ParetoFlow: Guided Flows in Multi-Objective Optimization

Ye Yuan [*1, 2], Can (Sam) Chen [*†1, 2], Christopher Pal [‡2, 3, 4], Xue Liu [‡1, 2]

McGill University [1] Mila [2] Polytechnique Montréal [3] CIFAR [4]

[*]Equal contribution with random order; [†]Tech Lead: can.chen@mila.quebec; [‡] Equal senior contribution with random order.

Problem Background

- Design objects with specific desired properties.
    - For example: design a neural network architecture to minimize the test loss on a classification task.

- Previous research primarily focuses on single-objective optimization, which fails to capture real-world complexities:
    - For example: design a neural network architecture that demands both low loss and minimal parameter counts.

- **Offline multi-objective optimization (MOO):** leveraging an offline dataset of designs and their associated labels to minimize multiple objectives simultaneously.

Problem Formulation

- Find $\boldsymbol{x}^* \in \mathcal{X}$ such that there is no $\boldsymbol{x} \in \mathcal{X}$ with $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$, where $\boldsymbol{f} : \mathcal{X} \rightarrow \mathbb{R}^m$ is a vector of m objective functions, and $\prec$ denotes Pareto dominance.

- A solution $\boldsymbol{x}$ is said to *Pareto dominate* another solution $\boldsymbol{x}^*$ (denoted as $\boldsymbol{f}(\boldsymbol{x}) \prec \boldsymbol{f}(\boldsymbol{x}^*)$) if:

$$\forall i \in \{1, \ldots, m\}, \quad f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$$
$$\text{and} \quad \exists j \in \{1, \ldots, m\} \text{ such that } f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*).$$

- A solution $\boldsymbol{x}^*$ is Pareto optimal if there is no other solution $\boldsymbol{x} \in \mathcal{X}$ that Pareto dominates $\boldsymbol{x}^*$. The set of all Pareto optimal solutions constitutes the *Pareto set (PS)*. The corresponding set of objective vectors, defined as $\{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in PS\}$, is known as the *Pareto front*.

Flow Matching

- A conditional probability $p_t(\boldsymbol{x} \mid \boldsymbol{x}_1),\ t \in [0,1]$, evolving from an initial distribution $p_0(\boldsymbol{x} \mid \boldsymbol{x}_1) = q(\boldsymbol{x})$ to an approximate Dirac delta function $p_1(\boldsymbol{x} \mid \boldsymbol{x}_1) \approx \delta(\boldsymbol{x} - \boldsymbol{x}_1)$. This evolution is conditioned on a specific point $\boldsymbol{x}_1$ from the distribution p_{data} and is driven by the conditional vector field $u_t(\boldsymbol{x} \mid \boldsymbol{x}_1)$.

- The process begins by drawing initial noise $\boldsymbol{x_0}$ from $q(\boldsymbol{x_0})$. This noise is then linearly interpolated with the data point $\boldsymbol{x}_1$:

$$\boldsymbol{x} \mid \boldsymbol{x}_1, t = (1-t) \cdot \boldsymbol{x_0} + t \cdot \boldsymbol{x_1}, \quad \boldsymbol{x_0} \sim q(\boldsymbol{x_0}).$$

