Gen-DFL: Decision-Focused Generative Learning for Robust Decision Making

Prince Zizhuang Wang[1], Shuyi Chen[1], Jinhao Liang[2], Ferdinando Fioretto[2], Shixiang Zhu[1]

[1]Heinz College, Carnegie Mellon University [2]Department of Computer Science, University of Virginia

Motivation

Decision-focused learning (DFL) trains predictors for downstream decision quality, but standard DFL usually commits to a **single point prediction**. This is brittle when uncertainty is high-dimensional, nonlinear, or concentrated in the tail.

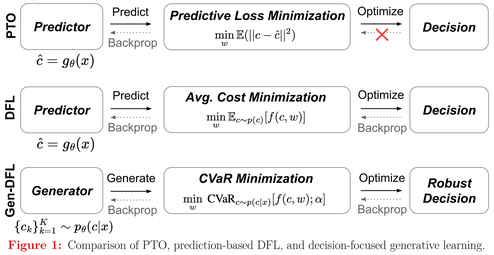

Figure 1: Comparison of PTO, prediction-based DFL, and decision-focused generative learning.

Takeaway: point predictions can optimize average behavior while missing adverse outcomes that drive risk-sensitive costs.

Problem Setup

We seek decisions that perform well in the worst-α% of outcomes:

$$w^\star(x;\alpha) = \arg\min_w \ \text{CVaR}_{c\sim p(c|x)}\left[f(c,w);\alpha\right].$$

The risk level α controls the trade-off:

- $\alpha \to 0$: robust, worst-case behavior.
- $\alpha \to 1$: expectation-based optimization.
- Smaller α: harder tail estimation, larger risk-sensitivity.

Generate-Then-Optimize (GTO) Paradigm

Within `Gen-DFL`, **generate-then-optimize** (GTO) replaces brittle point predictions with generated cost scenarios: learn $p(c|x)$, then optimize decisions over samples. This helps most when DFL faces:

- High-dimensional uncertainty and nonlinear objectives.
- Tail-risk requirements over the worst-α% outcomes.
- Unknown or multimodal conditional distributions.

GTO Decision Rule

Replace deterministic cost prediction with a conditional generative model, then solve the downstream CVaR problem over generated scenarios:

$$w_\theta^\star(x;\alpha) = \arg\min_w \ \text{CVaR}_{c\sim p_\theta(c|x)}\left[f(c,w);\alpha\right].$$

Generate $c_1,\dots,c_K \sim p_\theta(c|x)$ → **Rank risk** worst-α% scenarios → **Optimize** $w_\theta^\star$ via SAA

Core idea: learn plausible scenario distributions that preserve uncertainty, expose high-risk outcomes, and let the optimizer choose decisions using the same CVaR criterion used at evaluation time.

Gen-DFL: End-to-End Learning Framework

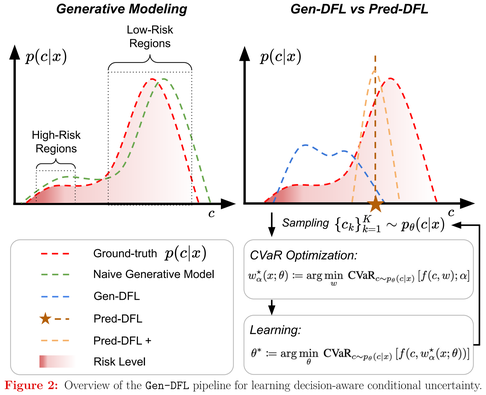

Figure 2: Overview of the `Gen-DFL` pipeline for learning decision-aware conditional uncertainty.

Takeaway: `Gen-DFL` models $p(c|x)$ with a conditional generator while emphasizing samples that matter for the downstream decision objective.

Training Objective

Define CVaR regret under the true distribution:

$$\text{Regret}_{\theta,p}(x;\alpha) = \text{CVaR}_{p(c|x)}\left[f(c,w_\theta^\star) - f(c,w^\star);\alpha\right].$$

Because $p(c|x)$ is unknown, a proxy $q(c|x)$ estimates the regret during training:

$$\ell_{\texttt{Gen-DFL}}(\theta;q,\alpha) = \beta\,\mathbb{E}_x[\text{Regret}_{\theta,q}(x;\alpha)] + \gamma\,\ell_{\text{gen}}(\theta).$$

Loss components

- **Decision term**: estimates CVaR regret with a proxy distribution $q(c|x)$ when the true $p(c|x)$ is unknown.
- **Generative term**: keeps generated samples faithful to the observed data distribution.
- **Trade-off**: larger β emphasizes downstream decision quality; γ controls distributional regularization.

Gen-DFL Algorithm

1. Sample scenarios $\{c_k\}_{k=1}^K \sim p_\theta(c|x)$.
2. Solve $w_\theta^\star(x;\alpha)$ by empirical CVaR minimization.
3. Evaluate proxy CVaR regret under fixed $q(c|x)$.
4. Update θ with decision loss plus generative regularization.

Implementation note: the paper uses conditional normalizing flows for $p_\theta(c|x)$, but the framework is model-agnostic and only requires sample generation for SAA.
Inference: sample from the learned $p_\theta(c|x)$ and solve the same CVaR optimization problem.
Training loop: alternate scenario generation, optimization, regret scoring, and generator updates until convergence.

