Discontinuity-Aware 2D Neural Fields

Yash Belhe - University of California San Diego
Michael Gharbi, Matthew Fisher, Iliyan Georgiev - Adobe
Ravi Ramamoorthi, Tzu-Mao Li - University of California San Diego

Path-tracing can produce arbitrarily high resolution images

Let’s re-render a 100x close-up here

100x zoom — image has discontinuities!

Discontinuity locations are analytically known

Most image formats do not use discontinuity information

Our contribution

Hybrid neural-mesh-based representation for images

- Is **optimizable**
- Can be **rendered** at any zoom scale in real time
- Can **preserve discontinuities** that are given

Common image representations

Raster images can represent complex signals

… but details are limited by resolution

Neural fields can compactly encode giga images!

InstantNGP: Muller 22

Martel 21: ACORN: Adaptive Coordinate Networks for Neural Representation

Muller 22: Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

… but they blur discontinuities

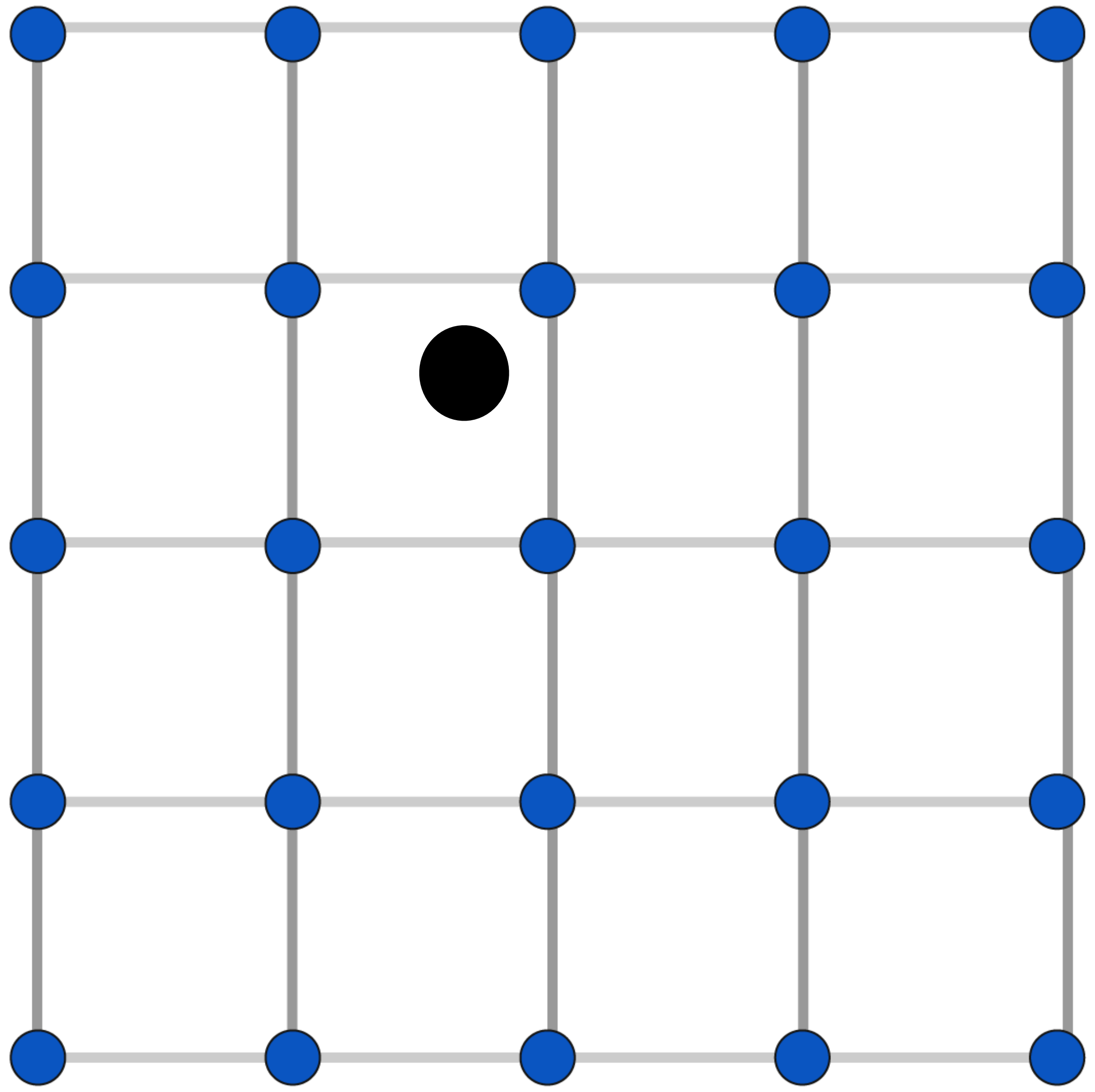
InstantNGP: Muller 22

Neural fields are continuous by construction

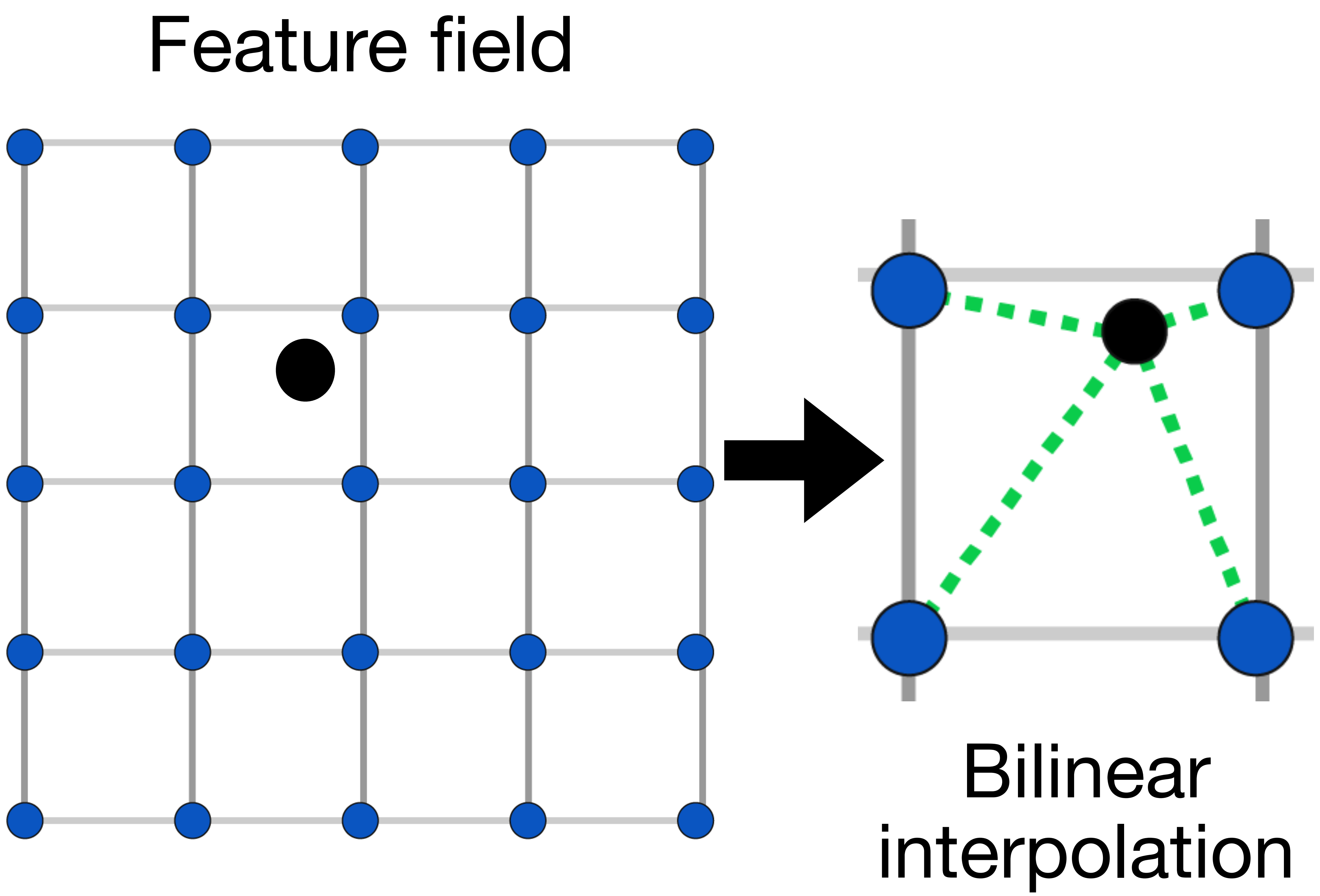
●

Neural fields are continuous by construction

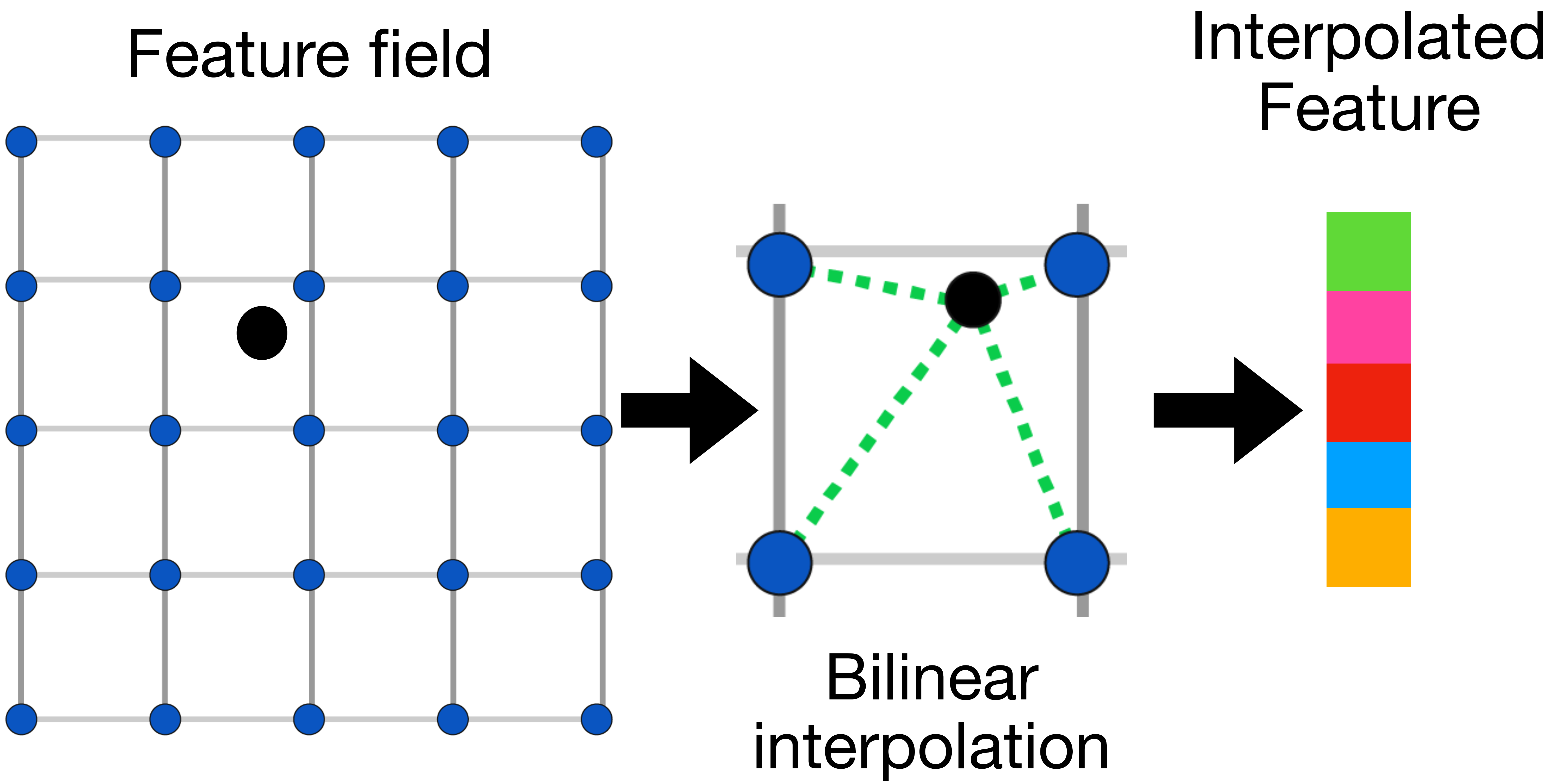
Feature field



Neural fields are continuous by construction



Neural fields are continuous by construction

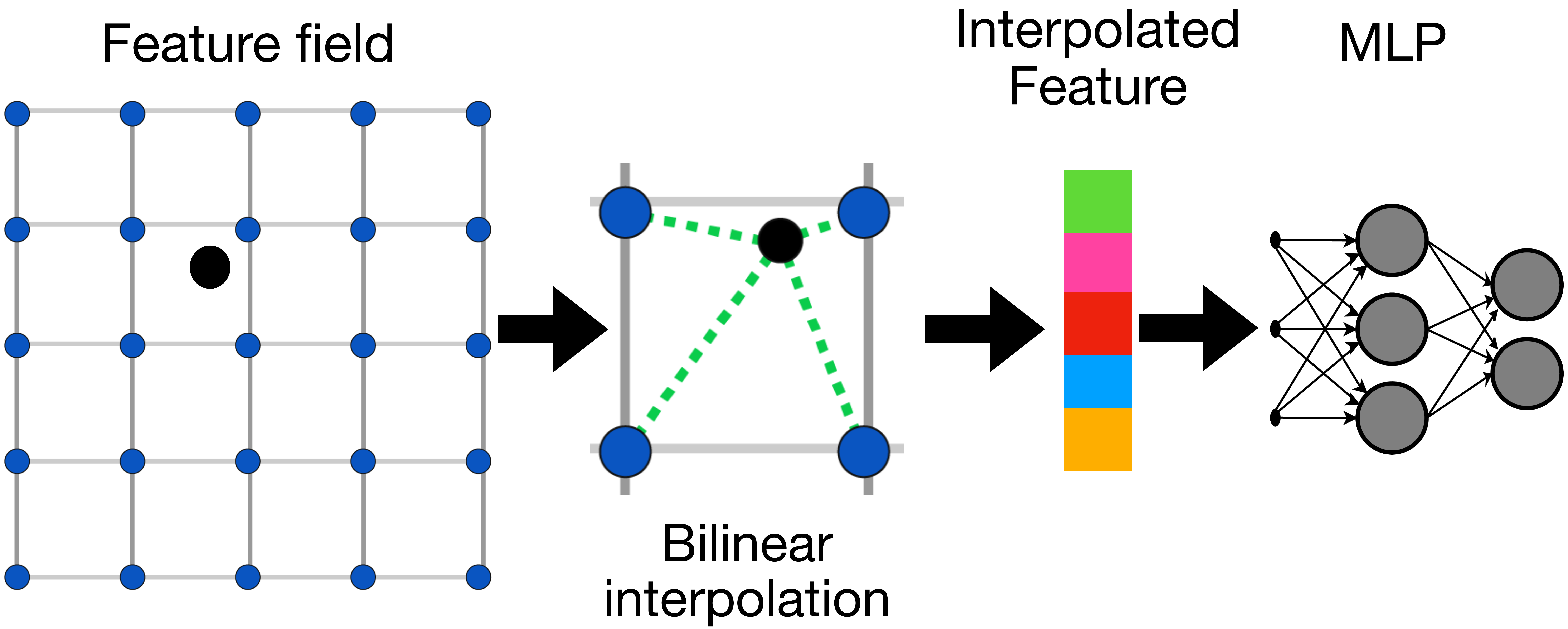


Feature field

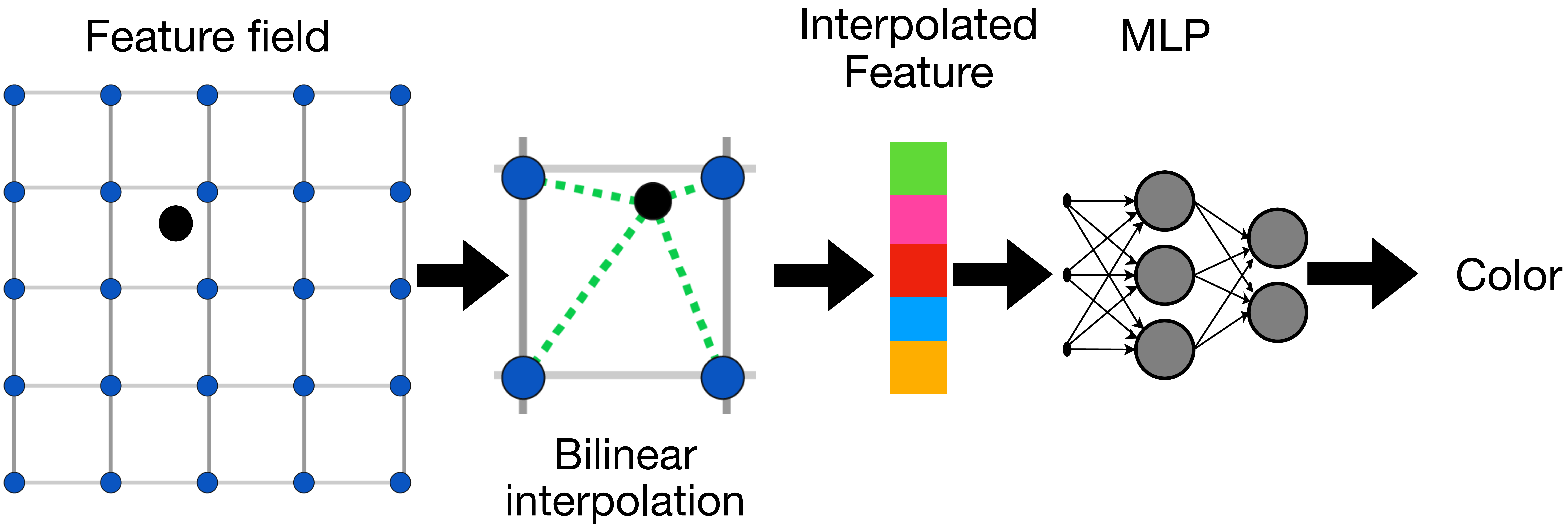
Bilinear interpolation

Interpolated Feature

Neural fields are continuous by construction



Neural fields are continuous by construction



Feature field

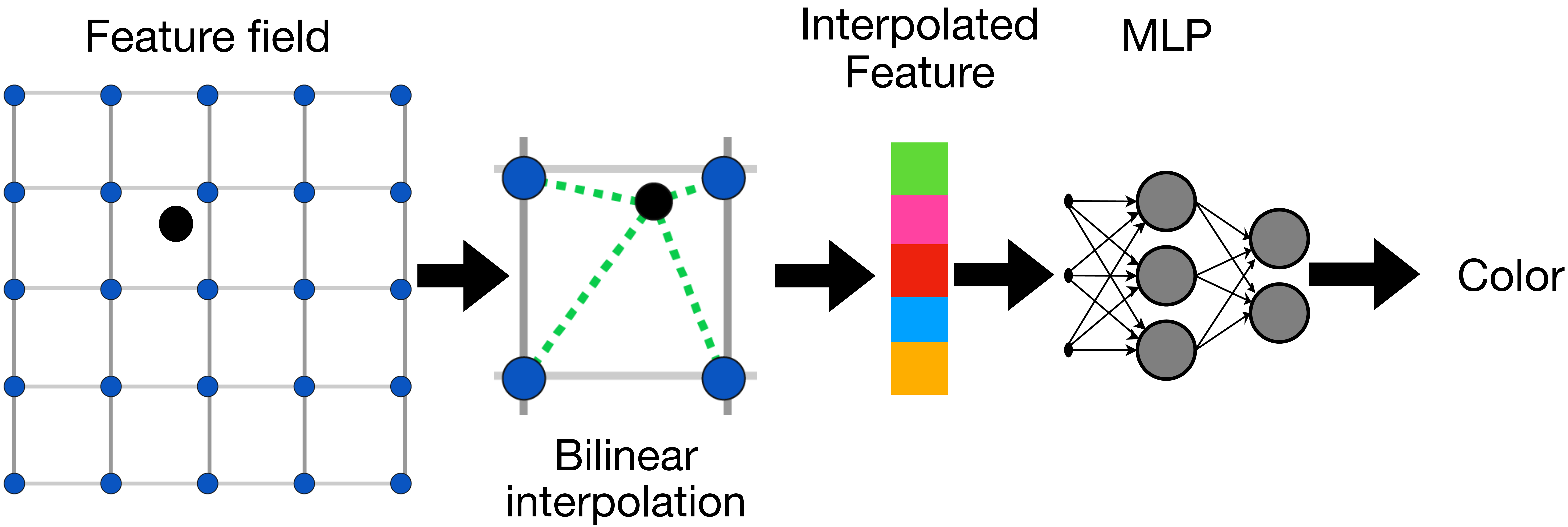
Bilinear interpolation

