AIM: Adaptive Intervention for Deep Multi-task Learning of Molecular Properties

Mason Minot, Gisbert Schneider
Department of Biosystems Science and Engineering
ETH Zürich
Basel, 4056, Switzerland
mminot@ethz.ch, gisbert.schneider@bsse.ethz.ch

Abstract

Simultaneously optimizing multiple, frequently conflicting, molecular properties is a key bottleneck in the development of novel therapeutics. Although a promising approach, the efficacy of multi-task learning is often compromised by destructive gradient interference, especially in the data-scarce regimes common to drug discovery. To address this, we propose AIM, an optimization framework that learns a dynamic policy to mediate gradient conflicts. The policy is trained jointly with the main network using a novel augmented objective composed of dense, differentiable regularizers. This objective guides the policy to produce updates that are geometrically stable and dynamically efficient, prioritizing progress on the most challenging tasks. We demonstrate that AIM achieves statistically significant improvements over multi-task baselines on subsets of the QM9 and targeted protein degraders benchmarks, with its advantage being most pronounced in data-scarce regimes. Beyond performance, AIM's key contribution is its interpretability; the learned policy matrix serves as a diagnostic tool for analyzing inter-task relationships. This combination of data-efficient performance and diagnostic insight highlights the potential of adaptive optimizers to accelerate scientific discovery by creating more robust and insightful models for multi-property molecular design.

1 Introduction

The *in silico* design of molecules frequently requires the simultaneous optimization of multiple, often competing, objectives [1, 2]. For instance, a candidate drug molecule must exhibit high binding affinity to its target while also possessing favorable pharmacokinetic properties such as aqueous solubility and metabolic stability [3]. In domains such as drug discovery, where high-quality experimental data is often scarce and costly to acquire, achieving data efficiency is crucial [4–7]. Multi-task learning (MTL) is a strategy to this end, as it enables a single model to learn a unified representation across related tasks [8].

However, the efficacy of MTL is often compromised by negative transfer [9, 10]. When task objectives conflict, their gradients can pull the model in opposing directions, leading to destructive interference that degrades overall performance. As a result, MTL's potential for data efficiency is often unrealized because the very tasks meant to help one another end up in conflict. To effectively leverage MTL, it is therefore desirable to minimize negative transfer while maximizing synergistic benefits.

A considerable body of work has sought to mitigate this negative interference, laying a crucial foundation for robust multi-task optimization. Methods such as PCGrad [10], CAGrad [9], and Nash-MTL [11] mitigate conflicts using principled, but often static, rules. Although effective, the uniform strategy applied by these hand-crafted heuristics creates an opportunity for

NeurIPS 2025 AI for Science Workshop

a learned, adaptive policy to discover more nuanced strategies for navigating these complex trade-offs.

In this work, we reframe the problem from one of heuristic gradient surgery to one of learned, adaptive optimization. We propose Adaptive Intervention for deep Multi-task learning of molecular properties (AIM), an optimization framework that learns a dynamic, context-aware policy to transform conflicting gradients into a more effective update step. The policy is trained jointly with the main model by optimizing a novel augmented objective guided by a generalization signal from a held-out validation set and is augmented with dense, differentiable regularizers. Key regularizers ensure dynamic efficiency by preserving the update step's magnitude and promoting progress on tasks with the highest current loss.

We validate AIM on a 2D toy problem, the foundational QM9 dataset [12], and a complex targeted protein degrader (TPD) ADME (absorption, distribution, metabolism, and excretion) benchmark [13]. By evaluating on data subsets to simulate real-world, data-scarce scenarios, we show that AIM's learned policy discovers more effective optimization strategies than handcrafted heuristics, particularly in low-data regimes.

2 Related work

This section lays the groundwork for our proposed method. We begin by formally introducing the MTL paradigm and its central challenge of destructive interference. We then review the foundational heuristic approaches, setting the stage for the adaptive learning-based methods we explore. Finally, we introduce the concept of a learned optimization policy as a natural extension of this prior work.

2.1 The challenge of multi-task optimization

Training a single neural network for multiple scientific tasks holds the promise of improving data efficiency and discovering shared representations between physical properties. However, this MTL paradigm introduces a fundamental optimization challenge, namely navigating the trade-offs between competing task objectives.

2.2 Heuristics for mitigating gradient interference

The common formulation of the MTL objective is to minimize a weighted sum of individual task losses $\mathcal{L}(\theta) = \sum_i w_i \mathcal{L}_i(\theta)$. A primary obstacle is destructive gradient interference, where gradients from different tasks $\mathbf{g_i} = \nabla_\theta \mathcal{L}_i$, pull model parameters θ in competing directions. This interference is multifaceted:

1. **Directional conflict:** The literature has traditionally defined directional conflict as an instance where gradients are opposed ($\mathbf{g_i} \cdot \mathbf{g_j} < 0$) [10]. Building on this, our work proposes a more flexible definition, positing that conflict exists across a spectrum of misalignments. Even positively correlated gradients can represent a suboptimal compromise, as their vector sum may neglect the needs of other tasks. The core challenge is thus not simply to avoid opposed gradients, but to learn the optimal degree of alignment for each specific pair of tasks.

2. **Magnitude conflict:** A task with a large gradient norm can dominate the update, hindering progress on other tasks. Although often treated separately, magnitude and directional conflicts are intertwined; an imbalance in magnitude can exacerbate directional disputes. An effective intervention mechanism, therefore, should ideally account for both.

The presence of task conflicts means that a single optimal solution is rarely feasible, leading to a Pareto front of trade-off solutions. The difficulty lies in choosing the point on this front that offers the best generalization. Whereas static heuristics address this with a fixed geometric rule, our work learns this selection process directly from the data, enabling an adaptive strategy.

