Supplementary Material for Action Space Reduction for Planning Domains

Harsha Kokel, **Junkyu Lee**, **Michael Katz**, **Kavitha Srinivas** and **Shirin Sohrabi**
IBM T.J. Watson Research Center, Yorktown Heights, USA
{harsha.kokel, junkyu.lee, michael.katz1, kavitha.srinivas}@ibm.com, ssohrab@us.ibm.com

The code and reference to the full paper are available at https://github.com/IBM/Parameter-Seed-Set.

A Complexity Proof of Parameter Seed Set Decision Problem

Definition 1. *(Bounded Parameter Seed Set Decision Problem (PSS))*
Input: *A set of n parameters $V = \{v_1, \cdots, v_n\}$, a number of objects per parameter $c(v) \in \mathbb{N}$, a set of m equations $E = \{e_1, \cdots, e_m\}$, and an integer k. Each equation is of the form $e = P_e \rightarrow V_e$, where $P_e, V_e \subseteq V$ and it signifies that the parameters in V_e can be deduced if the parameters in the set P_e are provided.*
$\overline{V} \subseteq V$ is a parameter seed set if all other parameters can be eventually deduced from it, i.e., there exists a sequence of equations $e_1 \ldots e_x$ such that $\overline{V} = \overline{V}_0$, $\overline{V}_x = V$, $\overline{V}_{i+1} = \overline{V}_i \cup V_{e_i}$, and $P_{e_i} \subseteq \overline{V}_i$.

Problem: *Is there a parameter seed set $\overline{V} \subseteq V$ such that $\prod_{v \in \overline{V}} c(v) \leq k$?*

Note that a special case of PSS for all parameters having the same number of objects > 1 is equivalent to an existing problem from planning literature called The Minimal Seed Set Problem [Gefen and Brafman, 2011].

Definition 2. *(Bounded Seed Set Decision Problem)*
Input: *A set of nutrients C, a set of reactions R with each $r = (X, Y) \in R$ and $X, Y \subseteq C$ and an integer n.*

$\overline{C} \subseteq C$ is a seed set if all other nutrients can be eventually deduced from it, i.e., there exists a sequence of reactions $r_1 \ldots r_x$ such that $\overline{C} = \overline{C}_0$, $\overline{C}_x = C$, $\overline{C}_{i+1} = \overline{C_i} \cup Y_{r_i}$, and $X_{r_i} \subseteq \overline{C}_i$.

Problem: *Is there a seed set $\overline{C} \subset C$ of size $\leq n$?*

Theorem 1. *The bounded PSS decision problem is at least as hard as the Bounded Seed Set decision problem.*

Proof. The proof is straightforward since the seed set is a special case of the parameter seed set with a uniform number of objects per parameter. □

The Minimal Seed Set Problem was shown to be NP-hard by reduction from set cover [Gefen and Brafman, 2011]. While the authors do not provide a detailed reduction, for completeness we present here the reduction, obtained via correspondence with the authors.

Definition 3. *(Set Cover Decision Problem SET-COVER)*
Input: *A set of elements U, a set of subsets $S = \{S_1, \cdots, S_b\}$ such that union of S covers all the elements in U, and an integer n.*
The set $\overline{S} \subset S$ is a set cover if

$$\bigcup_{S_i \in \overline{S}} S_i = U$$

Problem: *Is there a set cover $\overline{S} \subset S$ of size $\leq n$?*

Theorem 2. *Bounded Seed Set decision problem is NP-complete.*

Proof. The membership in NP is trivial. To prove that the bounded seed set decision problem is NP-hard, we reduce SET-COVER to bounded seed set. Given U and S, we define C and R as follows.

- For every element $u \in U$, we add a nutrient c_u to the nutrient set C. We denote these nutrients by C_U.
- For every subset $S_i \in S$ we add a nutrient x_i to C. We denote these nutrients by C_S.
- For every subset $S_i \in S$, we add a reaction $(\{x_i\}, \{c_u \mid u \in S_i\})$ to R.
- We add the reaction (C_U, C_S) to R.