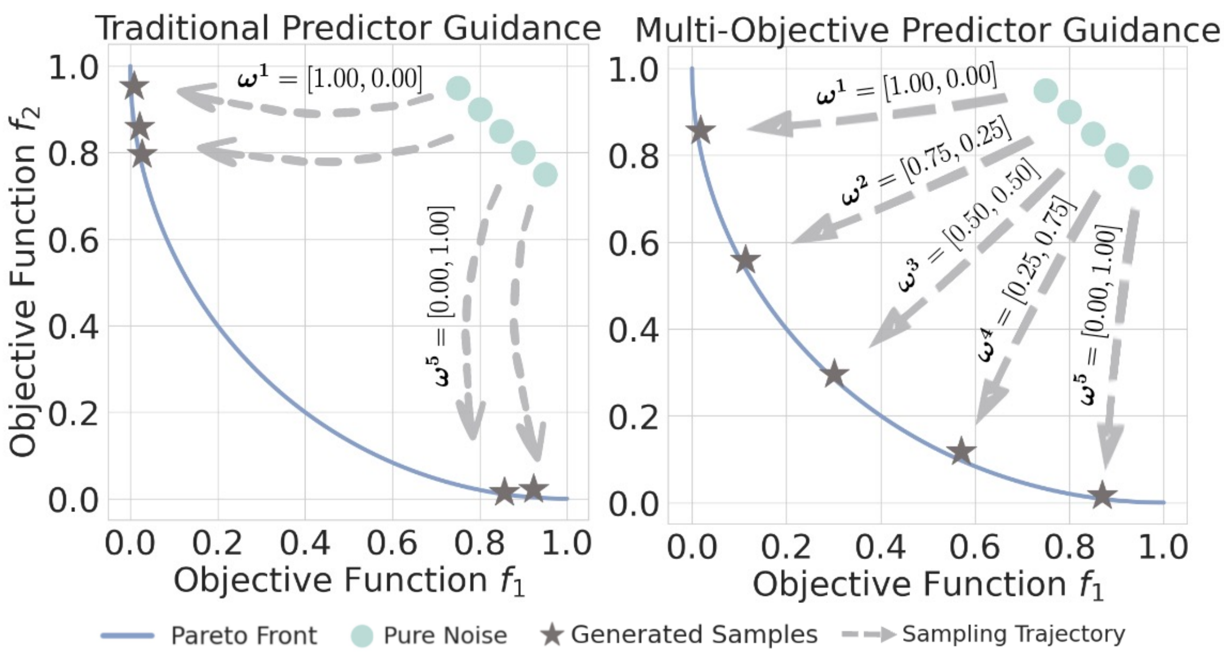
- Training this conditional flow matching model involves optimizing the loss function:

$$\mathbb{E}_{t, p_{\text{data}}(\boldsymbol{x}_1), q(\boldsymbol{x}_0)} \| \hat{v}(\boldsymbol{x}, t; \theta) - (\boldsymbol{x}_1 - \boldsymbol{x}_0) \|^2.$$

- We can then use the learned vector field $\hat{v}(\boldsymbol{x}, t; \theta)$ to generate samples by solving the neural ODE.

Multi-Objective Predictor Guidance

- Traditional predictor guidance only optimizes a single objective.
- We allow multi-objective guidance by decomposing it into individual weighted objective generation subproblems.



Multi-Objective Predictor Guidance

- Traditional predictor guidance in flow matching is derived as:

$$\tilde{v}(\boldsymbol{x}_t, t, y; \boldsymbol{\theta}) = \hat{v}(\boldsymbol{x}_t, t; \boldsymbol{\theta}) + \frac{1-t}{t} \nabla_{\boldsymbol{x}_t} \log p_{\boldsymbol{\beta}}(y \mid \boldsymbol{x}_t, t),$$

where $p_{\boldsymbol{\beta}}(y \mid \boldsymbol{x}_t, t)$ represents the predicted property distribution.

- We optimize multiple properties $[f_1(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})]$ simultaneously by defining a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \cdots, \omega_m]$, where each $\omega_i > 0$ and $\sum_{i=1}^m \omega_i = 1$. Then the weighted property prediction is written as:

$$\hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x}_t; \boldsymbol{\beta}) = \sum_{i=1}^{m} -\hat{f}_i(\hat{\boldsymbol{x}}_1(\boldsymbol{x}_t); \boldsymbol{\beta}_i)\omega_i,$$

where $\hat{f}_i$ predicts the i^{th} objective for $\boldsymbol{x}_t$, trained using only $\boldsymbol{x}_1$ data, and the negative sign indicates minimization.

