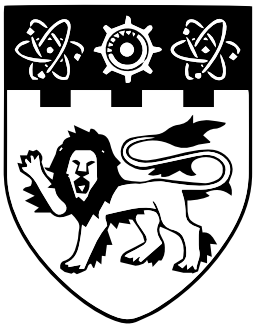

NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Inverse Machine Learning for Multi-criteria Decision Making

Tan Chin Sheng

College of Computing and Data Science

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2024

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

23/07/2024

.....

Date

.....

Tan Chin Sheng

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

23/07/2024

.....
Date

.....
Assoc Prof Lam Siew-Kei

Authorship Attribution Statement

This thesis contains material from 3 papers published and 1 papers submitted in the following peer-reviewed conference and journal which I am listed as an author.

Part of Chapter 4 is published as "Tan, C.S., Gupta, A. and Xu, C., 2022, December. Are Two Heads Always Better Than One? Human-AI Complementarity in Multi-criteria Order Planning. In 2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 0939-0943). IEEE.".

The contributions of the co-authors are as follows:

- Dr. Gupta and I shared our initial thoughts and direction for this work. The core aim of this research was to explore how human decision-makers and optimization (AI) algorithms can work together to enhance the convergence performance of multi-objective optimization problems. This was achieved by leveraging on the emerging transfer optimization methods within the proposed Human-AI complementarity framework.

- I conducted the experimental studies. Dr. Gupta and analyzed the numerical results.

- I prepared the manuscript drafts. The manuscript was revised by Dr. Gupta, with Dr. Xu providing valuable industry-related advice for improvement.

Chapter 5 is published as "Tan, C.S., Gupta, A., Ong, Y.S., Pratama, M., Tan, P.S. and Lam, S.K., 2023. Pareto optimization with small data by learning across common objective spaces. Scientific Reports, 13(1), p.7842.".

The contributions of the co-authors are as follows:

- Dr. Gupta, Prof Ong and I shared the novel concept of inverse transfer learning across common objective spaces.

- Dr. Gupta and I co-designed the methodology for multi-source inverse transfer learning that maximally utilize experiential source tasks to improve Pareto set learning performance in the target optimization task.
- I implemented the algorithm for the underlying methodology, conducted the experimental studies and analyzed the numerical results. Dr Gupta provided further suggestions to improve my experimental designs.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Gupta, with Assoc. Prof Pratama, Dr. Tan and Assoc. Prof Lam providing their feedback for improvement.

Chapter 6 is published as "Tan, C.S., Gupta, A., Ong, Y.S., Lam, S.K., Pratama, M. and Tan, P.S., 2024, December. Pareto Set Representation Learning with Application to Multi-criteria Order Optimization. In 2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 0947-0951). IEEE.".
The contributions of the co-authors are as follows:

- Dr. Gupta, Prof Ong and I shared the novel idea of Pareto set representation learning for large-scale multi-objective optimization problems.
- I designed and implemented the algorithm for Pareto set representation learning, conducted the experimental studies and analyzed the numerical results. Dr Gupta provided further suggestions to improve my experimental designs.
- I prepared the manuscript drafts. The manuscript was revised by Dr. Gupta, with Assoc. Prof Pratama, Dr. Tan and Assoc. Prof Lam providing their feedback for improvement.

Chapter 7 is submitted as "Tan, C.S., Gupta, A., Lam, S.K, Pratama, M. and Tan, P.S., 2024. Monte Carlo walk-pull for inverse modeling of highly nonlinear Pareto sets. IEEE Transactions on Emerging Topics in Computational Intelligence. (Pending review result)".
The contributions of the co-authors are as follows:

- Dr. Gupta and I shared our initial thoughts and direction for this work. This particular work was envisioned to pioneer the idea on Pareto data augmentation using the Monte Carlo walk-pull process, with the ultimate goal of enhancing the predictive performance of non-linear Pareto set learning.

- I designed and implemented the algorithms related to Pareto data augmentation and Pareto set learning, conducted the experimental studies and analyzed the numerical results. Dr Gupta provided further suggestions to improve my experimental designs and analysis.

- I prepared manuscript drafts and revised it based on the feedback from Dr. Gupta, Assoc. Prof Pratama, Dr. Tan and Assoc. Prof Lam.

23/07/2024

....

Date Tan Chin Sheng

Acknowledgements

The pursuit of a part-time PhD degree is like running a marathon, requiring both internal and external support. For this, I would like to thank my family and friends for their unconditional support and understanding, especially during times when I could not be around with them.

I would also like to thank my supervisors and Dr. Gupta for being at the marathon site with me, ensuring I always stayed on track and headed in the right direction. Dr. Tan, both my boss and supervisor, consistently checked on me, ensuring I maintained my pace and made good progress in both my career and PhD studies. Assoc. Prof. Pratama, thank you for agreeing to supervise me and providing excellent technical guidance even from overseas. For the second half of this marathon, I am grateful to have had Assoc. Prof. Lam and Dr. Gupta join me in this race. Assoc. Prof. Lam, your support, trust, and invaluable feedback have been instrumental in my research work. Dr. Gupta, your wisdom, mentorship, and the beer sessions we shared have kept me strong in the final stretch. At times, it felt like you were running alongside me, ensuring I never dropped out of the race.

Lastly, I would like to thank my younger self for having the courage to sign up for this marathon without fully knowing what I was getting into, for persevering throughout the journey, and for completing this thesis.

Table of Contents

Abstract

Multi-criteria decision making involves solving a multi-objective optimization problem (MOP) and selecting an optimal trade-off solution from the Pareto set (PS) based on preferred criteria/ objectives. The decision maker (DM) can articulate her preferences before, during, or after solving the MOP. However, specifying preferences *a priori* without knowledge of the Pareto data can be challenging. As an alternative, the DM can interactively refine her preferences during optimization, but this process can be cognitively demanding, especially when the number of query increases. A third approach involves solving the MOP to acquire the Pareto data first, allowing the DM to select a solution *a posteriori* based on her preferences. However, discovering solutions that lie on the PS manifold and satisfying the DM's preferences is often unlikely, particularly in domains where evaluation data is costly.

To overcome the limited decision choices in the acquired PS, an inverse machine learning model can be invoked to map arbitrary preferences in the objective space to new solutions in the PS. The accuracy of this inverse model, however, is constrained by the inherently small training data available in expensive optimization domains. To alleviate this small data challenge, this thesis introduces a pioneering study on *multi-source inverse transfer learning*. The proposed method leverages experiential *source* tasks to enhance PS learning in the *target* MOP task, uniquely facilitating information transfer between heterogeneous source-target pairs through common objective spaces.

Additionally, to mitigate the curse of dimensionality in the decision space, the concept of *PS representation learning* is introduced. This approach reduces the problem to its smallest possible dimensions while accurately capturing the Pareto optima. A denoising autoencoder is utilized to discover a compressed latent representation of a sparsely populated PS, leveraging its bottleneck architecture. This representation then serves to create compact inverse models, mapping DM's preferences in the objective space to the dimensionally reduced PS.

The final part of the thesis explores scenarios where source task is absent for inverse transfer learning. It presents the novel idea of *Monte Carlo walk-pull* as a means of PS augmentation. Techniques are proposed for the first time to couple the walk-pull process with an inverse machine learner, enabling accurate modeling of the optima in problems with highly non-linear PS shapes.

All three proposed approaches have been empirically tested on benchmarks MOPs and real-world MOPs based on an industrial supply chain planning problem and a high-fidelity multidisciplinary simulation data of composite materials manufacturing processes. The results are promising, showcasing significant improvements in the predictive accuracy and Pareto approximation capacity of PS learning. With such accurate inverse models, a future of on-demand Human-AI interaction facilitating multi-objective decisions can be envisioned.

Keywords: Inverse Machine Learning, Pareto Set Learning, Multi-source Inverse Transfer Learning, Pareto Set Augmentation, Multi-criteria Decision Making.

List of Figures

List of Tables

Chapter 1

Introduction

Decision making for real-world problems, such as engineering [1,2], economics [3, 4], logistics systems planning [5, 6], manufacturing operations optimization [7, 8] is typically complex to solve and consists of multiple conflicting criteria/ objectives. It involves optimization and selection of an optimal solution based on the preferred trade-offs among different criteria. The decision maker (DM) in this case, can provide her preferences among these conflicting criteria before, during, or after the course of optimization. The sequence of preference articulation can in turn drive the formulation and solver choice for the optimization problem, introducing respective research challenges.

1.1 Multi-criteria Decision Making and its Challenges

The involvement of the DM at various stages of the optimization process is as shown in Fig. 1.1. In the first scenario, the DM is involved prior to the initiation of the optimization process, where her preferences among the conflicting objectives is first make known. Some common methods to model the DM preferences include representing it as goals [9, 10], preference weights [11, 12] and utility functions [13, 14] with the ultimate aim of simplifying the MOP either into a single-objective or constraining its search direction. The simplified problem can then be efficiently solved, returning a

single or subset of optimal trade-off solutions that satisfy the *a priori* articulated preference. However, it forces the DM to state her preferences among the conflicting objectives without truly understanding the underlying comprise made in terms of the actual objective values. As a consequence, the initial preference articulated may not be a good representative and can result in changing preferences after being presented with the optimal objective values, requiring multiple re-runs of the optimization process.

One potential remedy is to perform interactive search [15, 16] where in this scenario, the DM is allow to refine her preferences during the course of optimization. It provides the freedom for the DM to fine-tune her preferences according to the objective values of the current best-performing solution obtained in the midst of the optimization process. With the updated preferences, the interactive methods then change the search direction accordingly in the remaining optimization process. However, this give raise to a new challenge of having to determine the optimal number of queries that should be made to the DM during the course of optimization. As an insufficient amount of queries will result in an inaccurate preference model, while on the other hand, a significant amount will cause unnecessary fatigue to the DM. Moreover, the optimal number of queries to be made is highly dependent on individual DM and increases rapidly with the number of objectives.

The final scenario involves directly solving the original MOP to obtain a diverse set of optimal trade-off solutions, where no objective can be improved further without worsening another. This set of solutions constitutes the Pareto Set (PS) in the decision space, with corresponding objective values forming the Pareto Front (PF). Solving the MOP provides a comprehensive view of all possible trade-offs in the objective space, allowing the DM to select a solution *a posteriori* based on her preferences. However, this approach increases the complexity of the optimization process and requires specific solvers designed for MOPs. These solvers include exact methods [17–19] and approximate sampling-based methods [20–22], which typically produce

discrete representations of possibly continuous PFs. A common procedure [23] is to scalarize a MOP with different weight combinations, decomposing it into a set of single-objective optimization sub-problems that are solved jointly. Alternatively, approaches such as [24, 25] evolve populations of solutions toward diverse regions of the PF simultaneously, without explicit problem decomposition, and are widely adopted in practice.

However, in most real-world cases, achieving good coverage of the PF for *a posteriori* decision making requires a large number of solution evaluations, which can be prohibitively expensive. This is particularly true for MOPs with objective functions that rely on time-consuming computer simulations or costly real-world procedures, where each evaluation is at a premium. Consequently, the Pareto optimal solutions obtained from a single run of any MOP solver are typically limited in quantity and rarely align with the sub-regions of the PF preferred by DMs.

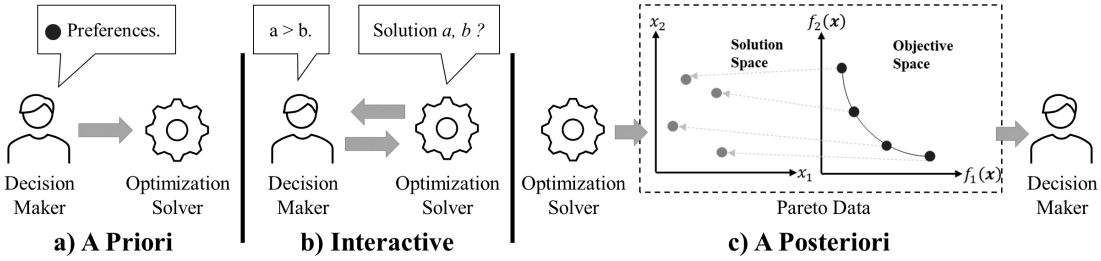

Figure 1.1: Different decision making scenarios involving the DM to articulate her preferences (a) *a priori*, (b) interactively or (c) *a posteriori* of optimization.

A promising approach to address the mismatch between acquired and preferred points along the PF is to initiate PS learning. As illustrated in Fig. 1.2, this involves training an inverse machine learning model to map points from the PF to the decision space [26], enabling the generation of new solutions corresponding to arbitrary, unexplored sub-regions of the PF. This capability paves the way for seamless Human-AI interaction in multi-criteria decision making, where solutions can be generated on-demand based on the postponed preferences of DMs. However, the accuracy of the inverse model

depends on the quantity of acquired training data, which is inherently limited in expensive optimization domains. This challenge is further compounded by problems with highly non-linear PS, requiring a larger volume of Pareto data to adequately represent the intricate geometry.

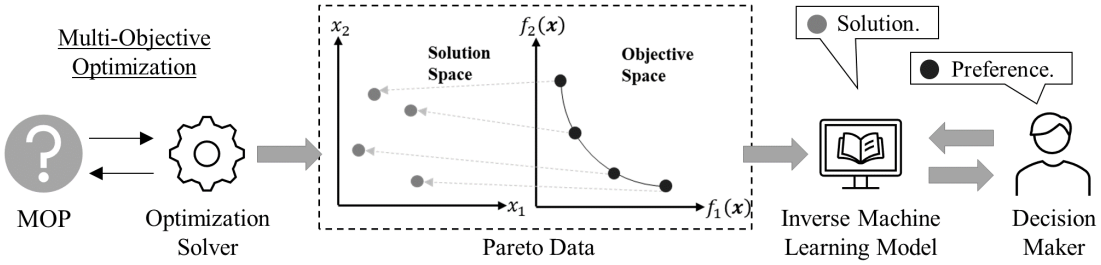

Figure 1.2: Inverse machine learning model for *a posterior* decision making.

1.2 Research Objectives

This thesis is motivated by open research issues in multi-criteria decision making discussed in the previous section. Particularly, it focuses on enhancing the performance of inverse machine learning models to facilitate *a posteriori* decision making in expensive and complex problems characterized by small Pareto data and highly non-linear PS. The research objectives are stated as follows.

1. Investigate into existing machine learning techniques for enhancing the process of multi-criteria decision making to unveil a comprehensive research landscape while identifying open research issues. Subsequently, conduct a background study to illustrate the concept of incorporating machine learning models to facilitate Human-AI multi-criteria decision making.

2. Pioneer the study of *multi-source inverse transfer learning* for *a posteriori* decision making. Develop a novel method that leverages experiential source tasks to enhance PS learning in the target optimization task. Through this unique inverse setting, it will unify common ob-

jective spaces for enabling information transfer between heterogeneous source-target pairs. Additionally, demonstrate the efficacy of the proposed approach in improving predictive accuracy and PF approximation capacity through experimental tests based on benchmark functions and high-fidelity simulation data from composite materials manufacturing processes.

3. Improve the computational efficiency of PS learning in high-dimensional decision spaces by introducing the concept of *PS representation learning*. This concept aims to reduce the problem to its smallest possible dimensions while accurately capturing Pareto-optima. Next, verify the concept with a denoising autoencoder to discover a compressed latent representation of a sparsely populated PS through its unique bottleneck architecture. In addition, assess the potential of this representation to create compact inverse models, mapping points from the PF in objective space to the dimensionally reduced PS in decision space. This is achieved by empirically validating the method on benchmark problems and an industrial supply chain planning problem to showcase its effectiveness in reducing the dimensionality of the PS and gains in PF approximation capacity.

4. Enhance the performance of inverse machine learning model in the absence of source data by presenting a new idea of *Monte Carlo walk-pull* as a means of Pareto data augmentation. Propose techniques to couple the walk-pull process with an inverse machine learner to accurately map points along the front to the PS in decision space. Investigate the ability of the proposed method to accurately model the optima of problems with highly non-linear PS shapes with test results to demonstrate its improvement in the predictive accuracy and PF approximation capacity.

5. Finally, conclude the research works conducted on inverse machine learning for *a posteriori* decision making under different scenarios. In addition, identify future research work for enhancing the performance of PS

learning for *a posteriori* multi-criteria decision making and its potential to integrate with existing multi-objective optimization algorithms.

1.3 Thesis Outline

Figure 1.3 shows the organization of the remainder thesis with the following chapters.

- Chapter 2 provides a comprehensive literature review on the application of machine learning approaches for multi-criteria optimization and decision making. It focuses on preference articulation in *a priori*, *interactive* and *a posterior* stages of the optimization process. Based on the review conducted, open research challenges are identified to guide the research direction of the thesis.

- Chapter 3 presents preliminaries on multi-objective optimization and inverse machine learning for multi-criteria decision making to enable on-demand mapping of DM preferences in the objective to the decision space.

- Chapter 4 conducts a background work on the application of machine learning models to facilitate Human-AI decision making, serving as a feasibility study and exemplifies the impact of the research conducted in this thesis.

- Chapter 5 proposes a new approach for *multi-source inverted transfer learning* to overcome the limited Pareto data. Under this inverse setup, it uniquely enabled information transfers between heterogeneous source-target pairs through the unification of the common objective spaces. Thereafter, a rigorous empirical analysis of the method on benchmark MOPs, with 4-D to 7-D objective spaces, and on a real-word composite manufacturing use-case is conducted.

- Chapter 6 provides a computationally efficient solution to address the high-dimensional PS learning. Concept of *PS representation learning* is introduced to reduce the problem to its minimal dimensions required for accurately capturing the Pareto optima. Furthermore, to illustrate the concept, a denoising autoencoder is invoked to discover a compressed latent representation of a sparsely populated PS to enable the creation of compact inverse models. Empirical tests based on benchmark problems and an industrial supply chain planning problem are then conducted.

- Chapter 7 put forward a different approach in handling limited Pareto data with the absence of source data. A novel idea of *Monte Carlo walk-pull* for Pareto data augmentation is proposed. Moreover, techniques to couple the walk-pull process with an inverse machine learner trained to map points along the front to the PS in decision space are presented for the first time. Thereafter, its performance is validated on multi-objective functions with nonlinear PS shape to highlight the robustness of the proposed approach.

- Chapter 8 concludes the thesis and discusses future research outlooks for these methods.

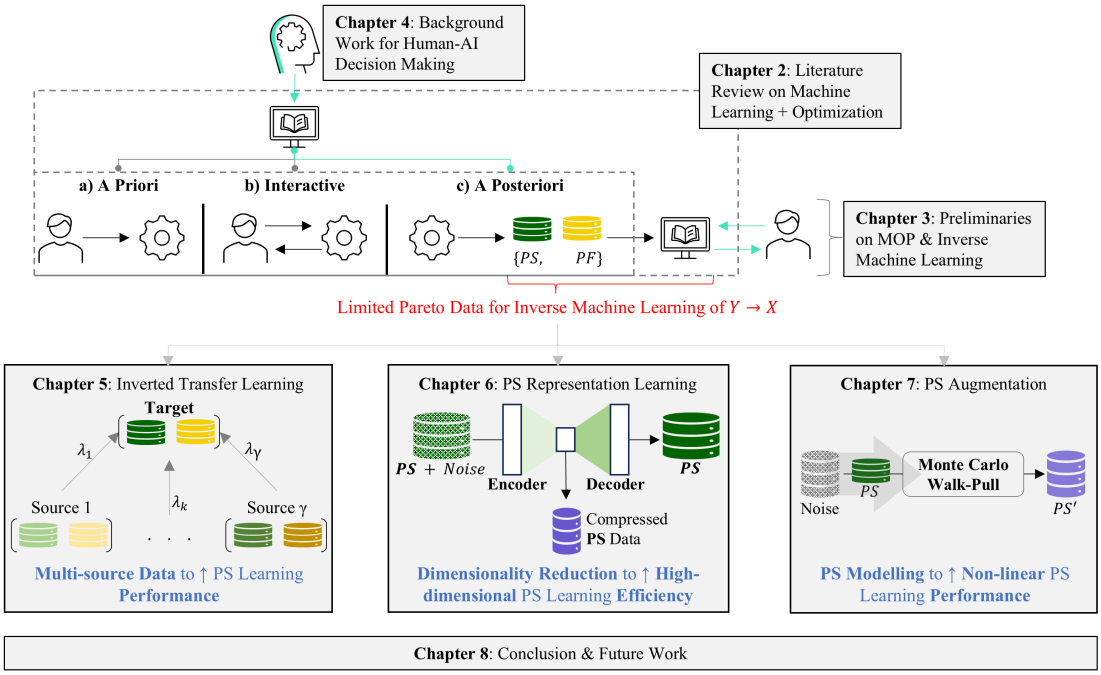

Figure 1.3: Organization of the key chapters in this thesis.

Chapter 2

Literature Review

This chapter provides a systematic review of existing approaches that facilitate multi-criteria optimization and decision making. The approaches are organized and classified based on the sequence of preference articulation, as shown in Fig. 2.1.

The first section reviews works related to preference-guided optimization, where preferences among solutions are elicited before or interactively during the course of optimization. This class of works aims to simplify and focus the search on the subregion of the PF based on known or learned preferences.

The second section focuses on two main applications of machine learning within the multi-objective optimization process, which are building a forward model for rapid function evaluation and unsupervised PS learning. This research stream aims to develop optimization algorithms leveraging machine learning techniques to efficiently solve expensive MOPs. The goal is to obtain a well-converged PF representation, enabling DMs to select a solution *a posteriori* based on their preferences.

The final section reviews inverse machine models for PS learning based on current Pareto data acquired either during[1] or after the multi-objective optimization process. This research stream aims to enhance the density of the approximated PF and enable the on-demand generation of new solutions

[1]Note that inverse machine learning models have also been applied during the optimization process to enhance convergence performance. However, the discussion is deferred to this final section to compare them with inverse models developed for *a posteriori decision making*.

based on the DM postponed preferences by training an inverse model.

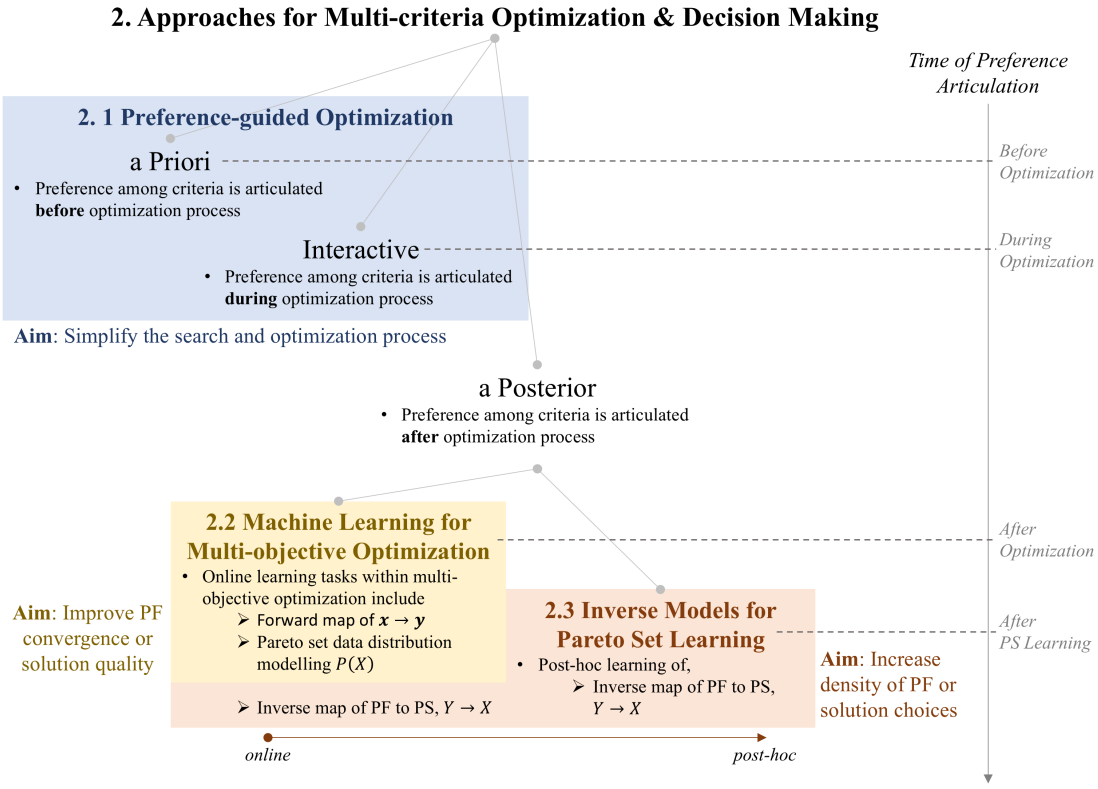

Figure 2.1: Organization of literature review on existing approaches for multi-criteria optimization and decision making.

2.1 Preference-guided Optimization

In multi-criteria optimization and decision making, the aim is to compute a set of Pareto optimal solutions with at least one of its solutions satisfying the DM preference. A potential approach to increase the likelihood of preference alignment under limited computational resources is preference-guided optimization, which models the DM preferences and incorporate it into the search process.

The DM preferences can be articulated *a priori* or interactively during the course of optimization. For *a priori optimization*, preferences are explicitly stated in the forms of goals or weights and are fixed during the search process. The goals are defined by asking the DM to state her ideal objective

values in the form of a m-dimensional vector with m as the number of objectives. This vector then serves as a reference point for comparing the objective values of different solution candidates generated during the optimization process. To this end, [9] considered scenarios involving multiple DMs with diverse objective goals. Solution candidates were selected based on its dominance relationships among these reference vectors, with the eventual aim of acquiring a subset of solutions that span across various subregions of the PF that were close to the predefined goals. In the context of expensive MOP, [27] introduced a new acquisition function that incorporated the probability of achieving a predefined goal into existing Bayesian optimization algorithms.

Another way to represent *a priori* preference is by specifying m weights that sum to 1, indicating the importance among the m number of conflicting objectives. This facilitates the scalarization of the original MOP, allowing the application of existing single-objective solvers. To overcome the cognitive demanding needs of precise preference articulation, [28] proposed using the weights to instead represent the trade-offs that the DM is willing to make. For instance, in a bi-objective optimization problem involving cost and performance, the DM might specify a willingness to sacrifice no more than 2-units of cost for a 1-unit improvement in performance. This helps identify subregions of the PF with the desired curvature or slope (in the case of m=2) that align with the trade-offs stated by the DM. Building on this idea, the authors in [29] have analyzed the curvature of different parts of the PF to derive various preference sets that represent potential trade-offs preferred by the DM.

In situations where DMs lack a clear and consistent preference, a simplified method of querying DMs through pairwise comparison of the available solutions can be employed. A pioneering idea in this area is the analytical hierarchy process [30], where the DM repeatedly rates the relative importance of different combinations of two objective pairs. These ratings are then

checked for inconsistencies, prompting the DM to re-evaluate as needed before converting the pairwise ratings into a set of weights indicating the importance among the objectives.

Subsequent works in the field of *interactive optimization* have adopted similar approaches to model pairwise comparison outcomes by learning various arbitrary utility functions. One approach involves using a linear scalarization function, where its weights are iteratively adjusted based on new feedback during the interactive process [16]. Extensions of this method have incorporated multiple scalarization functions to increase the number of Pareto optimal solutions identified at the end of the optimization process [31, 32]. Similarly, probabilistic models have been explored to efficiently balance preference learning and optimization, offering the advantage of reducing the number of queries required to avoid DM fatigue [14, 33].

To address high-dimensional objective spaces and bi-level MOPs, quasi-concave polynomial functions were proposed to learn the associated utility functions [34]. Likewise, [35] demonstrated the consistent effectiveness of ordinal regression functions when modeling the utility functions for MOPs with diverse PF shapes. Neural networks have also been studied to approximate more complex utility functions, demonstrating its robustness against inconsistent DM ratings during the interactive optimization process [13, 36].

The works on preference-guided optimization aim to avoid the intensive computational demands of finding a large set of solutions that comprehensively cover the PF to satisfy DM preferences. Instead, it focuses on computing a subset of solutions in highly desirable subregions of the PF, based on either *a priori* or learned preferences during the optimization process. However, the real challenge lies in the DM ability to provide such precise yet accurate preferences with limited knowledge of the true PF topology.

2.2 Machine Learning for Multi-objective Optimization

Assuming that DM is unable to provide any *a priori* information about her preferences, the goal is then shifted to solving the MOP to obtain a diverse set of Pareto optimal solutions for *a posteriori* decision making. Hence, this section provides a concise review on the recent developments in multi-objective optimization algorithms that harness the insights from problem-specific features to expedite the search process.

Existing MOP solvers can be broadly classified into exact mathematical programming methods [17–19] and approximate sampling-based methods [20–22], which iteratively evolve a population of solutions toward convergence. Compared to exact methods, sampling-based methods using evolutionary algorithms are more widely adopted in real-world applications [1–8] owing to their ability to handle more complex problem formulations [37, 38]. Furthermore, most multi-objective evolutionary algorithms (MOEAs) can obtain a set of solutions in a single run and offer the flexibility to customize the underlying algorithm to increase its efficiency in tackling different MOPs [39, 40].

The general steps in an MOEA begin with an initial parent population of randomly sampled solutions, whose objective values are assessed using a multi-objective function. Based on these evaluations, the solutions are ranked [2] according to their diversity and convergence toward the true PF. A new population of offspring is then generated by inheriting features from the top-ranking parent population, and their objective values are evaluated. The offspring and parent populations are subsequently combined and re-ranked.

[2]Based on the ranking approaches, MOEAs can be further divided into three categories, namely Pareto-, indicator-, and decomposition-based MOEAs. Pareto-based MOEAs [24, 41] perform non-dominated sorting to rank solutions based on its Pareto dominance. While Indicator-based MOEAs [42–44] use an indicator such as $I_{\epsilon+}$ [42], Hypervolume [45] and R2 [46] to determine the superiority of solutions. Lastly, decomposition-based MOEAs [20, 47] scalarize a MOP into a set of single-objective optimization sub-problems and are ranked separately accordingly to the single-objective evaluation functions. Commonly used scalarization function include, weighted sum [48], Tchebycheff [48] and penalty-based boundary intersection [49].

This process repeats for a fixed number of generations or terminates early if the solutions converge to the true PF.

To cope with the ever-increasing complexity of real-world MOPs, numerous studies have proposed innovative forms of MOEAs that leverage the data generated during the optimization process. Fig. 2.2 illustrates two learnable insights that can be potentially exploited online during the execution of the MOEA from a machine learning perspective. These insights aim to efficiently tackle the significant challenges inherent in solving real-world MOPs, aligning with the research objectives detailed in Chapter 1.2.

The first challenge involves costly evaluation functions that require time-intensive procedures, such as computer simulations or real-world experiments, to compute objective values for each solution candidate. This significantly limits the number of evaluations possible within a fixed computational budget. The second challenge pertains to highly non-linear dependencies among decision variables, which poses a challenge for traditional MOEAs relying on genetic operators. The inherent randomness of these algorithms can disrupt the solution building blocks or impose restrictive mating conditions that hinder the formation of high-quality solutions [40].

To address the first challenge, existing research has focused on building a more computationally efficient surrogate model of the expensive evaluation functions, as shown by the machine learning task 1 in Fig. 2.2. The surrogate model is first trained on the available premium dataset that has been evaluated by the expensive function $\boldsymbol{f}(\cdot)$. This trained model is then used as a substitute for evaluating new solutions and identifying high-potential solutions that are worth evaluating using $\boldsymbol{f}(\cdot)$. With the newly evaluated premium data, the surrogate model is then retrained and this process of alternating between cheap and expensive evaluation functions continues until the computational budget has been exhausted.

The goal of this research primarily focuses on methods that seek to improve the accuracy of the surrogate model trained under limited data and selec-

tion criteria of high-potential solutions worth re-evaluating with the expensive objective function. In [50], multiple radial basis functions were trained on different subsets of the costly evaluation data available, with the individual predictions of the final objective values averaged to reduce the generalization error. High-potential solutions were then chosen based on non-dominated solutions in different clusters, partitioned using weight vectors. To effectively distinguish solutions based on good convergence and uniqueness, which is important in maintaining diversity, a dual fuzzy classifier was developed in [51]. Subsequently, high-potential solutions were selected based on the sum of the two predicted membership degrees. To further enhance diversity, solutions that potentially lie in the PF bound have also been considered as an additional criterion for re-evaluation by the expensive objective function [52].

