```
@genjax.gen
def model():
  v = sample(flip(0.8), "v")
  (r, s) = (2.0, 1.0) if v else (3.0, 0.5)
  x = sample(normal(r, s), "x")
  observe(normal(x * x, 0.1), 4.0)
  return (v, x)

def density_model(tr):
  density = flip_density(tr["v"], 0.8)
  v = tr["v"]
  (r, s) = (2.0, 1.0) if v else (3.0, 0.5)
  density += normal_density(tr["x"], r, s)
  x = tr["x"]
  density += normal_density(4.0, x * x, 0.1)
  return density

@genjax.gen
def guide(p, mu_1, mu_2):
  v = sample(flip_reinforce(p), "v")
  (r, s) = (mu_1, 1.0) if v else (mu_2, 1.0)
  x = sample(normal_reparam(r, s), "x")
  return (v, x)

def sim_guide(p, mu_1, mu_2):
  v = flip_reinforce_adev(p)
  (r, s) = (mu_1, 1.0) if v else (mu_2, 1.0)
  x = normal_reparam_adev(r, s)
  tr = {"v": v, "x": x}
  density = (flip_density(v, p) *
             normal_density(x, r, s))
  return tr, density

@genjax.adev.E
def elbo(p, mu_1, mu_2):
  (tr, guide_p) = sim(guide)(p, mu_1, mu_2)
  model_p = density(model)(tr)
  return log(model_p / guide_p)
```