A Comparison of Sampling Strategies for Learning Policies

Farid Musayev[1], Dominik Drexler[1], Daniel Gnad[1,2], Jendrik Seipp[1]

[1]Linköping University, Sweden
[2]Heidelberg University, Germany
{farid.musayev, dominik.drexler, daniel.gnad, jendrik.seipp}@liu.se

Abstract

We address the problem of learning transition classifiers for classical planning domains that generalize across tasks of arbitrary size. Since the size of the induced state spaces typically grows exponentially with the number of objects, it is crucial to sample training data carefully. We systematically investigate strategies for sampling training data from complete state spaces to identify which transitions are most informative for learning. Our results show that goal distance and structural symmetries in states and transitions are key criteria for selecting valuable data, while other sampling heuristics provide only marginal additional gains. Further analysis reveals that, in many domains, transitions extracted solely from optimal plans already suffice for learning robust transition classifiers, with only a few domains requiring more diverse sampling. Using these insights, we construct transition classifiers that achieve state-of-the-art performance on the IPC 2023 benchmark set.

Introduction

Statistical learning has become an increasingly prominent approach to classical planning in recent years. A common objective for learning-based approaches is to train models capable of generalizing across tasks of arbitrary size. State spaces induced by classical planning tasks can be extremely large, often exponential in the number of objects in the task. However, relying solely on small tasks for learning may lead to poor generalization to larger tasks, as the state spaces of small tasks may not capture the important characteristics of larger ones. To address this challenge, different learning approaches try to extract data from large state spaces. A common approach is to extract data from optimal plans (Ferber, Helmert, and Hoffmann 2020; Chen, Trevizan, and Thiébaux 2024; Ståhlberg, Bonet, and Geffner 2022a) generated by traditional planners. Krajňanský et al. (2014) introduce a method for extracting multiple close-to-optimal plans instead of a single one to learn action pruning rules. Hao et al. (2025) propose to generate data using A^* search with admissible heuristics to obtain more states beyond those on optimal plans. Some approaches for learning policies (Ståhlberg, Bonet, and Geffner 2022b; Musayev et al. 2025) sample data from large but complete state spaces expanded using breadth-first search, since the size of these state spaces often makes it infeasible to use all available data for learning. When only a subset of data can be used, the question arises of *how* to effectively sample training data from large state spaces.

In this work, we focus on learning policies represented as transition classifiers (Musayev et al. 2025). We present a study comparing several strategies for sampling training data from complete state spaces. Our strategies rely on four criteria: (1) the distance to the goal; (2) the symmetry of states and transitions; (3) the ratio of "good" to all outgoing transitions in a given state; (4) the cost of making an error in a given state, prioritizing states where making an error incurs extra effort. Finally, we evaluate the effectiveness of sampling transitions from optimal plans.

Our empirical evaluation shows that goal-distance stratification and symmetry detection significantly improve generalization in several domains. Among state-based criteria, the cost of error demonstrates the most consistent benefits, though no single strategy dominates the others across all domains. Our findings also show that transitions from optimal plans can provide sufficient training data for many domains, while others require broader state space coverage for effective learning. Finally, we show that transition classifiers learned with the proposed strategies achieve state-of-the-art performance on the IPC 2023 benchmark set.

Background

Classical Planning. A planning task (or task) is a pair $P = \langle D, I \rangle$ where D is a planning domain that defines a set of first-order predicates $\mathcal{R}$ and action schemas, and I is a task description that consists of a finite set of objects O and two sets of ground atoms describing the initial state and the goal condition. A state s is a set of atoms induced by grounding predicates $\mathcal{R}$ from D with objects from O. An action a is a ground instance of an action schema from D. An action a is applicable in a state s if its preconditions are satisfied in s. A task P induces a state space $\mathcal{S}(P) = \langle S, s_0, G, Succ \rangle$ where S is a finite set of states, $s_0 \in S$ is the initial state, and G is a goal condition that induces a set of goal states $S_G = \{s \in S \mid G \subseteq s\}$. $Succ \subseteq S \times S$ is a set of state pairs $\langle s, s' \rangle$ such that there exists a ground action a applicable in s that yields the successor state s'. The set of outgoing transitions from a state s is $Succ(s) = \{\langle s, s' \rangle \mid \langle s, s' \rangle \in Succ\}$.