Multi-Objective Predictor Guidance

- We formulate the weighted distribution as:
$$p_{\beta}(y \mid \hat{\boldsymbol{x}}_1(\boldsymbol{x}_t), \boldsymbol{\omega}) = e^{\gamma \hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x}_t; \boldsymbol{\beta})}/Z,$$
where γ is a scaling factor and Z is the normalization constant.

- Similar to the single-objective guidance, we then have:
$$\tilde{v}(\boldsymbol{x}_t, t, y; \boldsymbol{\theta}) = \hat{v}(\boldsymbol{x}_t, t; \boldsymbol{\theta}) + \gamma \frac{1-t}{t} \nabla_{\boldsymbol{x}_t} \hat{f}_{\boldsymbol{\omega}}(\boldsymbol{x}_t; \boldsymbol{\beta}).$$

- For the sample $\boldsymbol{x}_t^i$ at time step t, we advance to the next time step:
$$\hat{\boldsymbol{x}}_s^i = \boldsymbol{x}_t^i + \tilde{v}(\boldsymbol{x}_t^i, t, y; \boldsymbol{\theta}) \Delta t + g\sqrt{\Delta t} \epsilon,$$
where $s = t + \Delta t$ indicates the next time step, $g = 0.1$ denotes the noise factor, and ϵ is a standard Gaussian noise term. By sampling different ϵ, we could obtain O offsprings at each time step.

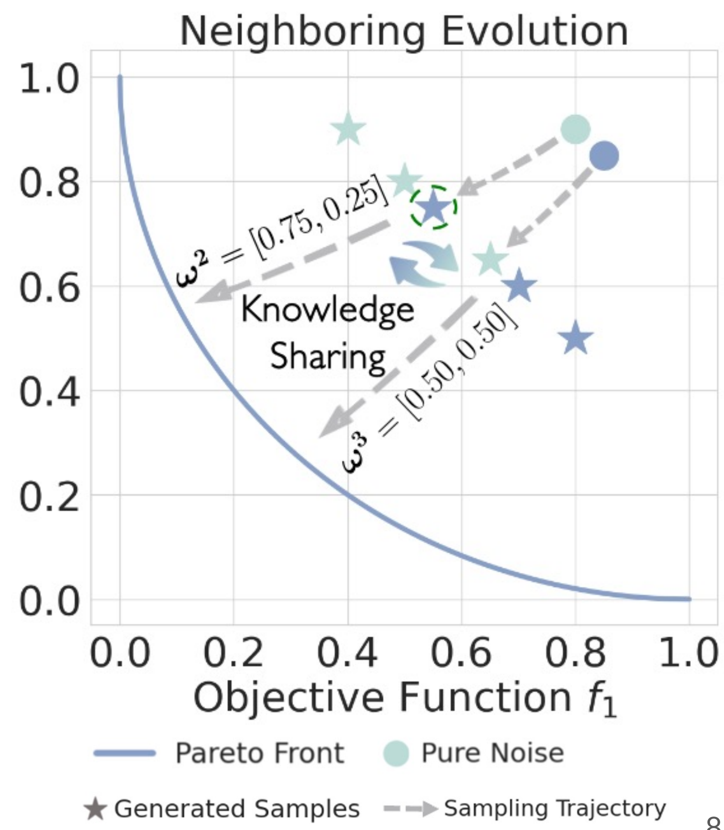
Neighboring Evolution

- Weighted distributions with similar weight vectors are likely to produce similar samples. For a distribution associated with $\boldsymbol{\omega}^i$, its neighbors are identified as the K distributions whose weight vectors have the smallest angular distances to $\boldsymbol{\omega}^i$.

- Given that there are K neighboring samples for sample i and the O offsprings we obtained before, this result in a set: $\boldsymbol{X}_i = \{\hat{\boldsymbol{x}}_s^{j,o} \mid j \in \mathcal{N}(i), o \in \{1, 2, \cdots, O\}\}$.

