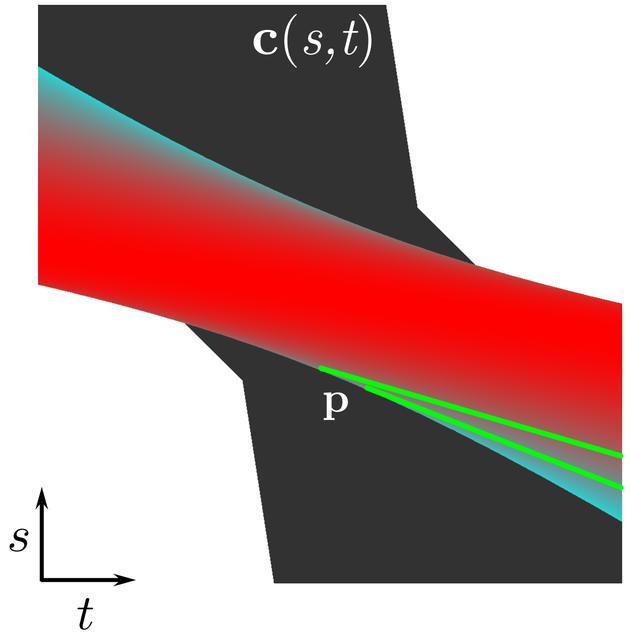
The geometry of LFNs







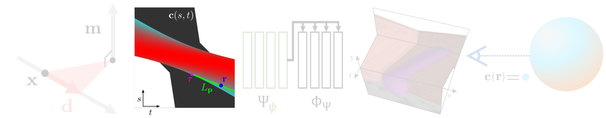
Epipolar Plane Image

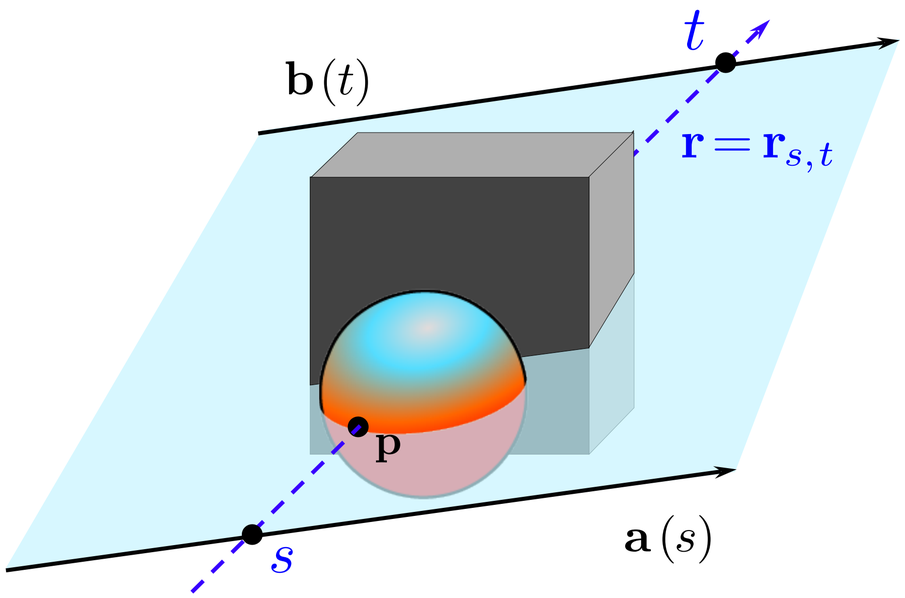


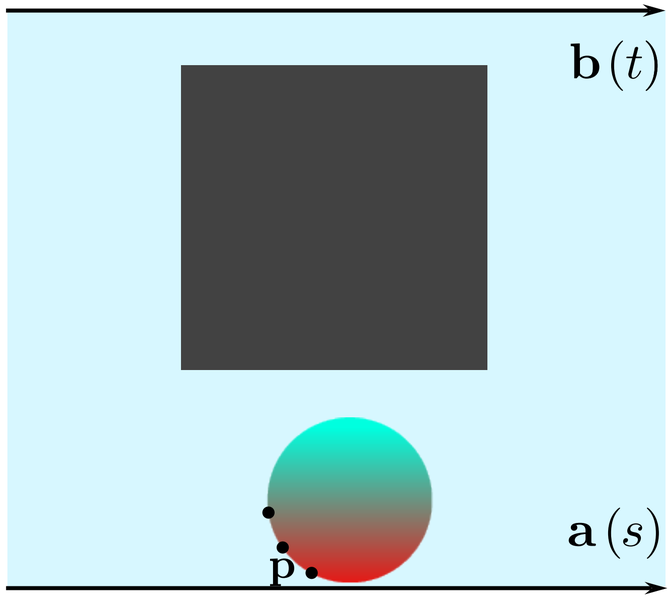
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

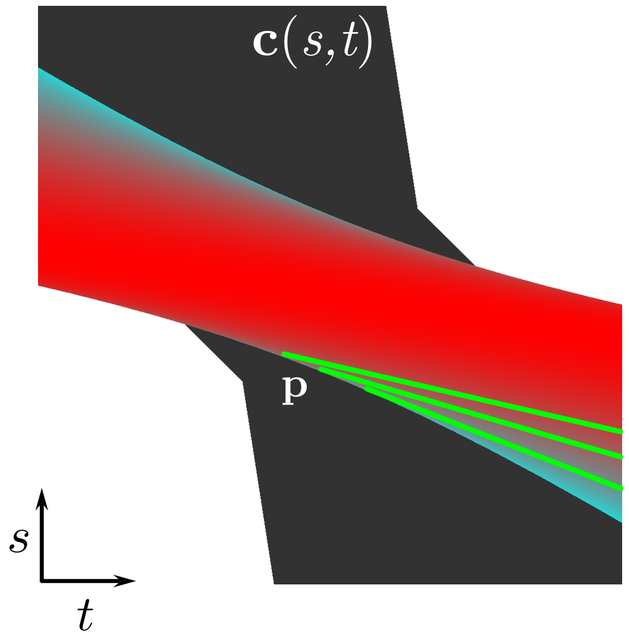
The geometry of LFNs







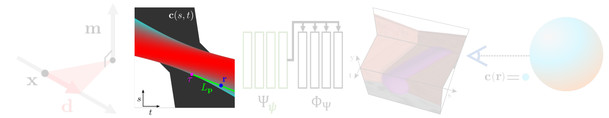
Epipolar Plane Image

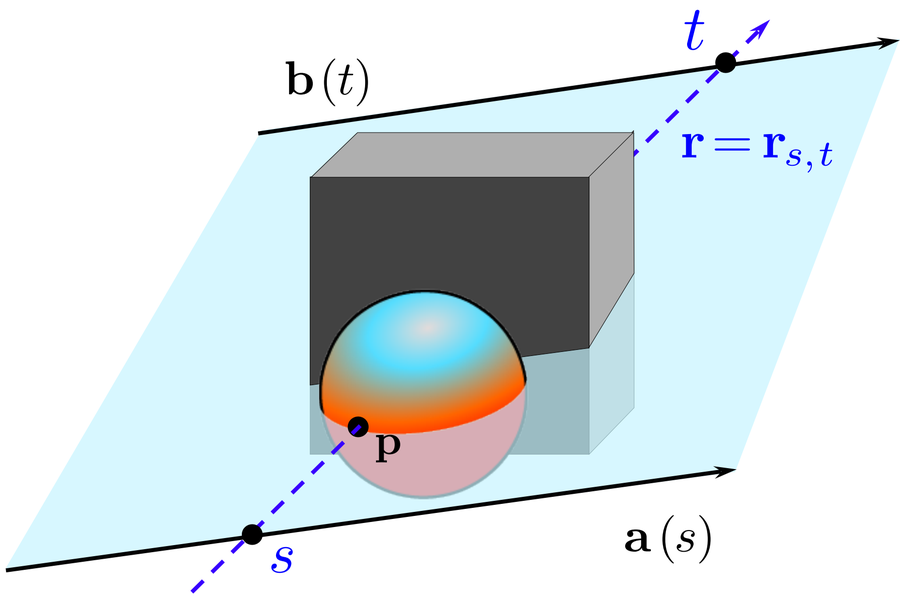


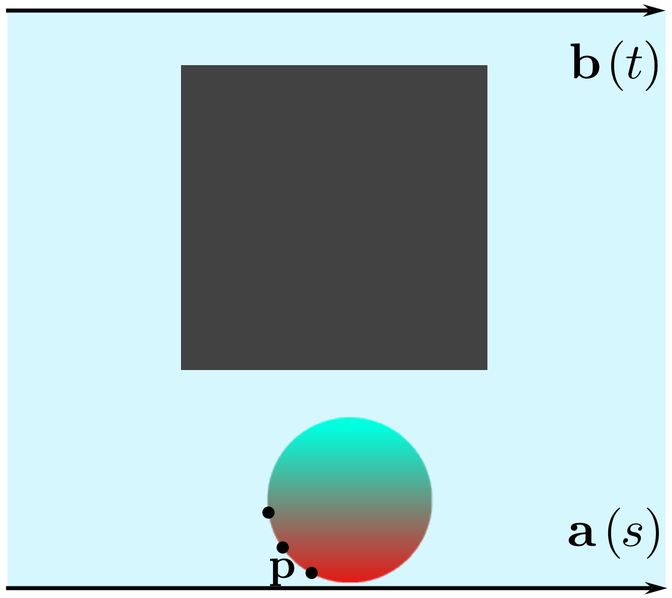
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

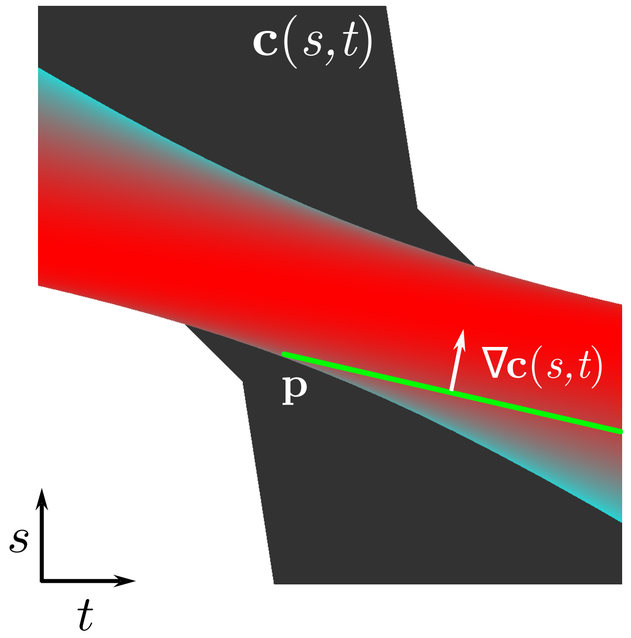
The geometry of LFNs







Epipolar Plane Image

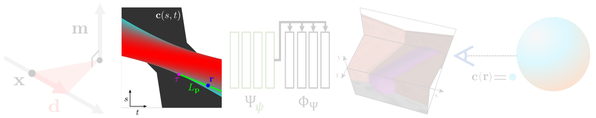


Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

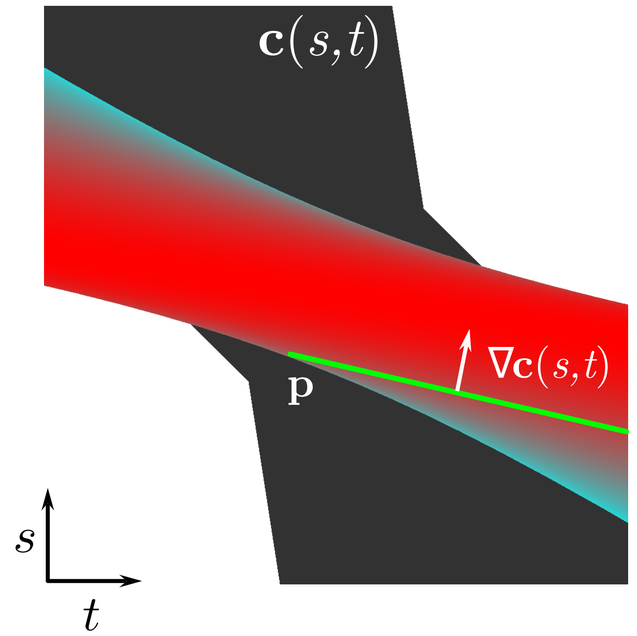
Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset -

The geometry of LFNs



$$d(s,t) = D\frac{\partial_t \mathbf{c}(s,t)}{\partial_s \mathbf{c}(s,t) + \partial_t \mathbf{c}(s,t)}$$

Epipolar Plane Image

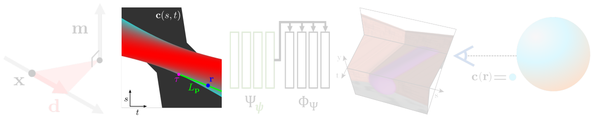


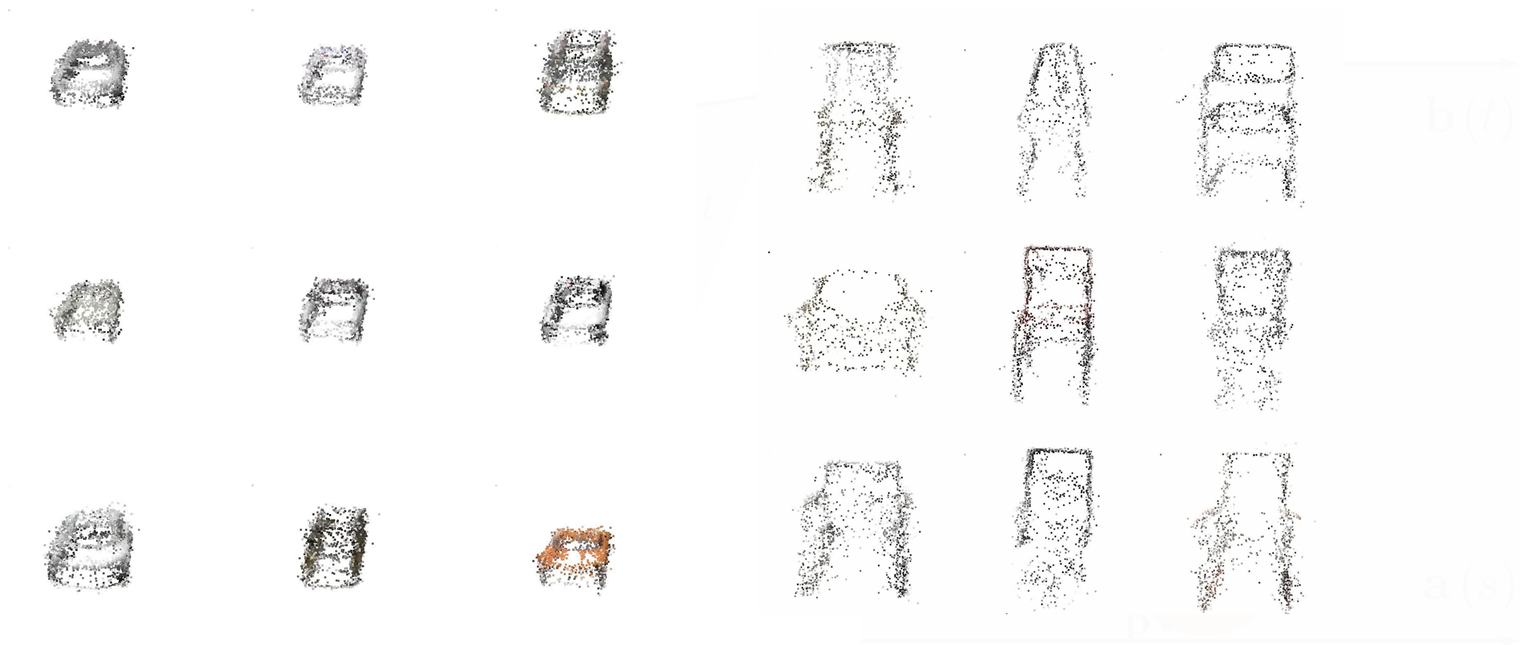
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

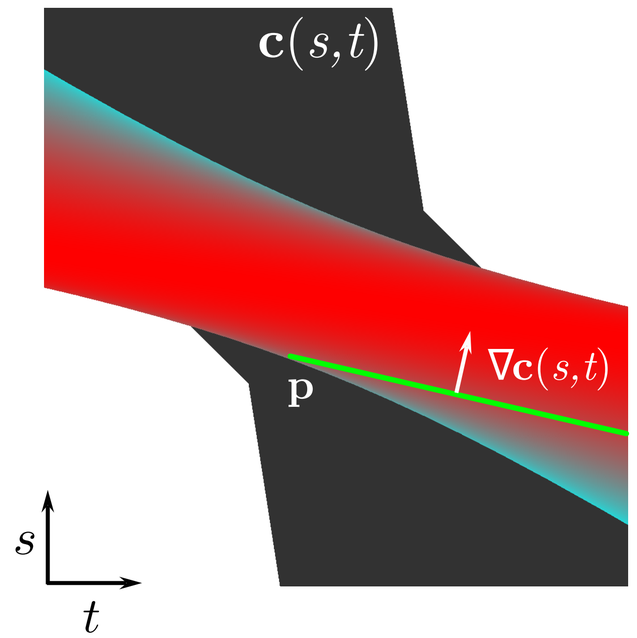
Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset - can extract depth from gradients of light field.

