Appendix

A Building HRRT policy

We employed a hierarchical reinforcement learning framework to model the complex decision-making process in RRT. This framework involved three distinct agents. The modality agent was responsible for high-level decisions, choosing between IHD, CRRT, or no RRT. Once a modality was selected, lower-level agents were activated to determine the specific dosage. The IHD agent decided among four levels of ultrafiltration volume, while the CRRT agent recommended a continuous effluent rate. The primary outcome of our study was ICU mortality. All agents aimed to decrease mortality by maximizing future rewards, learning to make optimal decisions through state-action pairs. The reward function was directly influenced by patient mortality outcomes and SOFA scores. We also explored an alternative end-to-end learning approach using the Option-Critic architecture[2], which theoretically allows for simultaneous optimization of both the high-level modality policy and the low-level dosage policies within a unified framework. However, this approach proved to be unstable in practice. The difficulty of coordinating multiple hierarchical agents concurrently led to severe convergence issues, where the high-level policy struggled to stabilize while the low-level options were still exploring or under-optimized. This lack of coordination resulted in policy values that failed to exhibit a consistent or meaningful correlation with patient mortality, undermining the safety and interpretability required for clinical deployment. Consequently, we adopted the decoupled hierarchical approach, which ensures robust and independent optimization of each decision layer.

A primary challenge in developing a clinical agent from existing medical records is the static, offline nature of the data. Standard online algorithms like D3QN are unsuitable as they suffer from extrapolation error, learning to exploit out-of-distribution (OOD) actions that were rare in the dataset but appear overly-optimistic. This was observed in our initial experiments, where D3QN would collapse to recommending IHD, a rare action associated with a less severe cohort, mistaking this correlation for causation. To mitigate this, our framework employs specialized offline algorithms for each agent. For the high-level modality agent, we selected BCQ[6]. BCQ addresses extrapolation error by learning a policy π that is constrained to the data distribution. It uses an imitation model to generate actions $\{a_i\}$ similar to the behavior policy β, and the agent is restricted to selecting the action that maximizes the Q-value from this safe set:

$$\pi(s) = \arg \max_{a_i \sim G(s)} Q(s, a_i)$$

This prevents the agent from choosing OOD actions like IHD and ensures robust, safe decisions.

For the low-level IHD agent, which also suffered from data sparsity, we employed CQL[14]. CQL is a soft constraint method that adds a conservative penalty to the standard Bellman loss. This penalty pushes down the Q-values of OOD actions while pushing up the Q-values for actions present in the data:

$$\mathcal{L}(\theta) = \alpha \mathbb{E}_{s \sim \mathcal{D}, a \sim \pi(\cdot|s)}[Q(s, a)] - \mathbb{E}_{s,a \sim \mathcal{D}}[Q(s, a)] + \mathcal{L}_{\text{Bellman}}(\theta)$$

By tuning the weight α, we force the agent to be pessimistic about unknown actions, preventing policy collapse.

For the low-level CRRT agent, the dataset was abundant, but the action space was continuous. We therefore employed IQL[13], a state-of-the-art offline algorithm for continuous control. IQL avoids querying OOD actions by learning the Q-function implicitly. It first learns a state-value function V using an expectile regression loss:

$$\mathcal{L}_V(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s, a) - V_\psi(s))]$$

where $L_2^\tau(\cdot)$ is the expectile loss, which asymmetrically penalizes errors to estimate the upper-expectile of the Q-values. The Q-function is then updated by regressing towards the observed rewards and the learned value function:

$$\mathcal{L}_Q(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}[(r + \gamma V_\psi(s') - Q_\theta(s, a))^2]$$

Finally, the policy is extracted using a weighted advantage function $A(s, a) = Q_{\hat{\theta}}(s, a) - V_\psi(s)$, which implicitly constrains the policy to actions with high advantages. This approach provides robust stability for the continuous control task and completes our fully offline framework.

B Hyperparameter tuning

Hyperparameter tuning is a crucial aspect of refining algorithm performance. To achieve optimal performance, we utilized a grid search approach, systematically testing various combinations of hyperparameters such as learning rates, hidden layers, batch sizes, and hidden sizes. This comprehensive exploration helps identify configurations that provide the most effective outcomes. For the Modality agent, a learning rate of 0.001, a batch size of 1024, a hidden size of

8, and 3 hidden layers are utilized. The IHD agent uses a learning rate of 0.001, batch size of 256, hidden size of 32, with 3 hidden layers. Meanwhile, the CRRT agent employs a learning rate of 0.001, a smaller batch size of 128, a hidden size of 16, and 4 hidden layers. These configurations ensure tailored learning dynamics and adaptability for each specific task. In the process of hyperparameter tuning, we use a 5-fold cross-validation to ensure the generalizability of the model and reduce overfitting.