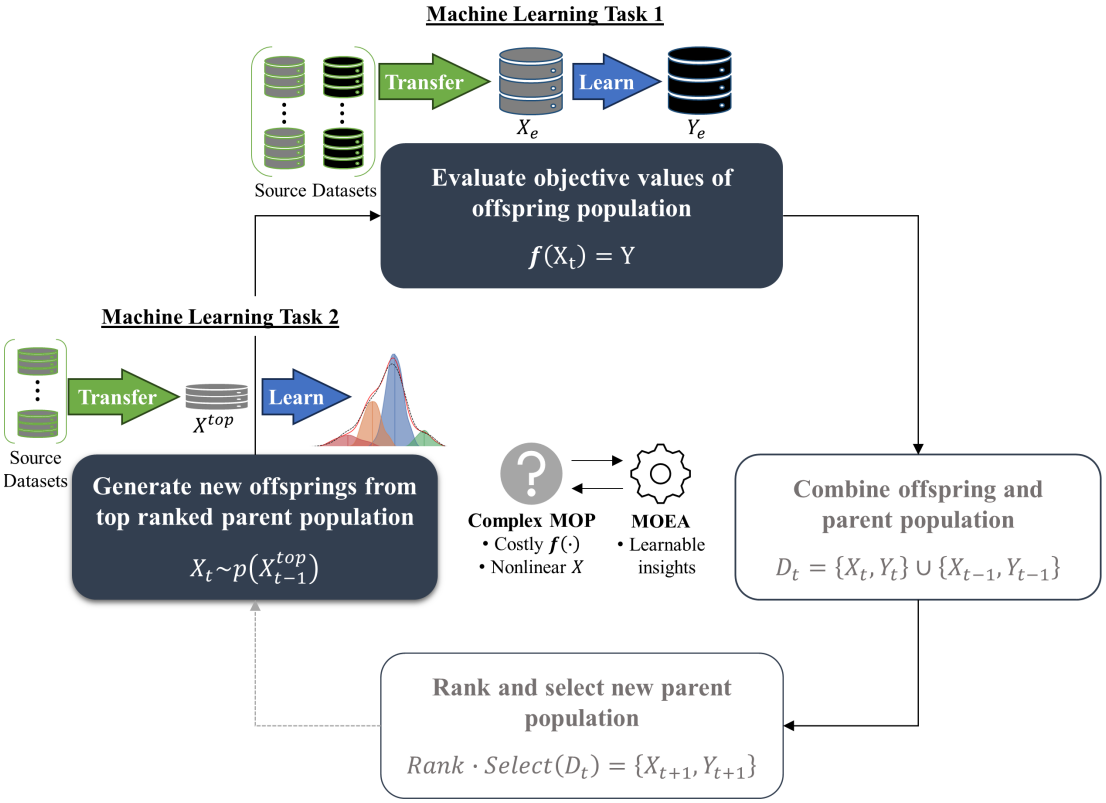

Figure 2.2: A machine learning perspective in MOEA for addressing issues arising from complex MOP.

To enhance the predictive accuracy of the surrogate model during the search process, selecting potential solutions that improve training performance was

another key consideration. In this regard, [53] employed a predefined threshold to alternate the selection of high-potential solutions based on their ability to enhance either PF convergence or surrogate model accuracy. PF convergence improvement was quantified using the distribution of objective values predicted by an ensemble of surrogate models, while surrogate model accuracy improvement was estimated by the reduction in the average distance within the premium training data in the decision space.

Another widely adopted approach to balancing exploitation and exploration, which ensures both good PF convergence and improved surrogate model accuracy, is Bayesian optimization [54]. This method employed a probabilistic regression model that provides both the mean and standard deviation of the predicted objective values. Based on the predictive probability distribution, various acquisition functions, such as the lower confidence bound [55], expected improvement [56], and predicted entropy search [57] can be formulated to choose high-potential solutions to balance exploitation and exploration. Building on this concept, the authors in [58] proposed a new adaptive acquisition function that increasingly emphasized exploitation as the computational budget for the expensive function evaluator decreases. To extend the modeling of expensive MOPs with constraints, [59] suggested using another probabilistic-based surrogate model to predict the degree of constraint violation. Leveraging these two surrogate models, high-potential solutions were chosen based on the maximum improvement that can be achieved in both objective and constraint violation values.

Advances in enhancing the efficiency of Bayesian optimization in high-dimensional decision and objective spaces have also garnered significant attention. To improve the scalability of surrogate models in these spaces, [60] proposed using multiple surrogate models, each trained on data subsets from different local regions of the objective space. This approach facilitates parallel searching for high-potential solutions through a coordinated strategy. Additionally, various dimensionality reduction methods in decision spaces have been explored. For

instance, [61] jointly trains a denoising autoencoder and a surrogate model, enabling efficient gradient-based optimization within the autoencoder's latent space. In [62], an efficient Monte Carlo sampling algorithm was introduced to rapidly identify sparse, axis-aligned subspaces for surrogate modeling of critical decision variables. Further research has focused on developing efficient machine learning models for handling a large number of objectives. For example, [63] investigates a computationally efficient dropout neural network and a method for adaptively balancing convergence and diversity when selecting promising solutions.

Besides improving the predictive accuracy of the surrogate model with new training data, transfer learning approaches have been another area explored. In [64], the authors divided the objective space into different partitions, where separate probabilistic surrogate models were first trained. Subsequently, a joint surrogate model learning was performed to exploit the correlation among different sub-objective spaces. In [65], MOPs with cheap and expensive objective evaluation functions were considered. The partial evaluation data from the cheap objective evaluator was then adaptively used to augment the surrogate model's predictive accuracy for the expensive objective function. Beyond transfer learning within a single MOP, [66] proposed multi-problem surrogates that maximally utilized other potentially correlated data sources. The transfer mechanism is based on a stacking approach involving a meta-regression stage to learn weighted predictions of the multiple base surrogates. Similarly in [67], a multi-output neural network trained for a specific injection molding process is transferred to a related target task by re-training the output-adjacent layers with a small amount of target data. Additionally, [68] evaluated the theoretical and empirical performance of a multi-task Gaussian process model for source-target transfer, showing faster convergence compared to the no-transfer scenario. While in [69], the separable structure of transfer covariance matrices in multi-task Gaussian processes was exploited to reduce the computational load in high-dimensional MOP.

In addition to improving the predictive accuracy of surrogate models with new training data, transfer learning has also been actively explored. In [64], the authors partitioned the objective space and trained separate probabilistic surrogate models for each partition. A joint surrogate model was then constructed to exploit correlations across different sub-objective spaces. In [65], MOPs with both cheap and expensive objective evaluation functions were considered. Partial evaluation data from the cheaper objective function was adaptively used to improve the surrogate model's accuracy for the more expensive objective function. Meanwhile, [69] exploited the separable structure of transfer covariance matrices in multi-task Gaussian processes across different objectives to reduce computational costs in high-dimensional MOPs.

Beyond transfer learning within a the scope of a single MOP, [66] introduced multi-problem surrogates that leveraged potentially correlated data sources. Their transfer mechanism employed a stacking approach with a meta-regression stage to generate weighted predictions from multiple base surrogates. Similarly, in [67], a multi-output neural network trained for a specific injection molding process was transferred to a related target task by retraining the output-adjacent layers with a small amount of target data. Additionally, [68] evaluated both the theoretical and empirical performance of a multi-task Gaussian process model for source-target transfer, demonstrating faster convergence compared to the no-transfer scenario.

The next challenge is tackled in the existing literature by modeling the highly non-linear dependencies among decision variables of current non-dominated or better performing solution candidates as a probability distribution function. Using the modeled distribution $\boldsymbol{p}(X_{t-1}^{top})$, the decision variable values of new offspring X_t were sampled, eliminating the need of genetic operators[3], as illustrated in machine learning task 2 of Fig. 2.2. Pioneering work in this area has focused on multi-objective estimation of distribution algorithms [72],

[3]Note, an alternative approach to fully replacing the genetic operators is the use of data analytics to complement the operators, which is not part of the review scope. Interested readers can refer to [70, 71] for relevant examples.

where different probabilistic models were employed to effectively capture dependencies among variables and the regularity property of the current PS.

To accurately represent the correlations, [73, 73] proposed modeling the probability distribution as a directed acyclic graphical model, with the nodes representing decision variables and the arcs representing dependencies among these nodes. Alternatively, the regularity property of MOPs, where the PS shape can be represented in a lower dimensional space than the number of decision variables, has been explored by [74, 75]. The authors used local principal component analysis to build simplified probability models in the reduced decision space. To ensure the generation of diverse solution candidates in different subregions of the PF, mixture distribution models have been utilized instead in [76, 77]. Different probabilistic models were built separately based on the clustering of the objective values of the solution candidates. More recently, to generate high-quality offspring, [78, 79] proposed using generative adversarial networks, where the current population of solutions were classified into high- and poor-performing groups for training.

Apart from learning the dependencies among target solution candidates, transfer learning techniques that exploit dependencies across different MOP tasks to accelerate PF convergence have also been extensively studied in recent years. The focus has primarily been on evaluating the similarity between different pairs of source-target tasks to determine the extent of transfer.

In [80], the extent of transfer from source to target task was determined based on the previous steps of the optimization process. For instance, if a previously transferred source solution led to a non-dominated solution in the target MOP, it will be considered as a positive transfer, and the extent of transfer from this source task will be increased. Subsequently, in [81], the same authors proposed a more elaborate method to better quantify the measure of source-target similarity. They introduced the idea of transfer rank, where all past transfers that led to positive results will cause an increase in the ranking. To further differentiate source tasks with the same rank, a k-

nearest neighbor clustering method based on regions of positive and negative transfer was additionally proposed. Meanwhile, the authors in [82] proposed measuring similarity by fitting a Gaussian mixture model for all the top-performing solutions of the source and target populations. The computed mixture weights were then used to define the extent of transfer. This approach was later adopted in [83] to optimize various problem instances of an integrated energy system, yielding improved optimization performance.

A selection strategy based on a Wasserstein-based similarity measure between the source and target population distributions was employed to better distinguish the extent of transfer for various source tasks [84]. This strategy selects the most similar source task to the target if its similarity measure is higher than those of the other source tasks. In cases where the variance of the similarity measure across source tasks is high, all source tasks, or the top K most similar source MOPs, are considered and weighted proportionally according to their similarity.

For more granular transfer, the objective space of the MOPs is partitioned into subspaces using different weight vectors, with solution transfer restricted to the same subspace [85]. Similarly, in [86], the likelihood of selecting a solution for transfer is assessed by comparing the solution density and convergence across sub-regions of the objective space for each source-target pair. To enhance the convergence of the primary constrained MOP, the authors in [87] introduced an auxiliary task without constraints. Solutions for both tasks are then generated by selecting parent solutions either within or across tasks, based on past improvements in both scenarios.

Machine learning techniques aimed at increasing the overlap between source and target MOP tasks have also been a key area of research. In this regard, [88] introduced the idea of representation learning via spatial transformations to handle mismatches in search space dimensionalities between distinct MOPs, thereby increasing the overlap between their optimized search distributions. Similarly, [89] proposed a refinement method for the source

solution to be transferred, where more drastic refinement was applied if the similarity measure was low. In [90], a discriminative reconstruction network was developed for each MOP task to directly transfer a source solution, control the extent of the transfer, and construct a Pareto optimal subspace for the efficient evolution of target MOP solutions.

The works reviewed herein focus on using machine learning techniques to tackle the challenges of solving MOPs with expensive-to-evaluate functions. This is achieved by developing surrogate models that efficiently predict objective values and by selecting high-potential solutions for costly evaluation. Additionally, to manage nonlinear dependencies among decision variables, various probabilistic models are explored to generate subsequent offspring solutions. However, for *a posteriori* decision making, these works address only part of the issue on faster PF convergence and do not tackle the challenge of insufficient PF representation.

2.3 Inverse Models for Pareto Set Learning

In this section, an overview of existing works on the application of inverse models for PS learning in MOPs is provided. The literature is broadly categorized into two research strands, referred to herein as (a) *online* PS learning and (b) *post-hoc* PS learning with the former being applied during the optimization process.

Unlike the probabilistic models discussed in the previous section, which do not explicitly consider the objective space, online PS learning aims to generate solution candidates more precisely based on different points along the approximated PF. This is achieved via inverse supervised machine learning model that are trained to map points from the objective to the decision space. Here, the inverse models are repeatedly updated based on data being generated during each optimization step, and subsequently inform the sampling of promising solution candidates in the next iterations. To this end, [91]

utilized multiple Gaussian process (labelled hereafter as *inv*GP). Each *inv*GP was tasked to predict a single decision variable value. The training data was first partitioned into subspace in objective space based on uniformly distributed reference vectors. Within a subspace, they applied a random grouping technique to determine which inverse models were to be built, training an *inv*GP for each.

To address the issue of irregular (non-uniform or disconnected) PFs, various objective space partitioning techniques have also been proposed in the literature. Adaptive reference vector generation in the context of online PS learning was explored in [92], adjusting or removing reference vectors based on the number of solutions associated with each partition. In [93], K-means clustering was applied to partition the data before training multiple inverse models. Alternatively, the random grouping mechanism by [91] has been the subject of further study and refinement. For instance, a feature importance method with random forests [94] was applied to determine better assignments of decision variables to objective functions. Likewise, a nonrandom grouping strategy [95] was put forth to enhance the reliability of the inverse model.

Recent efforts have explored neural PS learning in multi-objective combinatorial optimization [96] and multi-objective Bayesian optimization for computationally expensive problems [97]. Additionally, methods to leverage external datasets to enhance the predictive accuracy of inverse models have been investigated. In [98], the authors proposed using both non-dominated and dominated solutions within each optimization step to build an inverse generative model, improving optimization performance. In [99, 100], datasets from diverse optimization tasks were employed to develop inverse multi-task Gaussian process models, where each model was trained to map preference vectors in the objective space to a specific dimension in the decision space. Another emerging application of online PS learning involves optimizing dynamic, real-time systems, where the inverse model is continually retrained with new streaming data [101, 102].

On the contrary, post-hoc PS learning serves to aid decision making by enhancing the density of the PF approximation given a target MOP and its solution evaluation data acquired in the course of *a posteriori* multi-objective optimization. The goal is for a DM to have the ability to generate new near-optimal solutions on-demand, simply by prompting the inverse model at unexplored regions of the PF. An early work in this regard was carried out in [103], where the authors employed a radial basis function network (labelled hereafter as *inv*RBFNN) structure for the inverse model. While their method was agnostic to the choice and behavior of the underlying MOP solver, subsequent attempts to improve the accuracy of the *inv*RBFNN have sought to refine the distribution/ placement of training samples generated during the optimization run. One generally applicable idea, not restricted to *inv*RBFNNs, was to bias the optimizer to generate more data in regions of greater geometrical change in the PS [104], under the intuitive assumption that the topology of a function can be better interpolated if its high variation regions are well sampled. Similarly, [105] proposed an approach for generating new solutions using inexpensive sampling techniques in tandem with an inverse model. Subsequently, these solutions' objective values were evaluated and selected according to their Pareto optimality to update the inverse model. This entire procedure was then repeated several times to achieve a more well represented PF.

Other works in post-hoc PS learning have considered challenges arising from complexities of the PF. For example, [106] proposed a method for multi-modal MOPs, where a one-to-many mapping could arise from objective to decision space due to the presence of multiple solutions that result in identical objective function values along the PF. In [26], the performance of the inverse model for many-objective optimization problems (MaOPs: those with four or more objective functions) was investigated. The authors revealed a blessing of dimensionality of many-objective search, showing that training data obtained from an MaOP could result in better accuracy compared to the data generated from its dimensionally reduced counterpart. In an effort

to eliminate the need for preference articulation, [107] proposed an algorithm for detecting knee regions (that are naturally preferred by DMs) along high-dimensional/ complex PFs, facilitating the discovery of corresponding points in the PS by an *inv*RBFNN. In a separate study [108], the authors examined the performance of various machine learning models for post-hoc PS learning of MaOPs. From their extensive empirical investigation, they concluded that the independent *inv*GP model provided the best overall performance.

Despite the growing interest in both online and post-hoc PS learning, research in these areas is still in its early stages compared to the plethora of multi-objective optimization algorithms with forward models or preference models discussed in the previous two sections. Additionally, there is a notable lack of work focusing on improving the performance for post-hoc PS learning in limited data regimes.

2.4 Chapter Summary

Existing research has primarily focused on leveraging machine learning techniques to address the cognitive challenges faced by DMs during multi-criteria optimization and decision making processes. To mitigate the issue of incomplete knowledge for *a priori* preference articulation, interactive optimization methods allow DMs to iteratively express their preferences throughout the optimization process. Various machine learning models have been developed to accurately capture DM utility functions while minimizing the number of preference queries.

In contrast, enhancing decision quality during *a posteriori* decision-making has driven the integration of machine learning tasks to complement existing optimization algorithms. Research in this area has primarily focused on (i) developing cost-efficient surrogate models as alternatives to expensive solution evaluators, or (ii) modeling and sampling the distribution of promising solutions. Recent advancements have emphasized improving the scalability of

MOPs with numerous decision and objective variables through dimensionality reduction techniques. Another notable approach involves leveraging transfer learning from related optimization tasks to enhance surrogate model accuracy or improve non-dominated solution distribution modeling.

An emerging research direction gaining momentum focuses on increasing the quantity and diversity of decision choices during *a posteriori* decision-making. In this context, inverse machine learning models for PS learning have been explored to map solutions from the objective space back to the decision space. This can occur either during (online PS learning) or after (post-hoc PS learning) optimization, with the former also aiming to improve optimization convergence. However, training accurate inverse models remains challenging, particularly in domains with limited training data.

This thesis aims to fill the research gap of enhancing the accuracy of inverse models during post-hoc PS learning for the on-demand, controlled generation of new non-dominated solutions in complex and expensive MOPs characterized by high dimensionality, non-linear PS, and limited datasets. It explores the integration of existing post-hoc PS learning techniques with emerging approaches, such as transfer learning and low-dimensional manifold modeling. The methods developed in this work have the potential to advance online PS learning, paving the way for innovative optimization algorithms in future research.

Chapter 3

Preliminaries

In this chapter, the basics and definitions of keys concepts for multi-objective optimization, as well as inverse modeling of PS are presented.

3.1 Multi-objective Optimization

Without loss of generality, a multi-objective problem can be stated as a minimization problem as follows,

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})] \\ & s.t. \quad \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d, \end{aligned} \tag{3.1}$$

where m is the total number of objectives to be minimized, f_i being the i^{th} objective function, and $\mathcal{X}$ being the feasible region of a d-dimensional decision space. $\boldsymbol{f}(\boldsymbol{x})$ is thus a *forward map* from points in decision space to the objective space. Note that a maximization problem could simply be written as minimizing the negative of $\boldsymbol{f}(\boldsymbol{x})$.

Assuming conflicting objectives in Eq. (3.1) (such that no single solution exists that simultaneously optimizes all the objectives), the goal is to arrive at a *set* of so-called *Pareto optimal* solutions, with each solution embodying a different trade-off among the objectives. The definitions of key terms associated with the notion of Pareto optimality in MOPs [109] are presented as

follows.

Definition 1 (Pareto Dominance)

A solution $\boldsymbol{x}_a$ *is said to Pareto dominate solution* $\boldsymbol{x}_b$ *if* $\forall i \in \{1, 2, .., m\}$*:* $f_i(\boldsymbol{x}_a) \leq f_i(\boldsymbol{x}_b)$ *and* $\exists j \in \{1, 2, .., m\}$ *such that* $f_j(x_a) < f_j(x_b)$*.*

Definition 2 (Pareto Optimality)

A solution $\boldsymbol{x}^*$ *is called Pareto optimal if there exists no solution* $\boldsymbol{x} \in \mathcal{X}$ *that Pareto dominates* $\boldsymbol{x}^*$*.*

Definition 3 (Pareto Set)

The set of all Pareto optimal solutions constitutes the Pareto set (PS) in decision space.

Definition 4 (Pareto Front)

The image of the Pareto set in the objective space is called the Pareto font (PF).

Definition 5 (Ideal Point)

The ideal point is the vector in objective space whose components are the solution of each single-objective problem $\min_{\boldsymbol{x} \in \mathcal{X}} f_i(\boldsymbol{x})$*,* $i = 1, 2, \ldots, m$*.*

Definition 6 (Nadir Point)

The nadir point is the vector in objective space whose components are the solution of each single-objective problem $\max_{\boldsymbol{x} \in \mathcal{X}_P} f_i(\boldsymbol{x})$*,* $i = 1, 2, \ldots, m$*, where* $\mathcal{X}_P$ *denotes the PS.*

These concepts lie at the heart of post-hoc PS learning with the purpose of obtaining an accurate inverse map from the PF in objective space to the PS in decision space. In this regard, the ideal and nadir points provide the lower and upper bound vectors that constrain the set of possible points in the objective space.

3.2 Inverse Modeling of Pareto Set

In post-hoc PS learning, no strong assumption is made about the algorithm used to solve Eq. (3.1). Let the PF approximation data obtained by the

end of a run of any MOP solver be $Y \in \mathbb{R}^{n \times m}$, and the corresponding non-dominated solutions in decision space be $X \in \mathbb{R}^{n \times d}$, where n is the number of points along the PF acquired. For optimization in domains with expensive-to-evaluate objective functions, n would typically be small—e.g., in the order of hundreds or fewer points [22]—offering insufficient coverage of the PF. In such cases, PS learning can serve to enhance the density of the PF approximation, or satisfy a DM's postponed preferences by inversely generating corresponding solutions on-demand [103].

However, for a DM to precisely articulate her preferences along the approximated PF, prior knowledge of the PF's topology needs to be known. This information is inherently difficult to possess due to our initial assumption of data scarcity. Furthermore, MOPs characterized by complex, irregular PFs, such as those with discontinuities, exacerbate this difficulty. Therefore, the initial step towards facilitating inverse generative decision support involves transforming points along the approximated PF Y into a projected set $W \in \mathbb{R}^{n \times m}$, which can be queried independently of the PF's topology. This transformation maps each point in Y to a point in W, which we denote by the function,

$$\Pi^{-1} : Y \to W. \tag{3.2}$$

Fig. 3.1 illustrates one such realization of the transformation function in Eq. (3.2) for $m = 2$, adopted from [110]. The data in Y is first normalized to the range $[0, 1]$ based on the ideal and nadir points estimated from Y. The normalized points then undergo orthogonal projection onto the unit hyperplane $\mathcal{W}$ to produce the dataset W. The hyperplane is defined by the (m-1)-simplex $\{\boldsymbol{e}_1, ..., \boldsymbol{e}_m\}$, where $\boldsymbol{e}_i$ is a vector of zeros with a one in the i^{th} position. In the case of Fig. 3.1, the hyperplane reduces to a line passing through $(0, 1)$ and $(1, 0)$, along which the DM can easily articulate her preferences for f_1 or f_2 or a weighted combination of them, without having to deeply take into consideration the topology of the PF.

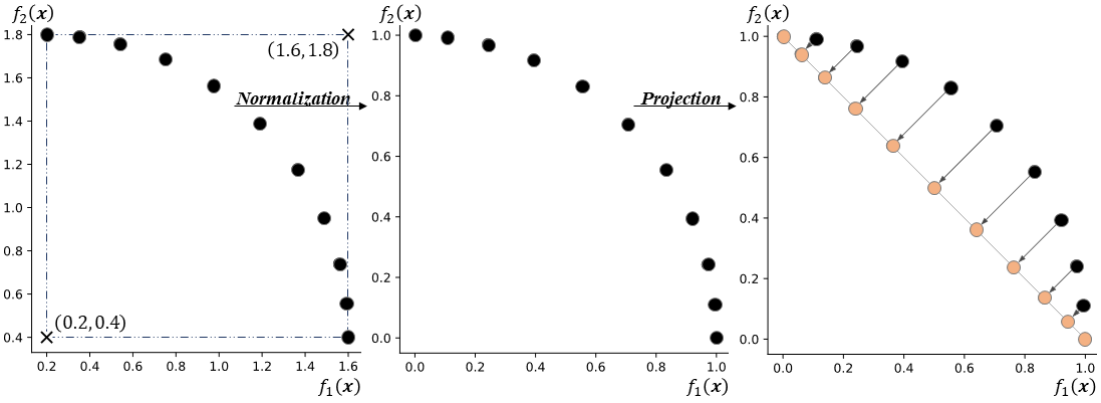

Figure 3.1: Illustration of the Π^{-1} mapping procedure from the approximated PF Y (in yellow) to the projected set W (in orange) along the unit hyperplane.

Given the projected set of points, post-hoc PS learning entails the construction of an inverse function $\boldsymbol{\psi}^{-1} : \mathcal{W} \rightarrow \mathcal{X}$, through training a machine learning model. It is worth emphasizing that if the Karush-Kuhn-Tucker (KKT) conditions hold in a given problem, both the PF and PS are (m-1)-dimensional piecewise continuous manifolds for m-objective optimization problems under certain mild conditions. This has led to the common assumption, albeit without guarantee, that the mapping from the PF to PS is indeed a one-to-one injective function [91, 111]. Nonetheless, even if the underlying inverse function is not one-to-one, it can still be approximated by an arbitrary function $\boldsymbol{\psi}_{\boldsymbol{\theta}}^{-1}$, parameterized by $\boldsymbol{\theta}$, though it cannot strictly be considered the inverse image. Moreover, it has been postulated that in practice, non-injectivity does not necessarily delimit the application and could in fact be beneficial for inverse modeling [103].

Fig. 3.2 shows the complete workflow of post-hoc PS learning with an inverse machine learner. The inverse model is trained in a supervised learning setting based on the derived dataset D, consisting of W and X as its respective inputs and outputs. With an accurate inverse model in hand, a DM can in principle prompt the model with an arbitrary set of preference points $W_q \subset \mathcal{W}$ in unexplored sub-regions of the projected PF, producing desired

solutions in the PS as,

$$\boldsymbol{\psi}_{\boldsymbol{\theta}}^{-1}(W_q) = X_q. \tag{3.3}$$

The solutions in X_q can then be evaluated with the forward map to validate the quality of outputs produced by the inverse model. For example, the model's PF approximation capacity can be quantified by the improvement in spread and convergence to the PF of $Y_q = \boldsymbol{f}(X_q)$ relative to the points used for training. (For synthetic problems where the theoretical PF is known, this can be achieved by means of various *generational distance* metrics [112].) Assuming a smooth one-to-one mapping between the PS and the (m-1)-dimensional unit hyperplane in objective space [91], the accuracy of the inverse model to a specific DM query could also be quantified by the Euclidean distance of its prediction to the true Pareto optimal solution.

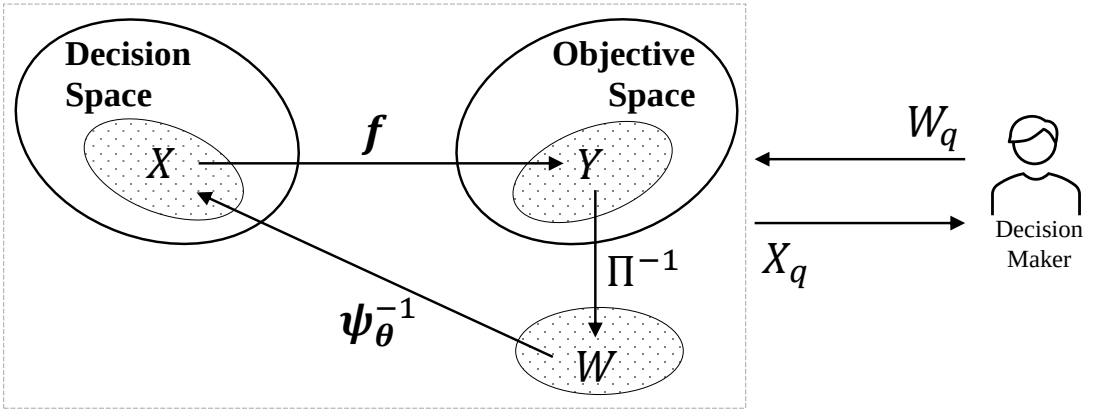

Figure 3.2: Workflow for multi-objective decision support via post-hoc PS learning. The DM provides preference vectors W_q in the projected unit hyperplane for the inverse model to generate corresponding solutions X_q within the PS.

3.3 Chapter Summary

In this chapter, the formulation of MOP, along with key concepts and its definitions were introduced. Additionally, the concept of PS learning and the essential steps involved in constructing an inverse machine learning model for *a posteriori* decision making were presented. To better illustrate these

concepts, the next chapter provides a background study on how they work together for *a posteriori* decision making, using a supply chain planning problem as an example.

Chapter 4

Background Study on Human-AI Multi-criteria Decision Making

Innovative solutions often arise from the synergistic contributions of diverse entities with complementary strengths and weaknesses. This trend is expected to continue with the advent of artificial intelligence (AI). Therefore, this chapter examines whether the synergistic collaboration between human DMs and AI algorithms in the field of optimization and machine learning can significantly enhance the convergence and diversity of the Pareto data.

To achieve this, two potential scenarios of Human-AI complementary decision making are proposed. The first scenario leverages human-supplied priors to inform the search process, thereby accelerating the convergence performance of optimization algorithms[1]. The second scenario involves an inverse machine learning model trained on acquired Pareto data to aid DMs in exploring new points from the PF to the PS.

Both scenarios are tested using a carefully designed multi-criteria order planning problem, demonstrating accelerated convergence during optimization and a denser approximation of the PF for *a posterior* decision making. This further reinforce the importance of omni-directional Human-AI collaboration in multi-criteria decision making.

[1]The work in this chapter has been published in [7].

4.1 Human-AI Complementary in Multi-criteria Decision Making

In the face of new and challenging problems, humans often resort to brainstorming activities where people with diverse backgrounds and perspectives can contribute ideas in the hope of formulating an effective solution in a short period of time. Taking this cue, in the realm of AI research, existing works in Human-AI complementarity has focused on leveraging human cognitive strengths to improve the performance of known AI algorithms. For example, [113] proposed an interactive image retrieval system where doctors could articulate the similarity criteria when retrieving past, medical images as a reference for new patient diagnosis. [114] developed a Bayesian framework for combining the predictions and different types of confidence scores from humans and AI algorithms to achieve higher accuracy in image classification tasks. In the area of multi-criteria optimization, many have focused on user preference articulation either before [115, 116] or interactively [32, 117] during the course of the optimization process to efficiently arrive at the desired subregion of the full PF.

Building on ideas from previous works, we focus on Human-AI collaboration in multi-objective optimization. Fig. 4.1 illustrates two scenarios for the involvement of DM: (a) during the optimization process to accelerate PF convergence, and (b) in *a posterior* decision making, as discussed in Chapter 1.

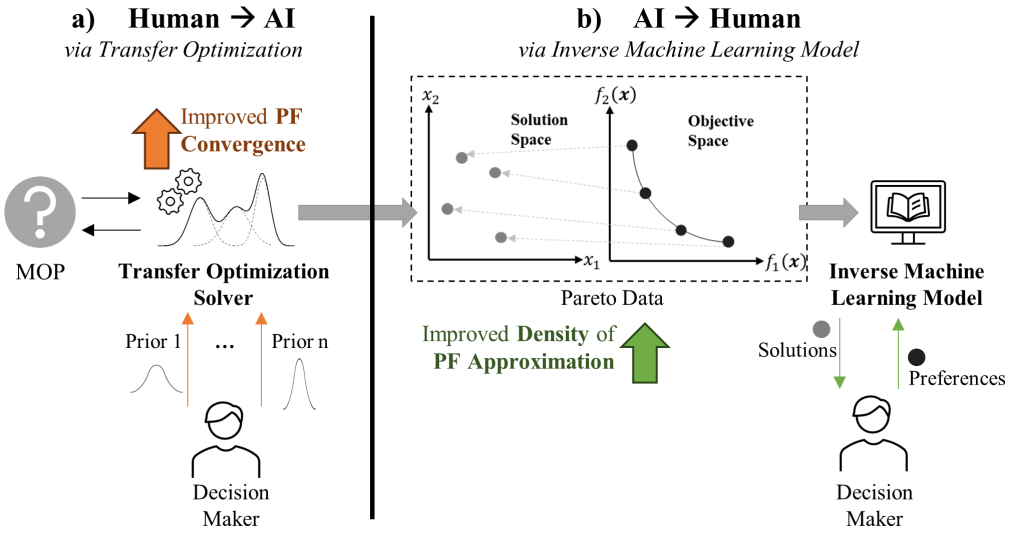

Figure 4.1: Human-AI complementary decision making framework, focusing on (a) Transfer optimization with human supplied priors and (b) Inverse machine learning models for *a posterior* decision making.

