Algorithm 1 Proposed LbD framework

Inputs: $\mathbf{\Sigma}_c, d, dt$ ▷ Camera and robot properties
Select values for: k_a, k_p, k_o
for each demonstration i **do**
  $t := 0$ ▷ Initialization
  while $t_r < T_r$ **do**
    Get $\mathbf{p}_c, \mathbf{R}_c, \mathbf{n}, \mathbf{J}(\mathbf{q})$
    Get $\hat{\mathbf{p}}_{cf}$ from camera
    $t := t + dt$
    if $i = 1$ **then** ▷ 1st iteration
      $t_r := t$
      $\mathbf{v}_1 := \mathbf{0}_6$ ▷ No prev. demos
      Calculate $\mathbf{v}_2$ from (17)
      $\overline{\mathbf{\Sigma}}_m(t_r) := \mathbf{\Sigma}_i$ ▷ Initialization
    else
      Find the corresponding t_r of the 1st demo.
      $\mathbf{n}_d$:= major eigenvector of $\overline{\mathbf{\Sigma}}_m(t_r)$
      Calculate current $\mathbf{\Sigma}_i$ from (2)
      $\overline{\mathbf{\Sigma}}_m(t_r) := \left(\overline{\mathbf{\Sigma}}_m^{-1}(t_r) + \mathbf{\Sigma}_i^{-1}\right)^{-1}$ ▷ Eq. (13)
      Calculate $\mathbf{v}_1, \mathbf{v}_2$ from (16), (17)
    end if
    Calculate $\dot{\mathbf{q}}_c$ from (15)
    Command $\dot{\mathbf{q}}_c$ to the robot
    Store $\hat{\mathbf{p}}_{cf}$ as a sequence of t_r
    Store $\mathbf{p}_c, \mathbf{R}_c$ as a sequence of t_r
    Store $\overline{\mathbf{\Sigma}}_m$ as a sequence of t_r
    if $i = 1$ AND user commands break **then**
      $T_r := t_r$
      Break
    end if
  end while
  if Target uncertainty reached **then**
    Break
  end if
end for
Define $\mathbf{W}$ from (8)
Calculate $\mathbf{A}, \mathbf{B}, \mathbf{P}_c, \hat{\mathbf{P}}_{cf}$ from (9), (5), (6)
Calculate $\hat{\mathbf{p}}_f$ sequence from (4) ▷ Estimator
Train an encoding mechanism (e.g. a DMP), based on the $\hat{\mathbf{p}}_f$ sequence of t_r