A plan is a state trajectory $s_1, s_2, \ldots, s_n$ where $s_1 = s_0$ and $s_n \in S_G$, and for each $1 \leq i < n$ there exists a state pair $\langle s_i, s_{i+1} \rangle \in Succ$. A plan for s is optimal if there is no shorter plan from s to a goal state. $h^*(s)$ is the length of an

optimal plan for s. $h^*(s) = \infty$ if no plan for s exists.

Transition Classifiers. A transition classifier c_Φ is a relation over state pairs evaluated over features Φ. We build on the definitions by Musayev et al. (2025) and represent a transition classifier as a set of rules over the features, each of the form $C \mapsto E$, where C is a set of feature conditions and E is a set of feature effects. A state pair $\langle s, s' \rangle$ is compatible with a rule iff s satisfies all conditions in C and $\langle s, s' \rangle$ satisfies all effects in E. A state pair $\langle s, s' \rangle$ is in c_Φ, or c_Φ-compatible, iff $\langle s, s' \rangle$ is compatible with a rule in c_Φ. A transition $\langle s, s' \rangle$ is labeled as *good* if $h^*(s') < h^*(s)$, *bad* if $h^*(s') \geq h^*(s)$ and *unsolvable* if $h^*(s') = \infty$.

Symmetries. We follow the definitions by Drexler et al. (2024). A relational structure $\mathfrak{A}^s$ for a state s from a task P over objects O and predicates $\mathcal{R}$ consists of a universe $U^s = O$ and interpretations $R^s \subseteq (U^s)^k$ for each predicate R of arity k, where $\langle o_1, \ldots, o_k \rangle \in R^s$ iff $R(o_1, \ldots, o_k)$ is true in s. Two relational structures $\mathfrak{A}^s$ and $\mathfrak{A}^t$ are **isomorphic**, denoted by $\mathfrak{A}^s \sim \mathfrak{A}^t$, iff there exists a bijection $\sigma : U^s \to U^t$ such that for every predicate R of arity k it holds that $\langle o_1, \ldots, o_k \rangle \in R^s$ iff $\langle \sigma(o_1), \ldots, \sigma(o_k) \rangle \in R^t$. Two states s and t are symmetric, denoted by $s \sim t$, iff their relational structures are isomorphic, i.e., $\mathfrak{A}^s \sim \mathfrak{A}^t$. Two transitions $\langle s_1, s_2 \rangle$ and $\langle t_1, t_2 \rangle$ are symmetric, denoted by $\langle s_1, s_2 \rangle \sim \langle t_1, t_2 \rangle$, iff $s_1 \sim t_1$ and $s_2 \sim t_2$.

Sampling Strategies

Given a set of training tasks $\mathcal{P}$ over a planning domain D, our objective is to learn a transition classifier c_Φ over features Φ that generalizes to larger tasks over D. In many domains, the state space grows exponentially with the problem size, limiting the size of tasks we can consider. Consequently, we impose computational limits on the number of states and transitions, as well as the time required to generate state spaces. Since learning from large spaces is computationally challenging, we sample from the generated state spaces, focusing on states and transitions that are most relevant for learning effective transition classifiers. We describe below several criteria for identifying informative samples.

Sampling Transitions vs. Sampling States. A training set $\mathcal{T}$ for learning transition classifiers is a subset of all transitions in the generated state spaces. Musayev et al. (2025) sample transitions uniformly at random from those state spaces, which we refer to as *transition-based sampling*. For each state space S, they generate a set of successors $Succ$ and uniformly sample transitions $\langle s, s' \rangle$ from $Succ$ until reaching a predefined number of samples or exhausting the state space.

In addition to transition-based sampling, we consider sampling states and including all their outgoing transitions, which we call *state-based sampling*. We do so because a learned transition classifier that considers more information about such a state can potentially make more informed decisions.

Symmetries. State spaces may contain an exponential number of symmetric states, leading to many redundant samples in the data (Drexler et al. 2024). To the best of our knowledge, we are the first to study the impact of symmetries on the generalization capabilities of learned models.

During transition-based (resp. state-based) sampling, we exclude a transition (resp. state) from the training data if it is symmetric to one already in the training samples. For state-based sampling, we additionally consider a variant that excludes an outgoing transition from a sampled state if the transition is symmetric to one in the training samples.

Goal Distances. In their work on learning general policies, Ståhlberg, Bonet, and Geffner (2022a) sample states based on the goal distances. Sampling by goal distance decreases the likelihood of sampling symmetric states, as two states with different goal distances are non-symmetric. We empirically evaluate goal-distance-based sampling and aim to strengthen it. First, we stratify the states in each generated state space by their goal distance, by grouping states with the same h^* value. Then, we sample uniformly from each stratum to ensure balanced representation across different distances to the goal. This stratified sampling can be combined with symmetry detection, where we exclude symmetric states or transitions within each stratum.