2.3 Navigating the front: from heuristics to a data-driven approach

A variety of methods have been developed to navigate the multi-task optimization landscape. The simplest approach, linear scalarization (LS), which minimizes the aggregate loss, often fails to converge to the Pareto front. Principled geometric and game-theoretic methods, including PCGrad [10], CAGrad [9], and Nash-MTL [11], apply fixed geometric rules to gradients. Complementary to these, adaptive weighting heuristics like FAMO [14] balance tasks by adjusting their weights based on relative progress. Our work explores a more flexible paradigm in which the intervention policy itself is learned, enabling an adaptive strategy that is sensitive to the unique relationships between tasks.

3 Method: Adaptive Intervention for deep Multi-task learning (AIM)

AIM reframes the resolution of gradient conflicts from a fixed heuristic to a learned, adaptive policy Ψ. This policy transforms a set of raw task gradients $\{\mathbf{g}_i\}$ into a unified update vector $\mathbf{g}_{\text{intervened}}$. This transformed gradient is then used to update the main model parameters, θ, in a standard optimization step. The policy's innovation lies in its data-driven definition of gradient conflict, which is trained by optimizing a novel augmented objective.

3.1 Differentiable gradient intervention

Instead of relying on a static geometric rule (e.g., intervening only when $\mathbf{g}_i \cdot \mathbf{g}_j < 0$), AIM's policy learns when and how strongly to intervene based on a learnable conflict threshold, τ. Crucially, the policy treats the raw task gradients $\{\mathbf{g}_i\}$ as fixed input vectors. Although formal meta-learning [15] or second-order methods could also guide the policy, such approaches can be computationally prohibitive. Instead, by applying a stop-gradient, we prevent the need to differentiate through the gradient computation itself, thus avoiding expensive second-order derivatives while retaining the benefits of a learned, adaptive policy.

The intervention is performed in a pairwise manner. For each task gradient $\mathbf{g}_i$, its modified counterpart $\mathbf{g}'_i$ is computed by considering its relationship with all other task gradients $\mathbf{g}_j$ (where $j \neq i$). The strength of intervention between any pair is determined by a soft, differentiable projection weight $w_{\text{proj}}^{(i,j)}$:

$$w_{\text{proj}}^{(i,j)} = \sigma\left(k \cdot (\tau_{ij} - \cos(\mathbf{g}_i, \mathbf{g}_j))\right) \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, k is a temperature parameter, and τ_{ij} is the learnable threshold for the pair. The modified gradient $\mathbf{g}'_i$ is formed by iteratively removing the conflicting components from other gradients, adapting the formulation from PCGrad [10]:

$$\mathbf{g}'_i = \mathbf{g}_i - \sum_{j \neq i} w_{\text{proj}}^{(i,j)} \cdot \text{proj}_{\mathbf{g}_j}(\mathbf{g}_i) \tag{2}$$

where $\text{proj}_{\mathbf{g}_j}(\mathbf{g}_i) = (\frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2})\mathbf{g}_j$ is the vector projection of $\mathbf{g}_i$ on $\mathbf{g}_j$. These modified gradients, $\{\mathbf{g}'_i\}$, are then summed to form the final update vector, $\mathbf{g}_{\text{intervened}} = \sum_{i=1}^{N} \mathbf{g}'_i$, which is used to update the model parameters: $\theta_{t+1} \leftarrow \text{Adam}(\theta_t, \mathbf{g}_{\text{intervened}})$. The complete process is detailed in Algorithm 1 in the Appendix.

We explore two policy variants:

1. **Scalar policy:** Learns a single global threshold τ, applying a uniform conflict tolerance across all pairs of tasks.
2. **Matrix policy:** Learns a unique threshold τ_{ij} for each pair of tasks, allowing the model to capture nuanced pairwise relationships and providing an interpretable diagnostic tool.

3.2 Guiding the policy

The policy parameters, Φ, are trained jointly with the main model by minimizing an augmented objective, $\mathcal{L}_{\text{policy}}$. To guide the policy with a generalization signal, the training data is partitioned once before training into two disjoint datasets: a primary set for calculating task gradients and a policy guidance set (10% of the training data) for evaluating the policy. At each training step, a batch

from the primary set is used to generate the raw gradients, while a separate batch from the policy guidance set is used to update the policy's parameters. This use of a held-out dataset is crucial, as it prevents the policy from learning a trivial solution (e.g., outputting a zero vector) by forcing it to produce updates that generalize to unseen data.

While a forward-looking objective could evaluate the policy based on a simulated "lookahead" step, we found this approach to be computationally expensive. Instead, we opt for a more efficient joint-training scheme where the policy guidance loss, $\mathcal{L}_{\text{guide}}$, is calculated on the current model parameters. This introduces a slight temporal mismatch, but we found it to be a favorable trade-off for the significant gains in computational efficiency. The objective is thus guided by the model's loss on the held-out data and regularizers that promote dynamic efficiency, with each component weighted by a scalar hyperparameter λ:

$$\mathcal{L}_{\text{policy}} = \lambda_g \mathcal{L}_{\text{guide}} + \lambda_m \mathcal{L}_{\text{magnitude}} + \lambda_p \mathcal{L}_{\text{progress}} \tag{3}$$

The components of this objective are:

- $\mathcal{L}_{\textbf{guide}}$ **(Policy guidance objective):** The model loss on a held-out subset of the training data, providing a direct signal of generalization performance.
- $\mathcal{L}_{\textbf{magnitude}}$ **(Magnitude preservation):** A stability penalty, formulated as $(\|\mathbf{g}_{\text{intervened}}\| - \sum_i \|\mathbf{g}_i\|)^2$, that encourages the policy to redistribute, not destroy, gradient energy.
- $\mathcal{L}_{\textbf{progress}}$ **(Progress penalty):** A penalty that rewards interventions for aiding tasks with high current loss. It is formulated as $-\sum_i \alpha_i \cdot p_i$, where the weight $\alpha_i = \mathcal{L}_i / \sum_j \mathcal{L}_j$ is the normalized loss of task i, and $p_i = \mathbf{g}_{\text{intervened}} \cdot \mathbf{g}_i$ is the projected progress. This encourages the policy to prioritize helping the most difficult tasks at each step.

