A THEOREM PROOF

A.1 Lemma Proof

Lemma 1 (Reward Space Constraint). *Given the optimization problem:* $\min_{\tilde{x}} J(\tilde{x}) = F(\tilde{x}) + \lambda \mathcal{R}_r(\tilde{r}), \quad \lambda > 0$, *where* $F(\tilde{x})$ *is the primary objective function and* $\mathcal{R}_r$ *is the regularization term defined as:* $\mathcal{R}_r(\tilde{r}) = (ReLU(\tilde{r} - r_{\max}))^2 + (ReLU(r_{\min} - \tilde{r}))^2$, *then for any optimal solution* $\tilde{x}^*$, *the reward component* $\tilde{r}^*$ *must satisfy* $\tilde{r}^* \in [r_{\min}, r_{\max}]$. *This holds under the assumption that* $F(\tilde{x})$ *is continuous with respect to* $\tilde{r}$.

Proof. We proceed by contradiction. Assume there exists an optimal solution $\tilde{x}^*$ such that $\tilde{r}^* \notin [r_{\min}, r_{\max}]$. There are two cases to consider:

Case 1: $\tilde{r}^* > r_{\max}$

Since $\tilde{r}^* > r_{\max} \geq r_{\min}$, the regularization term evaluates to:

$$\mathcal{R}_r(\tilde{r}^*) = (\tilde{r}^* - r_{\max})^2 > 0. \tag{1}$$

Consider a new candidate solution $\tilde{x}_{\text{new}}$ identical to $\tilde{x}^*$ except for the reward component, where we set:

$$\tilde{r}_{\text{new}} = \frac{\tilde{r}^* + r_{\max}}{2}. \tag{2}$$

Note that $\tilde{r}_{\text{new}} > r_{\max}$ since $\tilde{r}^* > r_{\max}$. The regularization term at this new point is:

$$\mathcal{R}_r(\tilde{r}_{\text{new}}) = \left(\frac{\tilde{r}^* + r_{\max}}{2} - r_{\max}\right)^2 = \left(\frac{\tilde{r}^* - r_{\max}}{2}\right)^2. \tag{3}$$

The difference in total objective values is:

$$\begin{aligned} J(\tilde{x}_{\text{new}}) - J(\tilde{x}^*) = &\underbrace{\left[F(\tilde{x}_{\text{new}}) - F(\tilde{x}^*)\right]}_{\Delta F} \\ &+ \lambda \left[\left(\frac{\tilde{r}^* - r_{\max}}{2}\right)^2 - (\tilde{r}^* - r_{\max})^2\right]. \end{aligned} \tag{4}$$

Simplify the regularization difference:

$$\lambda \left[\frac{(\tilde{r}^* - r_{\max})^2}{4} - (\tilde{r}^* - r_{\max})^2\right] = -\lambda \frac{3}{4} (\tilde{r}^* - r_{\max})^2 < 0. \tag{5}$$

By continuity of F, for $\epsilon = \lambda \frac{3}{8} (\tilde{r}^* - r_{\max})^2 > 0$, there exists $\delta > 0$ such that:

$$|\tilde{r}_{\text{new}} - \tilde{r}^*| < \delta \implies |\Delta F| < \epsilon. \tag{6}$$

Select $\tilde{r}_{\text{new}}$ close enough to $\tilde{r}^*$ to satisfy this condition. Then:

$$\begin{aligned} J(\tilde{x}_{\text{new}}) - J(\tilde{x}^*) &< \epsilon - \lambda \frac{3}{4} (\tilde{r}^* - r_{\max})^2 \\ &= \lambda \frac{3}{8} (\tilde{r}^* - r_{\max})^2 - \lambda \frac{3}{4} (\tilde{r}^* - r_{\max})^2 \\ &= -\lambda \frac{3}{8} (\tilde{r}^* - r_{\max})^2 < 0. \end{aligned} \tag{7}$$

Thus $J(\tilde{x}_{\text{new}}) < J(\tilde{x}^*)$, contradicting the optimality of $\tilde{x}^*$.

Case 2: $\tilde{r}^* < r_{\min}$

The proof is symmetric to Case 1. Set:

$$\tilde{r}_{\text{new}} = \frac{\tilde{r}^* + r_{\min}}{2} < r_{\min}. \tag{8}$$

The regularization difference is:

$$\lambda \left[\left(\frac{\tilde{r}^* - r_{\min}}{2}\right)^2 - (\tilde{r}^* - r_{\min})^2\right] = -\lambda \frac{3}{4} (\tilde{r}^* - r_{\min})^2 < 0. \tag{9}$$

By continuity of F, we can find $\tilde{r}_{\text{new}}$ sufficiently close to $\tilde{r}^*$ such that:

$$J(\tilde{x}_{\text{new}}) - J(\tilde{x}^*) < 0,$$

again contradicting optimality.