Interpolated Feature

MLP

Color

Neural fields are continuous by construction

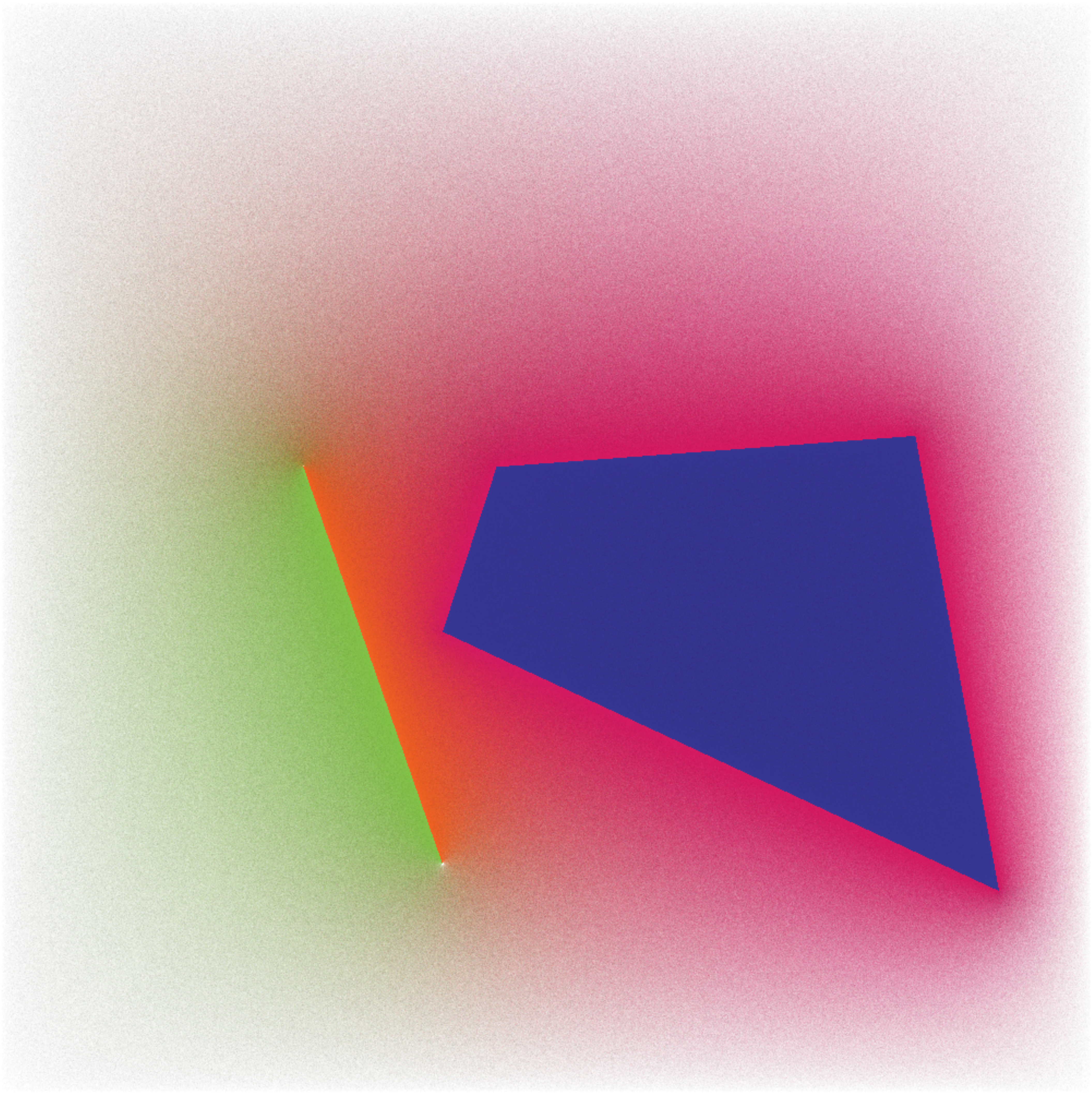


Vector graphics analytically store discontinuity locations

… but they have simplistic shading

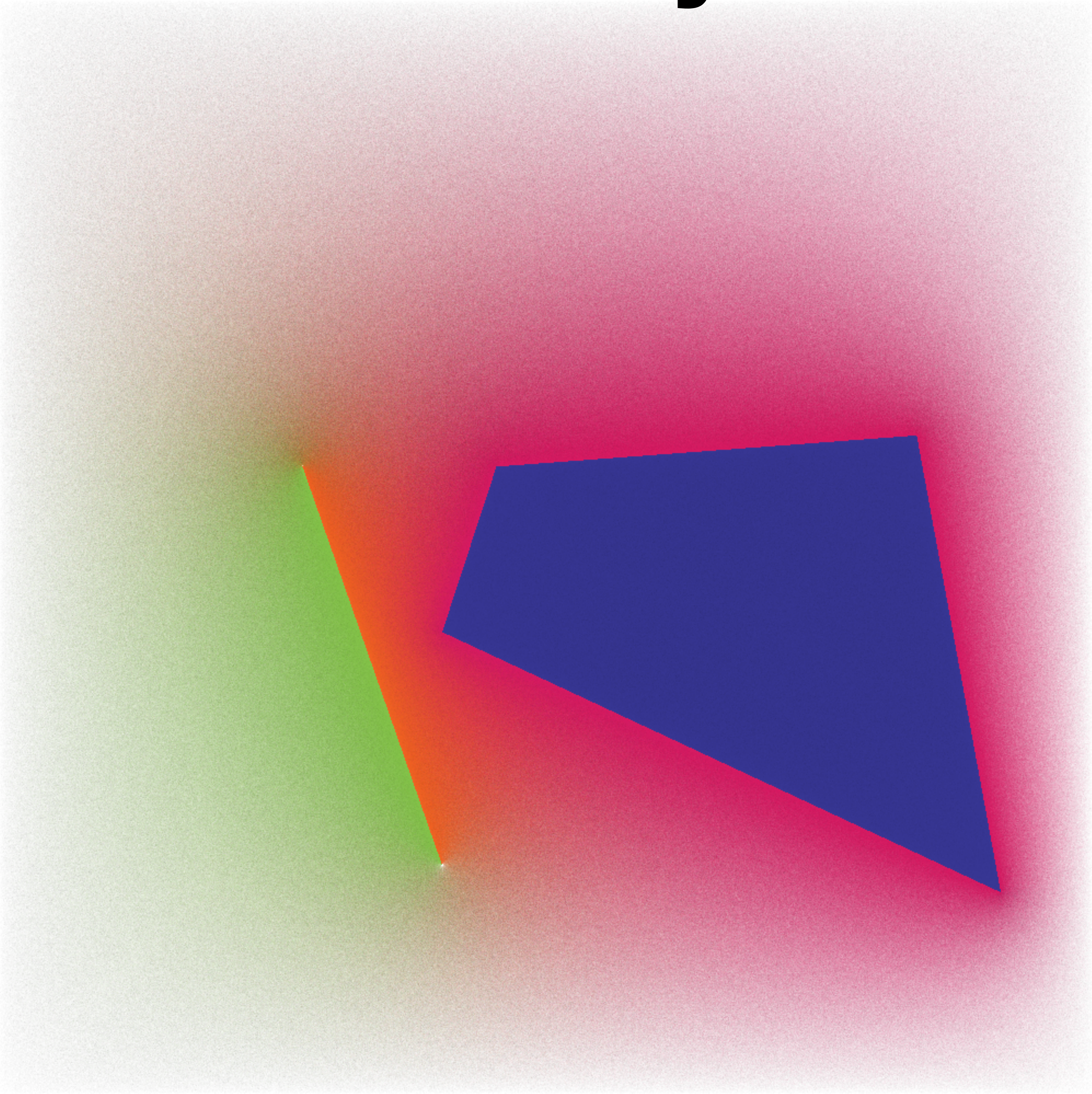
Our goal

Our goal: encode a target image

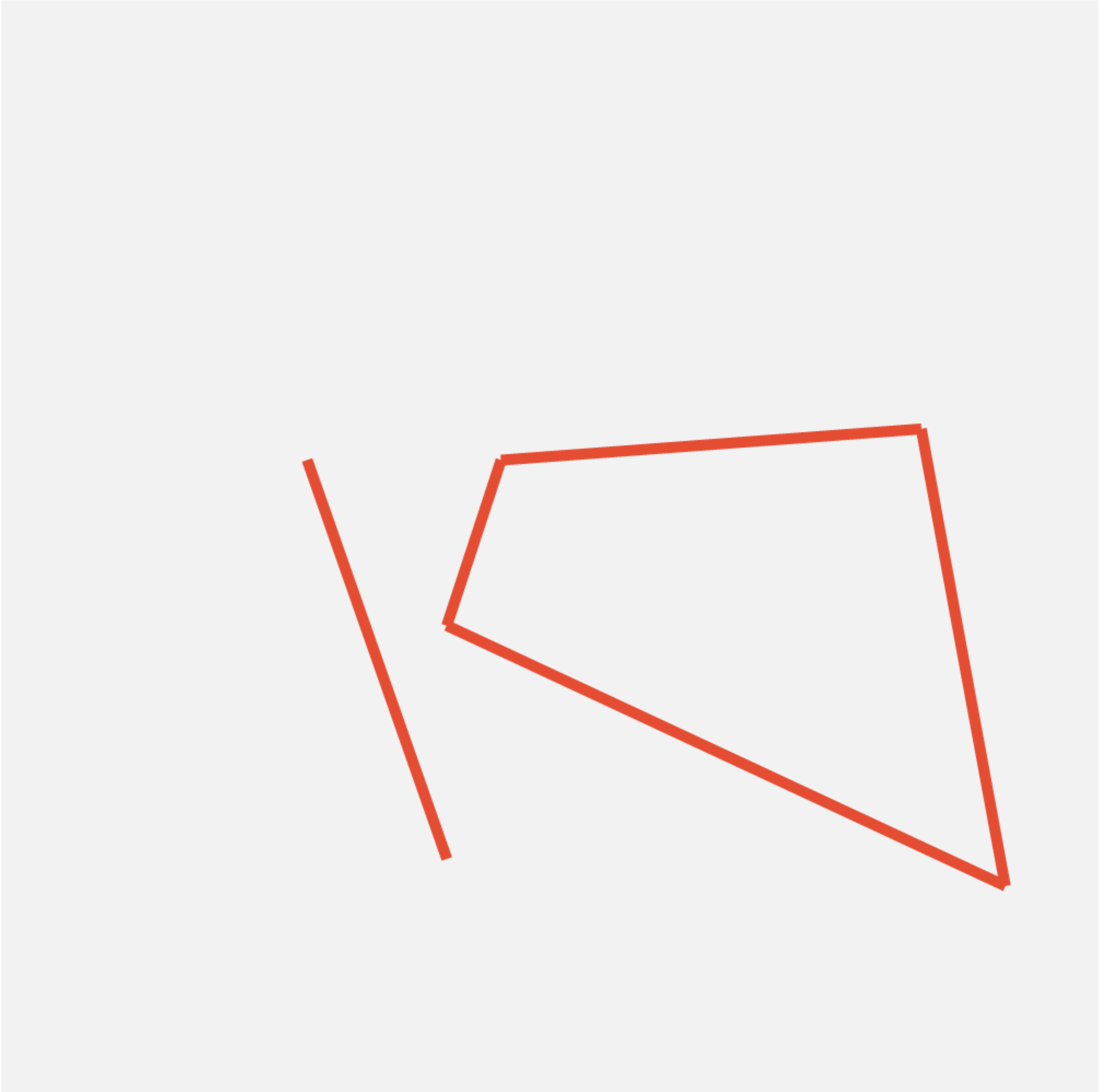


Target image

Our goal: encode a target image given its discontinuitiy locations

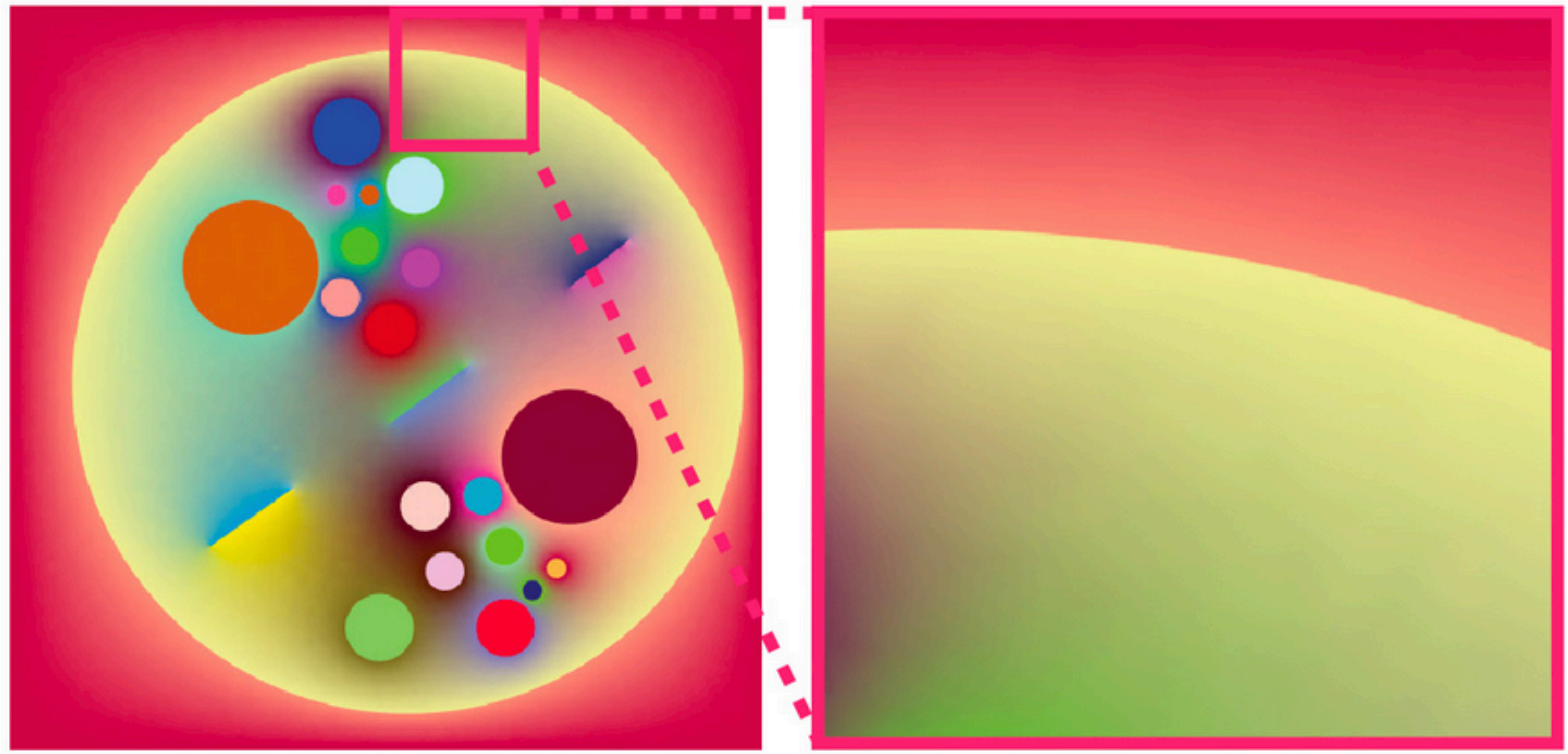


Target image



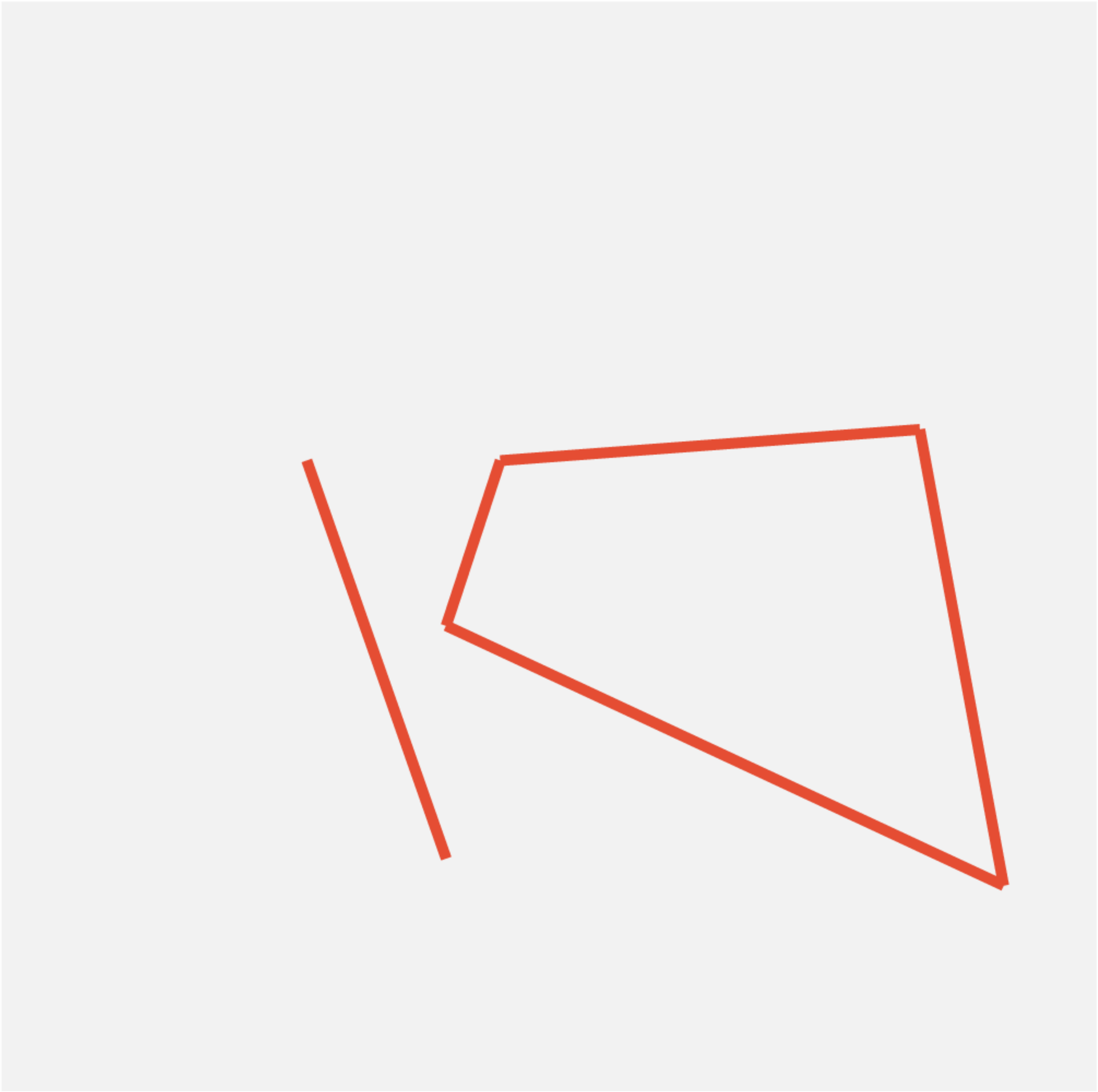
Discontinuity locations

Curved discontinuities



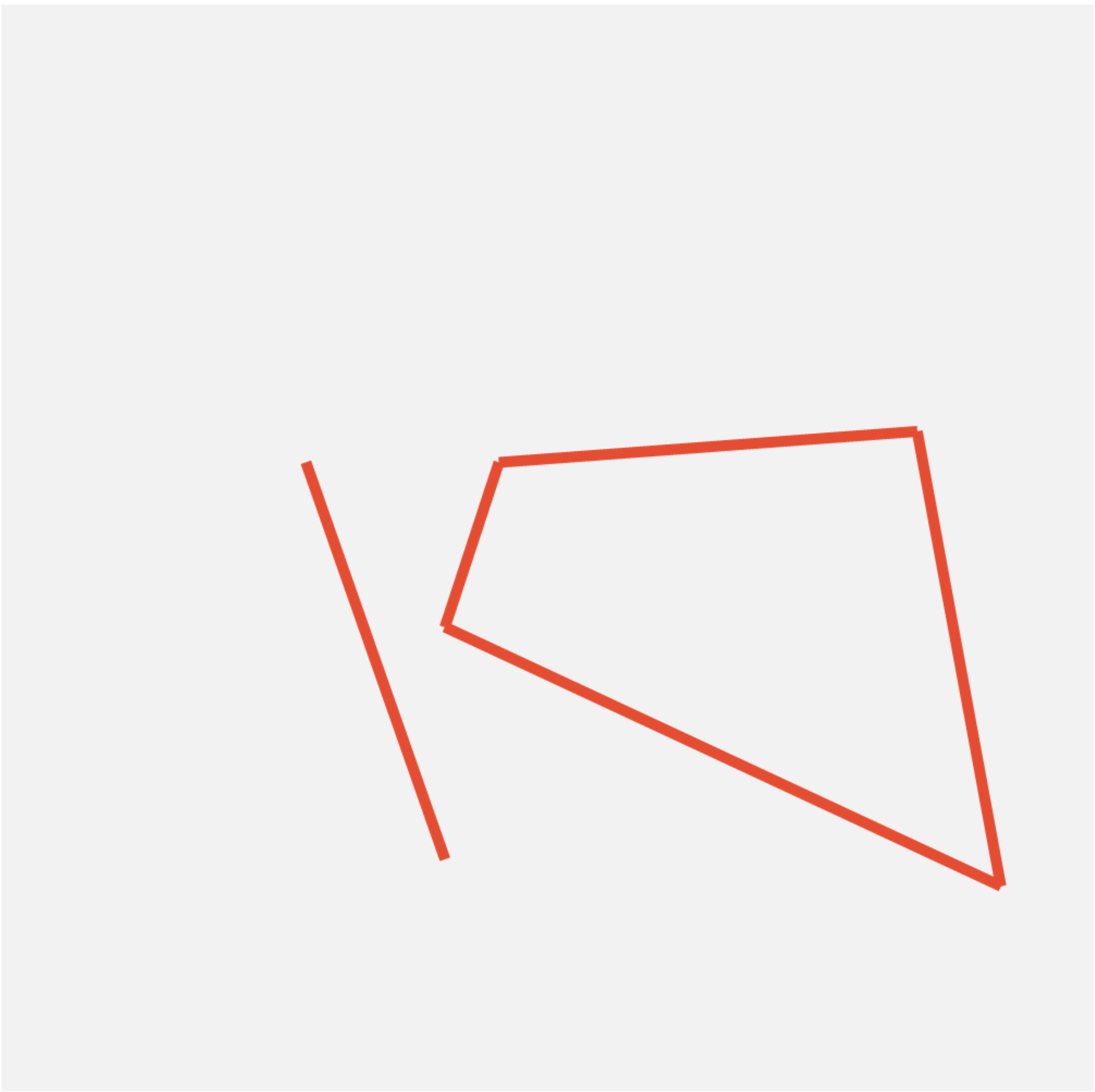
Feature field construction

Feature field construction

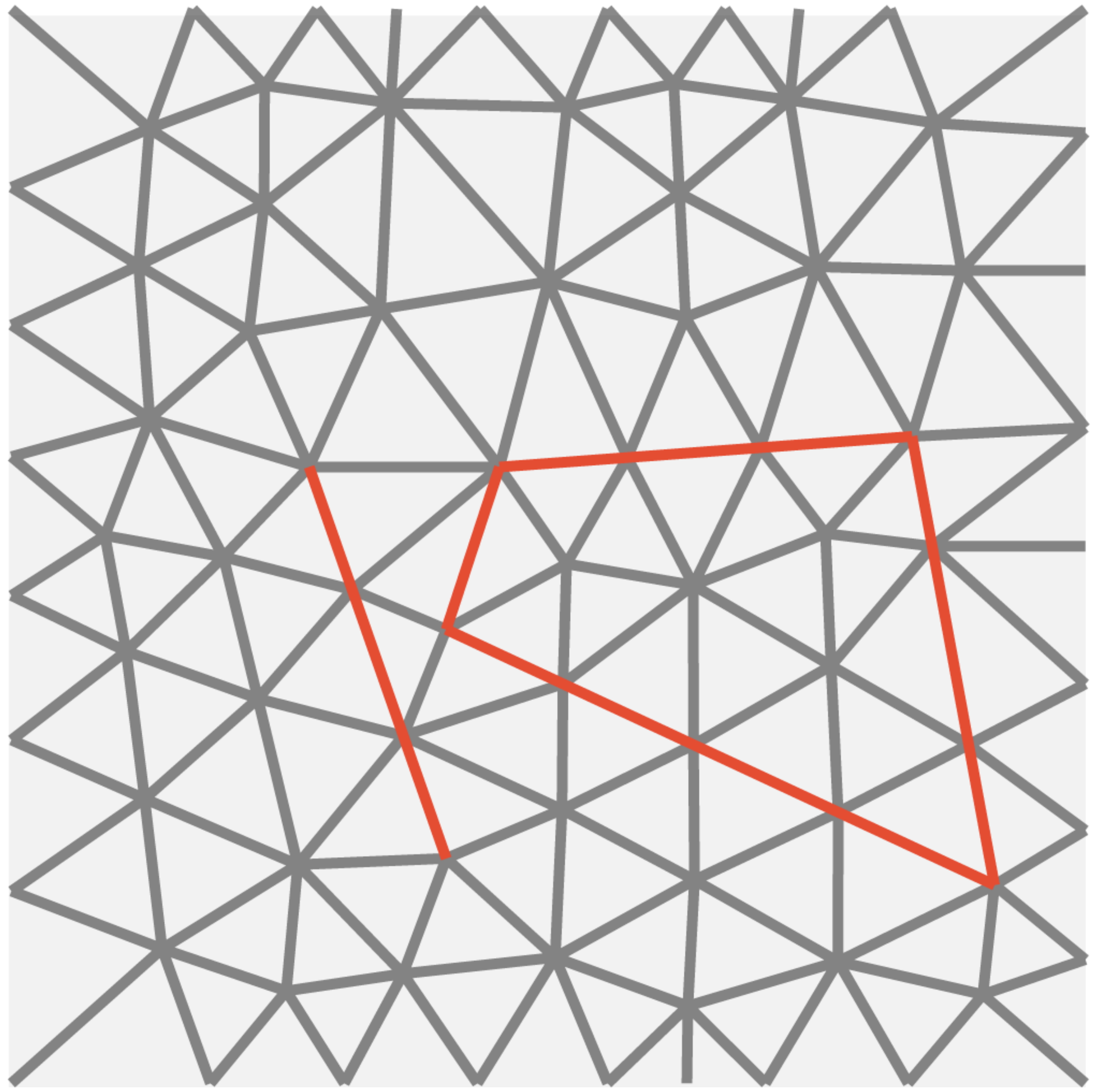


Discontinuity locations

Feature field construction



Discontinuity locations



Triangulation

Hu 19: TriWild: Robust Triangulation with Curve Constraints

