This appendix provides a detailed elaboration on several aspects of our study. In Section A, we outline our implementation procedure and the hyper-parameter settings used. Section B provides pseudocodes illustrating the processes of planning with Hierarchical Diffuser. We examine the robustness of HD with various K values in Section C. The Out-of-distribution (OOD) visualizations and the corresponding experiment details are outlined in Section E. Section D explains details of the wall clock measurement. Finally, starting from Section H, we present our theoretical proofs.

A Implementation Details

In this section, we describe the details of implementation and hyperparameters we used during our experiments. For the Out-of-distribution experiment details, please check Section E.

- We build our Hierarchical Diffuser upon the officially released Diffuser code obtained from https://github.com/jannerm/diffuser. We list out the changes we made below.
- In our approach, the high-level and low-level planners are trained separately using segments randomly selected from the D4RL offline dataset.
- For the high-level planner's training, we choose segments equivalent in length to the planning horizon, H. Within these segments, states at every K steps are selected. In the dense action variants, the intermediary action sequences between these states are then flattened concatenated with the corresponding jumpy states along the feature dimension. This approach of trajectory representation is also employed in the training of the high-level reward predictor.
- The sequence modeling at the low-level is the same as Diffuser except that we are using a sequence length of $K + 1$.
- We set $K = 15$ for the long-horizon planning tasks, while for the Gym-MuJoCo, we use $K = 4$.
- Aligning closely with the settings used by Diffuser, we employ a planning horizon of $H = 32$ for the MuJoCo locomotion tasks. For the Maze2D tasks, we utilize varying planning horizons; $H = 120$ for the Maze2D UMaze task, $H = 255$ for the Medium Maze task, and $H = 390$ for the Large Maze task. For the AntMaze tasks, we set $H = 225$ for the UMaze, $H = 255$ for the Medium Maze, and $H = 450$ for the Large Maze.
- For the MuJoCo locomotion tasks, we select the guidance scales ω from a set of choices, $\{0.1, 0.01, 0.001, 0.0001\}$, during the planning phase.

B Planning with High-Level Diffuser

We highlight the high-level planning and low-level planning in Algorithm 1 and Algorithm 2, respectively. The complete process of planning with HD is detailed in Algorithm 3.

B.1 Planning with High-Level Diffuser

The high-level module, Sparse Diffuser (SD), models the subsampled states and actions, enabling it to operate independently. We present the pseudocode of guided planning with the Sparse Diffuser in Algorithm 1.

B.2 Planning with Low-Level Diffuser

Given subgoals sampled from the high-level diffuser, segments of low-level plans can be generated concurrently. We illatrate generating one such segment as example in Algorithm 2.

B.3 Hierarchical Planning

The comprehensive hierarchical planning involving both high-level and low-level planners is outlined in Algorithm 3. For the Maze2D tasks, we employed an open-loop approach, while for more challenging environments like AntMaze, Gym-MuJoCo, and Franka Kitchen, a closed-loop strategy was adopted.

Algorithm 1 High-Level Planning

function SAMPLEHIGHLEVELPLAN(Current State $\mathbf{s}$, Sparse Diffuser $\boldsymbol{\mu}_{\theta_{\text{SD}}}$, guidance function $\mathcal{J}_{\phi_{\text{SD}}}$, guidance scale ω, variance σ_m^2)
  initialize plan $\mathbf{x}_M^{\text{SD}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $m = M-1, \ldots, 1$ **do**
    $\tilde{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu}_{\theta_{\text{SD}}}(\mathbf{x}_{m+1}^{\text{SD}}) + \omega \sigma_m^2 \nabla_{\mathbf{x}_m^{\text{SD}}} \mathcal{J}_{\phi_{\text{SD}}}(\mathbf{x}_m^{\text{SD}})$
    $\mathbf{x}_{m-1}^{\text{SD}} \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \sigma_m^2 \mathbf{I})$
    Fix $\mathbf{g}_0$ in $\mathbf{x}_{m-1}^{\text{SD}}$ to current state $\mathbf{s}$
  end for
  return High-level plan $\mathbf{x}_0^{\text{SD}}$
end function

Algorithm 2 Low-Level Planning

function SAMPLELOWLEVELPLAN(Subgoals (g_i, g_{i+1}), low-level diffuser $\boldsymbol{\mu}_\theta$, low-level guidance function $\mathcal{J}_\phi$, guidance scale ω, variance σ_m^2)
  Initialize all low-level plan $\mathbf{x}_M^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  for $m = M-1, \ldots, 1$ **do**
    $\tilde{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu}_\theta(\mathbf{x}_{m+1}^i) + \omega \sigma_m^2 \nabla_{\mathbf{x}_m^i} \mathcal{J}_\phi(\mathbf{x}_m^i)$
    $\mathbf{x}_{m-1}^i \sim \mathcal{N}(\tilde{\boldsymbol{\mu}}, \sigma_m^2 \mathbf{I})$
    Fix $\mathbf{s}_0$ in $\mathbf{x}_{m-1}^i$ to $\mathbf{g}_i$; Fix $\mathbf{s}_K$ in $\mathbf{x}_{m-1}^i$ to $\mathbf{g}_{i+1}$
  end for
  return low-level plan $\mathbf{x}_0^i$
end function

C ABLATION STUDY ON JUMPY STEPS K

In this section, we report the detailed findings from an ablation study concerning the impact of the parameter K in Hierarchical Diffuser. The results, which are detailed in Tables 7 and 8, correspond to Maze2D tasks and MuJoCo locomotion tasks, respectively. As we increased K, an initial enhancement in performance was observed. However, a subsequent performance decline was noted with larger K values. This trend aligns with our initial hypothesis that a larger K introduces more skipped steps at the high-level planning stage, potentially resulting in the omission of information necessary for effective trajectory modeling, consequently leading to performance degradation.

Table 7: Ablation on K - Maze2D. The model's performance increased with the value of K up until $K = 21$. We report the mean and standard error over 100 random seeds.