- We Update the current sample $\boldsymbol{x}_t^i$ using the neighboring set $\boldsymbol{X}_i$:

$$\boldsymbol{x}_s^i = \arg \max_{\hat{\boldsymbol{x}}_s^{j,o} \in \boldsymbol{X}_i^l} f_{\boldsymbol{\omega}^i}(\hat{\boldsymbol{x}}_s^{j,o}; \boldsymbol{\beta}).$$



Experiment: Tasks

- **Synthetic Function (Synthetic):** encompasses several subtasks involving popular functions with 2-3 objectives, aiming to identify the Pareto Set with offline designs;
- **Multi-Objective Neural Architecture Search (MO-NAS):** consists of tasks searching for a neural architecture that optimizes multiple metrics, such as latency and parameters count;
- **Multi-Objective Reinforcement Learning (MORL):** involves finding a control policy for a robot to maximize speed and energy efficiency or objectives related to running and jumping;
- **Scientific Design (Sci-Design):** includes tasks that concentrate on molecule or protein discovery to achieve certain desired properties.
- **Real-World Applications (RE):** encompasses a variety of practical optimization challenges, including four-bar truss and pressure vessel design. The MOPortfolio task, which focuses on optimizing expected returns and variance of returns is also included here.

Experiment: Evaluation Metrics Hypervolume (HV)

- The HV metric quantifies the size of the objective space that is dominated by the candidate set $\mathcal{B}$ and bounded by a reference point $\boldsymbol{r} = (r^1, r^2, \ldots, r^m)$. Mathematically, the HV is defined as:

$$HV(\mathcal{B}) = \mathrm{vol}\left(\bigcup_{\boldsymbol{y} \in \mathcal{B}} \prod_{i=1}^{m} [y^i, r^i]\right),$$

- where $\prod_{i=1}^{m} [y^i, r^i]$ represents an m-dimensional hyperrectangle (or box) spanning from the coordinates of $\boldsymbol{y}$ to the reference point $\boldsymbol{r}$ along each objective, and $\mathrm{vol}(\cdot)$ denotes the Lebesgue measure of the union of these hyperrectangles.

- In simple terms, a larger hypervolume indicates that the solution set is both close to the Pareto front and well-distributed across the objective space.

Experiment Results

- ParetoFlow consistently achieves the highest ranks across all tasks, underscoring its effectiveness.
- Both DNN-based and generative modeling-based methods frequently outperform D(best), illustrating the strength of predictor and generative modeling.
- MO-NAS and Sci-Design tasks are predominantly discrete, with MO-NAS having a higher dimensionality. Generative modeling methods show reduced effectiveness on MO-NAS, which may stem from the difficulty in modeling high-dimensional discrete data.

Table 1: Average rank of different methods on each type of task in Off-MOO-Bench.