The geometry of LFNs





Epipolar Plane Image

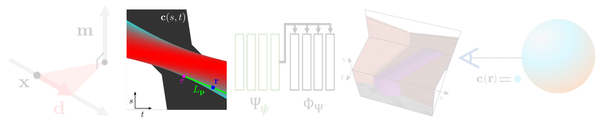


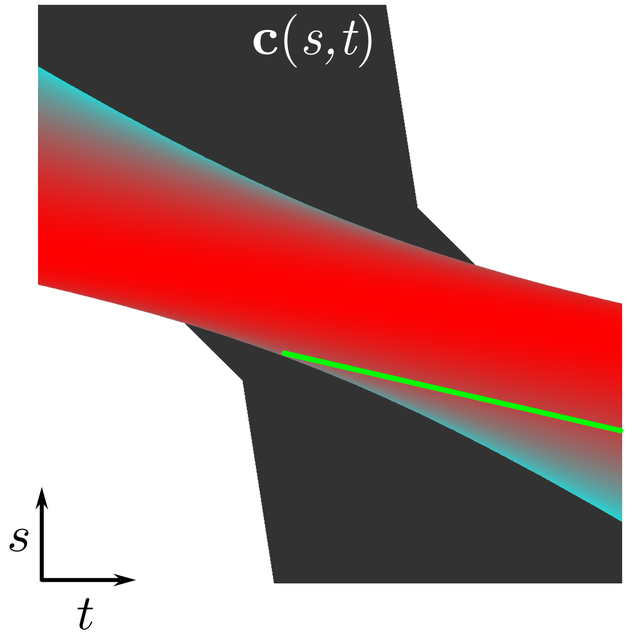
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

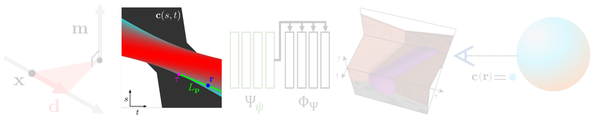
Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset - can extract depth from gradients of light field.

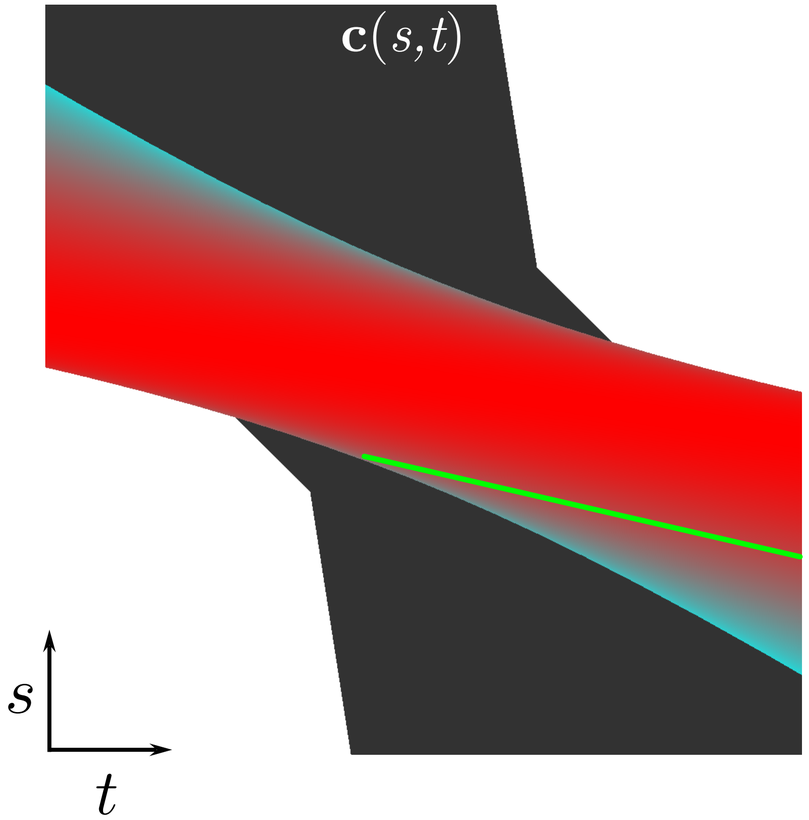
Multi-view consistency





Multi-view consistency





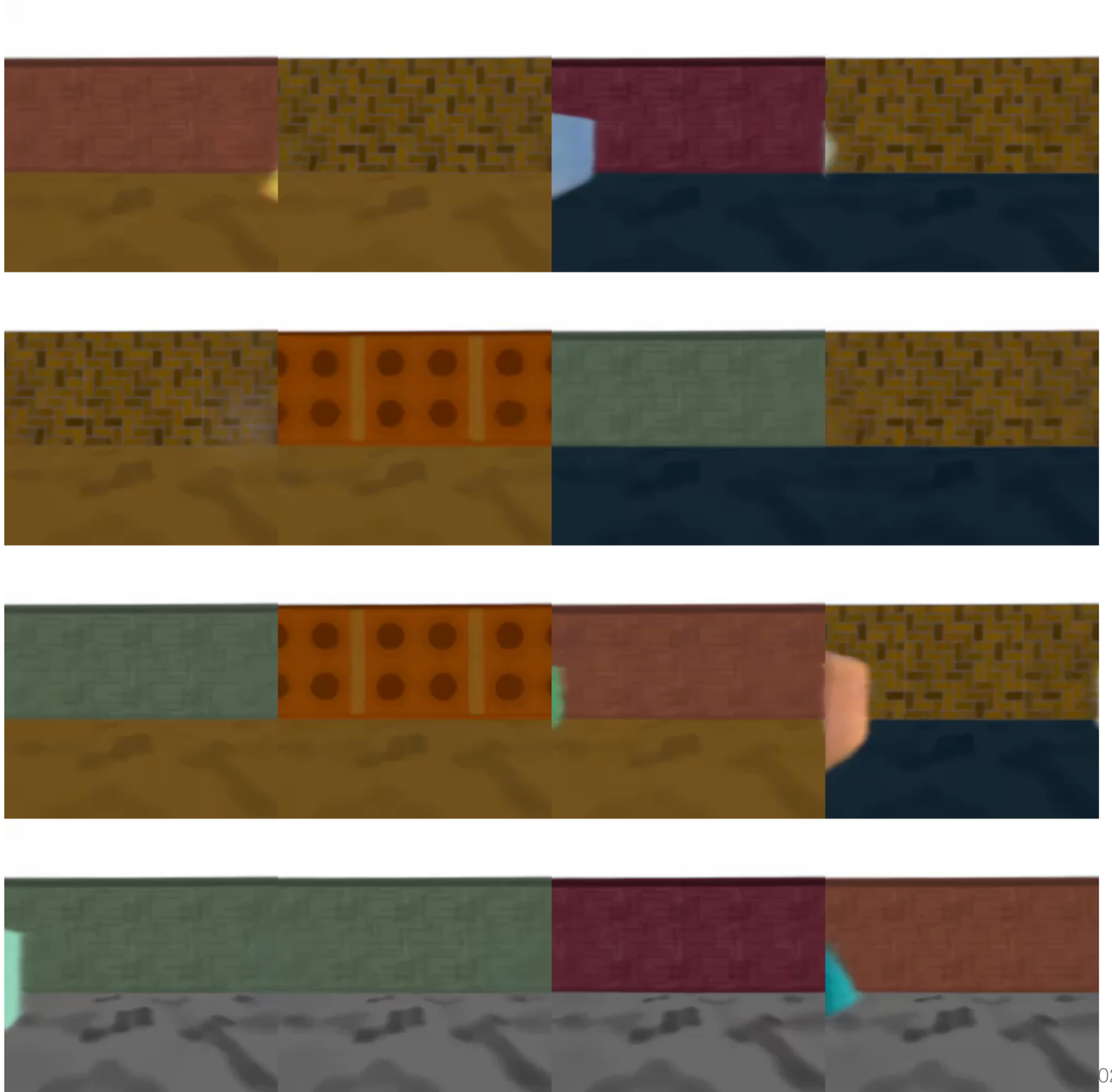
Light Field Networks
500 FPS
1 evaluation per ray

Volumetric Rendering (pixelNeRF)
0.033 FPS
196 evaluations per ray

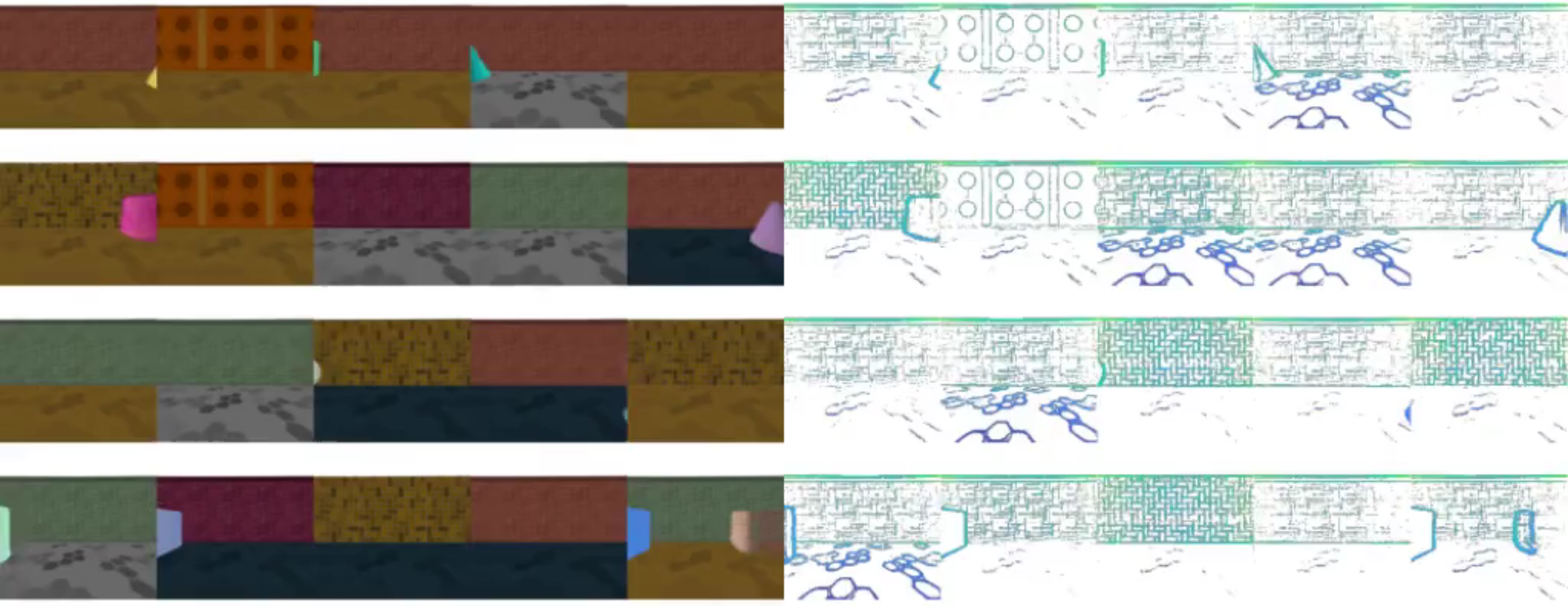
1x speed

Real-time. No post-processing, no discrete data structures (octrees, voxelgrids, …).
>100x reduction in memory: Can be trained on small GPUs!

Light Field Networks
500 FPS
1 evaluation per ray



Also Encode Depth in their 4D derivatives:
can be extracted via single evaluation of neural network and its gradient!



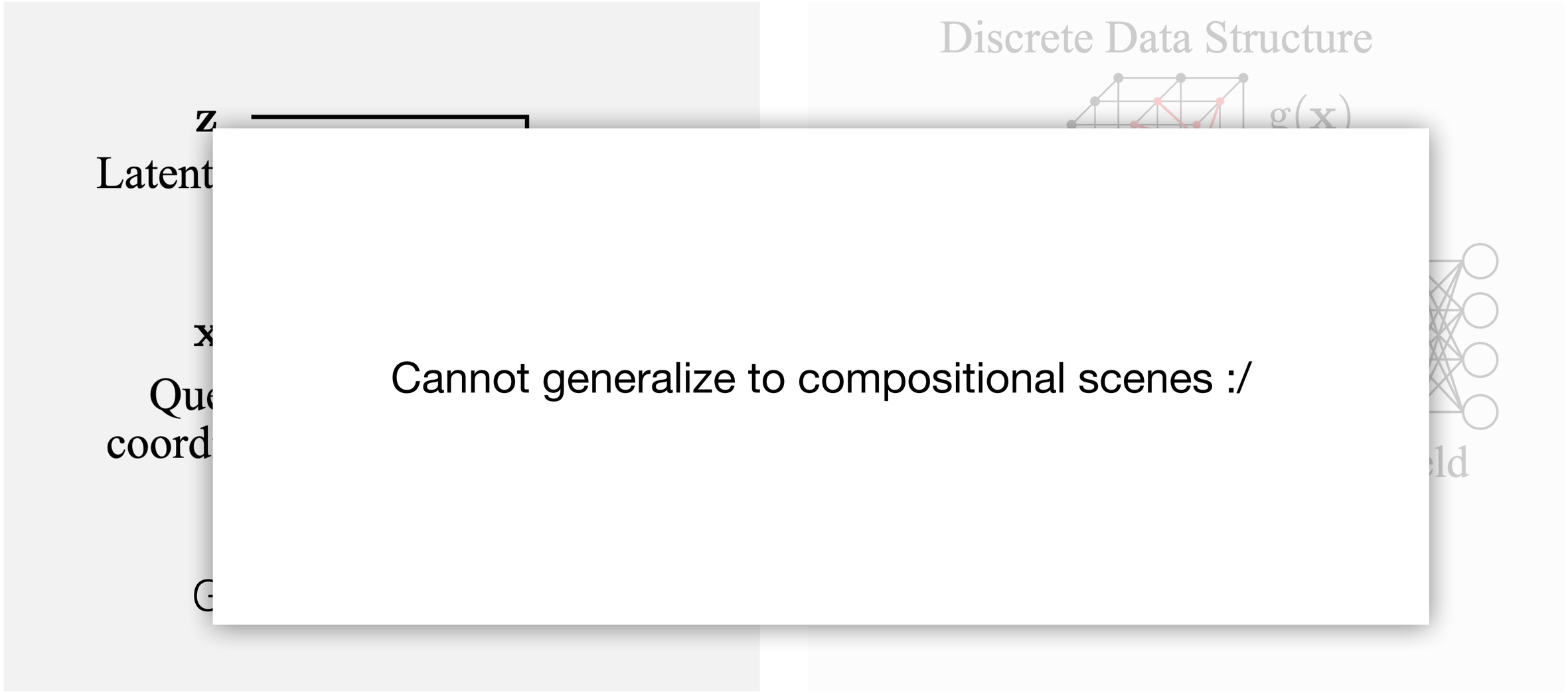
Outperforms global conditioning

	Chairs	Cars
SRNs [3]	**22.89** / 0.89	22.25 / **0.89**
LFN	22.26 / **0.90**	**22.42** / **0.89**

Does not outperform local conditioning

	Class-specific	Multi-Class
LFN	22.34 / 0.90	24.95 / 0.87
pixelNeRF	**23.45 / 0.91**	**26.80 / 0.91**

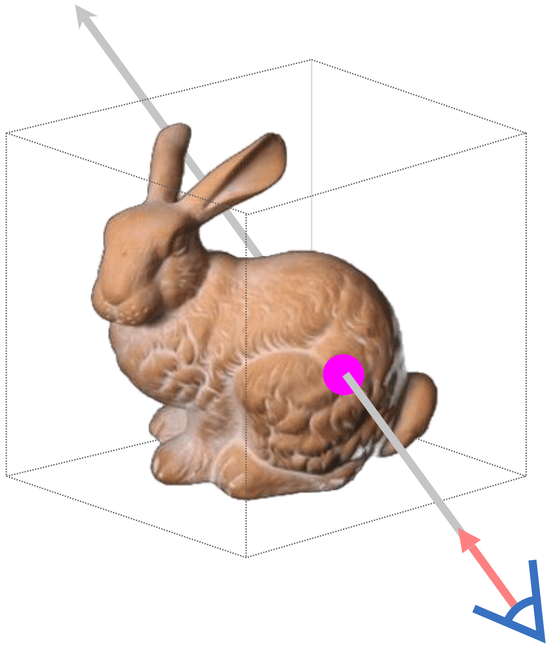
Global Conditioning: Single Latent Code for whole 3D Scene