Environment		**K1** (Diffuser default)	**HD-K7**	**HD-K15** (default)	**HD-K21**
Maze2D	U-Maze	113.9 ± 3.1	127.0 ± 1.5	$\mathbf{128.4 \pm 3.6}$	124.0 ± 2.1
Maze2D	Medium	121.5 ± 2.7	132.5 ± 1.3	$\mathbf{135.6 \pm 3.0}$	130.3 ± 2.4
Maze2D	Large	123.0 ± 6.4	153.2 ± 3.0	$\mathbf{155.8 \pm 2.5}$	158.9 ± 2.0
Sing-task Average		119.5	137.6	**139.9**	137.7
Multi2D	U-Maze	128.9 ± 1.8	135.4 ± 1.1	$\mathbf{144.1 \pm 1.2}$	133.7 ± 1.3
Multi2D	Medium	127.2 ± 3.4	135.3 ± 1.6	$\mathbf{140.2 \pm 1.6}$	134.5 ± 1.4
Multi2D	Large	132.1 ± 5.8	160.2 ± 1.9	$\mathbf{165.5 \pm 0.6}$	159.3 ± 3.0
Multi-task Average		129.4	143.7	**149.9**	142.5

D WALL CLOCK COMPARISON DETAILS

We evaluated the wall clock time by averaging the time taken per complete plan during testing and, for the training phase, the time needed for 100 updates. All models were measured using a single NVIDIA RTX 8000 GPU to ensure consistency. We employ the released code and default settings for the Diffuser model. We select the Maze2D tasks and Hopper-Medium-Expert, a representative for

Algorithm 3 Hierarchical Planning

function SAMPLEHIERARCHICALPLAN(High-level diffuser $\boldsymbol{\mu}_{\theta_{\text{SD}}}$, low-level diffuser $\boldsymbol{\mu}_\theta$, high-level guidance function $\mathcal{J}_{\phi_{\text{SD}}}$, low-level guidance function $\mathcal{J}_\phi$, high-level guidance scale ω_{SD}, low-level guidance scale ω, high-level variance $\sigma^2_{\text{SD},m}$, low-level variance σ^2_m)
  Observe state $\mathbf{s}$;
  if do open-loop **then**
    Sample high-level plan $\mathbf{x}^{\text{SD}}$ = SAMPLEHIGHLEVELPLAN($\mathbf{s}$, $\boldsymbol{\mu}_{\theta_{\text{SD}}}$, $\mathcal{J}_{\phi_{\text{SD}}}$, ω_{SD}, $\sigma^2_{SD,m}$)
    for $i = 0, \ldots, H-1$ **parallel do**
      Sample low-level plan $\mathbf{x}^{(i)}$ = SAMPLELOWLEVELPLAN((g_i, g_{i+1}), $\boldsymbol{\mu}_\theta$, $\mathcal{J}_\phi$, ω, σ^2_m)
    end for
    Form the full plan $\mathbf{x}$ with low-level plans $\mathbf{x}^{(i)}$ for $i = 0, H-1$
    for action $\mathbf{a}_t$ in $\mathbf{x}$ **do**
      Execute $\mathbf{a}_t$
    end for
  else
    while not done **do**
      Sample high-level plan $\mathbf{x}^{\text{SD}}$ = SAMPLEHIGHLEVELPLAN($\mathbf{s}$, $\boldsymbol{\mu}_{\theta_{\text{SD}}}$, $\mathcal{J}_\phi$, ω_{SD}, $\sigma^2_{\text{SD},m}$)
      // Sample only the first low-level segment
      Sample $\mathbf{x}^{(0)}$ = SAMPLELOWLEVELPLAN((g_0, g_1), $\boldsymbol{\mu}_\theta$, $\mathcal{J}_\phi$, ω, σ^2_m)
      Execute the first $\mathbf{a}_0$ of plan $\mathbf{x}^{(0)}$
      Observe state $\mathbf{s}$
    end while
  end if
end function

Table 8: Ablation on K - MuJoCo Locomotion. The model's performance increased with the value of K up until $K = 8$. We report the mean and standard error over 5 random seeds.

Dataset	**Environment**	**K1** (Diffuser default)	**HD-K4** (default)	**HD-K8**
Medium-Expert	HalfCheetah	88.9 ± 0.3	$\mathbf{92.5} \pm 0.3$	91.5 ± 0.3
Medium-Expert	Hopper	103.3 ± 1.3	$\mathbf{115.3} \pm 1.1$	113.0 ± 0.5
Medium-Expert	Walker2d	106.9 ± 0.2	107.1 ± 0.1	$\mathbf{107.6} \pm 0.3$
Medium	HalfCheetah	42.8 ± 0.3	$\mathbf{46.7} \pm 0.2$	45.9 ± 0.7
Medium	Hopper	74.3 ± 1.4	$\mathbf{99.3} \pm 0.3$	86.7 ± 7.4
Medium	Walker2d	79.6 ± 0.6	84.0 ± 0.6	$\mathbf{84.2} \pm 0.5$
Medium-Replay	HalfCheetah	37.7 ± 0.5	38.1 ± 0.7	$\mathbf{39.5} \pm 0.4$
Medium-Replay	Hopper	93.6 ± 0.4	$\mathbf{94.7} \pm 0.7$	91.3 ± 1.3
Medium-Replay	Walker2d	70.6 ± 1.6	$\mathbf{84.1} \pm 2.2$	76.4 ± 2.7
Average		77.5	**84.6**	81.8

the Gym-MuJoCo tasks, from the D4RL benchmark for our measurement purpose. On the Maze2D tasks, we set $K = 15$, and for the Gym-MuJoCo tasks, we set it to 4 as this is our default setting for RL tasks. The planning horizons of HD for each task, outlined in Table 9, are influenced by their need for divisibility by K, leading to slight deviations from the default values used by the Diffuser.