Since both cases lead to contradictions, our initial assumption must be false. Therefore, for any optimal solution $\tilde{x}^*$, we must have $\tilde{r}^* \in [r_{\min}, r_{\max}]$. □

Now, we combine the above remarks and lemma to give the following core theorem.

A.2 Theorem Proof

Theorem 1 (Regularization for Compressing the Solution Space). *Let $\mathcal{X}^*_{reg}$ denote the set of global optima to the regularized problem. By introducing a prior-informed regularization term $\mathcal{R}_{total}(x)$, the solution space is effectively compressed such that $\mathcal{X}^*_{reg} \subseteq \mathcal{X}^{\epsilon}_{orig} \cap C_{prior}$, where $\mathcal{X}^{\epsilon}_{orig}$ denotes the ϵ-approximate solution space with respect to gradient matching, and C_{prior} is the constraint set induced by the regularization priors.*

Proof. Consider the regularized optimization problem:

$$\min_{\tilde{x}} J(\tilde{x}) := \|\nabla_\theta \mathcal{L}(\tilde{x};\theta) - g_{\text{obs}}\|^2 + \lambda \left(\alpha \mathcal{R}_s(\tilde{s}) + \beta \mathcal{R}_r(\tilde{r}) + \gamma \mathcal{R}_f(\tilde{x})\right). \tag{P2}$$

Let $\tilde{x}^* = (\tilde{s}^*, \tilde{a}^*, \tilde{r}^*, \tilde{s}'^*)$ be any global minimizer of $J(\tilde{x})$. Then for any $\tilde{x}$,

$$J(\tilde{x}^*) \leq J(\tilde{x}). \tag{10}$$

In particular, consider the true data sample $x_{\text{real}} = (s_{\text{real}}, a_{\text{real}}, r_{\text{real}}, s'_{\text{real}})$ such that

$$g_{\text{obs}} = \nabla_\theta \mathcal{L}(x_{\text{real}};\theta). \tag{11}$$

Since x_{real} is physically valid, it satisfies:

$$\begin{aligned} \mathcal{R}_s(s_{\text{real}}) &= \|s_{\text{real}} - \mu_s\|^2 = \delta_s \geq 0, \\ \mathcal{R}_r(r_{\text{real}}) &= 0, \\ \mathcal{R}_f(x_{\text{real}}) &= \|f(s_{\text{real}}, a_{\text{real}}) - s'_{\text{real}}\|^2 = \delta_{\text{dyn}} \geq 0. \end{aligned} \tag{12}$$

Then the objective at x_{real} is:

$$J(x_{\text{real}}) = \underbrace{\|\nabla_\theta \mathcal{L}(x_{\text{real}};\theta) - g_{\text{obs}}\|^2}_{=0} + \lambda \left(\alpha \delta_s + \gamma \delta_{\text{dyn}}\right) \triangleq \epsilon_J. \tag{13}$$

Hence, since $\tilde{x}^*$ minimizes J, we have:

$$\left\|\nabla_\theta \mathcal{L}(\tilde{x}^*;\theta) - g_{\text{obs}}\right\|^2 + \lambda \mathcal{R}_{\text{total}}(\tilde{x}^*) \leq \epsilon_J. \tag{14}$$

From this, we can extract two implications:

(1) Gradient Matching Condition.

$$\left\|\nabla_\theta \mathcal{L}(\tilde{x}^*;\theta) - g_{\text{obs}}\right\|^2 \leq \epsilon_J. \tag{2}$$

This implies $\tilde{x}^* \in \mathcal{X}^{\epsilon}_{\text{orig}}$.

(2) Regularization Constraint. From (1), we also have:

$$\lambda \mathcal{R}_{\text{total}}(\tilde{x}^*) \leq \epsilon_J \quad \Longrightarrow \quad \mathcal{R}_{\text{total}}(\tilde{x}^*) \leq \frac{\epsilon_J}{\lambda}. \tag{15}$$

Expanding the regularizer gives:

$$\alpha \mathcal{R}_s(\tilde{s}^*) + \beta \mathcal{R}_r(\tilde{r}^*) + \gamma \mathcal{R}_f(\tilde{x}^*) \leq \frac{\epsilon_J}{\lambda}. \tag{16}$$

Now, we analyze each term separately.