By learning an intervention policy guided by this multi-faceted objective, AIM develops a nuanced, context-aware strategy that overcomes certain limitations of handcrafted heuristics.

4 Experiments and results

To evaluate our proposed optimization framework, we compare its performance against several established MTL baselines. To evaluate the effectiveness of the model-agnostic AIM optimizer, we use a consistent graph neural network (GNN) architecture for all methods to ensure a fair comparison, adopting the standard message-passing model used previously [14]. All performance metrics reported are the mean and standard deviation computed over $N = 3$ independent runs with different random seeds. For all experiments on a given benchmark, the test set was held fixed to ensure a fair comparison across different training subset sizes.

4.1 Visualizing learning trajectories on a toy problem

To visualize how different algorithms navigate a conflicted optimization landscape, we use a two-task problem with a known Pareto front, as shown in Figure 1. We compare the optimization trajectories of LS, the heuristic-based solver PCGrad, and our proposed method, AIM (Matrix). The results highlight a key difference in behavior: LS fails to converge to the Pareto front, whereas both PCGrad and AIM successfully navigate the landscape to find an optimal trade-off solution. This demonstrates that AIM can effectively resolve gradient conflicts, matching the performance of established methods in this idealized setting.

4.2 QM9 benchmark

The QM9 dataset [12] serves as a foundational benchmark in computational chemistry, comprising approximately 134,000 small organic molecules and their corresponding quantum mechanical properties, calculated using Density Functional Theory (DFT). These properties, ranging from electronic characteristics such as ϵ_{HOMO} and ϵ_{LUMO} to energetic values such as the internal energy at 298.15 K (U), provide a diverse and challenging set of prediction targets.

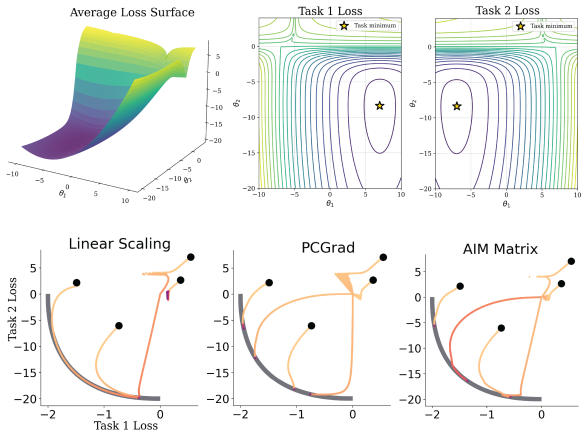

Figure 1: (Top) Loss landscape of the 2D toy problem, showing conflicting task minima (stars) and the resulting Pareto front. (Bottom) Optimization trajectories for linear scalarization (LS), PCGrad, and AIM on the conflicted landscape.

4.2.1 The learned policy as a diagnostic tool

A key advantage of AIM is that its learned policy matrix offers a diagnostic window into the MTL process. Figure 2 visualizes the policy's evolution, showing the learned conflict thresholds, the rates of conflict, raw gradient similarities, and resulting projection weights early (Epoch 10) and later (Epoch 100) in training. The policy learns an intuitive optimization strategy based on the alignment of task gradients, guided by its objective to aid struggling tasks. This is evident with the highly aligned energetic properties (U_0, U, H, G), where the policy applies near-zero projection weights. In contrast, as training progresses, stricter interventions for pairs like C_v and ϵ_{LUMO} drive a more independent optimization. This evolution demonstrates AIM's ability to adapt to nuanced task relationships, a crucial capability for navigating complex optimization landscapes.

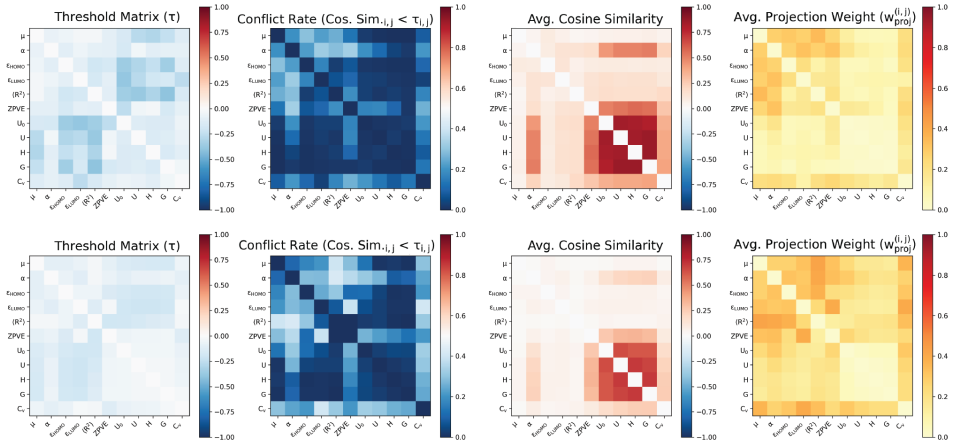

Figure 2: **Evolution of the learned AIM policy on QM9.** The heatmaps show four diagnostic metrics for the policy early in training (Epoch 10, top) and later in training (Epoch 100, bottom). From left to right: learned conflict thresholds (τ), conflict rate, raw gradient cosine similarity, and final projection weights.

4.2.2 Quantitative performance

As a strong performance benchmark, we use a single-task learning (STL) baseline, which consists of a separate model trained independently for each task on the full dataset. We evaluate our multi-task learning (MTL) approach against this baseline using Mean Rank (MR) across all tasks and $\Delta_m\%$, the mean percentage error improvement relative to STL. For both metrics, lower values indicate better performance.