C Behavior policy

Since we need to simulate clinician's policy during OPE evaluation, and the accuracy of the behavior policy decides how well OPE metrics can reflect the performance of our trained RL policy. We build a surrogate behavior model π_b based on multi-layer perceptron (MLP), which simulates how clinicians view current patient features and make decisions. We used a batch size of 512, hidden size of 256, and the hidden layers are 4. We trained the model with learning rate of 0.005 for 100 epochs and achieved an accuracy of 80% on test set.

D Theoretical scaling of intermediate rewards

The central challenge in designing R_{mix} is to scale the intermediate reward parameters b and c such that they provide a useful secondary signal without overpowering the primary mortality signal a. To determine this scaling a priori, we first analyzed the cumulative impact of R_{inter} over a full trajectory. The cumulative penalty from c (Equation 3) approximates a telescoping sum, dependent only on the net SOFA change, while the penalty from b accumulates with each stagnant step:

$$R_{\text{inter\_total}} = \sum_t R_{\text{inter}}(s_t, s_{t+1}) \approx -c \cdot (\text{SOFA}_{\text{final}} - \text{SOFA}_{\text{initial}}) - b \cdot (N_{\text{stagnant\_steps}})$$

Based on the distribution of trajectory lengths (Figure 10), we estimated a typical long-stay trajectory to have approximately $T = 600$ hours (100 steps with 6-hour interval). We posited that for a patient who reaches a terminal state, a net SOFA change of 4 points and 30 stagnant steps (30% of the time) were reasonable heuristics.

Our primary design goal was to ensure the intermediate reward magnitude remained a small nudge compared to the terminal reward (using $|a| = 10$). We set a target for $|R_{\text{intermediate\_total}}|$ to be approximately 20% of $|R_{\text{terminal}}|$, or 2.0. We also enforced a balance between the two penalty components ($4c \approx 30b$) according to previous study[24, 25]. This led to a system of equations:

$$4c + 30b \approx 2.0$$
$$4c \approx 30b$$

Solving this system by substituting $c \approx 7.5b$ yielded our candidate parameters for the optimal setting: $b \approx 0.033$ and $c \approx 0.25$.

E Off-policy evaluation metrics

To evaluate the RL policy, unlike applications in robotics or gaming, where we can collect samples from real environment, in healthcare, we are unable to collect samples from real patients. Therefore, we evaluate policies using OPE metrics due to ethical and safety concerns.

Suppose we have RL policy π and π_b, the Importance Sampling (IS) metric is defined as:

$$\text{IS} = \frac{1}{N} \sum_{i=1}^{N} \rho_i R_i, \tag{5}$$

where ρ_i is the importance weight for episode i, defined as:

$$\rho_i = \prod_{t=1}^{T} \frac{\pi(a_t \mid s_t)}{\pi_b(a_t \mid s_t)}, \tag{6}$$

The Weighted Importance Sampling (WIS) metric is expressed as:

$$\text{WIS} = \frac{\sum_{i=1}^{N} \rho_i R_i}{\sum_{i=1}^{N} \rho_i}, \tag{7}$$

which normalizes the importance weights for balanced evaluation.