Difficult States. States with a high number of bad or unsolvable outgoing transitions and a low number of good outgoing transitions can be seen as more important to classify correctly for the transition classifier. Acting randomly in such states would likely select a bad or unsolvable transition simply because they outnumber good ones. We therefore prioritize sampling from these challenging states.

Definition 1 (good-transition ratio)**.** *Consider a state space $\mathcal{S}(P) = \langle S, s_0, G, Succ \rangle$ of a task P. We define the **good-transition ratio** $\rho_+(s)$ for a state $s \in S$ as*

$$\rho_+(s) = \frac{|\{\langle s, s' \rangle \in Succ(s) \mid \langle s, s' \rangle \text{ is good}\}|}{|Succ(s)|}.$$

During sampling, we prioritize states s with a low $\rho_+(s)$.

Bad Decision Cost. We also consider prioritizing states where choosing an incorrect transition results in a large heuristic error. States with high potential heuristic errors can be critical for learning because misclassification at these decision points can lead to significant setbacks in reaching the goal. For example, in domains with unsolvable states, making a bad decision that leads to a dead-end region can be much more detrimental than making a bad decision that only slightly increases the distance to the goal.

Definition 2 (cost of error)**.** *Consider a state space $\mathcal{S}(P) = \langle S, s_0, G, Succ \rangle$ of a task P. We define the **cost of error** $C_{err}(\langle s, s' \rangle)$ **for a transition** $\langle s, s' \rangle \in Succ$ as $C_{err}(\langle s, s' \rangle) = h^*(s') - h^*(s) + 1$, where $C_{err}(\langle s, s' \rangle) = 0$ for good transitions and ∞ for unsolvable transitions. Moreover, we define the **cost of error** $C_{err}(s)$ **for a state** $s \in S$ as*

$$C_{err}(s) = \max_{\langle s, s' \rangle \in Succ(s)} C_{err}(\langle s, s' \rangle).$$

During sampling, we prioritize states s with a high $C_{\text{err}}(s)$.

States and Transitions from Optimal Plans. A common approach to learning value functions is to use only states that appear on optimal plans (Ståhlberg, Bonet, and Geffner 2022a; Chen, Trevizan, and Thiébaux 2024). However, in

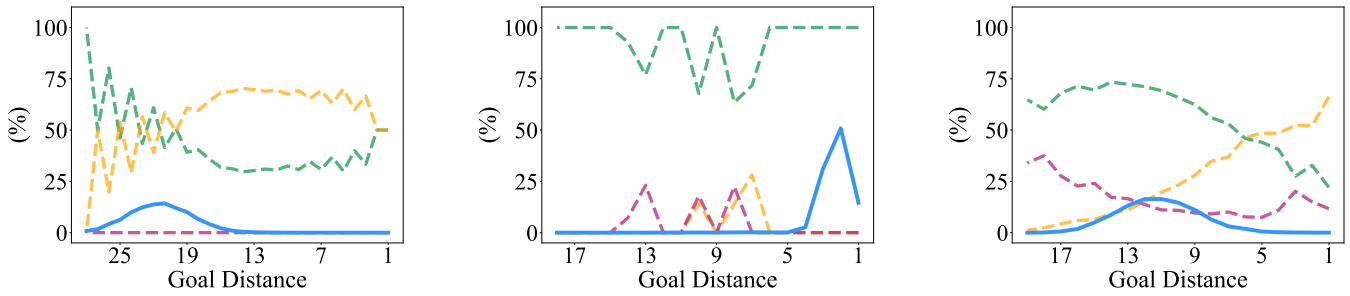

Figure 1: Transition statistics for one large training instance in Blocksworld (left), Spanner (middle) and Childsnack (right). The solid blue line shows the share of all transitions located at each goal distance, while the dashed lines show the proportion of good, bad and unsolvable transitions among the transitions at that distance.

many domains, optimal planning is NP-hard, and hence, optimal general strategies do not exist. In those cases, plan existence might still be tractable, meaning that suboptimal general strategies exist. We argue that learning such strategies might benefit from suboptimal training data. In our final experiment, we therefore compare the effectiveness of sampling from optimal plans to sampling from the entire state space.