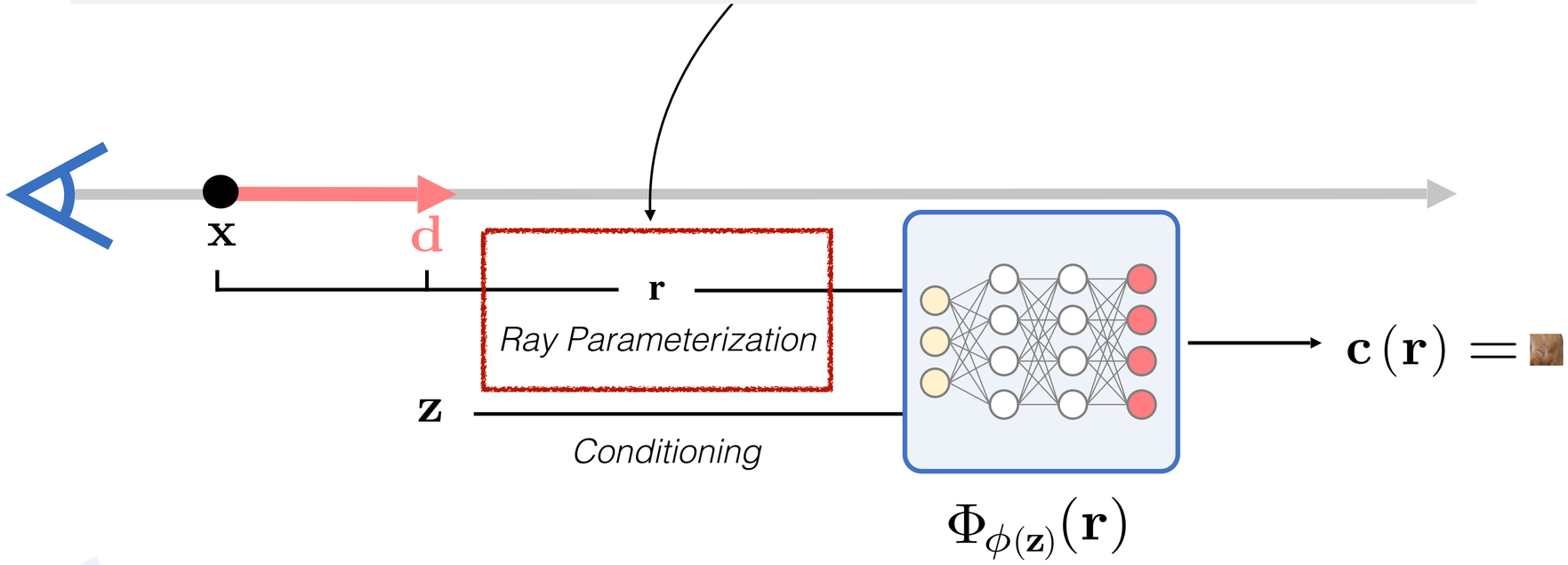
Cannot generalize to compositional scenes :/

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

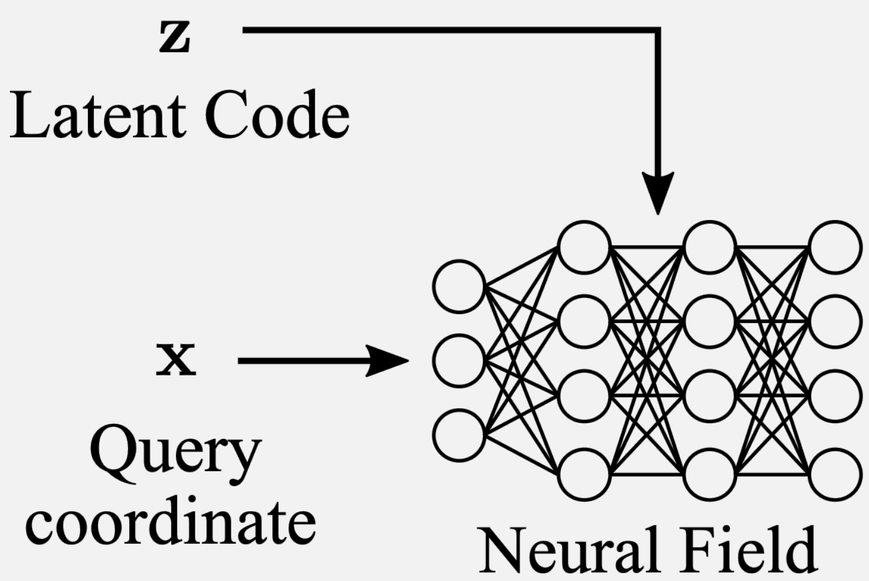
Now: *Learn Prior over Light Fields!*



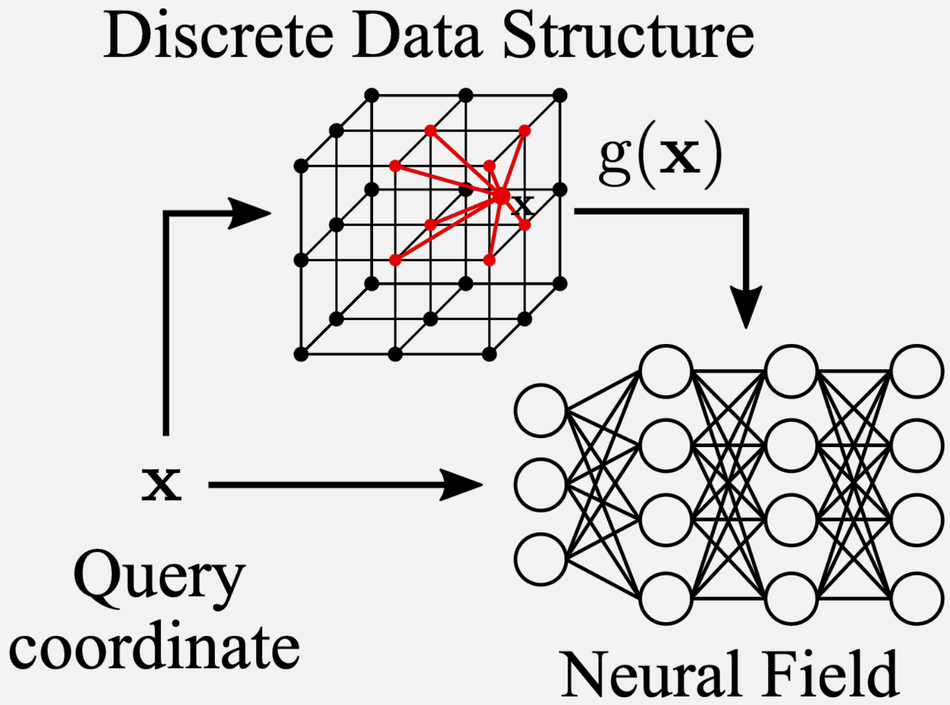
Five-dimensional: Hybrid Discrete-Continuous intractable



Recap: Global Conditioning and Local Conditioning



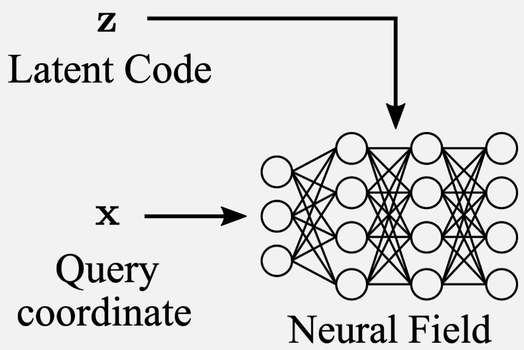
Global Conditioning



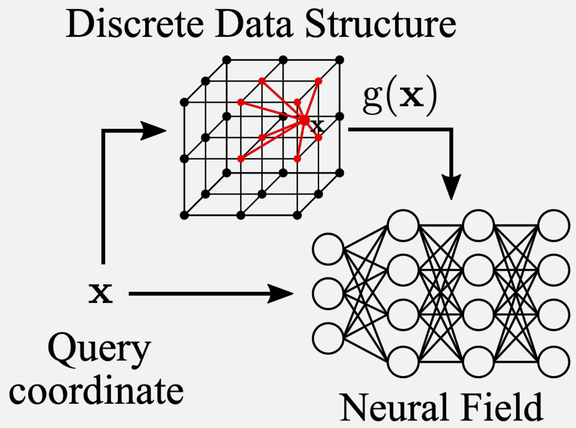
Local Conditioning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

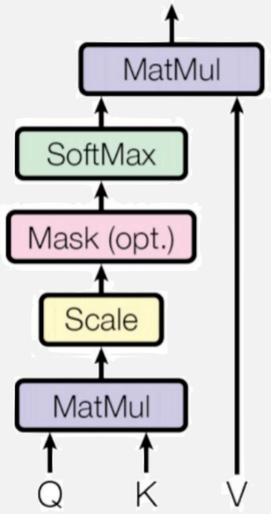
Other types of conditioning…?



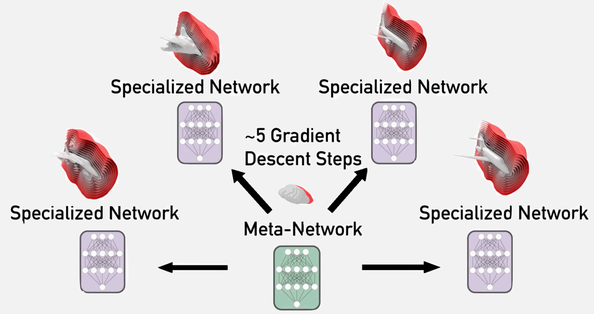
Global Conditioning



Local Conditioning



Attention-Based Conditioning



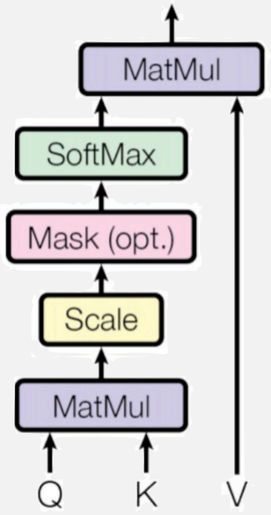
Gradient-Based Meta-Learning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Other types of conditioning…?

Global Conditioning

Local Conditioning

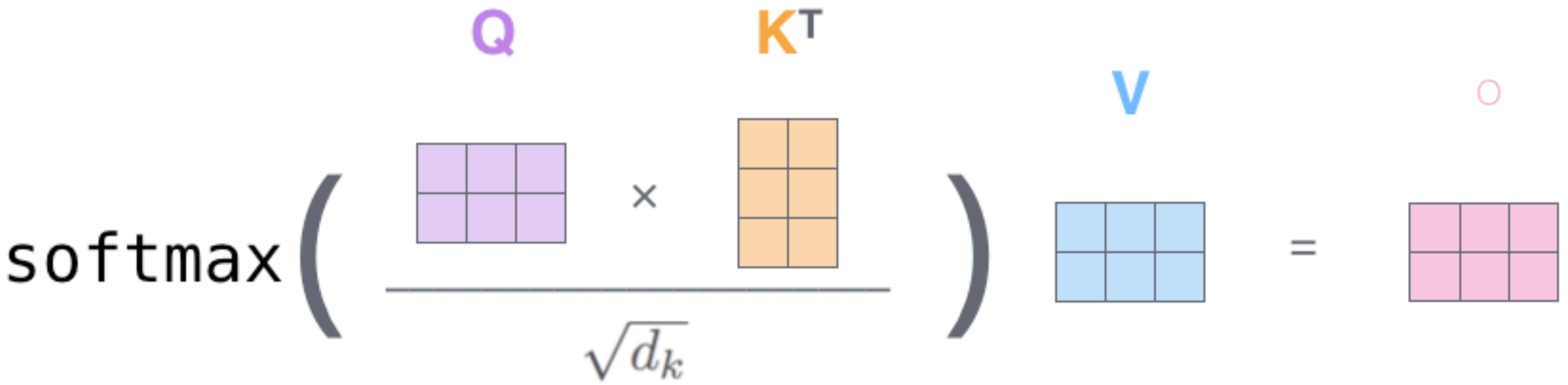


Attention-Based Conditioning

Gradient-Based Meta-Learning

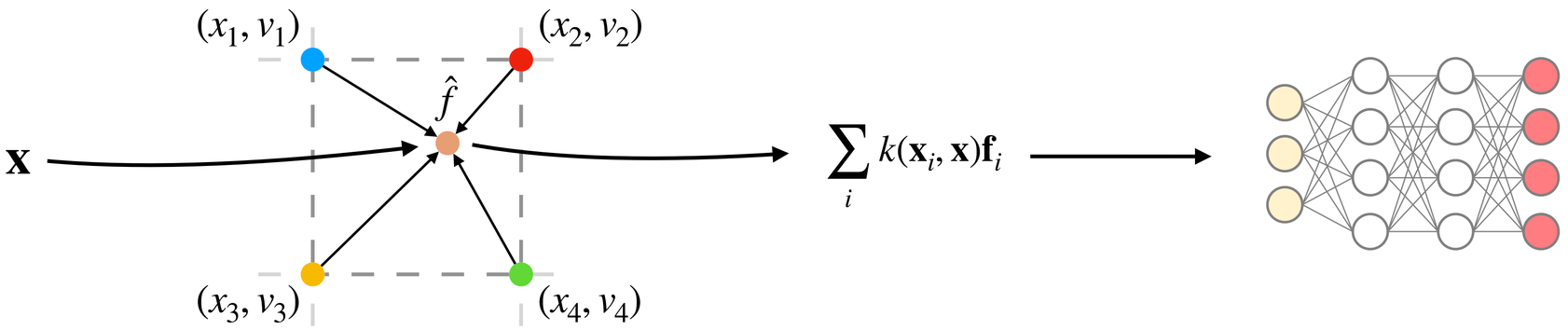
Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Recap: Scaled Dot-Product Attention



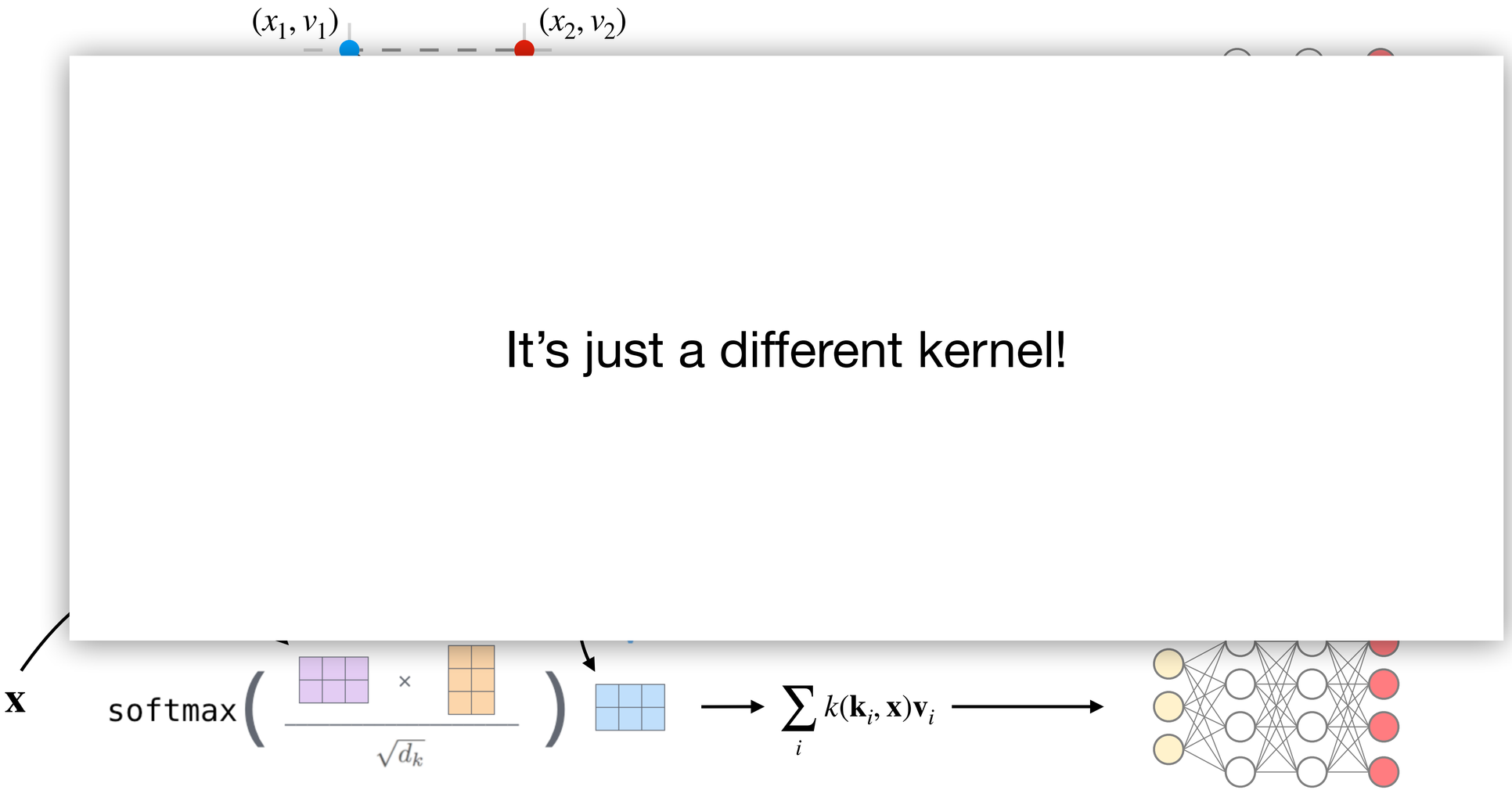
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Attention as a form of local conditioning