Theoretical Takeaways

The bound explains where distributional modeling helps most:

- **High variance**: point estimates discard important dispersion.
- **High dimension**: prediction bias accumulates across parameters.
- **Small** α: tail data become sparse, so generated scenarios help cover high-risk outcomes.

Main Theorem

Under Lipschitz assumptions, the regret gap $\Delta R(x) = R_{\text{pred}}(x;\alpha) - R_{\text{gen}}(x;\alpha)$ is controlled by distribution mismatch, predictor bias, conditional variance, and risk level:

$$\mathbb{E}_x[|\Delta R(x)|] \lesssim \mathbb{E}_x\Big[\tfrac{1}{\alpha}\mathcal{W}(p_\theta,p) + \tfrac{1}{\alpha}\|\text{Bias}[g]\| + \left(\tfrac{1}{\alpha}+1\right)\sqrt{\|\text{Var}[c|x]\|} + \text{CVaR}_{p(c|x)}[\|\text{Bias}[g(x)]\|;\alpha]\Big].$$

Implication: when variance, dimensionality, or risk sensitivity increase, `Pred-DFL` suffers from point bias while `Gen-DFL` uses distributional structure.

Experimental Setup

Synthetic tasks: portfolio CVaR, fractional knapsack, and structured shortest path under nonlinear, high-variance, and high-dimensional uncertainty.
Real-data tasks: energy-cost-aware scheduling and COVID hospital-bed allocation under forecast uncertainty.
Compared methods: best baseline selected from Pairwise, Listwise, NCE, MAP, SPO+, Diff-DRO, and 2Stage (PTO).
Metric: average relative regret; lower values mean the learned decision is closer to the task-specific oracle. Real-data baseline comparisons use $\alpha = 1$ to match standard Pred-DFL evaluation.

Main Experimental Results

Table 1: Average relative regret vs. the best baseline. Synthetic rows use polynomial degree; real-data rows use dataset name. Lower is better.

Task	Degree / Dataset	Best Baseline	`Gen-DFL`
Synthetic Degree Sweep			
Portfolio	Deg-2	6.92±0.26	**3.71±0.18**
Portfolio	Deg-4	7.23±0.60	**3.81±0.22**
Portfolio	Deg-6	7.01±0.26	**4.31±0.32**
Portfolio	Deg-8	6.98±0.98	**3.59±0.31**
Knapsack	Deg-2	19.63±4.50	**17.60±3.38**
Knapsack	Deg-4	16.58±3.68	**15.21±3.75**
Knapsack	Deg-6	**17.45±7.20**	17.91±2.44
Knapsack	Deg-8	20.38±6.70	**19.29±3.75**
Shortest Path	Deg-2	2.65±0.25	**1.87±0.20**
Shortest Path	Deg-4	11.78±2.89	**3.64±0.43**
Shortest Path	Deg-6	23.76±4.21	**6.52±0.71**
Shortest Path	Deg-8	39.81±5.46	**13.36±2.59**
Real-Data Tasks			
Energy	Scheduling	1.49±0.12	**1.09±0.09**
COVID Resource	Hospital beds	**16.41±3.80**	16.86±4.62

Takeaway: `Gen-DFL` wins on most settings, with the clearest gains on nonlinear shortest-path and portfolio tasks; the compact table also shows the few near-tie cases for transparency.

Robustness Stress Tests

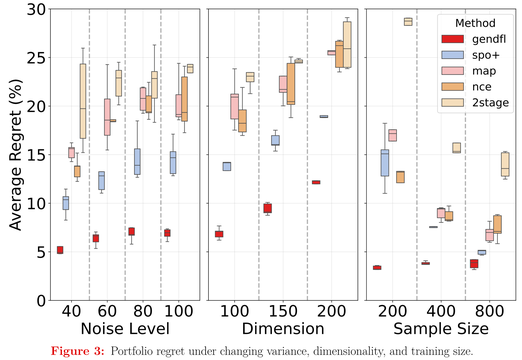

Figure 3: Portfolio regret under changing variance, dimensionality, and training size.

Takeaway: `Gen-DFL` remains stable as variance, dimensionality, and data scarcity increase.

Sensitivity Analysis

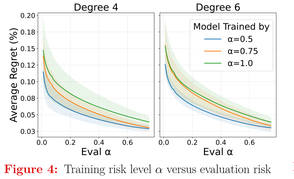

Figure 4: Training risk level α versus evaluation risk level in portfolio optimization.

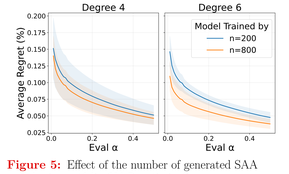

Figure 5: Effect of the number of generated SAA samples in portfolio CVaR optimization.

Takeaway: lower training α improves adverse-region decisions, while more SAA samples improve CVaR approximation and decision quality.
Reading: the sensitivity plots diagnose how `Gen-DFL` trades off robustness through α and numerical quality through the number of generated samples.

Conclusions

- **Generate-then-optimize**: `Gen-DFL` learns the conditional distribution $p_\theta(c|x)$ and optimizes decisions over generated scenarios, so uncertainty is represented as decision-relevant samples rather than a single point estimate.
- **Decision-focused training**: the learning objective combines downstream CVaR regret with generative regularization, encouraging samples that remain statistically plausible while emphasizing high-loss regions that affect robust decisions.
- **Robust decision quality**: theory and experiments show the largest gains when variance, dimensionality, nonlinearity, or tail sensitivity increase, with strong regret reductions across portfolio, shortest-path, and energy scheduling tasks.