Feature field construction

Discontinuity locations

Triangulation

Feature field

Hu 19: TriWild: Robust Triangulation with Curve Constraints

Our feature field is aligned with discontinuities

Discontinuity locations

Triangulation

Feature field

Hu 19: TriWild: Robust Triangulation with Curve Constraints

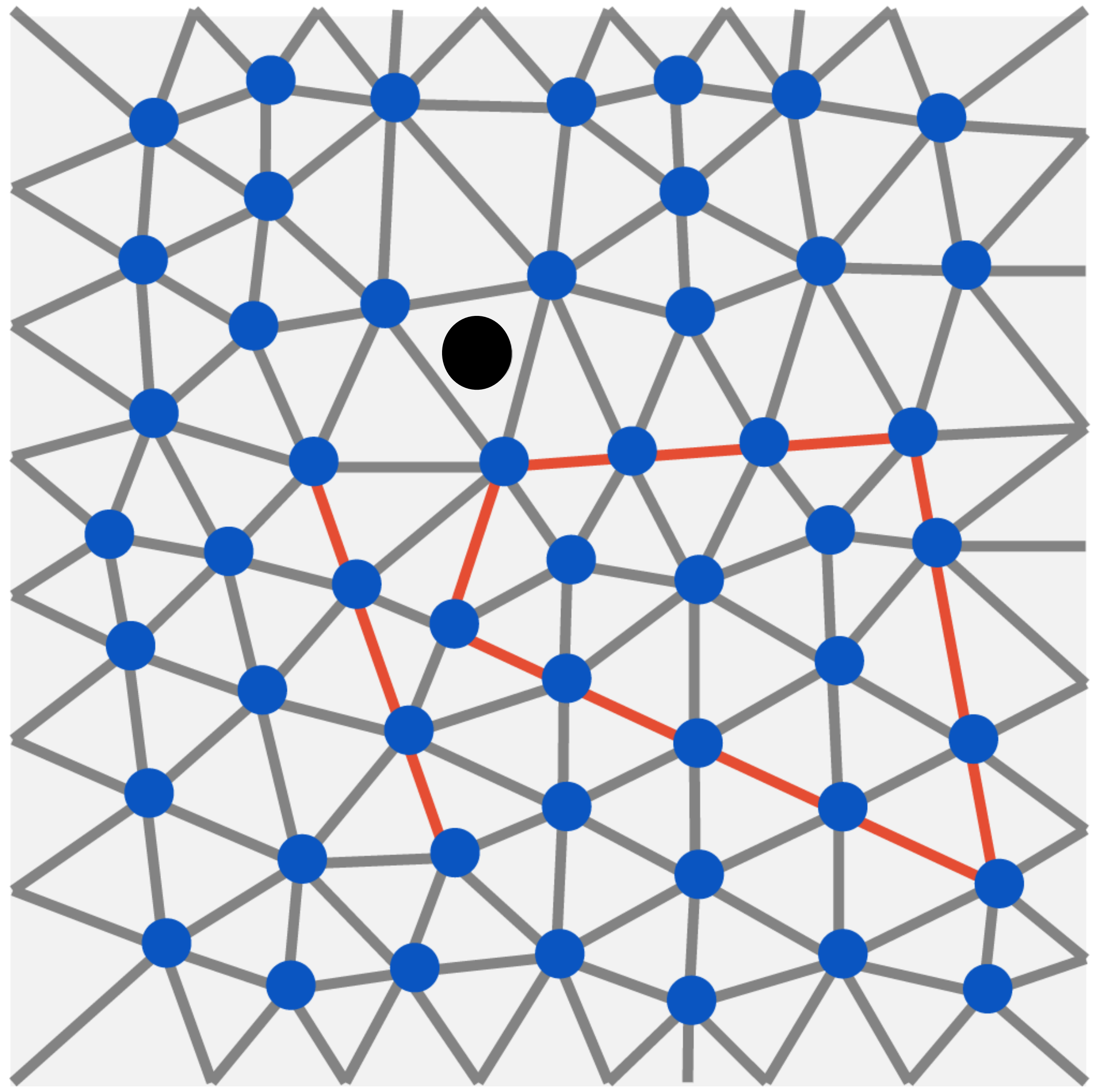
Our rendering pipeline

Mapping queries to colors

•

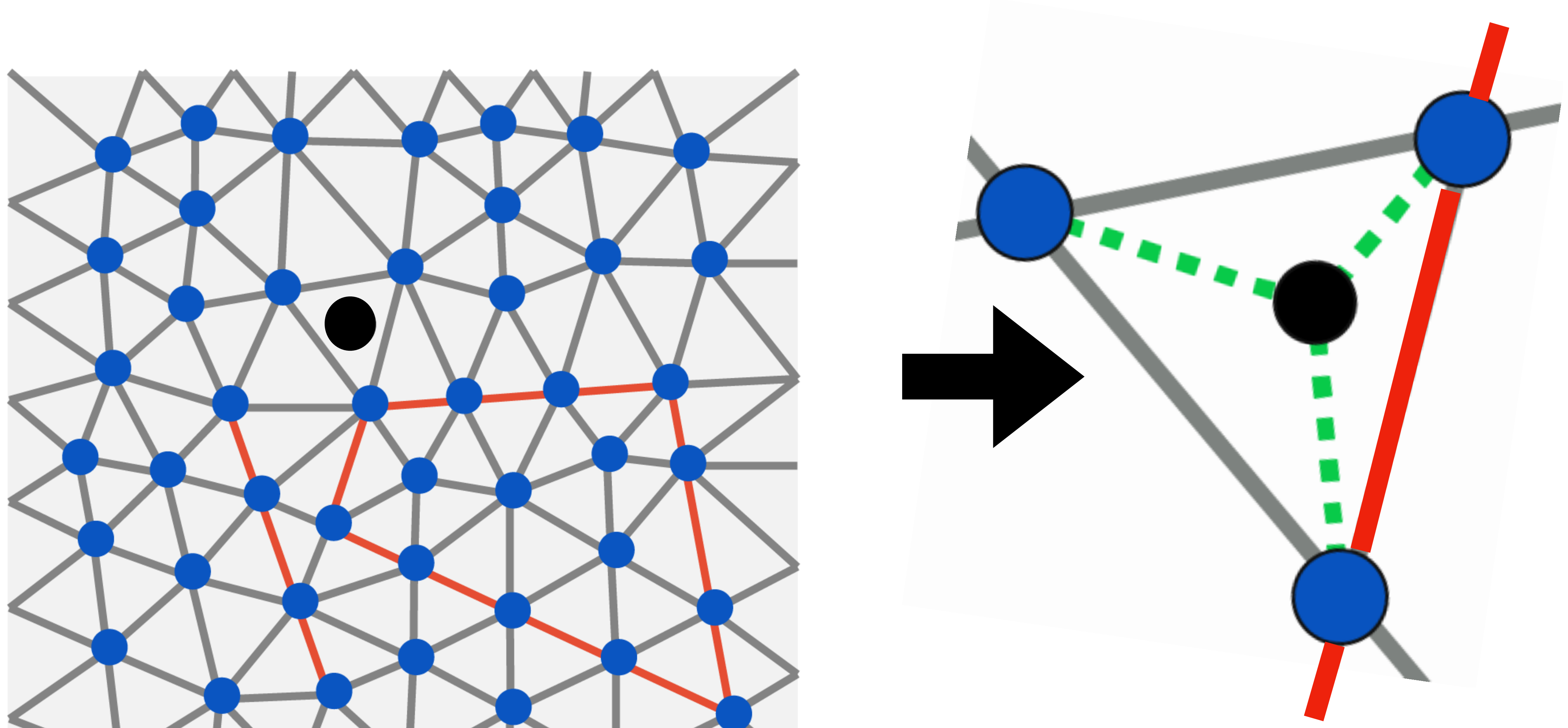
Mapping queries to colors

Feature field



Mapping queries to colors

Feature field

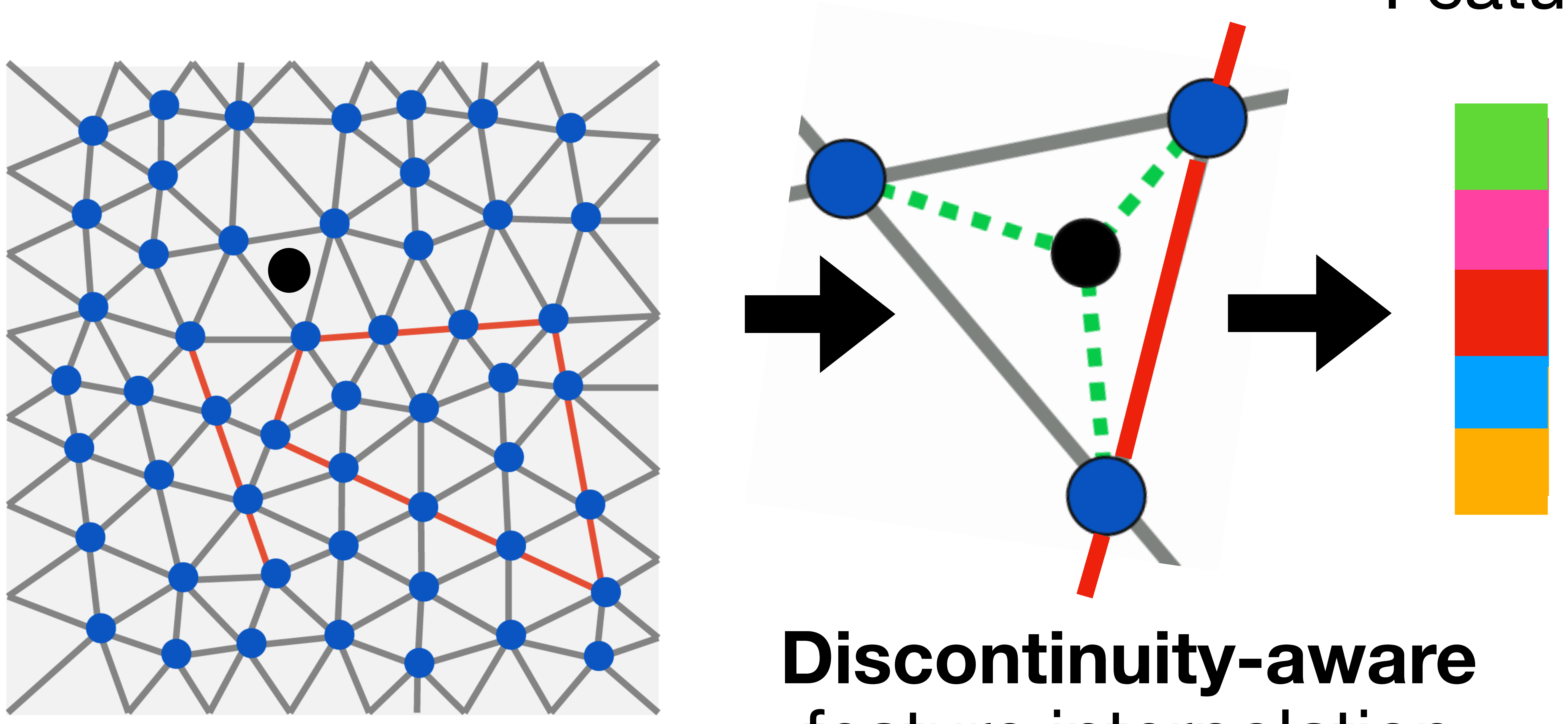


Discontinuity-aware
feature interpolation

Mapping queries to colors

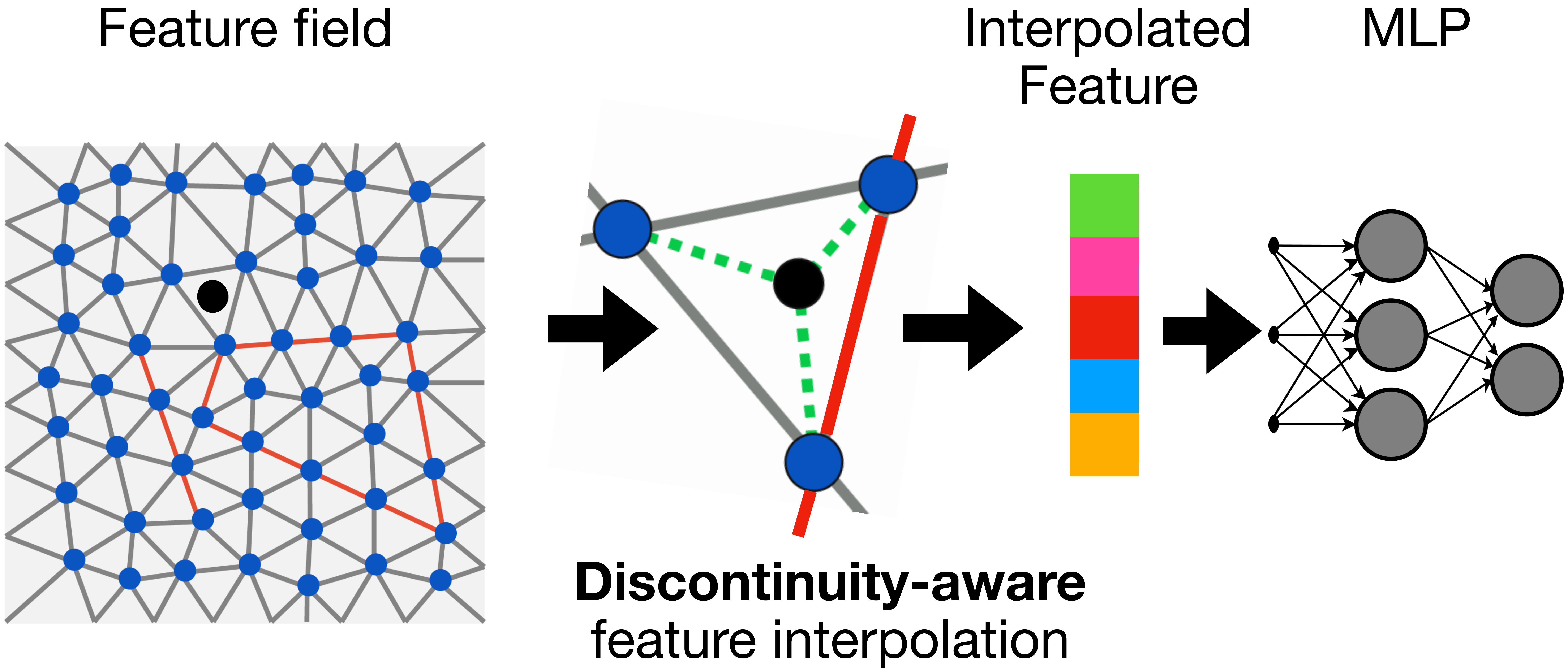
Feature field

Interpolated Feature



Discontinuity-aware feature interpolation

Mapping queries to colors



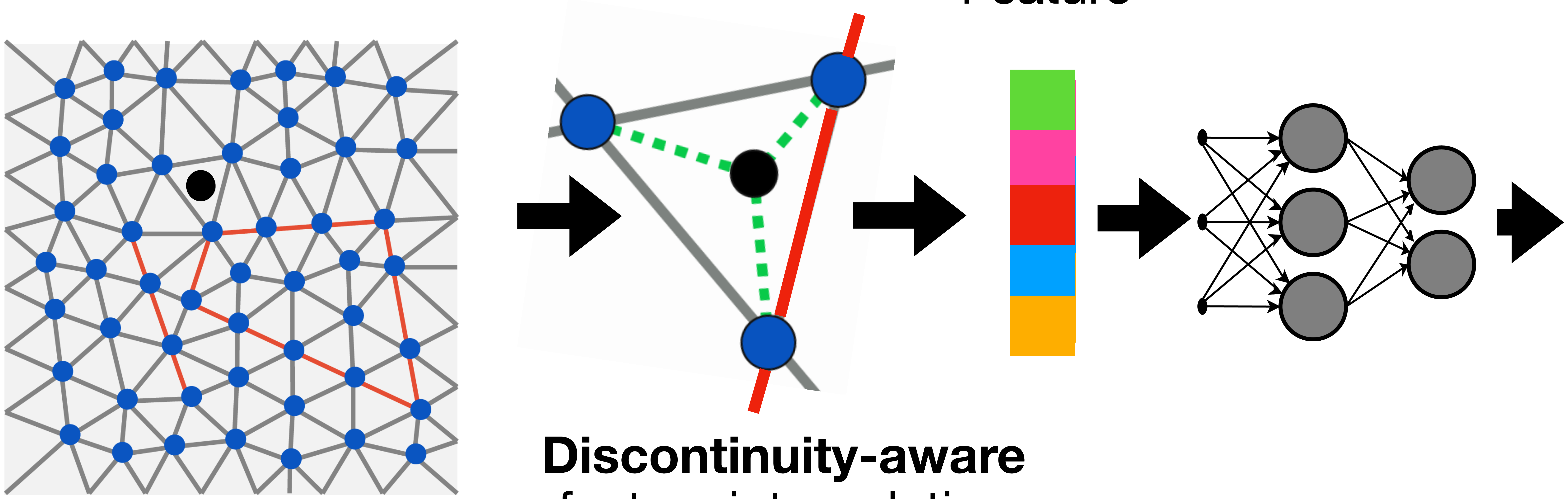