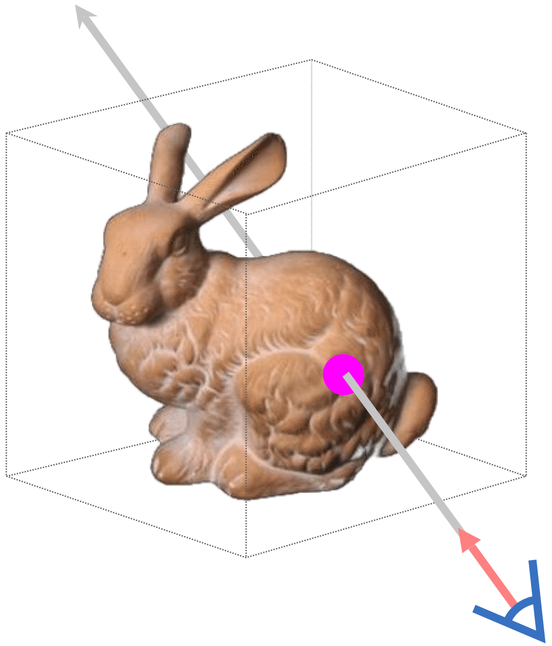
Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

Attention as a form of local conditioning

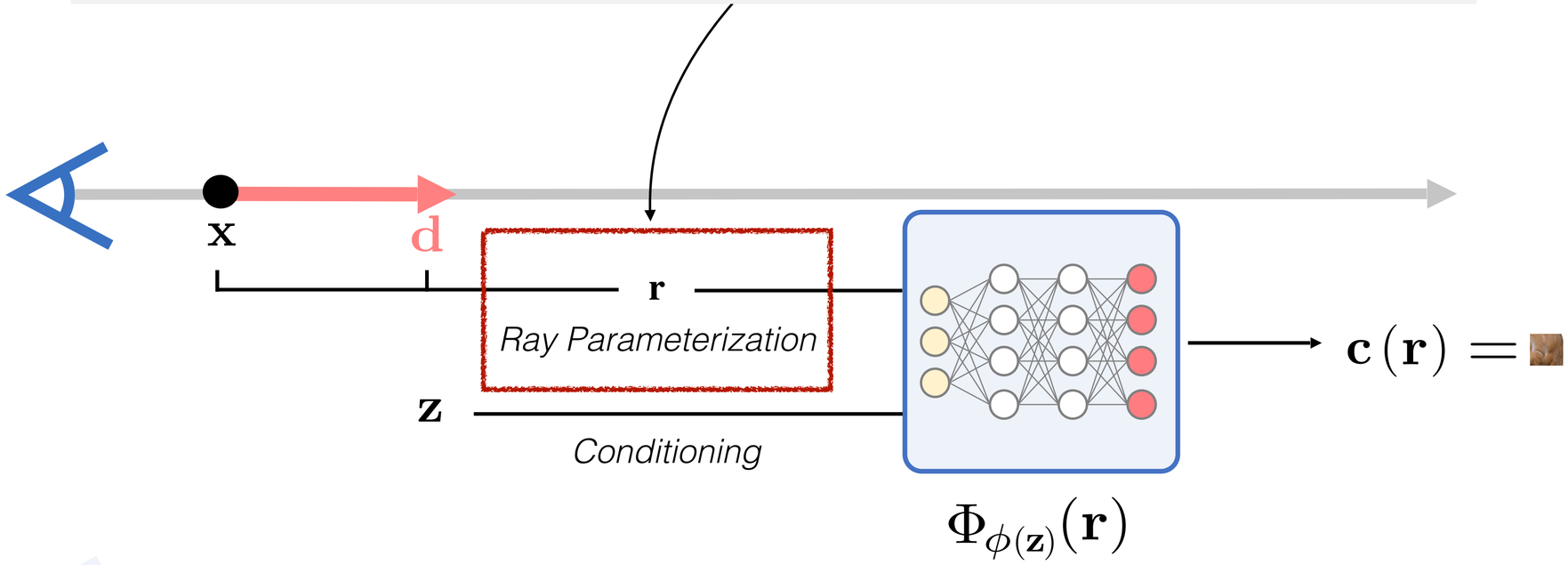


Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from https://jalammar.github.io/illustrated-transformer/

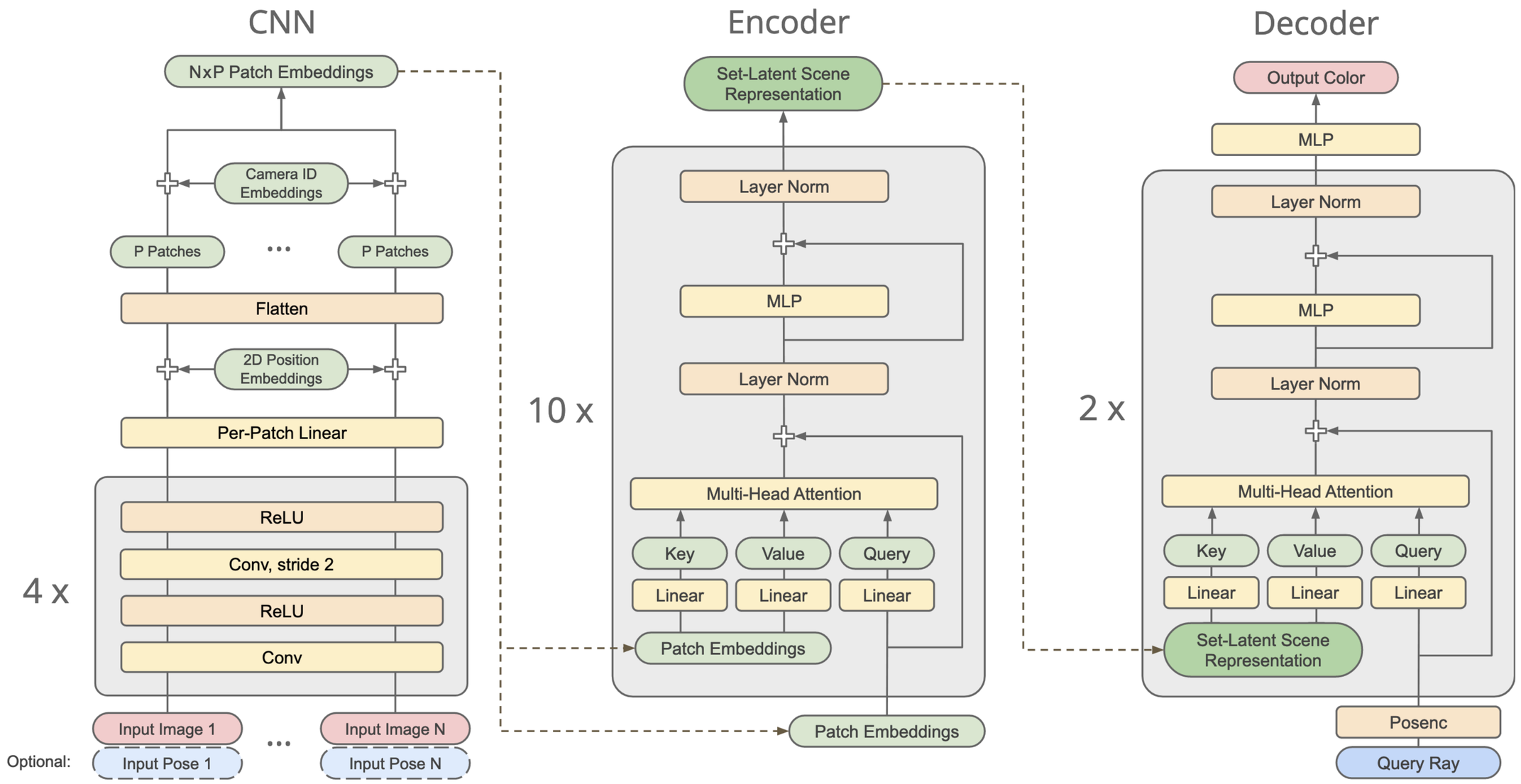
Now: *Learn Prior over Light Fields!*



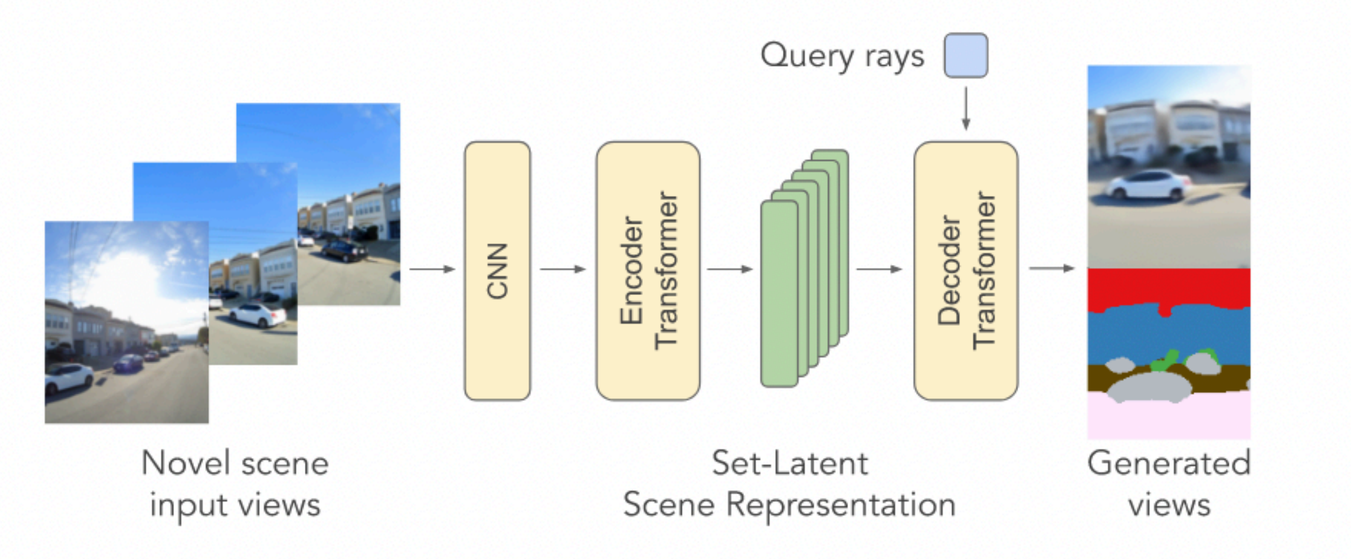
Five-dimensional: Hybrid Discrete-Continuous intractable



Transformers for Light Fields: Scene Representation Transformer



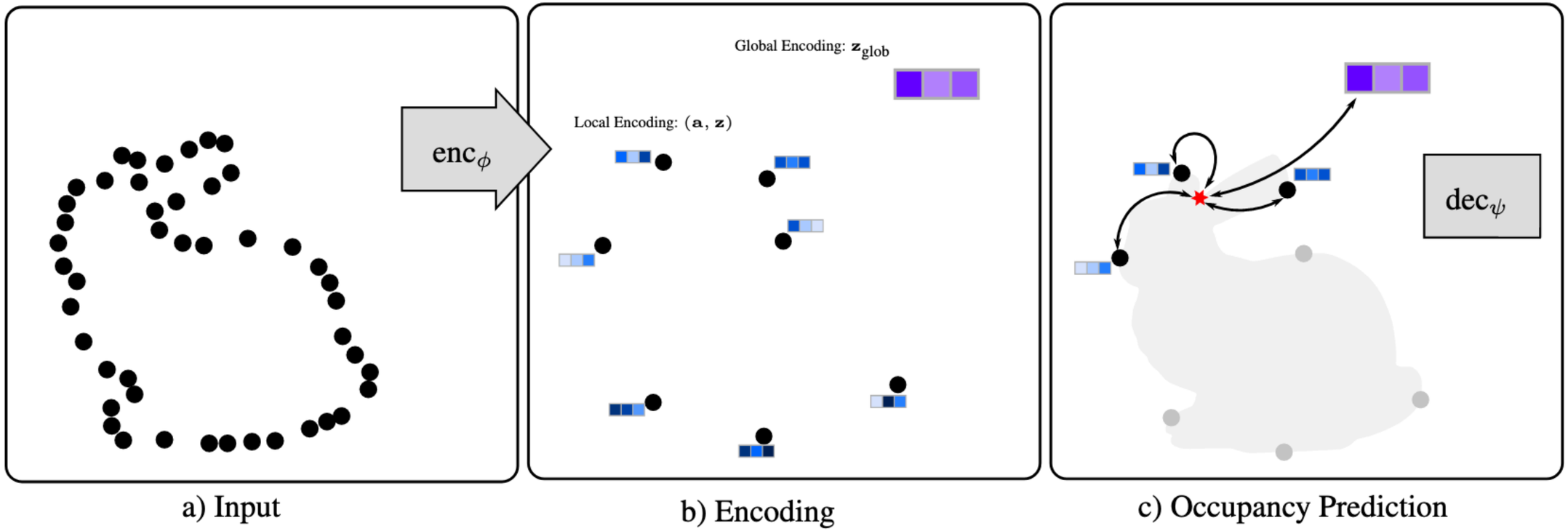
Attention-conditioned Light Fields



Scene Representation Transformer:
Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations

Mehdi S. M. Sajjadi Henning Meyer Etienne Pot Urs Bergmann Klaus Greff
Noha Radwan Suhani Vora Mario Lučić Daniel Duckworth Alexey Dosovitskiy*
Jakob Uszkoreit* Thomas Funkhouser Andrea Tagliasacchi‡*

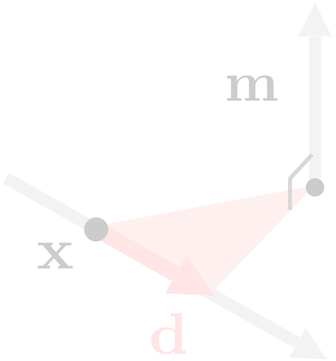
Transformer Conditioning for Point Clouds: AIR-Nets, Giebenhain et al. 2022

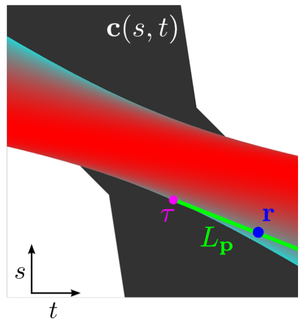


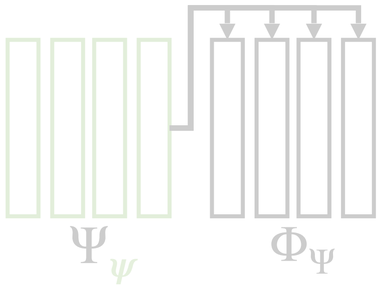
AIR-Nets, Giebenhain et al. 2022

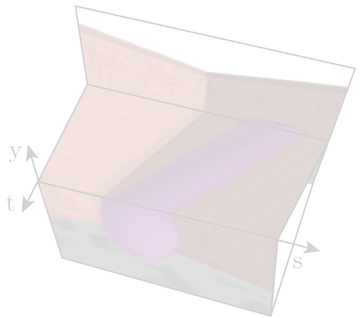
Where's the geometry in a Light Field?