The results in Table 1 highlight the strength of AIM in low data settings. In the 10k subset, both AIM (Scalar) and AIM (Matrix) significantly outperform all baselines on both MR and $\Delta_m\%$. This trend continues on the 50k subset, where AIM (Scalar) achieves the best performance on both metrics, indicating a robust and balanced performance across the diverse quantum chemical properties. As the dataset size increases to 100k, other methods become more competitive. While FAMO achieves the best $\Delta_m\%$, AIM (Scalar) secures the top Mean Rank, demonstrating that the method remains a strong performer. This overall pattern suggests that AIM's learned policy is particularly effective when data is limited. For a granular view, the MAE per task on the 100k subset is detailed in the Appendix.

Table 1: Benchmark results on the QM9 dataset for different training subset sizes. We report the test delta ($\Delta_m\%$) and the mean rank (MR) across all tasks. Lower values are better.

	10k		50k		100k	
Model	$\Delta_m\%$	MR	$\Delta_m\%$	MR	$\Delta_m\%$	MR
LS	588.6 ± 50.3	10.82	276.1 ± 23.9	6.36	163.6 ± 9.6	5.00
PCGrad	512.5 ± 44.2	9.18	203.0 ± 10.7	3.73	132.5 ± 1.8	4.18
NASH-MTL	608.8 ± 24.1	12.18	273.0 ± 25.9	5.45	185.4 ± 13.8	6.18
FAMO	559.0 ± 68.5	7.09	205.9 ± 31.3	3.55	$\mathbf{100.6 \pm 12.0}$	3.55
CAGrad	514.6 ± 23.3	9.18	188.6 ± 16.2	4.09	116.2 ± 11.5	4.09
AIM (Scalar)	478.9 ± 23.1	8.27	**186.4 ± 12.4**	**2.36**	120.2 ± 0.7	**2.27**
AIM (Matrix)	**436.6 ± 25.6**	**4.45**	194.5 ± 8.6	2.45	124.7 ± 2.1	2.73

4.3 Targeted protein degraders (TPD) ADME benchmark

We test our method on a frontier drug discovery challenge: predicting ADME properties for TPDs and other molecules using a recent benchmark [13]. The dataset's 274,00 molecules and 25 diverse tasks provide a complex test-bed for MTL.

Our evaluation on the TPD ADME dataset demonstrates the effectiveness of our AIM framework. We report the $\Delta_m\%$ over the single-task learning (STL) baseline, which was trained on the full dataset for each respective task, and the MR across all 25 tasks for training data subsets of 10k, 100k and 250k compounds. Similar to the QM9 benchmark, the evolution of the learned policy on the TPD dataset provides a diagnostic view of the inter-task relationships over time (Figure 3). Initially, it applies moderate interventions to navigate gradient conflicts and aid tasks like the CYP inhibition assays. This initial corrective action successfully steers the optimization into a more cooperative state later in training, characterized by higher gradient alignment that sharpens and refines the inherent clusters among functionally related tasks.

Table 2: Benchmark results on the TPD ADME dataset. We report the test delta ($\Delta_m\%$) and the mean rank (MR) across all tasks. Lower values are better.

	10k		100k		250k	
Model	$\Delta_m\%$	MR	$\Delta_m\%$	MR	$\Delta_m\%$	MR
LS	117.5 ± 15.3	5.40	34.7 ± 6.7	5.68	19.4 ± 7.5	4.40
PCGrad	310.7 ± 170.3	6.96	66.1 ± 22.1	6.92	29.2 ± 10.5	4.50
NASH-MTL	116.3 ± 4.1	5.04	30.5 ± 5.3	3.52	19.0 ± 5.8	3.76
FAMO	110.6 ± 4.5	4.12	30.5 ± 3.7	3.76	18.6 ± 4.1	3.80
CAGrad	105.0 ± 9.3	3.28	28.2 ± 1.3	3.12	$\mathbf{15.0 \pm 1.7}$	**2.56**
AIM (Scalar)	90.5 ± 1.4	1.92	$\mathbf{28.0 \pm 1.1}$	2.52	19.4 ± 3.3	3.84
AIM (Matrix)	$\mathbf{88.6 \pm 3.3}$	**1.28**	28.1 ± 3.4	**2.48**	16.4 ± 1.2	2.64

AIM delivers state-of-the-art performance, particularly in the more data-scarce regime (Table 2). In the 10k and 100k subsets, the AIM methods outperform all baselines. In the 100k subset, for example, the AIM (Matrix) and AIM (Scalar) variants achieve the best and second-best MR (2.48 and 2.52, respectively) and the second-best and best $\Delta_m\%$ (28.1 and 28.0, respectively). As the dataset size

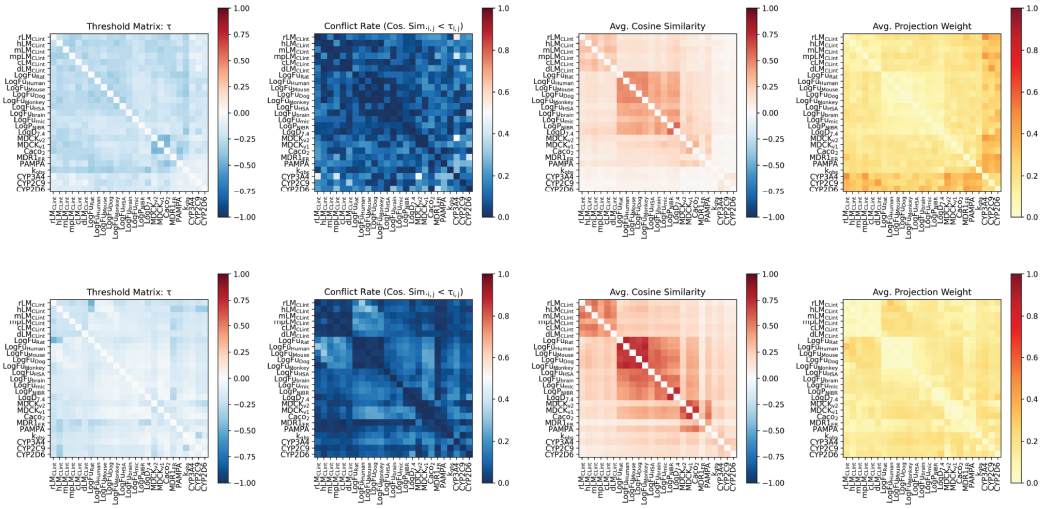

Figure 3: **Evolution of the learned AIM policy on TPD.** The heatmaps show the policy's diagnostic metrics early (Epoch 10, top) and late (Epoch 300, bottom) in training. From left to right: learned conflict thresholds (τ), conflict rate, raw gradient similarity, and final projection weights.

increases to 250k, AIM remains competitive. AIM (Matrix) achieves the second-best performance in both $\Delta_m\%$ (16.4) and Mean Rank (2.64), closely following the top-performing CAGrad. This highlights AIM's ability to find a robust and well-balanced solution across a wide array of competing objectives.