4.1.1 Human to AI Collaboration via Transfer Optimization

In the first scenario of Fig. 4.1a, the DM's prior belief about the location of optimum solutions after being presented with the MOP is modeled as a probabilistic model. The mean and variance of the model represent the DM's intuition on the possible location of good solutions and her confidence level, respectively. In the event where the decision maker has no prior knowledge of certain solution variables, a uniform distribution over all possible values can be utilized. Next, to adaptively transfer the human supplied prior during the course of the optimization process, this chapter adopts the transfer evolutionary optimization method from [118, 119], where the objective function $\boldsymbol{f}(\boldsymbol{x})$ is transformed to a probabilistic modeling perspective as [120]:

$$min_{\boldsymbol{p}(\boldsymbol{x})} \int_{\mathcal{X}} \boldsymbol{f}(\boldsymbol{x})\boldsymbol{p}(\boldsymbol{x}) \quad d\boldsymbol{x} \tag{4.1}$$

where $\boldsymbol{x} \in \mathcal{X}$ is a candidate solution vector in the feasible search space $\mathcal{X}$, and $\boldsymbol{p}(\boldsymbol{x})$ denotes the underlying probability distribution model of the search

population in an evolutionary algorithm. Accordingly, to model the intensity of transfer or complementarity in the human supplied prior, Eq. (4.1) can be reformulated into:

$$max_{w_1,w_2,...,w_{\mathcal{T}} \cdot \boldsymbol{p}_{\mathcal{T}}(\boldsymbol{x})} \int_{\mathcal{X}} \quad \boldsymbol{f}_{\mathcal{T}}(\boldsymbol{x}) \left[\sum_{h=1}^{k} w_h \cdot \boldsymbol{p}_h^*(\boldsymbol{x}) + w_{\mathcal{T}} \cdot \boldsymbol{p}_{\mathcal{T}}(\boldsymbol{x}) \right] \quad d\boldsymbol{x} \tag{4.2}$$

where, $\boldsymbol{f}_{\mathcal{T}}$ is the objective function of the target problem to solve, $\boldsymbol{p}_{\mathcal{T}}$ is the target search distribution model, w_h can be interpreted as the complementarity coefficient of the human supplied priors $\boldsymbol{p}_h^*(\boldsymbol{x})$, for $h \in [1, ..., k]$ assuming there to be k DMs, which are constrained to $\sum_{h=1}^{k} w_h + w_{\mathcal{T}} = 1$ and $w_h \geq 1, \ \forall\, h$. In the simplest case and in the remainder of this chapter, $k = 1$. For the sake of brevity, interested readers are referred to [119] for details on how the target mixture $\left[\sum_{h=1}^{k} w_h \cdot \boldsymbol{p}_h^*(\boldsymbol{x}) + w_{\mathcal{T}} \cdot \boldsymbol{p}_{\mathcal{T}}(\boldsymbol{x}) \right]$ is built and the theoretical analysis on how embedding the method into a probabilistic model-based evolutionary optimizer could cultivate knowledge transfers / complementarity.

4.1.2 AI to Human Collaboration via Inverse Model

The second scenarios of Fig. 4.1b focuses on the inverse mapping of points from the acquired PF to PS by learning the function $\boldsymbol{\psi}^{-1} : \mathcal{W} \rightarrow \mathcal{X}$, as discussed in Chapter 3.2. A prevalent choice of the inverse model structure is the nonparametric Gaussian process (GP) [121], which models uncertainty in predictions by defining a distribution over functions. A strong motivation behind this choice is the uncertainty-awareness of GPs, deemed invaluable for rationalizable Human-AI interactions [122]. Additionally, it has been explored in recent works [103, 123, 124] and shown in [124] to provide competitive predictive performance over other machine learning models, such as neural network that requires a large number of training data.

Hence, in this chapter, we make use of the state-of-art GP regressions in the realm of PS learning for modeling each decision variable. For the j^{th}

decision variable, the *inv*GP describes it as a distribution over functions $\psi_j^{-1} \sim \mathcal{GP}(\mu_j(\boldsymbol{w}), k_j(\boldsymbol{w}, \boldsymbol{w}^{'}))$, with $\mu_j(\boldsymbol{w})$ as the mean (typically set to a constant, zero) and $k_j(\cdot, \cdot)$ as some valid covariance function. The inverse map is thus a stochastic process wherein any finite subset of random variables follows a joint multivariate Gaussian distribution.

Given the observations in $\{W, X_j\}$, the posterior predictive distribution at any preference point $\boldsymbol{w}_q$, can be analytically obtained [121] as,

$$\mu_j(\boldsymbol{w}_q) = \boldsymbol{k}_{\boldsymbol{w}_q}^{\intercal}(K + \sigma_{\epsilon_j}^2)^{-1} X_j, \tag{4.3a}$$

$$\sigma_j^2(\boldsymbol{w}_q) = k(\boldsymbol{w}_q, \boldsymbol{w}_q) - \boldsymbol{k}_{\boldsymbol{w}_q}^{\intercal}(K + \sigma_{\epsilon_j}^2 I)^{-1} \boldsymbol{k}_{\boldsymbol{w}_q} + \sigma_{\epsilon_j}^2, \tag{4.3b}$$

where $\boldsymbol{k}_{\boldsymbol{w}_q}$ is the kernel vector between $\boldsymbol{w}_q$ and W computed using any valid covariance function with K as the overall covariance matrix of the *inv*GP and σ_ϵ is the noise term. The (hyper-)parameters of the selected covariance function and the noise term, denoted jointly as $\boldsymbol{\theta}$, are optimized by maximizing the log marginal likelihood $\log p(X_j|W)$, given as,

$$-\frac{1}{2} X_j^{\intercal}(K + \sigma_j^2 I_n)^{-1} X_j - \frac{1}{2} \log |K + \sigma_j^2 I_n| + const.$$

4.2 Multi-criteria Supply Chain Planning Case Study

The vision on Human-AI complementary in multi-criteria decision making is illustrated using a supply chain order optimization problem [7]. It focuses on computing the order allocation and production plans under demand uncertainties that minimizes the expected order fulfillment time and unutilized production capacity for a given planning horizon. The decision variables thus include the allocated order quantities from each customer to its respective factories and the minimum production quantity required to start a production at each factory.

Fig. 4.2 shows the dynamics of the order planning model in each time step in the planning horizon. During the order allocation stage, the customer demand is generated from a given distribution. The order quantity is then split and allocated to various factories based on the computed order allocation plan. At the production stage, the available factories will begin its production for a given production time if the allocated quantities meet or exceed the computed minimum production quantity. Once the production has been completed, the orders are shipped based on a given transportation time for order fulfillment.

The first objective on the order fulfillment time is calculated as a sum of the queuing, production and transportation time of the factory that took the longest time to fulfill it. While the second objective on the unutilized capacity is the difference between its production capacity and production quantities in each production cycle. The 2 objectives are conflicting as reducing the order fulfillment time would require lowering the minimum production quantity such that the queuing time is shortened. However, with a lower minimum production quantity, it will cause the unutilized production capacity to increase.

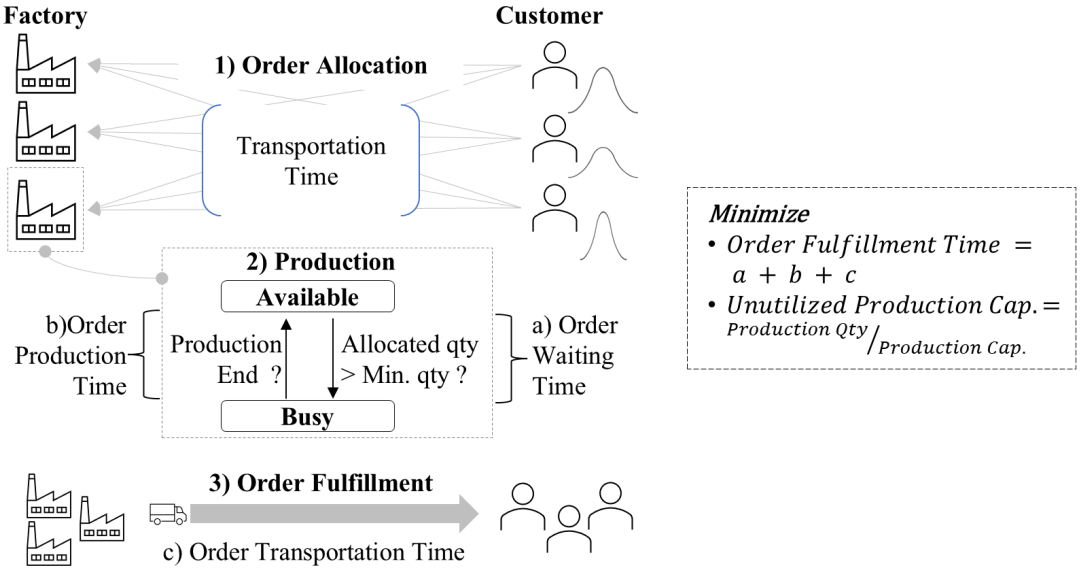

Figure 4.2: Dynamics of the multi-criteria supply chain planning model.

4.2.1 Experiment Design

A Supply Chain network consisting of 3 factories and customers with its parameters related to the 3 variants of the multi-criteria order planning problem is shown in Table 4.1. It has a total of 12 decision variables, with its first 9 as the allocation percentage from each customer to its respective factories while the last 3 as the minimum production quantities required to trigger a production in each factory. For all the problem instances, the planning horizon is set to 20 time steps and the objective values are evaluated as the average across the 20 replications.

Table 4.1: Problem parameter settings for problem instance 1 to 3.

	Production Time	Maximum Capacity		Daily Demand
Problem Instance 1				
Factory 1	4	480	Customer 1	$\max(\mathcal{N}(100, 80), 0)$
Factory 2	3	216	Customer 2	$\max(\mathcal{N}(60, 48), 0)$
Factory 3	2	96	Customer 3	$\max(\mathcal{N}(40, 32), 0)$
Problem Instance 2				
Factory 1	3	240	Customer 1	$\max(\mathcal{N}(100, 80), 0)$
Factory 2	3	240	Customer 2	$\max(\mathcal{N}(60, 48), 0)$
Factory 3	3	240	Customer 3	$\max(\mathcal{N}(40, 32), 0)$
Problem Instance 3				
Factory 1	2	96	Customer 1	$\max(\mathcal{N}(100, 80), 0)$
Factory 2	3	216	Customer 2	$\max(\mathcal{N}(60, 48), 0)$
Factory 3	4	480	Customer 3	$\max(\mathcal{N}(40, 32), 0)$

The experiment is designed such that one obvious prior belief that a DM might form, is to allocate each customer orders to its nearest factories with the Minimal Transportation Time (MinTT) following Table 4.2. The prior belief can be then represented as a probabilistic model using a multivariate, independent normal distribution. The mean values will represent the average allocation percentage with a value of 100% for customer and factory pairs with the minimal transportation time of 1 unit (i.e., Customer 1 to Factory 1, Customer 2 to Factory 2 and Customer 3 to Factory 3) and 0%, otherwise. The variances of the 9 solution variables are all set to 10 to denote a 90%

confidence level on the prior belief.

Table 4.2: Transportation time for problem instance 1-3.

Transportation Time			
	Customer 1	Customer 2	Customer 3
Factory 1	1	2	3
Factory 2	2	1	2
Factory 3	3	2	1

4.2.2 Result on Transfer Optimization

The multi-criteria evolutionary algorithm, NSGA II [125] is employed as the base solver with a population size of 200 and being run for 300 generations in all the experiments. Eq. (4.2) with the MinTT probabilistic model as the human supplied prior, p_H is incorporated into NSGA II for enabling adaptive transfer during the course of the optimization process.

Fig. 4.3a shows the average hypervolume [115] over the 10 replication runs for NSGA II with and without adaptive transfer of the human supplied prior. While Fig. 4.3b shows the distributions of the human supplied prior and the Pareto solutions obtained in NSGA II with adaptive transfer (Human + AI). Through analyzing the results and solution distributions, the following insights are derived.

Insight 1: Perfect human supplied prior

The human supplied prior (MinTT probabilistic model) accurately represents the Pareto solution distributions of problem instance 1. As it minimizes the fulfilment lead time by reducing the transportation time to the minimal. In addition, it allows each factory to have sufficient production capacity to cope with its allocated customer demand. This is verified by studying the first column of Fig. 4.3b where there is a complete overlap between the distributions of the human supplied prior and the Pareto solutions (Human-AI Pareto Set). The 157% improvement in the hypervolume implies that

the perfect human supplied prior has significant complementary effect on the optimizer (Human + AI) performance.

Insight 2: Imperfect human supplied prior

The solutions based on the MinTT probabilistic model when applied to problem instance 2 will cause Factory 1 to have insufficient production capacity when fulfilling all orders from Customer 1. However, during the course of the optimization process, this imperfectness is curbed through the adaptive transfer mechanism. This can be observed through the non-overlaps between the distributions of the human supplied prior and the Pareto solutions shown in the second column of Fig. 4.3b. The main differences in these 2 distributions are that 30-40% of Customer 1 order from Factory 1 (C1-F1) is reallocated to Factory 2 (C1-F2). Additionally, to prevent overloading Factory 2, 30-50% of Customer 2 order is also reallocated to Factory 3 (C2-F3). Nevertheless, in situations of imperfect human supplied prior, a significant 63% hypervolume improvement can still be achieved.

Insight 3: Flawed human supplied prior

The solutions generated from the MinTT probabilistic model is far from the Pareto solutions for problem instance 3 as shown in the third column of Fig. 4.3b. This is because the solutions from the human supplied prior will cause production shortages in Factory 1. In addition, it will cause Factory 3 to have a high percentage of unutilized production capacity. However, during the course of the optimization process, the human supplied prior is meticulously transferred while searching for a better reallocation plan. The main differences between the distributions of the human supplied prior and the Pareto solutions, are that 90% of Customer 1 order is reallocated from Factory 1 (C1-F1) to Factory 2 (C1-F2) and 3 (C1-F3). Furthermore, about 30-60% of Customer 2 order to Factory 2 (C2-F2) is reallocated to Factory

1 (C2-F1) to ensure a more balanced production loading. From the 24% hypervolume improvement, the transfer mechanism is able to adaptively adopt the complementary part of the human supplied prior while optimizing the remaining solutions from scratch.

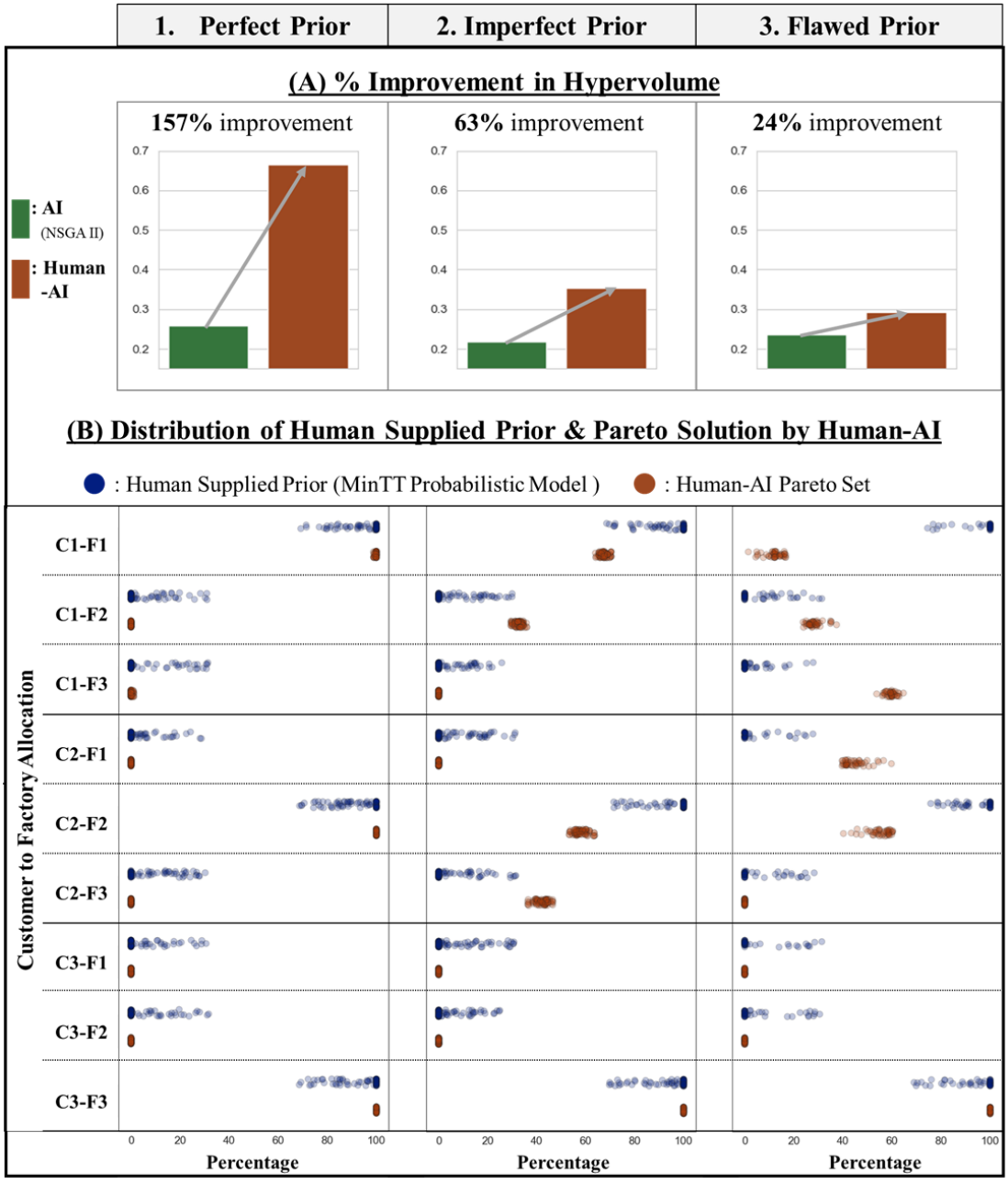

Figure 4.3: Top: Comparison of the hypervolume improvements. Bottom: distributions of the human supplied prior based on 1 standard deviation and Pareto solutions obtained by Human-AI on the allocation plan.

4.2.3 Result on Inverse Modeling

The independent *inv*GPs for post-hoc PS learning employ a squared exponential covariance function and are implemented using the GPyTorch library [126]. The training process involves 100 training iterations with a learning rate of 0.1. A total of $n = 50$ training data acquired from the the course of transfer optimization in the previous section., while a test set of $n_q = 1000$ query points (e.g., those supplied by a DM) not contained in the training data are used for performance evaluation.

Figure 4.4 illustrates the approximated PF before and after applying inverse modeling for problem instances 1 to 3. The inverse model demonstrates its capability to enhance decision making by producing a denser representation of the approximated PF. This enables DMs to conduct a more informed trade-off analysis between the two conflicting objectives.

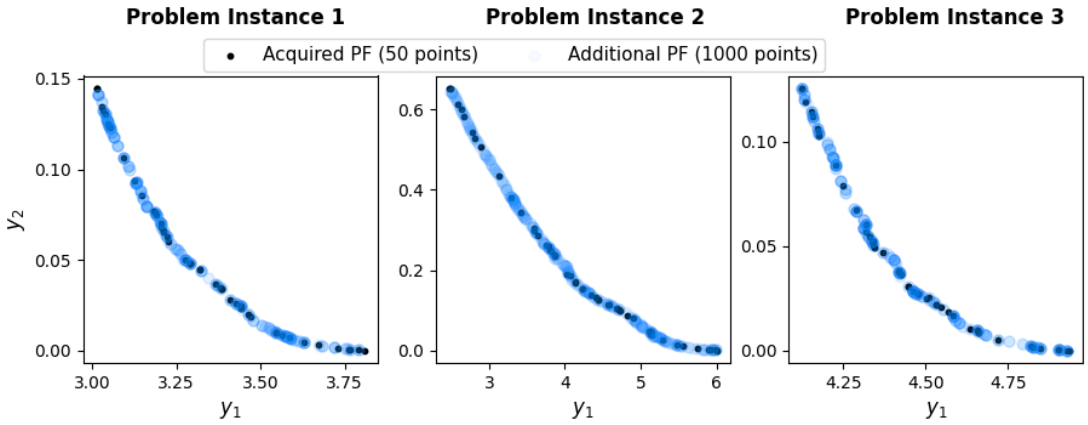

Figure 4.4: 2-D scatter plot of the acquired (in black) and additional PF (in blue) for problem instance 1 to 3. The x-axis and y-axis represents the average order fulfillment and unutilized capacity across the 3 factories respectively.

4.3 Chapter Summary

In this chapter, the potential of Human-AI omni-directional collaboration to significantly enhance both the quality and quantity of Pareto data are ex-

plored. The first section focuses on improving PF quality in terms of better convergence and diversity through emerging *transfer optimization methods* that leverage experiential priors to guide the search process. The second section focuses on increasing the quantity of Pareto data acquired through the concept of *post-hoc PS learning* for informed *a posterior* decision making.

However, the challenge of learning an accurate inverse model in expensive optimization domains, where evaluation data is scarce, remains unsolved. Therefore, in the remaining chapters, the focus is on proposing innovative approaches to enhance the performance of inverse models in problem domains characterized by limited training data, without imposing constraints on the workings of the underlying MOP solver.

Chapter 5

Pareto Set Learning across Common Objective Spaces

The Pareto data obtained from a single run of the MOP solver is inherently limited in expensive optimization domains, where evaluation data is at a premium. This challenge becomes more pronounced as the number of objectives increases, causing the number of points required to adequately cover the entire PF to scale exponentially. Consequently, this makes it nearly impossible to build an accurate inverse machine learning model to map preferred but unexplored regions along the front to the PS in decision space.

To address this small data challenge, this chapter introduces a pioneering study on multi-source inverse transfer learning to maximally leverage experiential *source* tasks to enhance PS learning in the *target* optimization task[1]. Information transfer between heterogeneous source-target pairs is uniquely enabled in the inverse setting through the unification provided by common objective spaces.

[1]The work in this chapter has been published in [123].

5.1 Harnessing Small Datasets in Pareto Set Learning

An accurate inverse model can offer significant benefits to a DM in controlled generation of desired PS solutions. However, the accuracy of $\boldsymbol{\psi}_{\boldsymbol{\theta}}^{-1}$ depends on the quality and quantity of available training data, which is inherently scarce/small in high-dimensional/expensive objective spaces. Hence, to overcome the challenge of limited data regimes, a novel *inverse transfer learning* method is proposed in this section.

Consider γ source datasets $\{D_{\mathcal{S}_1}, ..., D_{\mathcal{S}_\gamma}\}$ with $D_{\mathcal{S}_k} = \{W_{\mathcal{S}_k},\ X_{\mathcal{S}_k}\}$, $\forall k = 1, ..., \gamma$, alongside target data $D_{\mathcal{T}} = \{W_{\mathcal{T}}, X_{\mathcal{T}}\}$ derived from the optimization task at hand. It is assumed that these datasets originate from varied but related MOPs within a given application area, such that the unit hyperplane containing $W_{\mathcal{S}_k} \in \mathbb{R}^{n_{\mathcal{S}_k} \times m}$ and $W_{\mathcal{T}} \in \mathbb{R}^{n_{\mathcal{T}} \times m}$ may lie in a common objective space; i.e., $\mathcal{W}_{\mathcal{S}_k} = \mathcal{W}_{\mathcal{T}}$, $\forall k$. (A real-world exemplar of this is presented in Section 5.2.) Given high-dimensional/expensive objectives, the target data is inevitably sparse or small, whereas a sizeable *cumulative* volume of source data is deemed available from past problems solved (i.e., $n_{\mathcal{T}} << \sum_{k=1}^{\gamma} n_{\mathcal{S}_k}$ even if each $n_{\mathcal{S}_k}$ may be small). This motivates maximal utilization of information from the experiential sources to augment target.

Crucially, PS learning through a common objective space allows for information transfers in scenarios where the decision spaces $\mathcal{X}_{\mathcal{S}_k} \subset \mathbb{R}^{d_{\mathcal{S}_k}}$ and $\mathcal{X}_{\mathcal{T}} \subset \mathbb{R}^{d_{\mathcal{T}}}$ of a source and target task may differ. In particular, the dimensionality of the space could change (i.e., $d_{\mathcal{S}_k} \neq d_{\mathcal{T}}$) with some decision variables/dimensions being added (or removed) in the target MOP relative to the source [127]. The common objectives (which form the inputs to the inverse model) provide the necessary unification for transfer learning to occur even between such heterogeneous source-target pairs. For practicality, our proposed inverse learner models each decision variable independently; a useful implication of this is given in Subsection 5.1.2. Inverse transfer learning is

activated *only* between those source and target decision variables that bear the same physical meaning. We leverage this assumption to condense the exposition in subsequent subsections to only a single (the j^{th}) target variable $x_{\mathcal{T},j}$. An overlapping source decision variable bearing the same physical meaning is denoted as $x_{\mathcal{S}_k,j}$.

5.1.1 Inverse TGPS for Single-source Transfer

First consider standard (no-transfer) PS learning with stochastic, nonparametric GPs. Let the target data be $D_{\mathcal{T},j} = \{W_{\mathcal{T}}, X_{\mathcal{T},j}\}$ where $X_{\mathcal{T},j}$ represents the j^{th} column of $X_{\mathcal{T}}$. In this case, an *inv*GP model, from $\boldsymbol{w} \in \mathcal{W}$ to $x_{\mathcal{T},j} \in \mathbb{R}$, describes a distribution over functions as $\psi^{-1}(\boldsymbol{w}) \sim \mathcal{GP}(\mu(\boldsymbol{w}), k(\boldsymbol{w}, \boldsymbol{w}'))$, where $\mu(\boldsymbol{w})$ is the mean (typically set to a constant, zero) and $k(\cdot,\cdot)$ is some valid covariance function. The inverse map is thus a stochastic process wherein any finite subset of random variables follows a joint multivariate Gaussian distribution. Given the observations in $D_{\mathcal{T},j}$, the posterior predictive distribution at any query point $\boldsymbol{w}_q$ can then be analytically obtained [121].

In the transfer learning setting with a single source dataset $D_{\mathcal{S},j} = \{W_{\mathcal{S}}, X_{\mathcal{S},j}\}$, an *inv*TGP model can account for the similarity between the source and target tasks by extending the covariance function $k(\cdot,\cdot)$ as,

$$\tilde{k}_j(\boldsymbol{w}, \boldsymbol{w}') = \begin{cases} \lambda_j k(\boldsymbol{w}, \boldsymbol{w}'), & \text{if } \boldsymbol{w} \in W_{\mathcal{S}} \ \& \ \boldsymbol{w}' \in W_{\mathcal{T}} \\ & \text{or } \boldsymbol{w} \in W_{\mathcal{T}} \ \& \ \boldsymbol{w}' \in W_{\mathcal{S}}\,, \\ k(\boldsymbol{w}, \boldsymbol{w}'), & \textit{otherwise} \end{cases} \tag{5.1}$$

where $\tilde{k}_j(\cdot,\cdot)$ is referred to as the transfer kernel. λ_j is a measure of source-target correlation, with $|\lambda_j| \leq 1$ being a sufficient condition for the transfer kernel to be valid. As such, if $|\lambda_j|$ is learned to be close to 1, it indicates high relevance of the source to the target task, whereas λ_j close to zero signifies that the source may be unrelated to the target. In the geostatistics literature, this model corresponds to the *intrinsic coregionalization model*, a

specific case of *co-kriging* that uses only a single (scalar) λ to capture the inter-task similarity [128]. In contrast, the *linear model of coregionalization* from geostatistics may offer greater flexibility by using multiple kernels, but at the added cost of complicating model training and inference [129]. We therefore limit our implementation here to a scalar λ, achieving encouraging performance as shown in the experiments.

For posterior inference, the closed-form predicted mean and variance of the *inv*TGP at a query point $\boldsymbol{w}_q$ is given by,

$$\mu_j(\boldsymbol{w}_q) = \tilde{\boldsymbol{k}}_{\boldsymbol{w}_q}(\tilde{K} + \Lambda)^{-1} \begin{bmatrix} X_{\mathcal{S},j} \\ X_{\mathcal{T},j} \end{bmatrix}, \tag{5.2a}$$

$$\sigma_j^2(\boldsymbol{w}_q) = \tilde{k}(\boldsymbol{w}_q, \boldsymbol{w}_q) - \tilde{\boldsymbol{k}}_{\boldsymbol{w}_q}(\tilde{K} + \Lambda)^{-1}\tilde{\boldsymbol{k}}_{\boldsymbol{w}_q}^{\intercal}, \tag{5.2b}$$

where $\tilde{\boldsymbol{k}}_{\boldsymbol{w}_q}$ is the kernel vector between $\boldsymbol{w}_q$ and $W = \{W_{\mathcal{S}}, W_{\mathcal{T}}\}$ computed using the transfer kernel in Eq. (5.1), $\Lambda = \begin{bmatrix} \sigma_{\mathcal{S}}^2 I_{n_{\mathcal{S}}}, & 0 \\ 0, & \sigma_{\mathcal{T}}^2 I_{n_{\mathcal{T}}} \end{bmatrix}$ where $\sigma_{\mathcal{S}}^2$ and $\sigma_{\mathcal{T}}^2$ are the source and target noise terms, respectively, and $\tilde{K} = \begin{bmatrix} \tilde{K}_{\mathcal{SS}}, & \tilde{K}_{\mathcal{ST}} \\ \tilde{K}_{\mathcal{TS}}, & \tilde{K}_{\mathcal{TT}} \end{bmatrix}$ is the overall covariance matrix of the *inv*TGP. In $\tilde{K}$, $\tilde{K}_{\mathcal{SS}}$ and $\tilde{K}_{\mathcal{TT}}$ are the kernel matrices of the data in the source and target tasks, respectively. $\tilde{K}_{\mathcal{ST}}$ $(= \tilde{K}_{\mathcal{TS}}^{\intercal})$ is the kernel matrix across the data in the source and target datasets.

Parameter Learning

One way to learn the (hyper-)parameters of the *inv*TGP would be to consider the joint distribution of source and target tasks [130]. This may however cause the model to bias towards the source task when the volume of target data is less than that of the source. Thus, in this paper, a two-stage training process is employed instead. In the first stage, the parameters of the standard covariance function $k(\cdot, \cdot)$ and $\sigma_{\mathcal{T}}$ are learned based on the target data $D_{\mathcal{T},j}$

alone by maximizing,

$$-\frac{1}{2}\, X_{\mathcal{T},j}^{\intercal}\, (\tilde{K}_{\mathcal{T}\mathcal{T}} + \sigma_{\mathcal{T}}^2 I_{n_{\mathcal{T}}})^{-1}\, X_{\mathcal{T},j} - \frac{1}{2}\, \log(|\tilde{K}_{\mathcal{T}\mathcal{T}} + \sigma_{\mathcal{T}}^2 I_{n_{\mathcal{T}}}|) + const.$$

In the second stage, the parameters found for $k(\cdot,\cdot)$ are kept fixed while searching for λ_j and $\sigma_{\mathcal{S}}$ that optimize the following log marginal likelihood considering both the source and the target data,

$$-\frac{1}{2}\, \left[X_{\mathcal{S},j}^{\intercal},\ X_{\mathcal{T},j}^{\intercal}\right](\tilde{K} + \Lambda)^{-1} \begin{bmatrix} X_{\mathcal{S},j} \\ X_{\mathcal{T},j} \end{bmatrix} - \frac{1}{2}\, \log(|\tilde{K} + \Lambda|) + const.$$

Note that the training complexity of the second stage scales cubically with the size of the data, i.e., as $\mathcal{O}\big((n_{\mathcal{S}} + n_{\mathcal{T}})^3\big)$, due to the need for inversion and the determinant of $\tilde{K} + \Lambda$.

5.1.2 Product-of-invTGPs for Multi-source Transfer

The cubic complexity poses a major challenge while extending the TGP model to multi-source transfer learning since the total data size grows rapidly with the number of sources. A full TGP would additionally involve the modeling of correlations between *all* (source-target and source-source) task pairs, such that the number of parameters to be learned would grow as the square of the number of sources. This makes parameter optimization difficult as well.