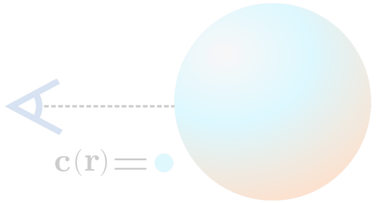
Parameterization | LFN Geometry | Meta-Learning | Results | Limitations



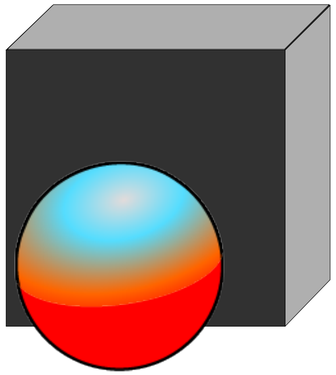




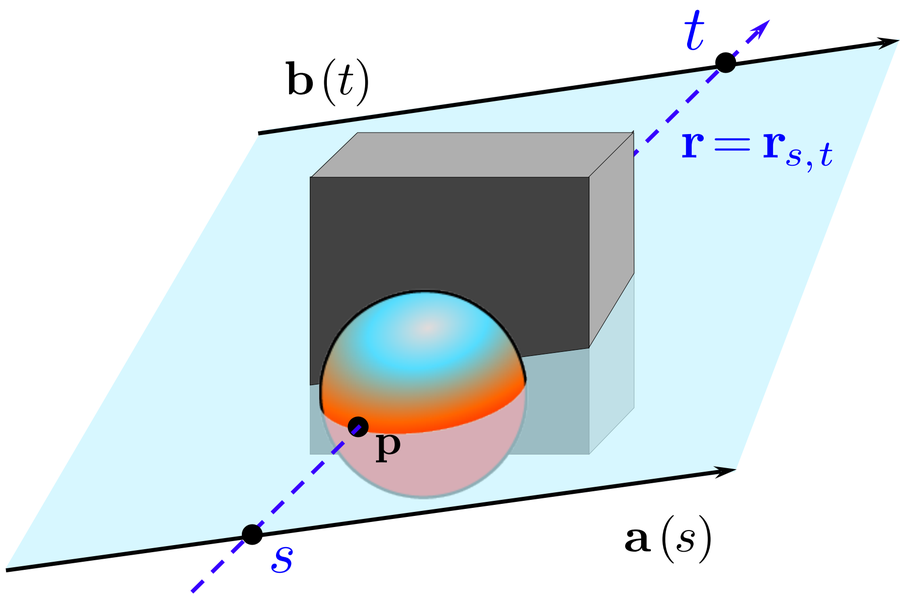




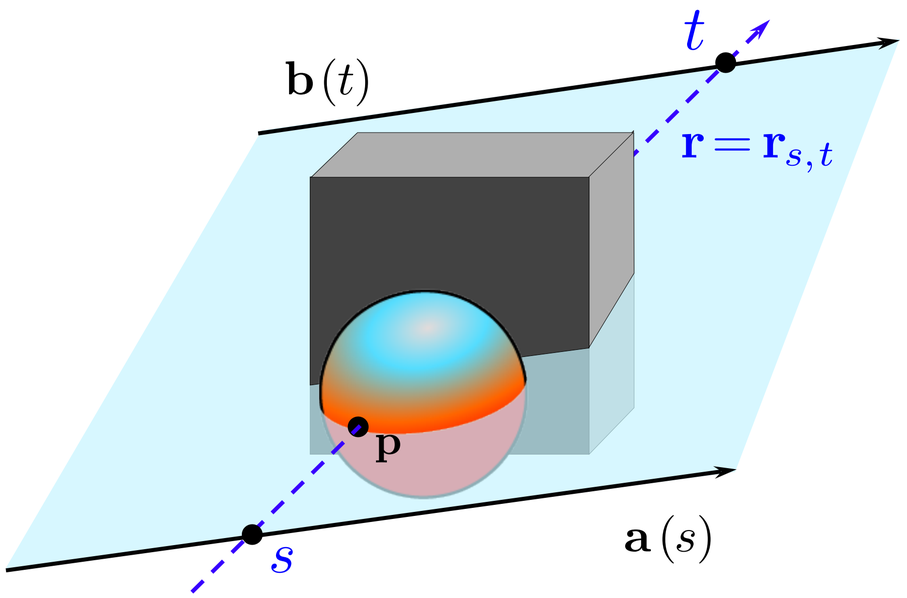
The geometry of LFNs

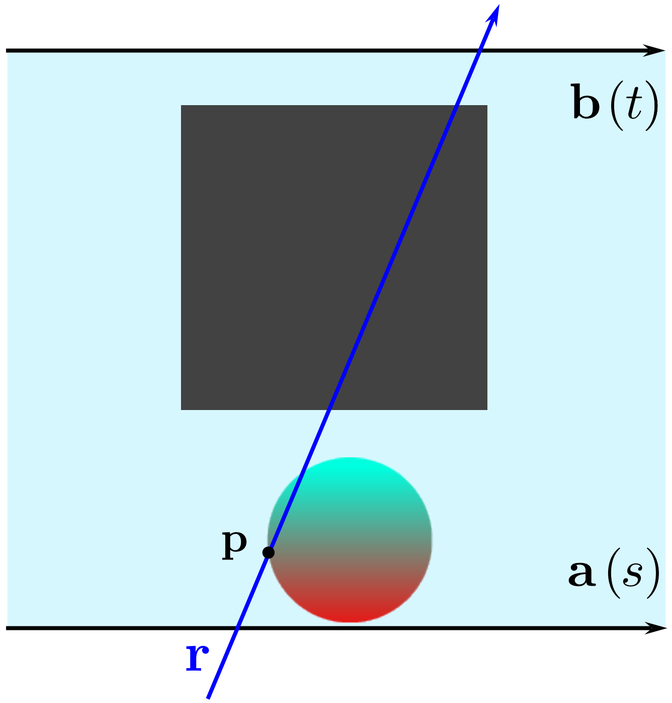


The geometry of LFNs

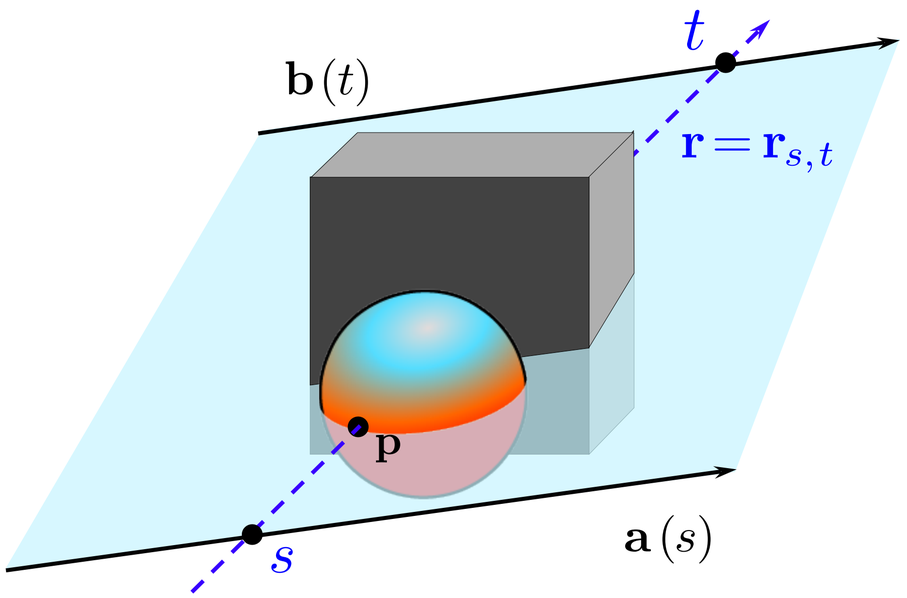


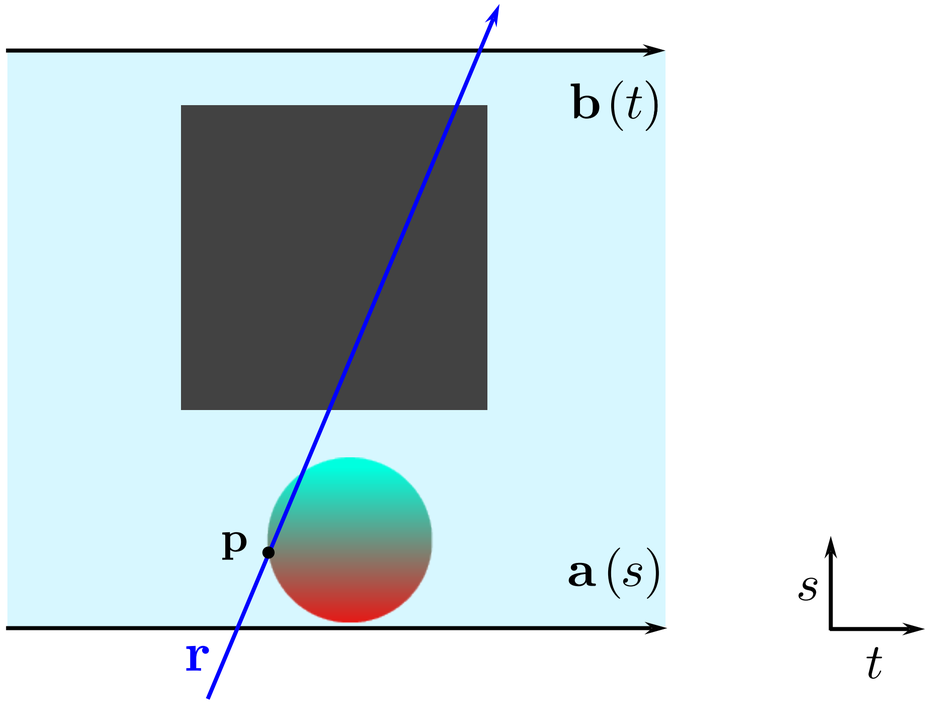
The geometry of LFNs



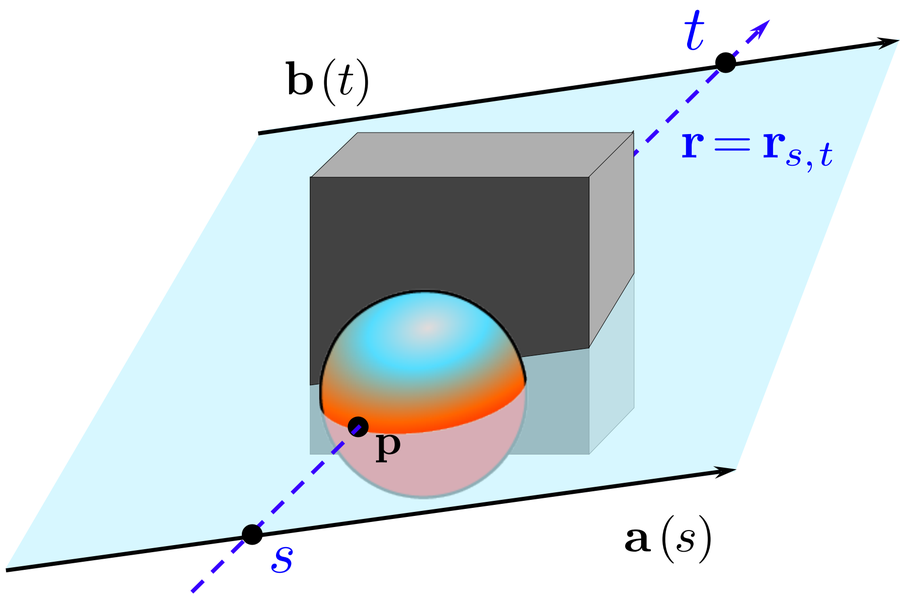


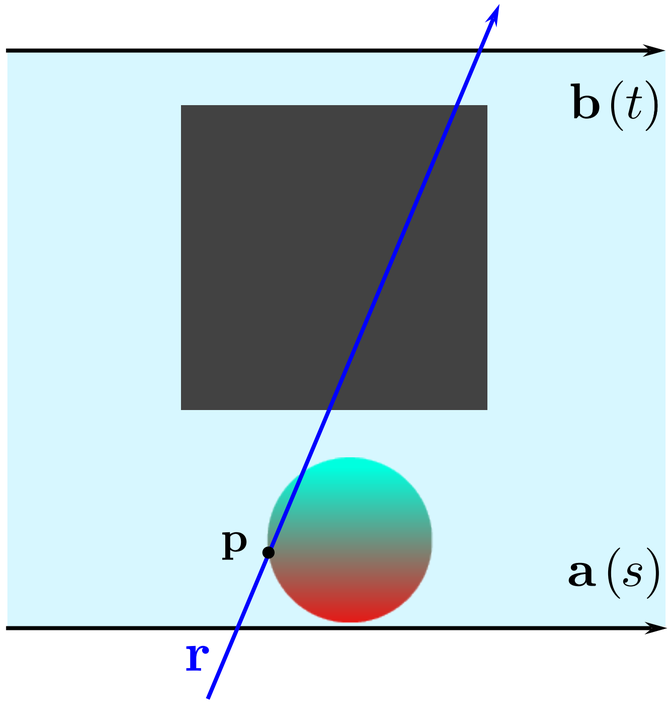
The geometry of LFNs

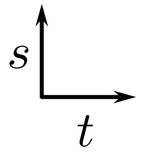




The geometry of LFNs

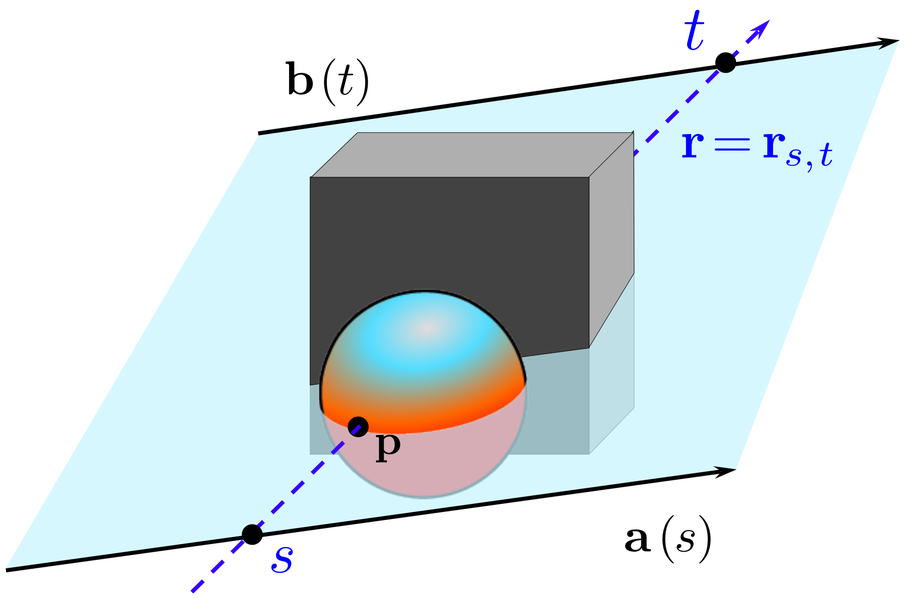


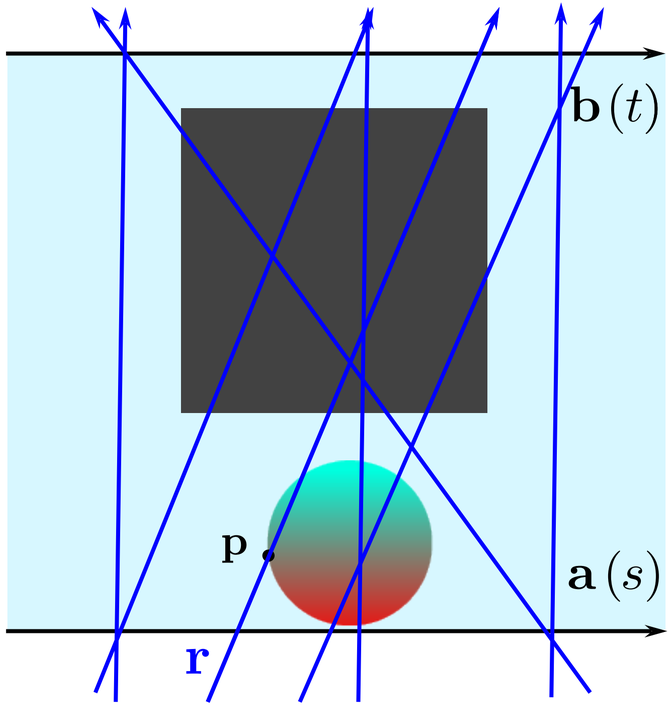


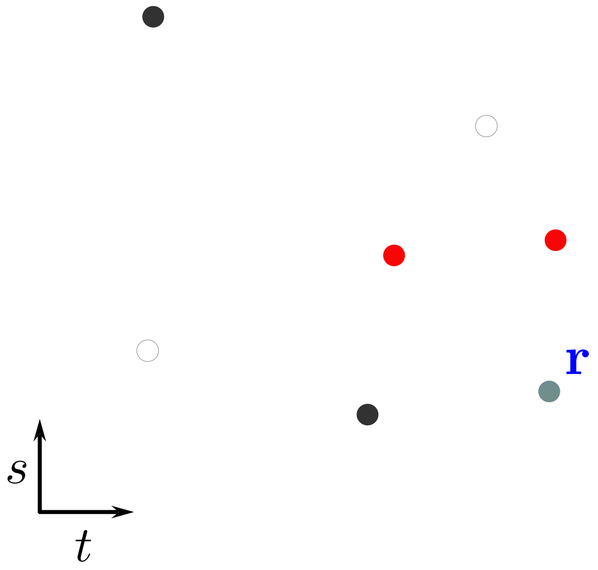