Empirical Evaluation

We conduct experiments on the training and testing benchmark sets from the Learning Track of the International Planning Competition (IPC) 2023. From the training set, we consider state spaces fully expandable within a 1-minute time limit and with at most 10 million states. For the state spaces of each domain, we apply the sampling strategies discussed above. When combining multiple criteria, we first stratify states by goal distance, then apply symmetry detection and prioritize states within each stratum using ρ_+ or C_{err}. In each case, we limit the overall training data to 100 000 sampled transitions per domain. Moreover, we use a 10-minute time limit per state space for strategies that involve symmetry detection. We compare the following sampling strategies:

- **Transition-based sampling:**
  - $\sigma(T)$: random transition sampling
  - $\sigma(T_{\text{g}})$: goal-distance-stratified transition sampling
  - $\sigma(\widetilde{T}_{\text{g}})$: goal-distance-stratified transition sampling with symmetry detection
- **State-based sampling** (all use goal-distance stratification with symmetry detection):
  - $\sigma(\widetilde{S}_{\text{g}})$: uniform sampling over states
  - $\sigma(\widetilde{S}_{\text{g}}, \rho_+)$: prioritizing states with lower ρ_+
  - $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$: prioritizing states with higher C_{err}
- **Optimal plans vs. complete state spaces** (both with symmetry detection):
  - $\sigma(\widetilde{T}_*)$: transitions from optimal plans
  - $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}}, |*|)$: $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$ with training data size matched to $\sigma(\widetilde{T}_*)$

For each domain, we (1) obtain training data by running a sampling strategy with five random seeds, (2) generate and evaluate description logic features on the obtained data, (3) train five decision tree models, (4) convert them into policies following Musayev et al. (2025), (5) run them in our planner and report average results. For training a model, we follow the IPC 2023 setup and use 1 CPU core, 24 hours and 32 GiB per domain. We run policies in a lookahead search. Starting from the initial state, we iteratively follow the policy, store all successors along the trajectory in the open list, and, whenever the policy fails to predict a good transition, select a state from the open list sorted by predicted unsolvability (0 or 1) and g-value. We use a 30-minute time limit and an 8 GiB memory limit per test instance. We implemented our planner within the Fast Downward planning framework (Helmert 2006). We note that for tasks from five domains—Childsnack (24), Ferry (14), Rovers (14), Sokoban (4) and Transport (24)—the Fast Downward translator fails to ground instances within the 8 GiB memory limit, reducing the test set size. All code and data will be published upon acceptance. The code is already included in the supplementary material.

Is random transition sampling enough? The coverage data in Table 1 shows that $\sigma(T)$ is outperformed by all other sampling strategies. To understand this weakness, Figure 1 visualizes the distribution of transitions per goal distance in Blocksworld, Spanner and Childsnack. We consider one of the large training instances in each domain, noting that we observe similar distributions across other instances. In Blocksworld, transitions concentrate in the first half of the goal-distance range, leading random sampling to miss critical transitions closer to the goal. In Spanner, transitions are highly imbalanced across all goal distances, with good transitions dominating bad and unsolvable ones. A critical region is where the agent must choose between picking up spanners or moving towards the goal. If the agent fails to pick up spanners early, it cannot complete the goal later. If random sampling undersamples these critical regions, it can lead to poor model performance. Notably, stratification by goal distance degrades performance in Childsnack. This counterintuitive result suggests that the uniform distribution across goal-distance strata may also oversample less informative regions while undersampling critical transitions. We address this issue further with the C_{err} criterion.

	Baseline		Transition-based			State-based			Optimal plans	
Domain	LAMA-f	$h^{\text{WLF}}_{\text{GPR}}$	$\sigma(T)$	$\sigma(T_{\text{g}})$	$\sigma(\widetilde{T}_{\text{g}})$	$\sigma(\widetilde{S}_{\text{g}})$	$\sigma(\widetilde{S}_{\text{g}}, \rho_{+})$	$\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$	$\sigma(\widetilde{T}_{*})$	$\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}}, \lvert*\rvert)$
Blocksworld (90)	60	**72**	20±9	51±16	47±13	66±19	69±8	65±10	68±2	65±9
Childsnack (66)	35	31	61±3	43±19	55±12	49±17	43±17	**62±1**	32±0	28±12
Ferry (76)	68	**76**	74±1	75±1	75±0	75±1	75±1	75±1	67±0	75±1
Floortile (90)	11	2	37±12	31±1	44±2	35±1	36±1	36±1	**60±0**	49±9
Miconic (90)	**90**	**90**	**90±0**	**90±0**	**90±0**	**90±0**	**90±0**	**90±0**	**90±0**	**90±0**
Rovers (76)	**68**	37	50±5	50±6	51±5	48±6	50±5	51±5	52±3	51±6
Satellite (90)	**89**	48	55±3	52±3	55±4	54±5	53±5	55±3	52±7	49±7
Sokoban (86)	**40**	38	22±1	22±2	23±1	22±2	22±1	25±1	20±0	20±3
Spanner (90)	30	**73**	48±15	64±0	63±0	65±0	65±0	65±0	64±0	64±1
Transport (66)	**66**	28	**66±0**	**66±0**	**66±0**	65±1	65±1	65±1	**66±0**	62±6
Sum (820)	557	495	523±49	544±48	569±37	569±52	568±39	**589±23**	571±12	553±54