(a) State Prior: If $\alpha > 0$,

$$\mathcal{R}_s(\tilde{s}^*) = \|\tilde{s}^* - \mu_s\|^2 \leq \frac{\epsilon_J}{\lambda \alpha}. \tag{17}$$

Thus, $\tilde{s}^*$ lies in a hypersphere centered at μ_s.

(b) Reward Prior: If $\beta > 0$,

$$\mathcal{R}_r(\tilde{r}^*) = \left(\text{ReLU}(\tilde{r}^* - r_{\max})\right)^2 + \left(\text{ReLU}(r_{\min} - \tilde{r}^*)\right)^2 \leq \frac{\epsilon_J}{\lambda \beta}. \tag{18}$$

This implies:

$$\tilde{r}^* \in [r_{\min} - \epsilon, r_{\max} + \epsilon], \quad \text{for small } \epsilon. \tag{19}$$

(c) Dynamics Prior: If $\gamma > 0$,

$$\mathcal{R}_f(\tilde{x}^*) = \|f(\tilde{s}^*, \tilde{a}^*) - \tilde{s}'^*\|^2 \leq \frac{\epsilon_J}{\lambda \gamma}. \tag{20}$$

This implies $(\tilde{s}^*, \tilde{a}^*, \tilde{s}'^*)$ lies close to the manifold defined by the transition model.

Conclusion: From inequalities (2), (5), (7), and (8), we conclude:

$$\tilde{x}^* \in \mathcal{X}^{\epsilon}_{\text{orig}} \cap C_{\text{prior}}, \tag{21}$$

which completes the proof. □

B ADDITIONAL ENVIRONMENTAL SETUP INFORMATION

B.1 Environmental Information

The Atari environment serves as a classic benchmark for both offline and online RL, widely used to evaluate agent performance based on pixel input. The following provides brief descriptions of four widely studied Atari games used in RL benchmarks:

- Pong is a table tennis match game where the objective is to score points and defeat opponents in a left-right duel. In this environment, the agent must learn to precisely rebound the ball using its paddle while simultaneously anticipating the opponent's behavior.
- Breakout is a brick-breaking game where the player controls a paddle to rebound a ball and break the bricks positioned above. To achieve this efficiently, the player must master the relationship between the rebound angle and the speed of the ball to maximize breaking efficiency.
- Qbert is a jumping game played on an isometric grid, where the goal is to make the character jump onto each cell to change its color. In this setting, the agent needs to handle complex spatial navigation, jump control, and avoid collisions with enemies.
- Seaquest is a side-scrolling underwater shooting game where an agent controls a submarine to repel enemies and rescue divers, while periodically surfacing to replenish oxygen. These gameplay dynamics require the agent to coordinate multiple strategic goals, including offense, defense, navigation, and resource management.

MuJoCo is a high-performance physics engine widely used for simulating continuous control tasks in RL, particularly well-suited for detailed modeling of robot dynamics and contact mechanics. Below are the four common control tasks in MuJoCo:

- Hopper is a single-legged hopping robot designed to maintain stable forward locomotion. The challenge of the Hopper lies in maintaining body balance and absorbing impact during single–leg landings. Hence, the agent must learn to coordinate leg joints of to achieve continuous hopping without falling.
- Walker2d simulates a two-dimensional bipedal robot designed to walk forward, which requires the agent to maintain dynamic balance and coordinate multiple joints. Consequently, the policy must learn to achieve an efficient gait while preventing falls.
- Ant is a quadruped robot with 8 controllable joints, designed to crawl forward on a two-dimensional plane. The crawling task presents a high-dimensional action space, where the primary challenge is to coordinate the movements of multiple legs to achieve stable and fast locomotion.
- Halfcheetah is a two-dimensional robotic agent that represents a simplified, bipedal version of a cheetah, equipped with multiple joints in its hind legs and torso to simulate agile movement. This design enables dynamic locomotion patterns similar to running. In this environment, the goal of the agent is to move forward efficiently by learning to coordinate the movements of its hind legs and torso.

Car Racing is a continuous control task with pixel-based observations and is commonly used to evaluate the performance of RL algorithms under high-dimensional visual inputs and continuous action spaces. In this environment, the agent controls a car that navigates a randomly generated track, aiming to cover as much of the track as possible within a limited time to maximize its score. Each state is represented by an RGB image capturing the road layout and the position of the car from a top-down perspective. The agent operates in a continuous action space defined by three control variables: steering angle, throttle, and brake.

