| Episodes | PGIRL |  |  |  |  | GIRL |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | R | RB | G | GB | eNAC | R | RB | G | GB | eNAC |
| 10 | 0.031<br>±0.011 | 0.020<br>±0.001 | 0.020<br>±0.002 | 0.021<br>±0.002 | 0.037<br>±0.009 | 16.1<br>±1.5 | 8.4<br>±1.1 | 46.1<br>±6.2 | 8.6<br>±1.3 | 2.2<br>±0.3 |
| 100 | 0.172<br>±0.007 | 0.168<br>±0.001 | 0.166<br>±0.003 | 0.176<br>±0.003 | 0.172<br>±0.003 | 74.0<br>±19.8 | 70.2<br>±9.8 | 249.7<br>±73.6 | 56.4<br>±8.3 | 32.0<br>±6.5 |
| 1000 | 1.064<br>±0.006 | [illegible]<br>±0.011 | [illegible]<br>±0.031 | 1.683<br>±0.005 | 1.040<br>±0.038 | 971.3<br>±270.0 | [illegible]<br>±25.2 | [illegible]<br>±1128.2 | [illegible]<br>±32.5 | [illegible]<br>±14.7 |

Table 1: Average Computational Time (s) for the generation of the results presented in Figure 2b (5D LQR).