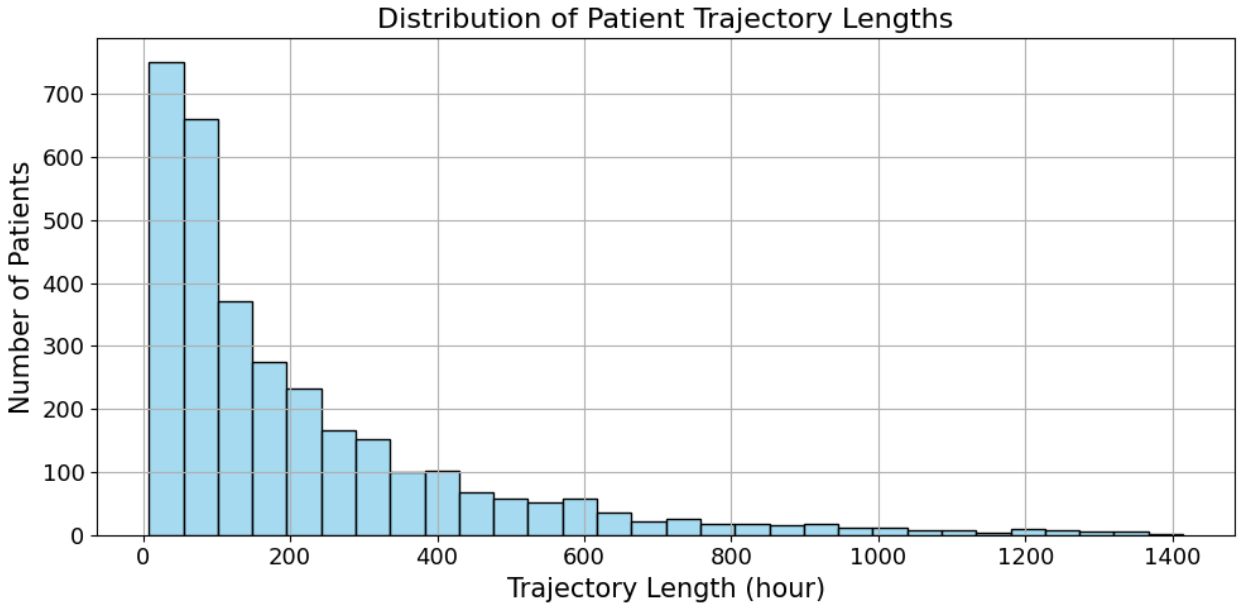

Figure 10: Distribution of patient trajectory length for MIMIC-IV dataset.

F Implementation of IHD oversampling

To implement oversampling, we first calculate the original probability of each dialysis action. We then compute a specific sampling weight for each class using the formula $w_{\text{class}} = P_{\text{target}}/P_{\text{original}}$, where P_{target} is our desired distribution. To correct for the bias introduced by this new distribution during training, we also calculate and store a normalized Importance Sampling (IS) weight for each sample, $w_{\text{IS}} = (P_{\text{original}}/P_{\text{target}})^{\beta}$, which is then applied during the model's loss calculation.

G CRRT effluent rate calculation

The total effluent dose (mL/kg/h) for CRRT is defined by the following equations:

Continuous Veno-Venous Hemofiltration (CVVH/CVVHF)

$$\text{Dose} = \frac{Q_{rep,pre} + Q_{rep,post} + Q_{net}}{W}$$

Continuous Veno-Venous Hemodialysis (CVVHD)

$$\text{Dose} = \frac{Q_{dial} + Q_{net}}{W}$$

Continuous Veno-Venous Hemodiafiltration (CVVHDF)

$$\text{Dose} = \frac{Q_{rep,pre} + Q_{rep,post} + Q_{net} + Q_{dial}}{W}$$

Notation:

$Q_{rep,pre}$ = Pre-Filter Replacement Fluid Rate (mL/h)
$Q_{rep,post}$ = Post-Filter Replacement Fluid Rate (mL/h)
Q_{dial} = Dialysate Fluid Rate (mL/h)
Q_{net} = Fluid Removal Rate (mL/h)
W = Patient Current Weight (kg)

Table 3: Feature missing rates(%) of 3 datasets.

Feature	MIMIC-IV	AmsterdamUMCdb	Dr.ECC
Age	0	0	0
Gender	0	0	0
Race	0	100	100
Weight	67.99	7.8	21.81
SBP	3.46	4.25	12.62
DBP	3.43	4.23	12.62
MBP	3.34	4.25	12.70
Heart Rate	1.83	1.96	12.62
SpO2	4.13	4.32	12.96
pH	88.35	72.94	47.26
Urea	65.53	93.68	42.09
Creatinine	65.56	91.64	42.08
Sodium	63.05	72.06	42.07
Potassium	62.70	71.65	42.07
Bicarbonate	63.62	73.02	47.26
PaO_2/FiO_2	74.31	73.03	47.26
Urine Output (6hr)	61.00	17.76	12.76
Urine Output Rate (6hr)	66.04	17.76	24.22

Table 4: The list of features for RRT.

Category	Feature Name
Demographics	Age, Gender, Weight, Race
Vital Signs	SBP, DBP, MBP, SpO_2, Heart Rate
Lab Tests	Urea, Creatinine, Potassium, Sodium, pH, PaO_2/FiO_2, Bicarbonate
Urine Output	Total urine output in the past 6 hours, Change rate of urine output in the past 6 hours

Abbreviations - SBP: Systolic blood pressure; MBP: Mean blood pressure; DBP: Diastolic blood pressure; SpO_2: Peripheral oxygen saturation; PaO_2/FiO_2: Arterial oxygen partial pressure to fractional inspired oxygen ratio.