To overcome these challenges, the factorized product-of-GP experts for alleviating the cubic training cost [131, 132] and arriving at a novel product-of-*inv*TGPs is adopted. A significant advantage of factorization is that it allows for massively distributed, decentralized computations in model training and posterior inference[2]. The *inv*TGPs learned for all source-target pairs form in-

[2]In addition to computational gains, the scope for decentralization has positive implications in terms of source data privacy. Privacy preservation is a vital research theme in artificial intelligence [122], with little prior work done from the perspective of multi-objective transfer optimization. Our proposed methodology takes an important stride forward in filling this gap.

dependent components that are efficiently trainable on distributed hardware. *As a useful aside, the assumed independence of target decision variables implies even greater scope for parallelization.* What's more, when limiting to sequential computations, the time complexity of the product-of-experts (PoE) scales only linearly with respect to the number of sources. Table 5.1 compares the computational complexity of *inv*TGPs and product-of-*inv*TGPs with and without parallel computations.

	Training	**Inference**
***inv*TGPs**	$\mathcal{O}\big((n_{\mathcal{T}} + \sum_{k=1}^{\gamma} n_{\mathcal{S}_k})^3\big)$	$\mathcal{O}\big((n_{\mathcal{T}} + \sum_{k=1}^{\gamma} n_{\mathcal{S}_k})^2\big)$
Product-of-*inv*TGPs	$\sum_{k=1}^{\gamma} \mathcal{O}\big((n_{\mathcal{T}} + n_{\mathcal{S}_k})^3\big)$	$\sum_{k=1}^{\gamma} \mathcal{O}\big((n_{\mathcal{T}} + n_{\mathcal{S}_k})^2\big)$
Product-of-*inv*TGPs (with Parallel Compute)	$\mathcal{O}\big((n_{\mathcal{T}} + n_{\mathcal{S}_{max}})^3\big)$	$\mathcal{O}\big((n_{\mathcal{T}} + n_{\mathcal{S}_{max}})^2\big)$

Table 5.1: Computational complexity of *inv*TGPs and Product-of-*inv*TGPs, with $n_{\mathcal{S}_{max}}$ denoting the largest source dataset.

Beyond computational gains, the PoE offers a principled fusion of individual *inv*TGP predictive distributions. This can be shown as follows. For the j^{th} target decision variable, let $\mu_{k,j}(\boldsymbol{w}_q)$ and $\sigma^2_{k,j}(\boldsymbol{w}_q)$ be the predicted mean and variance at query point $\boldsymbol{w}_q$ of the *inv*TGP trained (as per the procedure in Subsection 5.1.1) with the k^{th} source $D_{\mathcal{S}_k,j}$ and the target data $D_{\mathcal{T},j}$. The product of γ such Gaussian predictions is then proportional to a Gaussian with mean and variance given by,

$$\mu_{PoE,j}(\boldsymbol{w}_q) = \sigma^2_{PoE,j} \sum_{k=1}^{\gamma} \sigma^{-2}_{k,j}(\boldsymbol{w}_q)\mu_{k,j}(\boldsymbol{w}_q), \tag{5.3a}$$

$$\sigma^2_{PoE,j}(\boldsymbol{w}_q) = 1/\big(\sum_{k=1}^{\gamma} \sigma^{-2}_{k,j}(\boldsymbol{w}_q)\big). \tag{5.3b}$$

As indicated by Eq. (5.3), the PoE composes the final prediction taking into account each *inv*TGP's predictive uncertainty. Lower predicted variances

(indicating more confident/certain predictions) are more strongly weighted, leading to an intuitively sound fused prediction. Imagine a situation where a source k' results in an *inv*TGP whose predictive variance is large, such that $\sigma^{-2}_{k',j} << \sigma^{-2}_{k,j}, \forall k \neq k'$. This could happen if $\lambda_{k',j}$ is much smaller in magnitude than the source-target correlations uncovered by the other *inv*TGPs. In such cases, Eq. (5.3a) implies that the k' term will vanish from the PoE aggregation, providing a fused prediction that depends only on those *inv*TGPs that are confident at $\boldsymbol{w}_q$.

By replicating the PS learning and prediction procedure (as shown for the j^{th} variable) for all $d_{\mathcal{T}}$ target decision space dimensions, a complete solution $\boldsymbol{\mu}_{PoE}(\boldsymbol{w}_q)$ corresponding to query point $\boldsymbol{w}_q$ is constructed.

5.1.3 A Generalized Product-of-invTGPs

The product-of-*inv*TGPs offers both computational and predictive advantages in the multi-source transfer setting. However, as the number of source datasets (or *inv*TGPs) increases, Eq. (5.3b) implies that the predicted variance of the PoE would quickly drop to zero, suggesting overconfident predictions [133]. This is undesirable, as well-calibrated uncertainty-aware prediction is a key to rationalizable human-AI interaction [122]. *An overconfident prediction could mislead a DM into adopting a solution where the PoE is confident but wrong.* To alleviate this issue, a tunable parameter β can be introduced into Eq. (5.3) to form the following generalized PoE (gPoE) prediction,

$$\mu_{gPoE,j}(\boldsymbol{w}_q) = \sigma^2_{gPoE,j} \sum_{k=1}^{\gamma} \beta_k \sigma^{-2}_{k,j}(\boldsymbol{w}_q)\mu_{k,j}(\boldsymbol{w}_q), \tag{5.4a}$$

$$\sigma^2_{gPoE,j}(\boldsymbol{w}_q) = 1/\big(\sum_{k=1}^{\gamma} \beta_k \sigma^{-2}_{k,j}(\boldsymbol{w}_q)\big), \tag{5.4b}$$

where $\sum_{k=1}^{\gamma} \beta_k = 1$. In the current implementation, β_k i set to $1/\gamma$. This makes the aggregated mean in Eq. (5.4a) identical to Eq. (5.3a)—hence preserving the intuitively sound fused prediction—while preventing the predicted

variance in Eq. (5.4b) from degenerating to zero for large γ.

5.1.4 A Summary of Salient Features

Inverse transfer through common objective spaces is what enables post-hoc PS learning to maximally benefit from mutual information between heterogeneous source-target pairs. Here, a recap on some of the salient features of the proposed approach brought by the generalized product-of-*inv*TGPs, supporting PS learning in small data regimes are provided.

- *Computationally efficient multi-source transfer.* The method gives rise to a factorized training scheme where *inv*TGPs for all source-target pairs form independent components that are efficiently trainable on distributed hardware. Hence, given a fully parallel computation setup, the training complexity is limited only by the largest data size among all paired source-target datasets. The cubic complexity in the number of sources is overcome.

- *Uncertainty-aware fusion of predicted means.* The aggregation method of gPoE weights individual *inv*TGPs inversely to their predictive uncertainty. This leads to a fused prediction that depends more strongly on *inv*TGPs with low predicted variance (higher confidence), while adaptively weighing out those with large predicted variance.

- *Calibrated predicted variance.* The gPoE does not lead to overconfident predictions under increasing number of sources (*inv*TGP models), facilitating rationalizable human-AI interactions with models that know what they don't know.

5.2 Empirical Analysis

The generalized product-of-*inv*TGPs is implemented using the GPyTorch library [126]. The method is first verified on the pedagogical DTLZ 1-3 benchmarks [134], with slight modifications to synthetically create different source and target MOPs. Modified DTLZ 1-3 with 4 to 7 objective functions are used to analyse the performance of the method under: i) increasing levels of (target) data scarcity, ii) varying source-target similarity, and iii) multi-source transfer. A set of computationally expensive MOPs from the lightweight composites manufacturing domain are considered next. The use-case establishes the validity of the assumption (of common objective spaces) and the practical applicability of the method in augmenting post-hoc PS learning under small data by means of inverse transfer learning.

5.2.1 Evaluation Metrics

To evaluate the quality of post-hoc PS learning, two different metrics, namely, the *Inverted Generational Distance (IGD) Ratio* and the *Root Mean Square Error (RMSE)* are considered. The two metrics capture distinctive attributes of the candidate solutions generated from the perspective of a DM with postponed preferences.

The IGD Ratio adapted from [103] gives a broad understanding of the overall PF approximation capacity of PS learning. It quantifies the improvement in the quality of PF approximation before and after solutions generation via inverse model as,

$$IGD\ Ratio = \frac{IGD_b}{IGD_a}, \tag{5.5}$$

where IGD_b and IGD_a are the IGD values before and after the generation of additional solutions via PS learning, respectively. A ratio of 1 indicates that the PF approximation has not improved despite the generation of additional points, while a value greater than 1 provides a scalar indicator of the relative

convergence and diversity improvement. Values less than 1 do not occur as IGD_a combines the generated points with the initial points along the approximated PF. It is important to note that the IGD is a measure of the Euclidean distance between elements in the approximated PF and the true PF [112];

$$IGD = \frac{1}{|Y^*|} \sum_{q=1}^{|Y^*|} \min\{||\boldsymbol{y}_q^* - \boldsymbol{y}_1||_2, ..., ||\boldsymbol{y}_q^* - \boldsymbol{y}_{n_q}||_2\}, \tag{5.6}$$

where $Y^* = \{\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_{n_q}^*\}$ is a set of n_q well-distributed reference points along the true PF and $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n_q}$ are the set of approximate points generated as $\boldsymbol{y}_q = \boldsymbol{f}(\boldsymbol{\mu}_{gPoE}(\boldsymbol{w}_q))$. A lower IGD is clearly better.

In contrast to the IGD Ratio, the RMSE provides a more fine-grained evaluation of the accuracy of PS learning on a test set of n_q query points (e.g., those supplied by a DM) not contained in the training data. For benchmark functions whose analytical expressions are known, the RMSE value is measured in the objective space as per (5.7a). The error thus quantifies how closely the inverse model is able to satisfy specific DM preferences articulated in the objective space. On the other hand, calculating exact objective function values for predicted solutions in real-world MOPs can call for expensive evaluations. To avoid this, the RMSE can be measured in decision space instead, as per (5.7b). The latter is meaningful when we consider a smooth one-to-one mapping between the PS and the PF. The two instantiations of the RMSE are stated as,

$$RMSE_{\boldsymbol{f}} = \sqrt{\frac{\sum_{q=1}^{n_q} ||\boldsymbol{y}_q^* - \boldsymbol{y}_q||_2^2}{n_q}}, \tag{5.7a}$$

$$RMSE_{\boldsymbol{x}} = \sqrt{\frac{\sum_{q=1}^{n_q} ||\boldsymbol{x}_q^* - \boldsymbol{x}_q||_2^2}{n_q}}, \tag{5.7b}$$

where $\boldsymbol{x}_q^*$ and $\boldsymbol{x}_q$ are the true and predicted solutions, respectively, given the q^{th} query/test point $\boldsymbol{w}_q$. Note, the predicted mean of the product-of-*inv*TGPs is taken as its point estimate for accuracy evaluation, i.e., $\boldsymbol{x}_q = \boldsymbol{\mu}_{gPoE}(\boldsymbol{w}_q)$.

5.2.2 Results on Modified DTLZ Benchmarks

We begin by modifying the DTLZ 1-3 benchmarks (denoted as DTLZ 1a - 3a) to create different problem instances with heterogeneous decision spaces. These problems make up source and target MOPs with common objective spaces and PF topology, but with varying characteristics of the PS. DTLZ 1a-3a take the general form [135] of,

$$\begin{aligned} \min_{\boldsymbol{x}_I, \boldsymbol{x}_{II}} \quad & \boldsymbol{f}\big(\boldsymbol{x}, s, g(\boldsymbol{x}_{II})\big) = \big[f_1\big(\boldsymbol{x}_I, s, g(\boldsymbol{x}_{II})\big), ..., f_m(\boldsymbol{x}_I, s, g(\boldsymbol{x}_{II}))\big], \\ & s.t. \quad 0 \leq x \leq 1, \ \forall x \in \{\boldsymbol{x}_I, \boldsymbol{x}_{II}\}, \end{aligned} \tag{5.8}$$

where m is the number of objectives to be minimized. d is the total number of decision variables constituting $\boldsymbol{x}_I = [x_1, ..., x_{m-1}]$ and $\boldsymbol{x}_{II} = [x_m, ..., x_d]$ with $d \geq m$, and s changes the distribution of the non-dominated solutions.

The objective values of DTLZ 1a are given by Eq. (5.9a) while those of DTLZ 2a and 3a are given by Eq. (5.9b);

$$\begin{bmatrix} f_1 \\ f_2 \\ ... \\ f_{m-1} \\ f_m \end{bmatrix}^\intercal = 0.5\big(1 + g(\boldsymbol{x}_{II})\big) \begin{bmatrix} {x_1}^s \ {x_2}^s ... \ {x_{m-1}}^s \\ {x_1}^s \ {x_2}^s ... \ (1 - {x_{m-1}}^s) \\ ... \\ {x_1}^s \ (1 - {x_2}^s) \\ (1 - {x_1}^s) \end{bmatrix}^\intercal, \tag{5.9a}$$

$$\begin{bmatrix} f_1 \\ f_2 \\ ... \\ f_{m-1} \\ f_m \end{bmatrix}^\intercal = \big(1 + g(\boldsymbol{x}_{II})\big) \begin{bmatrix} cos(\frac{{x_1}^s \pi}{2}) ... \ cos(\frac{{x_{m-2}}^s \pi}{2}) \ cos(\frac{{x_{m-1}}^s \pi}{2}) \\ cos(\frac{{x_1}^s \pi}{2}) ... \ cos(\frac{{x_{m-2}}^s \pi}{2}) \ sin(\frac{{x_{m-1}}^s \pi}{2}) \\ cos(\frac{{x_1}^s \pi}{2}) ... \ sin(\frac{{x_{m-1}}^s \pi}{2}) \\ ... \\ sin(\frac{{x_{m-1}}^s \pi}{2}) \end{bmatrix}^\intercal, \tag{5.9b}$$

where s is set to 1 for all target MOPs, and $s \in (0, 1)$ for source MOPs to simulate different degrees of source-target similarity. A value of s closer to 1 indicates higher similarity.

The function $g(\boldsymbol{x}_{II})$ in Eq. (5.9) is given by Eq. (5.10a) for DTLZ 2a and Eq. (5.10b) for DTLZ 1a and 3a;

$$g(\boldsymbol{x}_{II}) = \sum_{x_j \in \boldsymbol{x}_{II}} (x_j - p_j)^2, \tag{5.10a}$$

$$g(\boldsymbol{x}_{II}) = 100\,|\boldsymbol{x}_{II}| \sum_{x_j \in \boldsymbol{x}_{II}} \big[(x_j - p_j)^2 - cos\big(2\pi(x_j - p_j)\big)\big], \tag{5.10b}$$

where $p_j = 0.5$ for all target MOPs, and $p_j = \frac{j-|\boldsymbol{x}_I|}{k|\boldsymbol{x}_{II}|}$ for all source MOPs $k = 1, 2, ..., \gamma$.

To produce the source and target datasets for DTLZ 1a-3a, the NSGA-III algorithm from the pymoo library [136] is run to generate the PF and PS approximations. All results of post-hoc PS learning are averaged over 20 runs of GP training with the squared exponential covariance function optimized by Adam [137]. We consider heterogeneous source and target MOPs with $d_{\mathcal{S}} = 10$ and $d_{\mathcal{T}} = 12$ decision variables. Table 5.2 shows the experimental settings where the amount of source data (per source MOP) is about twice that of available target data. The set of n_q query/test points of potential interest to a DM are evenly spaced along the projected hyperplane in the objective space. n_q is relatively large, allowing for rigorous evaluation of Pareto approximation capacity as indicated by the IGD Ratio.

DTLZ 1a-3a	**7-Obj**	**6-Obj**	**5-Obj**	**4-Obj**
$n_{\mathcal{S}}$	413	498	246	236
$n_{\mathcal{T}}$	161	246	135	108
n_q	5005	2002	1820	1330

Table 5.2: Experiment settings used for the size of the source data ($n_{\mathcal{S}}$), the target data ($n_{\mathcal{T}}$), and the number of query points (n_q) employed for testing PS learning on the DTLZ 1a-3a benchmarks with 4 to 7 objective functions.

Impact of Target Data Scarcity on Pareto Set Learning

The effect of small target data in high-dimensional optimization domains is illustrated on DTLZ 1a-3a with 4 and 7 objectives. The experimental setup,

including the configuration of source and target problems, follows the settings outlined in Table 5.3. To assess the effect on PS learning quality, the target data is progressively reduced to 50% and 25% for each problem instance.

DTLZ 1a-3a	**# Data Points**	**7-Obj**	**4-Obj**
Source	$n_{\mathcal{S}} =$	413	236
$(s = 0.9, d_{\mathcal{S}} = 10)$			
	$n_{\mathcal{T}} =$	161	108
Target	$50\% \cdot n_{\mathcal{T}} =$	80	54
$(s = 1, d_{\mathcal{T}} = 12)$	$25\% \cdot n_{\mathcal{T}} =$	40	27
	$n_q =$	5005	1330

Table 5.3: Target data size settings for analyzing the impact of data scarcity on PS learning in DTLZ 1a–3a benchmarks with 4 and 7 objective functions.

From Fig. 5.1, a monotonic worsening (increasing) trend is observed in the RMSE value as the amount of target data is decreased. This is not surprising. Interestingly, Fig. 5.1 shows that by transfer learning from a correlated source MOP with $s = 0.9$, the *inv*TGP is able to resist the negative effects of data scarcity to a large extent. In particular, the RMSE is lowered by up to ∼50% when compared to the *inv*GP with no transfer.

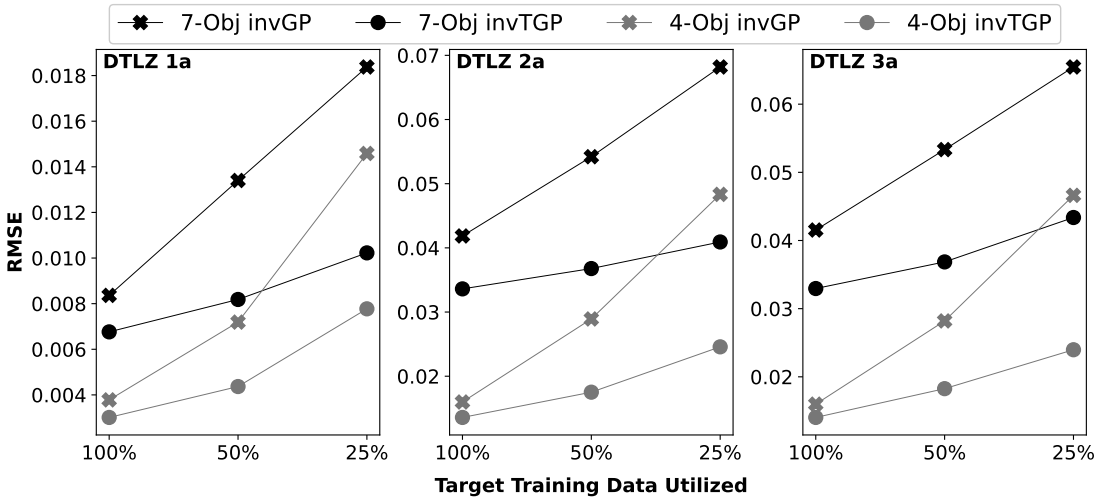

Figure 5.1: Accuracy of PS learning measured in $RMSE_{\boldsymbol{f}}$ (y-axis) for different amounts of target training data utilized (x-axis). Results for DTLZ 1a to 3a with 7 objectives (black line) or 4 objectives (grey line) are presented. The marker "o" and "x" represent PS learning by *inv*TGP with single-source transfer and *inv*GP without transfer, respectively.

Effect of Source-Target Similarity

The second set of experiments for DTLZ 1a-3a aims at investigating the performance of *inv*TGP under different levels of source-target similarity, compared against the baseline case of *inv*GP with no transfer. The source-target similarity levels and source dataset sizes are configured as specified in Table 5.4.

7-Obj DTLZ 1a-3a	**Source-Target Similarity**, s	
Source $(d_{\mathcal{S}} = 10, n_{\mathcal{S}} = 413)$	Very High	0.9
	High	0.75
	Mid	0.5
	Low	0.125
Target $(d_{\mathcal{T}} = 12, n_{\mathcal{T}} = 161, n_q = 5005)$	-	1

Table 5.4: Source-target similarity settings for analyzing the impact on PS learning in DTLZ 1a–3a benchmarks with 7 objective functions.

The quality of solutions generated via PS learning, measured by the IGD Ratio and the RMSE value are depicted in Figure 5.2. From the results, not only does the *inv*TGP outperform the *inv*GP, but also as the source-target similarity increases, the quality of the generated solutions tends to improve consistently for the *inv*TGP. This improvement makes intuitive sense and indicates that the *inv*TGP successfully leverages the correlation between the target task and the different source MOPs, transferring the external information weighted by λ_j in (5.1) to augment its performance.

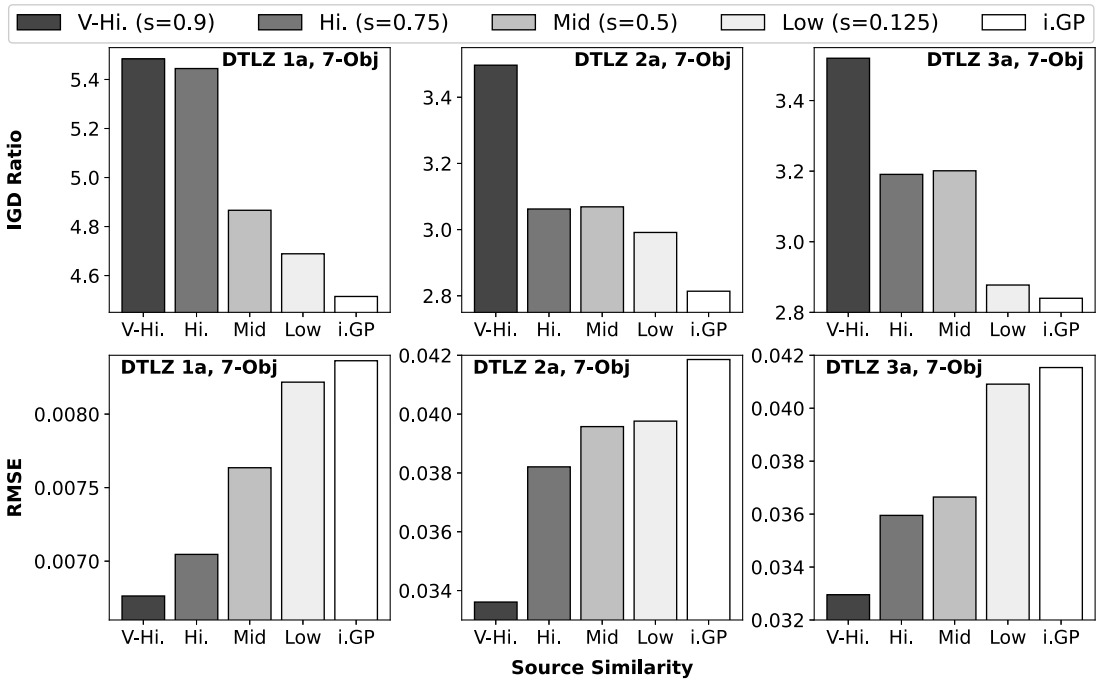

Figure 5.2: Performance of PS learning measured in IGD Ratio (y-axis, top row) and $RMSE_{\boldsymbol{f}}$ value (y-axis, bottom row) with different levels (s) of source-target similarity (x-axis). i.GP refers to the baseline *inv*GP with no transfer; all other results are from the *inv*TGP.

Utilizing Multi-source Transfers

The final set of experiments with benchmark functions investigates the performance of the generalized product-of-*inv*TGPs under multi-source transfer. The number of source datasets available for transfer varies from 1 to 3, with source-target similarity configured according to Table 5.5. The sizes of source and target datasets follow the specifications in Table 5.2. The number of decision variables is fixed at $d_{\mathcal{S}} = 12$ for all source data and $d_{\mathcal{T}} = 10$ for target data.

DTLZ 1a-3a (with 4-Obj to 7-Obj)	**Source-Target Similarity**, s		
	0.9	0.75	0.5
Without transfer	-	-	-
1-Source transfer	-	-	✓
2-Source transfer	-	✓	✓
3-Source transfer	✓	✓	✓

Table 5.5: Source-target similarity settings for analyzing the impact of varying source transfers on PS learning in DTLZ 1a–3a benchmarks with 4 to 7 objective functions.

Given a high 7-D objective space, Fig. 5.3 shows that the performance of the model improves substantially when additional data from source MOPs with larger source-target correlation are introduced. Note that in most practical situations, inter-task correlations would not be known beforehand. Hence, an important property of an effective transfer learning algorithm is to be able to selectively exploit useful information sources without the need for a human in the loop, while curbing harmful negative transfer from unrelated data. The aggregation equations Eq. (5.3a) and Eq. (5.4a) suggest this to be the case in theory. The experimental results substantiate that the model is indeed able to fuse information from all available sources to construct more accurate predicted solutions.

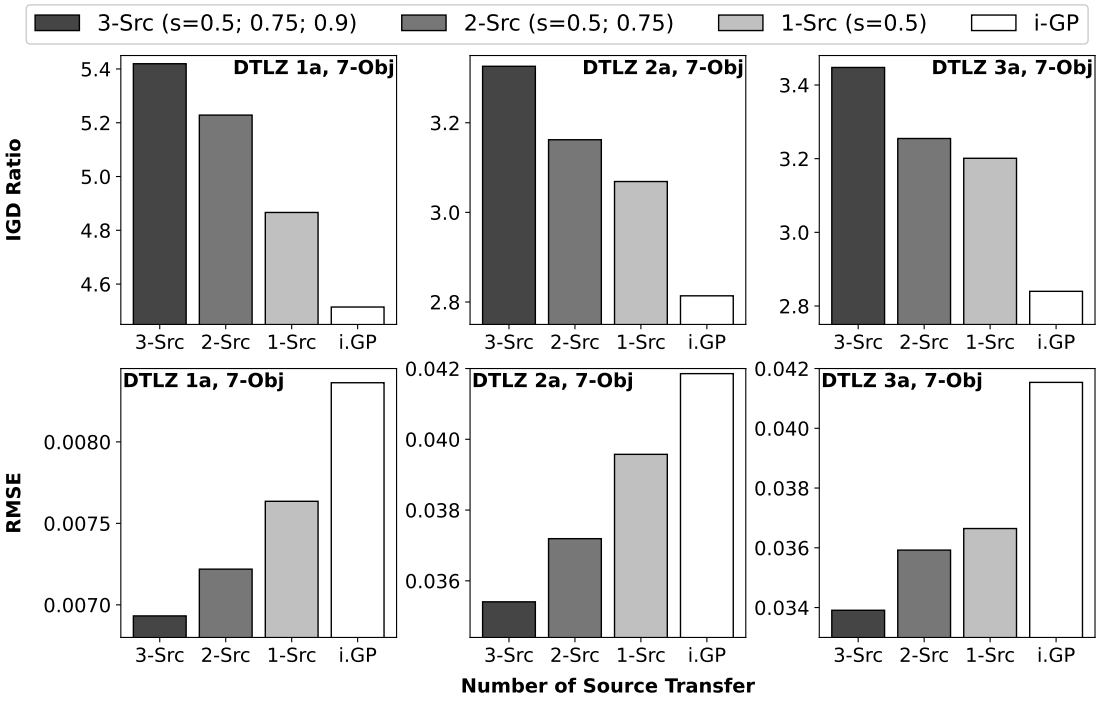

Figure 5.3: Performance of PS learning measured in IGD Ratio (y-axis, top row) and $RMSE_{\boldsymbol{f}}$ value (y-axis, bottom row) with decreasing number of source MOPs (x-axis) for DTLZ 1a to 3a with 7 objectives. i.GP refers to the baseline *inv*GP with no transfer.

The complete experiments for DTLZ 1a-3a with 4 to 7 objective functions are prensented in Tables 5.6 and 5.7. The findings highlight that the product-of-*inv*TGPs often leads to superior PS learning. Interestingly, monotonically improving performance is observed here as the number of source MOPs increases. Table 5.6 includes yet another commonly used inverse machine learning model, namely, the inverse radial basis function neural network (*inv*RBFNN), as a baseline for comprehensive comparison. The network structure and hyperparameters of the *inv*RBFNN were implemented according to the specifications in [103]. The *inv*RBFNN was found to under-perform relative to the *inv*GP and hence has been left out from the engineering case-study presented next.

Pareto Estimator	**# Objective Functions**			
	7-Obj	**6-Obj**	**5-Obj**	**4-Obj**
***inv*TGPs**	**IGD Ratio for DTLZ 1a**			
3 Sources	**5.42** (0.01)	**7.51** (0.01)	**7.98** (0.02)	**12.13** (0.16)
2 Sources	5.23 (0.01)	7.29 (0.01)	7.50 (0.02)	10.62 (0.10)
1 Source	4.87 (2e-3)	6.90 (0.01)	6.96 (0.01)	9.29 (0.10)
***inv*GP**	4.51 (5e-8)	6.95 (1e-7)	6.83 (2e-7)	8.55 (4e-7)
***inv*RBFNN**	3.57 (2e-6)	1.42 (4e-5)	1.30 (9e-6)	1.10 (1e-4)
***inv*TGPs**	**IGD Ratio for DTLZ 2a**			
3 Sources	**3.33** (0.02)	**4.33** (0.03)	**5.72** (0.04)	**6.46** (0.02)
2 Sources	3.16 (0.02)	4.04 (0.03)	5.36 (0.02)	5.66 (0.01)
1 Source	3.07 (0.02)	4.00 (0.02)	5.22 (0.01)	5.56 (0.12)
***inv*GP**	2.81 (1e-07)	3.63 (2e-07)	5.19 (3e-07)	5.48 (4e-07)
***inv*RBFNN**	1.66 (2e-10)	1.70 (6e-8)	2.68 (9e-8)	2.29 (7e-4)
***inv*TGPs**	**IGD Ratio for DTLZ 3a**			
3 Sources	**3.45** (0.02)	**4.07** (0.04)	**5.16** (0.01)	**6.22** (0.03)
2 Sources	3.25 (0.02)	3.83 (0.07)	5.13 (0.01)	5.99 (0.04)
1 Source	3.20 (0.02)	3.43 (0.01)	4.70 (4e-3)	5.85 (0.06)
***inv*GP**	2.84 (1e-7)	3.27 (0.01)	4.26 (3e-7)	5.55 (4e-7)
***inv*RBFNN**	1.72 (6e-10)	1.41 (5e-7)	1.63 (1e-6)	1.21 (6e-4)

Table 5.6: Performance of PS learning measured in IGD Ratio given 1 source ($s = 0.5$), 2 sources ($s = 0.5, 0.75$) or 3 sources ($s = 0.5, 0.75, 0.9$) for transfer. Values in bold mark the best averaged performance for a given target MOP over 20 independent runs. Values in brackets represent standard deviations in performance over these runs.

Pareto Estimator	**# Objective Functions**			
	7-Obj	**6-Obj**	**5-Obj**	**4-Obj**
***inv*TGPs**	**RMSE** ·1E-2 **for DTLZ 1a**			
3 Sources	**0.69** (1E-3)	**0.46** (3E-4)	**0.45** (9E-4)	**0.24** (3E-3)
2 Sources	0.72 (7E-4)	0.48 (6E-4)	0.47 (1E-3)	0.27 (3E-3)
1 Source	0.76 (3E-4)	0.49 (4E-4)	0.52 (9E-4)	0.33 (0.01)
***inv*GP**	0.84 (1E-8)	0.48 (6E-9)	0.56 (8E-8)	0.38 (3E-8)
***inv*TGPs**	**RMSE** ·1E-2 **for DTLZ 2a**			
3 Sources	**3.54** (0.02)	**2.29** (0.01)	**1.85** (0.01)	**1.33** (2E-3)
2 Sources	3.72 (0.02)	2.43 (0.02)	1.95 (4E-3)	1.48 (2E-3)
1 Source	3.96 (0.05)	2.46 (0.01)	2.02 (4E-3)	1.54 (0.02)
***inv*GP**	4.19 (1E-7)	2.77 (1E-7)	2.10 (9E-8)	1.60 (1E-7)
***inv*TGPs**	**RMSE** ·1E-2 **for DTLZ 3a**			
3 Sources	**3.39** (0.01)	**2.46** (0.03)	**2.05** (4E-3)	**1.41** (0.01)
2 Sources	3.59 (0.02)	2.60 (0.04)	2.07 (4E-3)	1.46 (0.01)
1 Source	3.66 (0.02)	2.85 (0.05)	2.22 (1E-3)	1.46 (0.01)
***inv*GP**	4.15 (1E-7)	3.05 (5E-8)	2.51 (1E-7)	1.60 (1E-7)

Table 5.7: Performance of PS learning measured in $RMSE_f$ value given 1 source ($s = 0.5$), 2 sources ($s = 0.5, 0.75$) or 3 sources ($s = 0.5, 0.75, 0.9$) for transfer. Values in bold mark the best averaged performance for a given target MOP over 20 independent runs. Values in brackets represent standard deviations in performance over these runs.