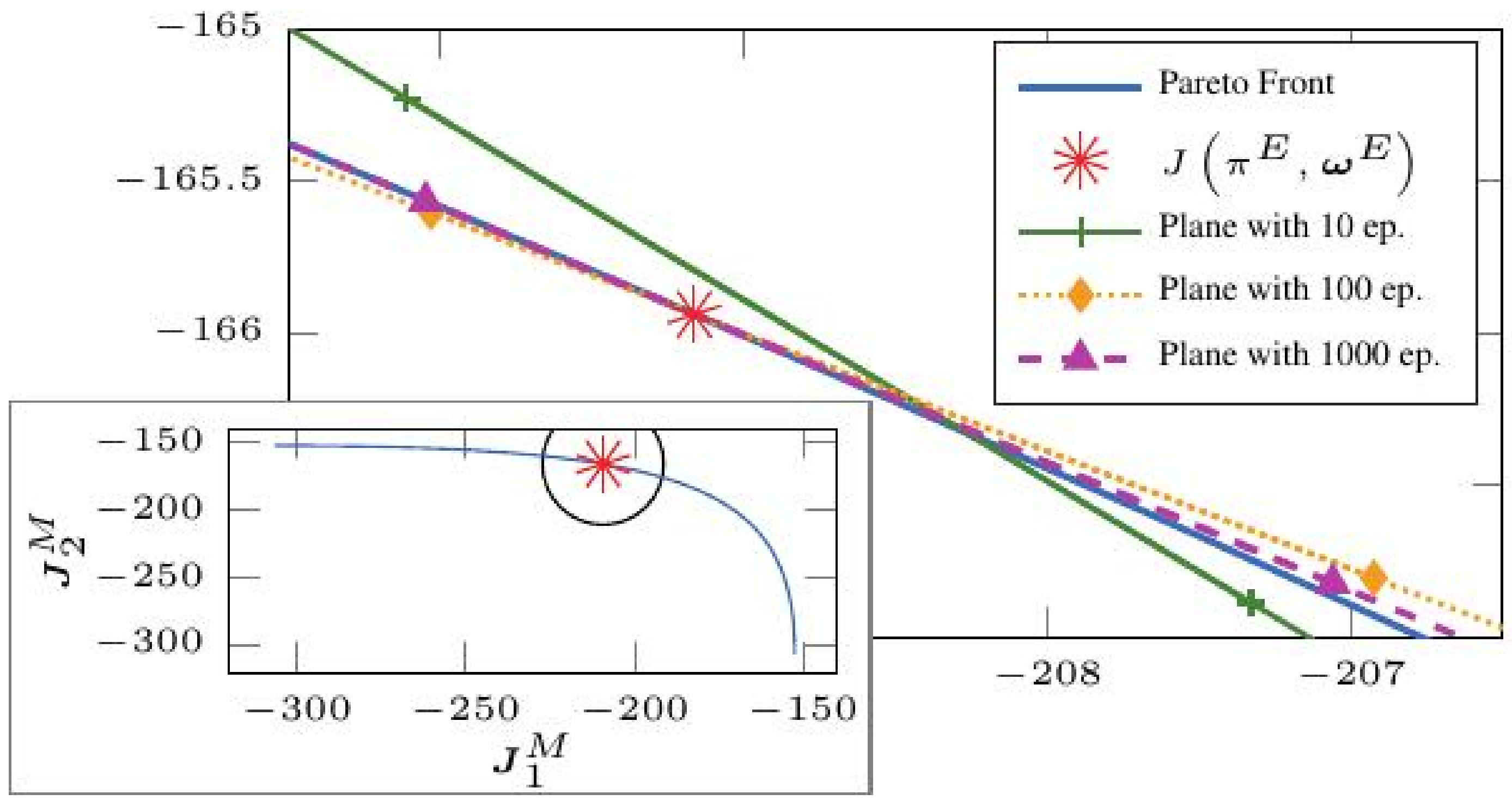

Figure 1: Behavior of the eNAC-PGIRL in 2D LQR. Figure reports the planes in the objective space identified by the PGIRL algorithm with 10, 100 and $1,000$ trajectories. This figure represents a zoom of the frontier around the current solution. The entire frontier is shown in the corner figure.

## Experiments

This section is devoted to the empirical analysis of the proposed algorithms. The first domain, a linear quadratic regulator, is used to illustrate the main characteristics of the proposed approach, while the mountain car domain is used to compare it against the most related approaches.

### Linear Quadratic Regulator

In this section we provide a set of experiments in the well-known Linear Quadratic Regulator (LQR) problem (Peters and Schaal 2008b). These experiments are meant to be a proof of concept of our algorithm behavior. We consider the multi-dimensional, multi-objective version of the problem provided in (Pirotta, Parisi, and Restelli 2015), the reader may refer to it for the settings. We consider a linear parametrization of the reward function:
$\mathcal{R}(s,a;\boldsymbol{\omega}) = -\sum_{i=1}^{q} \boldsymbol{\omega}_i(s^{\mathrm{T}} Q_i s + a^{\mathrm{T}} R_i a)$.

**Exact Expert's Policy Parameters** In the first test, we focus on the PGIRL approach where the gradient directions are computed using the eNAC algorithm (eNAC–PGIRL) in the 2D LQR domain. The goal is to provide a geometric interpretation of what the algorithm does. Figure 1 reports the planes (lines in 2D) and the associated weights obtained by the eNAC-PGIRL algorithm with different data set sizes. As the number of samples increases, the accuracy of the plane identified by the algorithm improves. With $1,000$ trajectories, the plane is almost tangent to the Pareto frontier. The points on the planes are obtained from the Gram matrix (after a translation from the origin).

The next set of experiments deals with the accuracy and time complexity of the proposed approaches (GIRL and PGIRL) with different gradient estimation methods (REINFORCE w/ and w/o baseline (RB, R), GPOMDP w/ and w/o baseline (GB,G) and eNAC). We selected 5 problem dimensions: (2, 5, 10, 20). For each domain we selected 20 random expert's weights in the unit simplex and we generated $5$ different datasets. It is known that the contribute of the baseline for the gradient estimation is important and cannot be neglected (Peters and Schaal 2008b). Consider Figure 2a, using plain R and G the GIRL algorithm is not able to recover the correct weights. Although the error decreases as the number of trajectories increases, the error obtained with $1,000$ trajectories is larger than the one obtained by the baseline–versions (RB and GB) with only $10$ trajectories. For this reason we have removed the plain gradient algorithms from the other tests. Figures 2b–2d replicate the test for increasing problem dimensions. All the algorithms show a decreasing error as the number of samples increases, but no significant differences can be observed. From such results, we can conclude that, when the expert's policy is known, GIRL is able to recover a good approximation of the reward function even with a few sample trajectories.

In Table 1 we show how the computational times of the different algorithms change as a function of the number of available trajectories. PGIRL algorithm outperforms GIRL for any possible configuration. Recall that PGIRL has to compute a fixed number of gradients, equal to the reward dimensionality, while GIRL is an iterative algorithm. We have imposed a maximum number of function evaluations to $500$ for the convex optimization algorithm. The results show that the difference in the time complexity exceeds two orders of magnitude.[5] Although, the best choice for linear reward parametrizations is PGIRL, GIRL has the advantage of working even with non–linear parametrizations.

**Approximated Expert's Policy Parameters** In the following the parameters of the expert's policy are unknown and we have access only to expert's trajectories. In order to apply GIRL and PGIRL algorithms we have to learn a parametric policy from the data, that is, we have to solve a MLE problem (see Section ). We consider a standard 1–dimensional LQR problem. Under these settings the policy is a Gaussian $a_t \sim \mathcal{N}(ks, \sigma^2)$ and the reward is $r_t = -\boldsymbol{\omega}_1 s_t^2 - \boldsymbol{\omega}_2 a_t^2$. The initial state is randomly selected in the interval $[-3, 3]$. Since the action space is continuous and

[5] The performance of the GIRL algorithm depends on the implementation of the convex algorithm and its parameters. Here we have exploited NLopt library (http://ab-initio.mit.edu/nlopt).