The detailed results per task on the 100k subset are provided in the Appendix. Together, these results demonstrate AIM's strong performance in specific molecular properties related to absorption, distribution, and metabolism.

5 Discussion

Our work introduced AIM, an optimization framework that learns an adaptive policy to mediate gradient conflicts in multi-task learning. The experimental results across a toy problem, the QM9 benchmark, and a frontier drug discovery problem, given by the TPD ADME dataset, confirm our central hypothesis: A learned, data-driven intervention policy can outperform static heuristics, particularly in the data-scarce regimes common to early phases of scientific discovery. Importantly, the learned policy itself serves as an interpretable diagnostic tool that offers insights into the relationships between tasks. This discussion synthesizes these findings, addresses the performance nuances, and frames the practical utility of AIM in light of its demonstrated strengths and limitations.

5.1 The value and trade-offs of an adaptive policy

A key finding of this work is that the value of AIM's adaptive policy is most pronounced when data is limited. In both the QM9 and TPD ADME benchmarks, AIM's advantage over heuristic-based methods like CAGrad and FAMO was greatest on the smaller data subsets. This performance trend leads us to hypothesize that the primary value of an adaptive policy is in discovering a more data-efficient optimization trajectory. In data-limited settings, the fixed geometric rules of heuristics may be too rigid, whereas AIM's learned policy appears to find a more effective path for navigating gradient conflicts. As the amount of training data increases, this advantage narrows, and simpler heuristics become more competitive. We therefore conclude that AIM's core strength provides the most value when data, not compute, is the primary bottleneck. This complexity-versus-robustness trade-off is also observed within AIM itself. The simpler AIM (Scalar) variant sometimes outperforms the more granular Matrix policy, a result we attribute to the scalar policy's reduced risk of overfitting to the training data, making it a more robust, albeit less nuanced, choice in certain scenarios.

5.2 From black box to diagnostic: interpreting the learned policy

Beyond pure performance, AIM's primary contribution is its interpretability. The evolution of the policy matrix validates the learning process, showing how AIM learns a sophisticated optimization strategy that adapts to the natural alignment of task gradients. On QM9 (Figure 2), for instance, the policy preserves beneficial gradient sharing between highly-aligned energetic properties (U_0, U, H, G) with minimal intervention, while enforcing independent optimization paths for non-aligned tasks through stricter interventions. This diagnostic capability extends to the more complex TPD benchmark (Figure 3), where the policy's evolution reveals a progression from applying moderate interventions early in training to fostering a more cooperative state characterized by refined task clusters and higher gradient alignment. The ability to learn and visualize these nuanced strategies, which align with established physical and ADME principles, builds confidence in the model and its predictions.

5.3 The practical utility of multi-task learning in scientific discovery

A crucial observation is that the single-task learning (STL) baseline sets a formidable performance benchmark. It is important to highlight that the STL baseline was trained on the full dataset for each task, whereas the MTL models had access to only a fraction of the data. Although this data disparity explains much of the performance gap in low-data regimes, a gap persists even when MTL models are trained on larger subsets, particularly for QM9. This points to a fundamental trade-off: an STL model is a specialist, dedicating its entire capacity to the nuances of a single task. An MTL model, in contrast, must find a single, generalized representation that serves as a compromise across multiple, often conflicting, objectives. As a result, for some datasets, including QM9, it has become common practice in the literature to train a single model per task.

However, this idealized STL scenario is often impractical. In scientific discovery, particularly in fields like drug development, the simultaneous optimization of many properties is a hard constraint. Training and deploying hundreds of specialist models is logistically and computationally infeasible, making multi-task learning (MTL) the only viable path forward for creating a single, unified, and operationally efficient model. The fundamental challenge of MTL, therefore, is to effectively manage these inherent trade-offs. Our results demonstrate that while simple MTL methods struggle with this compromise, AIM's adaptive policy consistently pushes the performance frontier, outperforming other MTL heuristics in the data-scarce regimes most relevant to these challenges. AIM's contribution is therefore twofold. First, it offers superior performance within this necessary, yet inherently compromised, MTL paradigm. Second, and critically, its interpretable policy (as seen in Figures 2 and 3) provides a diagnostic tool to understand the very task conflicts that necessitate this compromise in the first place. AIM offers not just a better answer, but deeper insight into the complex inter-task relationships that govern multi-property molecular design.

5.4 Outlook and future directions

The insights gained from this analysis directly inform the most promising future applications of AIM. The demonstrated ability of the policy to identify task groups (Figure 2, 3) provides a data-driven foundation for automated curriculum design. An early stage policy could be used to pre-train a model on synergistic task groups before introducing more conflicting objectives, a strategy that is more principled than manual curriculum design.

Similarly, the policy's function as a scientific hypothesis engine is grounded in its proven ability to learn known physical relationships. Its potential to uncover unexpected task compatibilities, by learning a permissive policy between two seemingly unrelated tasks, is therefore not mere speculation. It represents a concrete mechanism for generating machine-driven hypotheses that can guide targeted experimental validation, turning the optimizer into an active participant in the scientific discovery process.