Feature field

Interpolated Feature

MLP

Discontinuity-aware feature interpolation

Mapping queries to colors



Feature field

Discontinuity-aware feature interpolation

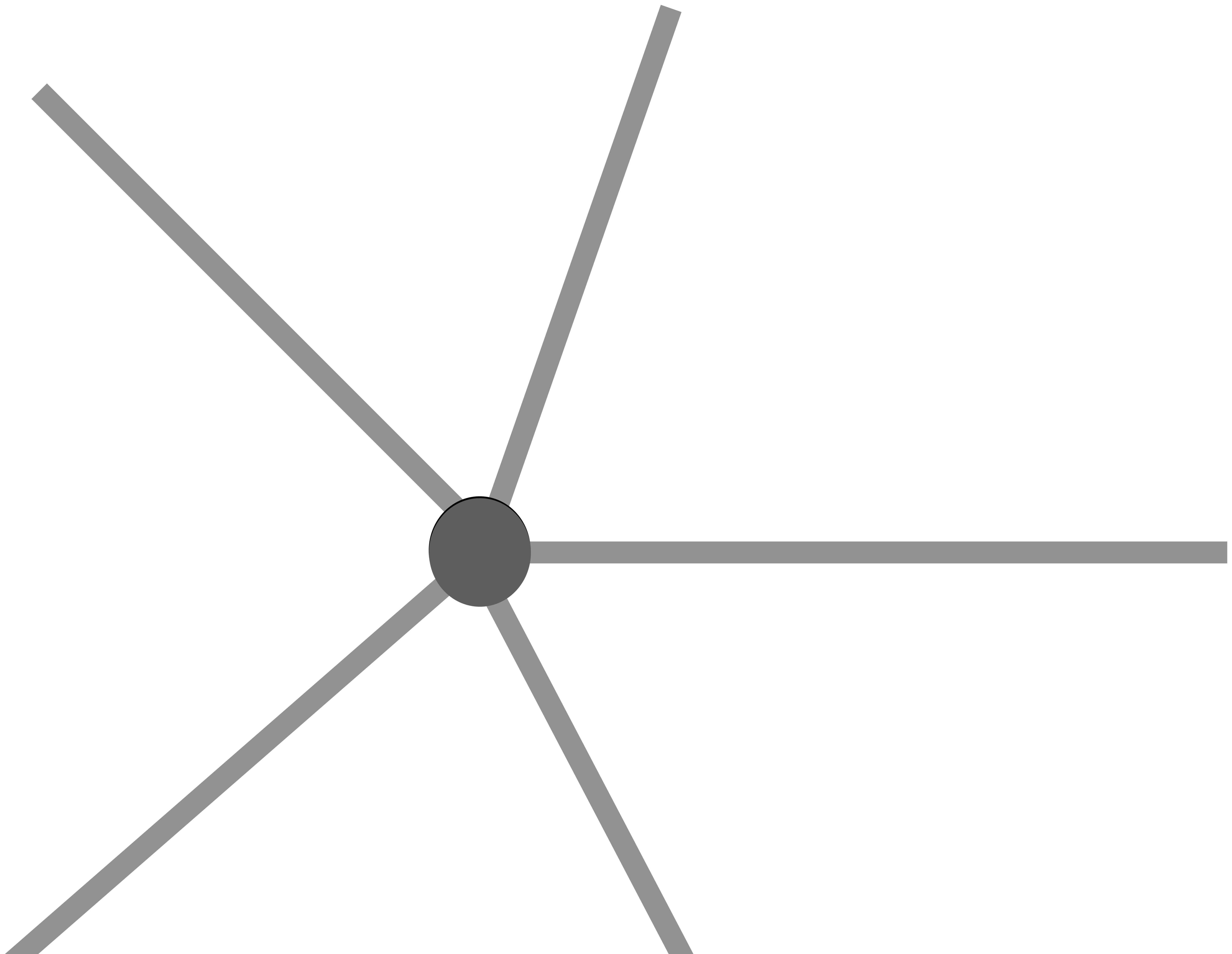
Interpolated Feature

MLP

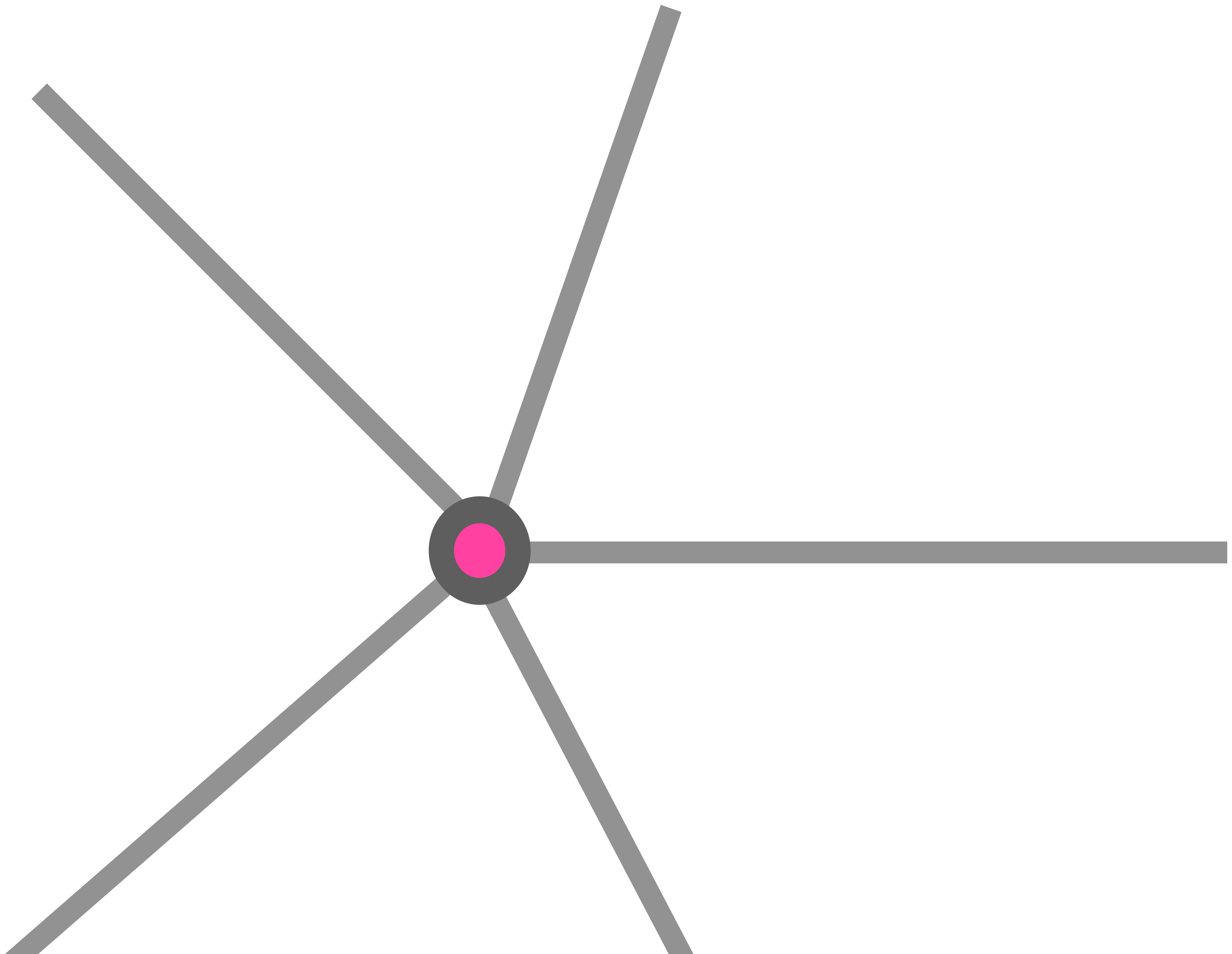
Color

Discontinuity-aware feature interpolation

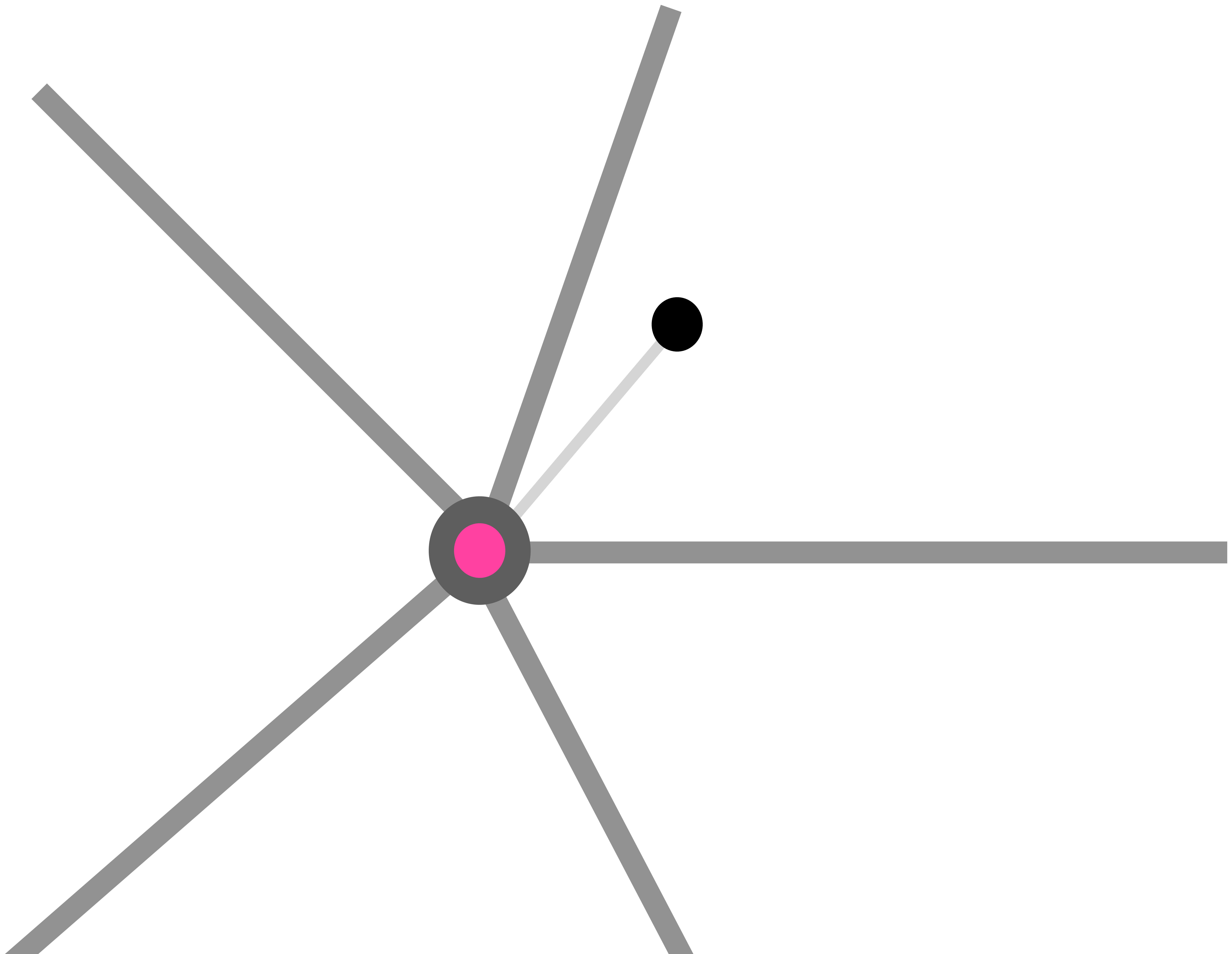
Continuous vertex



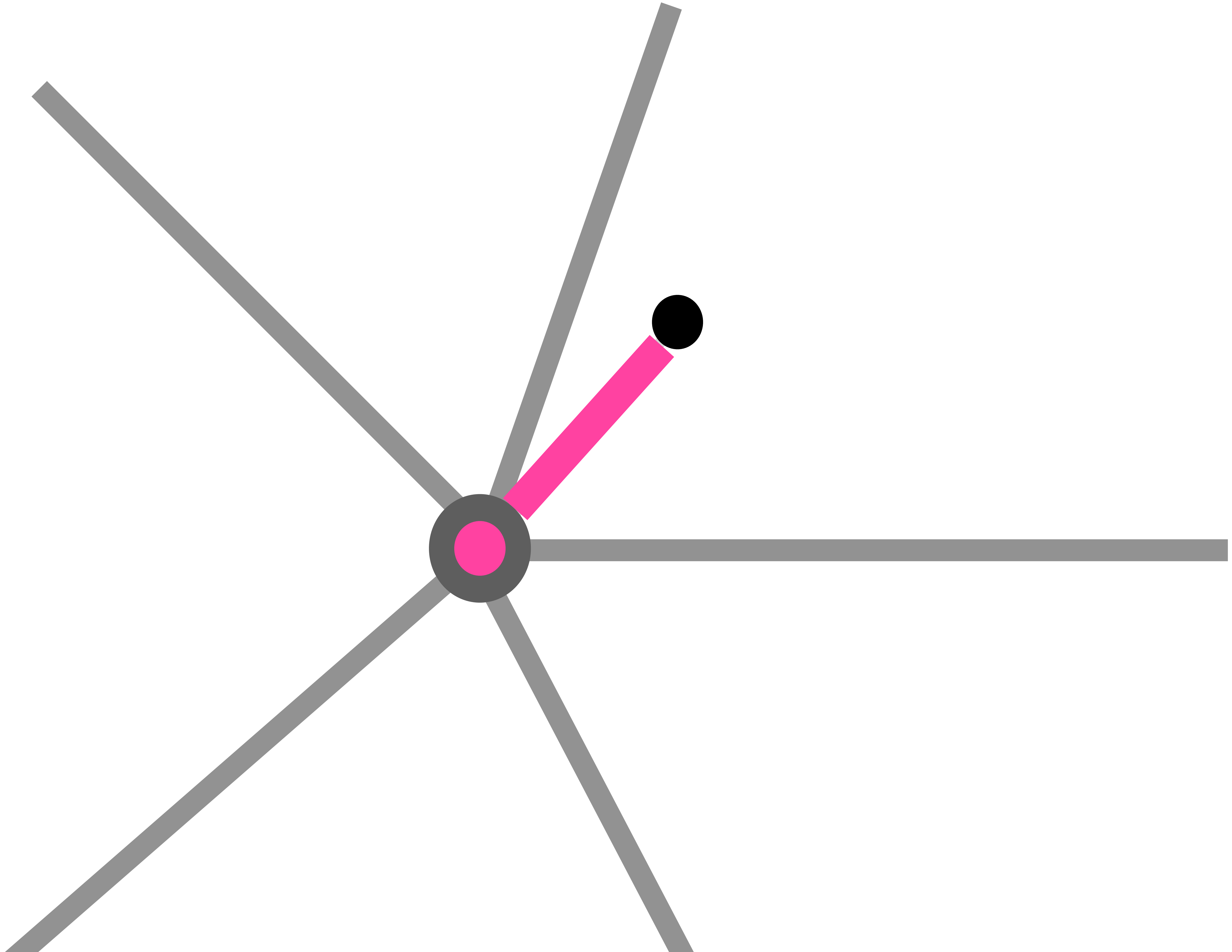
Continuous vertex



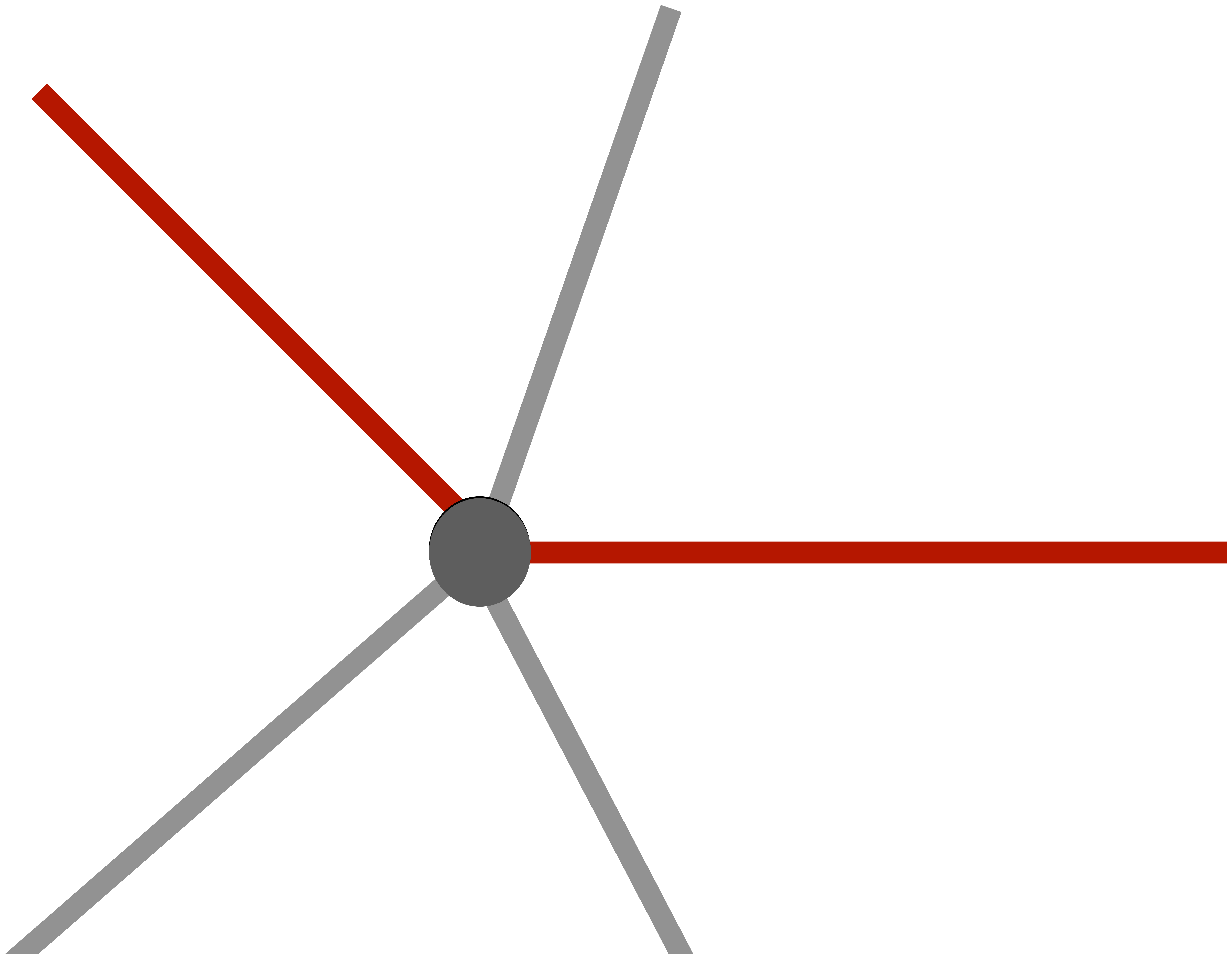
Continuous vertex



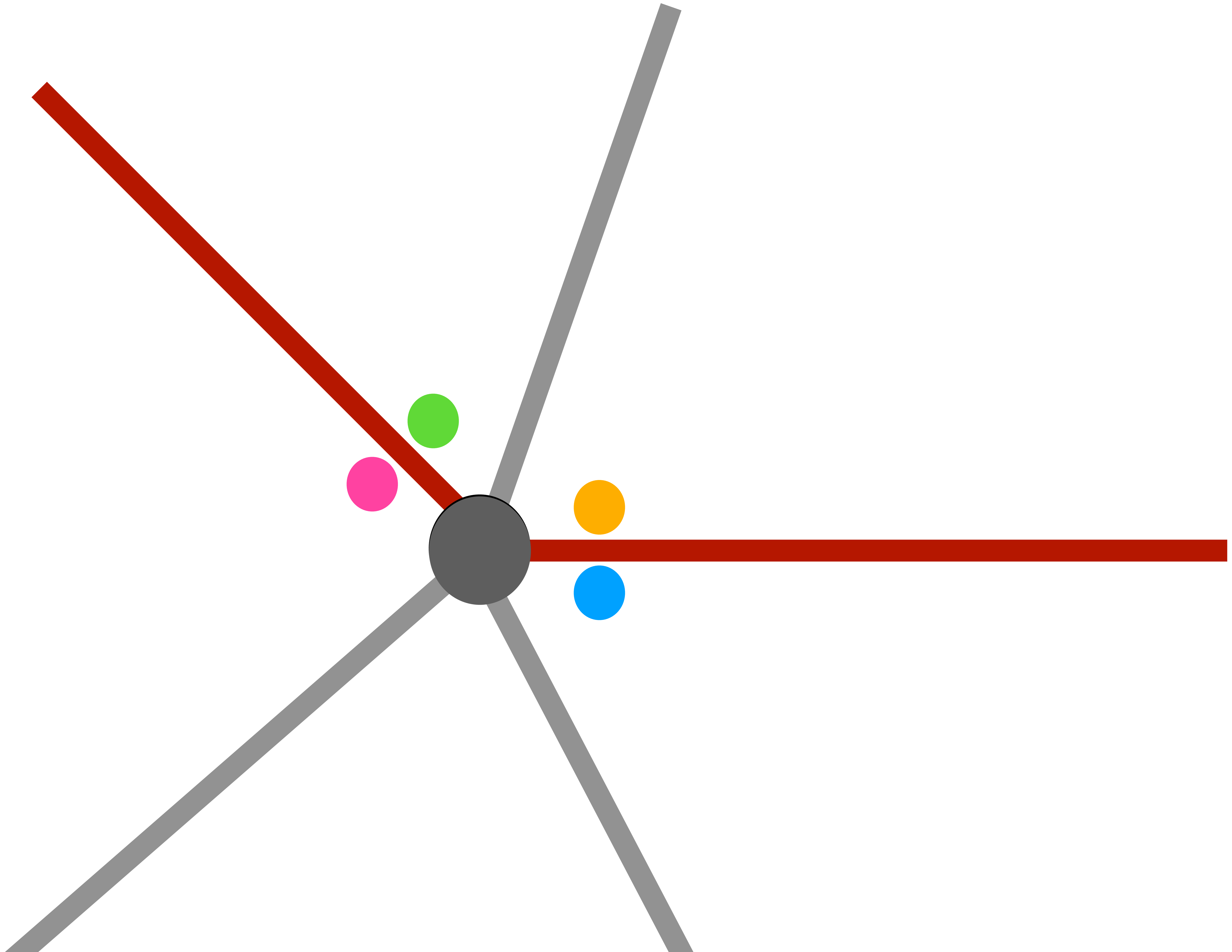
Continuous vertex



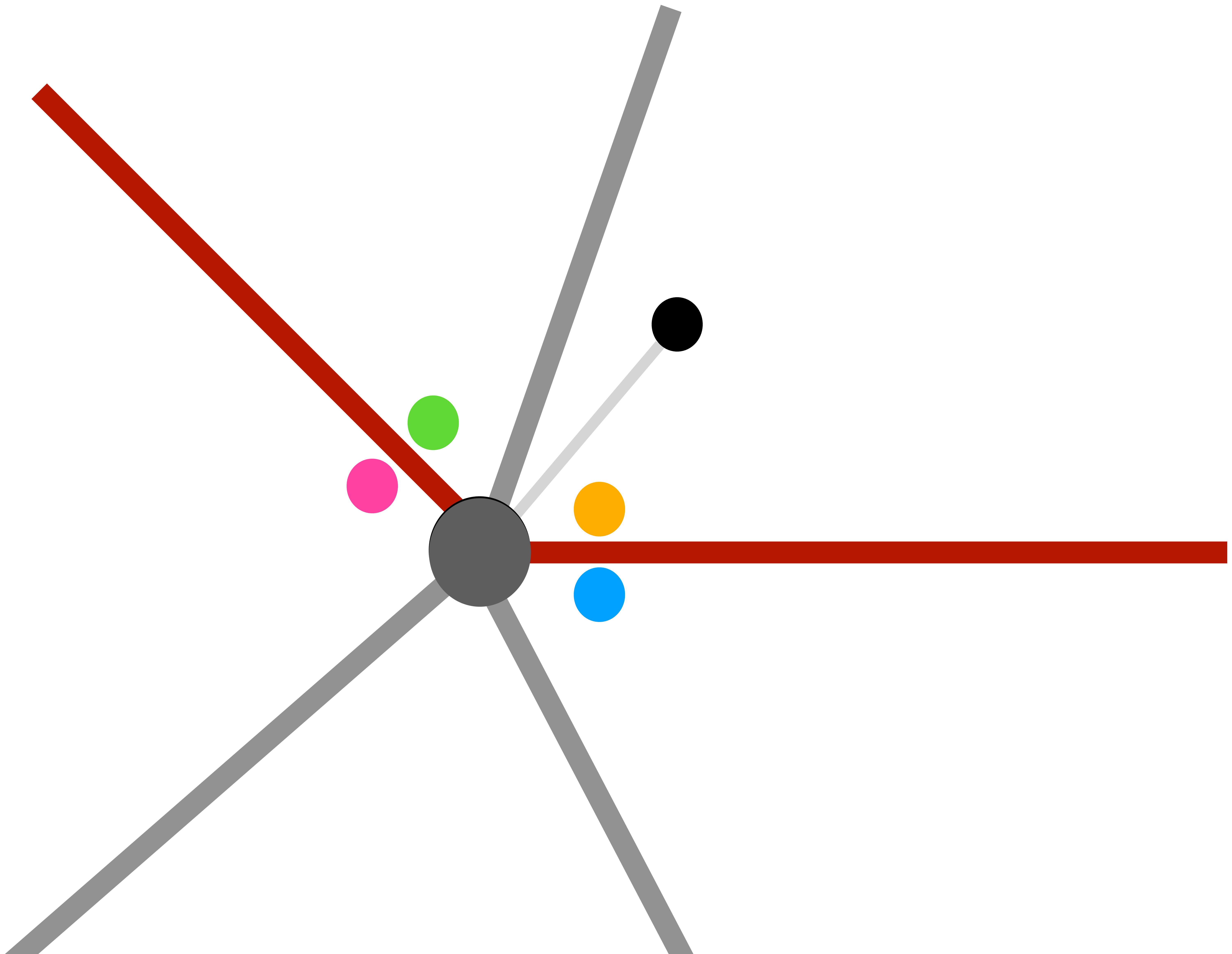
Discontinuous vertex



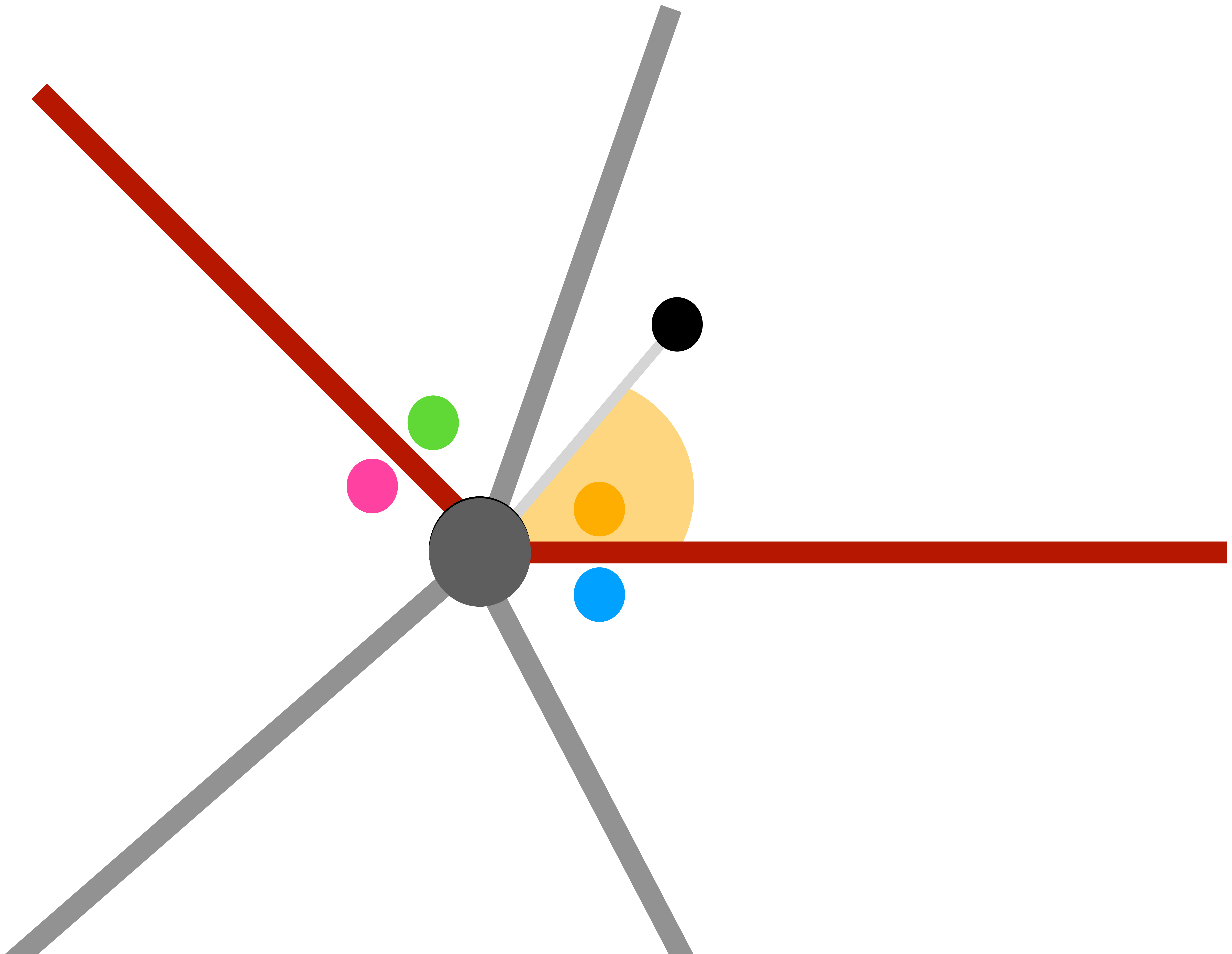
Different features above and below each discontinuity



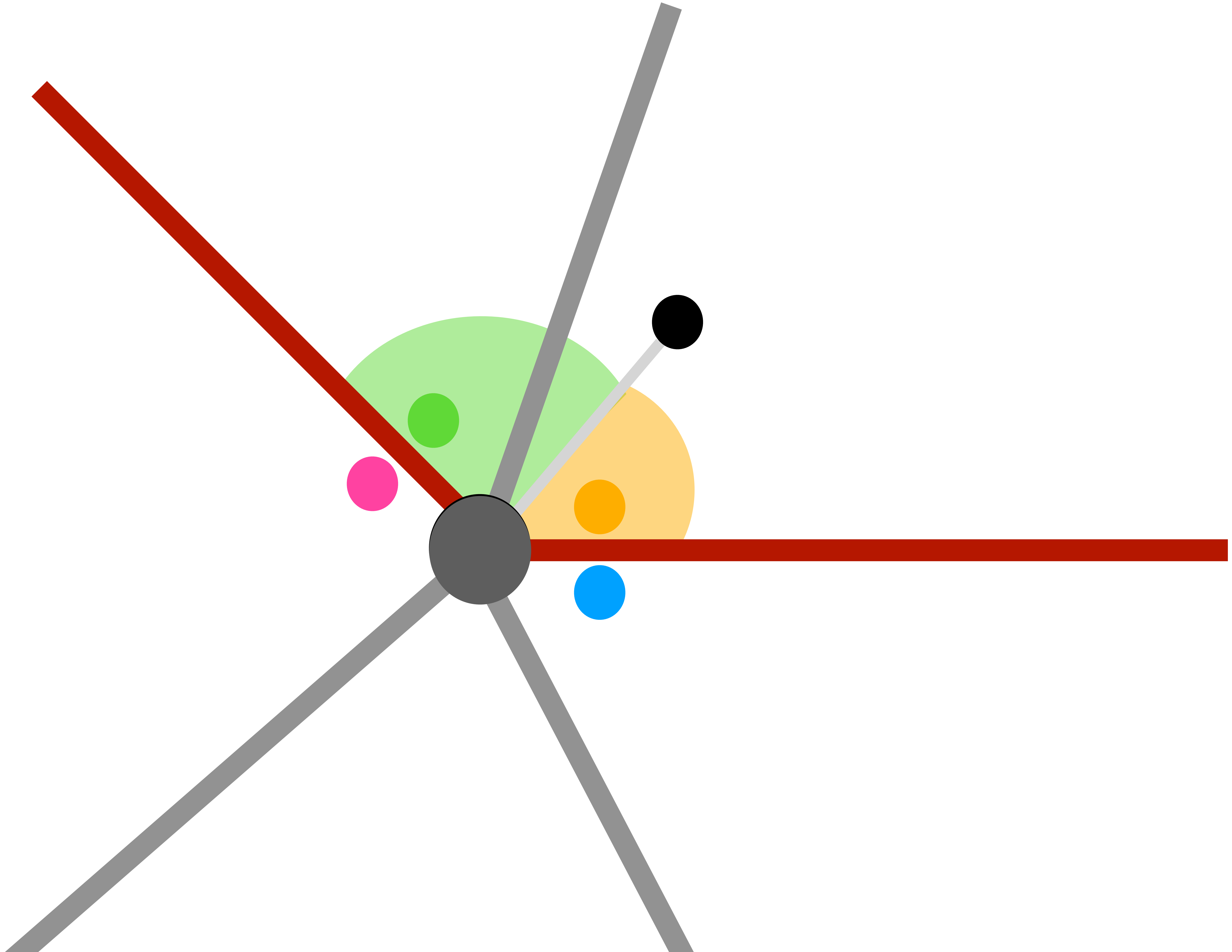
Evaluating vertex feature for a query point