Table 5: Variables used for modality and fluid management decisions.

Level	Action Type	Variable	Unit
Modality Selection	Modality choice	IHD and CRRT (CVVH, CVVHD, CVVHDF, SCUF)	Discrete {0, 1}
IHD Parameters	Ultrafiltration volume	Total ultrafiltrate	mL
	Dialysate rate	Dialysate rate	mL/hr
	Prefilter replacement rate	Prefilter rate	mL/hr
CRRT Parameters	Postfilter replacement rate	Postfilter rate	mL/hr
	Fluid removal rate	Hourly fluid removal or goal	mL/hr

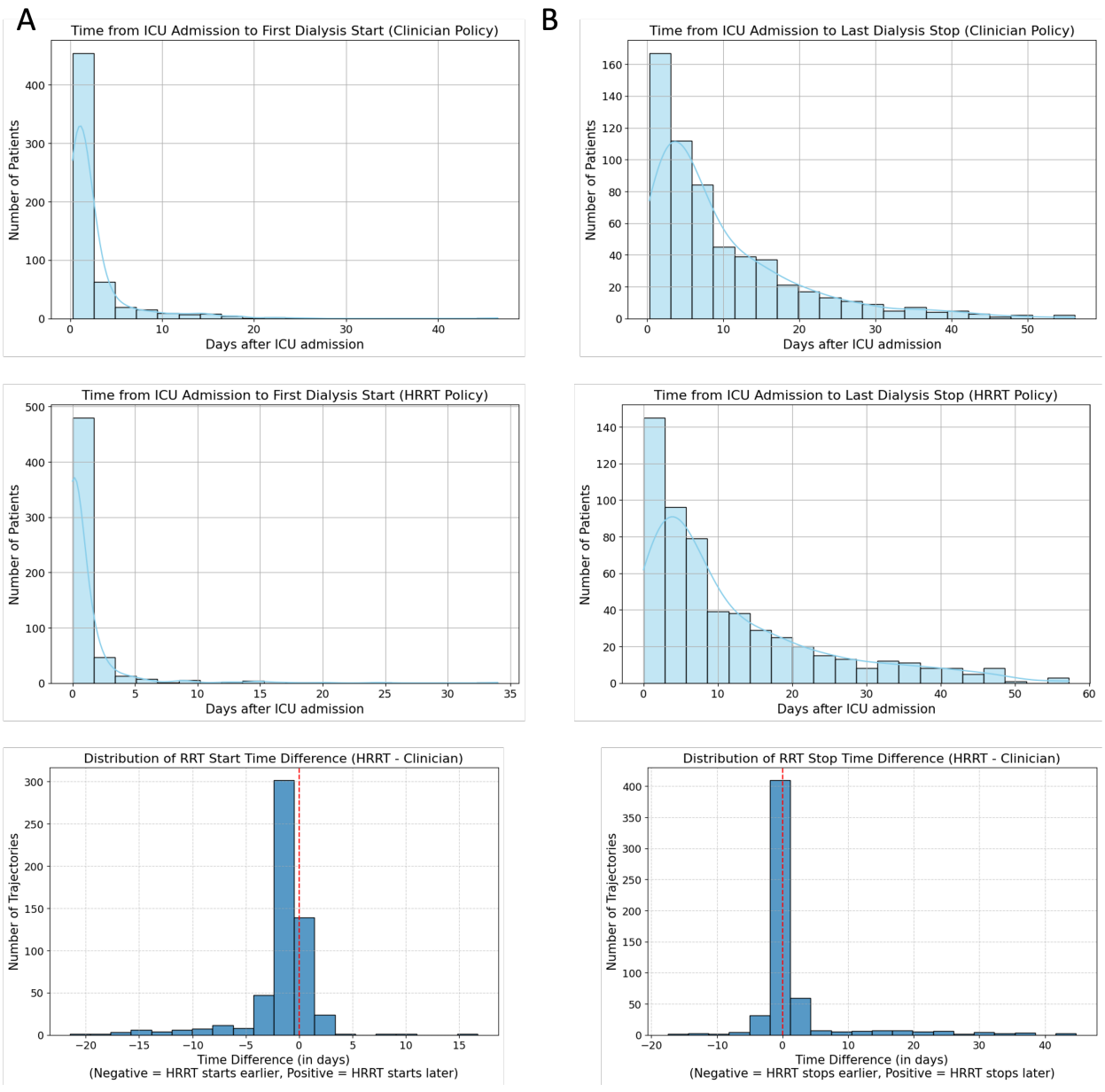

Figure 11: Temporal validation of RRT initiation and termination on the external AmsterdamUMCdb cohort.