$\mathbf{r}$

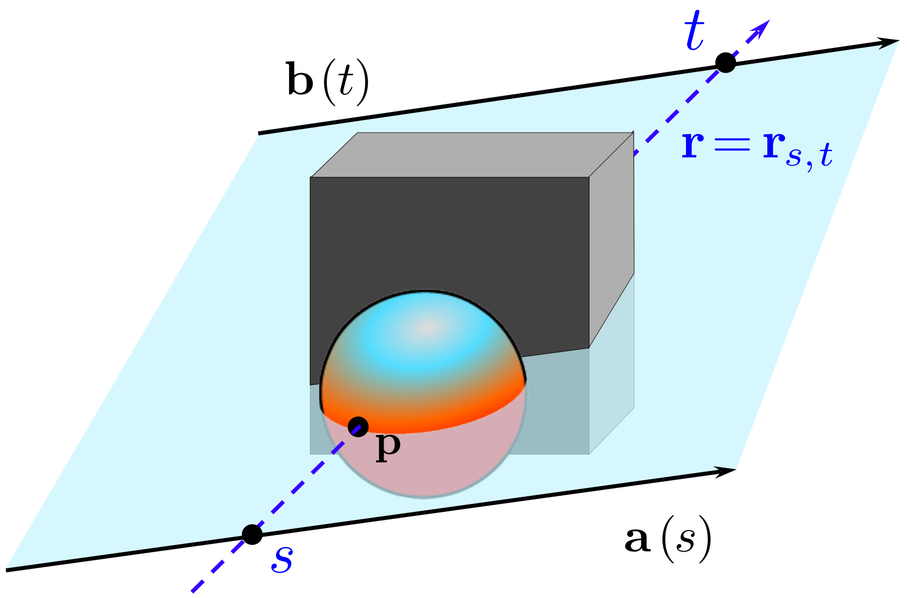
The geometry of LFNs

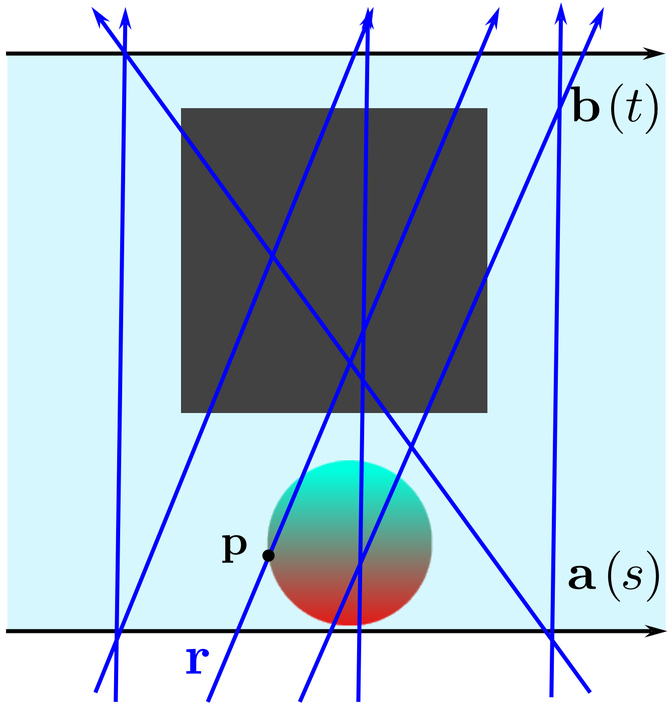




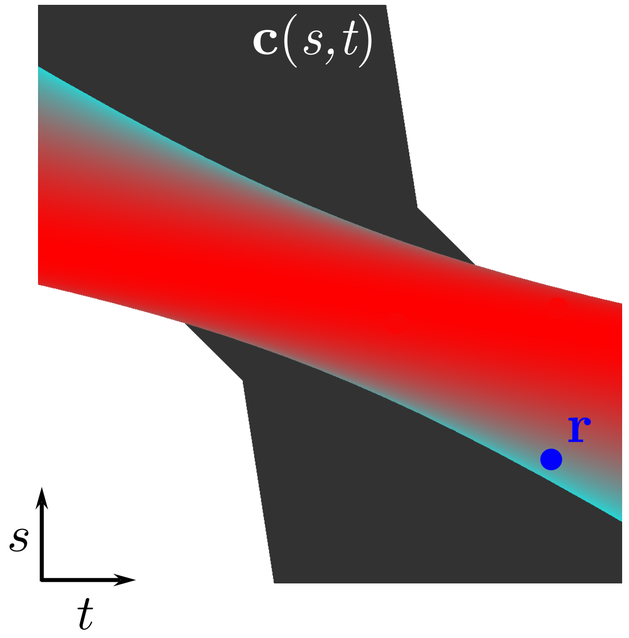


The geometry of LFNs

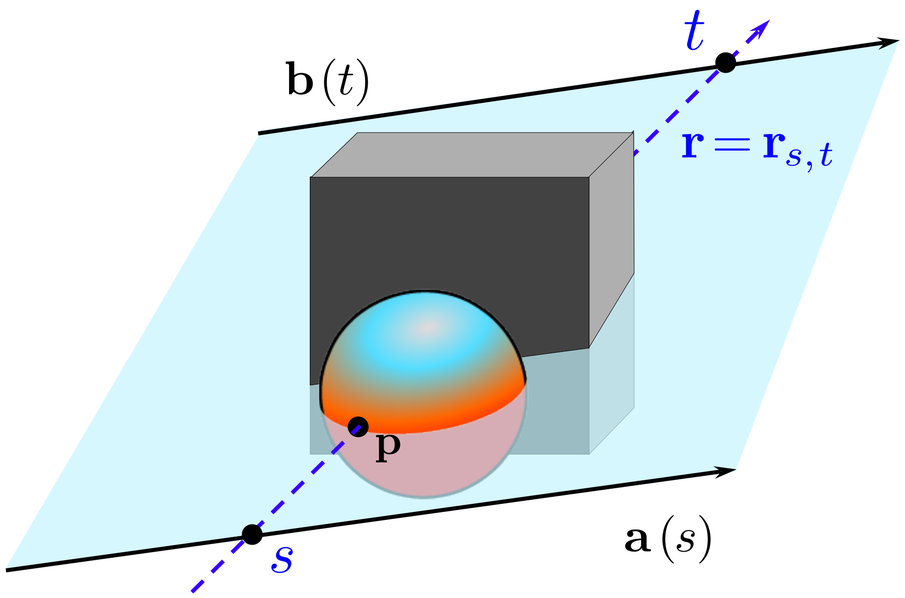


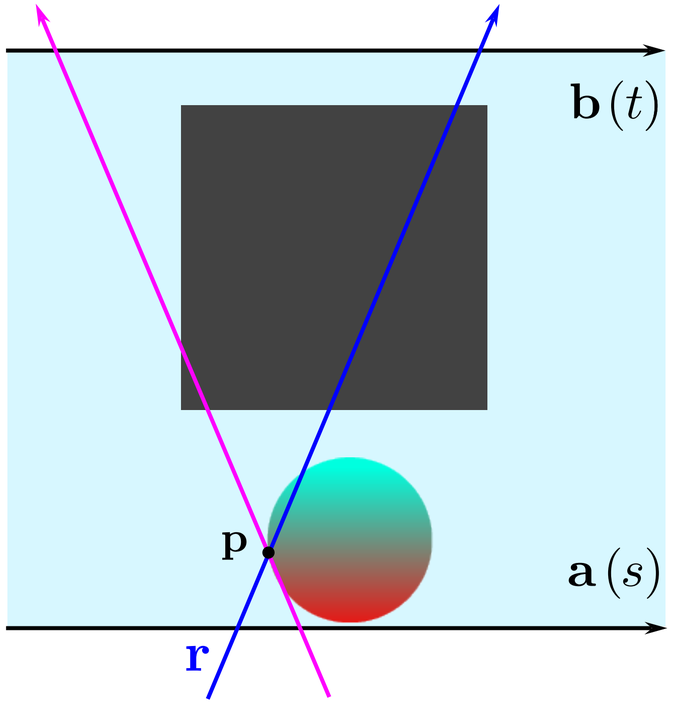


Epipolar Plane Image

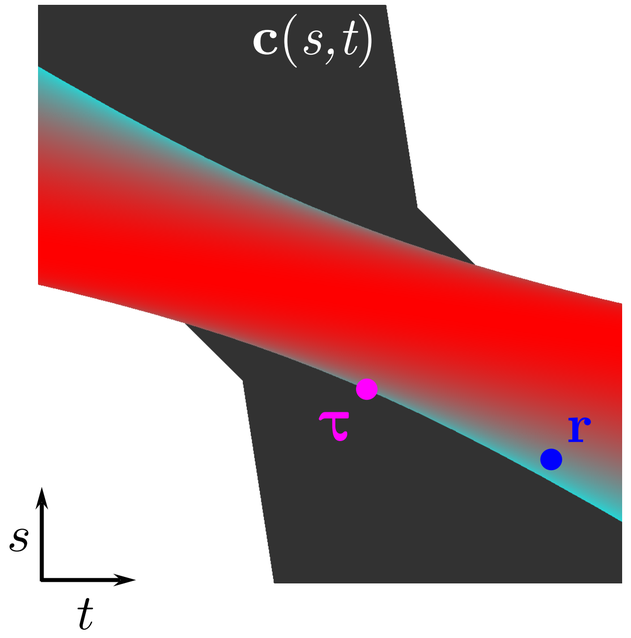


The geometry of LFNs

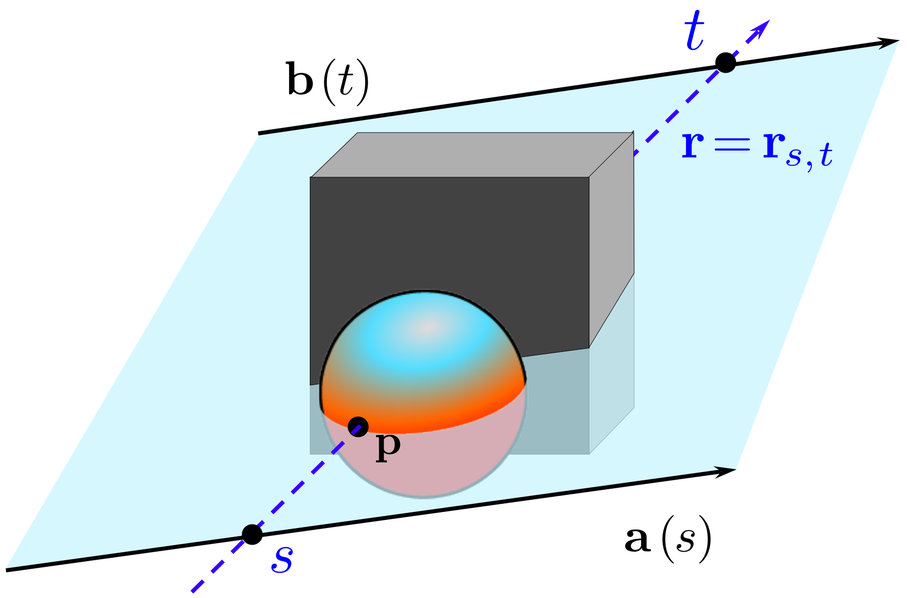


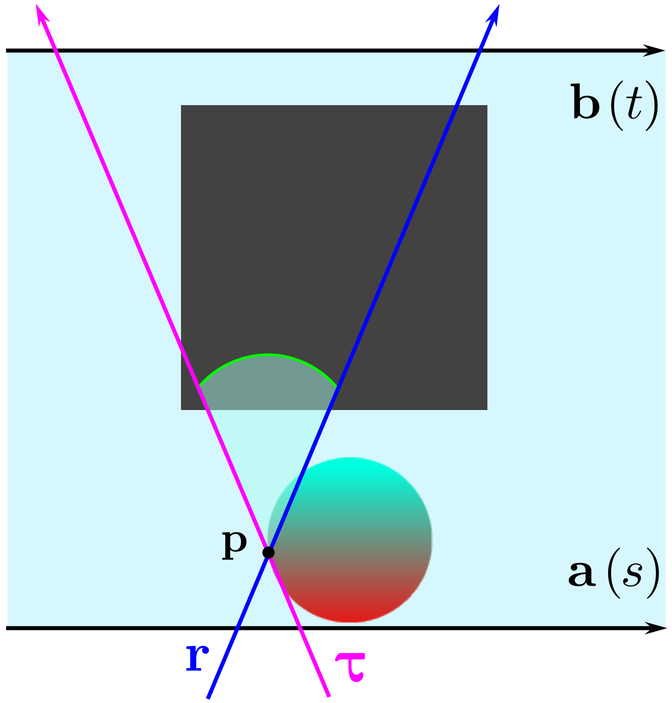


Epipolar Plane Image

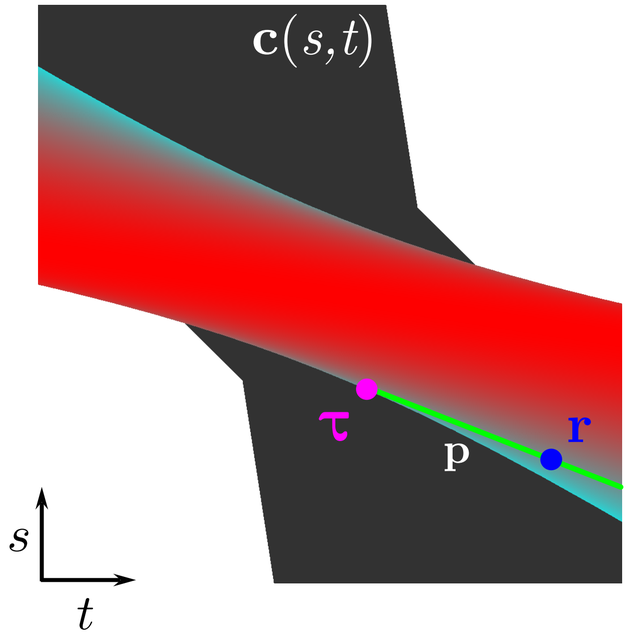


The geometry of LFNs

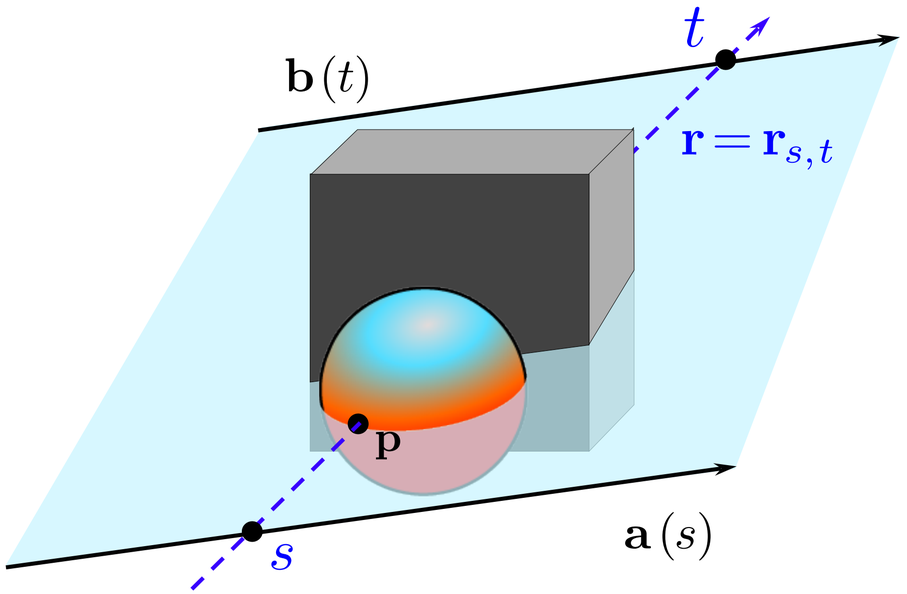


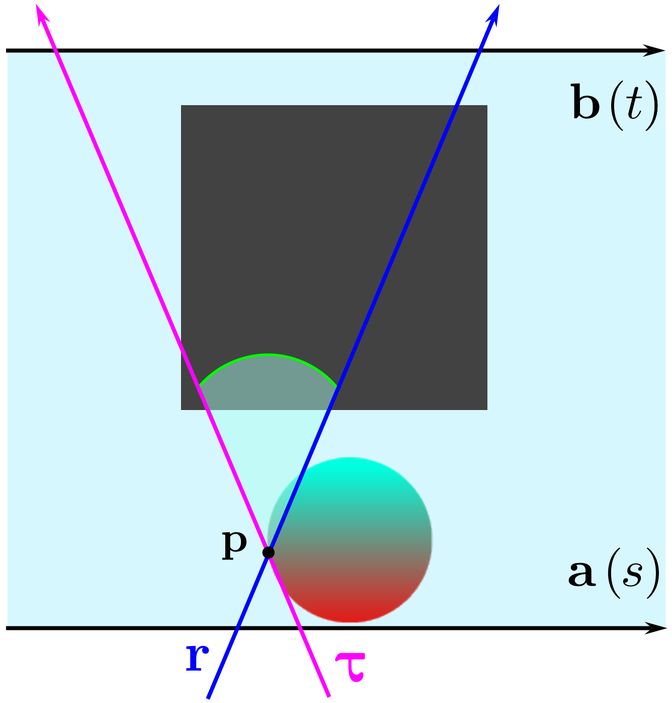


Epipolar Plane Image

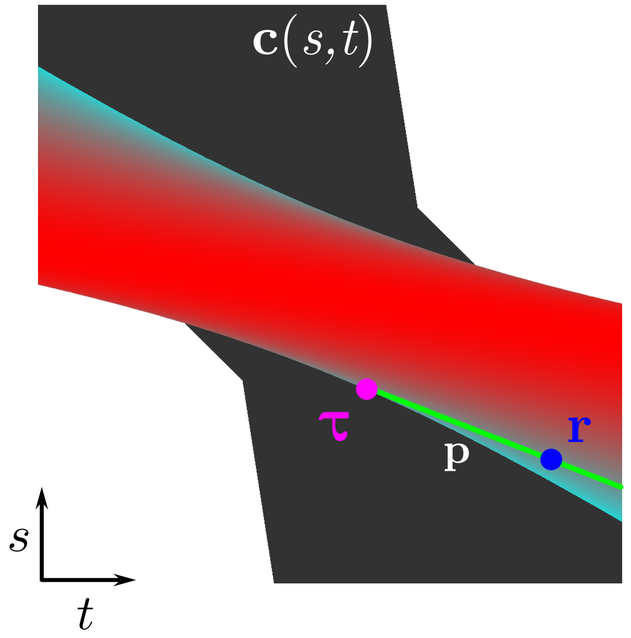


The geometry of LFNs