FrozenLake is a classic discrete reinforcement learning environment where agents need to move from the starting point to the target position on a frozen lake surface while avoiding falling into an ice cave. The environment is composed of a grid, where each cell can be a start point (S), safe ice (F), hole (H), or goal (G). In this environment, the agent can take four actions: move up, down, left, or right. However, due to slippery ground conditions, actions may stochastically deviate from the intended direction, which introduces stochasticity into policy learning.

B.2 Evaluation Metric Describe

For gradient inversion attacks involving image inputs, we primarily assess the similarity between the reconstructed and original images using PSNR and SSIM. In particular, PSNR quantifies image distortion based on the pixel-wise mean squared error (MSE), with higher values indicating better reconstruction quality and lower distortion. Unlike PSNR, SSIM focuses on measuring structural similarity in images and aligns with the perception of the human visual system. Specifically, SSIM evaluates the similarity between two images by comparing their luminance, contrast, and structure components. SSIM ranges from -1 to 1, with values closer to 1 signifying greater structural similarity between two images. For low-dimensional state inputs, we employ MSE to directly measure the mean squared difference between the features of the original and reconstructed states. For actions, we use recovery accuracy (RA) to measure the reconstruction capability of RGIA, defined as $\frac{n}{m} \times 100\%$, where n is the number of correctly reconstructed actions and m is the total number of evaluated actions. Lastly, since rewards are continuous variables, we also use MSE to evaluate the accuracy of reconstructed rewards.

In addition, to better evaluate the impact of regularization terms on the reconstructed samples, we introduce consistency metrics, which include Euclidean distance (ED), Silhouette Score (SS), Covariance Determinant (CD), and Transition Error (TE). Specifically, ED calculates the average pairwise Euclidean distance between reconstructed states, where a lower value indicates greater similarity. SS quantifies structural consistency by measuring the clustering quality of reconstructed samples under multiple random initializations. CD is the determinant of the state covariance matrix, which measures the spread of the reconstruction distribution, with lower values indicating tighter, more consistent

Table 1: Laplace noise defense results.

Variance	Environment	Loss↓	Rewards↑	MSE↓	RA↑
1e-1	Hopper	0.81	21.4	0.31	0.03
	Walker2d	0.48	30.3	0.27	0.08
	Halfcheetah	0.91	20.7	0.38	0.09
1e-2	Hopper	0.56	34.4	0.23	0.13
	Walker2d	0.30	40.8	0.20	0.14
	Halfcheetah	0.67	42.5	0.27	0.12
1e-3	Hopper	0.025	70.3	0.17e-2	0.62
	Walker2d	0.027	87.6	0.17e-1	0.70
	Halfcheetah	0.034	65.9	0.18e-1	0.68
1e-4	Hopper	0.24e-2	89.7	0.11e-3	0.79
	Walker2d	0.84e-2	102.3	0.14e-3	0.82
	Halfcheetah	0.73e-2	86.7	0.17e-3	0.74
1e-5	Hopper	0.09-2	94.5	0.09e-5	0.83
	Walker2d	0.11e-2	106.9	0.13e-5	0.83
	Halfcheetah	0.11e-2	89.6	0.18e-3	0.82

solutions. TE is an MSE between $f(\tilde{s}, \tilde{a})$ and $\tilde{s}'$, which indicates whether the reconstructed samples are consistent with the environment dynamics.

B.3 Baselines

- DGL iteratively updates virtual inputs and labels through an optimization algorithm, so that the gradients generated by the virtual data are close to the real gradients, thereby accurately restoring the original training data.
- GradInversion transforms gradient inversion into a joint optimization problem with strong prior constraints, which uses gradient matching as the main loss and BN statistics-total variation-group consistency as regularization to drive multiple randomly initialized noisy images to converge to the high-fidelity original image.
- GIFD performs layer-wise optimization of intermediate features in the generator under an L1-ball constraint, which significantly enlarges the search space while suppressing distortion. This strategy enables the attack to achieve pixel-level reconstruction even under distribution shift and noise.
- DFLeak progressively integrates high-frequency details from Prior-Free Face Restoration (PFFR) into the reconstructed image in a residual manner during gradient matching iterations. In addition, it introduces a pixel-wise update scheduling strategy, which applies a decay factor to the gradients in the fusion regions to suppress the smoothing side effects of regularization terms, thereby continuously preserving facial textures.