Table 9: Wall-clock time H value

Environment	Diffuser	Ours
Maze2d-Umaze	128	120
Maze2d-Medium	256	255
Maze2d-Large	384	390
Hopper-Medium-Expert	32	32

E Compositional Out-Of-Distribution (OOD) Experiment Details

While an increase in kernel size does indeed provide a performance boost for the Diffuser model, this enlargement inevitably augments the model's capacity, which potentially increases the risk of overfitting. Therefore, Diffuser models may underperform on tasks demanding both a large receptive field and strong generalization abilities. To illustrate this, inspired by Janner et al. (2022a), we designed a compositional out-of-distribution (OOD) Maze2D task, as depicted in Figure 4. During training, the agent is only exposed to offline trajectories navigating diagonally. However, during testing, the agent is required to traverse between novel start-goal pairs. We visualized the 32 plans generated by the models in Figure 4. As presented in the figure, only the Hierarchical Diffuser can generate reasonable plans approximating the optimal solution. In contrast, all Diffuser variants either create plans that lead the agent crossing a wall (i.e. Diffuser, Diffuser-KS13, and Diffuser-KS19) or produce plans that exceed the maximum step limit (i.e. Diffuser-13, Diffuser-KS19, and Diffuser-KS25).

To conduct this experiment, we generated a training dataset of 2 million transitions using the same Proportional-Derivative (PD) controller as used for generating the Maze2D tasks. Given that an optimal path typically requires around 230 steps to transition from the starting point to the end goal, we set the planning horizon H for the Diffuser variants at 248, while for our proposed method, we set it at 255, to ensure divisibility by $K = 15$. For the reinforcement learning task in the testing phase, the maximum steps allowed were set at 300. Throughout the training phase, we partitioned 10% of the training dataset as a validation set to mitigate the risk of overfitting. To quantitatively measure the discrepancy between the generated plans and the optimal solution, we used Cosine Similarity and Mean Squared Error (MSE). Specifically, we crafted 10 optimal paths using the same controller and sampled 100 plans from each model for each testing task. To ensure that the optimal path length aligned with the planning horizon of each model, we modified the threshold distance used to terminate the controller once the agent reached the goal state. Subsequently, we computed the discrepancy between each plan and each optimal path. The mean of these results was reported in Table 5.

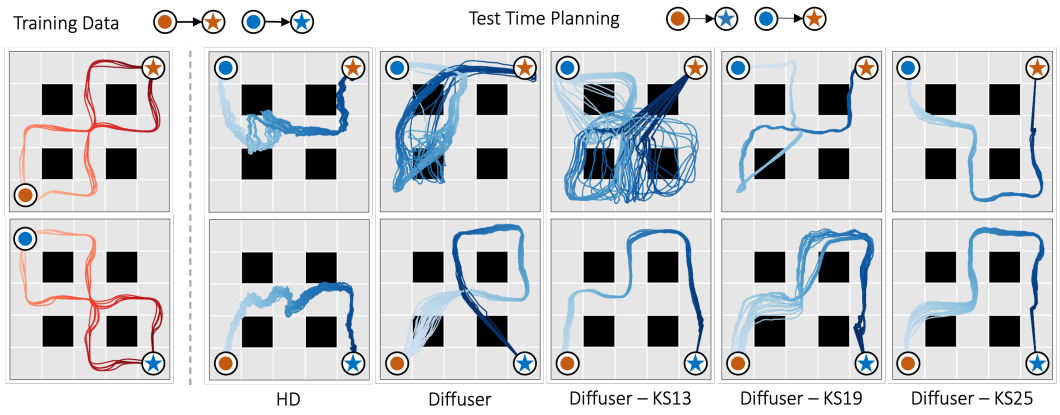

Figure 4: Large Kernel Size Hurts the OOD Generalization. Increasing kernel size generally improves the offline RL performance of Diffuser model. However, when a large receptive field and compositional out-of-distribution (OOD) generalization are both required, Diffuser models offer no simple solution. We demonstrate this with the sampled plans from both the standard Difuser and a Difuser with varied kernel sizes (KS). None of them can come up with an optimal plan by stiching training segments together. Conversely, our proposed Hierarchical Diffuser (HD) posseses both a large receptive field and the flexibility needed of compositional OOD tasks.

F Additional Ablation Studies

F.1 Transformer-based Diffusion

We compare our model (based on U-Net (CNN)) with Transformer-based diffusion and the results are in the table below. For this experiment, we use the hyperparameter setting in the Decision Transformer (Chen et al., 2021) as a starting point for our investigation. The results, as shown in the table, reveal that the HD-Transformer achieves similar performance to the HD-UNet in Maze2D tasks, though it is slightly less effective in the Gym-MuJoCo tasks. While the HD-Transformer shows promise, we would like to emphasize that our primary contribution is not the backbone architecture but the benefits of hierarchical structures. Nonetheless, we are grateful for the reviewer's insightful and constructive recommendation. This ablation study will make the paper better.

Table 10: Ablation Study on Backbone Architecture. HD-Transformer achieves comparable with HD-Unet on a wide rang of tasks.

Task	HD-UNet	HD-Transformer
Maze2d-Large	**128.4** $\pm$ 3.6	127.9 $\pm$ 3.2
Maze2d-Medium	135.6 $\pm$ 3.0	**136.1** $\pm$ 2.6
Maze2d-UMaze	**155.8** $\pm$ 2.5	154.1 $\pm$ 3.6
Maze2d Average	**139.9**	139.4
MedExp-HalfCheetah	**92.5** $\pm$ 0.3	88.4 $\pm$ 0.6
MedExp-Hopper	**115.3** $\pm$ 1.1	103.9 $\pm$ 5.9
MedExp-Walker2d	**107.1** $\pm$ 0.1	107.0 $\pm$ 0.3
Medium-HalfCheetah	**46.7** $\pm$ 0.2	45.3 $\pm$ 0.5
Medium-Hopper	**99.3** $\pm$ 0.3	94.0 $\pm$ 5.4
Medium-Walker2d	**84.0** $\pm$ 0.6	82.8 $\pm$ 1.7
MedRep-HalfCheetah	38.1 $\pm$ 0.7	**39.5** $\pm$ 0.2
MedRep-Hopper	**94.7** $\pm$ 0.7	91.4 $\pm$ 1.5
MedRep-Walker2d	**84.1** $\pm$ 2.2	81.2 $\pm$ 1.1
Gym Average	**84.6**	81.5

F.2 Sub-goal Selection Strategies

Hierarchical Diffuser (HD) select sub-goals with fixed time interval for simplicity. Here, we consider other choises:

- **Route Sampling** (Lai et al., 2020) (RS): In line with HDMI, we also consider choosing waypoint with fixed length interval as sub-goals. Specifically, denote the distance moved after action a_t as δ_t. Then, the route length can be computed as $S = \sum_{t=0}^{T1} \delta_t$. We pick the waypoints with fixed interval of S/k, where k is the number of sub-goals.
- **Value Sampling** (Correia & Alexandre, 2023) (VS): Also inspired by HDMI, we also test the value sampling method, where the most valuable states are chosen as sub-goals. Specifically, the distance weighted accumulated reward is used to value each states after state s_i: $W(s_j) = \sum_{k=i+1}^{j} \frac{r_k}{j-i}$.
- **Future Sampling** (Andrychowicz et al., 2017) (FS): Beyond RS and VS, we also explored a hindsight heuristic method, randomly selecting future states as sub-goals.