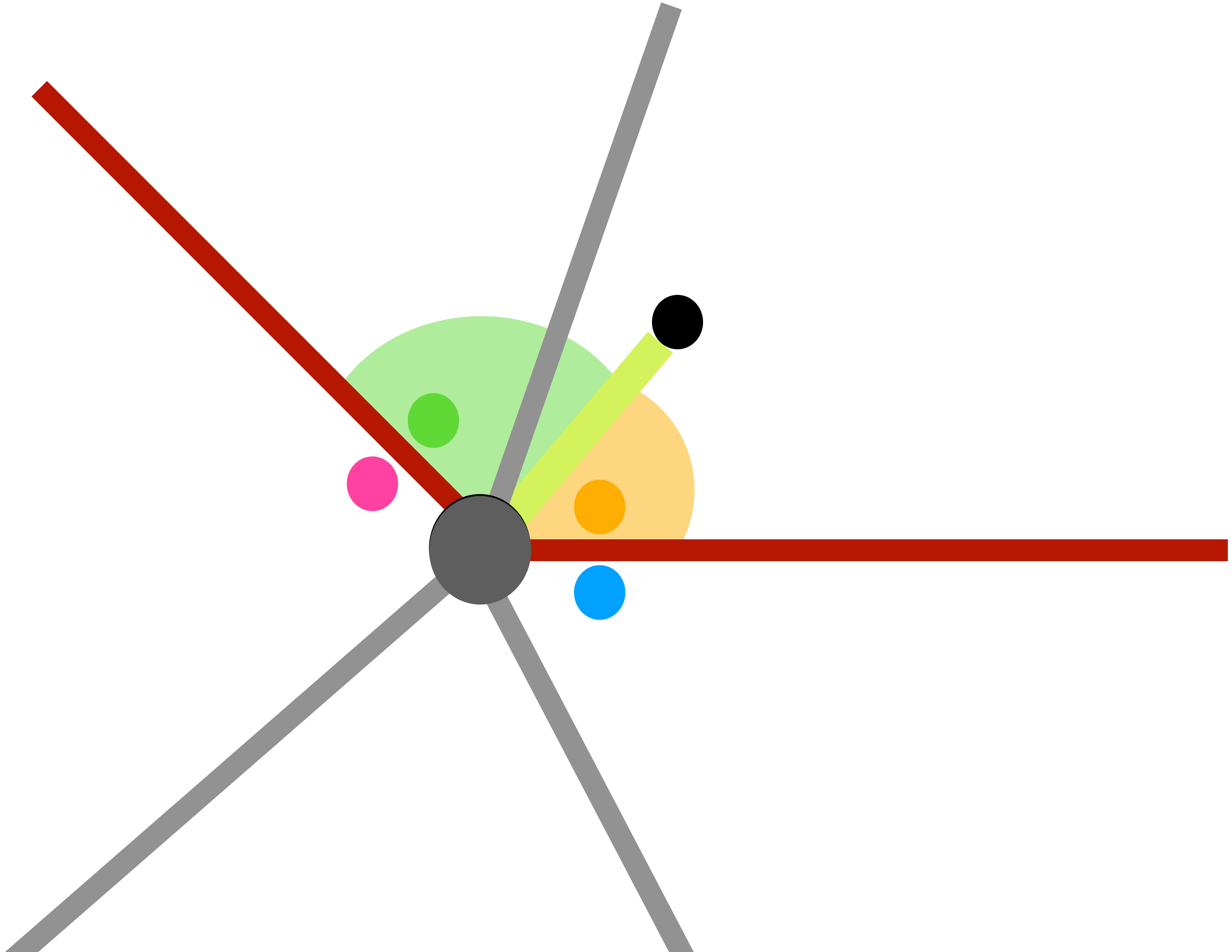
Closest clockwise feature



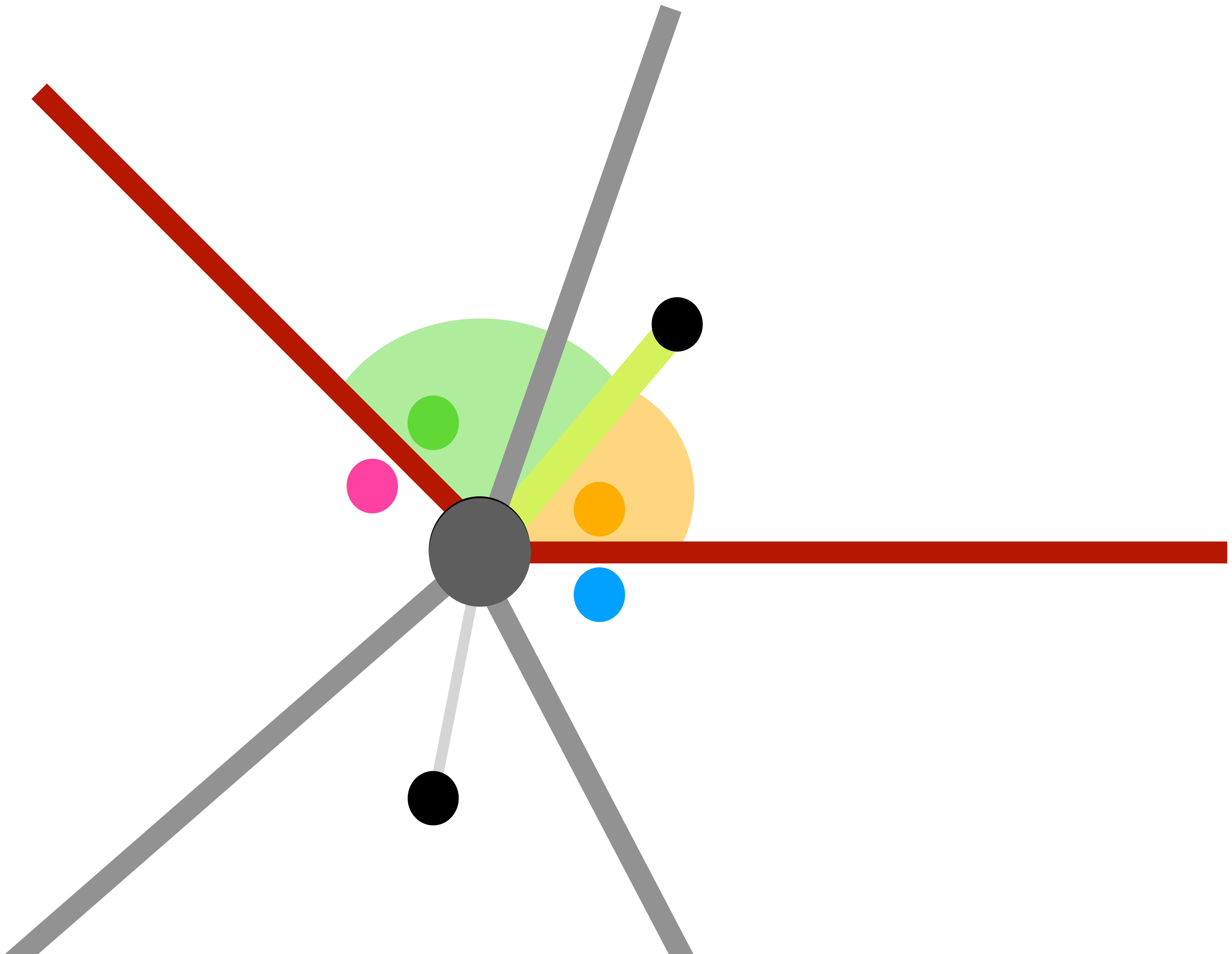
Closest counter-clockwise feature



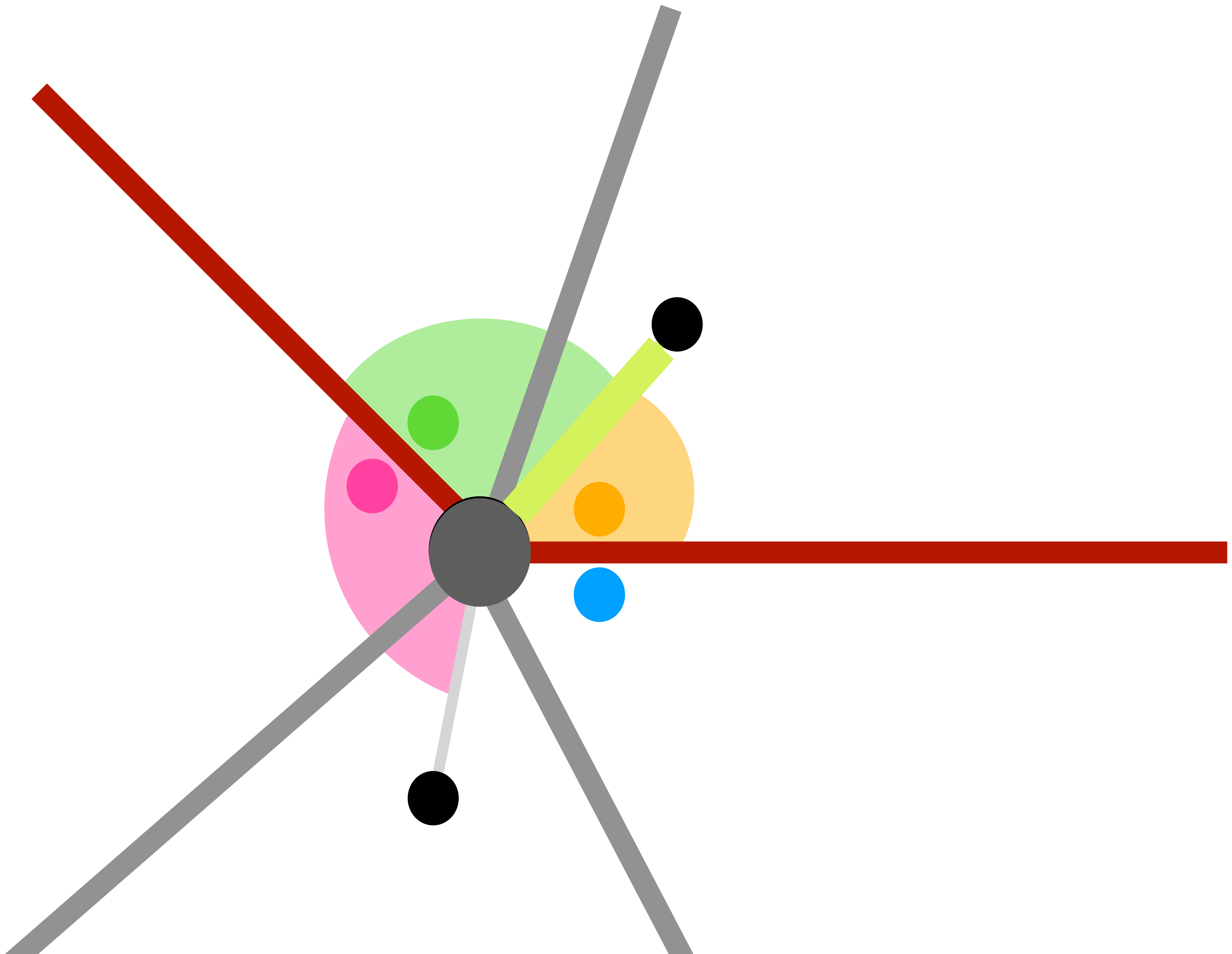
Vertex feature = radially interpolate closest features



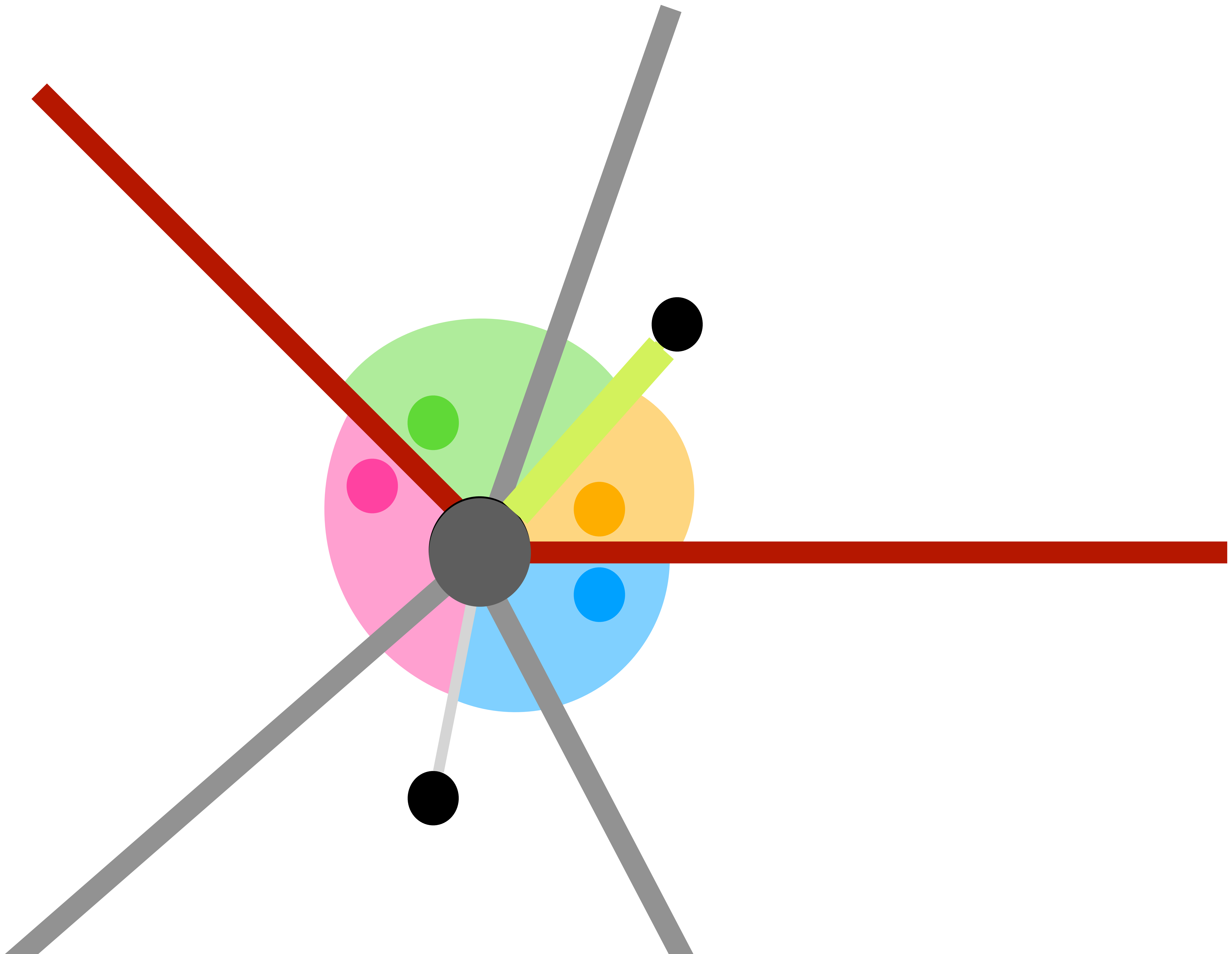
Closest features change on other side of discontinuity



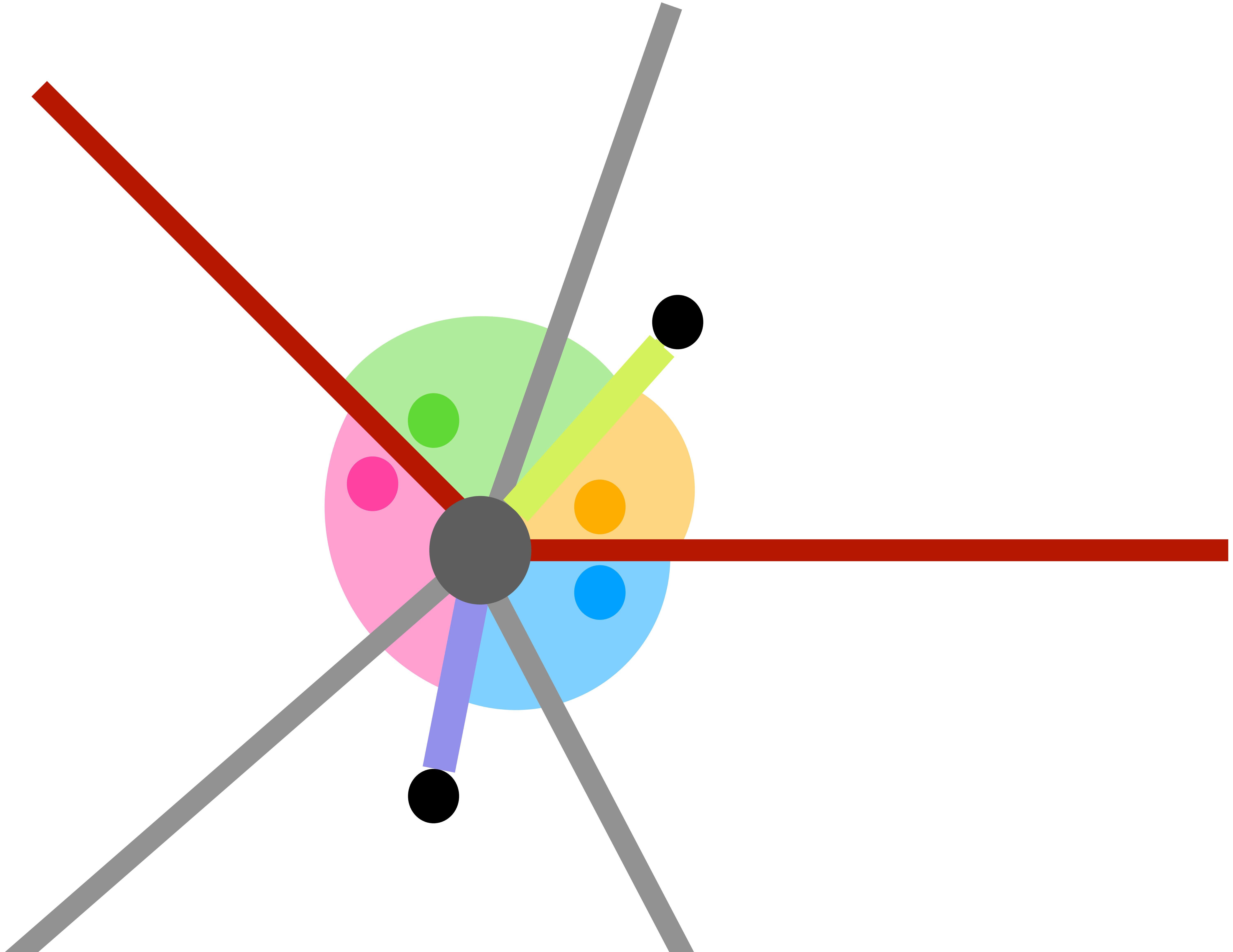
Closest features change on other side of discontinuity