Let $\overline{C} \subseteq C$ be a seed set of size k. Assume W.L.O.G. that $\overline{C} \subseteq C_S$. The assumption is valid, as some subset of these nutrients is sufficient for deriving all the nutrients in V_U, which, in turn, can be used to derive all of C_S. Further, since it is a seed set, we have

$$C_U = \bigcup_{x_i \in \overline{C}} \{c_u \mid u \in S_i\},$$

and therefore
the corresponding set $\overline{S} = \{S_i \in S \mid x_i \in \overline{C}\}$ is a set cover, of the same size k.

□

B Example

Consider the `unload_truck` lifted action in a logistics domain and two LMGs shown below.

Schematic Operator:

$$
\begin{aligned}
&\texttt{unload\_truck} \\
&\quad :\texttt{params}\ \{?\texttt{p} : \texttt{pkg}, ?\texttt{t} : \texttt{truck}, ?\texttt{l} : \texttt{loc}\} \\
&\quad :\texttt{pre}\ \{\texttt{at}(?\texttt{t}, ?\texttt{l}), \texttt{in}(?\texttt{p}, ?\texttt{t})\} \\
&\quad :\texttt{add}\ \{\texttt{at}(?\texttt{p}, ?\texttt{l})\} \\
&\quad :\texttt{del}\ \{\texttt{in}(?\texttt{p}, ?\texttt{t})\}
\end{aligned}
$$

LMGs:

$l_1 = \langle\{?\texttt{a}\}, \{?\texttt{b}\}, \{\texttt{in}(?\texttt{a}, ?\texttt{b})\}\rangle$
$\triangleright$ $\because$ Package can only be in one truck
$l_2 = \langle\{?\texttt{c}\}, \{?\texttt{d}\}, \{\texttt{at}(?\texttt{c}, ?\texttt{d})\}\rangle$
$\triangleright$ $\because$ Object can only be at one location

Here, since `in(?p, ?t)` $\sqsubseteq$ `in(?a, ?b)`, the parameter `?t` can be inferred if the parameter `?p` is known. So, the parameter `?t` can be removed from the seed set if we have `?p` in the seed set. Further, as `at(?t, ?l)` $\sqsubseteq$ `at(?c, ?d)`, the parameter `?l` can be inferred if the parameter `?t` is known. So, parameter `l` can also be removed from the seed set. Note that `?l` can be removed from the seed set even if `?t` is not in the seed set as the parameter `?p` guarantees that we can infer `?t`.

C Additional Results

In this section, we present the additional results from our experiments.

C.1 Reduction in label set

Table 1 present the max and mean percent (number) of the parameters identified as seeds in each domain. Additionally, it also presents the time taken (in seconds) for finding the parameter seeds for all the schematic operators. The results are aggregated over all the problem files in that domain. The IPC problem files and domains were obtained from the downward benchmark at the following GitHub repository https://github.com/aibasel/downward-benchmarks and the HTG domains were obtained from the following GitHub repository https://github.com/abcorrea/htg-domains.

In the main paper, we present the action reduction on seven HTG domains. Here, we present the reduction on the remaining three domains: organic-synthesis-alkene, organic-synthesis-MIT, organic-synthesis-MIT. Figure 1 compares the size of the label sets L' and L, obtained with and without the reduction, respectively. As these values are quite large and the log plot might not adequately reflect the magnitude of the reduction, we additionally present the values in Table 2. The 5^{th} column in the table shows the reduction in action labels achieved by our approach.

C.2 Learning reinforcement learning policies

To evaluate the advantage of reducing the label set size in planning as RL, we cast the PDDL task as an MDP with either the regular (all ground actions) or the reduced label set and learn an RL policy. We focus on 4 classical planning domains, *Ferry, Gripper, Blocks,* and *Logistics*. Table 4 summarizes the number of objects and the number of action labels in each of these domains. Column 3 presents the number of ground actions identified by pyperplan[1], used by PDDLenv of Gehring *et al.* (2022). Column 4 presents the number of reduced actions identified by our approach. We employ the Double DQN implementation from the ACME RL library [Hoffman *et al.*, 2020] to learn a state-action value function. Table 3 describes all the hyperparameters used in the RL experiments. These experiments were performed on computing clusters with Intel(R) Xeon(R) CPU E5-2667 v2 @ 3.30GHz and Tesla K80.