C ADDITIONAL EXPERIMENTS

C.1 Comparative Experiment of High-dimensional Environment

To intuitively evaluate the effectiveness of RGIA, we conduct gradient inversion attacks in RFL based on the AC framework. In this RFL-AC setting, each local agent interacts with the environment and updates its local parameters independently, without sharing raw trajectories. During the policy optimization stage, the agent uploads gradients of the value function to the central server for aggregation. RGIA intercepts these uploaded gradients and aims to reconstruct the private training data, including states, rewards, and actions. To quantitatively evaluate the reconstruction performance, we compare RGIA with four representative baselines: DGL, GradInv, GIFD, and DFLeak.

C.2 Defense Experiments

To evaluate the robustness of RGIA against differential privacy (DP)-based defense mechanisms, we simulate a privacy-preserving training scenario by adding random noise to the gradients during policy training. Specifically, we consider two commonly used DP noise distributions: Gaussian noise and Laplace noise. The noise is injected into the gradients of the value function network in the AC architecture. The configuration of each noise is as follows:

- Distributions: (1) Gaussian noise: $\mathcal{N}(0, \sigma^2)$ and (2) Laplace noise: $\text{Lap}(0, b)$, where $b = \sqrt{2}\sigma$ to match variance with Gaussian noise.
- Mean: 0
- Variance (σ^2) range: $[10^{-1},\ 10^{-2},\ 10^{-3},\ 10^{-4},\ 10^{-5}]$

This range allows us to evaluate the impact of different privacy levels, from strong (high noise) to weak (low noise) protection.

Table 2: The impact of different defense methods on RGIA performance.

Environment	Defense method	GME↓	Loss↓	MSE↓	Rewards↑
Pong	HE	0.89	2.34	0.56	20.3
	GQ (4 bit)	0.51	1.35	0.27	-17.4
	GQ (8 bit)	0.11	0.27	0.071	4.3
Hopper	HE	0.63	1.78	0.79	93.7
	GQ (4 bit)	0.34	0.73	0.21	34.6
	GQ (8 bit)	0.09	0.25	0.038	70.3

Table 3: Effect of reward range constraint (β) on reward reconstruction validity.

	Reward Error ↓	Invalid $\tilde{r}$ Ratio(%)↓
β	Pong/Qbert	Pong/Qbert
0.0	0.33/0.28	0.27/0.19
0.01	0.29/0.19	0.12/0.07
0.1	0.14/0.13	0.00/0.00
1.0	0.08/0.09	0.00/0.00
10.0	0.08/0.08	0.00/0.00

Table 1 presents the performance of the RGIA attack under Laplace noise defense across three environments (Hopper, Walker2d, and Halfcheetah) with varying noise variances from $1e^{-1}$ to $1e^{-5}$. We observe a consistent trend across all metrics, indicating that increasing the strength of differential privacy (i.e., higher noise variance) significantly reduces the attack effectiveness, while weakening the performance of the learned policy.

Furthermore, we investigate the impact of homomorphic encryption (HE) and gradient quantization (GQ) on the performance of RGIA. To simplify experimental complexity, we implement a lightweight HE scheme that performs gradient encryption via additive operations before uploading the encrypted gradients. Concurrently, we apply GQ to map gradients into low-bit representations (e.g., 8-bit or 4-bit). Since these techniques obscure gradient distributions, we propose the gradient matching error (GME) metric to intuitively quantify gradient variations. The combined effects of HE and GQ on RGIA are summarized in Table 2.

From the experimental results, homomorphic encryption can effectively defend against data-reconstruction attacks launched by RGIA without compromising the original algorithmic performance (i.e., reward). However, it is well known that homomorphic encryption introduces substantial computational overhead, which significantly limits its practical application in FRL. Although gradient quantization can prevent data reconstruction, it adversely affects the original performance of the FRL algorithm. In particular, the performance is notably degraded when gradient values are converted to low precision (4 bit).

C.3 Visualization Experiment

We visualize reconstructed images from five methods (i.e., RGIA, DGL, GrandInv, GIFD, and DFLeak) across four environments: Pong, Breakout, Qbert, and Car Racing. As shown in Figure 1, GrandInv, GIFD, and DFLeak introduce improved feature loss functions, enabling the reconstruction of samples that are more similar to the original images. However, since these methods cannot address the pseudo-solution problem in FRL, they occasionally reconstruct failed samples. In particular, although DFLeak is capable of reconstructing high-quality states in most environments, the reconstructed states in the Qbert environment contain some noise (as indicated by the red ellipse). Moreover, GrandInv reconstructed high-quality images in the Car Racing environment, but the reconstructed state is not the original state (as indicated by the red box).

C.4 Low-dimensional Reconstruction Experiment.

