Supplementary Material: Context-guided Continual Reinforcement Learning for Landmark Detection with Incomplete Data

Kaiwen Wan[1], Boming Wang[1], Fuping Wu[2], Haiyu Gong[1], and Xiahai Zhuang[1] ✉

[1] School of Data Science, Fudan University, Shanghai, 200433, China
[2] Nuffield Department of Population Health, University of Oxford, Oxford, UK
{16210180100,zxh}@fudan.edu.cn, {bmwang21,hygong23}@m.fudan.edu.cn, Fuping.Wu@ndph.ox.ac.uk

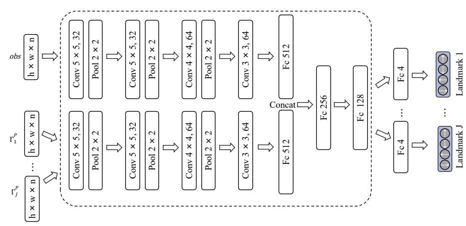

Fig. 1. Illustration of 2D CgMtQ network architecture. The outputted vector represents the utility of the action taken by the agent. The magnitude of this utility determines the Q value. $h(w)$ represents width (height) of cropped patches and n represents the time lag term. J represents the number of targets.

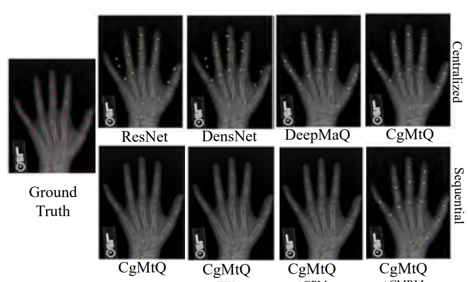

Fig. 2. The visualization of predictions with different methods and settings on handsets. The dots (●) represent Ground Truth (GD) landmarks. The dots (●) represent predictions of landmarks which are close to GD while the crosses (×) represent abnormal predictions. Noted that every image contains predictions for all 17 landmarks, but some abnormal predictions may overlapped.

Algorithm 1: Training stage of CgCRL

Data: Sequential Training Set $\mathcal{T}_k = \{(I_{k,i},\ Y_{k,i})|_{i\in\{1,\cdots,N_k\}}\}$, number of subtasks K, maximum number of episodes N, budget T, greedy parameter ϵ.
Result: Optimal network parameters ω

Initialize CgMtQ with random weights ω , the lag weight $\omega^- = \omega$, replay memory $\mathbf{D}$, prompt library $\mathit{\Gamma}$;
When progressing to k^{th} subtask, update $\mathit{\Gamma}$ with $\mathcal{T}_k$. If k $>$ 1, augment $\mathcal{T}_k$ with $\mathit{\Gamma}$ using CMRM;
for *episode = 1, N* **do**
  Select a random sample pair from $\mathcal{T}_k$: a random image $I_{k,i}$ with j-th target landmark y_j; a prompt $\mathit{\Gamma}_j^p$ from $\mathit{\Gamma}_j$;
  Initialize the starting position $\hat{y}^{[0]}$ randomly, and the state S accordingly;
  for t = *1, T* **do**
    If agent is not terminated, performs $A^{[t]}$ and move to $\hat{y}^{[t+1]}$ according to CgMtQ with probability $1-\epsilon$, otherwise selects a random action;
    Get $\hat{y}^{[t+1]}$, $R^{[t]}$, $S^{[t+1]}$ and store transition $(S^{[t]}, A^{[t]}, R^{[t]}, S^{[t+1]})$ in $\mathbf{D}$;
    Sample a random batch of transitions from $\mathbf{D}$, perform a gradient descent step with Eq. (1) for CgMtQ parameters ω;
    Terminate agent if it finds the target, oscillates or reaches maximum steps, break when all agents are terminated;
  end
  For every c_1 steps, reset $\omega^- = \omega$ and update ϵ;
end

Algorithm 2: Test stage of CgMtQ

Data: Test image set $\{I_i|_{i=1,\cdots,N'}\}$, number of test images N', index of all target landmarks $\mathcal{J}$, prompt library $\mathit{\Gamma}$, budget of CgMtQ T, maximum number of iterations M', number of prompts for j-th target N_j.
Result: Prediction for j-th landmark $\{y_{ij}^{pred}|_{i=1,\cdots,N'}\}$

Select image I_i from $\{I_i|_{i=1,\cdots,N'}\}$;
for p = *1,* N_j **do**
  Initialise $y^{[0]}$ with random landmarks, select $\mathit{\Gamma}_j^m$ from $\mathit{\Gamma}$;
  for t = *1,T* **do**
    If agent is not terminated, perform $A^{[t]}$ according to CgMtQ, move from $y^{[t]}$ to $y^{[t+1]}$;
    Terminate agent if it oscillates or reaches max steps, update $\hat{y}^p$ with $y^{[t+1]}$, crop $\mathit{\Gamma}_j^{\hat{p}}$;
  end
end
get y_{ij}^{pred} with Eq. (2) in the main manuscript.