Notably, in RS and VS, certain states might never be chosen as sub-goals, unlike FS and the fixed time interval sampling (TS) used in HD, which offers equal probability for each state to be selected as a sub-goal.

Given the varying lengths of sub-tasks generated by these selection methods, integrating dense action at the high level was impractical. Hence, we focused our experiments on HD rather than HD-DA. At the low level, sub-trajectories were padded to a consistent length L. It's important to note that excluding dense action data at the high level may slightly hinder the learning of the value function, potentially leading to a marginal decrease in performance. The results, as presented in the table **??**,

demonstrate that our hierarchical framework is generally resilient across different sub-goal selection methods. While HD-VS and HD-RS exhibited somewhat lower performance, we hypothesize this may be due to uneven sampling of valuable states, which could impact the planning guidance function's effectiveness.

Table 11: Ablation Study on Sub-goal Selection. HD is generally resilient across different sub-goal selection methods.

Dataset	HD-DA	HD	HD-FS	HD-VS	HD-RS
MedExp-Halfcheetah	**92.5** $\pm$ 0.3	92.1 $\pm$ 0.5	87.6 $\pm$ 0.7	87.6 $\pm$ 0.6	88.4 $\pm$ 0.4
MedExp-Hopper	**115.3** $\pm$ 1.1	104.1 $\pm$ 8.2	106.5 $\pm$ 5.5	108.9 $\pm$ 4.8	106.4 $\pm$ 5.0
MedExp-Walker2d	107.1 $\pm$ 0.1	**107.4** $\pm$ 0.3	107.0 $\pm$ 0.1	**107.4** $\pm$ 0.2	**107.4** $\pm$ 0.3
Medium-Halfcheetah	**46.7** $\pm$ 0.2	45.2 $\pm$ 0.2	43.9 $\pm$ 0.4	43.2 $\pm$ 0.3	43.6 $\pm$ 0.9
Medium-Hopper	99.3 $\pm$ 0.3	99.2 $\pm$ 0.7	**100.9** $\pm$ 0.8	92.3 $\pm$ 4.2	95.8 $\pm$ 1.3
Medium-Walker2d	**84.0** $\pm$ 0.6	82.6 $\pm$ 0.8	83.1 $\pm$ 1.0	82.4 $\pm$ 0.9	82.9 $\pm$ 1.1
MedRep-Halfcheetah	38.1 $\pm$ 0.7	37.5 $\pm$ 1.7	**39.7** $\pm$ 0.3	38.1 $\pm$ 0.7	38.4 $\pm$ 0.8
MedRep-Hopper	**94.7** $\pm$ 0.7	93.4 $\pm$ 3.1	90.9 $\pm$ 1.7	91.3 $\pm$ 1.3	92.6 $\pm$ 1.2
MedRep-Walker2d	**84.1** $\pm$ 2.2	77.2 $\pm$ 3.3	80.9 $\pm$ 1.7	75.7 $\pm$ 2.10	76.4 $\pm$ 2.7
Average	**84.6**	82.1	82.3	80.8	81.3

G Theoretical Analysis

In this section, we show that the proposed method can improve the generalization capability when compared to the baseline. Our analysis also sheds light on the tradeoffs in K and the kernel size. Let $K \in \{1, \dots, T\}$, $\ell(x) = \tau \mathbb{E}_{m,\epsilon}[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_m}x + \sqrt{1-\bar{\alpha}_m}\epsilon, m)\|^2]]$, where $\tau > 0$ is an arbitrary normalization coefficient that can depend on K: e.g., $1/d$ where d is the dimensionality of ϵ. Given the training trajectory data $(\mathbf{x}_0^{(i)})_{i=1}^n$, the training loss is defined by $\hat{\mathcal{L}}(\theta) = \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{x}_0^{(i)})$ where $\mathbf{x}_m^{(i)} = \sqrt{\bar{\alpha}_m}\mathbf{x}_0^{(i)} + \sqrt{1-\bar{\alpha}_m}\epsilon$, and $\mathbf{x}_0^{(1)}, \dots, \mathbf{x}_0^{(n)}$ are independent samples of trajectories. We have $\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)]$. Define $\hat{\theta}$ to be an output of the training process using $(\mathbf{x}_0^{(i)})_{i=1}^n$, and φ to be the (unknown) value function under the optimal policy. Let Θ be the set of θ such that $\hat{\theta} \in \Theta$ and Θ is independent of $(\mathbf{x}_0^{(i)})_{i=1}^n$. Denote the projection of the parameter space Θ onto the loss function by $\mathcal{H} = \{x \mapsto \tau \mathbb{E}_{m,\epsilon}[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_m}x + \sqrt{1-\bar{\alpha}_m}\epsilon, m)\|^2] : \theta \in \Theta\}$, the conditional Rademacher complexity by $\mathcal{R}_t(\mathcal{H}) = \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n, \xi}[\sup_{h \in \mathcal{H}} \frac{1}{n_t}\sum_{i=1}^{n_t} \xi_i h(\mathbf{x}_0^{(i)}) \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t]$, where $\mathcal{C}_t = \{\mathbf{x}_0 \in \mathcal{X} : t = \arg\max_{j \in [H]} \varphi(\mathbf{g}_j)$ where $[\mathbf{g}_1\ \mathbf{g}_2\ \cdots\ \mathbf{g}_H]$ is the first row of $\mathbf{x}_0\}$ and $n_t = \sum_{i=1}^n \mathbb{1}\{\mathbf{x}_0^{(i)} \in \mathcal{C}_t\}$. Define $\mathcal{T} = \{t \in [H] : n_t \geq 1\}$ and $C_0 = d\tau c((1/\sqrt{2}) + \sqrt{2})$ for some $c \geq 0$ such that $c \geq \mathbb{E}_{m,\epsilon}[((\epsilon - \epsilon_\theta(\mathbf{x}_m, m))_i)^2]]$ for $i = 1, \dots, d$, where d is the dimension of $\epsilon \in \mathbb{R}^d$. Here, both the loss values and C_0 scale linearly in d. Our theorem works for any $\tau > 0$, including $\tau = 1/d$, which normalizes the loss values and C_0 with respect to d. Thus, the conclusion of our theorem is invariant of the scale of the loss value.