We also acknowledge several limitations that provide clear avenues for future research. A thorough ablation study is needed to isolate the specific contributions of the guidance loss and the regularizers within our augmented objective. Furthermore, a detailed analysis of the computational overhead introduced by the policy and its guidance set would be crucial for practical deployment. Finally, evaluating AIM's generalizability across diverse model architectures beyond the GNN used here will be essential to validate its broader utility as a model-agnostic optimizer.

Acknowledgments and Disclosure of Funding

We thank Andrea Anelli, Leonid Komissarov, Natasa Tagasovska, Pan Kessel, and Kenneth Atz for inspiring discussions. This study was financially supported by the Roche Access to Distinguished Scientists (ROADS) Program (grant no. ROADS-073 to G.S.).

References

[1] Petra Schneider, W Patrick Walters, Alleyn T Plowright, Norman Sieroka, Jennifer Listgarten, Robert A Goodnow Jr, Jasmin Fisher, Johanna M Jansen, José S Duca, Thomas S Rush, et al. Rethinking drug design in the artificial intelligence era. *Nature reviews drug discovery*, 19(5): 353–364, 2020.

[2] Kenneth Atz, David F Nippa, Alex T Müller, Vera Jost, Andrea Anelli, Michael Reutlinger, Christian Kramer, Rainer E Martin, Uwe Grether, Gisbert Schneider, et al. Geometric deep learning-guided suzuki reaction conditions assessment for applications in medicinal chemistry. *RSC Medicinal Chemistry*, 15(7):2310–2321, 2024.

[3] David F Nippa, Kenneth Atz, Yannick Stenzhorn, Alex T Müller, Andreas Tosstorff, Jörg Benz, Hayley Binch, Markus Bürkler, Achi Haider, Dominik Heer, et al. Expediting hit-to-lead progression in drug discovery through reaction prediction and multi-objective molecular optimization. 2025.

[4] Andrea Anelli, Hanno Dietrich, Philipp Ectors, Frank Stowasser, Tristan Bereau, Marcus Neumann, and Joost van den Ende. Robust and efficient reranking in crystal structure prediction: a data driven method for real-life molecules. *CrystEngComm*, 26(41):5845–5849, 2024.

[5] Mason Minot and Sai T Reddy. Meta learning addresses noisy and under-labeled data in machine learning-guided antibody engineering. *Cell systems*, 15(1):4–18, 2024.

[6] Leonid Komissarov, Nenad Manevski, Katrin Groebke Zbinden, Torsten Schindler, Marinka Zitnik, and Lisa Sach-Peltason. Actionable predictions of human pharmacokinetics at the drug design stage. *Molecular Pharmaceutics*, 21(9):4356–4371, 2024.

[7] Mason Minot, Yannick Stenzhorn, Jens Wolfard, Sebastian Strobel, Philippe Jablonski, Daniel Zimmerli, Martin Binder, Uwe Grether, Rainer E Martin, Alex T Müller, et al. Active and geometric deep learning advances chemical reaction prediction in data-scarce drug discovery. 2025.

[8] Stephan Allenspach, Jan A Hiss, and Gisbert Schneider. Neural multi-task learning in drug design. *Nature Machine Intelligence*, 6(2):124–137, 2024.

[9] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in Neural Information Processing Systems*, 34:18878–18890, 2021.

[10] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[11] Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

[12] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

[13] Giulia Peteani, Minh Tam Davide Huynh, Grégori Gerebtzoff, and Raquel Rodríguez-Pérez. Application of machine learning models for property prediction to targeted protein degraders. *Nature communications*, 15(1):5764, 2024.

[14] Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. Famo: Fast adaptive multitask optimization. *Advances in Neural Information Processing Systems*, 36:57226–57243, 2023.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

A Appendix

A.1 Algorithm

The AIM update step is summarized in Algorithm 1.

Algorithm 1 AIM Update Step

Require: Main model parameters θ, Policy parameters Φ (containing τ or τ_{ij})
Require: Learning rates η_θ, η_Φ
Require: Hyperparameters $\lambda_g, \lambda_m, \lambda_p$
Require: Batch from primary training set D_{primary}, Batch from policy guidance set D_{guide}

// Step 1: Compute Raw Task Gradients on Primary Data
for each task $i \in \{1, \ldots, N\}$ **do**
  Compute task loss $\mathcal{L}_i(\theta)$ on D_{primary}
  Compute raw task gradient $\mathbf{g}_i = \nabla_\theta \mathcal{L}_i(\theta)$ ▷ Stop gradient applied to $\mathbf{g}_i$ for policy input
end for
// Step 2: Differentiable Gradient Intervention (Policy Ψ)
Initialize $\mathbf{g}_{\text{intervened}} = \mathbf{0}$
for each task $i \in \{1, \ldots, N\}$ **do**
  Initialize $\mathbf{g}'_i = \mathbf{g}_i$
  for each conflicting task $j \in \{1, \ldots, N\}, j \neq i$ **do**
    Compute cosine similarity: $\cos(\mathbf{g}_i, \mathbf{g}_j) = \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_i\| \|\mathbf{g}_j\|}$
    Retrieve learnable threshold τ_{ij} (or global τ for scalar policy)
    Compute projection weight: $w_{\text{proj}}^{(i,j)} = \sigma\left(k \cdot (\tau_{ij} - \cos(\mathbf{g}_i, \mathbf{g}_j))\right)$
    Compute vector projection: $\text{proj}_{\mathbf{g}_j}(\mathbf{g}_i) = \left(\frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2}\right) \mathbf{g}_j$
    Modify gradient: $\mathbf{g}'_i \leftarrow \mathbf{g}'_i - w_{\text{proj}}^{(i,j)} \cdot \text{proj}_{\mathbf{g}_j}(\mathbf{g}_i)$
  end for
  $\mathbf{g}_{\text{intervened}} \leftarrow \mathbf{g}_{\text{intervened}} + \mathbf{g}'_i$
end for
// Step 3: Compute Policy Loss on Guidance Data
Compute task losses $\mathcal{L}_1(\theta), \ldots, \mathcal{L}_N(\theta)$ on D_{guide}
// Policy Guidance Objective
$\mathcal{L}_{\text{guide}} = \sum_{i=1}^{N} \mathcal{L}_i(\theta)$
// Magnitude Preservation
$\mathcal{L}_{\text{magnitude}} = \left(\|\mathbf{g}_{\text{intervened}}\| - \sum_{i=1}^{N} \|\mathbf{g}_i\|\right)^2$
// Progress Penalty
Initialize $\mathcal{L}_{\text{progress}} = 0$
for each task $i \in \{1, \ldots, N\}$ **do**
  $\alpha_i = \mathcal{L}_i(\theta) / \sum_{j=1}^{N} \mathcal{L}_j(\theta)$
  $p_i = \mathbf{g}_{\text{intervened}} \cdot \mathbf{g}_i$
  $\mathcal{L}_{\text{progress}} \leftarrow \mathcal{L}_{\text{progress}} - \alpha_i \cdot p_i$
end for
Total Policy Loss: $\mathcal{L}_{\text{policy}} = \lambda_g \mathcal{L}_{\text{guide}} + \lambda_m \mathcal{L}_{\text{magnitude}} + \lambda_p \mathcal{L}_{\text{progress}}$
// Step 4: Update Parameters
Update policy parameters: $\Phi \leftarrow \text{Optimizer}(\Phi, \nabla_\Phi \mathcal{L}_{\text{policy}}, \eta_\Phi)$ ▷ e.g., Adam or SGD
Update main model parameters: $\theta \leftarrow \text{Optimizer}(\theta, \mathbf{g}_{\text{intervened}}, \eta_\theta)$ ▷ e.g., Adam or SGD