5.2.3 A Multidisciplinary Design Use-case

Here, the generalized product-of-*inv*TGPs model is applied to a practical use-case in the manufacturing of lightweight fiber-reinforced polymer (FRP) composites. Two distinct manufacturing techniques are considered, naturally forming source and target tasks in a transfer learning setting; detailed descriptions of these techniques can be found in the work by Gupta [138]. The first, labelled resin transfer moulding (RTM), involves placing a fibrous reinforcement inside a mould cavity whose geometry is precisely machined according to the FRP part to be produced. The mould is completely closed at the start of the manufacturing cycle, fully compressing the dry fibres to the desired fibre volume fraction. The mould is then heated to an operation temperature at which liquid thermosetting resin is injected into it at high pressure until the cavity is filled. After mould filling, the part rests

and cures under controlled temperature until the liquid resin sufficiently solidifies. The two phases (filling and curing) of the manufacturing cycle form a multidisciplinary design problem, deeply coupled by the thermal conditions induced in the part at the end of filling. A candidate process design is therefore evaluated by first running the mould filling simulation code, the output of which gives the initial thermal condition for the curing simulation.

Compression resin transfer moulding (CRTM) is an alternate technique that can shorten manufacturing cycle time but usually at the cost of larger peripheral equipment. This is achieved by a slight modification to the filling phase of the RTM cycle. Specifically, in CRTM, the mould is only partially closed before resin injection, reducing the resistance to the resin's flow. Full closure to the final fibre volume fraction occurs after fibre wetting with the required volume of liquid resin. The need for larger equipment (e.g., hydraulic press) thus originates from having to jointly compress the resin + fibre system.

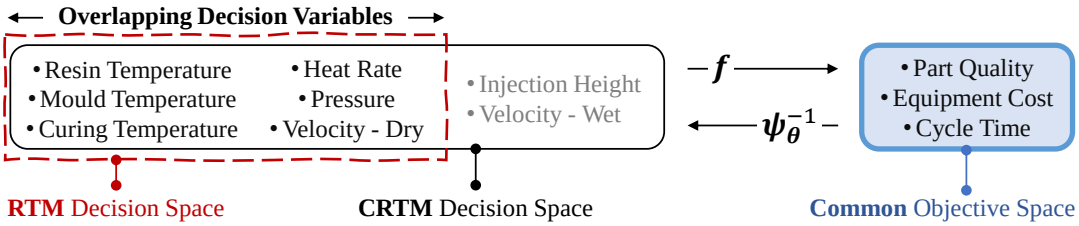

Figure 5.4: The common 3-D objective space and the overlapping decision variables of the heterogeneous RTM and CRTM manufacturing processes.

Despite the difference in the design (and hence the decision space) of the RTM and CRTM processes, their objective functions from a manufacturing standpoint are identical. In both cases the goal is to maximize *part quality* while minimizing *equipment cost* and *cycle time*, forming MOPs with 3-D objective spaces as descried in [26]. The finite element simulation codes for approximating these objectives are generally expensive, allowing small but high-quality data to be generated. The scenario thus perfectly encompasses the assumptions made in this paper. Fig. 5.4 illustrates the common objective space and the heterogeneous but overlapping decision spaces of the MOPs

under consideration. The six overlapping decision variables pertain to the thermal conditions of the resin and the mould (namely, *Resin Temperature, Mould Temperature, Heat Rate, Curing Temperature*), liquid injection pressure (*Pressure*) and the dry fibre compression velocity (*Velocity - Dry*). CRTM introduces two additional decision variables, namely, the *Injection Height* of the mould prior to resin injection and the wet fibre compression velocity (*Velocity - Wet*).

MOPs arising from the manufacture of FRP parts of circular geometry made of glass-fibre reinforced epoxy is considered. The plates are of 1 m diameter with a central injection hole of 20 mm. The final part fibre volume fraction is either 35% or 40%. By accounting for two different manufacturing processes we get a total of four MOPs: R35, R40, C35, and C40. Here R represents RTM, C represents CRTM, and the numerical value represents the part's final fibre volume fraction. At the end of multi-objective optimization runs for each task, datasets containing 500 optimized solution samples are collected. For assessing post-hoc PS learning, the target dataset is further divided into training and testing splits of 10 and 490 points, respectively, serving as an example of machine learning under expensive and extremely small data. The amount of source data (per source MOP) is taken to be 50 points. Given the computational expense of running evaluations at a large number of query points, only the $RMSE_{\boldsymbol{x}}$ value on the test set is used as the metric for comparison herein.

Table 5.8 shows the accuracy of PS learning under different source-target combinations. The high degree of overlap in the objective and decision spaces of related manufacturing tasks intuitively suggests the existence of transferrable information between them. It is therefore not surprising that both single-source and multi-source transfer learning with *inv*TGPs show benefits over the standard *inv*GP model trained only on limited target data. In the case of R35 as target task, a reduction in RMSE of up to $\sim$17% is achieved as a consequence of transfer. Unlike in the case of benchmark functions,

the best averaged performance in Table 5.8 is not achieved when all source data is utilized for multi-source transfers. This observation warrants future investigation. *It is however striking that multi-source transfer always leads to significantly better predictions than the least performant single-source invTGPs*, thus motivating joint utilization of all available sources in practical scenarios where source-target correlations may be *a priori* unknown.

	RMSE (·1e-2) on Target Problem			
Source Task(s)	**C35**	**C40**	**R35**	**R40**
No Transfer	19.36 (8e-15)	22.56 (1e-14)	25.97 (6e-15)	28.68 (6e-15)
C35	-	21.49 (8e-2)	25.13 (4e-2)	28.02 (1e-1)
C40	18.41 (7e-2)	-	26.10 (2e-1)	**25.73** (2e-1)
R35	18.20 (3e-2)	21.99 (9e-2)	-	27.4 (3e-1)
R40	**18.02** (3e-2)	21.82 (2e-1)	**21.47** (5e-1)	-
C35-C40	-	-	25.51 (8e-2)	25.87 (1e-1)
C35-R35	-	21.61 (7e-2)	-	26.93 (2e-1)
C35-R40	-	**21.44** (1e-1)	22.98 (3e-1)	-
C40-R35	18.26 (4e-2)	-	-	26.38 (2e-1)
C40-R40	18.12 (6e-2)	-	23.37 (3e-1)	-
R35-R40	18.08 (3e-2)	21.87 (1e-1)	-	-
All	18.13 (4e-2)	21.59 (1e-1)	23.87 (2e-1)	26.29 (2e-1)

Table 5.8: Performance of PS learning measured in $RMSE_{\boldsymbol{x}}$ value for the composite part manufacturing use-case. The values in bold mark the best averaged performance for a given target MOP over 20 independent runs. Strikingly, multi-source transfer utilizing all sources (last row of the table) always leads to significantly lower RMSE than the least performant single-source *inv*TGPs.

5.3 Chapter Summary

This chapter takes an important step towards effective Human-AI interactions in multi-objective decision making, particularly in high-dimensional/expensive optimization domains characterized by data scarcity. To this end, a novel methodology for PS learning under small data to recover non-dominated solutions along sparsely populated PFs is proposed. The proposed method is the first to explore the concept of multi-source, inverse transfer Gaussian processes (*inv*TGPs) for post-hoc PS learning, leveraging MOPs with com-

mon objective spaces to maximally utilize information between heterogeneous source-target pairs. To avoid computational bottlenecks arising from a large number of source datasets, a factorized product-of-experts procedure is put forth. The advantage of the adapted product-of-experts is that it not only facilitates massively distributed training, but also gives rationalizable predictive distributions that fuse together *inv*TGPs drawn from multiple sources to augment PS learning in the *target* optimization task at hand.

The resulting product-of-*inv*TGPs model is put through extensive empirical tests. Experiments are carried out on modified DTLZ benchmarks as well as on practical MOPs with computationally expensive, multidisciplinary evaluation data. The results obtained are promising and clearly highlight the benefits of jointly utilizing all available source datasets for transfer, especially in complex real-world scenarios where source-target correlations may not be known beforehand.

A major focus of this chapter has been on post-hoc PS learning in high-dimensional objective spaces that lead to sparse PF approximations. In the next chapter, the curse of dimensionality in decision space is considered, with dimensionality reduction techniques (to discover low-dimensional, piecewise continuous manifolds on which Pareto optimal solutions tend to lie [91]) for efficient learning of the inverse model(s).

Chapter 6

Pareto Set Representation Learning for High-dimensional Decision Space

To address the curse of dimensionality in the decision space that impedes the development of efficient inverse models, this chapter introduces the concept of *PS representation learning*[1]. This approach reduces the problem to its smallest possible dimensions while accurately capturing the Pareto optima. A denoising autoencoder is employed to discover a compressed latent representation of a sparsely populated PS by leveraging its unique bottleneck architecture. This representation then serves as the basis for creating compact inverse models that map points from the PF in objective space to the dimensionally reduced PS in decision space.

6.1 Proposed Representation Learning Method

For an m-objective continuous optimization problem, learning of a low-dimensional representation of the PS is motivated by the observation that under certain smoothness assumptions, the KKT conditions imply that its PS is a piecewise continuous $(m-1)$-dimensional manifold in the decision space [123]. This holds even when the dimensionality d of the decision space is high, i.e., $d \gg m$.

[1]The work in this chapter has been published in [139].

In this section, a DAE with a bottleneck layer to discover the compressed PS representation as illustrated in Fig. 6.1 is put forward. First, let the non-dominated solutions acquired from a run of an optimization algorithm be $X \in \mathbb{R}^{n \times d}$, where n is the total number of solutions. The training procedure of the DAE is initiated by corrupting each point $\mathbf{x}_h$ in X with isotropic Gaussian noise $\boldsymbol{z} \in \mathbb{R}^d$ with zero mean. The task of the DAE is hence to restore the corrupted PS data by minimising,

$$loss(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_{l=1}^{n \cdot n_p} \sum_{h=1}^{n} \left| \boldsymbol{h}_{\boldsymbol{\theta}}\left(\boldsymbol{g}_{\boldsymbol{\phi}}(\boldsymbol{x}_h + \boldsymbol{z}_l)\right) - \boldsymbol{x}_h \right|_2^2, \tag{6.1}$$

where, n_p is the number of perturbations introduced to each PS data, $\boldsymbol{g}_{\boldsymbol{\theta}}$ and $\boldsymbol{h}_{\boldsymbol{\phi}}$ are the encoding and decoding functions of the DAE, respectively, with $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ being the corresponding model parameters to be optimized.

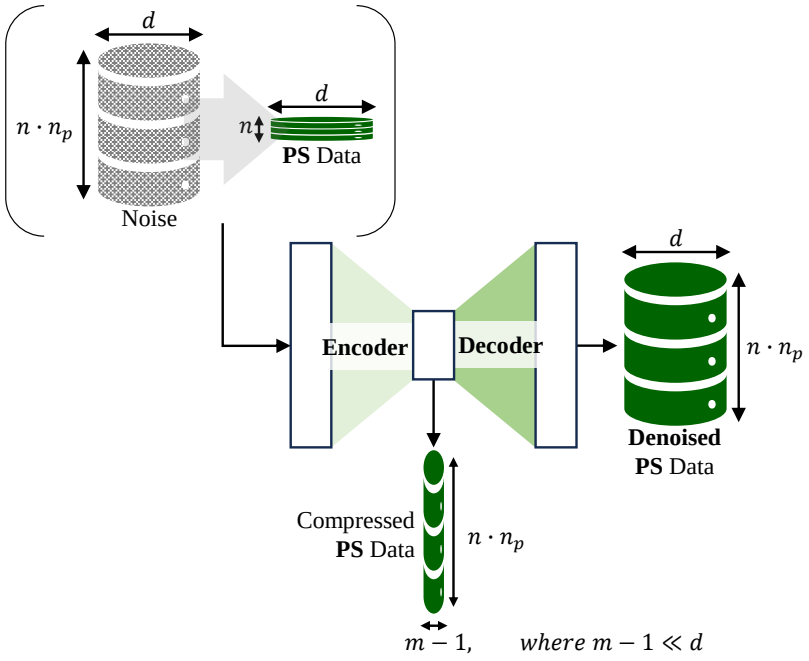

Figure 6.1: The proposed workflow of PS representation learning for reducing a problem to its smallest possible dimensions through the bottleneck layer of a DAE.

The learning efficacy of the DAE is first demonstrated on a benchmark test problem DTLZ-5 [134] with $m = 3$ objectives and $d = 500$ decision variables. The decoder and encoder of the DAE are configured to each have 1 hidden

layer with 10 neurons, while the bottleneck layer contains 2 neurons. *The highly reduced size of the bottleneck layer is informed by the theoretical conclusion drawn from the KKT conditions, that the PS must be an* $(m-1)$*-dimensional piecewise continuous manifold. Therefore, in general (discounting edge cases such as those with a degenerate PF or PS), choosing the bottleneck layer to be* $(m-1)$*-dimensional shall provide the most succinct representation of the PS possible.* The activations of all neurons in the DAE are set to the sigmoid function. The amount of PS data is $n = 100$, with each data point repeatedly corrupted $n_p = 10$ times by isotropic Gaussian noise of covariance scale 0.05. The trained DAE[2] is then tasked to denoise 1000 random solutions, following the procedure in [141], where each decision variable is sampled uniformly from 0 to 1. The denoised output closely approximates $x_{3,\ldots,500} = 0.5$ of the Pareto optimal solutions of DTLZ-5, with a root mean squared error of 1E-4. Fig. 6.2 depicts the uniformly drawn samples (in red) and the DAE's output (in purple) for the first three decision variables, showcasing its proficiency in discovering the low-dimensional 2-D manifold (a plane in this example) formed by the PS data.

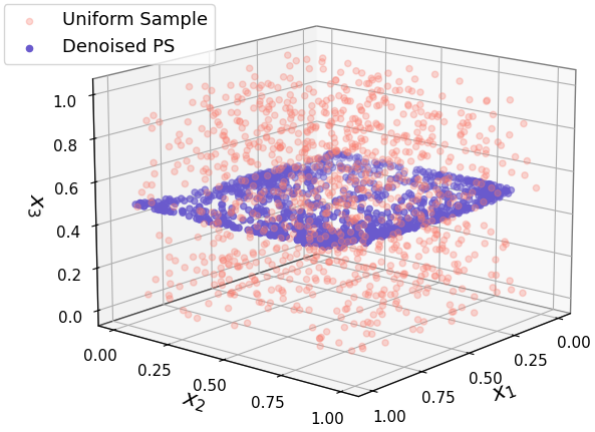

Figure 6.2: 3-D scatter plot of the uniform samples and the denoised output for the first 3 decision variables of DTLZ-5.

[2]The DAE is trained using the Adam optimizer, with a learning rate of 0.01 for 3000 iterations with early termination if the validation performance fails to improve over a consecutive 600 iterations. The training process is implemented in Python's TensorFlow library [140].

6.2 Application in Pareto Set Learning

Leveraging on the DAE's ability to discover a compressed representation of the PS, a potential application for the inverse modeling of the PS manifold is elucidated. The procedure for compact inverse models that synergize with DAE is outlined in Fig. 6.3.

Initially, isotropic Gaussian noise is introduced to the sparse PS data obtained from a run of any multi-objective optimization algorithm. Subsequently, the DAE is trained to restore the noisy PS data by minimizing Eq. (6.1). Upon completion of the training, the DAE encodes the PS data to facilitate the training of compact inverse models that map points from the PF to the reduced PS representation space. At the time of inference, the inverse models are applied to generate solutions in the low-dimensional space based on arbitrary points along the unexplored sub-regions of the PF. The DAE then serves to decode the generated solutions back to the original (high-dimensional) decision space.

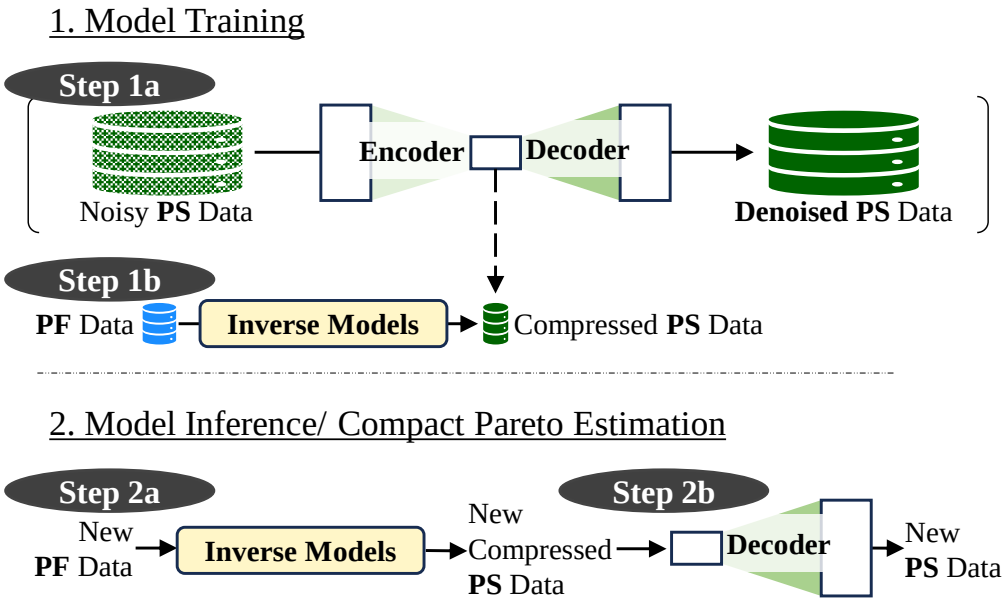

Figure 6.3: Compact inverse modeling workflow involving DAE.

The solutions decoded by the DAE are evaluated with the forward mapping function, i.e., the known objectives of the problem, to quantify the overall

quality of compact inverse models based on the *Inverted Generational Distance (IGD) Ratio* in Eq. (5.5).

In the case of DAE and independent *inv*GPs for post-hoc PS learning, each regression maps a point along the PF to a particular dimension in the DAE's bottleneck layer. This results in a reduction of the total number of inversions to $(m-1)$, marking a massive reduction in compute (especially in the case of high dimensional problems with $d >> m$), as illustrated in Table 6.1.

	Training	**Inference**
invGPs	$\mathcal{O}\big(d \cdot n^3\big)$	$\mathcal{O}\big(d \cdot n^3\big)$
Compact *inv*GPs	$\mathcal{O}\big(2C\big)+\mathcal{O}\big((m-1) \cdot n^3\big)$	$\mathcal{O}\big(C\big)+\mathcal{O}\big((m-1) \cdot n^3\big)$

Table 6.1: Computational complexity of *inv*GPs with and without dimensionality reduction. $\mathcal{O}\big(2C\big)$ and $\mathcal{O}\big(C\big)$ are the training and inference complexity of DAE, respectively, where C represents the total number of connections.

To further demonstrate its application, an empirical study on DTLZ test problems and a multi-criteria supply chain planning problem [7] is conducted in the subsequent sections. The DAE network architecture and training configurations remain consistent with those in Section 6.1.

6.2.1 DTLZ-2 and 5 Test Problems

Fig. 6.4 illustrates the approximated PF before and after compressed inverse modeling for DTLZ-2 and 5. The outcomes reveal that the compact inverse models effectively generates a more densely populated set of points along the PF, with IGD Ratios of 2.515 and 3.205 for DTLZ-2 and DTLZ-5, respectively.

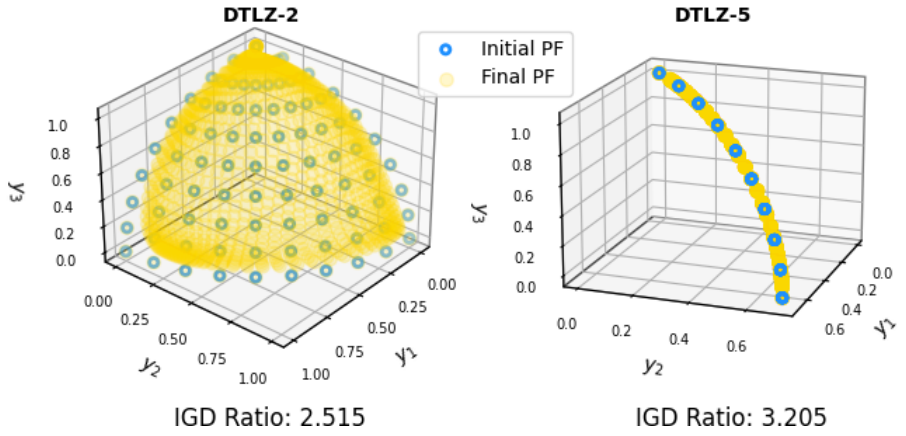

Figure 6.4: 3-D scatter plot of points in the approximated PF before (100 blue points) and after compact inverse modeling (1100 yellow points) and their respective IGD ratios, for DTLZ 2 and 5.

6.2.2 Multi-criteria Supply Chain Planning Problem

The same multi-criteria supply chain planning problem based on problem instance 1 in Chapter 4 is considered. For this problem, the DAE is initially trained on a PS data comprising 50 points. Similar to the methodology outlined in Section 6.1, the trained DAE is employed to denoise 1000 uniformly sampled solutions. Fig. 6.5 shows the denoised output for the last 3 decision variables, which forms a 1-D piecewise continuous curve—in agreement with the theoretical conclusions drawn from the KKT conditions for an $m = 2$ dimensional objective space. When applying the DAE in conjunction with the inverse model for post-hoc PS learning, a denser PF with a IGD Ratio of 2.09 is observed; see Fig. 6.6.

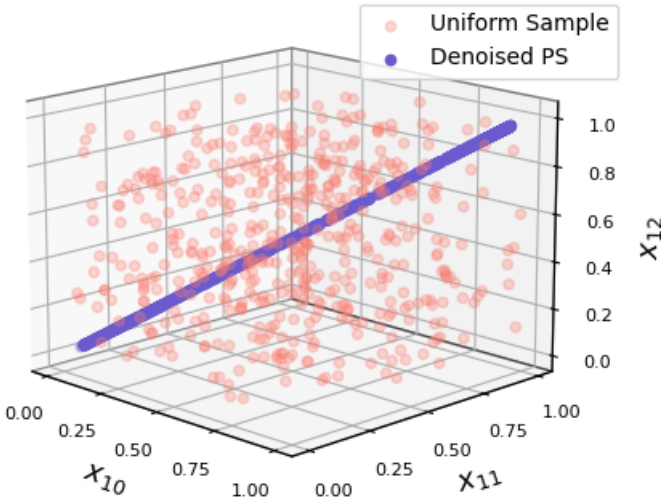

Figure 6.5: 3-D scatter plot of the uniform samples (in red) and the denoised output (in purple) for the last 3 decision variables of the supply chain planning problem.

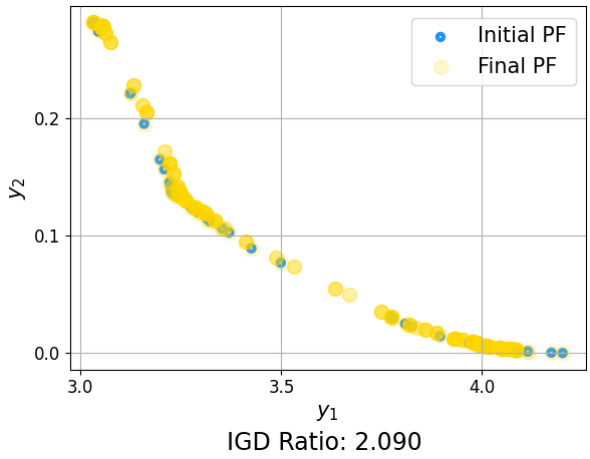

Figure 6.6: 3-D scatter plot of the points in the PF before (50 blue points) and after compact inverse modeling (1050 yellow points) and the achieved IGD ratio for the supply chain planning problem. The x and y-axis represents the average order fulfillment time (y_1) and unutilized production capacity (y_2) respectively.

6.3 Chapter Summary

This chapter presents a technique for discovering highly compressed PS representations achieving a 99.6% dimensionality reduction for test problems by means of the bottleneck layer of a DAE. The reduced PS is shown to sup-

port the development of compact inverse models via test problems and a real-world supply chain planning example yielding promising results.

Chapter 7

Monte Carlo Walk-pull for Inverse Modeling of Highly Non-linear Pareto Sets

In this chapter, the worst-case scenario for post-hoc PS learning is addressed, characterized by limited target data and the absence of a source dataset. To overcome this challenge, the novel concept of *Monte Carlo walk-pull* for Pareto data augmentation is introduced. For the first time, techniques that couple the walk-pull process with an inverse machine learner trained to map points along the front to the PS in decision space are presented. A distinguishing feature of the proposed method is its ability to accurately model the optima of problems with highly non-linear PS shapes.

7.1 Basic Idea

To overcome the small data challenge in PS learning, the basic idea of generative modeling of the PS manifold represented by current non-dominated solutions X is presented. The low-dimensional manifold (possibly $m-1$, if KKT condition holds) can be modeled explicitly by various dimensionality reduction methods. Through the compressed latent space, new solutions are generated and mapped back to the original space for augmenting the initial PS data.

This concept has been primarily explored to accelerate the convergence of solutions during the optimization process. Within each optimization step, principal component analysis [142] or autoencoder (AE) [143] has been studied to discover the low-dimensional representation for genetic operators in evolutionary algorithms to efficiently sample and generate new solution candidates. These new solutions are then mapped back to the original decision space for evaluating its Pareto optimality and the whole procedure is repeated. Alternatively, AE has also been used to iteratively learn and generate new solution candidates in the compressed space, replacing the need for genetic operators [144].

Extending upon these prior works, the generative modeling of the low-dimensional PS manifold is applied, focusing on the converged non-dominated solutions acquired in the course of *a posteriori* optimization. Two experiments based on test problems with $d = 10$ decision variables, each having a different PS shapes are conducted. The first test problem, DTLZ-2 [134] has $m = 3$ objectives and a linear PS shape. The second test problem, $\tilde{\boldsymbol{f}}$-2 has $m = 2$ objectives and a more complicated, non-linear PS shape with its detailed formulation listed in Table 7.2.

Following the generative approach in [144], an AE with a single hidden layer in both the encoder and decoder is utilized. The hidden layer consists of 10 neurons, while the bottleneck layer has $m - 1$ neurons. All the neurons in the AE have sigmoid activation functions. The AE is trained using the Adam optimizer [137] with a learning rate of 0.01. Training is conducted over 3000 iterations, with early stopping triggered if validation performance does not improve for 600 consecutive iterations. The implementation is done in Python using the Keras library [145].

Fig. 7.1 depicts the initial 100 data points within the true PS data used for training and the additional 1000 data points generated by the AE for DTLZ-2 for its first three decision variables. The AE proficiently models the linear shape of the PS with the generated points falling closely within the

2-D plane of the PS manifold.

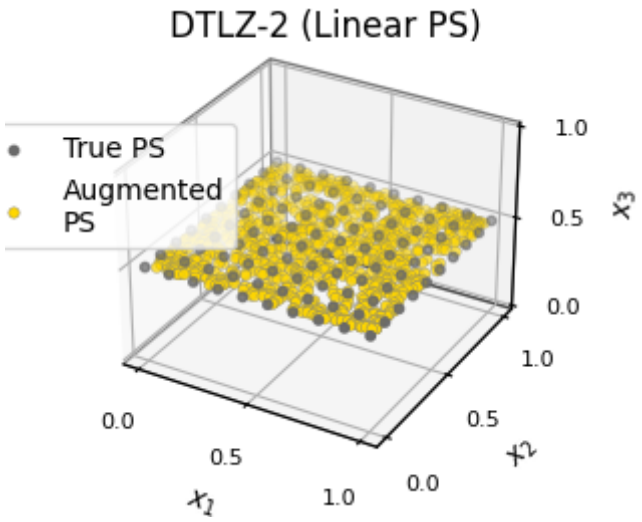

Figure 7.1: 3-D scatter plots of the true (in grey) and augmented (in yellow) PS for first three decision variables of DTLZ-2 with a linear PS shape.

7.1.1 Limitation in non-linear Pareto Set

For the second test problem $\tilde{\boldsymbol{f}}$-2, the AE struggles to effectively model the non-linear shape of the PS, despite its ability to capture complex patterns in data distributions. As shown in Fig. 7.2, there is a significant discrepancy in both magnitude and offset angle between the true 1-D sinusoidal shape of the PS and the points generated by the AE.

This deviation is likely caused by the restrictive bottleneck layer of the AE, which enforces a low-dimensional representation, thereby increasing the learning difficulty, especially with a limited training dataset. Additionally, for PS data augmentation, extracting a low-dimensional representation is not strictly required.

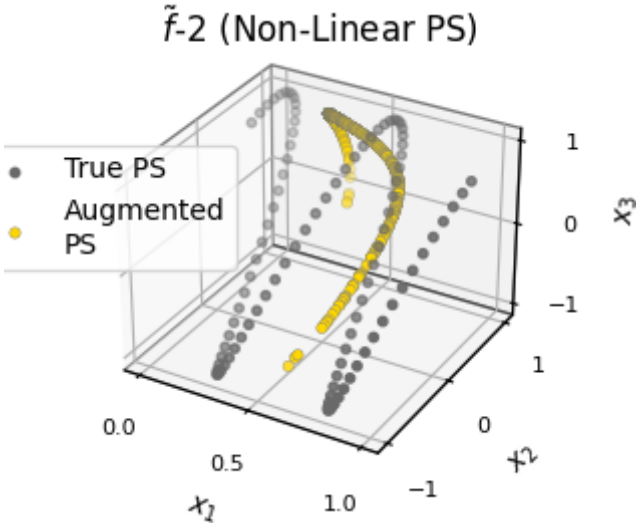

Figure 7.2: 3-D scatter plots of the true (in grey) and augmented (in yellow) PS for first three decision variables of $\tilde{f}$-2 with a non-linear PS shape.

7.2 Monte Carlo Walk-pull for Non-linear Pareto Set

In this section, a novel approach on generative modeling of the complex PS shapes and methods to integrate it with the *inv*GP is put forward, thereby addressing the issue of non-linear PS learning with limited data.

7.2.1 The Monte Carlo Walk-pull Process

The generative modeling approach discussed in Section 7.1 requires extracting the low-dimensional PS manifold for data augmentation. However, for MOPs with non-linear PS shapes or where the KKT conditions do not hold, precisely identifying and uncovering the low-dimensional representation poses an additional challenge.

Hence, to effectively model intricate PS shapes from various MOPs, the concept of *Monte Carlo walk-pull* process [146, 147] to model and sample additional points along the piecewise continuous manifold $\mathcal{M}_X$ of the acquired PS

data X is first introduced. Specifically, each d-dimensional non-dominated solution $\boldsymbol{x}_k \in X$, is perturbed by an isotropic Gaussian noise $\boldsymbol{z} \in \mathbb{R}^d$ with zero mean and covariance sI. This random walk process is repeated n_p number of times by adding a different $\boldsymbol{z} \sim \mathcal{N}(0, sI)$, to $\forall \boldsymbol{x}_k \in X$, for increasing the number of available training examples.

While during the pull process, efforts are made to reconstruct the perturbed data points back onto $\mathcal{M}_X$ using a machine learning model. The model is trained to minimize the overall reconstruction error between the perturbed and original locations of the non-dominated solutions and is formulated as,

$$loss(\boldsymbol{\phi}) = \sum_{l=1}^{n \cdot n_p} \sum_{h=1}^{n} \left|\left| \boldsymbol{\chi}_{\boldsymbol{\phi}}(\boldsymbol{x}_h + \boldsymbol{z}_l) - \boldsymbol{x}_h \right|\right|_2^2, \tag{7.1}$$

where $\boldsymbol{\chi}_{\boldsymbol{\phi}}$ is the function learned by the machine learning model to pull the perturbed points back to its initial location by optimizing its parameters $\boldsymbol{\phi}$.

The main difference over existing dimensionality reduction methods, is that the unnecessary constricting requirement to explicitly learn the low-dimensional PS manifold is removed when optimizing Eq. (7.1). Additionally, under this training setup, an optimal projection that maps perturbed data points back onto the local region of $\mathcal{M}_X$ can be potentially learned – a pivotal feature, that will be rigorously demonstrated next through theoretical and empirical analysis.