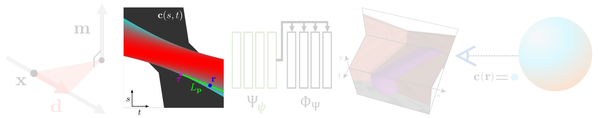


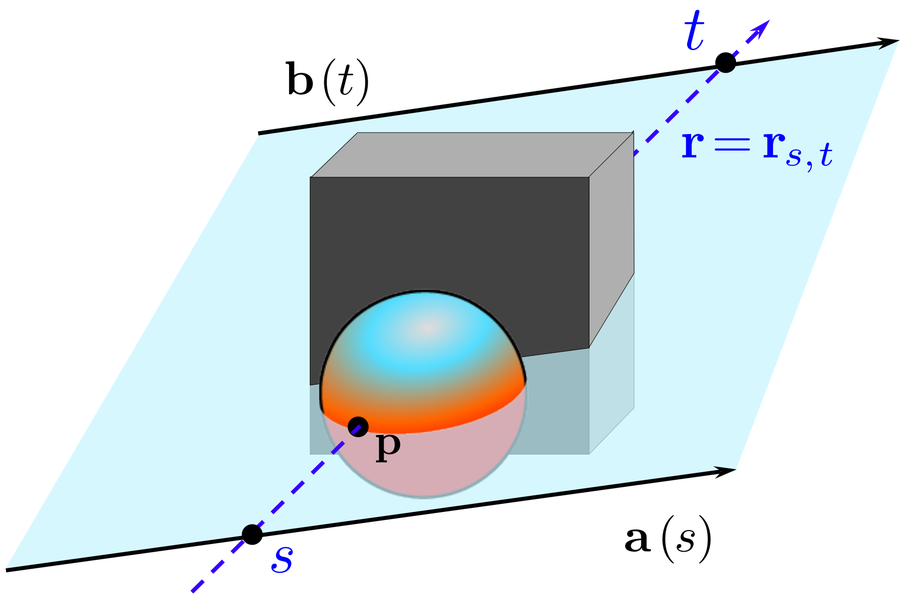
Epipolar Plane Image

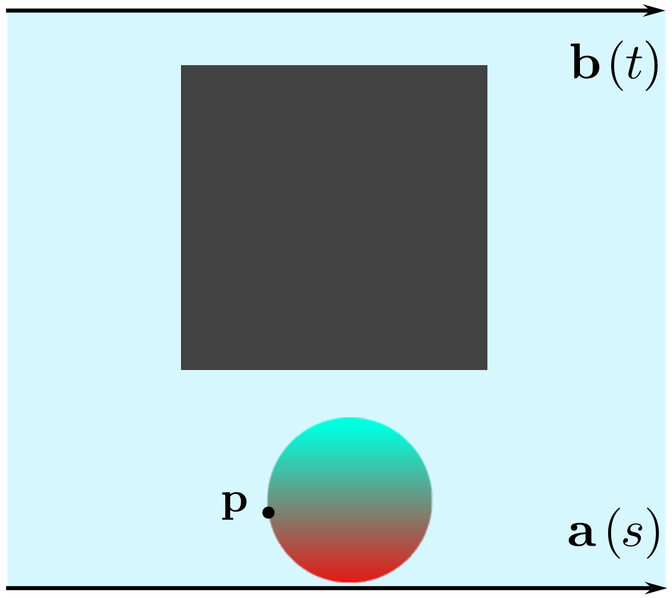


Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

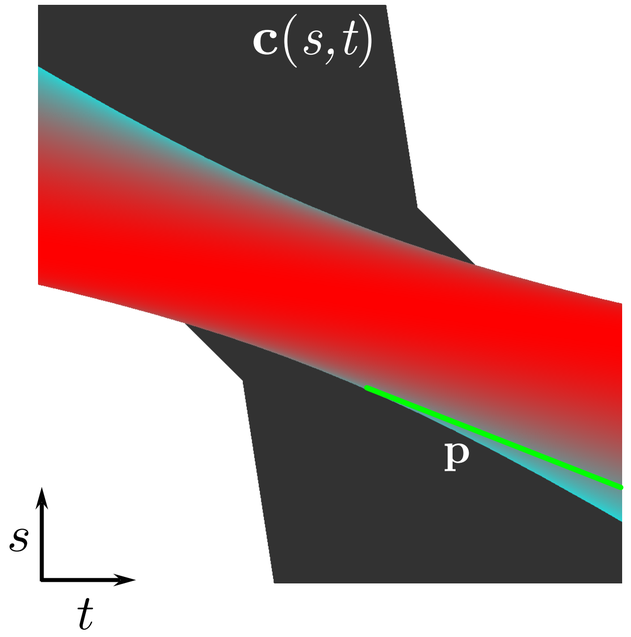
The geometry of LFNs







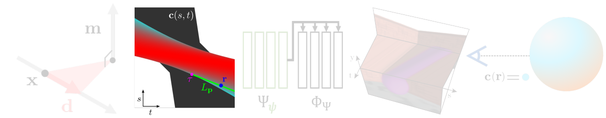
Epipolar Plane Image

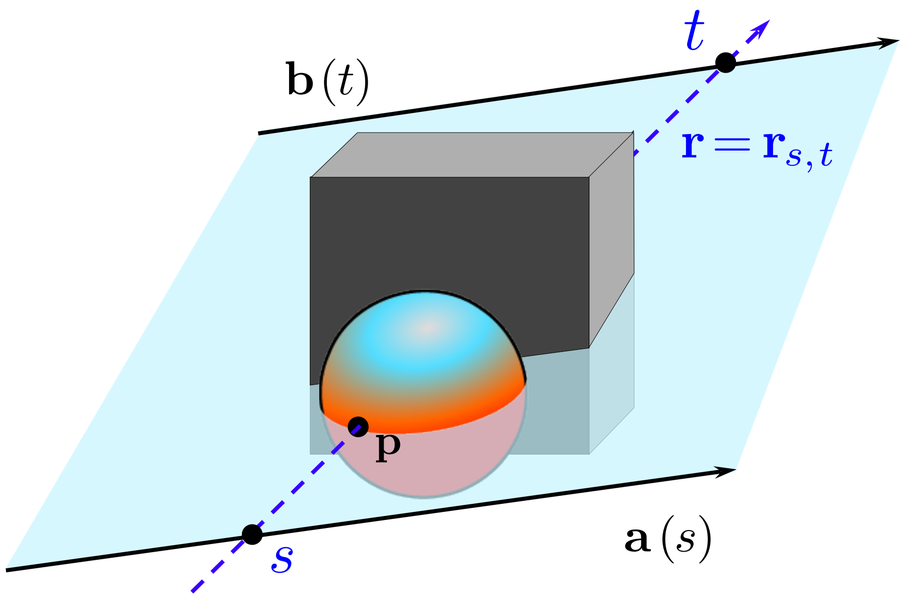


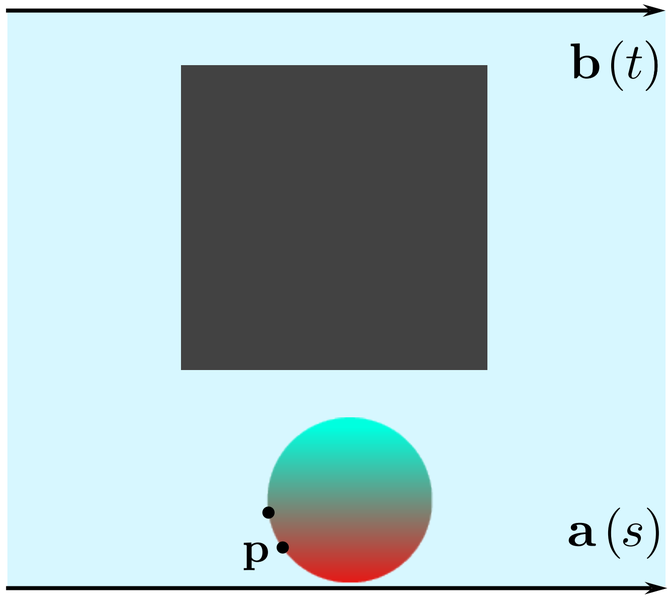
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

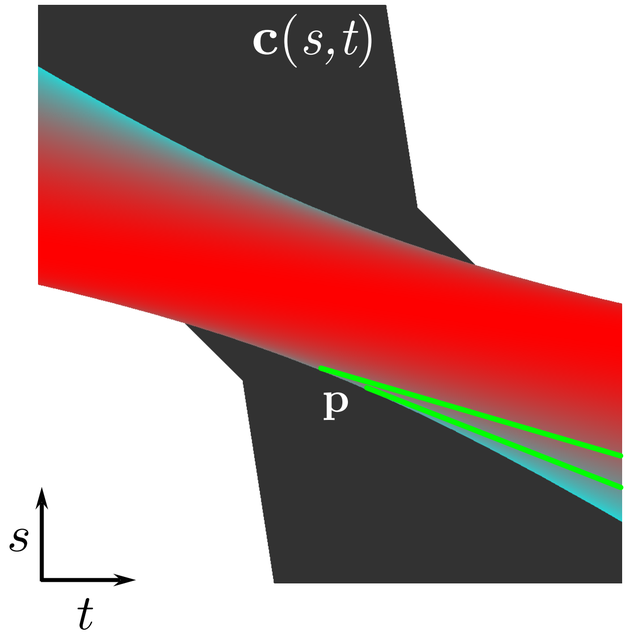
The geometry of LFNs







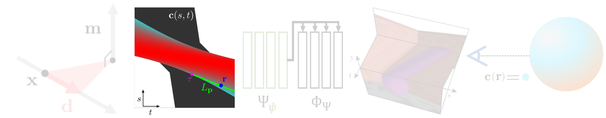
Epipolar Plane Image

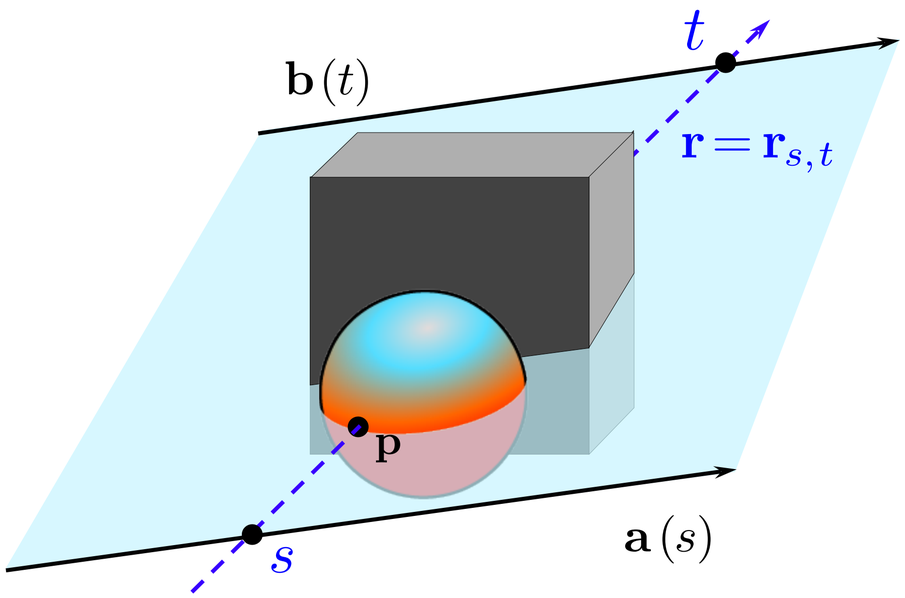


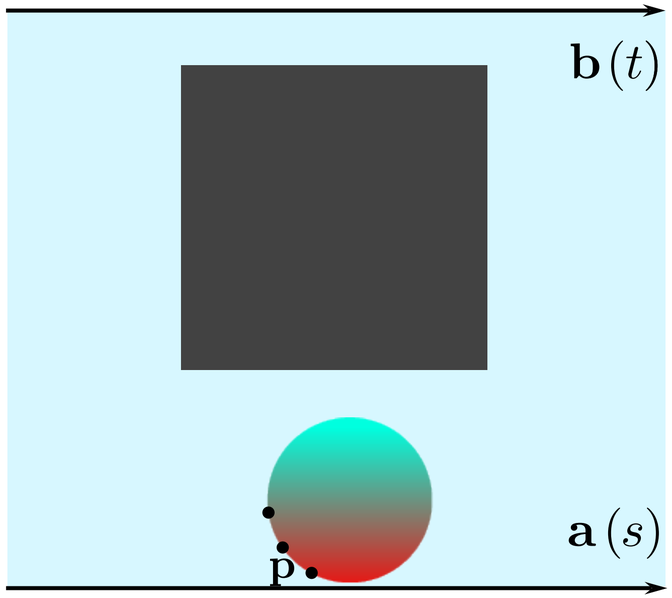
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