Theorem 1. *For any $\delta > 0$, with probability at least $1 - \delta$,*

$$\mathcal{L}(\hat{\theta}) \leq \hat{\mathcal{L}}(\hat{\theta}) + C_0\sqrt{\left\lceil \frac{T}{K} \right\rceil \frac{\ln(\lceil \frac{T}{K} \rceil \frac{2}{\delta})}{n}} + \sum_{t \in \mathcal{T}} \frac{2n_t \mathcal{R}_t(\mathcal{H})}{n}. \tag{13}$$

The proof is presented in Appendix I. The baseline is recovered by setting $K = 1$. Thus, Theorem 1 demonstrates that the proposed method (i.e., the case of $K > 1$) can improve the generalization capability of the baseline (i.e., the case of $K = 1$). Moreover, while the upper bound on $\mathcal{L}(\hat{\theta}) - \hat{\mathcal{L}}(\hat{\theta})$ decreases as K increases, it is expected that we loose more details of states with a larger value of K. Therefore, there is a tradeoff in K: i.e., with a larger value of K, we expect a better generalization for the diffusion process but a more loss of state-action details to perform RL tasks. On the other hand, the conditional Rademacher complexity term $\mathcal{R}_t(\mathcal{H})$ in Theorem 1 tends to increase as the number of parameters increases. Thus, there is also a tradeoff in the kernel size: i.e., with a larger kernel size, we expect a worse generalization for the diffusion process but a better receptive field to perform RL tasks. We provide the additional analysis on $\mathcal{R}_t(\mathcal{H})$ in Appendix H.

H On the conditional Rademacher complexity

In this section, we state that the term $\sum_{t\in\mathcal{T}} \frac{2n_t\mathcal{R}_t(\mathcal{H})}{n}$ in Theorem 1 is also smaller for the proposed method with $K \geq 2$ when compared to the base model (i.e., with $K = 1$) under the following assumptions that typically hold in practice. We assume that we can express $\epsilon_\theta(\mathbf{x}_m, m) = Wg(V\mathbf{x}_m, m)$ for some functions g and some matrices W, V such that the parameters of g do not contain the entries of W and V, and that Θ contains θ with W and V such that $\|W\|_\infty \leq \zeta_W$ and $\|V\|_\infty < \zeta_V$ for some ζ_W and ζ_V. This assumption is satisfied in most neural networks used in practice as g is arbitrarily; e.g., we can set $g = \epsilon_\theta$, $W = I$ and $V = I$ to have any arbitrary function $\epsilon_\theta(\mathbf{x}_m, m) = Wg(V\mathbf{x}_m, m) = g(\mathbf{x}_m, m)$. We also assume that $\mathcal{R}_t(\mathcal{H})$ does not increase when we increase n_t. This is reasonable since $\mathcal{R}_t(\mathcal{H}) = O(\frac{1}{n_t})$ for many machine learning models, including neural networks. Under this setting, the following proposition states that the term $\sum_{t\in\mathcal{T}} \frac{2n_t\mathcal{R}_t(\mathcal{H})}{n}$ of with the proposed method is also smaller than that of the base model:

Proposition 1. *Let $q \geq 2$ and denote by $\bar{\mathcal{R}}_t(\bar{\mathcal{H}})$ and $\tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$ the conditional Rademacher complexities for $K = 1$ (base case) and $K \geq q$ (proposed method) respectively. Then, $\bar{\mathcal{R}}_t(\bar{\mathcal{H}}) \geq \tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$ for any $t \in \{1, \dots, T\}$ such that s_t is not skipped with $K = q$.*

The proof is presented in Appendix I.

I Proofs

I.1 Proof of Theorem 1

Proof. Let $K \in \{1, \dots, T\}$. Define $[H] = \{1, \dots, H\}$. Define

$$\ell(x) = \tau\mathbb{E}_{m,\epsilon}[\|\epsilon - \epsilon_{\hat{\theta}}(\sqrt{\bar{\alpha}_m}x + \sqrt{1-\bar{\alpha}_m}\epsilon, m)\|^2]]$$

Then, we have that $\hat{\mathcal{L}}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{x}_0^{(i)})$ and $\mathcal{L}(\hat{\theta}) = \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)]$. Here, $\ell(\mathbf{x}_0^{(1)}), \dots, \ell(\mathbf{x}_0^{(n)})$ are not independent since $\hat{\theta}$ is trained with the trajectories data $(\mathbf{x}_0^{(i)})_{i=1}^n$, which induces the dependence among $\ell(\mathbf{x}_0^{(1)}), \dots, \ell(\mathbf{x}_0^{(n)})$. To deal with this dependence, we recall that

$$\mathbf{x}_0 = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_H \\ \mathbf{a}_0 & \mathbf{a}_K & \cdots & \mathbf{a}_{HK} \\ \mathbf{a}_1 & \mathbf{a}_{K+1} & \cdots & \mathbf{a}_{HK+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{K-1} & \mathbf{a}_{2K-1} & \cdots & \mathbf{a}_{(H+1)K-1} \end{bmatrix} \in \mathcal{X} \subseteq \mathbb{R}^d,$$

where the baseline method is recovered by setting $K = 1$ (and hence $H = T/K = T$). To utilize this structure, we define $\mathcal{C}_k$ by

$$\mathcal{C}_k = \left\{ \mathbf{x}_0 = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_H \\ \mathbf{a}_0 & \mathbf{a}_K & \cdots & \mathbf{a}_{HK} \\ \mathbf{a}_1 & \mathbf{a}_{K+1} & \cdots & \mathbf{a}_{HK+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_{K-1} & \mathbf{a}_{2K-1} & \cdots & \mathbf{a}_{(H+1)K-1} \end{bmatrix} \in \mathcal{X} : k = \arg\max_{t\in[H]} \varphi(\mathbf{g}_t), \right\}.$$