Theoretical Analysis

The optimal projection $\boldsymbol{\chi}^*$ that a machine learning model can possibly learn is derived analytically from the reconstruction error function in Eq. (7.1). Let p_{lh} be the probability that $\boldsymbol{x}_h$ is the original location of the perturbed point $\tilde{\boldsymbol{x}}_l$. Furthermore, based on the Gaussian noise introduced during the walk process, p_{lh} can be computed from the multivariate Gaussian probability function, $exp\big(-0.5(\tilde{\boldsymbol{x}}_l - \boldsymbol{x}_h)^T (sI)^{-1} (\tilde{\boldsymbol{x}}_l - \boldsymbol{x}_h)\big) \big/ \big((2\pi)^{0.5d} |Il|^{0.5}\big)$. The reconstruc-

tion error in Eq. (7.1) can thus be reformulated as,

$$loss = \sum_{l=1}^{n \cdot n_p} \sum_{h=1}^{n} p_{lh} [\boldsymbol{\chi}(\tilde{\boldsymbol{x}}_l) - \boldsymbol{x}_h]^2, \tag{7.2}$$

where $\tilde{\boldsymbol{x}}_l = \boldsymbol{x}_h + \boldsymbol{z}_l$ represents the perturbed data point. The optimal $\boldsymbol{\chi}^*$ can then be computed by setting its partial derivative ${}^{\partial loss}/_{\partial \boldsymbol{\chi}(\tilde{\boldsymbol{x}}_l)} = 0$ and re-arranging it as,

$$\begin{aligned}
2 \sum_{l=1}^{n \cdot n_p} \sum_{h=1}^{n} \Big(p_{lh} \boldsymbol{\chi}^*(\tilde{\boldsymbol{x}}_l) - p_{lh} \boldsymbol{x}_h \Big) &= 0 \\
\sum_{l=1}^{n \cdot n_p} \Big(\boldsymbol{\chi}^*(\tilde{\boldsymbol{x}}_l) \sum_{h=1}^{n} p_{lh} \Big) &= \sum_{l=1}^{n \cdot n_p} \sum_{h=1}^{n} p_{lh} \boldsymbol{x}_h \\
\sum_{l=1}^{n \cdot n_p} \boldsymbol{\chi}^*(\tilde{\boldsymbol{x}}_l) &= \sum_{l=1}^{n \cdot n_p} \Big(\frac{\sum_{h=1}^{n} p_{lh} \boldsymbol{x}_h}{\sum_{h=1}^{n} p_{lh}} \Big) \\
\boldsymbol{\chi}^*(\tilde{\boldsymbol{x}}_l) = \frac{\sum_{k=1}^{n} p_{lh} \boldsymbol{x}_h}{\sum_{h=1}^{n} p_{lh}}, \quad \forall l &= 1, 2, .., n \cdot n_p.
\end{aligned} \tag{7.3}$$

From Eq. (7.3), the optimal projection $\boldsymbol{\chi}^*$ maps each perturbed data point $\tilde{\boldsymbol{x}}_l$ onto a locally euclidean space within $\mathcal{M}_X$. The projected points under $\boldsymbol{\chi}^*(\tilde{\boldsymbol{x}}_l)$, is hence weighted proportionally based on the likelihood of $\forall \boldsymbol{x}_h \in X$ being the actual location of the perturbed solution $\tilde{\boldsymbol{x}}_l$.

Experimental Analysis

Next, the viability of learning the optimal projection $\boldsymbol{\chi}^*_{\boldsymbol{\phi}}$, derived in Eq. 7.3 across three MOP test problems with distinctive PS shape is assessed. The first test problem $\tilde{\boldsymbol{f}}$-2, has a sinusoidal PS shape and is from the same problem discussed in Section 7.1. The second test problem $\tilde{\boldsymbol{f}}$-4, has a slightly more complex sinusoidal PS shape while the final test problem $\tilde{\boldsymbol{f}}$-7, has a different PS shape defined by an exponential function. All three test problems have $m = 2$ objectives and $d = 10$ decision variables with its formulations listed in Table 7.2.

A feedforward neural network, whose structure is detailed in Table 7.1, is employed to learn the optimal projection function within the walk-pull process. The primary motivation behind this choice, is its differentiable model structure that is deemed valuable when integrating it with the *inv*GP. The details of this integration will be discussed in the subsequent section.

Table 7.1: Neural network structure used for MOP test problems with different non-linear PS functions.

PS	**Neural Network Structure**			
Function	**# Hidden**	**# Neuron per**	**Act. Func. for**	**Act. Func. for**
Type	**Layer**	**Hidden Layer**	**Hidden Layer**	**Output Layer**
Sinusoidal	5	200	Sigmoid	Linear
Exponential	4	200	Sigmoid	Sigmoid

For each test instance, the availability of $n = 100$ data points uniformly distributed along the true PS is assumed. These points are normalized to a value between 0 to 1 before the introduction of the Gaussian noise $z \sim \mathcal{N}(0, sI)$ with $s = 0.1$. Each point within the true PS is perturbed $n_p = 20$ times to obtain a denser approximation of the PS with an additional 2000 points by the end of the reconstruction process. The dataset is then randomly split into training and validation sets with a ratio of 0.8 to 0.2. The remaining training settings are configured according to the training setup of the AE discussed in Section 7.1.

Fig. 7.3 illustrates the locations of the first three decision variables across the test problems during the Monte Carlo walk (left plot) and pull (right plot) processes. For test problem $\tilde{\boldsymbol{f}}$-2, unlike the AE, the neural network without the bottleneck layer is able to reconstruct the perturbed PS data successfully during the pull process, leading to a denser PS approximation. Similarly, for $\tilde{\boldsymbol{f}}$-4 and $\tilde{\boldsymbol{f}}$-7, the neural network proficiently generates a denser approximation of the PS by projecting perturbed points back onto unexplored sub-regions of the PS manifold.

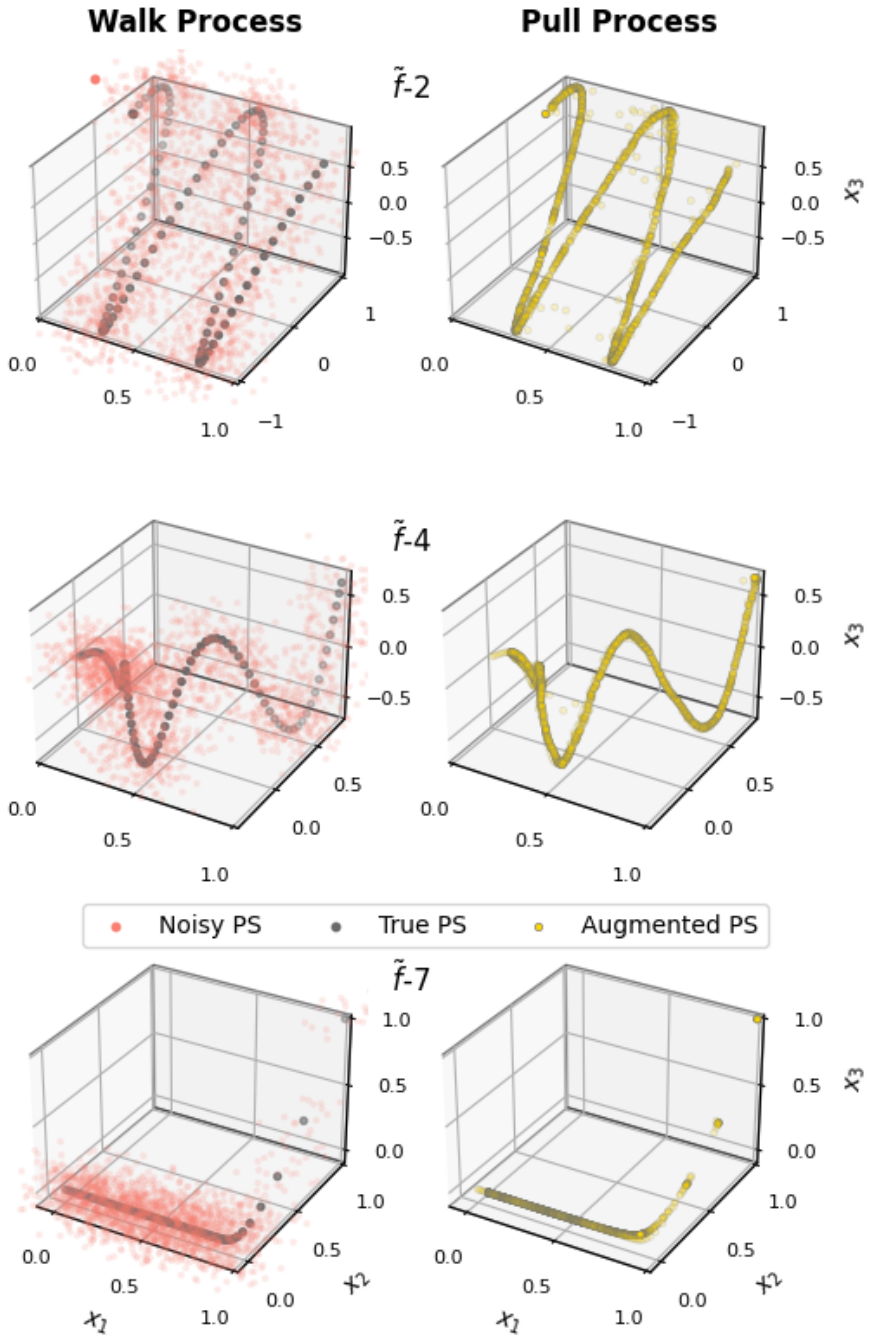

Figure 7.3: 3-D scatter plot of the first three decision variables, for benchmark problem $\tilde{f}$-2, $\tilde{f}$-4 and $\tilde{f}$-7. The orientation of the In the walk process (left), the grey and red points correspond to the true and noisy PS data, respectively. In the pull process (right), the yellow points represent the additional points generated at the end of the training process.

7.2.2 Synergizing Monte Carlo Walk-pull and Inverse Modeling

The Monte Carlo walk-pull process presents a means for data augmentation within the non-linear PS manifold, independent of the articulated preferences. Conversely, the inverse model facilitates the mapping of points from the projected PF to PS, with its accuracy compromised by the small training data. To enable controlled generation of solutions with improved accuracy and reliability, techniques to synergize the walk-pull process within the inverse modeling framework are proposed next, as illustrated in Fig. 7.4.

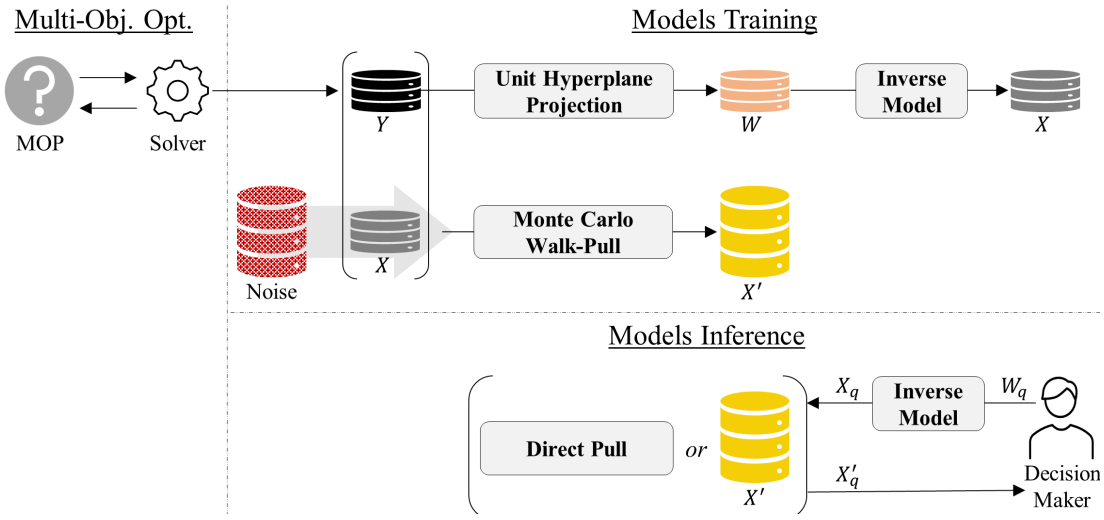

Figure 7.4: Complete workflow for enhancing non-linear PS learning under limited data. It involves the integration of the Monte Carlo walk-pull process to the inverse modeling framework.

Beginning with the dataset $D = \{Y, X\}$, the approximated PF Y, is first transformed to W, distributed along a unit hyperplane for ease of preference articulation. Subsequently, *inv*GPs are trained to learn the mapping from $\mathcal{W}$ to $\mathcal{X}$. Simultaneously, the Monte Carlo walk-pull process is invoked to model the PS manifold $\mathcal{M}_X$, formed by X. As a direct outcome of this process, a neural network is trained to reconstruct perturbed PS data back towards $\mathcal{M}_X$, creating a denser PS approximation $X^{'}$, consisting of initial PS data X and the reconstructed points.

At the time of inference, the *inv*GPs initially map any arbitrary preferences W_q articulated by the DM to its corresponding decision variable values X_q. Within the common decision space, two projection methods are introduced to enhance the accuracy and reliability X_q: the direct pull method, which leverages the neural network trained during the walk-pull process and the nearest point projection method by referencing the augmented PS data. The following sections describe these two methods in detail and highlight their complementary benefits.

Direct Pull Method

The initial prediction inferred by *inv*GP, albeit slightly inaccurate, can be interpreted as PS data with added noise (or error), analogous to a walk process. Consequently, the neural network $\boldsymbol{\chi}_{\boldsymbol{\phi}}$, trained in the pull process, can be directly applied to project this perturbed point back onto the local piecewise continuous manifold $\mathcal{M}_X$. This projection incurs a minor additional computational complexity of $\mathcal{O}(C)$, where C denotes the number of connections in the neural network, and is performed as,

$$\boldsymbol{\chi}_{\boldsymbol{\phi}}\big(\boldsymbol{\mu}(\boldsymbol{w}_q)\big) = \boldsymbol{x}_q^{'}, \tag{7.4}$$

where $\boldsymbol{\mu}(\boldsymbol{w}_q) = [\mu_1(\boldsymbol{w}_q), \mu_2(\boldsymbol{w}_q), ..., \mu_d(\boldsymbol{w}_q)]^\intercal$ is the mean prediction vector computed by each independent *inv*GP using Eq. (4.3a). While, the corresponding uncertainty estimates are computed by leveraging the local gradients derived from the differentiable structure of the neural network and the variance values inferred by the *inv*GP in Eq. (4.3b).

The direct pull method provides an analytical approach for efficiently computing the predictive distribution from the solutions inferred by the *inv*GP. Additionally, it provides a smooth approximation along the PS manifold given solutions with different decision variable values. Following the optimal projection delineated in Eq. 7.3, it is weighted proportionally based on the

distance between the solutions to be projected and the obtained PS data X. This helps to preserve or enhance the diversity among the solutions slated for projection, a particularly valuable feature when projecting solutions with closely similar decision variable values. However, akin to many machine learning models, when confronted with out-of-distribution data (those significantly diverging from the distribution of the perturbed PS data during the walk process), its performance might be adversely impacted. This could potentially happen when the solutions predicted by *inv*GP are distant from the true PS.

Nearest Point Projection Method

As an alternative, the augmented PS data $X^{'}$, can be effectively leveraged to enhance the reliability of the solutions predicted by *inv*GP. This is accomplished by using $X^{'}$ as a set of reference points, projecting solutions onto the nearest point $\boldsymbol{x}_l \in X^{'}$ based on the Euclidean distance. The additional computational complexity of this step is $\mathcal{O}\big(|X^{'}| \cdot d\big)$, where $|X^{'}|$ denotes the number of reference points and d represents the dimensionality of the decision variables, and is computed as,

$$\boldsymbol{\chi}_{X^{'}}(\boldsymbol{x}_q) = \underset{\boldsymbol{x}_l \in X^{'}}{\operatorname{argmin}} \, ||\boldsymbol{x}_q - \boldsymbol{x}_l||_2. \tag{7.5}$$

The predictive distribution in this case is empirically approximated by performing multiple runs of the Monte Carlo simulation based on Eq. (7.5), by sampling $\boldsymbol{x}_q$ from $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{w}_q), \Sigma_{\boldsymbol{w}_q})$. Here, $\Sigma_{\boldsymbol{w}_q}$ is the diagonal covariance matrix $\text{diag}(\sigma_1^2(\boldsymbol{w}_q), ..., \sigma_d^2(\boldsymbol{w}_q))$, computed from Eq. (4.3b).

The predictive distribution computed by the nearest point projection (NPP) method, in this case, relies on a subset of points within the local region of the augmented PS with its objective values bounded by this region. This ensures reliable estimates and complements the direct pull method in extreme cases where the *inv*GP suffers from poor accuracy. However, compared to the direct

pull method, the NPP has slightly higher computational complexity when calculating the uncertainty estimates. Additionally, the projection granularity depends on the number and diversity of points in the augmented PS. When projecting closely clustered solutions, multiple solutions might map onto the same point within the augmented PS, resulting in potential loss of diversity, despite having improvements made to the overall predictive accuracy.

7.2.3 A Summary of Salient Features

In this section, the key aspects of the proposed approach, emphasizing the noteworthy features arising from the collaborative synergy between the *inv*GP and Monte Carlo walk-pull process is further elaborated. These features play a crucial role to enable non-linear PS learning in scenarios characterized by small data.

- *Generative modeling of non-linear PS.* The Monte Carlo walk-pull is able to model the non-linear PS manifold through training a neural network to reconstruct additional perturbed PS data, thereby generating a denser approximation of the PS representation.

- *Data-efficient PS learning.* The performance of *inv*GP trained on limited data is enhanced by coupling it with the walk-pull process. Solutions inferred by *inv*GP are projected onto a localized region within the manifold of the acquired PS data, either through the trained neural network or by referencing the augmented PS data.

- *Seamless end-to-end uncertainty estimates.* The projection methods derived from the walk-pull process retain the predictive distribution of the decision variable values inferred by the *inv*GP. These values are computed either analytically through the differentiable neural network model or empirically from a subset of the augmented PS.

7.3 Empirical Analysis

The proposed methodology is tested on nine MOP benchmarks with $m = 2$ to 3 objectives, $d = 10$ decision variables and a diverse PS shapes characterized by either a complex exponential or sinusoidal functions. The independent *inv*GPs for inverse modeling employ a squared exponential covariance function and are implemented using the GPyTorch library [126]. The training process involves 100 training iterations with a learning rate of 0.1. While the neural network involved in the walk-pull process is implemented following the training configurations in Section 7.2.1 with its structure configured according to Table 7.1. Similarly, a total of $n = 100$ training data is assumed to be available, while a test set of $n_q = 1000$ query points (e.g., those supplied by a DM) not contained in the training data are used for performance evaluation. Both the training and test data are obtained from uniformly distributed points along the true PF.

The experiments are conducted over 50 independent runs with its performance evaluated based on the *Root Mean Square Error (RMSE)* and *Inverted Generational Distance (IGD) Ratio*. The two metrics capture distinctive performance attributes of the generated solutions from the perspective of a DM with postponed preferences. The RMSE metric is measured in both the decision and objective spaces to capture the fine-grained evaluation of the quality of post-hoc PS learning. The RMSE values assessed in the decision space offer direct insights into the accuracy of post-hoc PS learning. While, evaluation in the objective space delves into the extent to which the objective values of the generated solutions align with the DM's postponed preferences, representing the primary goal of this study. In contrast to the RMSE metric, the IGD Ratio [103] offers a broader perspective on the overall PF approximation capacity by quantifying the improvement in the PF approximation before and after post-hoc PS learning. The details of the benchmark MOPs and instantiations of the evaluation metrics are further discussed in the subsequent sections.

7.3.1 Test Problems with non-linear Pareto Set

The nine MOP benchmark problems used for the experiments are adopted from [148] with its objective functions taking the general form of,

$$\begin{aligned} f_1 &= \alpha_1(\boldsymbol{x}_I) + \beta_1(\boldsymbol{x}_{II} - \boldsymbol{g}(\boldsymbol{x}_I)), \\ &\vdots \\ f_m &= \alpha_m(\boldsymbol{x}_I) + \beta_m(\boldsymbol{x}_{II} - \boldsymbol{g}(\boldsymbol{x}_I)), \end{aligned} \tag{7.6}$$

$$s.t. \quad \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}, \ \forall x \in \{\boldsymbol{x}_I, \boldsymbol{x}_{II}\},$$

where the decision variables constitute $\boldsymbol{x}_I = [x_1, ..., x_{m-1}]$ and $\boldsymbol{x}_{II} = [x_m, ..., x_d]$, with $\alpha_{i=1,2,..,m}$, defining the shape of PF based on $\boldsymbol{a}_I \leq \boldsymbol{x}_I \leq \boldsymbol{b}_I$, while $\beta_{i=1,2,..,m}$, is a non-negative function defined based on the difference between the current values of $\boldsymbol{x}_{II}$, and its Pareto optimal values calculated from $\boldsymbol{g}(.)$, given the values of $\boldsymbol{x}_I$. Note, $[f_1, f_2, ..., f_m]^\intercal = [\alpha_1(\boldsymbol{x}_I), \alpha_2(\boldsymbol{x}_I), ..., \alpha_m(\boldsymbol{x}_I)]^\intercal$, i.e., a point along the PF, when $\boldsymbol{x}_{II}$ corresponds to the Pareto optimal solutions calculated from $g(x_I)$, with $\beta_i(\cdot) = 0$, for $i = 1, 2, ..., m$.

Following Eq. (7.6), two minor modifications are made to the original nine MOP problem formulations in [148] for the purpose of this study. The first adaptation involves refining the β function to increase its sensitivity of the objective values for solutions that are far from Pareto optimal. While the second modification targets the PS function $\boldsymbol{g}$, aiming to introduce sufficient complexity for post-hoc PS learning utilizing only inverse models. The detailed formulations of the nine modified benchmark problems is denoted as $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9, are as shown in Table 7.2.

Table 7.2: Objective functions for $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9, with its highly non-linear PS functions.

Instance	**Objective and PS functions**	Variable Bounds
$\tilde{\boldsymbol{f}}$-1	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(x_j - x_1^{50(1.0+\frac{3(j-2)}{d-2})})^2$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}(x_j - x_1^{50(1.0+\frac{3(j-2)}{d-2})})^2$, where $\boldsymbol{j}_1 = \{j \mid j$ is odd and $2 \le j \le d\}$ and $\boldsymbol{j}_2 = \{j \mid j$ is even and $2 \le j \le d\}$. Its PS is $x_j = x_1^{50(1.0+\frac{3(j-2)}{d-2}))}, \quad j = 2, ..., d.$	$[0,1]^d$
$\tilde{\boldsymbol{f}}$-2	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(x_j - \sin(4\pi x_i + \frac{j\pi}{d}))^2$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}(x_j - \sin(4\pi x_i + \frac{j\pi}{d}))^2$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1. Its PS is $x_j = \sin(4\pi x_i + \frac{j\pi}{d})), \quad j = 2, ..., d.$	$[0,1] \times [-1,1]^{d-1}$
$\tilde{\boldsymbol{f}}$-3	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(x_j - x_1\cos(4\pi x_i + \frac{j\pi}{d}))^2$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}(x_j - x_1\sin(4\pi x_i + \frac{j\pi}{d}))^2$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1. Its PS is $x_j = \begin{cases}\cos(4\pi x_i + \frac{j\pi}{d}) & j \in \boldsymbol{j}_1,\\ \sin(4\pi x_i + \frac{j\pi}{d}) & j \in \boldsymbol{j}_2.\end{cases}$	$[0,1] \times [-1,1]^{d-1}$
$\tilde{\boldsymbol{f}}$-4	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(x_j - 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{d}}{3}))^2$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}(x_j - x_1\sin(\frac{6\pi x_1 + \frac{j\pi}{d}}{3}))^2$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1. Its PS is $x_j = \begin{cases}0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{d}}{3}) & j \in \boldsymbol{j}_1,\\ 0.8x_1\sin(\frac{6\pi x_1 + \frac{j\pi}{d}}{3}) & j \in \boldsymbol{j}_2.\end{cases}$	$[0,1] \times [-1,1]^{d-1}$
$\tilde{\boldsymbol{f}}$-5	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}\left\{x_j - [0.3x_1^2\cos(6\pi x_1 + \frac{4j\pi}{d}) + 0.6x_1]\cos(2\pi x_1 + \frac{j\pi}{d})\right\}^2$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}\left\{x_j - [0.3x_1^2\cos(6\pi x_1 + \frac{4j\pi}{d}) + 0.6x_1]\sin(2\pi x_1 + \frac{j\pi}{d})\right\}^2$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1. Its PS is $x_j = \begin{cases}[0.3x_1^2\cos(6\pi x_1 + \frac{4j\pi}{d}) + 0.6x_1]\cos(2\pi x_1 + \frac{j\pi}{d}) & j \in \boldsymbol{j}_1,\\ [0.3x_1^2\cos(6\pi x_1 + \frac{4j\pi}{d}) + 0.6x_1]\sin(2\pi x_1 + \frac{j\pi}{d}) & j \in \boldsymbol{j}_2.\end{cases}$	$[0,1] \times [-1,1]^{d-1}$
$\tilde{\boldsymbol{f}}$-6	$\tilde{f}_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + 20\sum_{j\in \boldsymbol{j}_1}(x_j - 2x_2\sin(4\pi x_1 + \frac{j\pi}{d}))^2$, $\tilde{f}_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + 20\sum_{j\in \boldsymbol{j}_2}(x_j - 2x_2\sin(4\pi x_1 + \frac{j\pi}{d}))^2$, $\tilde{f}_3 = \sin(0.5x_1\pi) + 20\sum_{j\in \boldsymbol{j}_3}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{d}))^2$, where $\boldsymbol{j}_1 = \{j \mid 3 \le j \le d,$ and $j-1$ is a multiplication of $3\}$, $\boldsymbol{j}_2 = \{j \mid 3 \le j \le d,$ and $j-2$ is a multiplication of $3\}$, $\boldsymbol{j}_3 = \{j \mid 3 \le j \le d,$ and j is a multiplication of $3\}$. Its PS is $x_j = 2x_2\sin(4\pi x_1 + \frac{j\pi}{d}), \quad j = 3, ..., d.$	$[0,1]^2 \times [-2,2]^{d-2}$
$\tilde{\boldsymbol{f}}$-7	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(4v_j^2 - cos(8v_j\pi) + 1)$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20\sum_{j\in \boldsymbol{j}_2}(4v_j^2 - cos(8v_j\pi) + 1)$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1, and $v_j = x_j - x_1^{50(1+\frac{3(j-2)}{d-2})}, \quad j = 2, ..., d.$ Its PS is $x_j = x_1^{50(1+\frac{3(j-2)}{d-2})}, \quad j = 2, ..., d.$	$[0,1]^d$
$\tilde{\boldsymbol{f}}$-8	$\tilde{f}_1 = x_1 + 20(4\sum_{j\in \boldsymbol{j}_1} v_j^2 - 2\prod_{j\in \boldsymbol{j}_1} cos(\frac{20v_j\pi}{\sqrt{j}}) + 2)$, $\tilde{f}_2 = 1 - \sqrt{x_1} + 20(4\sum_{j\in \boldsymbol{j}_2} v_j^2 - 2\prod_{j\in \boldsymbol{j}_2} cos(\frac{20v_j\pi}{\sqrt{j}}) + 2)$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1, and $v_j = x_j - x_1^{50(1+\frac{3(j-2)}{d-2})}, \quad j = 2, ..., d.$ Its PS is $x_j = x_1^{50(1+\frac{3(j-2)}{d-2})}, \quad j = 2, ..., d.$	$[0,1]^d$
$\tilde{\boldsymbol{f}}$-9	$\tilde{f}_1 = x_1 + 20\sum_{j\in \boldsymbol{j}_1}(x_j - sin(5\pi x_1 + \frac{j\pi}{d}))^2$, $\tilde{f}_2 = 1 - x_1^2 + 20\sum_{j\in \boldsymbol{j}_2}(x_j - sin(5\pi x_1 + \frac{j\pi}{d}))^2$, where $\boldsymbol{j}_1$ and $\boldsymbol{j}_2$ are the same as those of $\tilde{\boldsymbol{f}}$-1, Its PS is $x_j = sin(5\pi x_1 + \frac{j\pi}{d}), \quad j = 2, ..., d.$	$[0,1]^2 \times [-1,1]^{d-1}$

7.3.2 Performance in Decision Space

The accuracy of PS learning given a set of n_q query/ test points in the objective space, is measured based on the RMSE between the actual and predicted decision variable values is computed based on Eq. 5.7b.

The $RMSE_x$ values for *inv*GP and the projection methods are compared in Table 7.3. The results underscore the overall effectiveness of both the direct pull and NPP methods in enhancing the predictive accuracy of *inv*GP across all test instances $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9. A more than 30% reduction in $RMSE_x$ values of the *inv*GP is observed for test problem $\tilde{\boldsymbol{f}}$-5 and $\tilde{\boldsymbol{f}}$-9. In comparison, the direct pull outperforms NPP method in majority of the test problems except for $\tilde{\boldsymbol{f}}$-5 and $\tilde{\boldsymbol{f}}$-7.

Table 7.3: Accuracy of PS learning measured in $RMSE_{\boldsymbol{x}}$ for *inv*GP, direct pull and NPP methods for $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9. Values in bold indicate the best averaged performance for a given MOP over 50 independent runs. Values in brackets denote the standard deviations in performance over these runs.

Instance	***inv*GP**	**+Direct Pull**	**+NPP**
		$RMSE_x$	
$\tilde{\boldsymbol{f}}$-1	0.048 (0.002)	**0.039** (0.004)	**0.039** (0.003)
$\tilde{\boldsymbol{f}}$-2	0.445 (0.051)	**0.363**(0.089)	0.374 (0.093)
$\tilde{\boldsymbol{f}}$-3	0.449 (0.051)	**0.376** (0.085)	0.395 (0.093)
$\tilde{\boldsymbol{f}}$-4	0.180 (0.008)	**0.147** (0.013)	0.167 (0.014)
$\tilde{\boldsymbol{f}}$-5	0.020 (0.004)	0.017 (0.002)	**0.013** (0.001)
$\tilde{\boldsymbol{f}}$-6	0.721 (0.000)	**0.716** (0.005)	0.736 (0.005)
$\tilde{\boldsymbol{f}}$-7	0.048 (0.002)	0.041 (0.003)	**0.040** (0.003)
$\tilde{\boldsymbol{f}}$-8	0.048 (0.002)	**0.038** (0.002)	0.039 (0.002)
$\tilde{\boldsymbol{f}}$-9	0.458 (0.084)	**0.312** (0.113)	0.324 (0.122)

To investigate further into the unique characteristics of the projection methods, the true PS and its corresponding predictions for the first three decision variables of $\tilde{\boldsymbol{f}}$-9 are plotted in Fig. 7.5. The *inv*GP struggles to provide an accurate prediction with large deviation both in the offset angle and magnitude between the sinusoidal shape of the true and predicted PS. Through the integration of the two projection methods, the predictive accuracy of the *inv*GP has been greatly improved with the projected points closely approxi-

mate the true PS shape.

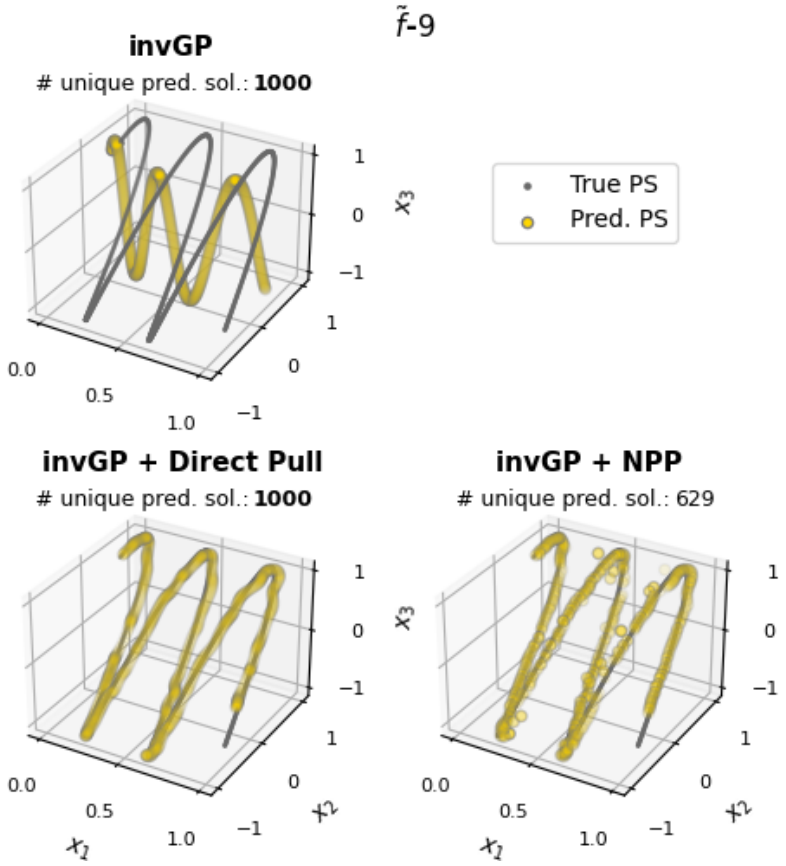

Figure 7.5: 3-D scatter plots of the true (in grey) and predicted (in yellow) PS by *inv*GP (top, left), direct pull (bottom, left) and NPP (bottom, right) methods for first three decision variables of $\tilde{\boldsymbol{f}}$-9, under a particular run.