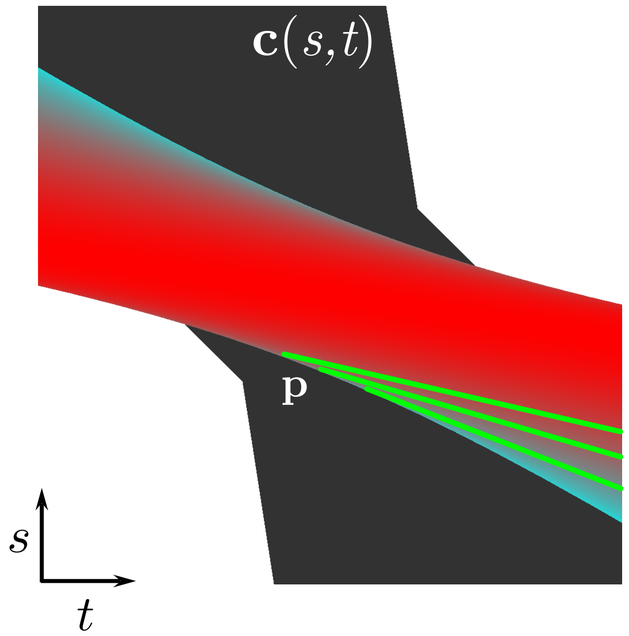
The geometry of LFNs







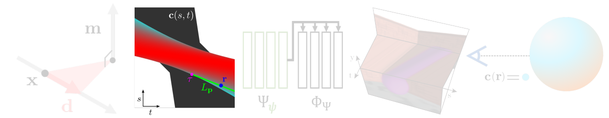
Epipolar Plane Image

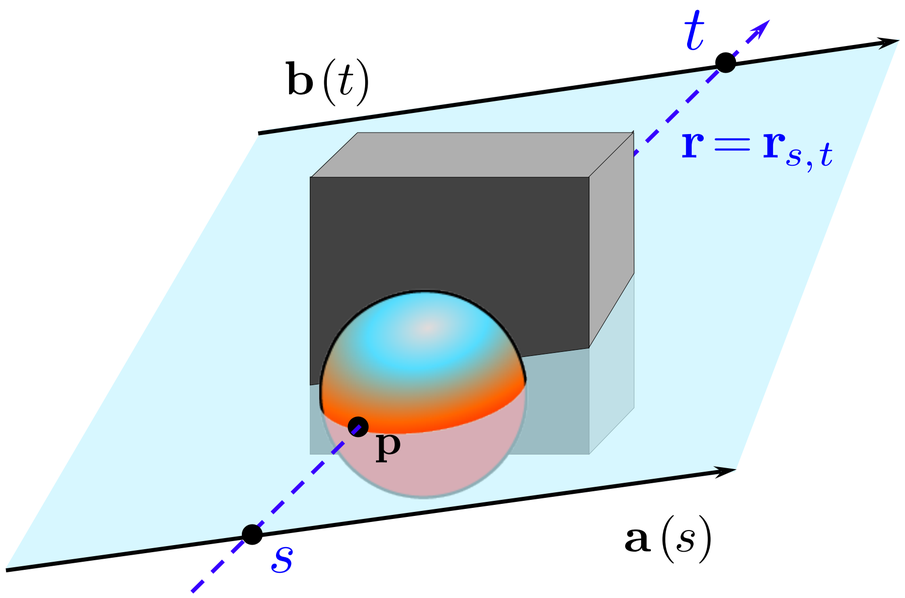


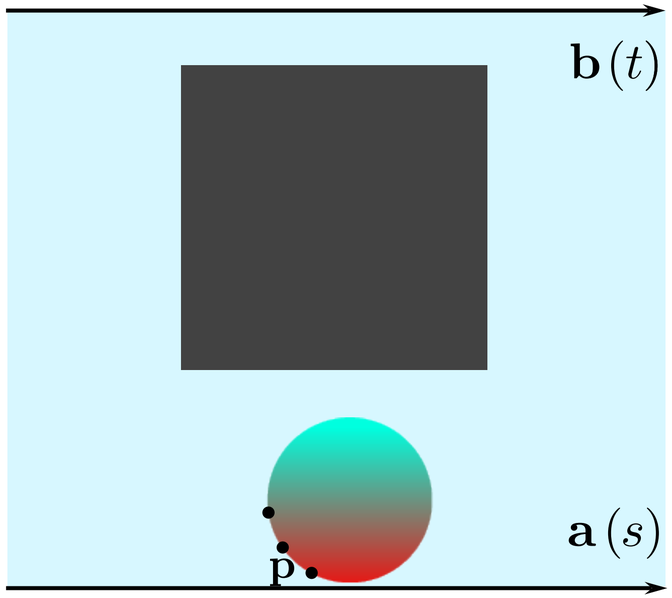
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

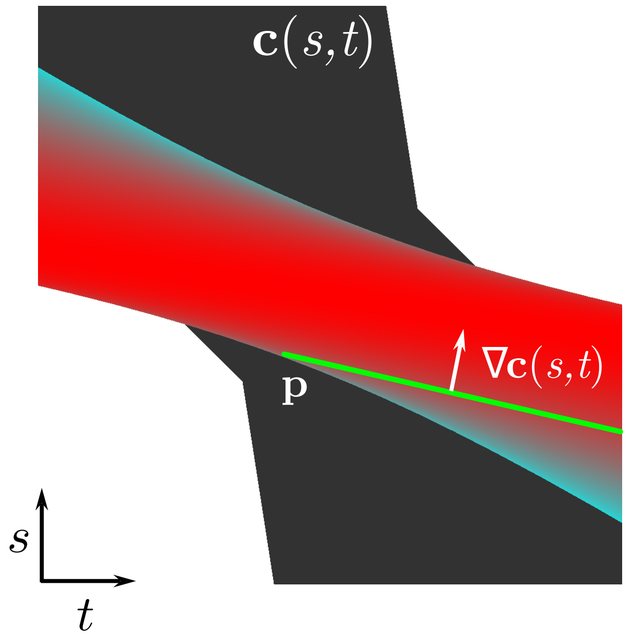
The geometry of LFNs







Epipolar Plane Image

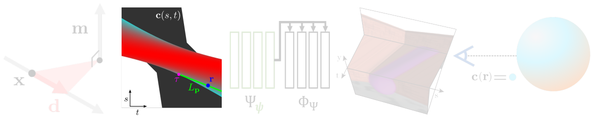


Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

Slope of line decreases as point moves closer.

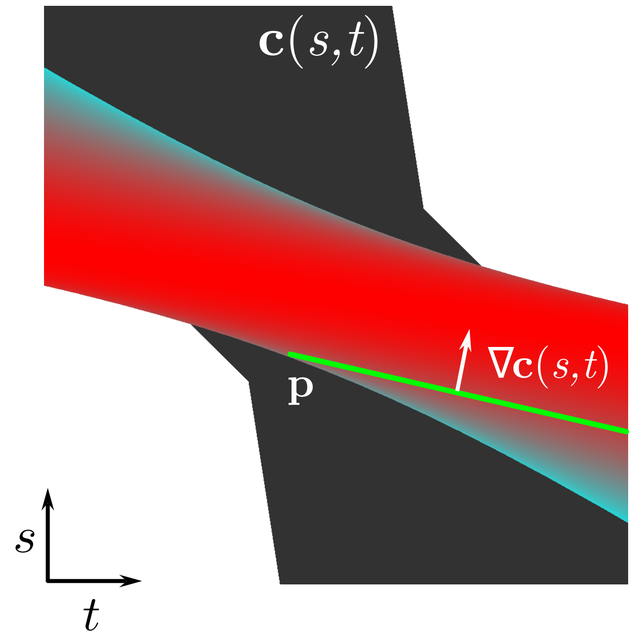
Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset -

The geometry of LFNs



$$d(s,t) = D\frac{\partial_t \mathbf{c}(s,t)}{\partial_s \mathbf{c}(s,t) + \partial_t \mathbf{c}(s,t)}$$

Epipolar Plane Image

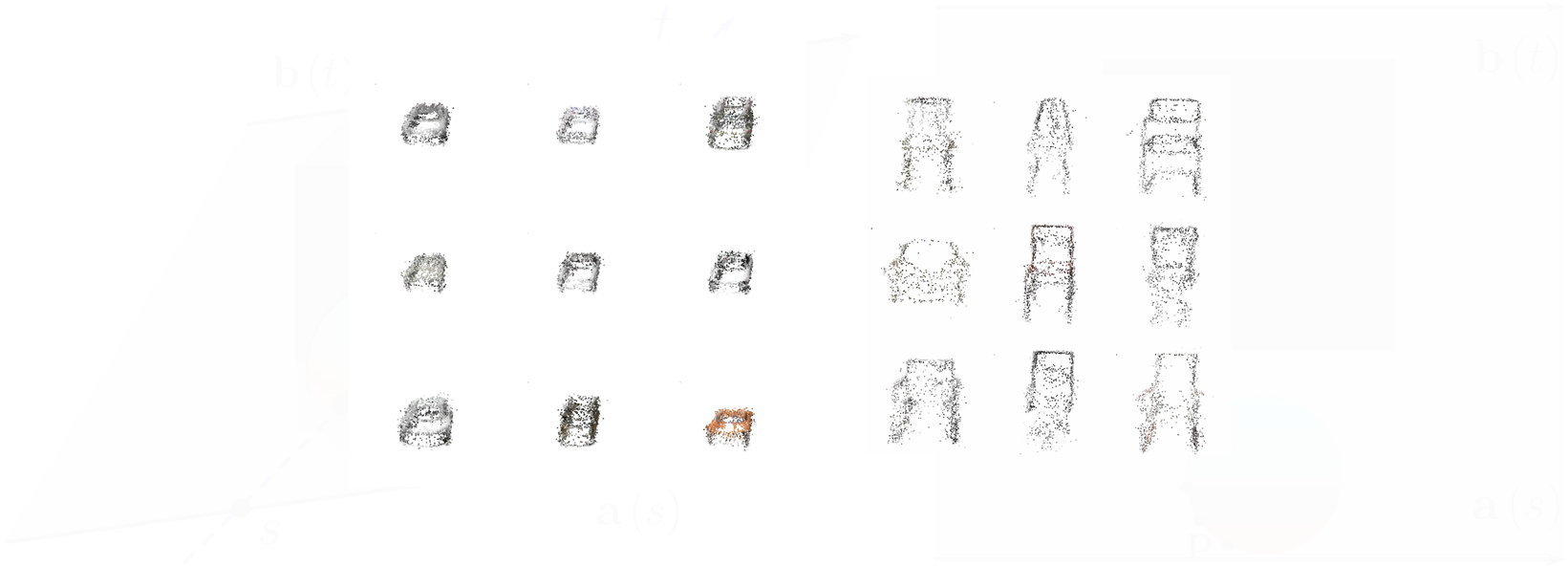


Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

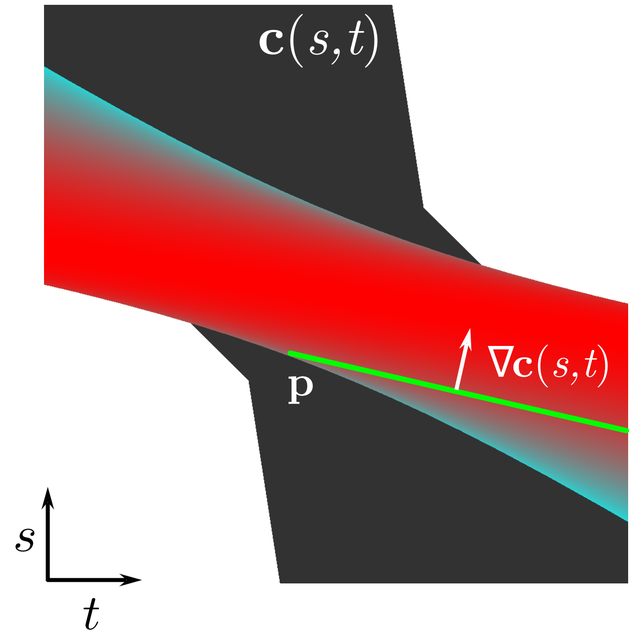
Slope of line decreases as point moves closer.

Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset - can extract depth from gradients of light field.

The geometry of LFNs



Epipolar Plane Image



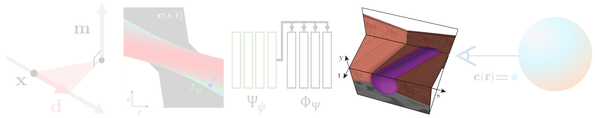
Points give lines of constant color in EPI $\mathbf{c}(s,t)$ – line is a levelset of the EPI.

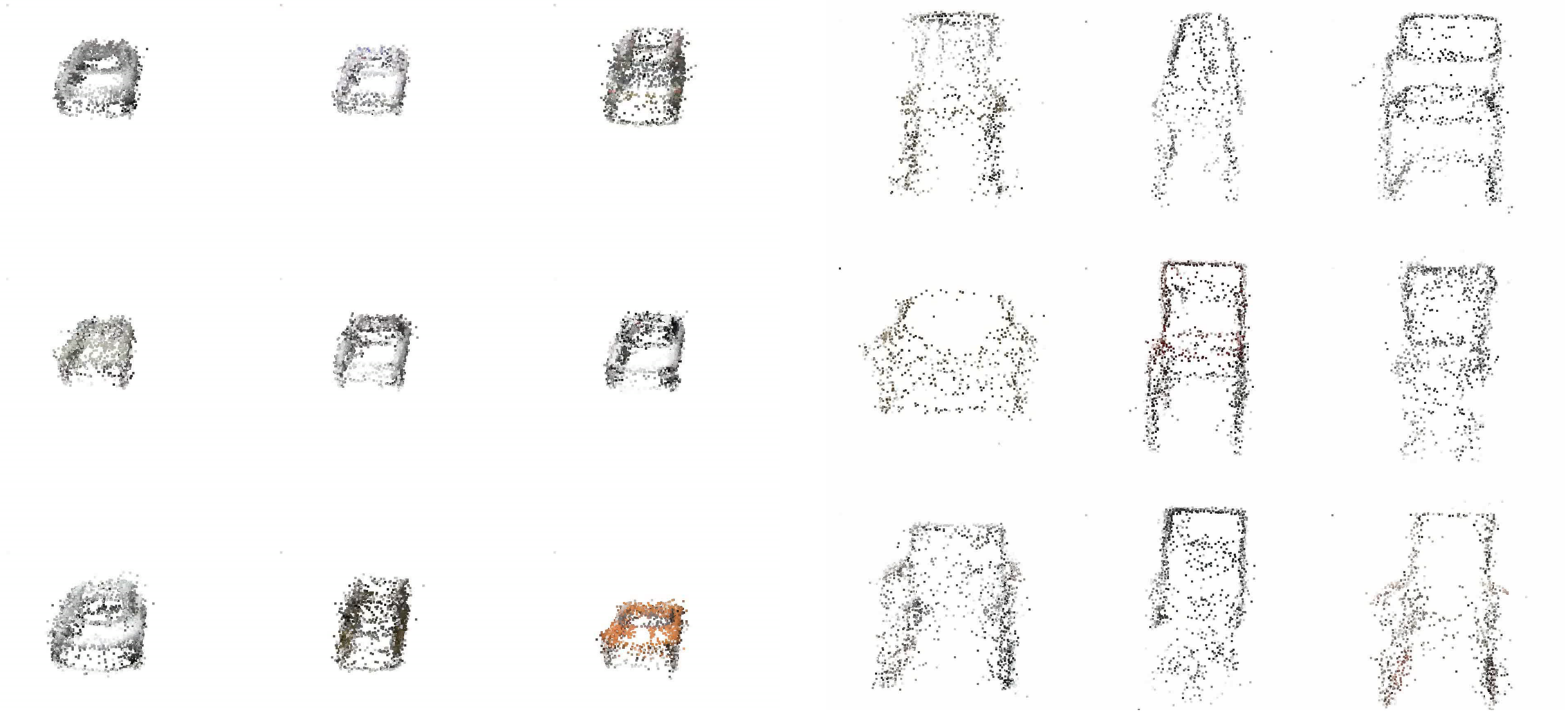
Slope of line decreases as point moves closer.

Gradient of $\mathbf{c}(s,t)$ is orthogonal to levelset - can extract depth from gradients of light field.

Pointclouds extracted from 4 views of cars & chairs

Single evaluation of network & its gradient per ray, constant complexity



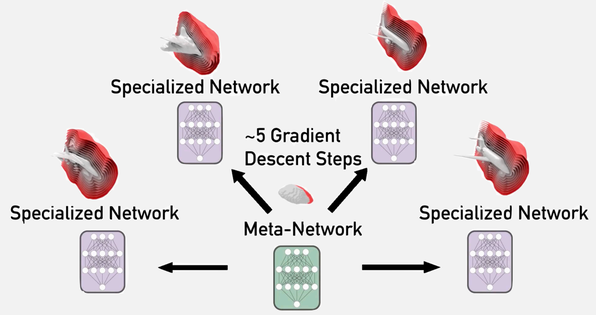


Other types of conditioning…?

Global Conditioning

Local Conditioning

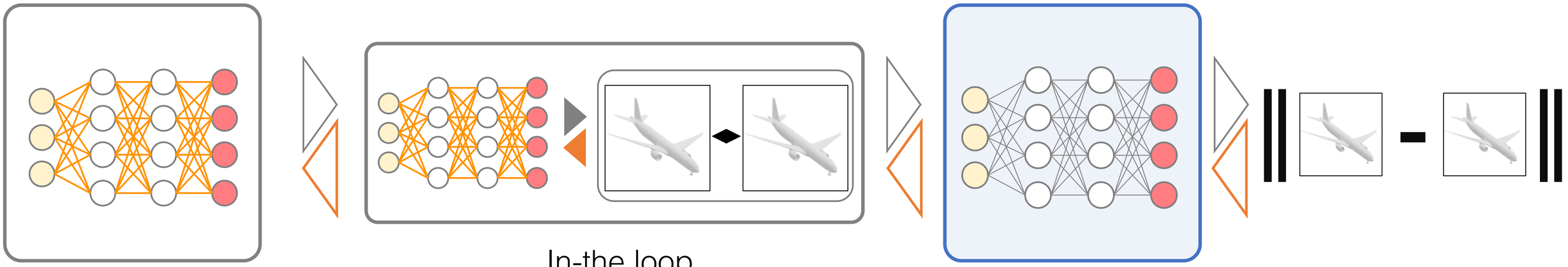
Attention-Based Conditioning



Gradient-Based Meta-Learning

Neural Fields in Visual Computing and Beyond, Xie et al., EG STAR 2022

Other forms of Generalization: Gradient-based meta-learning



Meta-Representation

In-the loop
specialization via gradient descent

Representation

Other forms of Generalization: Gradient-based meta-learning

Meta-Represe

Behaves somewhat similar to global conditioning.

gradient-
ining time.

explains
held-out observations when fit to
context observation.

MetaSDF: Meta-learning Signed Distance Functions, NeurIPS 2020

Next steps

- Oct 6th: Paper Session
- Oct 11th: Student Holiday
- Oct 18th: Paper Session
- Oct 20th: Lecture on Unconditional Generative Models of 3D Scenes!