We first write the expected error as the sum of the conditional expected error:

$$\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)] = \sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k] \Pr(\mathbf{x}_0 \in \mathcal{C}_k).$$

Similarly,

$$\frac{1}{n}\sum_{i=1}^n \ell(\mathbf{x}_0^{(i)}) = \frac{1}{n}\sum_{k\in I_\mathcal{K}} \sum_{i\in\mathcal{I}_k} \ell(\mathbf{x}_0^{(i)}) = \sum_{k\in I_\mathcal{K}} \frac{|\mathcal{I}_k|}{n} \frac{1}{|\mathcal{I}_k|} \sum_{i\in\mathcal{I}_k} \ell(\mathbf{x}_0^{(i)}),$$

where $\mathcal{I}_k = \{i \in [n] : \mathbf{x}_0^{(i)} \in \mathcal{C}_k\}$ and $I_{\mathcal{K}} = \{k \in [H] : |\mathcal{I}_k| \geq 1\}$. Using these, we decompose the difference into two terms:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)] - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_0^{(i)}) &= \sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k]\left(\Pr(\mathbf{x}_0 \in \mathcal{C}_k) - \frac{|\mathcal{I}_k|}{n}\right) \\ &\quad + \left(\sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k]\frac{|\mathcal{I}_k|}{n} - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_0^{(i)})\right). \\ &= \sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k]\left(\Pr(\mathbf{x}_0 \in \mathcal{C}_k) - \frac{|\mathcal{I}_k|}{n}\right) \\ &\quad + \frac{1}{n}\sum_{k\in I_{\mathcal{K}}}|\mathcal{I}_k|\left(\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k] - \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\ell(\mathbf{x}_0^{(i)})\right).\end{aligned} \tag{14}$$

By following the proof of Lemma 5 of (Kawaguchi et al., 2023) and invoking Lemma 1 of (Kawaguchi et al., 2022), we have that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\begin{aligned}&\sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k]\left(\Pr(\mathbf{x}_0 \in \mathcal{C}_k) - \frac{|\mathcal{I}_k|}{n}\right) \\ &\leq \left(\sum_k \mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k]\sqrt{\Pr(\mathbf{x}_0 \in \mathcal{C}_k)}\right)\sqrt{\frac{2\ln(H/\delta)}{n}} \\ &\leq C\left(\sum_k \sqrt{\Pr(\mathbf{x}_0 \in \mathcal{C}_k)}\right)\sqrt{\frac{2\ln(H/\delta)}{n}}.\end{aligned} \tag{15}$$

Here, note that for any (f, h, M) such that $M > 0$ and $B \geq 0$ for all X, we have that $\mathbb{P}(f(X) \geq M) \geq \mathbb{P}(f(X) > M) \geq \mathbb{P}(Bf(X) + h(X) > BM + h(X))$, where the probability is with respect to the randomness of X. Thus, by combining equation 14 and equation 15, we have that for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:

$$\begin{aligned}\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)] - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_0^{(i)}) &\leq \frac{1}{n}\sum_{k\in I_{\mathcal{K}}}|\mathcal{I}_k|\left(\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k] - \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\ell(\mathbf{x}_0^{(i)})\right) \\ &\quad + C\left(\sum_k \sqrt{\Pr(\mathbf{x}_0 \in \mathcal{C}_k)}\right)\sqrt{\frac{2\ln(H/\delta)}{n}}\end{aligned} \tag{16}$$

We now bound the first term in the right-hand side of equation equation 16. Define

$$\mathcal{H} = \{x \mapsto \tau\mathbb{E}_{m,\epsilon}[\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_m}x + \sqrt{1-\bar{\alpha}_m}\epsilon, m)\|^2] : \theta \in \Theta\},$$

and

$$\mathcal{R}_t(\mathcal{H}) = \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n}\mathbb{E}_\xi\left[\sup_{h\in\mathcal{H}}\frac{1}{|\mathcal{I}_t|}\sum_{i=1}^{|\mathcal{I}_t|}\xi_i h(\mathbf{x}_0^{(i)}) \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t\right].$$

with independent uniform random variables $\xi_1, \ldots, \xi_n$ taking values in $\{-1, 1\}$. We invoke Lemma 4 of (Pham et al., 2021) to obtain that for any $\delta > 0$, with probability at least $1 - \delta$,

$$\begin{aligned}&\frac{1}{n}\sum_{k\in I_{\mathcal{K}}}|\mathcal{I}_k|\left(\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)|\mathbf{x}_0 \in \mathcal{C}_k] - \frac{1}{|\mathcal{I}_k|}\sum_{i\in\mathcal{I}_k}\ell(\mathbf{x}_0^{(i)})\right) \\ &\leq \frac{1}{n}\sum_{k\in I_{\mathcal{K}}}|\mathcal{I}_k|\left(2\mathcal{R}_k(\mathcal{H}) + C\sqrt{\frac{\ln(H/\delta)}{2|\mathcal{I}_k|}}\right) \\ &= \sum_{k\in I_{\mathcal{K}}}\frac{2|\mathcal{I}_k|\mathcal{R}_k(\mathcal{H})}{n} + C\sqrt{\frac{\ln(H/\delta)}{2n}}\sum_{k\in I_{\mathcal{K}}}\sqrt{\frac{|\mathcal{I}_k|}{n}} \\ &\leq \sum_{k\in I_{\mathcal{K}}}\frac{2|\mathcal{I}_k|\mathcal{R}_k(\mathcal{H})}{n} + C\sqrt{\frac{H\ln(H/\delta)}{2n}},\end{aligned} \tag{17}$$

where the last line follows from the Cauchy–Schwarz inequality applied on the term $\sum_{k\in I_K}\sqrt{\frac{|\mathcal{I}_k|}{n}}$ as

$$\sum_{k\in I_K}\sqrt{\frac{|\mathcal{I}_k|}{n}} \le \sqrt{\sum_{k\in I_K}\frac{|\mathcal{I}_k|}{n}}\sqrt{\sum_{k\in I_K}1} = \sqrt{\sum_{k\in I_K}1} \le \sqrt{H}.$$