C.3 Lifted successor generation

In our preliminary experiments on using the label reduction for lifted successor generation, we modify the procedure of the Powerlifted planner by pre-processing the tables. We pre-join the precondition tables with the corresponding lifted mutex group table, over non-seed parameters. This allows us to reduce the size of the tables in the query. The reduction in the table sizes is presented in the main paper. We reproduce those plots, individually, for each domain in Figure 2 for clarity.

Further, we present the resulting difference in time taken to generate applicable actions. The comparison of the time of our approach with the original procedure on the HTG benchmark set is depicted in Figure 3, with the three domains where the time improvement is clearly visible shown in (a) and the domains with no visible improvement depicted in (b). Although pre-joining the tables have an additional cost, we can see from Figure 3 that the computational overhead is negligible. Further, it is clear from Figure 3 (a) that the reduced set of parameters can speed-up lifted successor generation. Even though the computation time gained in Figure 3 might seem small, note that the applicable actions are queried multiple times in the search process, including search node expansion and heuristic computation. Hence, this gain can make a notable difference in the planning process. All the experiments on lifted successor generations were run with Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz.

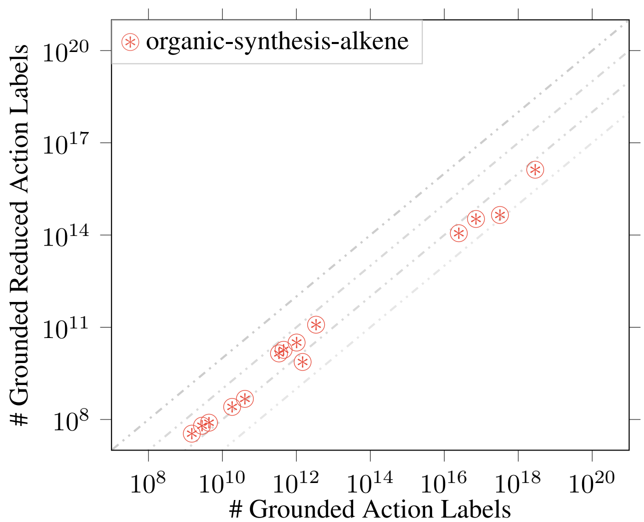
References

[Gefen and Brafman, 2011] Avitan Gefen and Ronen I. Brafman. The minimal seed set problem. In *ICAPS*, number 1, pages 319–322, 2011.

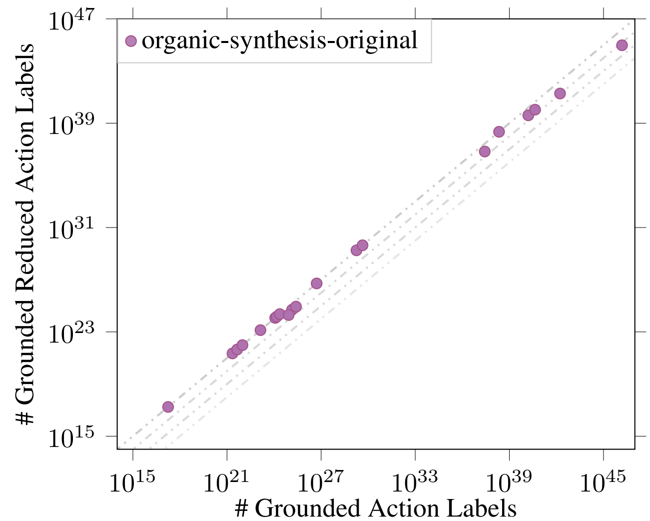
[Gehring *et al.*, 2022] Clement Gehring, Masataro Asai, Rohan Chitnis, Tom Silver, Leslie Pack Kaelbling, Shirin Sohrabi, and Michael Katz. Reinforcement learning for classical planning: Viewing heuristics as dense reward generators. In *ICAPS*, pages 588–596, 2022.

[Hoffman *et al.*, 2020] Matt Hoffman, Bobak Shahriari, John Aslanides, Gabriel Barth-Maron, et al. Acme: A research framework for distributed reinforcement learning. *CoRR*, abs/2006.00979, 2020.