Methods	Synthetic	MO-NAS	MORL	Sci-Design	RE	All Tasks
D-Best	16.82 ± 6.28	14.42 ± 4.11	15.00 ± 4.00	13.75 ± 6.91	18.06 ± 3.93	16.02 ± 5.13
E2E	10.91 ± 8.20	6.05 ± 3.32	12.50 ± 1.50	9.75 ± 4.97	9.69 ± 5.65	8.73 ± 5.88
E2E + GradNorm	12.64 ± 6.68	13.42 ± 5.54	8.50 ± 0.50	13.50 ± 5.12	14.19 ± 5.87	13.31 ± 5.87
E2E + PcGrad	9.45 ± 6.37	6.42 ± 3.18	16.50 ± 2.50	14.00 ± 3.16	10.88 ± 6.17	9.40 ± 5.70
MH	11.55 ± 7.19	$\underline{5.26} \pm \underline{3.93}$	12.00 ± 4.00	12.50 ± 3.28	10.00 ± 5.67	8.87 ± 6.00
MH + GradNorm	10.45 ± 6.21	16.42 ± 4.84	18.00 ± 2.00	14.75 ± 4.44	17.00 ± 4.72	15.27 ± 5.64
MH + PcGrad	11.45 ± 4.58	6.84 ± 2.83	18.50 ± 0.50	13.50 ± 5.41	11.06 ± 6.24	10.08 ± 5.46
MM	$\underline{4.91} \pm \underline{4.17}$	6.74 ± 3.81	16.50 ± 1.50	6.75 ± 4.32	6.69 ± 3.46	$\underline{6.71} \pm \underline{4.31}$
MM + COMs	13.00 ± 3.86	9.53 ± 4.42	12.50 ± 2.50	12.25 ± 6.83	14.62 ± 4.75	12.15 ± 5.06
MM + RoMA	13.27 ± 7.53	8.21 ± 5.75	10.00 ± 3.00	12.00 ± 2.45	10.25 ± 5.14	10.27 ± 6.06
MM + IOM	6.91 ± 3.78	5.37 ± 3.60	6.50 ± 0.50	10.75 ± 1.92	7.25 ± 4.02	6.73 ± 3.88
MM + ICT	14.45 ± 5.77	8.53 ± 3.12	9.50 ± 3.50	12.50 ± 7.12	11.75 ± 6.54	11.12 ± 5.77
MM + Tri-Mentor	11.00 ± 5.89	9.05 ± 5.71	10.50 ± 1.50	13.00 ± 3.54	10.50 ± 5.82	10.27 ± 5.65
MOEA/D + MM	10.55 ± 4.83	12.58 ± 5.02	11.00 ± 1.00	10.75 ± 6.87	12.12 ± 6.62	11.81 ± 5.66
MOBO	10.91 ± 4.42	14.74 ± 3.82	17.00 ± 0.00	8.25 ± 6.61	11.00 ± 5.79	12.37 ± 5.32
MOBO-qParEGO	13.36 ± 3.98	16.63 ± 3.77	21.00 ± 0.00	12.75 ± 8.04	17.69 ± 4.55	16.13 ± 4.91
MOBO-JES	17.27 ± 3.11	22.00 ± 0.00	21.00 ± 0.00	18.75 ± 5.63	13.62 ± 5.19	18.13 ± 5.00
PROUD	8.55 ± 6.33	14.53 ± 4.43	$\underline{2.50} \pm \underline{0.50}$	6.25 ± 3.49	5.75 ± 5.02	9.46 ± 6.39
LaMBO-2	10.18 ± 6.55	14.37 ± 4.66	3.00 ± 1.00	$\underline{5.00} \pm \underline{1.22}$	$\underline{5.00} \pm \underline{4.72}$	9.44 ± 6.49
CorrVAE	11.73 ± 6.14	17.74 ± 2.95	4.50 ± 0.50	8.00 ± 4.18	9.56 ± 6.00	12.69 ± 6.35
MOGFN	10.55 ± 6.04	15.95 ± 3.98	3.50 ± 1.50	5.50 ± 4.50	5.88 ± 4.97	10.42 ± 6.63
ParetoFlow **(ours)**	**4.00 ± 3.88**	**3.47 ± 4.26**	**1.00 ± 0.00**	**2.75 ± 1.48**	**2.44 ± 3.45**	**3.12 ± 3.77**

Conclusion

- In this work, we apply flow matching to offline multi-objective optimization, introducing ParetoFlow.
- Our multi-objective predictor guidance module employs a uniform weight vector for each sample generation, guiding samples to approximate the Pareto-front.
- Additionally, our neighboring evolution module enhances knowledge sharing between neighboring distributions.
- Experiments across various benchmarks confirm the effectiveness of our approach.

Thanks for your attention!

Paper

Code

McGill University

Mila

Polytechnique Montréal

CIFAR