On the other hand, by using Jensen's inequality,

$$\frac{1}{H}\sum_{k=1}^{H}\sqrt{\Pr(\mathbf{x}_0\in\mathcal{C}_k)} \le \sqrt{\frac{1}{H}\sum_{k=1}^{H}\Pr(\mathbf{x}_0\in\mathcal{C}_k)} = \frac{1}{\sqrt{H}}$$

which implies that

$$\sum_{k=1}^{H}\sqrt{\Pr(\mathbf{x}_0\in\mathcal{C}_k)} \le \sqrt{H}. \tag{18}$$

By combining equations equation 16 and equation 17 with union bound along with equation 18, it holds that any $\delta>0$, with probability at least $1-\delta$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)] - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_0^{(i)}) \\
&\le \sum_{k\in I_K}\frac{2|\mathcal{I}_k|\mathcal{R}_k(\mathcal{H})}{n} + C\sqrt{\frac{H\ln(2H/\delta)}{2n}} + C\left(\sum_{k}\sqrt{\Pr(\mathbf{x}_0\in\mathcal{C}_k)}\right)\sqrt{\frac{2\ln(2H/\delta)}{n}} \\
&\le \sum_{k\in I_K}\frac{2|\mathcal{I}_k|\mathcal{R}_k(\mathcal{H})}{n} + C\left(\sqrt{2}^{-1}+\sqrt{2}\right)\sqrt{\frac{H\ln(2H/\delta)}{n}}
\end{aligned}$$

Since $H\le\lceil T/K\rceil$, this implies that any $\delta>0$, with probability at least $1-\delta$,

$$\mathbb{E}_{\mathbf{x}_0}[\ell(\mathbf{x}_0)] - \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{x}_0^{(i)}) \le C_0\sqrt{\left\lceil\frac{T}{K}\right\rceil\frac{\ln(\lceil\frac{T}{K}\rceil\frac{2}{\delta})}{n}} + \sum_{t\in I_K}\frac{2|\mathcal{I}_t|\mathcal{R}_t(\mathcal{H})}{n}.$$

where $C_0 = C\left(\sqrt{2}^{-1}+\sqrt{2}\right)$. This proves the first statement of this theorem. □

I.2 Proof of Proposition 1

Proof. For the second statement, let $K=1$ and we consider the effect of increasing K from one to an arbitrary value greater than one. Denote by $\mathcal{R}_t(\mathcal{H})$ and $\tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$ the conditional Rademacher complexities for $K=1$ (base case) and $K>1$ (after increasing K) respectively: i.e., we want to show that $\mathcal{R}_t(\mathcal{H})\ge\tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$. Given the increasing value of K, let $t\in\{1,\dots,T\}$ such that s_t is not skipped after increasing K. From the definition of $\mathcal{H}$,

$$\begin{aligned}
\mathcal{R}_t(\mathcal{H}) =& \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n}\mathbb{E}_\xi\left[\sup_{h\in\mathcal{H}}\frac{1}{|\mathcal{I}_t|}\sum_{i=1}^{|\mathcal{I}_t|}\xi_i h(\mathbf{x}_0^{(i)}) \mid \mathbf{x}_0^{(i)}\in\mathcal{C}_t\right] \\
=& \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n}\mathbb{E}_\xi\left[\sup_{\theta\in\Theta}\frac{1}{|\mathcal{I}_t|}\sum_{i=1}^{|\mathcal{I}_t|}\xi_i\mathbb{E}_{m,\epsilon}[\|\epsilon-\epsilon_\theta(\varsigma(\mathbf{x}_0^{(i)}),m)\|^2] \mid \mathbf{x}_0^{(i)}\in\mathcal{C}_t\right].
\end{aligned} \tag{19}$$

where everything is defined for $K=1$ and $\varsigma(\mathbf{x}_0^{(i)}) = \sqrt{\bar{\alpha}_m}\mathbf{x}_0^{(i)} + \sqrt{1-\bar{\alpha}_m}\epsilon$. Here, we recall that $\epsilon_\theta(\mathbf{x}_m,m) = Wg(\mathbf{x}_m,m)$ for some function g and an output layer weight matrix W such that the parameters of g does not contain the entries of the output layer weight matrix W. This implies that $\epsilon_\theta(\varsigma(\mathbf{x}_0^{(i)}),m) = W\tilde{g}_m(V\mathbf{x}_0^{(i)})$ where $\tilde{g}_m(x) = g(\tilde{\varsigma}(x),m)$ where $\tilde{\varsigma}(x) = \sqrt{\bar{\alpha}_m}x + \sqrt{1-\bar{\alpha}_m}V\epsilon$,

and that we can decompose $\Theta = \mathcal{W} \times \mathcal{V} \times \tilde{\Theta}$ with which θ can be decomposed into $W \in \mathcal{W}$, $V \in \mathcal{V}$, and $\tilde{\theta} \in \tilde{\Theta}$. Using this,

$$\mathcal{R}_t(\mathcal{H}) = \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{\theta \in \Theta} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \mathbb{E}_{m,\epsilon}[\|\epsilon - W \tilde{g}_m(V \mathbf{x}_0^{(i)})\|^2] \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right] \tag{20}$$

$$= \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W} \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \sum_{j=1}^{d} \mathbb{E}_{m,\epsilon}[(\epsilon_j - W_j \tilde{g}_m(V \mathbf{x}_0^{(i)}))^2] \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right].$$

where W_j is the j-th row of W. Recall that when we increase K, some states are skipped and accordingly d decreases. Let d_0 be the d after K increased from one to some value greater than one: i.e., $d_0 \leq d$. Without loss of generality, let us arrange the order of the coordinates over $j = 1, 2 \ldots, d_0, d_0 + 1, \ldots, d$ so that $j = d_0 + 1, d_0 + 2, \ldots, d$ are removed after K increases.