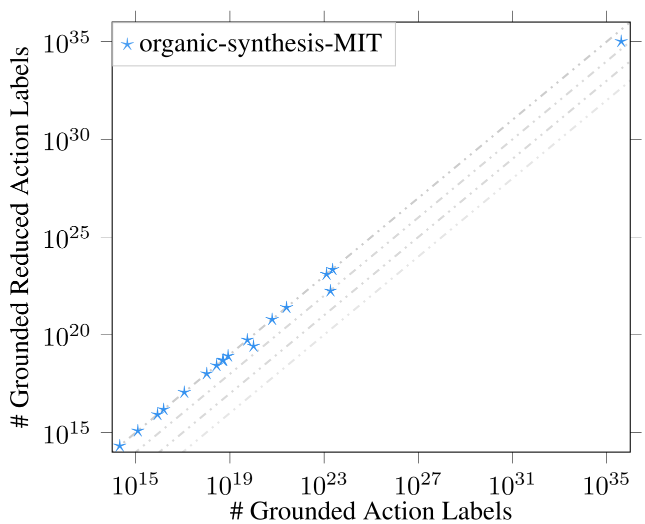
[1]https://github.com/aibasel/pyperplan



(a)



(b)



(c)

Figure 1: Comparison of label set sizes on three HTG domains. (a) organic-synthesis-alkene, (b) organic-synthesis-original, and (c) organic-synthesis-MIT.

Domain	seed parameters		Time (sec)		
	max % (#)	mean % (#)	mean	min	max
IPC domains					
blocks	100.0% (1.0)	50.0% (0.75)	0.33	0.33	0.34
barman	100.0% (4.0)	58.06% (2.5)	0.98	0.97	0.98
driverlog	66.67% (2.0)	52.78% (1.67)	0.48	0.48	0.49
thoughtful	100.0% (3.0)	26.97% (1.19)	1.72	1.7	1.73
gripper	66.67% (2.0)	50.0% (1.33)	0.25	0.25	0.29
pipesworld (no t.)	53.71% (2.8)	40.19% (2.47)	0.5	0.49	0.53
pipesworld (no s.)	36.67% (4.4)	34.17% (3.62)	0.37	0.35	0.43
pipesworld	43.32% (3.38)	34.31% (2.74)	0.51	0.5	0.6
tpp	66.67% (2.13)	39.52% (2.1)	0.33	0.32	0.36
freecell	66.67% (2.0)	34.71% (1.6)	0.83	0.81	0.88
logistics00	66.67% (2.0)	44.05% (1.4)	0.5	0.49	0.54
rovers	84.17% (5.05)	53.5% (2.27)	0.75	0.72	0.85
satellite	65.28% (1.94)	48.01% (1.34)	0.41	0.4	0.47
visitall	50.0% (1.0)	50.0% (1.0)	0.95	0.17	1.67
depot	66.67% (2.0)	53.33% (2.0)	0.41	0.4	0.41
zenotravel	63.33% (1.9)	37.77% (1.52)	0.41	0.4	0.47
HTG domains					
visitall-3dim	25.0% (1.0)	25.0% (1.0)	0.27	0.26	0.29
visitall-4dim	20.0% (1.0)	20.0% (1.0)	0.35	0.34	0.39
visitall-5dim	16.67% (1.0)	16.67% (1.0)	0.44	0.42	0.47
GED	100.0% (2.0)	38.1% (0.86)	1.14	1.12	1.19
GED-split	100.0% (1.0)	26.19% (0.43)	1.7	1.67	1.91
GED-positional	100.0% (3.0)	100.0% (2.67)	–	–	–
pipesworld (no s.)	39.67% (4.76)	35.9% (3.82)	0.35	0.33	0.39
rovers	100.0% (6.0)	100.0% (4.0)	–	–	–
blocks	100.0% (1.0)	50.0% (0.75)	0.51	0.35	0.79
childsnack_parsize1	100.0% (3.5)	72.71% (2.33)	0.35	0.34	0.39
childsnack_parsize2	100.0% (4.0)	63.89% (2.33)	0.35	0.32	0.38
childsnack_parsize3	100.0% (5.0)	62.05% (2.58)	0.35	0.32	0.38
childsnack_parsize4	100.0% (6.0)	60.83% (2.83)	0.34	0.33	0.39
logistics	50.0% (1.5)	34.03% (1.08)	0.62	0.53	0.81
OS-MIT	100.0% (27.94)	92.09% (10.38)	4.44	4.36	4.67
OS-alkene	80.59% (9.89)	62.72% (6.28)	1.13	1.07	1.18
OS-original	100.0% (28.15)	92.86% (10.43)	4.47	4.38	4.69