Table 1: Average coverage ± standard deviation over test tasks per domain and model learned with different sampling strategies.

Does symmetry detection help? Comparing $\sigma(T_{\text{g}})$ and $\sigma(\widetilde{T}_{\text{g}})$ reveals that symmetry detection impacts performance in several domains. As Drexler et al. (2024) have shown, Childsnack and Spanner possess symmetries and benefit significantly from symmetry detection in terms of data efficiency. We observe a significant reduction in the training data for Spanner, from 100 000 to around 7 000 transitions, and for Childsnack, from 100 000 to around 27 000 transitions. From a generalization perspective, we observe an improvement in Childsnack (from 43 to 55 solved tasks) but do not observe a significant change in Spanner (65 vs. 64 tasks). For Childsnack, we observe that learned decision trees have a different structure (e.g., different selected features and tree shapes), which can explain performance variance. Floortile also shows a notable improvement (from 31 to 44 tasks) with symmetry detection. Excluding symmetric transitions results in a more diverse set of transitions, improving the quality of generated features. Other domains, with fewer symmetries, show minor to no improvements in generalization.

Is it better to sample transitions or states? We compare $\sigma(\widetilde{T}_{\text{g}})$ and $\sigma(\widetilde{S}_{\text{g}})$, both using goal-distance stratification, and observe no clear advantage of one approach over the other across all domains. Blocksworld and Spanner show improvements with state-based sampling, though Blocksworld exhibits high variance. Childsnack and Floortile show noticeable degradation, while other domains perform comparably.

Do different state-based strategies matter? Comparing different state-based strategies against $\sigma(\widetilde{S}_{\text{g}})$ reveals that most criteria perform comparably across domains. Only two domains—Blocksworld and Childsnack—show notable differences based on the chosen criteria. $\sigma(\widetilde{S}_{\text{g}}, \rho_{+})$ achieves the best results in Blocksworld, while $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$ demonstrates superior performance in Childsnack. The effectiveness of $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$ in Childsnack stems from its ability to prioritize critical states where both good and unsolvable transitions are prevalent. We observe that decision trees learned in Blocksworld with different training sets exhibit significant variation in the features they select and the shape of the trees. We leave further investigation of model optimization for future work. Overall, the $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$ strategy provides competitive performance across both domains while achieving the highest total coverage with the lowest variance.

Is using optimal plans sufficient for learning? Our final experiment evaluates the $\sigma(\widetilde{T}_{*})$ sampling strategy, which uses only transitions from optimal plans. To make a fair comparison, we also consider $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}}, \lvert*\rvert)$ that uses the same amount of training data as $\sigma(\widetilde{T}_{*})$. Results in Table 1 show that $\sigma(\widetilde{T}_{*})$ solves more tasks in total than $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}}, \lvert*\rvert)$ while also achieving the lowest variance across domains. This indicates that for many domains, using transitions from optimal plans is sufficient for learning. Notably, Floortile, one of the most challenging domains for learning-based approaches, shows a significant improvement when using $\sigma(\widetilde{T}_{*})$, suggesting that optimal plans (with only 2 600 non-symmetric transitions) effectively capture critical information in this domain. Similar reductions in training data across other domains lead to simpler decision trees that maintain or improve generalization while enabling lower runtime. However, when we compare $\sigma(\widetilde{T}_{*})$ with our best sampling strategy $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$, we observe a performance drop (from 589 to 571 in total coverage). This suggests that for some domains, for example, Childsnack and Ferry, relying solely on optimal plans may not provide sufficient diversity in the training data.