Since Θ contains θ with W and V such that $\|W\|_\infty \leq \zeta_W$ and $\|V\|_\infty < \zeta_V$ for some ζ_W and ζ_V, the set $\mathcal{W}$ contains W such that $W_j = 0$ for $j = d_0 + 1, d_0 + 2, \ldots, d$. Define $\mathcal{W}_0$ such that $\mathcal{W} = \mathcal{W}_0 \times \tilde{\mathcal{W}}_0$ where $(W_j)_{j=1}^{d_0} \in \mathcal{W}_0$ and $(W_j)_{j=d_0+1}^{d} \in \tilde{\mathcal{W}}_0$. Notice that $\mathcal{W} = \{(W_j)_{j=1}^d : \|(W_j)_{j=1}^d\|_\infty \leq \zeta_W\}$ and $\mathcal{W}_0 = \{(W_j)_{j=1}^{d_0} : \|(W_j)_{j=1}^{d_0}\|_\infty \leq \zeta_W\}$. Since we take supremum over $W \in \mathcal{W}$, setting $W_j = 0$ for $j = d_0 + 1, d_0 + 2, \ldots, d$ attains a lower bound as

$$\mathcal{R}_t(\mathcal{H}) = \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W} \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \sum_{j=1}^{d} \mathbb{E}_{m,\epsilon}[(\epsilon_j - W_j \tilde{g}_m(V \mathbf{x}_0^{(i)}))^2] \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

$$\geq \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i A_i \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

$$= \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \sum_{j=1}^{d_0} \mathbb{E}_{m,\epsilon}[(\epsilon_j - W_j \tilde{g}_m(V \mathbf{x}_0^{(i)}))^2] \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

where $A_i = \sum_{j=1}^{d_0} \mathbb{E}_{m,\epsilon}[(\epsilon_j - W_j \tilde{g}_m(V \mathbf{x}_0^{(i)}))^2] + \sum_{j=d_0+1}^{d} \mathbb{E}_{m,\epsilon}[(\epsilon_j)^2]$ and the last line follows from the fact that

$$\mathbb{E}_\xi \sup_{(W,\tilde{\theta}) \in \mathcal{W} \times \tilde{\Theta}} \sum_{j=d_0+1}^{d} \xi_i \mathbb{E}_{m,\epsilon}[(\epsilon_j)^2] = \mathbb{E}_\xi \sum_{j=d_0+1}^{d} \xi_i \mathbb{E}_{m,\epsilon}[(\epsilon_j)^2] = \sum_{j=d_0+1}^{d} \mathbb{E}_\xi[\xi_i] \mathbb{E}_{m,\epsilon}[(\epsilon_j)^2] = 0.$$

Similarly, since Θ contains θ with W and V such that $\|W\|_\infty \leq \zeta_W$ and $\|V\|_\infty < \zeta_V$ for some ζ_W and ζ_V, the set $\mathcal{V}$ contains V such that $V_j = 0$ for $j = d_0 + 1, d_0 + 2, \ldots, d$, where V_j is the j-th row of V. Define $\mathcal{V}_0$ such that $\mathcal{V} = \mathcal{V}_0 \times \tilde{\mathcal{V}}_0$ where $(V_j)_{j=1}^{d_0} \in \mathcal{V}_0$ and $(V_j)_{j=d_0+1}^{d} \in \tilde{\mathcal{V}}_0$. Notice that $\mathcal{V} = \{(V_j)_{j=1}^d : \|(V_j)_{j=1}^d\|_\infty \leq \zeta_V\}$ and $\mathcal{V}_0 = \{(V_j)_{j=1}^{d_0} : \|(V_j)_{j=1}^{d_0}\|_\infty \leq \zeta_V\}$. Since we take supremum over $V \in \mathcal{V}$, setting $V_j = 0$ for $j = d_0 + 1, d_0 + 2, \ldots, d$ attains a lower bound as

$$\mathcal{R}_t(\mathcal{H}) \geq \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \sum_{j=1}^{d_0} \mathbb{E}_{m,\epsilon}[(\epsilon_j - W_j \tilde{g}_m(V \mathbf{x}_0^{(i)}))^2] \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

$$= \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V} \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i B_i(d) \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

$$\geq \mathbb{E}_{(\mathbf{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(W,V,\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V}_0 \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i B_i(d_0) \mid \mathbf{x}_0^{(i)} \in \mathcal{C}_t \right]$$

$$= \mathbb{E}_{(\tilde{x}_0^{(i)})_{i=1}^n} \mathbb{E}_\xi \left[\sup_{(\tilde{W},\tilde{V},\tilde{\theta}) \in \mathcal{W}_0 \times \mathcal{V}_0 \times \tilde{\Theta}} \frac{1}{|\mathcal{I}_t|} \sum_{i=1}^{|\mathcal{I}_t|} \xi_i \mathbb{E}_{m,\epsilon}[\|\tilde{\epsilon} - \tilde{W} \tilde{g}_m(\tilde{V} \tilde{x}_0^{(i)})\|^2] \mid \tilde{x}_0^{(i)} \in \tilde{\mathcal{C}}_t \right]$$

$$\geq \tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$$

where $B_i(d) = \sum_{j=1}^{d_0} \mathbb{E}_{m,\epsilon}\left[\left(\epsilon_j - W_j \tilde{g}_m\left(\sum_{k=1}^{d} V_k(\mathbf{x}_0^{(i)})_k\right)\right)^2\right]$, $\tilde{\epsilon} = (\epsilon_j)_{j=1}^{d_0}$, $\tilde{x}_0^{(i)} = ((\tilde{x}_0^{(i)})_j)_{j=1}^{d_0}$, $\tilde{\mathcal{C}}_t$ is the $\mathcal{C}_t$ for $\tilde{x}_0^{(i)}$ with skipping states, and $\tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$ is the conditional Rademacher complexity after increasing $K > 1$. The last line follows from the same steps of equation 19 and equation 20 applied for $\tilde{\mathcal{R}}_t(\tilde{\mathcal{H}})$ and the fact that $|\mathcal{I}_t|$ of $\mathcal{R}_t(\mathcal{H})$ is smaller than that of $\mathcal{R}_t(\tilde{\mathcal{H}})$ (due to the effect of removing the states), along with the assumption that $\mathcal{R}_t(\mathcal{H})$ does not increase when we increase n_t. This proves the second statement.

□