Closest features change on other side of discontinuity

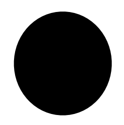


Closest features change on other side of discontinuity

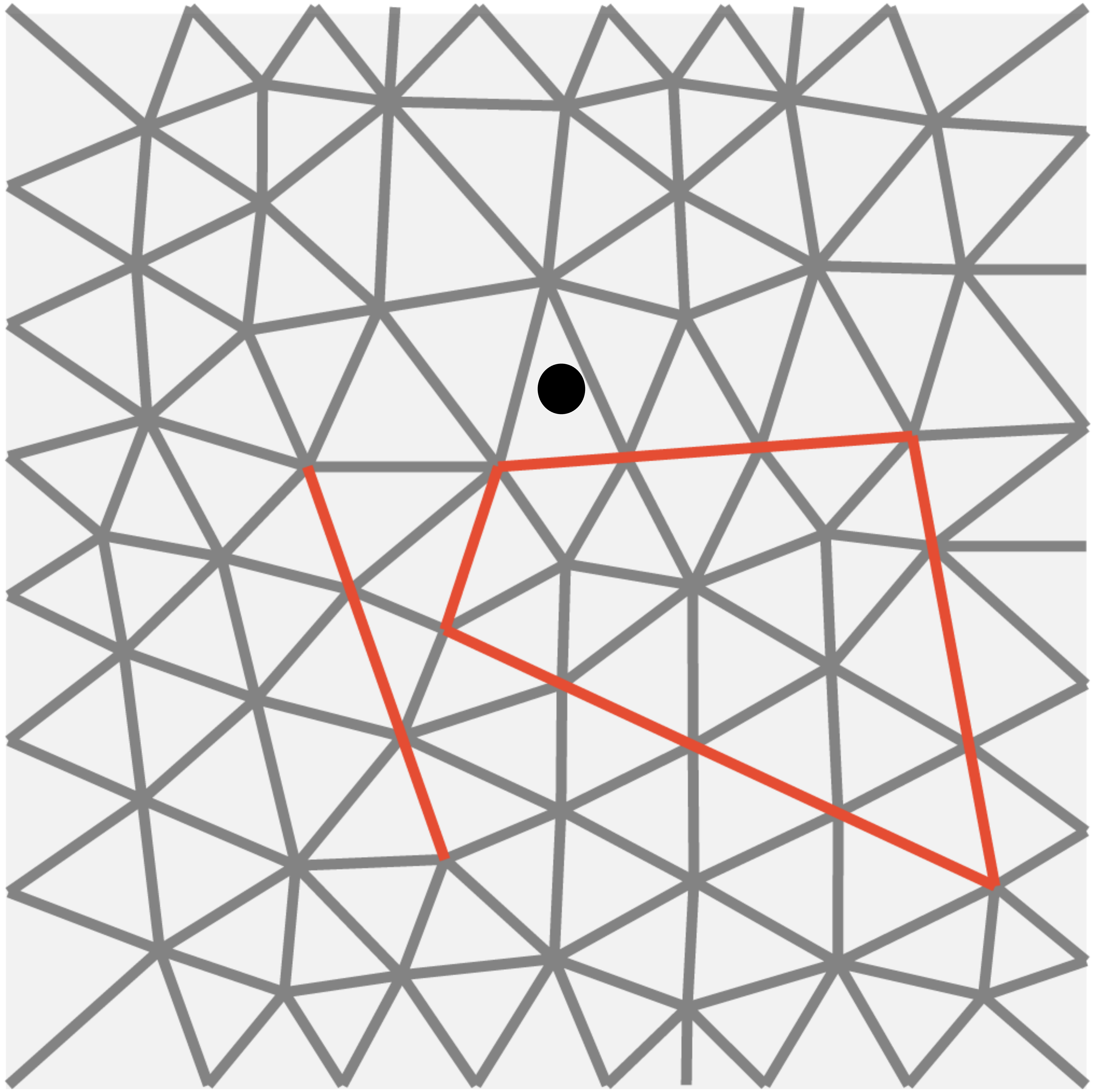


Putting it all together

Query point



Find triangle that contains query point



Zooming in to query point

Directly retrieve feature for continuous vertex

Retrieve features for discontinuous vertices

Find closest features

Radially interpolate closest features

Retrieve features for discontinuous vertices

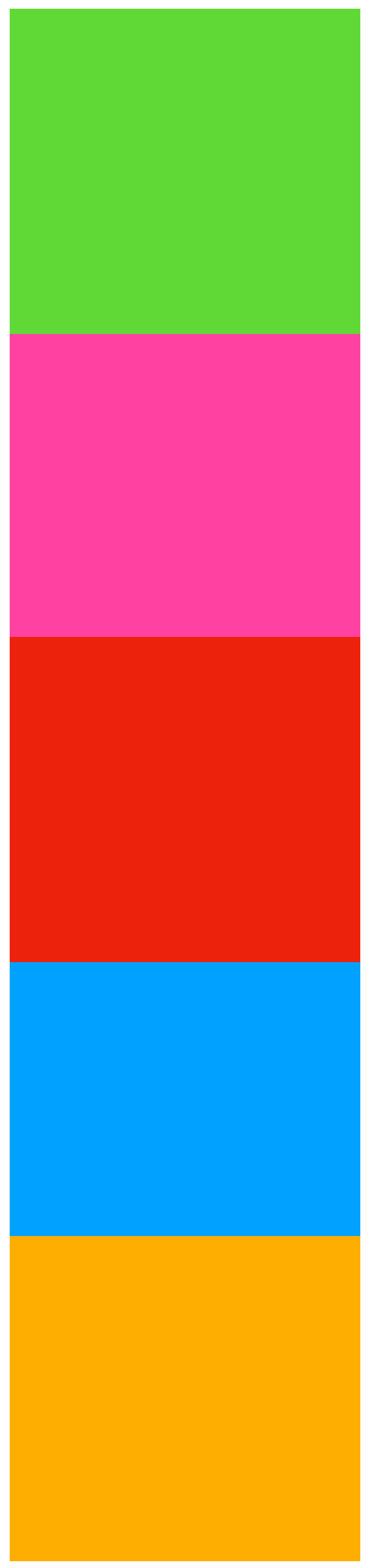
Find closest features

Radially interpolate nearest features

Barycentrically interpolate three vertex features

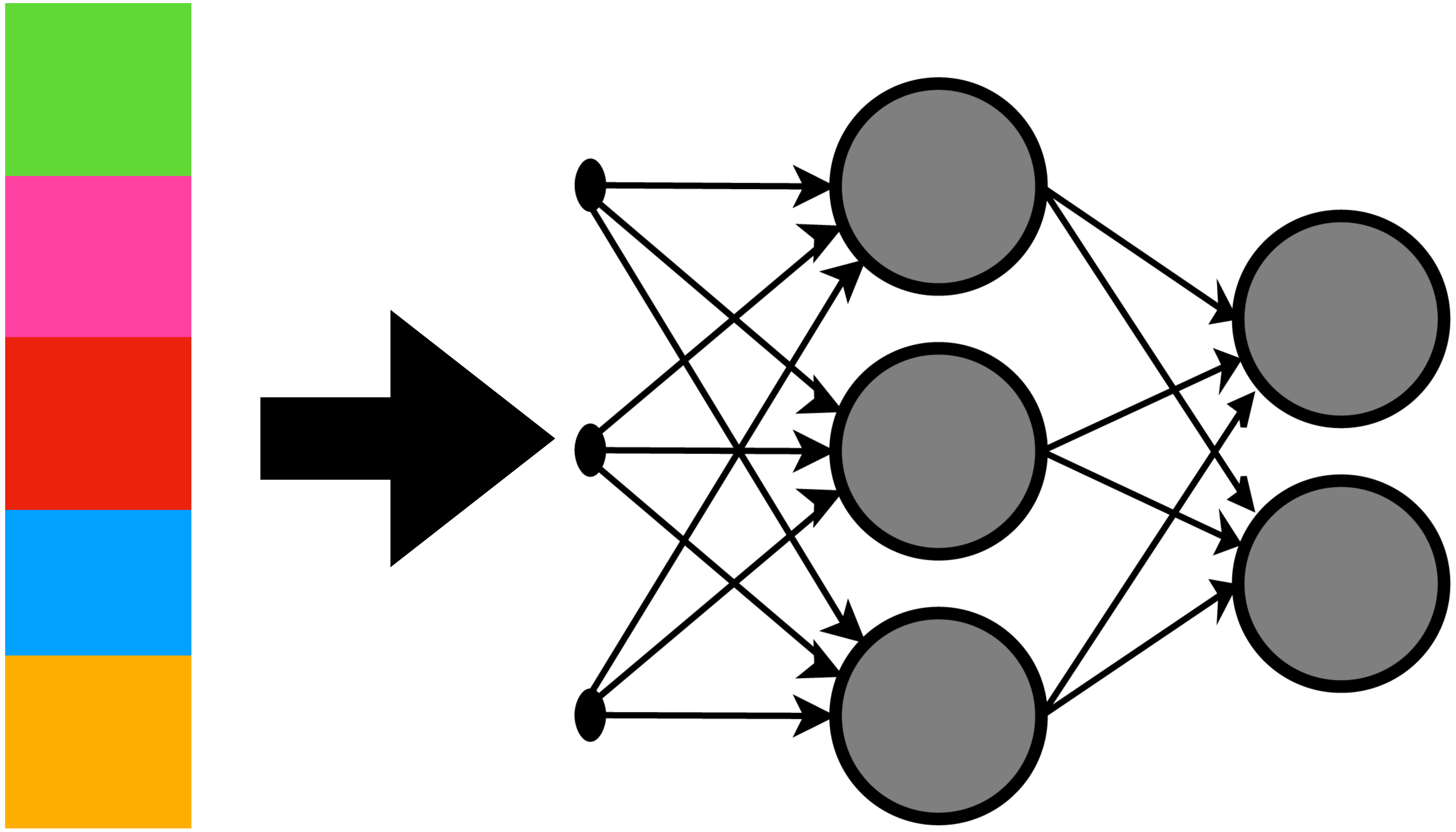
Barycentrically interpolate three vertex features

Decode interpolated features using MLP



Interpolated feature

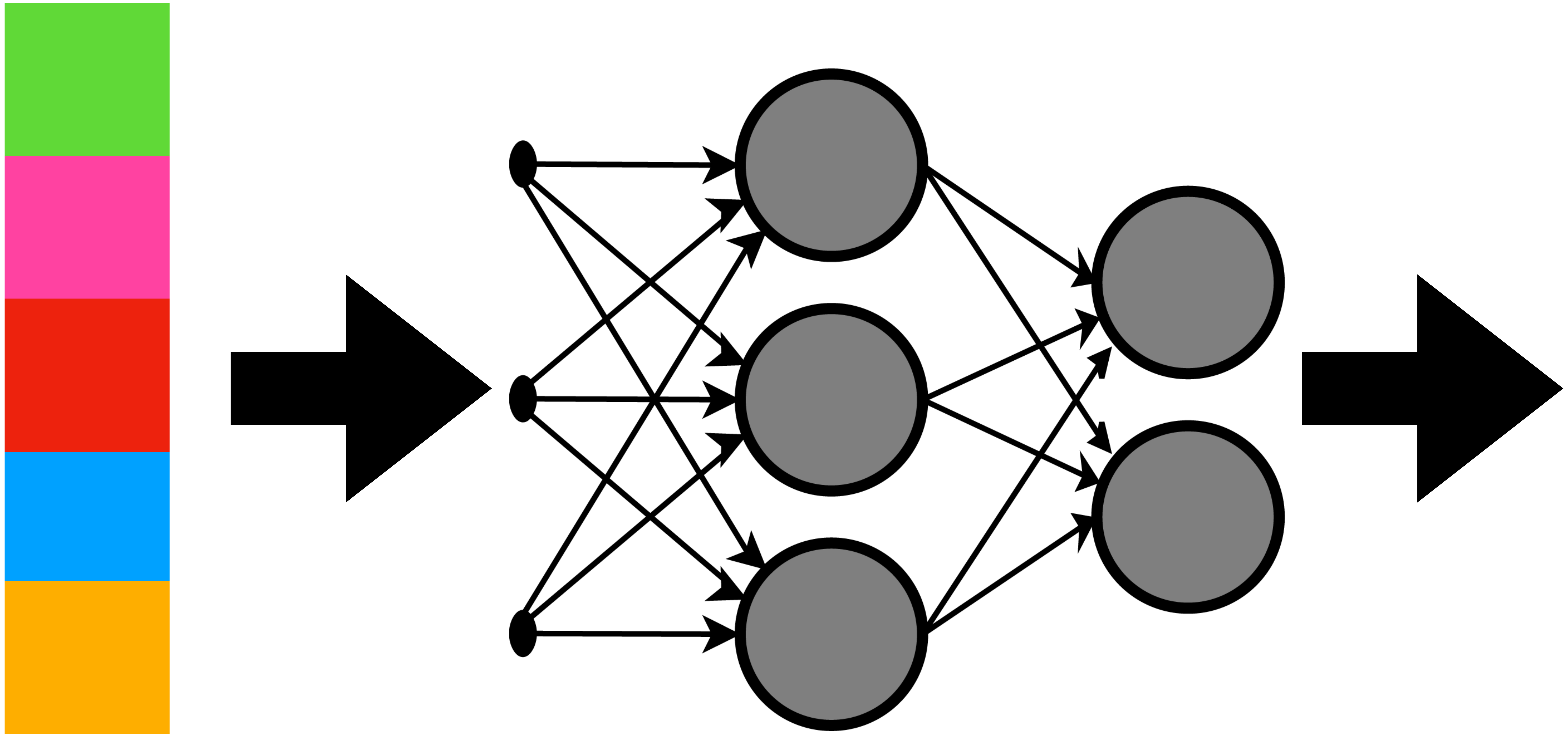
Decode interpolated features using MLP



Interpolated feature

MLP

Decode interpolated features using MLP

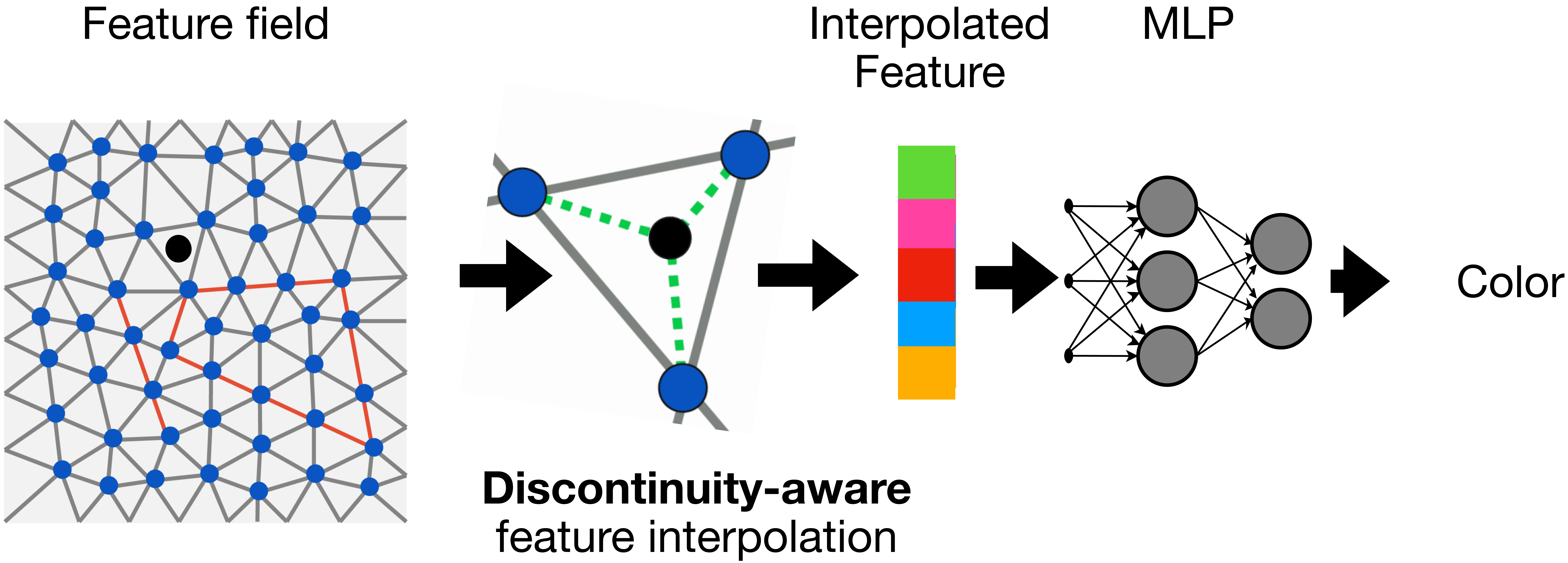


Interpolated feature

MLP

Color

Recap



Feature field

Discontinuity-aware feature interpolation

Interpolated Feature

MLP

Color

Performance

60 FPS inference @1080p

- 60-120 FPS inference on our examples
- Training is typically < 2 mins

All numbers are reported on an RTX 3090Ti

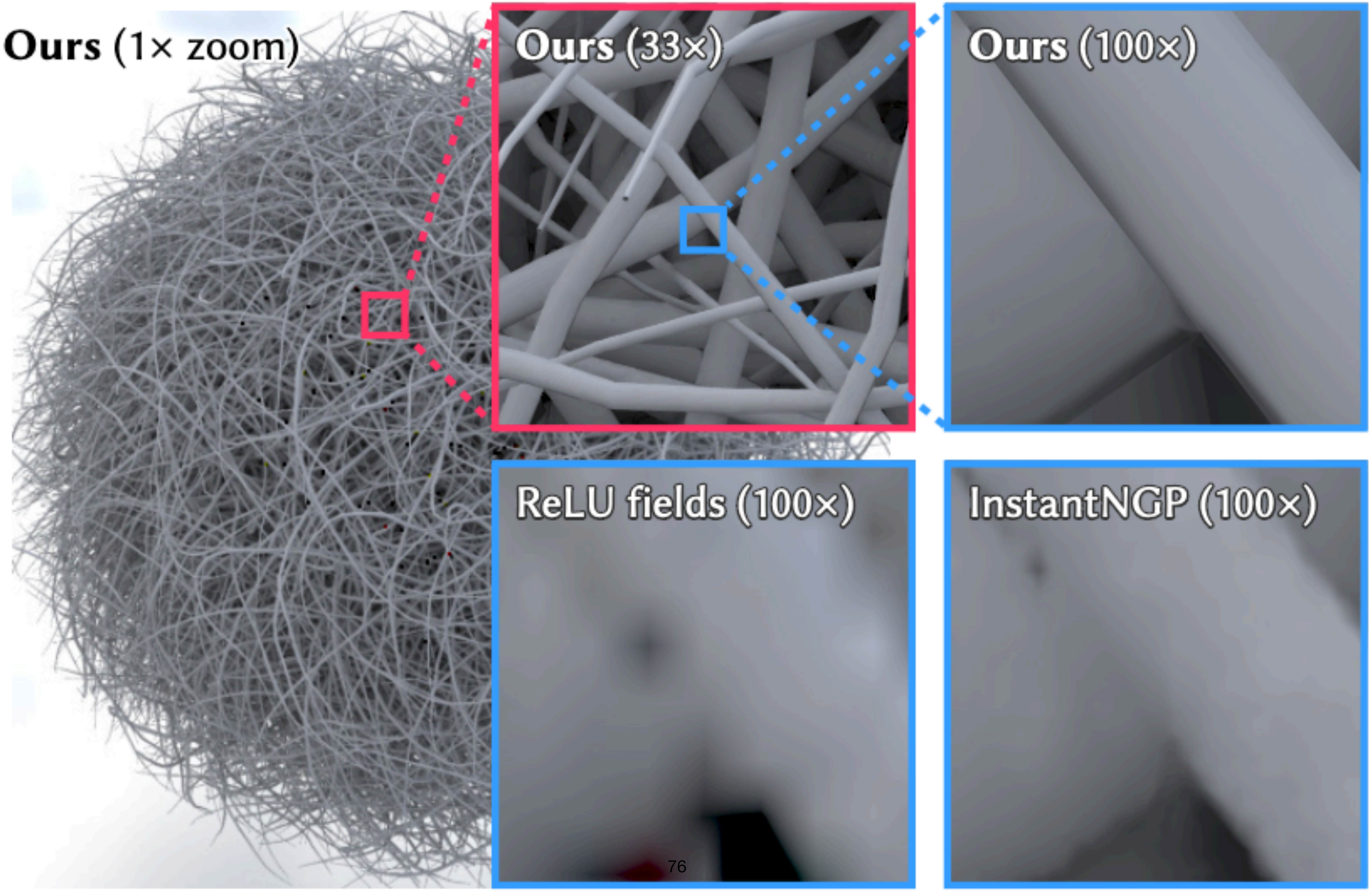
Results

Application: path-traced images

Application: path-traced image

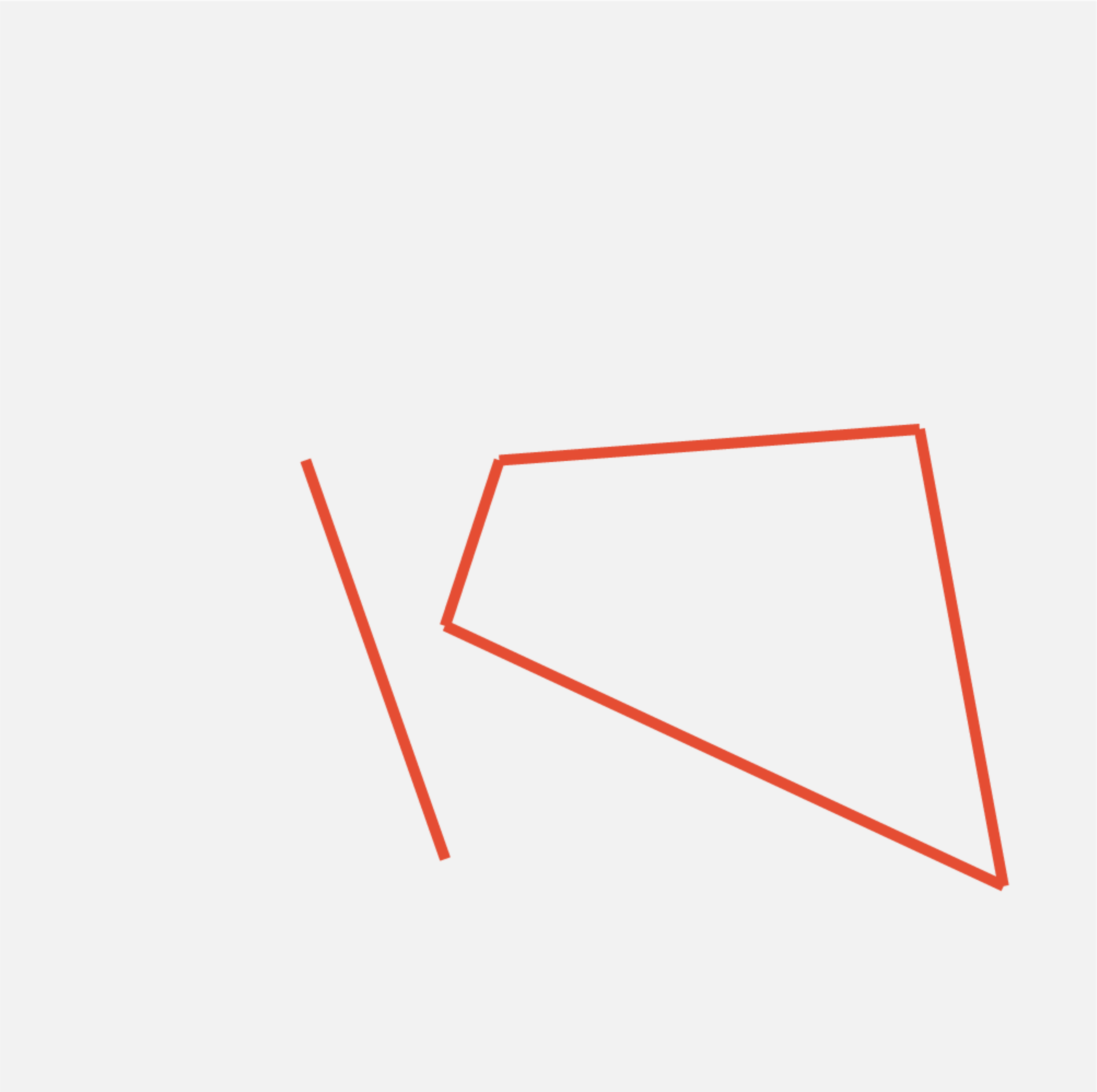
Reference
Ours
InstantNGP
ReLU Fields
Zoom: 1.739x