A.2 Training details and hyperparameters

All models were trained on the ETH Zürich Euler HPC using a single NVIDIA A100 GPU. The Adam optimizer was used for all experiments. The learning rate was managed by a ReduceLROnPlateau scheduler, which reduced the learning rate when the mean validation loss across all tasks plateaued. For the AIM policy optimizer, we used a CosineAnnealingWarmRestarts scheduler with the following parameters: $T_0 = 20$ epochs, $T_{mult} = 1$, and $\eta_{min} = 10^{-6}$.

Specific hyperparameters for the main model and the AIM policy were tuned for each dataset and are detailed in Table 3.

Table 3: Hyperparameter settings.

Hyperparameter	**Value**
General Training	
Epochs	[300, 400]
Early Stopping Patience	75 epochs
Main Model Learn Rate	$[1 \times 10^{-3}, 5 \times 10^{-4}]$
AIM Policy Specific	
Policy Learn Rate	$[5 \times 10^{-4}, 1 \times 10^{-3}, 5 \times 10^{-3}]$
k (temperature)	10
λ_g ($\mathcal{L}_{guide}$)	[0, 1]
λ_m (magnitude preservation)	[0, 0.01]
λ_p (progress penalty)	[0, 0.08, 0.8]

A.3 Additional extended results

Table 4: Mean Absolute Error (MAE) on the QM9 test set. Models were trained on the 100k subset. Lower values are better.

Method	μ	α	ϵ_{HOMO}	ϵ_{LUMO}	R^2	ZPVE	U_0	U	H	G	C_v
STL	0.067	0.181	60.58	53.91	0.503	4.54	58.84	64.24	63.85	66.22	0.072
LS	0.10	0.31	70.98	86.65	4.84	12.94	137.3	138.2	138.7	133.6	0.12
PCGrad	0.11	0.31	78.72	91.31	3.98	8.97	121.1	121.8	122.0	119.0	0.11
NASH-MTL	0.11	0.33	77.46	93.78	5.23	14.39	148.4	149.2	149.5	144.6	0.13
FAMO	0.39	0.37	147.1	134.7	**1.16**	**4.83**	**68.65**	**69.14**	**69.08**	**68.42**	0.12
CAGrad	0.11	0.32	79.82	94.89	3.34	7.34	116.2	116.8	117.1	114.1	0.12
AIM (Scalar)	0.089	0.268	**59.37**	**71.29**	4.18	9.82	111.5	111.6	112.0	113.2	**0.103**
AIM (Matrix)	**0.088**	**0.251**	60.94	72.31	4.07	12.16	116.3	116.4	118.2	111.7	**0.103**

Table 5: Mean Absolute Error (MAE) for clearance tasks on the TPD ADME 100k subset. Lower values are better.

Method	rLM	hLM	mLM	pigLM	cyno	dLM
STL	0.098	0.105	0.101	0.077	0.096	0.088
CAGrad	0.123 ± 0.000	0.128 ± 0.002	0.124 ± 0.001	0.103 ± 0.002	0.118 ± 0.001	0.121 ± 0.001
Nash-MTL	0.123 ± 0.004	0.129 ± 0.004	0.124 ± 0.004	0.103 ± 0.004	0.119 ± 0.005	0.121 ± 0.004
FAMO	0.124 ± 0.002	0.133 ± 0.002	0.127 ± 0.002	0.104 ± 0.002	0.121 ± 0.003	0.123 ± 0.002
LS	0.126 ± 0.003	0.133 ± 0.005	0.127 ± 0.005	0.107 ± 0.004	0.122 ± 0.004	0.125 ± 0.005
PCGrad	0.137 ± 0.008	0.151 ± 0.010	0.136 ± 0.003	0.120 ± 0.009	0.129 ± 0.003	0.138 ± 0.006
AIM (Scalar)	0.119 ± 0.003	0.125 ± 0.006	$\mathbf{0.119 \pm 0.001}$	0.101 ± 0.002	0.116 ± 0.004	$\mathbf{0.118 \pm 0.001}$
AIM (Matrix)	$\mathbf{0.117 \pm 0.003}$	$\mathbf{0.123 \pm 0.004}$	0.120 ± 0.003	$\mathbf{0.100 \pm 0.003}$	$\mathbf{0.115 \pm 0.004}$	0.119 ± 0.004

Table 6: Mean Absolute Error (MAE) for distribution and binding tasks (Part 1) on the TPD ADME 100k subset. Lower values are better.