Comparison to the state of the art. We compare our best sampling strategy $\sigma(\widetilde{S}_{\text{g}}, C_{\text{err}})$ against two state-of-the-art approaches: the first iteration of LAMA (Richter and Westphal 2010), a traditional planner that has been a strong baseline in satisficing planning for many years, and WL-GOOSE (Chen, Trevizan, and Thiébaux 2024), a planner that exclusively relies on a heuristic function learned using WL features and Gaussian Process Regression (GPR). Our approach outperforms both baselines in 5 out of 10 domains, matches LAMA in Miconic and Transport, and matches WL-GOOSE in Ferry and Miconic. Detecting symmetries *during search* could further improve our planner, as observed by a recent study (Bai, Thiébaux, and Trevizan 2025). Based on our results and previous studies, Rovers, Satellite and Sokoban emerge as the most challenging domains for learning-based approaches in

the IPC 2023 benchmark set. None of these domains benefit from our sampling strategies. In particular, we observe a lack of expressiveness in the features for Rovers and Sokoban, leading to poor model performance. We leave further investigation of feature engineering for future work.

Conclusions

We presented a comprehensive empirical study of sampling strategies for learning policies in classical planning. Our experiments on the IPC 2023 benchmarks demonstrate that different criteria can benefit learning in different ways. In particular, goal-distance stratification is important, as random sampling may struggle to capture critical regions. Symmetry detection significantly improves data efficiency in symmetric domains while maintaining or improving generalization. Among state-based criteria, the cost of error shows the most consistent improvements, although no single strategy dominates the others across all domains. Our results also reveal that using transitions from optimal plans suffices for learning robust models in many domains with the exception of a few cases where data diversity is crucial. Overall, our sampling strategies enable transition classifiers to achieve competitive results when compared to state-of-the-art planners.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) partially funded by the Swedish Research Council through grant agreement no. 2022-06725.

References

Bai, Y.; Thiébaux, S.; and Trevizan, F. 2025. Learning Efficiency Meets Symmetry Breaking. In Lipovetzky, N.; Sardina, S.; Harabor, D.; and Ramirez, M., eds., *Proceedings of the Thirty-Fifth International Conference on Automated Planning and Scheduling (ICAPS 2025)*, 154–159. AAAI Press.

Chen, D. Z.; Trevizan, F.; and Thiébaux, S. 2024. Return to Tradition: Learning Reliable Heuristics with Classical Machine Learning. In Bernardini, S.; and Muise, C., eds., *Proceedings of the Thirty-Fourth International Conference on Automated Planning and Scheduling (ICAPS 2024)*, 68–76. AAAI Press.

Drexler, D.; Ståhlberg, S.; Bonet, B.; and Geffner, H. 2024. Symmetries and Expressive Requirements for Learning General Policies. In Marquis, P.; Ortiz, M.; and Pagnucco, M., eds., *Proceedings of the Twenty-First International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*. IJCAI Organization.

Ferber, P.; Helmert, M.; and Hoffmann, J. 2020. Neural Network Heuristics for Classical Planning: A Study of Hyperparameter Space. In De Giacomo, G., ed., *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI 2020)*, 2346–2353. IOS Press.

Hao, M.; Chen, D. Z.; Trevizan, F.; and Thiébaux, S. 2025. Effective Data Generation and Feature Selection in Learning for Planning. In Lynce, I.; and Murano, A., eds., *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI 2025)*, 4969–4976. IOS Press.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26: 191–246.

Krajňanský, M.; Hoffmann, J.; Buffet, O.; and Fern, A. 2014. Learning Pruning Rules for Heuristic Search Planning. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, 483–488. IOS Press.

Musayev, F.; Drexler, D.; Gnad, D.; and Seipp, J. 2025. Combining Heuristics and Transition Classifiers in Classical Planning. In Lynce, I.; and Murano, A., eds., *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI 2025)*, 4694–4701. IOS Press.

Richter, S.; and Westphal, M. 2010. The LAMA Planner: Guiding Cost-Based Anytime Planning with Landmarks. *Journal of Artificial Intelligence Research*, 39: 127–177.

Ståhlberg, S.; Bonet, B.; and Geffner, H. 2022a. Learning General Optimal Policies with Graph Neural Networks: Expressive Power, Transparency, and Limits. In Thiébaux, S.; and Yeoh, W., eds., *Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling (ICAPS 2022)*, 629–637. AAAI Press.

Ståhlberg, S.; Bonet, B.; and Geffner, H. 2022b. Learning Generalized Policies without Supervision Using GNNs. In *ICAPS 2022 Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)*.