Since the state in FrozenLake is a discrete variable, we do not directly optimize the state variables but instead handle these variables similarly to discrete actions. Specifically, we first generate a random vector of shape $[N, M, C]$, where N is the batch size, M is the state dimension, and C is the number of grid value intervals. Next, we optimize these variables using gradient descent and obtain the reconstructed state by determining the state values via index C. The experimental results are shown in Figure 2, where each result is the average result of 10 random seeds. As can be seen, all methods achieve remarkably high performance in the FrozenLake environment. This is primarily because FrozenLake is a relatively simple and deterministic environment, where the training states and actions can be precisely manipulated without the need for additional constraints.

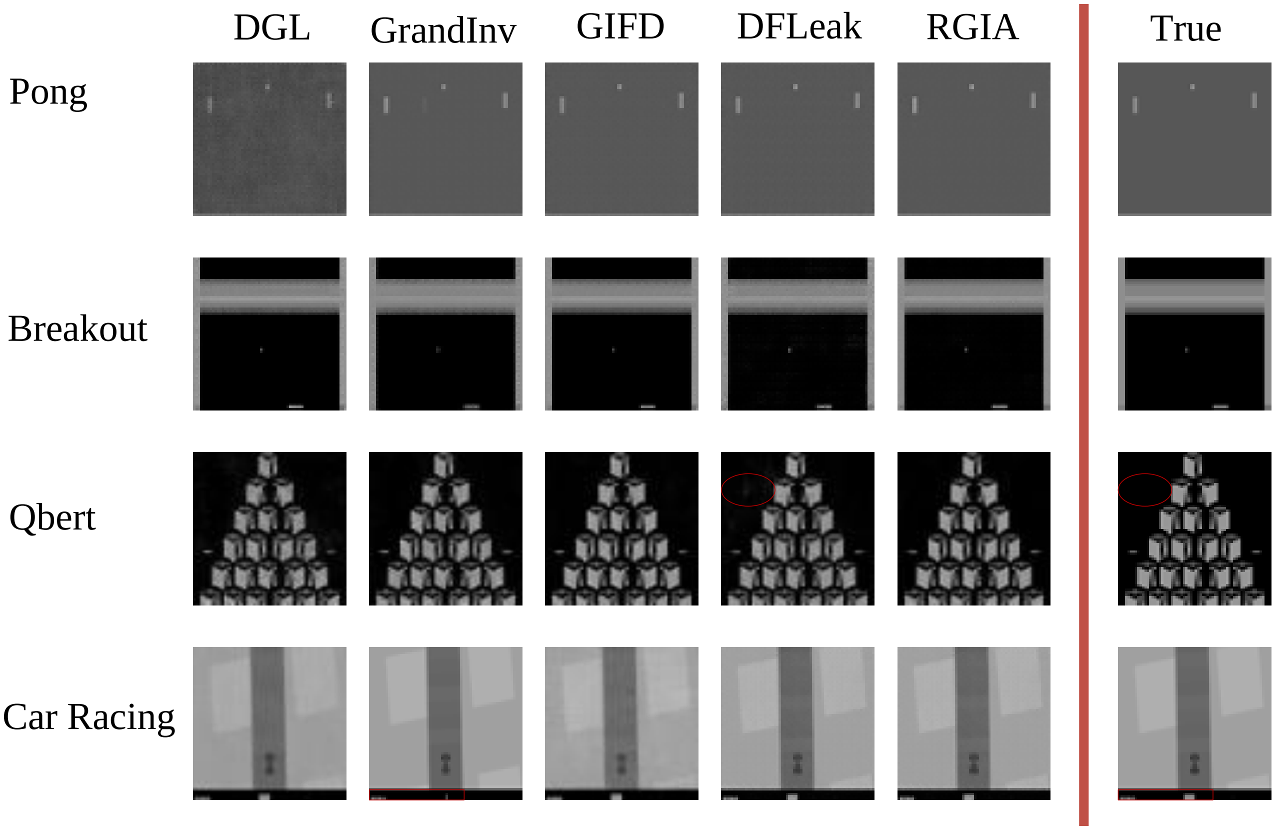

Figure 1: Reconstruction examples of different methods.

