Nerf Architecture

- a forward pass involves 2 rendering pass — coarse
  fine
- datastructure

  - self._renderer = {

    coarse:

    ImplicitRenderer (NeRFRaySampler,
    raymarcher)

    fine : ImplicitRenderer(ProbablisticRaySampler,
    raymarcher)

    }

  - self._implicit_function = {

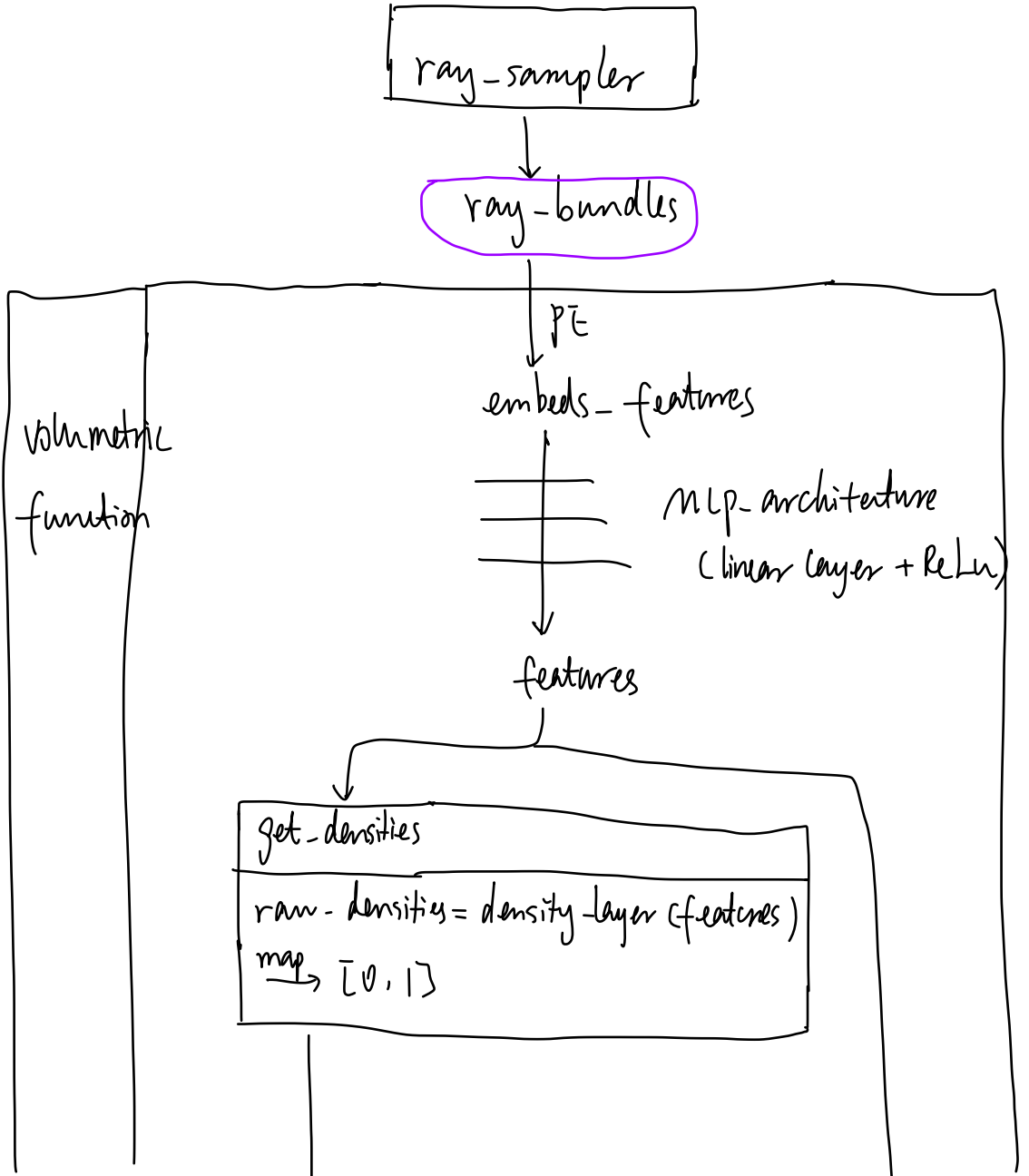
    coarse : NeuralRadiameField

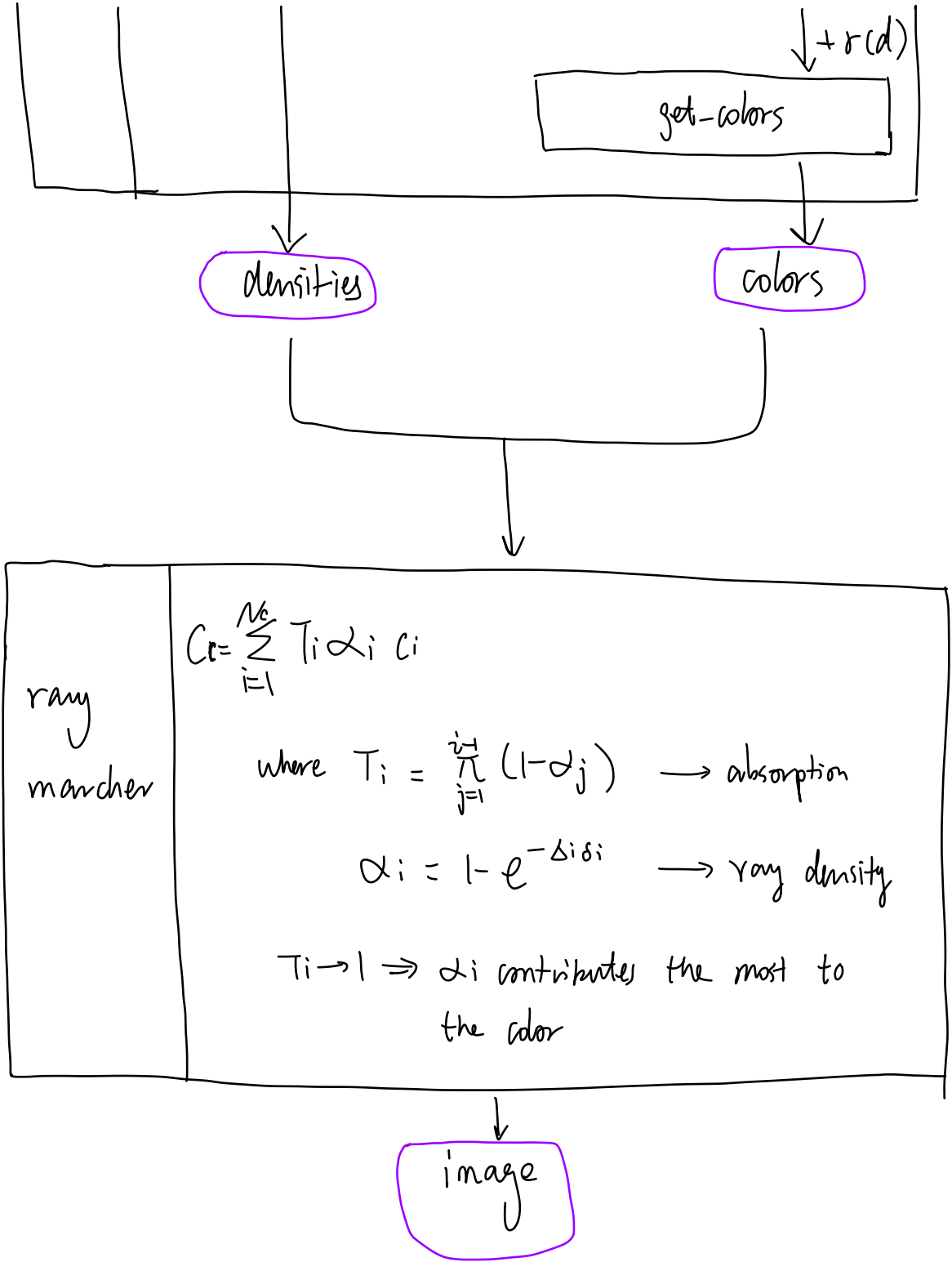
    fine : ...

    }

- forward

1. Coarse rendering pass




+r(d)
get-colors
densities
colors
ray
marcher
$C_r = \sum_{i=1}^{N_c} T_i \alpha_i c_i$
where $T_i = \prod_{j=1}^{i-1} (1-\alpha_j)$ → absorption
$\alpha_i = 1 - e^{-\Delta_i \sigma_i}$ → ray density
$T_i \to 1 \Rightarrow \alpha_i$ contributes the most to the color
image

2. fine rendering pass

almost the same, except:

- take coarse-weights
  to generate new ray-bundles
  via inverse transformation sampling
  $w_i = T_i \alpha_i$
- $C_f = \sum_{i=1}^{N_f+N_c} w_i c_i$

→ MSE:

$$\text{loss} = \sum_{r \in R} \| \hat{C}_c(r) - C(r) \|_2^2 + \| \hat{C}_f(r) - C(r) \|_2^2$$

differentiable. ⇒ gd, backwardprop

optimizer: Adam