Application: path-traced image

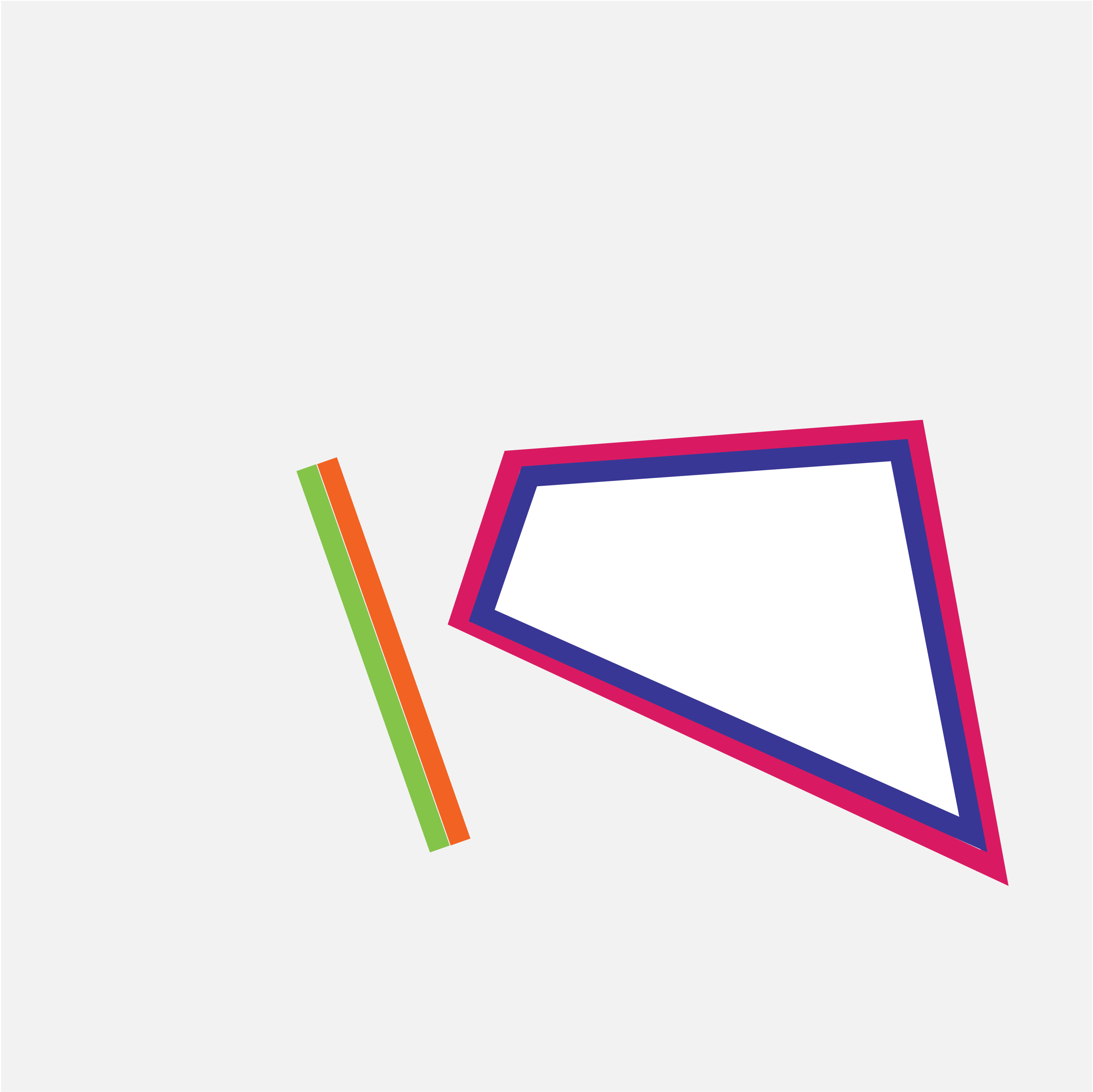


Application: diffusion curve images

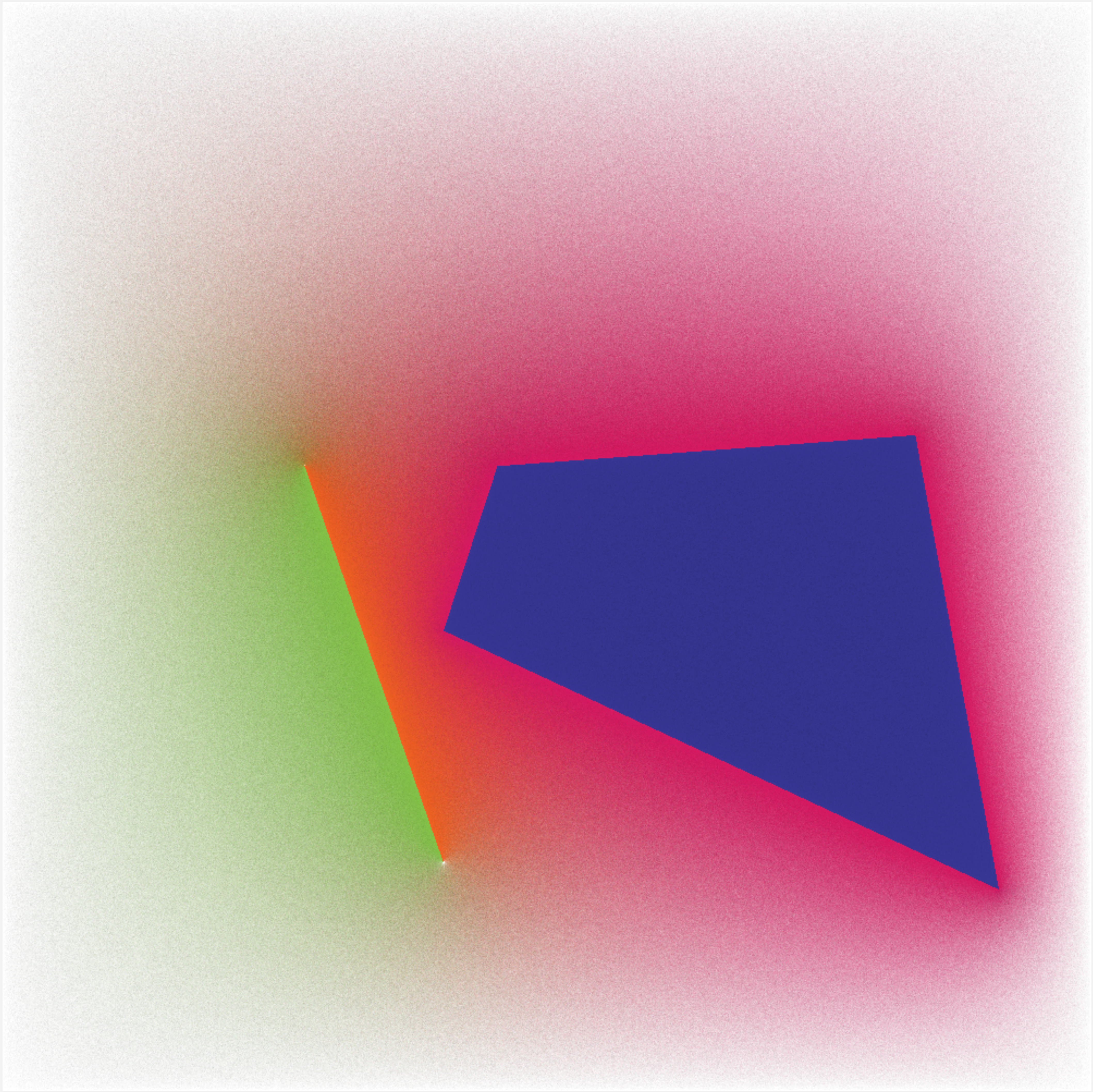
We start with some curves



Colors on both sides of curves



Diffuse colors from curves



Diffusion curve image

Monte Carlo estimate

Sawhney 20: Monte Carlo Geometry Processing

Monte Carlo data

Sawhney 20: Monte Carlo Geometry Processing

Ours

Monte Carlo data

Sawhney 20: Monte Carlo Geometry Processing

Ours

InstantNGP

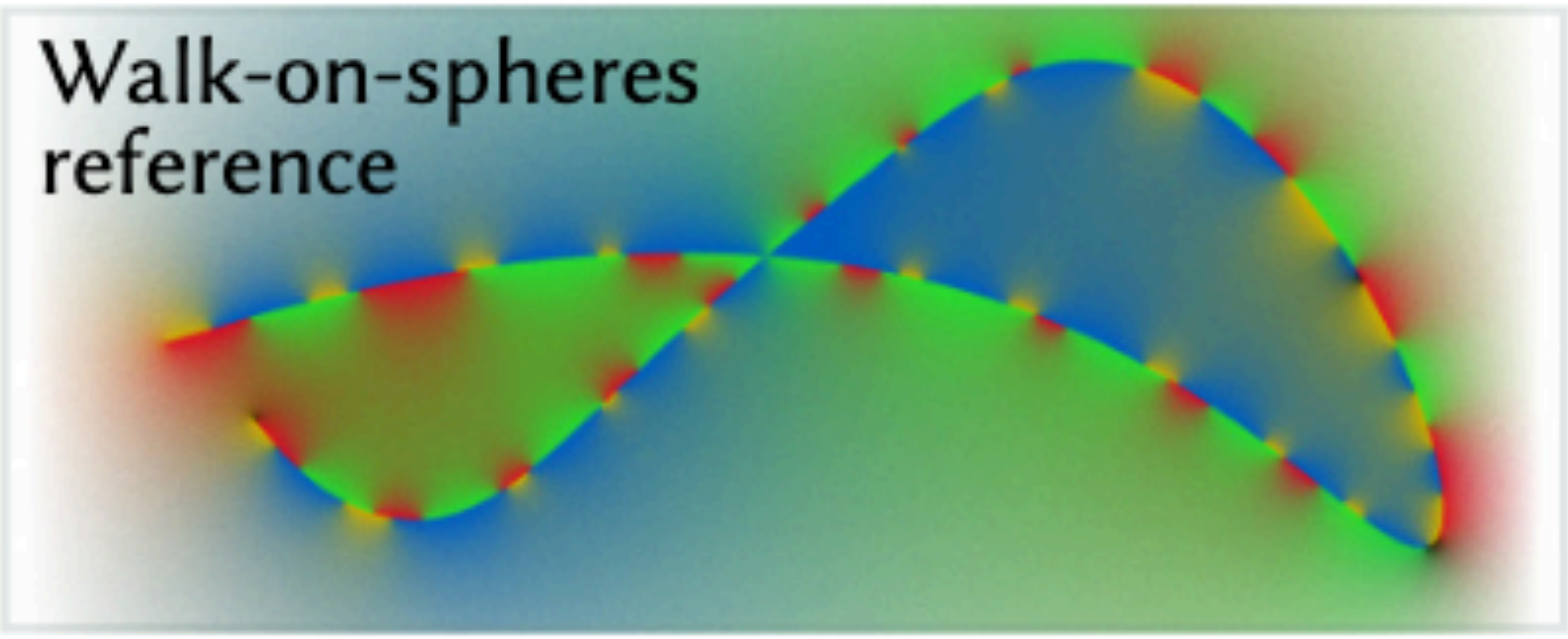
Muller 22: Instant Neural Graphics Primitives with a Multiresolution Hash Encoding

Our result: curved discontinuities

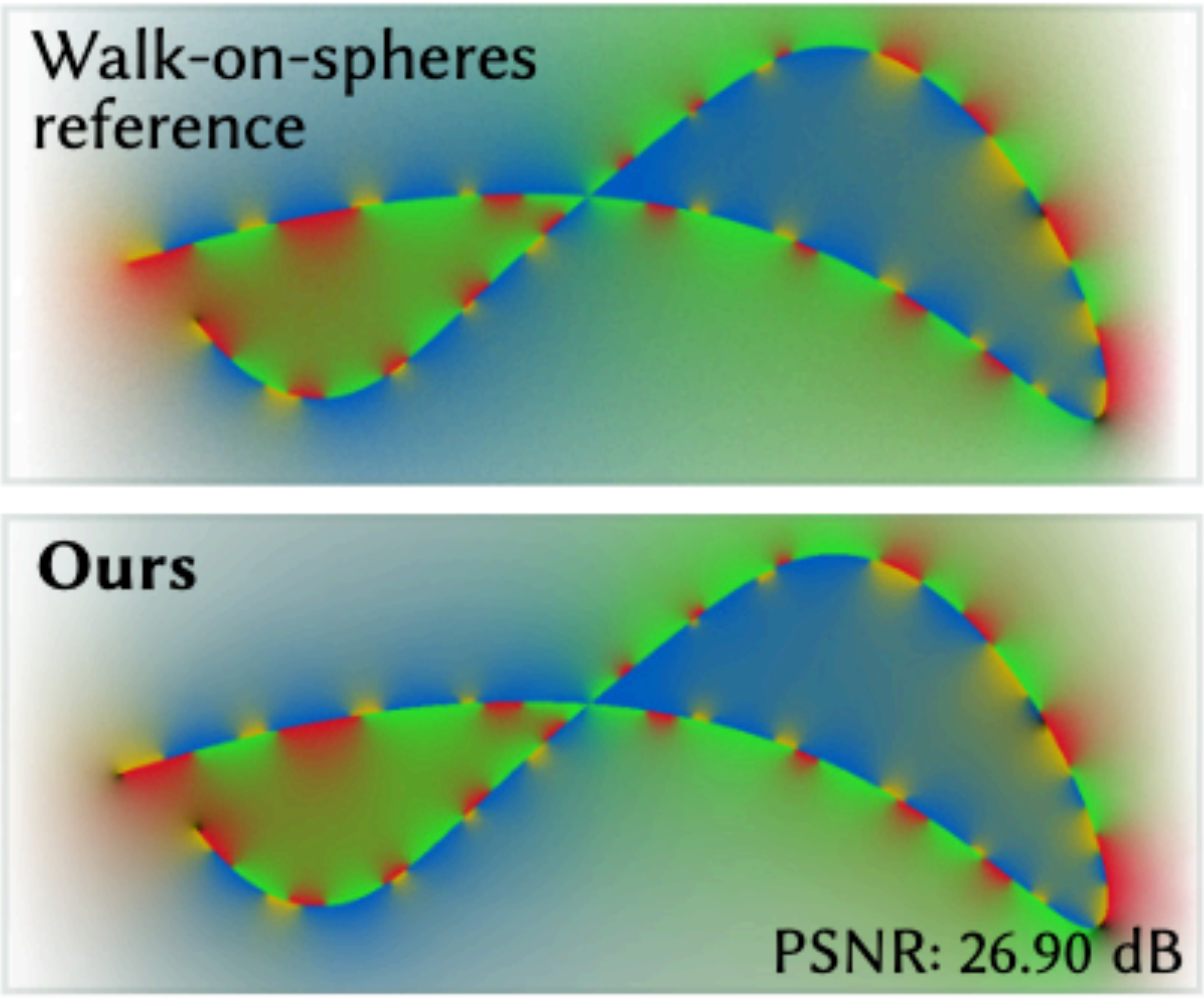
Our result: open edges

Application: physics-informed diffusion curve

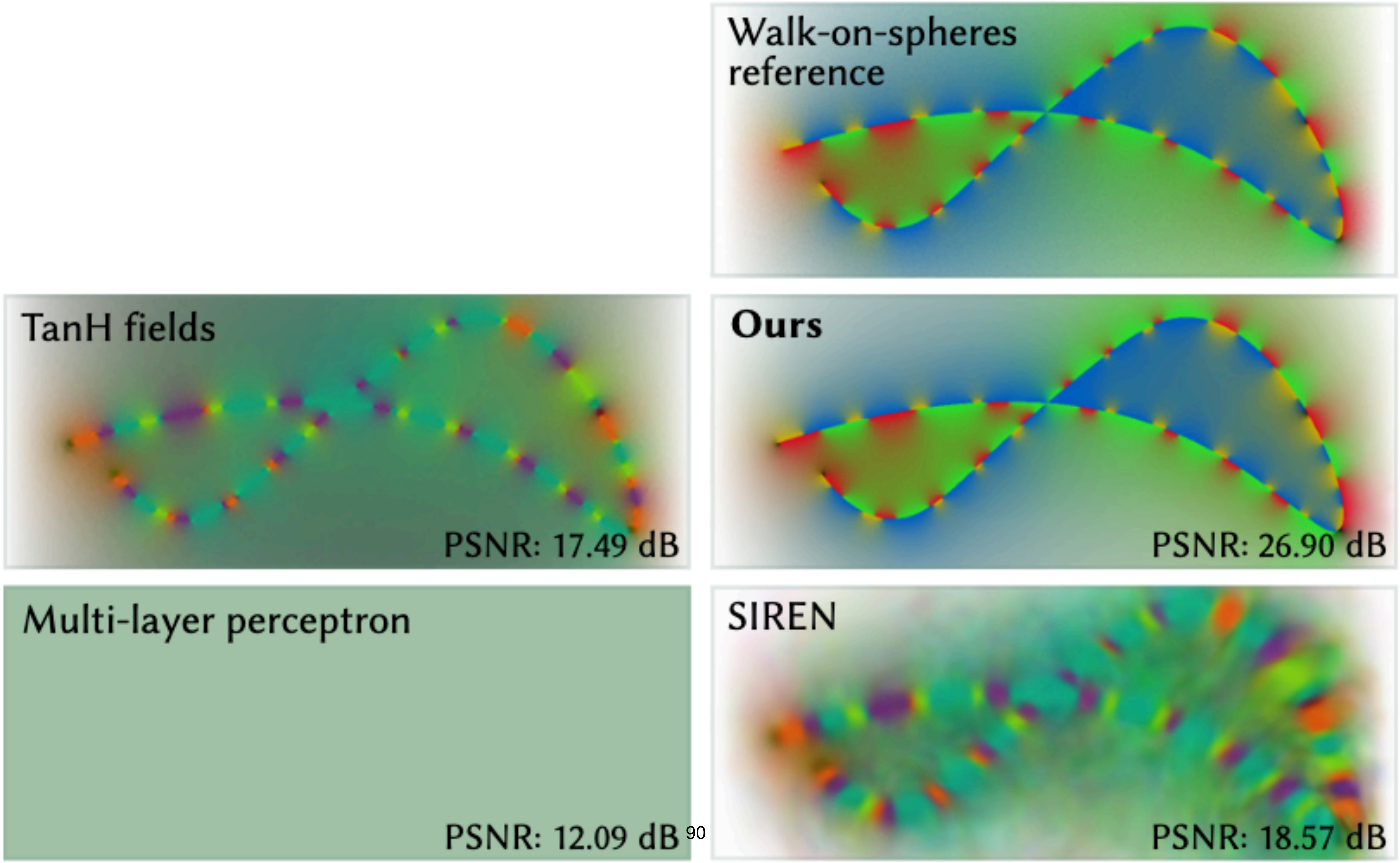
Application: physics informed diffusion curves



Application: physics informed diffusion curves

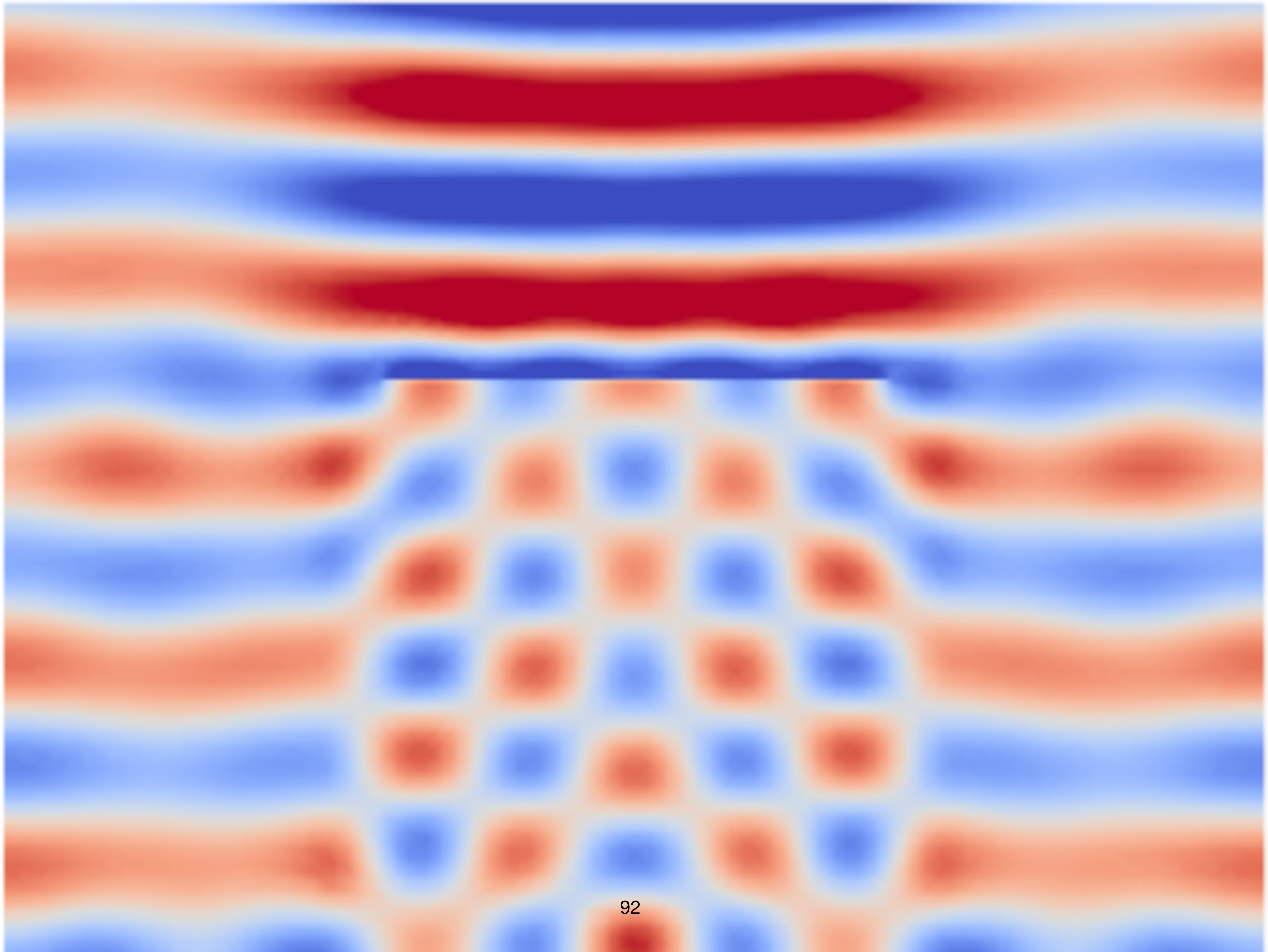


Application: physics informed diffusion curves



Application: store FEM solutions

Application: store solution to Helmholtz equation



Application: store solution to wave equation

Limitations

- We require discontinuity locations
- Different data structure needed for high frequency continuous variation

> Converting an image to pixels requires choosing a resolution and throwing away information beyond that resolution… When you really think about it, representing an image as pixels is really a bad compression technique… we need better image atoms…

Jim Blinn's Corner Notation Notation Notation

yashbelhe.github.io

Code available!