Method	Fu-Rat	Fu-Hum	Fu-Mou	Fu-Dog	Fu-Mon
STL	0.083	0.083	0.080	0.081	0.081
LS	0.103 ± 0.005	0.102 ± 0.004	0.101 ± 0.005	0.101 ± 0.005	0.098 ± 0.005
PCGrad	0.113 ± 0.005	0.168 ± 0.064	0.148 ± 0.033	0.115 ± 0.008	0.148 ± 0.053
Nash-MTL	0.100 ± 0.003	0.099 ± 0.003	0.098 ± 0.003	0.097 ± 0.003	0.095 ± 0.004
FAMO	$\mathbf{0.091 \pm 0.002}$	$\mathbf{0.090 \pm 0.002}$	$\mathbf{0.089 \pm 0.002}$	$\mathbf{0.089 \pm 0.002}$	$\mathbf{0.086 \pm 0.002}$
CAGrad	0.102 ± 0.001	0.101 ± 0.001	0.100 ± 0.001	0.100 ± 0.001	0.097 ± 0.001
AIM (Scalar)	0.097 ± 0.001	0.095 ± 0.001	0.095 ± 0.002	0.094 ± 0.001	0.092 ± 0.001
AIM (Matrix)	0.098 ± 0.002	0.096 ± 0.002	0.095 ± 0.002	0.095 ± 0.003	0.093 ± 0.002

Table 7: Mean Absolute Error (MAE) for distribution and binding tasks (Part 2) on the TPD ADME 100k subset. Lower values are better.

Method	Fu-HSA	Fu-brain	Fu-mic	LogP	LogD
STL	0.060	0.074	0.062	0.112	0.124
LS	0.088 ± 0.004	0.099 ± 0.004	0.083 ± 0.004	0.143 ± 0.006	0.156 ± 0.007
PCGrad	0.105 ± 0.005	0.110 ± 0.006	0.124 ± 0.041	0.177 ± 0.011	0.217 ± 0.057
Nash-MTL	0.086 ± 0.003	0.096 ± 0.004	0.081 ± 0.002	$\mathbf{0.137 \pm 0.005}$	$\mathbf{0.150 \pm 0.005}$
FAMO	$\mathbf{0.077 \pm 0.002}$	$\mathbf{0.089 \pm 0.002}$	$\mathbf{0.072 \pm 0.002}$	0.146 ± 0.003	0.161 ± 0.003
CAGrad	0.084 ± 0.000	0.097 ± 0.000	0.080 ± 0.000	0.140 ± 0.002	0.153 ± 0.002
AIM (Scalar)	0.082 ± 0.001	0.094 ± 0.001	0.077 ± 0.001	0.140 ± 0.002	0.152 ± 0.001
AIM (Matrix)	0.083 ± 0.000	0.094 ± 0.003	0.077 ± 0.002	0.139 ± 0.004	0.154 ± 0.005

Table 8: Mean Absolute Error (MAE) for permeability and metabolism tasks (Part 1) on the TPD ADME 100k subset. Lower values are better.

Method	MDCKv2	MDCKv1	Caco2	MDR1	PAMPA
STL	0.097	0.100	0.116	0.082	0.123
LS	0.123 ± 0.005	0.129 ± 0.004	0.153 ± 0.006	0.127 ± 0.007	0.155 ± 0.007
PCGrad	0.140 ± 0.006	0.193 ± 0.066	0.206 ± 0.005	0.142 ± 0.005	0.180 ± 0.010
Nash-MTL	$\mathbf{0.120 \pm 0.003}$	$\mathbf{0.125 \pm 0.003}$	$\mathbf{0.148 \pm 0.004}$	0.124 ± 0.005	$\mathbf{0.149 \pm 0.005}$
FAMO	0.130 ± 0.003	0.140 ± 0.004	0.171 ± 0.005	0.135 ± 0.005	0.175 ± 0.006
CAGrad	$\mathbf{0.120 \pm 0.001}$	0.126 ± 0.001	0.149 ± 0.001	$\mathbf{0.123 \pm 0.001}$	0.151 ± 0.002
AIM (Scalar)	0.121 ± 0.000	0.128 ± 0.002	0.153 ± 0.002	$\mathbf{0.123 \pm 0.001}$	0.154 ± 0.003
AIM (Matrix)	$\mathbf{0.120 \pm 0.003}$	0.126 ± 0.002	0.151 ± 0.003	$\mathbf{0.123 \pm 0.002}$	0.151 ± 0.003

Table 9: Mean Absolute Error (MAE) for permeability and metabolism tasks (Part 2) on the TPD ADME 100k subset. Lower values are better.

Method	kobs	CYP3A4	CYP2C9	CYP2D6
STL	0.063	0.088	0.064	0.069
LS	0.108 ± 0.003	0.125 ± 0.008	0.095 ± 0.003	0.109 ± 0.003
PCGrad	0.104 ± 0.003	0.133 ± 0.006	0.172 ± 0.065	0.110 ± 0.006
Nash-MTL	0.107 ± 0.004	0.119 ± 0.006	0.093 ± 0.003	0.103 ± 0.002
FAMO	0.103 ± 0.002	0.119 ± 0.003	0.089 ± 0.001	0.100 ± 0.003
CAGrad	$\mathbf{0.091 \pm 0.002}$	$\mathbf{0.107 \pm 0.001}$	$\mathbf{0.080 \pm 0.002}$	$\mathbf{0.094 \pm 0.001}$
AIM (Scalar)	0.101 ± 0.000	0.115 ± 0.004	0.089 ± 0.000	0.102 ± 0.003
AIM (Matrix)	0.101 ± 0.001	0.114 ± 0.001	0.090 ± 0.001	0.106 ± 0.007