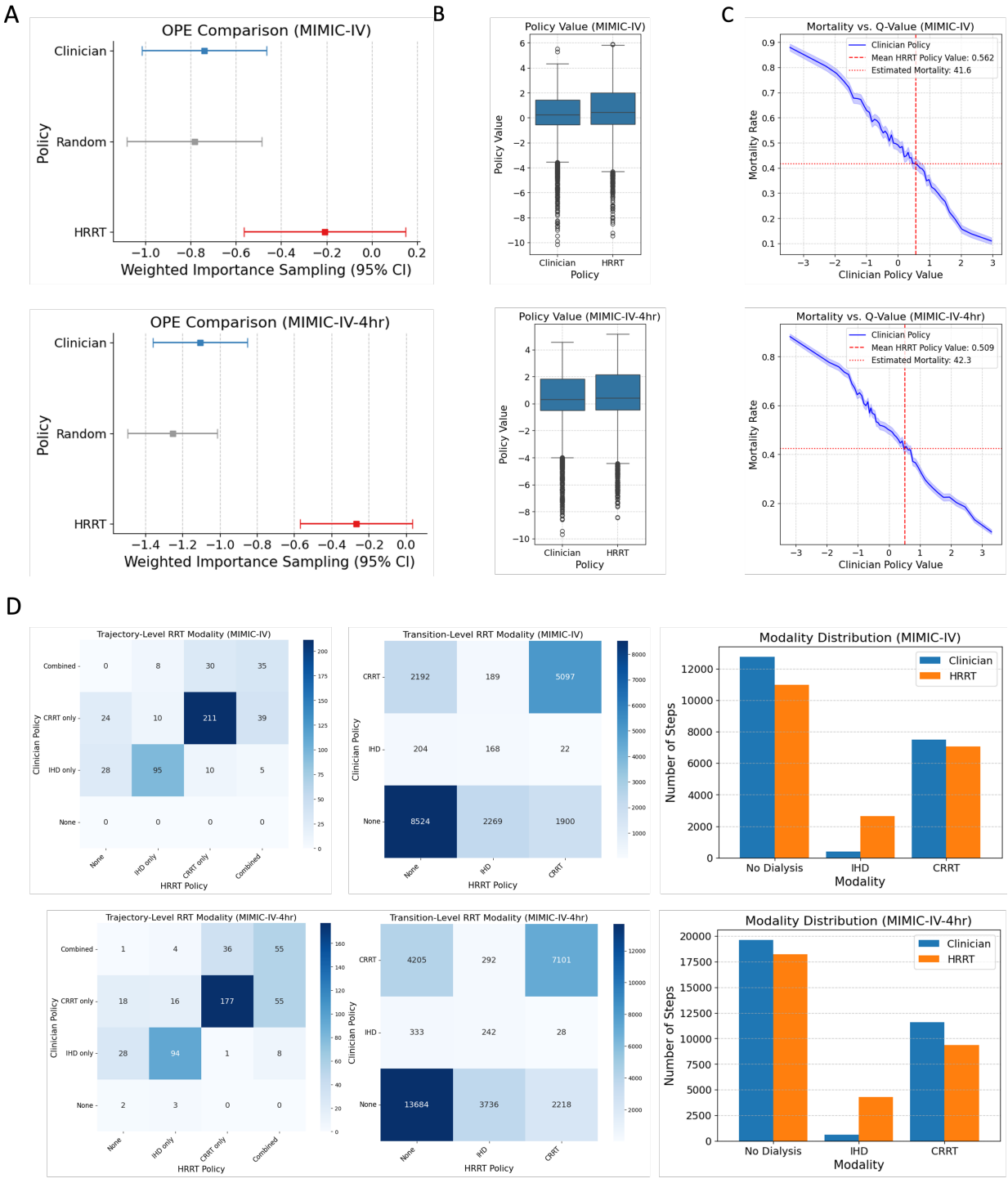

Figure 12: Sensitivity analysis on the binning interval. For each panel, the top row uses 6 hour interval while the bottom rows uses 4 hour interval.