In contrast, the direct pull method offers a smoother approximation of the true PS compared to the NPP method, which yields discontinuous and clustered points in various sub-regions of the true PS. This discrepancy stems from the NPP method's reliance on the augmented PS when selecting the closest point for projection. Such dependence can pose a slight issue for the case of $\tilde{\boldsymbol{f}}_9$, where the predictions made by *inv*GP are clustered close together. As a result, the NPP method projects multiple solutions onto the same point within the augmented PS, leading to only 629 unique solutions out of the 1000 test points.

7.3.3 Performance in Objective Space

To assess the primary goal of generating solutions that satisfy the articulated preferences, the RMSE evaluated in the objective space is calculated based on Eq. 5.7a.

The $RMSE_f$ values in Table 7.4 highlight the efficacy of the projection methods in mitigating the $RMSE_f$ values of *inv*GP. Overall, the NPP method performs better across a wider range of test problems, spanning from $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-3, $\tilde{\boldsymbol{f}}$-6 and $\tilde{\boldsymbol{f}}$-8, whereas the direct pull method, excels primarily in $\tilde{\boldsymbol{f}}$-4, $\tilde{\boldsymbol{f}}$-7 and $\tilde{\boldsymbol{f}}$-9.

Table 7.4: Quality of PS learning measured in $RMSE_{\boldsymbol{f}}$ for *inv*GP, direct pull and NPP methods for $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9. Values in bold and bracket indicate the best averaged and standard deviations of the performance for each MOP over 50 independent runs.

Instance	***inv*GP**	**+Direct Pull**	**+NPP**
	$RMSE_f$		
$\tilde{\boldsymbol{f}}$-1	0.575 (0.056)	0.551 (0.262)	**0.508** (0.195)
$\tilde{\boldsymbol{f}}$-2	24.194 (4.523)	19.260 (7.760)	**13.352** (8.638)
$\tilde{\boldsymbol{f}}$-3	24.567 (4.460)	18.381 (9.127)	**17.969** (5.833)
$\tilde{\boldsymbol{f}}$-4	6.498 (0.475)	**2.652** (1.079)	3.099 (1.598)
$\tilde{\boldsymbol{f}}$-5	**0.071**(0.024)	0.138 (0.070)	0.138 (0.061)
$\tilde{\boldsymbol{f}}$-6	47.625 (0.005)	11.517 (4.273)	**2.631** (0.604)
$\tilde{\boldsymbol{f}}$-7	32.196 (1.931)	**18.888** (5.597)	19.287 (4.897)
$\tilde{\boldsymbol{f}}$-8	3.514 (0.222)	2.096 (0.866)	**2.043** (0.746)
$\tilde{\boldsymbol{f}}$-9	25.380 (5.569)	**17.919** (9.427)	39.154 (20.240)

Notably, the $RMSE_{\boldsymbol{f}}$ values of the *inv*GP have been significantly reduced by up to 76% and 94% using the direct pull and NPP methods, respectively, for test problem $\tilde{\boldsymbol{f}}$-6 with $m = 3$ objectives. Fig. 7.6 further illustrates the distribution of points for the true and approximated PF before and after projections. Initially, the objective values of the *inv*GP's predicted solutions diverge considerably from the true PF, as shown in the top row of Fig. 7.6. However, leveraging on the projection methods, it substantially improve the convergence towards the true PF.

Both the direct pull and NPP methods exhibit distinct projection trajectories, each offering complementary advantages. In the middle row of Fig. 7.6, the direct pull method increases the diversity of solutions, covering a wider area of the true PF compared to NPP. This helps to achieve a more consistent alignment between the objective values of the predicted solutions and the preferences articulated by the DM across different sub-regions of the true PF. Conversely, the NPP method demonstrates a better convergence towards the true PF, as depicted in the bottom row of Fig 7.6. This is attributed to its inherent mechanism that ensures the projected solutions remain within the objective value bounds of the augmented PS. Consequently, it provides reliable estimates for scenarios like $\tilde{\boldsymbol{f}}$-6, where the inverse model performs poorly in the objective space.

Another interesting finding emerges from comparing the RMSE values of the projection methods evaluated in the objective and decision spaces. An inconsistency in performance gains is observed for test problems $\tilde{\boldsymbol{f}}$-2, $\tilde{\boldsymbol{f}}$-3, $\tilde{\boldsymbol{f}}$-5 and $\tilde{\boldsymbol{f}}$-6. Fig. 7.7 compares the average and standard deviation of the RMSE values of the projection methods for these problems based on the 50 independent runs. For $\tilde{\boldsymbol{f}}$-2, $\tilde{\boldsymbol{f}}$-3 and $\tilde{\boldsymbol{f}}$-6, the direct pull method showcases a lower RMSE values in the decision space. However, when assessing RMSE values in the objective space, the NPP method performs better. Similarly, for $\tilde{\boldsymbol{f}}$-5, the NPP method yields a lower $RMSE_{\boldsymbol{x}}$ value but have comparable $RMSE_{\boldsymbol{f}}$ value with the direct pull method. These observations highlight the non-monotonic relationship between the performances in the decision space used for guiding the PS learning and the objective space which dictates the solution optimality.

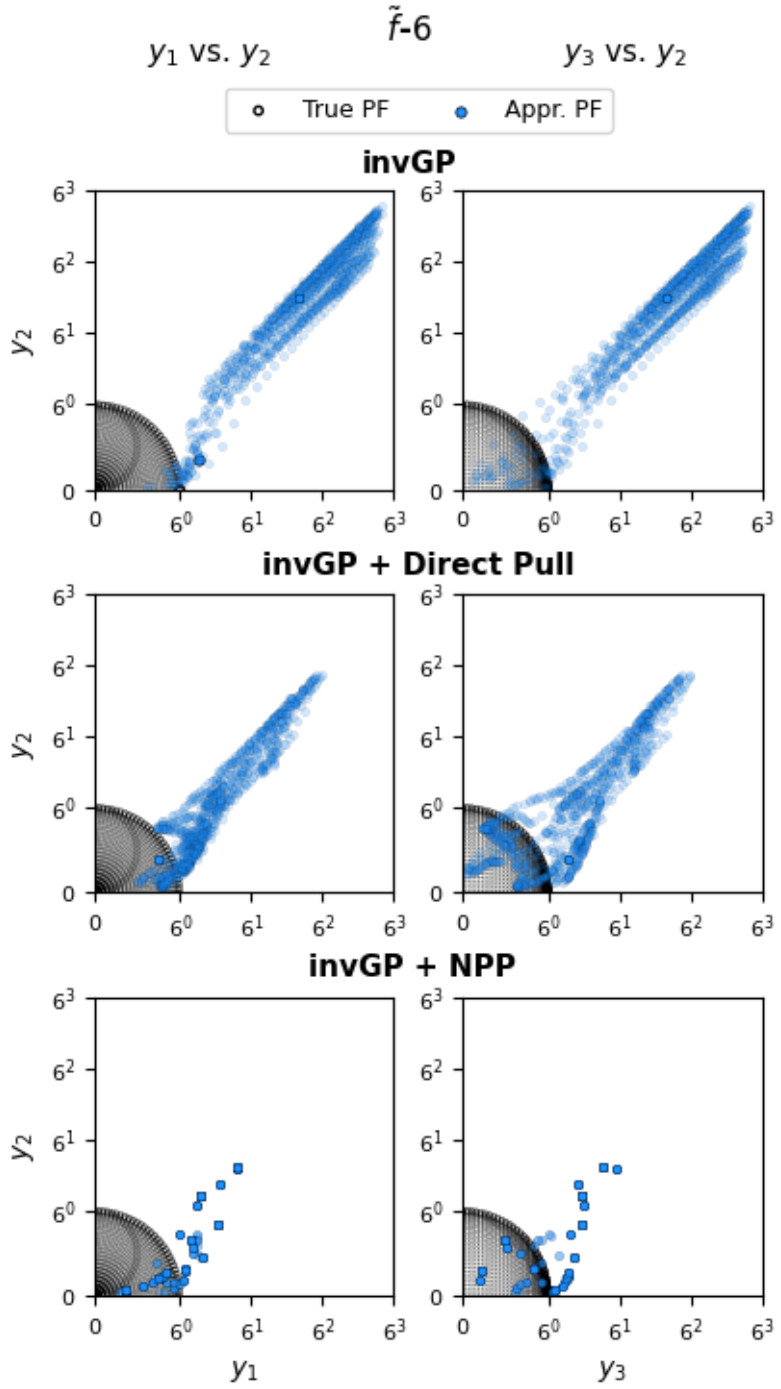

Figure 7.6: 2-D scatter plots of the true (in black) and approximated (in blue) PF by the *inv*GP (top row), direct pull (middle row) and NPP methods (bottom row) for $\tilde{\boldsymbol{f}}$-6, under a particular run. The 3 objectives are plotted separately in the first ($\boldsymbol{y}_1$ vs. $\boldsymbol{y}_2$) and second ($\boldsymbol{y}_3$ vs. $\boldsymbol{y}_2$) column.

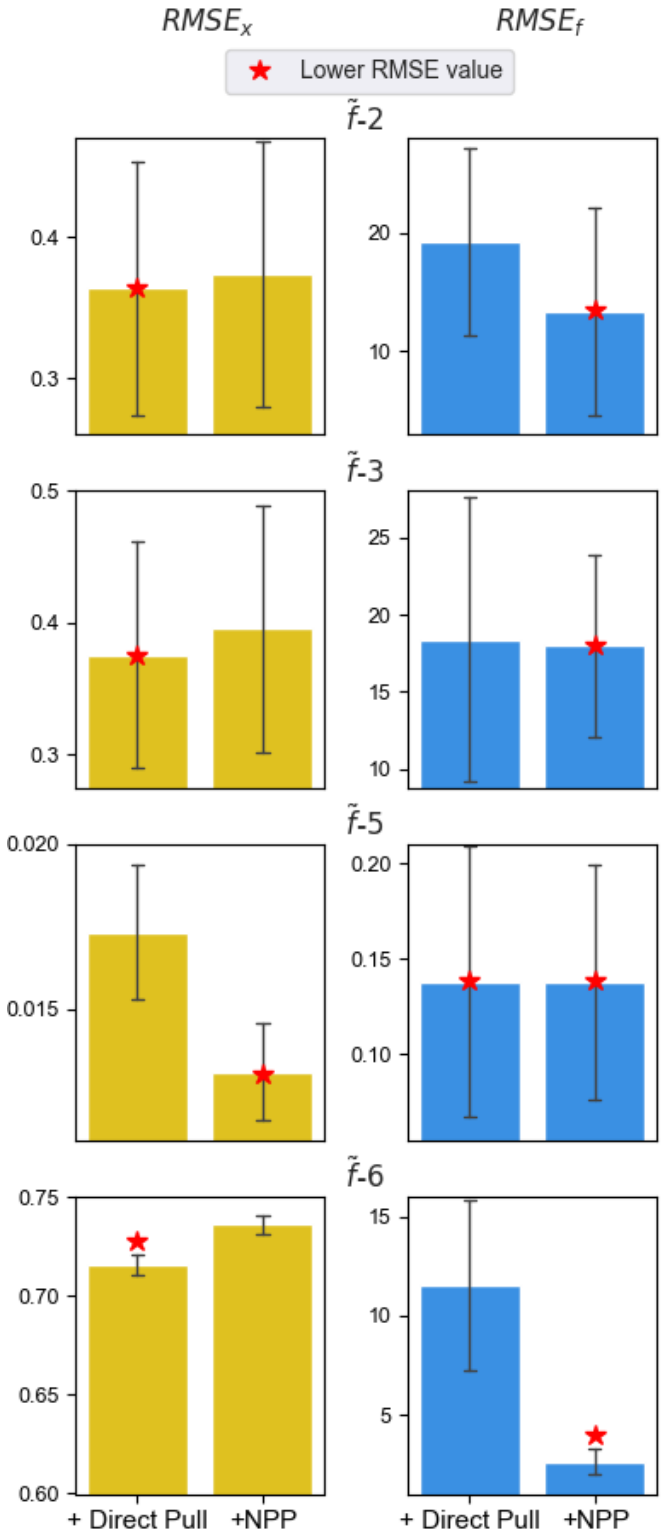

Figure 7.7: Comparison of the average $RMSE_{\boldsymbol{x}}$ (yellow bar) and $RMSE_{\boldsymbol{f}}$ (blue bar) values and its standard deviation (error bar) over 50 independent runs for $\tilde{\boldsymbol{f}}$-2, $\tilde{\boldsymbol{f}}$-3, $\tilde{\boldsymbol{f}}$-5 and $\tilde{\boldsymbol{f}}$-6. The x and y axis represent the projection methods utilized and average RMSE values, respectively, while the star symbol highlights a lower average RMSE value.

Additionally, a closer examination of the $RMSE_{\boldsymbol{f}}$ values in Table 7.4 highlights an exceptional case in $\tilde{\boldsymbol{f}}$-5. Despite achieving a notable $\sim 35\%$ reduction in $RMSE\boldsymbol{x}$ values by the NPP method, it stands out as the sole test problem exhibiting worsened $RMSE_{\boldsymbol{f}}$ values after projections. To elucidate this counter-intuitive finding, further analysis of the distribution of the solutions provided by the NPP method in both the objective and decision spaces is conducted, as depicted in Fig. 7.8.

Examining the top row of Fig. 7.8, the NPP method manages to improve the convergence of most solutions, except for the remaining 10% of the outlier solutions (highlighted in red) that are further away from the true PF. These outliers have mainly contributed to the overall increase of the $RMSE_{\boldsymbol{f}}$ value of the NPP method. However, when analyzing the solutions in the decision space, shown in the middle row of Fig. 7.8, the NPP method provides a closer approximation of the true PS, as compared to *inv*GP. Furthermore, focusing only on the outlier solutions in the decision space, the NPP method has a surprisingly lower $RMSE_{\boldsymbol{x}}$ value than the initial solutions before projection (predicted by *inv*GP), as observed in the bottom row of Fig. 7.8.

This underscores a critical insight: merely optimizing predictive error in the decision space, without considering its performance in the objective space, may prove inadequate. As exemplified by the case of $\tilde{\boldsymbol{f}}_5$, solutions close to a sub-region of the true PS, though appearing near-optimal, can actually be far from PF convergence. Therefore, incorporating insights on the performance evaluated in the objective space is essential for robustly learning the PS manifold. This allows for the generation of solutions that faithfully represent different sub-regions of the true PF, as desired by the DM.

This finding closely aligns with the recent paradigm in *Smart, Predict, then Optimize* [149–151], which integrates predictive models with downstream tasks. Here, the predictive model is trained based on the performance losses in the downstream task caused by inaccurate predictions. This encourages the exploration of a new research direction for the inverse modeling framework guided

by similar principles in the field of *Smart, Predict, then Optimize*.

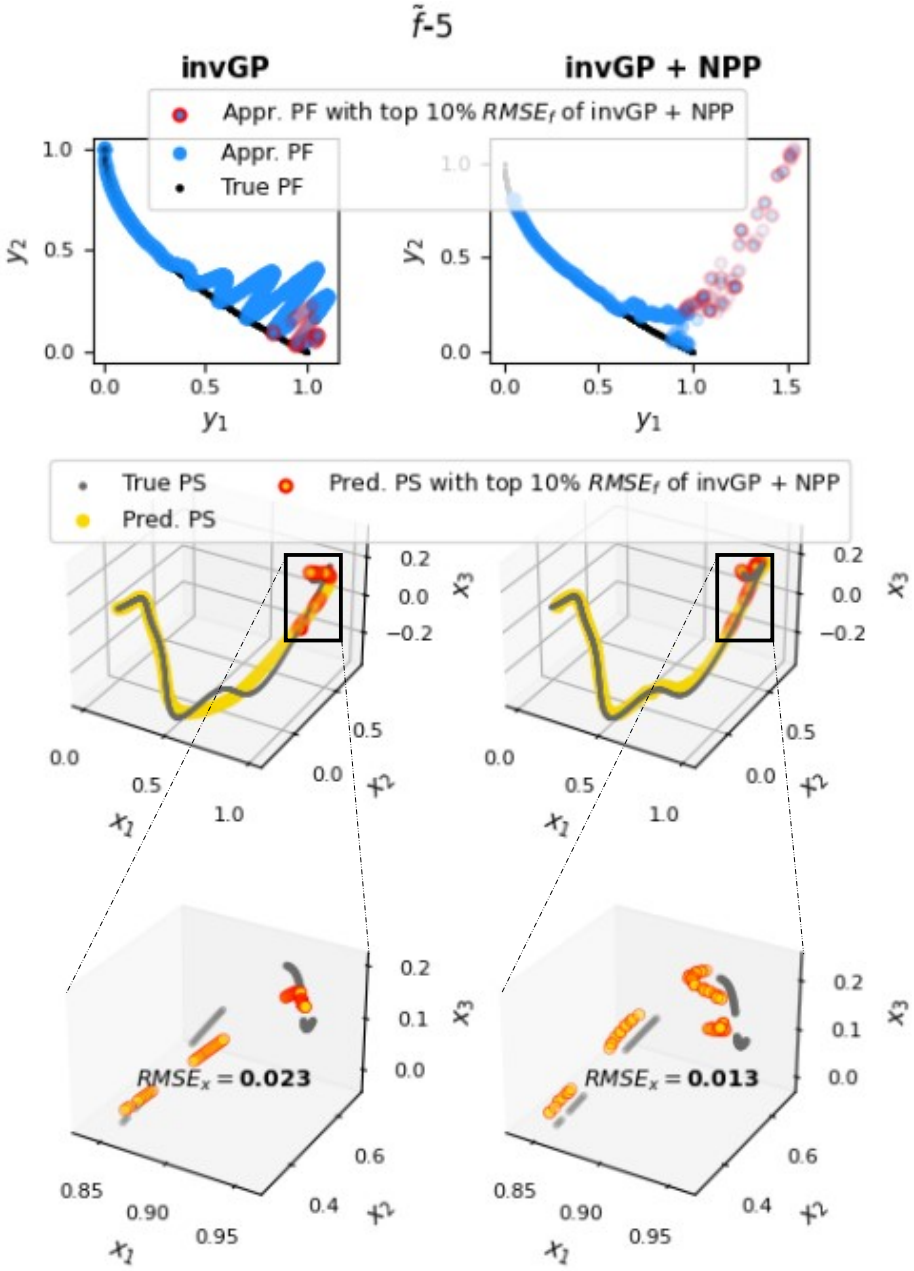

Figure 7.8: Scatter plots of the predicted solutions by *inv*GP (left) and NPP method (right) for $\tilde{\boldsymbol{f}}$-5, within a specific run. Outlier solutions from the NPP method that, contribute to the top 10% $RMSE_{\boldsymbol{f}}$ values are highlighted in red across all plots. The top row shows the true (in black) and approximated (in blue) PF. The middle row displays true (in grey) and predicted (in yellow) PS for the first three decision variables. The bottom row focuses on outlier solutions and its $RMSE_{\boldsymbol{x}}$ values.

7.3.4 Performance in Pareto Front Approximation

The performance of post-hoc PS learning in enhancing the capacity of PF approximation is evaluated using the IGD ratio in Eq. (5.5). This metric offers a board overview to the convergence and spread of the predicted solutions towards the true PF.

Table 7.5 presents the IGD ratios for the *inv*GP and the projection methods. The *inv*GP generally fails to improve the IGD values across most test problems, except for $\tilde{\boldsymbol{f}}$-5, where a 1.5-fold improvement is noted. Through the projection methods, the IGD ratios are improved for nearly half of the nine test problems, with the direct pull method performing slightly better. Notably, a substantial $\sim 89\%$ improvement to the IGD ratio of the *inv*GP is achieved by the direct pull method for test problem $\tilde{\boldsymbol{f}}$-1.

Table 7.5: PF approximation improvement measured in IGD ratio for *inv*GP, direct pull and NPP method for $\tilde{\boldsymbol{f}}$-1 to $\tilde{\boldsymbol{f}}$-9. Values in bold and bracket indicate the best averaged and standard deviations of the performance for each MOP.

Instance	***inv*GP**	**+Direct Pull**	**+NPP**
		IGDRatio	
$\tilde{\boldsymbol{f}}$-1	1.201 (0.069)	**2.273** (0.186)	2.228 (0.190)
$\tilde{\boldsymbol{f}}$-2	1.000 (0.000)	**1.003** (0.004)	1.002 (0.003)
$\tilde{\boldsymbol{f}}$-3	1.000 (0.001)	1.000 (0.000)	1.000 (0.000)
$\tilde{\boldsymbol{f}}$-4	**1.003** (0.003)	1.000 (0.000)	1.000 (0.000)
$\tilde{\boldsymbol{f}}$-5	**1.516** (0.042)	1.037 (0.043)	1.037 (0.044)
$\tilde{\boldsymbol{f}}$-6	1.004 (0.000)	**1.008**(0.002)	1.005 (0.002)
$\tilde{\boldsymbol{f}}$-7	1.000 (0.000)	1.224 (0.195)	**1.239** (0.219)
$\tilde{\boldsymbol{f}}$-8	1.001 (0.001)	**1.236** (0.270)	1.234 (0.263)
$\tilde{\boldsymbol{f}}$-9	1.000 (0.000)	**1.001**(0.003)	1.000 (0.001)

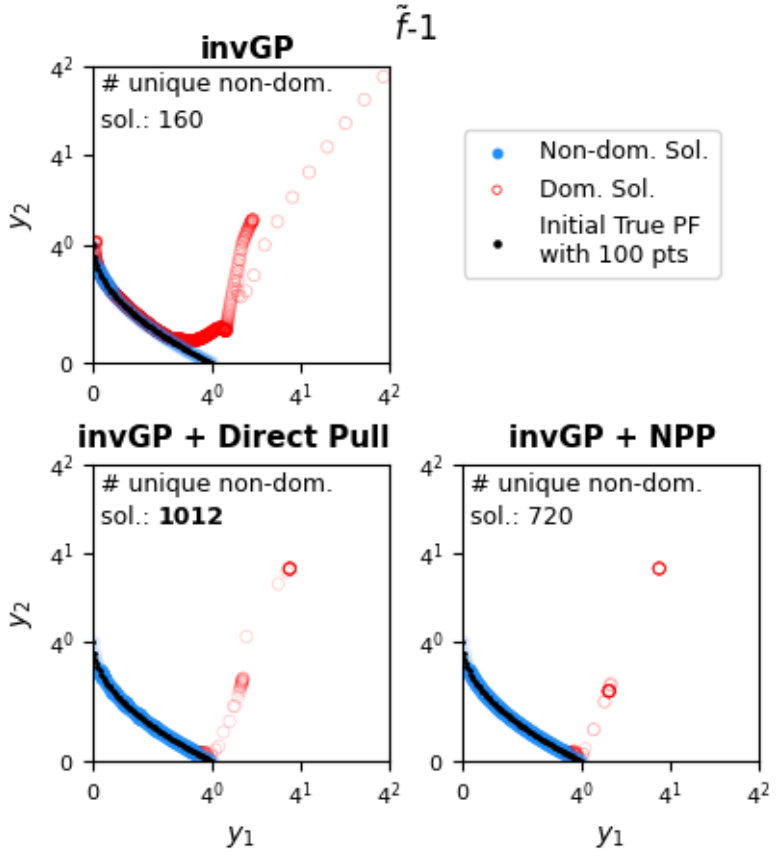

Figure 7.9: 2-D scatter plots of the true PF with 100 data points (in black) and approximated PF by the *inv*GP (top), direct pull (bottom, left) and NPP methods (bottom, right). The approximated PFs are further differentiated based on dominated (in red) and non-dominated (in blue) solutions.

Fig. 7.9 shows the PF approximated by the *inv*GP and the projection methods. The predicted solutions within the approximated PF are further separated into non-dominated and dominated solutions for the ease of analysis. The projection methods greatly complement the *inv*GP predictions by increasing the number of non-dominated solutions to up to 6.3 times, resulting in a denser approximation of the true PF. As compared to NPP method, the direct pull is able to better increase the solution diversity with $\sim 29\%$ higher number of unique non-dominated solutions. This again, is due to the underlying workings of NPP method, as discussed in the Section 7.3.2, where multiple solutions are projected onto the same point within the augmented PS.

7.4 Chapter Summary

This chapter takes an important advancement towards generative multi-objective decision support via preference prompts, particularly in expensive optimization domains marked by data scarcity. To this end, a pioneering concept in Monte Carlo walk-pull for data augmentation within the non-linear PS is introduced. Additionally, two projection functions derived from the walk-pull process are proposed to effectively map predictions from inverse models closer to the desired solutions.

The proposed methodology is verified through extensive empirical tests, featuring nine MOP benchmarks with highly non-linear PS shapes. The findings demonstrate promising results and highlight distinct advantages of the two projection methods for non-linear PS learning. Specifically, the direct pull method stands out in enhancing PS learning accuracy, while the NPP method, which references the augmented PS data, provides reliable estimates for mitigating highly inaccurate predictions.

A key insight from an exceptional case reveals that optimizing accuracy loss in the decision space alone is insufficient to generate solutions with objective values that closely satisfy the DM's preferences. This points towards future endeavour of *preference-aligned, post-hoc PS learning*, integrating PS learning with *Smart, Predict, then Optimize* framework. Such an endeavor shall potentially illuminate a new paradigm of learning the PS manifold guided by performance losses in the objective space.

Chapter 8

Conclusion and Future Work

This thesis tackles the unresolved research challenges in *a posteriori* multi-criteria decision making for MOPs marked by data scarcity. To address these challenges, three innovative approaches for improving performance in post-hoc PS learning have been introduced.

The first approach leverages multi-source inverse transfer learning, exploiting MOPs with common objective spaces to maximize information use between heterogeneous source-target pairs. The second approach focuses on improving the computational efficiency of inverse models for MOPs with many decision variables. This is achieved through PS representation learning that reduces the high-dimensional decision space to its smallest possible dimensions to enable the development of compact yet accurate inverse models. The final approach assumes a pessimistic case of limited target Pareto data and no source data for transfer. In this worst-case scenario, a novel Monte Carlo walk-pull concept is introduced for Pareto data augmentation. Two complementary techniques are devised to couple the outcomes of the walk-pull process with inverse machine models, improving overall performance. All three approaches have been empirically validated with benchmark MOPs and real-world problems, showcasing notable performance gains in the decision and objective spaces of inverse models trained with limited data.

8.1 Future Work

Given the promising results from the experimental studies, two board future research strands for inverse modeling in the area of PS learning are proposed. The first research strand involves continuously improving post-hoc PS learning performance. The second focuses on developing innovative generative multi-objective algorithms that integrate lightweight inverse models to generate new promising non-dominated solutions during the optimization process.

8.1.1 Post-hoc Pareto Set learning

In the area of post-hoc PS learning, two key extensions building on the approaches from Chapter 5 to Chapter 7 are planned for future work, as illustrated in Figure 8.1.

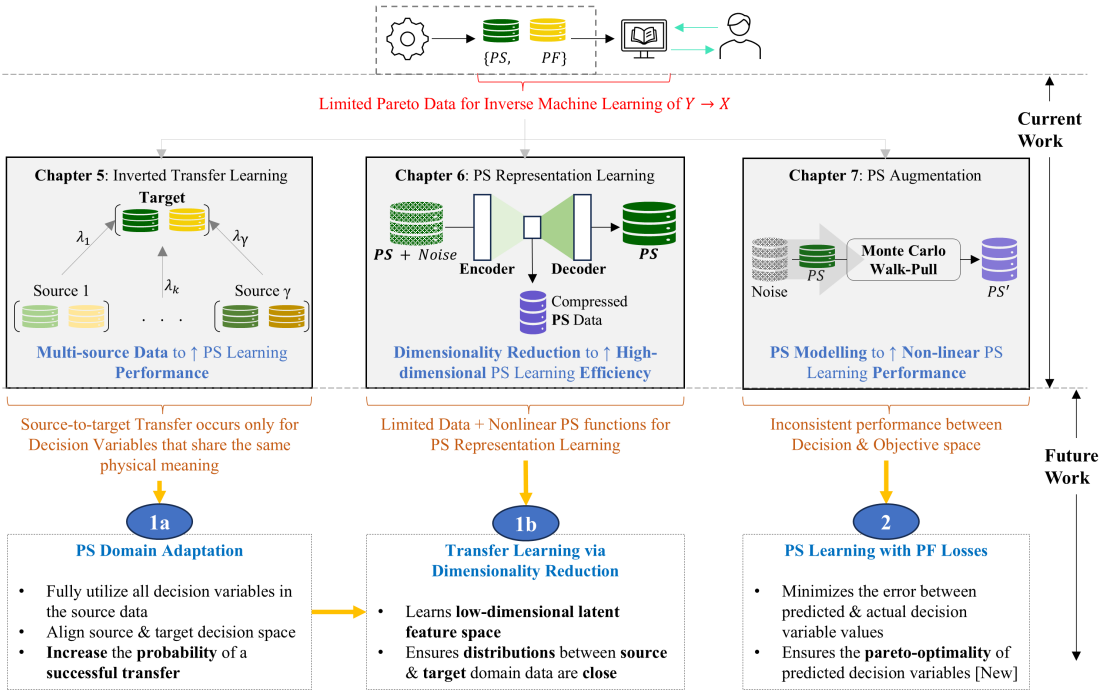

Figure 8.1: Future work for post-hoc PS learning.

To address the limited scope of source-to-target transfer in Chapter 5, which focuses only on decision variables with the same physical meaning, domain adaptation techniques can be applied. A promising approach to leveraging all

decision variables in the source data is to learn a mapping from the source to target decision space [152]. This increases the overlap between source and target decision spaces, enhancing the likelihood of successful transfer during PS learning. Figure 8.2 illustrates this concept for a single-source transfer, where the source and target decision spaces differ in magnitude and dimensionality. In this scenario, a machine learning model can be trained to map each point in the source PS $X_{\mathcal{S}}$ to the target PS $X_{\mathcal{T}}$ by aligning them based on their corresponding locations in the PF.

Furthermore, to enhance PS representation learning in Chapter 6, especially for high-dimensional, limited, and non-linear PS data, transfer learning via dimensionality reduction can be explored. Given an available source dataset, existing dimensionality reduction methods can be reformulated to identify a low-dimensional latent space that minimizes the distributional divergence between domains. A notable example in this area, proposed by [153], employs a universal kernel to minimize the maximum mean discrepancy between source and target data distributions. Once the kernel is obtained, principal component analysis is applied to select leading eigenvectors for constructing the low-dimensional representations.

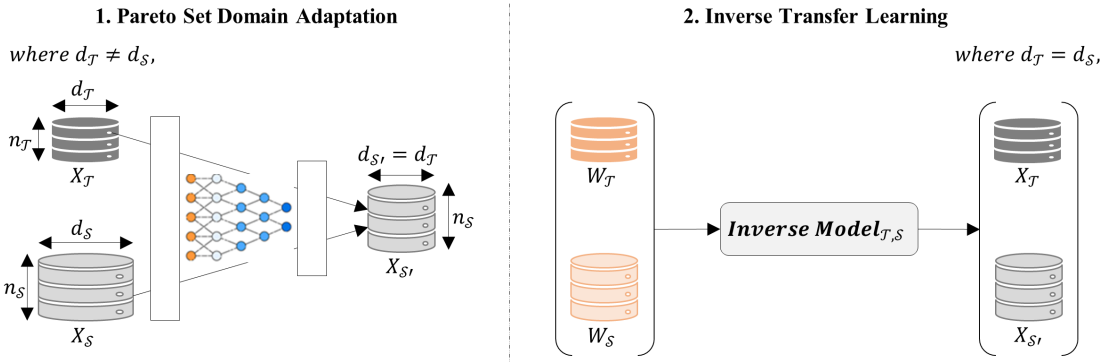

Figure 8.2: Domain adaptation of heterogeneous decision spaces of source-target task for efficient inverse transfer learning. The dimensionality of source and target PS are denoted by $d_{\mathcal{S}}$ and $d_{\mathcal{T}}$, respectively. Additionally, $n_{\mathcal{S}}$ and $n_{\mathcal{S}}$ represent the number of available Pareto data for the source and target tasks.