Table 1: Summary of seed parameters identified by our approach. Columns 2 & 3 present the maximum & mean of the percent (number) of seed parameters per operator, aggregated over all the problems in that domain. Columns 4, 5, & 6 present the mean, min, and max time taken to identify the seed parameters, respectively.

Domain	problem	# of ground actions ($\|L\|$)	# of reduced action ($\|L'\|$)	reduction ($\|L\| - \|L'\|$)
OS-alkene	p1	3.20E+17	4.49E+14	3.20E+17
OS-alkene	p10	3.41E+12	1.21E+11	3.29E+12
OS-alkene	p11	2.88E+18	1.33E+16	2.87E+18
OS-alkene	p12	18319428180	256608820	18062819360
OS-alkene	p13	18319428180	256608820	18062819360
OS-alkene	p14	2810413424	62656784	2747756640
OS-alkene	p15	2810413424	62656784	2747756640
OS-alkene	p16	1.48E+12	7435419200	1.47E+12
OS-alkene	p17	40551526400	468875264	40082651136
OS-alkene	p18	4.51E+11	18461971584	4.33E+11
OS-alkene	p2	3.39E+11	14094998658	3.25E+11
OS-alkene	p3	2.47E+16	1.15E+14	2.46E+16
OS-alkene	p4	2810413424	62656784	2747756640
OS-alkene	p5	100781250	4406250	96375000
OS-alkene	p6	1.01E+12	31835497386	9.83E+11
OS-alkene	p7	4328521728	78643200	4249878528
OS-alkene	p8	1487318658	34588806	1452729852
OS-alkene	p9	7.19E+16	3.33E+14	7.16E+16
OS-original	prob01	1.79E+29	1.79E+29	9.85E+14
OS-original	prob02	2.23E+21	2.23E+21	3.63E+14
OS-original	prob03	1.41E+25	5.01E+24	9.13E+24
OS-original	prob04	2.31E+24	2.31E+24	1.17E+18
OS-original	prob05	2.70E+37	6.75E+36	2.02E+37
OS-original	prob06	1.76E+17	1.76E+17	43008
OS-original	prob07	2.19E+38	2.19E+38	0
OS-original	prob08	1.61E+40	4.02E+39	1.21E+40
OS-original	prob09	1.52E+46	9.48E+44	1.42E+46
OS-original	prob10	1.47E+24	1.47E+24	8.83E+19
OS-original	prob11	1.18E+24	1.18E+24	1.32E+16
OS-original	prob12	4.35E+40	1.09E+40	3.27E+40
OS-original	prob13	8.53E+24	1.96E+24	6.58E+24
OS-original	prob14	2.50E+25	8.66E+24	1.63E+25
OS-original	prob15	5.18E+26	5.18E+26	1.24E+21
OS-original	prob16	1.36E+23	1.36E+23	0
OS-original	prob17	1.70E+42	1.89E+41	1.51E+42
OS-original	prob18	4.37E+29	4.37E+29	3.58E+23
OS-original	prob19	4.46E+21	4.46E+21	1.01E+17
OS-original	prob20	9.66E+21	9.66E+21	3.82E+14
OS-MIT	p10	2.29E+23	2.22E+23	7.49E+21
OS-MIT	p11	1.00E+20	2.68E+19	7.37E+19
OS-MIT	p12	1.27E+23	1.27E+23	47009759232
OS-MIT	p13	1.87E+23	1.78E+22	1.69E+23
OS-MIT	p14	1.16E+17	1.16E+17	19992000
OS-MIT	p15	1.53E+16	1.53E+16	0
OS-MIT	p16	6.22E+20	6.22E+20	0
OS-MIT	p17	2.54E+21	2.54E+21	0
OS-MIT	p19	8.38E+18	8.38E+18	4.52E+12
OS-MIT	p2	2.72E+18	2.72E+18	1155279872
OS-MIT	p20	5.20E+18	5.20E+18	1000448
OS-MIT	p3	2.10E+14	2.10E+14	0
OS-MIT	p4	4.93E+18	4.93E+18	272260096
OS-MIT	p5	4.15E+35	1.04E+35	3.11E+35
OS-MIT	p6	8.42E+15	8.42E+15	0
OS-MIT	p7	1.04E+18	1.04E+18	6.35E+12
OS-MIT	p8	1.23E+15	1.23E+15	261360
OS-MIT	p9	5.50E+19	5.50E+19	5496832