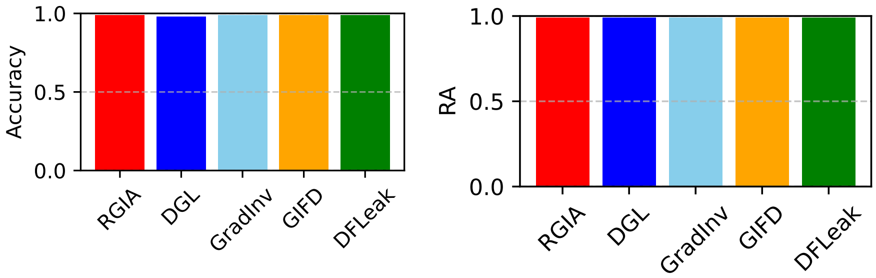

Figure 2: The comparison results of FrozenLake environments.

C.5 Parameter Sensitivity Analysis

Similar to the sensitivity experiment on α, we vary β while keeping $\alpha = 1.0$ and $\gamma = 1.0$ fixed. Experimental results include the absolute reward reconstruction error and the proportion of invalid samples, where invalid samples are defined as reward values $\tilde{r}$ outside the legal range $[r_{\min}, r_{\max}]$ imposed by the environment. As shown in Table 3, varying β within the range of 0.0 to 10.0 significantly alters the reward error and the proportion of invalid reconstruction samples. In particular, a small value (e.g., $\beta = 0.01$) effectively eliminates invalid samples. Moreover, experimental results show that, unlike state regularization, reward constraints are not highly sensitive to precise tuning.

To assess the contribution of the dynamics consistency regularization, we conduct a sensitivity analysis by varying the regularization weight $\gamma \in \{0.0, 0.01, 0.1, 1.0, 10.0\}$, while keeping the other regularization terms fixed ($\alpha = 1.0, \beta = 1.0$). This setup allows us to evaluate the effect of the transition prior that penalizes inconsistencies between the reconstructed next state $\tilde{s}'$ and the prediction of a pre-trained forward model $f(\tilde{s}, \tilde{a})$. Here, we use three metrics: Transition Error (TS), Silhouette Score (SS), and Covariance Determinant (CD) to comprehensively evaluate the impact of different γ values.

Table 4: Effect of dynamics consistency regularization (γ) on transition realism and sample concentration. The TE values reported in the table are the actual values scaled by a factor of 10^{-4}.

	TE↓	SS↑	CD↓
γ	Pong/Qbert	Pong/Qbert	Pong/Qbert
0.0	0.31/0.35	0.28/0.19	5.3e-2/4.9e-2
0.01	0.21/0.27	0.42/0.39	3.7e-2/4.1e-2
0.1	0.12/0.18	0.66/0.59	1.5e-2/2.1e-2
1.0	**0.10/0.13**	**0.72/0.68**	**1.2e-3/1.3e-3**
10.0	0.11/0.13	0.69/0.66	2.1e-3/1.9e-3

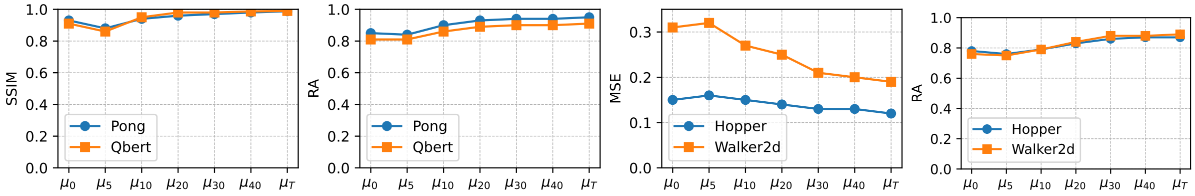

Figure 3: The impact of state prior μ_s on the performance of RGIA.

Table 4 presents experimental results for varying γ values in RGIA reconstruction. When no dynamics prior is applied ($\gamma = 0.0$), the reconstructed samples exhibit larger TE, lower SS, and higher CD compared to those with constraints. These results indicate that many generated trajectories violate environment dynamics and are highly dispersed in the state space. As γ increases, three metrics improve significantly. For instance, at $\gamma = 1.0$, the TE decreases by over 67%, and the SS increases to 0.72. Additionally, CD drops by nearly two orders of magnitude, demonstrating a pronounced shrinkage of the solution space. Interestingly, further increasing γ to 10.0 results in degraded metrics (i.e., increased TE and CD, and decreased SS). This phenomenon suggests that overly aggressive regularization may over-constrain the optimization process, which reduces flexibility and diversity.