The second extension is motivated by the exception case in Chapter 7, where

improvements in the decision space do not always translate to corresponding gains in the objective space. This limitation can reduce its effectiveness in providing DMs with alternative decision variables that remains Pareto optimal while offering different trade-offs among objective values. To address this inconsistency, a promising research direction is to ensure preference-aligned PS learning. The key idea is to integrate the inverse modeling framework with the *Smart, Predict, then Optimize* approach [149–151]. Specifically, this involves training an inverse model to minimize the error between the objective values of predicted and actual decision variables. This synergy could introduce a novel paradigm for learning the PS manifold, guided by performance losses in the objective space. Figure 8.3a illustrates this concept, where a loss function approximator estimates objective losses during PS learning. Recent studies [154, 155] have explored this idea, further supporting its potential.

Additionally, to enhance the performance of the two projection methods in Chapter 7, this approach can be adapted to the Monte Carlo walk-pull process. Specifically, during the pull process, the neural network responsible for restoring noisy PS data can be trained and evaluated by backpropagating objective losses, as shown in Figure 8.3b. This adaptation could improve the robustness of projection functions against out-of-distribution projections while preserving the diversity of projected points.

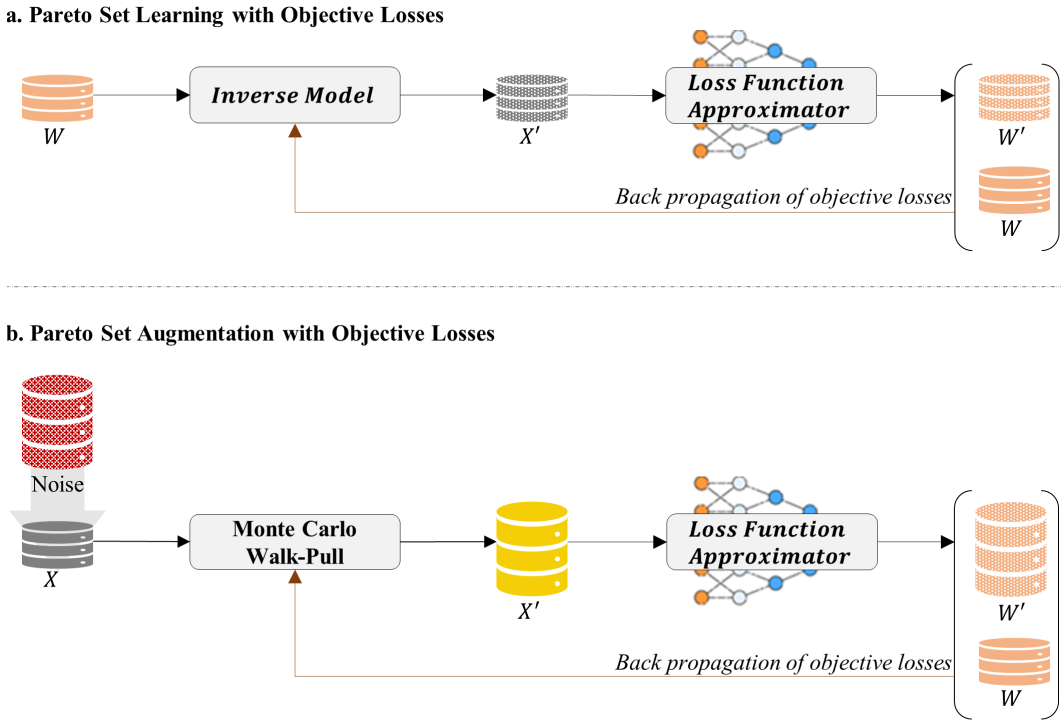

Figure 8.3: Improving the performances of (a) PS learning and (b) PS data augmentation through objective losses estimated by a loss function approximator.

8.1.2 Online Pareto Set Learning

The second research strand is motivated by the promising improvements observed in the IGD ratio, highlighting the potential to extend the post-hoc PS learning algorithms developed in this thesis to lightweight online learning models. These models could enable the generative modeling of non-dominated solutions during optimization. Figure 8.4 illustrates how the approaches from Chapter 5 and Chapter 7 could be integrated with existing MOEAs.

Recent pioneering work in [99] has begun exploring this direction, leveraging online PS transfer learning to solve related optimization tasks simultaneously. This approach was later extended by the same authors to incorporate multi-task surrogate modeling, enabling the efficient evaluation of expensive objective functions across source and target tasks [100].

This emerging research direction holds significant promise for developing gen-

erative multi-objective algorithms with faster convergence and greater diversity in the final Pareto optimal solutions.

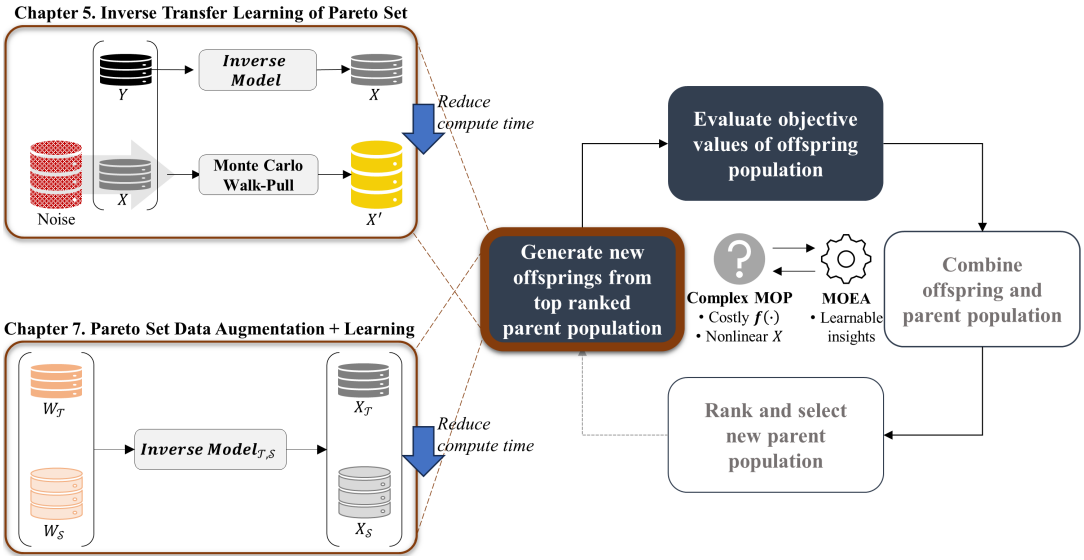

Figure 8.4: Generative multi-objective algorithms incorporating online PS learning algorithms.

References

[1] Xinsong Niu and Jiyang Wang. A combined model based on data preprocessing strategy and multi-objective optimization algorithm for short-term wind speed forecasting. *Applied Energy*, 241:519–539, 2019.

[2] Nauman Aslam, William Phillips, William Robertson, and Shyamala Sivakumar. A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks. *Information Fusion*, 12(3):202–212, 2011.

[3] Hongze Wang, Xuerong Li, Wenjing Hong, and Ke Tang. Multi-objective approaches to portfolio optimization with market impact costs. *Memetic Computing*, pages 1–11, 2022.

[4] Vadlamani Ravi, Dadabada Pradeepkumar, and Kalyanmoy Deb. Financial time series prediction using hybrids of chaos theory, multi-layer perceptron and multi-objective evolutionary algorithms. *Swarm and Evolutionary Computation*, 36:136–149, 2017.

[5] Abhishek Gupta, Chen Kim Heng, Yew-Soon Ong, Puay Siew Tan, and Allan N Zhang. A generic framework for multi-criteria decision support in eco-friendly urban logistics systems. *Expert Systems with Applications*, 71:288–300, 2017.

[6] Zizhen Zhang, Hu Qin, and Yanzhi Li. Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1987–2001, 2019.

[7] CS Tan, A Gupta, and C Xu. Are two heads always better than one? human-ai complementarity in multi-criteria order planning. In *2022 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0939–0943. IEEE, 2022.

[8] Qihao Liu, Xinyu Li, Liang Gao, and Guangchen Wang. A multiobjective memetic algorithm for integrated process planning and scheduling problem in distributed heterogeneous manufacturing systems. *Memetic Computing*, 14(2):193–209, 2022.

[9] Rui Wang, Robin C Purshouse, and Peter J Fleming. Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(4):474–494, 2012.

[10] Ran Cheng, Tobias Rodemann, Michael Fischer, Markus Olhofer, and Yaochu Jin. Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 1(2):97–111, 2017.

[11] Rui Wang, Robin C Purshouse, and Peter J Fleming. Preference-inspired co-evolutionary algorithms using weight vectors. *European Journal of Operational Research*, 243(2):423–441, 2015.

[12] Rui Wang, Zhongbao Zhou, Hisao Ishibuchi, Tianjun Liao, and Tao Zhang. Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 22(1):3–18, 2016.

[13] Luciana R Pedro and Ricardo HC Takahashi. Modeling decision-maker preferences through utility function level sets. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 550–563. Springer, 2011.

[14] Ian Dewancker, Michael McCourt, and Samuel Ainsworth. Interactive preference learning of utility functions for multi-objective optimization. *arXiv preprint arXiv:1612.04453*, 2016.

[15] Raul Astudillo and Peter Frazier. Multi-attribute bayesian optimization with interactive preference learning. In *International Conference on Artificial Intelligence and Statistics*, pages 4496–4507. PMLR, 2020.

[16] Nawal Benabbou, Cassandre Leroy, and Thibaut Lust. An interactive regret-based genetic algorithm for solving multi-objective combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34-03, pages 2335–2342, 2020.

[17] Lizhen Shao and Matthias Ehrgott. Discrete representation of non-dominated sets in multi-objective linear programming. *European Journal of Operational Research*, 255(3):687–698, 2016.

[18] Silvia Carpitella, Antonella Certa, Joaquín Izquierdo, and Concetta Manuela La Fata. k-out-of-n systems: An exact formula for the stationary availability and multi-objective configuration design based on mathematical programming and topsis. *Journal of Computational and Applied Mathematics*, 330:1007–1015, 2018.

[19] Sune Lauth Gadegaard, Lars Relund Nielsen, and Matthias Ehrgott. Bi-objective branch-and-cut algorithms based on lp relaxation and bound sets. *INFORMS Journal on Computing*, 31(4):790–804, 2019.

[20] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007.

[21] Lie Meng Pang, Hisao Ishibuchi, and Ke Shang. Nsga-ii with simple modification works well on a wide variety of many-objective problems. *IEEE Access*, 8:190240–190250, 2020.

[22] Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

[23] Anupam Trivedi, Dipti Srinivasan, Krishnendu Sanyal, and Abhiroop Ghosh. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462, 2016.

[24] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[25] Jesús Guillermo Falcón-Cardona and Carlos A Coello Coello. Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 53(2):1–35, 2020.

[26] Abhishek Gupta, Yew-Soon Ong, Mojtaba Shakeri, Xu Chi, and Allan Zhang NengSheng. The blessing of dimensionality in many-objective search: An inverse machine learning insight. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 3896–3902. IEEE, 2019.

[27] Kyohei Hanaoka. Bayesian optimization for goal-oriented multi-objective inverse material design. *Iscience*, 24(7), 2021.

[28] Jürgen Branke and Kalyanmoy Deb. Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge incorporation in evolutionary computation*, pages 461–477. Springer, 2005.

[29] Pradyumn Kumar Shukla, Michael Emmerich, and André Deutz. A theoretical analysis of curvature based preference models. In *Evolutionary Multi-Criterion Optimization: 7th International Conference, EMO 2013, Sheffield, UK, March 19-22, 2013. Proceedings 7*, pages 367–382. Springer, 2013.

[30] Thomas L Saaty. Decision making with the analytic hierarchy process. *International journal of services sciences*, 1(1):83–98, 2008.

[31] Kalyanmoy Deb, Ankur Sinha, Pekka J Korhonen, and Jyrki Wallenius. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010.

[32] Michał K Tomczyk and Miłosz Kadziński. Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation*, 24(2):320–334, 2019.

[33] Kendall Taylor, Huong Ha, Minyi Li, Jeffrey Chan, and Xiaodong Li. Bayesian preference learning for interactive multi-objective optimisation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 466–475, 2021.

[34] Ankur Sinha. *Progressively interactive evolutionary multiobjective optimization*. Aalto University, 2011.

[35] Jürgen Branke, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz. Learning value functions in interactive evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 19(1):88–102, 2014.

[36] Ke Li, Guiyu Lai, and Xin Yao. Interactive evolutionary multi-objective optimization via learning-to-rank. *IEEE Transactions on Evolutionary Computation*, 2023.

[37] Alfredo Arias-Montano, Carlos A Coello Coello, and Efrén Mezura-Montes. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE transactions on evolutionary computation*, 16(5):662–694, 2012.

[38] Antonin Ponsich, Antonio Lopez Jaimes, and Carlos A Coello Coello. A survey on multiobjective evolutionary algorithms for the solution of the

portfolio optimization problem and other finance and economics applications. *IEEE Transactions on evolutionary computation*, 17(3):321–344, 2012.

[39] Carlos A Coello Coello, Silvia González Brambila, Josué Figueroa Gamboa, Ma Guadalupe Castillo Tapia, and Raquel Hernández Gómez. Evolutionary multiobjective optimization: open research areas and some challenges lying ahead. *Complex & Intelligent Systems*, 6:221–236, 2020.

[40] Songbai Liu, Qiuzhen Lin, Jianqiang Li, and Kay Chen Tan. A survey on learnable evolutionary algorithms for scalable multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2023.

[41] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK report*, 103, 2001.

[42] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer, 2004.

[43] Johannes Bader and Eckart Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.

[44] Raquel Hernández Gómez and Carlos A Coello Coello. Mombi: A new metaheuristic for many-objective optimization based on the r2 indicator. In *2013 IEEE congress on evolutionary computation*, pages 2488–2495. IEEE, 2013.

[45] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[46] Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the properties of the r2 indicator. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 465–472, 2012.

[47] Wen-xiang Wang, Kang-shun Li, Xing-zhen Tao, and Fa-hui Gu. An improved moea/d algorithm with an adaptive evolutionary strategy. *Information Sciences*, 539:1–15, 2020.

[48] Kaisa Miettinen. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media, 1999.

[49] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[50] Yuanchao Liu, Jianchang Liu, Shubin Tan, Yongkuan Yang, and Fei Li. A bagging-based surrogate-assisted evolutionary algorithm for expensive multi-objective optimization. *Neural Computing and Applications*, 34(14):12097–12118, 2022.

[51] Jinyuan Zhang, Linjun He, and Hisao Ishibuchi. Dual fuzzy classifier-based evolutionary algorithm for expensive multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2022.

[52] Huachao Dong, Jinglu Li, Peng Wang, Baowei Song, and Xinkai Yu. Surrogate-guided multi-objective optimization (sgmoo) using an efficient online sampling strategy. *Knowledge-Based Systems*, 220:106919, 2021.

[53] Yi Zhao, Chaoli Sun, Jianchao Zeng, Ying Tan, and Guochen Zhang. A surrogate-ensemble assisted expensive many-objective optimization. *Knowledge-Based Systems*, 211:106520, 2021.

[54] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[55] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[56] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13:455–492, 1998.

[57] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

[58] Xilu Wang, Yaochu Jin, Sebastian Schmitt, and Markus Olhofer. An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization. *Information Sciences*, 519:317–331, 2020.

[59] Zhenshou Song, Handing Wang, Bing Xue, Mengjie Zhang, and Yaochu Jin. Balancing objective optimization and constraint satisfaction in expensive constrained evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 2023.

[60] Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Multi-objective bayesian optimization over high-dimensional search spaces. In *Uncertainty in Artificial Intelligence*, pages 507–517. PMLR, 2022.

[61] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning*, pages 20459–20478. PMLR, 2022.

[62] David Eriksson and Martin Jankowiak. High-dimensional bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pages 493–503. PMLR, 2021.

[63] Dan Guo, Xilu Wang, Kailai Gao, Yaochu Jin, Jinliang Ding, and Tianyou Chai. Evolutionary optimization of high-dimensional multiob-

jective and many-objective expensive problems assisted by a dropout neural network. *IEEE transactions on systems, man, and cybernetics: systems*, 52(4):2084–2097, 2021.

[64] Jianping Luo, Abhishek Gupta, Yew-Soon Ong, and Zhenkun Wang. Evolutionary optimization of expensive multiobjective problems with co-sub-pareto front gaussian process surrogates. *IEEE transactions on cybernetics*, 49(5):1708–1721, 2018.

[65] Xilu Wang, Yaochu Jin, Sebastian Schmitt, Markus Olhofer, and Richard Allmendinger. Transfer learning based surrogate assisted evolutionary bi-objective optimization for objectives with different evaluation times. *Knowledge-Based Systems*, 227:107190, 2021.

[66] Alan Tan Wei Min, Yew-Soon Ong, Abhishek Gupta, and Chi-Keong Goh. Multiproblem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):15–28, 2017.

[67] Jinsu Gim, Huaguang Yang, and Lih-Sheng Turng. Transfer learning of machine learning models for multi-objective process optimization of a transferred mold to ensure efficient and robust injection molding of high surface quality parts. *Journal of Manufacturing Processes*, 87:11–24, 2023.

[68] Jiao Liu, Abhishek Gupta, Chinchun Ooi, and Yew-Soon Ong. Extremo: Transfer evolutionary multiobjective optimization with proof of faster convergence. *IEEE Transactions on Evolutionary Computation*, 2024.

[69] Wesley J Maddox, Maximilian Balandat, Andrew G Wilson, and Eytan Bakshy. Bayesian optimization with high-dimensional outputs. *Advances in neural information processing systems*, 34:19274–19287, 2021.

[70] Stephen Friess, Peter Tiňo, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. Learning transferable variation operators in a continuous genetic

algorithm. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2027–2033. IEEE, 2019.

[71] Dhish Kumar Saxena, Sukrit Mittal, Kalyanmoy Deb, and Erik D Goodman. Investigating innovized progress operators with different ml methods. In *Machine Learning Assisted Evolutionary Multi-and Many-Objective Optimization*, pages 197–215. Springer, 2024.

[72] Mark Hauschild and Martin Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and evolutionary computation*, 1(3):111–128, 2011.

[73] Chang Wook Ahn and Rudrapatna S Ramakrishna. Multiobjective real-coded bayesian optimization algorithmrevisited: diversity preservation. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 593–600, 2007.

[74] Yong Wang, Jian Xiang, and Zixing Cai. A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator. *Applied Soft Computing*, 12(11):3526–3538, 2012.

[75] Yangyang Li, Xia Xu, Peidao Li, and Licheng Jiao. Improved rm-meda with local learning. *Soft Computing*, 18:1383–1397, 2014.

[76] Peter AN Bosman and Dirk Thierens. Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *International Journal of Approximate Reasoning*, 31(3):259–289, 2002.

[77] Hossein Karshenas, Roberto Santana, Concha Bielza, and Pedro Larranaga. Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables. *IEEE Transactions on Evolutionary Computation*, 18(4):519–542, 2013.

[78] Cheng He, Shihua Huang, Ran Cheng, Kay Chen Tan, and Yaochu Jin. Evolutionary multiobjective optimization driven by generative adversar-

ial networks (gans). *IEEE transactions on cybernetics*, 51(6):3129–3142, 2020.

[79] Zhenzhong Wang, Haokai Hong, Kai Ye, Guang-En Zhang, Min Jiang, and Kay Chen Tan. Manifold interpolation for large-scale multiobjective optimization via generative adversarial networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):4631–4645, 2021.

[80] Jiabin Lin, Hai-Lin Liu, Kay Chen Tan, and Fangqing Gu. An effective knowledge transfer approach for multiobjective multitasking optimization. *IEEE transactions on cybernetics*, 51(6):3238–3248, 2020.

[81] Hongyan Chen, Hai-Lin Liu, Fangqing Gu, and Kay Chen Tan. A multiobjective multitask optimization algorithm using transfer rank. *IEEE Transactions on Evolutionary Computation*, 27(2):237–250, 2022.

[82] Kavitesh Kumar Bali, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. *IEEE Transactions on Evolutionary Computation*, 24(1):69–83, 2019.

[83] Ting Wu, Siqi Bu, Xiang Wei, Guibin Wang, and Bin Zhou. Multitasking multi-objective operation optimization of integrated energy system considering biogas-solar-wind renewables. *Energy conversion and management*, 229:113736, 2021.

[84] Jun Zhang, Weien Zhou, Xianqi Chen, Wen Yao, and Lu Cao. Multisource selective transfer framework in multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(3):424–438, 2019.

[85] Xianpeng Wang, Zhiming Dong, Lixin Tang, and Qingfu Zhang. Multiobjective multitask optimization-neighborhood as a bridge for knowledge transfer. *IEEE Transactions on Evolutionary Computation*, 27(1):155–169, 2022.

[86] Zhengping Liang, Jiabiao Yan, Fan Zheng, Jigang Wang, Ling Liu, and Zexuan Zhu. Multi-objective multi-task particle swarm optimization based on objective space division and adaptive transfer. *Expert Systems with Applications*, 255:124618, 2024.

[87] Kangjia Qiao, Jing Liang, Kunjie Yu, Minghui Wang, Boyang Qu, Caitong Yue, and Yinan Guo. A self-adaptive evolutionary multi-task based constrained multi-objective evolutionary algorithm. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(4):1098–1112, 2023.

[88] Ray Lim, Lei Zhou, Abhishek Gupta, Yew-Soon Ong, and Allan N Zhang. Solution representation learning in multi-objective transfer evolutionary optimization. *IEEE Access*, 9:41844–41860, 2021.

[89] Zhengping Liang, Weiqi Liang, Zhiqiang Wang, Xiaoliang Ma, Ling Liu, and Zexuan Zhu. Multiobjective evolutionary multitasking with two-stage adaptive knowledge transfer based on population distribution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(7):4457–4469, 2021.

[90] Songbai Liu, Qiuzhen Lin, Liang Feng, Ka-Chun Wong, and Kay Chen Tan. Evolutionary multitasking for large-scale multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 2022.

[91] Ran Cheng, Yaochu Jin, Kaname Narukawa, and Bernhard Sendhoff. A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *IEEE Transactions on Evolutionary Computation*, 19(6):838–856, 2015.

[92] Ran Cheng, Yaochu Jin, and Kaname Narukawa. Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 127–140. Springer, 2015.

[93] Lucas RC Farias and Aluizio FR Araújo. Im-moea/d: An inverse modeling multi-objective evolutionary algorithm based on decomposition. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 462–467. IEEE, 2021.

[94] Pezhman Gholamnezhad, Ali Broumandnia, and Vahid Seydi. An inverse model-based multiobjective estimation of distribution algorithm using random-forest variable importance methods. *Computational Intelligence*, 38(3):1018–1056, 2022.

[95] Zhechen Zhang, Sanyang Liu, Weifeng Gao, Jingwei Xu, and Shengqi Zhu. An enhanced multi-objective evolutionary optimization algorithm with inverse model. *Information Sciences*, 530:128–147, 2020.

[96] Xi Lin, Zhiyuan Yang, and Qingfu Zhang. Pareto set learning for neural multi-objective combinatorial optimization. In *International Conference on Learning Representations*, 2021.

[97] Xi Lin, Zhiyuan Yang, Xiaoyuan Zhang, and Qingfu Zhang. Pareto set learning for expensive multi-objective optimization. *arXiv preprint arXiv:2210.08495*, 2022.

[98] Jiao Liu, Abhishek Gupta, Yew-Soon Ong, and Puay Siew Tan. Inverse multiobjective optimization by generative model prompting. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 737–740. IEEE, 2024.

[99] Jiao Liu, Abhishek Gupta, and Yew-Soon Ong. Bayesian inverse transfer in evolutionary multiobjective optimization. *ACM Transactions on Evolutionary Learning*, 4(4):1–27, 2024.

[100] Tingyang Wei, Jiao Liu, Abhishek Gupta, Puay Siew Tan, and Yew-Soon Ong. Bayesian forward-inverse transfer for multiobjective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 135–152. Springer, 2024.

[101] Keith D Humfeld, Dawei Gu, Geoffrey A Butler, Karl Nelson, and Navid Zobeiry. A machine learning framework for real-time inverse modeling and multi-objective process optimization of composites for active manufacturing control. *Composites Part B: Engineering*, 223:109150, 2021.

[102] Xiaxia Li, Jingming Yang, Hao Sun, Ziyu Hu, and Anran Cao. A dual prediction strategy with inverse model for evolutionary dynamic multiobjective optimization. *ISA transactions*, 117:196–209, 2021.

[103] Ioannis Giagkiozis and Peter J Fleming. Pareto front estimation for decision making. *Evolutionary computation*, 22(4):651–678, 2014.

[104] Yiming Yan, Ioannis Giagkiozis, and Peter J Fleming. Improved sampling of decision space for pareto estimation. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 767–774, 2015.

[105] Azam Asilian Bidgoli, Shahryar Rahnamayan, Bilgehan Erdem, Zekiye Erdem, Amin Ibrahim, Kalyanmoy Deb, and Ali Grami. Machine learning-based framework to cover optimal pareto-front in many-objective optimization. *Complex & Intelligent Systems*, 8(6):5287–5308, 2022.

[106] Rajesh Kudikala, Ioannis Giagkiozis, and P Fleming. Increasing the density of multi-objective multi-modal solutions using clustering and pareto estimation techniques. In *The 2013 World Congress in Computer Science Computer Engineering and Applied Computing*, 2013.

[107] Guo Yu, Yaochu Jin, Markus Olhofer, Qiqi Liu, and Wenli Du. Solution set augmentation for knee identification in multiobjective decision analysis. *IEEE Transactions on Cybernetics*, 2021.

[108] Anirudh Suresh and Kalyanmoy Deb. Machine learning based prediction of new pareto-optimal solutions from pseudo-weights. *IEEE Transactions on Evolutionary Computation*, 2023.

[109] Charles Audet, Jean Bigeon, Dominique Cartier, Sébastien Le Digabel, and Ludovic Salomon. Performance indicators in multiobjective optimization. *European journal of operational research*, 292(2):397–422, 2021.

[110] Ioannis Giagkiozis and Peter J Fleming. Increasing the density of available pareto optimal solutions. 2012.

[111] Wei Xing, Shireen Y Elhabian, Vahid Keshavarzzadeh, and Robert M Kirby. Shared-gp: learning interpretable shared hidden structure across data spaces for design space analysis and exploration. *J. Mech. Des*, pages 1–16, 2020.

[112] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer, 2015.

[113] Carrie J Cai, Emily Reif, Narayan Hegde, Jason Hipp, Been Kim, Daniel Smilkov, Martin Wattenberg, Fernanda Viegas, Greg S Corrado, Martin C Stumpe, et al. Human-centered tools for coping with imperfect algorithms during medical decision-making. In *Proceedings of the 2019 chi conference on human factors in computing systems*, pages 1–14, 2019.

[114] Mark Steyvers, Heliodoro Tejeda, Gavin Kerrigan, and Padhraic Smyth. Bayesian modeling of human–ai complementarity. *Proceedings of the National Academy of Sciences*, 119(11):e2111547119, 2022.

[115] Eduardo Fernández, Nelson Rangel-Valdez, Laura Cruz-Reyes, Claudia G Gomez-Santillan, and Carlos A Coello-Coello. Preference incor-

poration in moea/d using an outranking approach with imprecise model parameters. *Swarm and Evolutionary Computation*, 72:101097, 2022.

[116] Zhanglu Hou, Cheng He, and Ran Cheng. Reformulating preferences into constraints for evolutionary multi-and many-objective optimization. *Information sciences*, 541:1–15, 2020.

[117] Shamik Chaudhuri and Kalyanmoy Deb. An interactive evolutionary multi-objective optimization and decision making procedure. *Applied Soft Computing*, 10(2):496–511, 2010.

[118] Ray Lim, Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Allan N Zhang. Non-linear domain adaptation in transfer evolutionary optimization. *Cognitive Computation*, 13:290–307, 2021.

[119] Bingshui Da, Abhishek Gupta, and Yew-Soon Ong. Curbing negative influences online for seamless transfer evolutionary optimization. *IEEE transactions on cybernetics*, 49(12):4365–4378, 2018.

[120] Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

[121] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[122] Yew-Soon Ong and Abhishek Gupta. Air 5: Five pillars of artificial intelligence research. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(5):411–415, 2019.

[123] Chin Sheng Tan, Abhishek Gupta, Yew-Soon Ong, Mahardhika Pratama, Puay Siew Tan, and Siew Kei Lam. Pareto optimization with small data by learning across common objective spaces. *Scientific Reports*, 13(1):7842, 2023.

[124] Kalyanmoy Deb, Aryan Gondkar, and Suresh Anirudh. Learning to predict pareto-optimal solutions from pseudo-weights. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 191–204. Springer, 2023.

[125] Kalyanmoy Deb and Jayavelmurugan Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642, 2006.

[126] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

[127] Alan Tan Wei Min, Abhishek Gupta, and Yew-Soon Ong. Generalizing transfer bayesian optimization to source-target heterogeneity. *IEEE Transactions on Automation Science and Engineering*, 18(4):1754–1765, 2020.

[128] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

[129] Pengfei Wei, Ramon Sagarna, Yiping Ke, and Yew Soon Ong. Uncluttered domain sub-similarity modeling for transfer regression. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1314–1319. IEEE, 2018.

[130] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 2007.

[131] Marc Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490. PMLR, 2015.

[132] Samuel Cohen, Rendani Mbuvha, Tshilidzi Marwala, and Marc Deisenroth. Healing products of gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077. PMLR, 2020.

[133] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *IEEE transactions on neural networks and learning systems*, 31(11):4405–4423, 2020.

[134] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer, 2005.

[135] Marco Farina, Kalyanmoy Deb, and Paolo Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on evolutionary computation*, 8(5):425–442, 2004.

[136] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[137] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[138] Abhishek Gupta. *Numerical modelling and optimization of non-isothermal, rigid tool liquid composite moulding processes.* PhD thesis, ResearchSpace@ Auckland, 2013.

[139] CS Tan, A Gupta, YS Ong, SK Lam, M Pratama, and PS Tan. Pareto set representation learning with application to multi-criteria order optimization. In *2024 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0947–0951. IEEE, 2024.

[140] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[141] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, 26, 2013.

[142] Rui Wang, Nan-Jiang Dong, Dun-Wei Gong, Zhong-Bao Zhou, Shi Cheng, Guo-Hua Wu, and Ling Wang. Pca-assisted reproduction for continuous multi-objective optimization with complicated pareto optimal set. *Swarm and Evolutionary Computation*, 60:100795, 2021.

[143] Ye Tian, Chang Lu, Xingyi Zhang, Kay Chen Tan, and Yaochu Jin. Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks. *IEEE transactions on cybernetics*, 51(6):3115–3128, 2020.

[144] Malte Probst and Franz Rothlauf. Harmless overfitting: Using denoising autoencoders in estimation of distribution algorithms. *The Journal of Machine Learning Research*, 21(1):2992–3022, 2020.

[145] Francois Chollet et al. Keras, 2015.

[146] Saeed Saremi and Aapo Hyvarinen. Neural empirical bayes. *arXiv preprint arXiv:1903.02334*, 2019.

[147] Ziming Liu and Max Tegmark. Machine learning conservation laws from trajectories. *Physical Review Letters*, 126(18):180604, 2021.

[148] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE transactions on evolutionary computation*, 13(2):284–302, 2008.

[149] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33-01, pages 1658–1665, 2019.

[150] Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

[151] Jayanta Mandi, Peter J Stuckey, Tias Guns, et al. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34-02, pages 1603–1610, 2020.

[152] Ray Chee Chuan Lim. *Solution representation learning in transfer evolutionary optimization*. PhD thesis, Nanyang Technological University, 2021.

[153] Sinno Jialin Pan, James T Kwok, Qiang Yang, et al. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[154] Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. Decision-focused learning without decision-making: Learning locally optimized decision losses. *Advances in Neural Information Processing Systems*, 35:1320–1332, 2022.

[155] Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *International conference on machine learning*, pages 14935–14947. PMLR, 2022.