Table 2: Comparison of action label set sizes with and without reduction on 3 HTG domain: organic-synthesis-alkene, organic-synthesis-original and organic-synthesis-MIT.

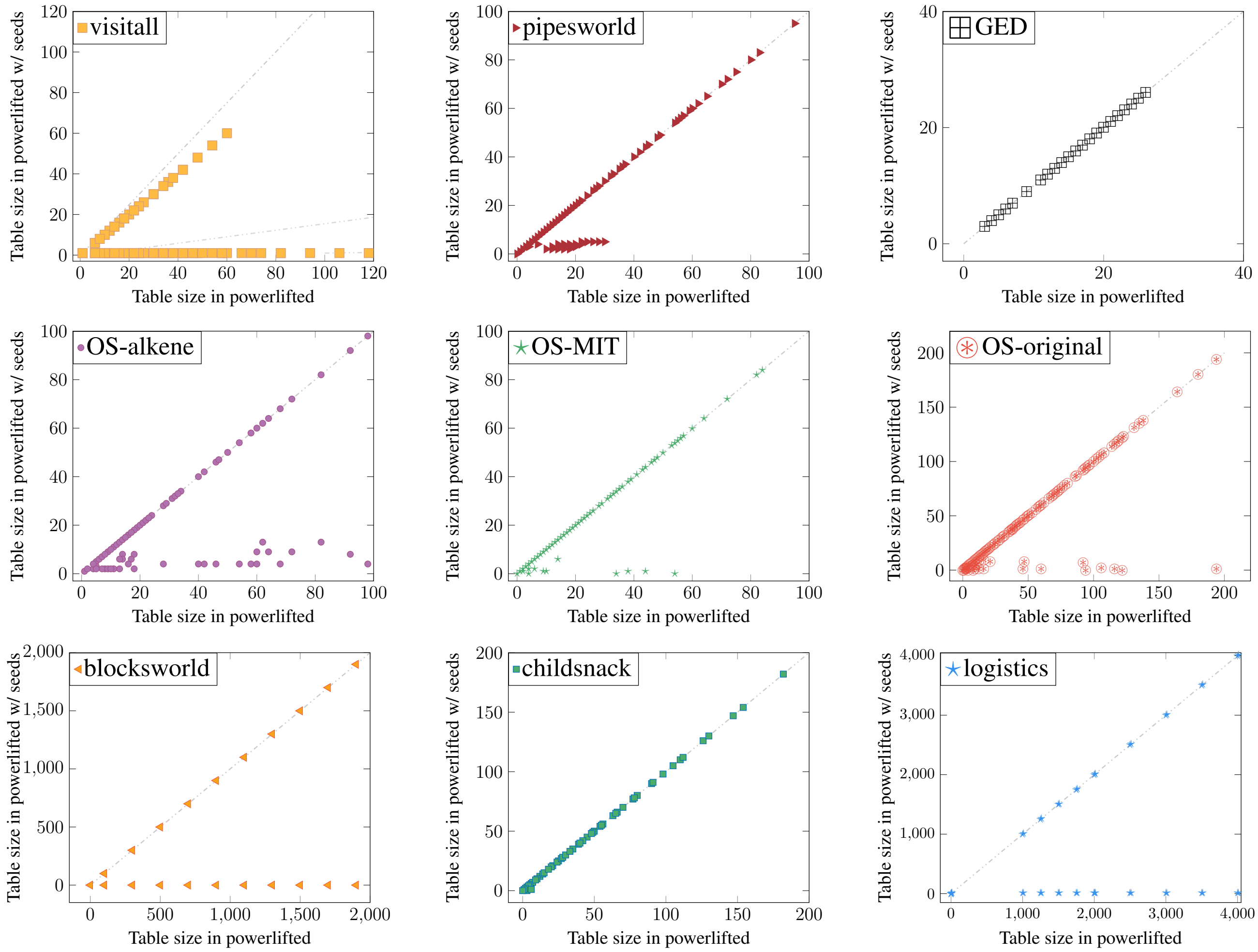
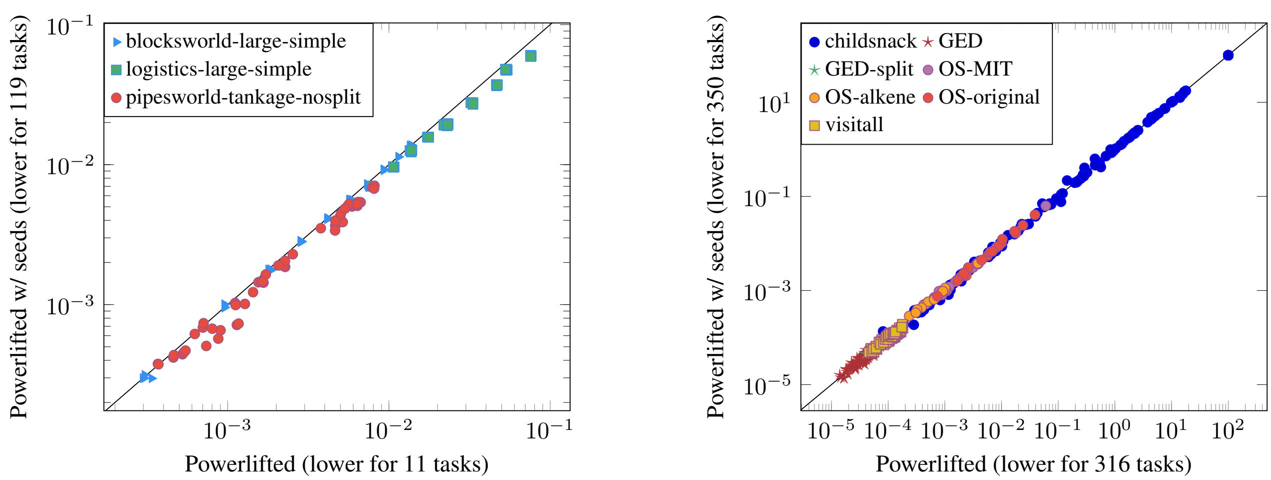


Figure 2: Comparison of table sizes in nine HTG domain before the query is performed. X-axis represents the table size in original powerlifted implementation and Y-axis represents the table size in powerlifted modified to account for the parameter seeds.



(a) (b)

Figure 3: Time of generating applicable actions for the initial state on HTG domains, (a) domains where our approach performs visibly better and (b) domains where our approach is on par with the state-of-the-art.

Hyperparameters	**Values**	**Exception**
Learning Rate	0.003	–
Batch Size	4	16 in Logistics
Input Size	# possible state literals	
Output Size	# Action labels	
Hidden layers	3	–
Hidden units	64	512 in Logistics
Discount	0.95	–
Max Episode Length	100	–

Table 3: Summary of hyperparameters used in RL evaluations.

Domain	Objects	# Action Labels		diff
		Ground ($\lvert L\rvert$)	Reduced ($\lvert L'\rvert$)	
Ferry	3 cars 3 loc.	24	7	17
Gripper	4 balls 2 loc.	36	14	21
Blocks	4 blocks	40	13	27
Logistics	2 pkgs, 2 cities, 2 trucks, 1 airplane	68	20	48

Table 4: Summary of domains used in RL evaluations.