C.6 State μ_s

To investigate the impact of deviations in the state prior μ_s used for regularization in RGIA on attack performance, we design an ablation experiment with biased priors. Specifically, the experiment involves a state-free prior and mean priors derived from datasets of varying sizes. For the latter, we select 500, 1000, 2000, 3000, 4000 and all training samples to calculate the mean prior. Ablation experiments are conducted in the Pong, Qbert, Hopper, and Walker2d environments, and the evaluation is performed using the SSIM, MSE, and RA metrics to quantify performance impact. The experimental results are shown in Figure 3, where μ_0 denotes the state regularization not applied, and u_5, u_{10}, u_{20}, u_{30}, u_{40}, u_T represent the state priors calculated with different data volumes, respectively.

Experimental results demonstrate that higher-quality priors can significantly improve the accuracy of data reconstruction. Specifically, in the Pong and Qbert environments, as the amount of data used to estimate the state prior increases, the reconstructed states become increasingly similar to the original ones, as measured by SSIM. Meanwhile, the action recovery accuracy (RA) also improves significantly. However, when the prior data exceeds 2000 samples, neither SSIM nor RA increases significantly. This plateau may be explained by the fact that 2000 samples are sufficient to accurately estimate the state prior distribution μ_s. An interesting exception occurs when only 500 samples are used to calculate μ_s: in this case, the reconstruction ability of RGIA is actually worse than that without state regularization. This decline likely results from a significant deviation of μ_s from the true distribution, which causes the optimization process to converge to a semantically incorrect solution. Similarly, Hopper and Walker2d show a similar trend. When the sample size exceeds 3000, μ_s no longer has a significant effect on the state MSE and RA.

In summary, our ablation study demonstrates that while an accurate state prior μ_s is crucial for RGIA to achieve optimal attack performance, only a moderate amount of data (typically ranging from 2000 to 3000 samples across environments, representing a mere 0.3% of the 1M total training dataset) is sufficient to estimate a prior that saturates the reconstruction accuracy improvements.

C.7 Effect of transition model Training Data Size on RGIA Performance.

To evaluate the impact of transition model accuracy on the proposed RGIA method, we conduct an ablation study by varying the amount of data used to train the transition model f in the dynamic consistency regularizer $\mathcal{R}_f = \|f(s, a) - s'\|^2$. For each environment (Walker2d and Breakout), we train f using different amounts of prior data $\{500, 1000, 2000, 3000, 4000, all\}$, where *all* denotes that the transition model is trained using the entire dataset.. The model architecture and optimization settings are kept identical across all data settings, and the remaining RGIA hyperparameters are fixed to their default values. After training f, we run RGIA with the learned transition model and evaluate reconstruction quality on 1,000 randomly selected target gradients. For each data setting, the experiment is repeated with 10

Table 5: The influence of transition models on RGIA reconstruction capabilities. 0 **means that the dynamic consistency regularization is not used, and 1M means that the transition model is trained by all training data.**

Numbers	Walker2d			Breakout		
	MSE↓	RA↑	TE↓	MSE↓	RA↑	TE↓
0	0.29e-7	0.77	0.16e-5	0.83e-6	0.82	0.23e-4
500	0.11e-6	0.77	0.18e-5	0.84e-6	0.80	0.27e-4
1000	0.28e-7	0.81	0.14e-5	0.80e-6	0.84	0.22e-4
2000	0.24e-7	0.84	0.13e-5	0.79e-6	0.87	0.13e-4
3000	0.21e-7	0.88	0.12e-5	0.79e-6	0.88	0.12e-4
4000	0.20e-7	0.89	0.12e-5	0.71e-6	0.87	0.11e-4
1M	0.19e-7	0.91	0.11e-5	0.69e-6	0.88	0.09e-4

different random seeds, and we report the mean values for three metrics: state MSE, action recovery accuracy (RA), and state transition error (TE).

As shown in Table 5, increasing the number of samples used to train the transition model f generally improves RGIA performance across state MSE, RA, and TE. For Walker2d, RA rises steadily from 0.77 without dynamic consistency regularization to 0.88 with 3,000 samples, while TE decreases from 0.16×10^{-5} to 0.12×10^{-5}. The MSE shows minor fluctuations at small sample sizes but converges to the lowest value (0.21×10^{-7}) with the largest dataset. For Breakout, RA improves from 0.82 to 0.88, and TE drops markedly from 0.23×10^{-4} to 0.12×10^{-4}, with MSE gradually decreasing from 0.83×10^{-6} to 0.79×10^{-6}. These results indicate that larger training sets enable the transition model to more accurately approximate environment transitions, thereby reducing reconstruction error and improving both state and action recovery in RGIA attacks. However, the improvement in each metric becomes less pronounced when the sample size reaches 3,000. Even when the transition model is trained on the entire training dataset, there is no significant improvement in MSE, RA, or TE.