WHAT DECISIONS NEED FROM A DISTRIBUTION: DECISION-FOCUSED LEARNING VIA OPTIMALITY-FIELD MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decision-Focused Learning (DFL) trains predictive models for downstream decision quality rather than predictive accuracy alone. For point predictions, recent work relates the regret gradient to prediction error and local solution sensitivity. In stochastic optimization, however, the optimal decision may depend on distributional information that a point prediction cannot represent. We show that a stochastic objective uses a predictive distribution through its *distributional optimality field*, a function over the decision space whose values are distributional functionals. Differentiating realized cost through the stochastic decision yields a sample gradient with the same structure: the field residual replaces prediction error and is transformed through the local solution sensitivity. When the field is available analytically, this gradient requires one solve and no predictive scenario sampling or differentiable optimization layer. Since the field being learned also determines the induced decision, this gradient constrains the field only where the decision currently lies. We therefore introduce **Functional-DFL**, which keeps the exact regret gradient and uses the same observation to match the field at additional decisions, so that regret remains controlled as the decision moves. Across two synthetic and two real-world problems, Functional-DFL achieves the best average decision quality among the compared baselines at low computational cost. Code is available at .

1 INTRODUCTION

Decision-Focused Learning (DFL) trains a predictive model for the decisions it induces rather than for predictive accuracy alone (Donti et al., 2017; Mandi et al., 2024). A prediction is passed to an optimization problem, the resulting decision is evaluated against the realized outcome, and the model is updated to reduce its regret. A central challenge in DFL is obtaining this learning signal through the optimization problem, whether by differentiating through the solver, surrogate losses, or smoothing (Amos & Kolter, 2017; Agrawal et al., 2019; Berthet et al., 2020; Elmachtoub & Grigas, 2022). For point predictions, recent work has shown that the signal has a simple form. The regret gradient is the prediction error projected through the local geometry of the solution, so the curvature and active constraints determine which errors move the decision, and the gradient follows from a single solve and its sensitivity (Lee et al., 2026).

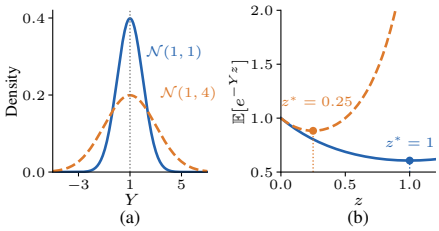


Figure 1: Two distributions with the same mean, $\mathcal{N}(1,1)$ and $\mathcal{N}(1,4)$, induce optima $z^* = 1$ and $z^* = 0.25$ for $\mathbb{E}[e^{-Yz}]$ on $[0,1]$, while a point prediction at the mean induces $z = 1$ for both.

Point predictions, however, can be restrictive in stochastic decision problems, where the optimal decision may depend on uncertainty that no single value can summarize (Sadana et al., 2025). In Figure 1, two normal distributions with the same mean lead to different optimal decisions under an exponential expected cost, whereas a point prediction at the mean leads to the same decision for both. Prior work has therefore trained predictive distributions or generated scenarios through the stochastic problem, or replaced the distribution by a learned expected objective (Donti et al., 2017;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Wang et al., 2026; Zhao et al., 2026b; Kong et al., 2025). A predictive distribution has no prediction-error vector, however, so it is unclear which of its errors move the decision, and the regret gradient is typically recovered by sampling scenarios and differentiating through the sampled problem. This raises the question of what information in a predictive distribution enters optimality and regret.

We show that this information is a function over the decision space, which we call the *distributional optimality field*. At each decision it is a distributional functional chosen by the objective, such as a moment or a tail expectation. A perturbation of the distribution enters the decision only through this field, so matching the full predictive distribution is unnecessary. When a predictive distribution is trained by differentiating realized cost through the stochastic decision, the sample gradient is exactly the residual between the predictive field and the realized cost gradient at the induced decision, projected through the same local solution sensitivity, and for point predictions with bilinear costs this reduces to the prediction-error form. When the field is available analytically, as it is for Gaussian mixtures in our tasks, this gradient follows from one solve and its sensitivity, with no scenario sampling and no optimization layer, and expressive predictive distributions can be trained where a layer tied to its own solver fails to converge in our experiments.

A prediction error is the same at every decision, whereas the field residual is evaluated at the induced decision only, so the regret gradient constrains the field where the decision currently lies even though the decision moves during training. A single observed outcome, however, provides an unbiased target for the field at any feasible decision through its realized cost gradient. We therefore introduce **Functional-DFL**, which keeps the exact regret gradient and uses the same observation to match the field at additional decisions without additional solves, so that regret remains controlled as the decision moves and the population minimizer of a well-specified model is preserved.

Our contributions are as follows.

1. **Optimality field and the regret gradient.** We identify the distributional optimality field as what a stochastic objective uses from a predictive distribution and prove that the sample regret gradient is the field residual at the induced decision projected through this sensitivity, which contains the point-prediction case and needs one solve for analytic fields.
2. **Functional-DFL.** We show that this gradient constrains the field only at the induced decision and introduce Functional-DFL, which matches the same residual at additional decisions from one observation, with regret and curvature guarantees for the added supervision.
3. **Empirical evaluation.** On four stochastic optimization problems, Functional-DFL achieves the lowest mean decision loss among the compared baselines at low computational cost. Ablations separate the regret gradient from the added field supervision and from a differentiable optimization layer with the same predictor.

2 Related Work

Decision-Focused Learning. DFL methods obtain learning signals through differentiable optimization, relaxations, surrogate losses, and perturbations (Donti et al., 2017; Mandi et al., 2024; Amos & Kolter, 2017; Agrawal et al., 2019; Wilder et al., 2019; Elmachtoub & Grigas, 2022; Berthet et al., 2020; Pogančić et al., 2020). Stochastic decision problems have also been addressed with predictive distributions or generated scenarios, integrated estimation and optimization, learned expected objectives, and distributional discrepancies designed for decision quality (Donti et al., 2017; Wang et al., 2026; Zhao et al., 2026b; Elmachtoub et al., 2023; Qi et al., 2026; Kong et al., 2025; Liu & Liu, 2026). Recent work trains generative outcome models through sampled stochastic programs or learns the expected objective over decisions directly (Zhao et al., 2026b; Kong et al., 2025), whereas we retain a predictive distribution and work with its optimality field.

Sensitivity Analysis of Optimal Solutions. Sensitivity analysis characterizes how optimal solutions respond to perturbations in problem data (Fiacco, 1976; Robinson, 1980; Bonnans & Shapiro, 2000). In stochastic programs, the perturbed object may be the distribution itself (Shapiro, 1990; Dupačová, 1990; Römisch & Schultz, 1991). Active constraints and local optimality systems also support efficient differentiation through optimization (Magoon et al., 2025; Zhao et al., 2026a), while recent work relates prediction error to the regret gradient through local decision geometry (Lee et al., 2026). We extend this sensitivity view to predictive distributions through the optimality field.

3 DISTRIBUTIONAL DECISION-FOCUSED LEARNING

3.1 PROBLEM SETTING

We observe a context $x \in \mathcal{X}$ and choose a decision $z \in \mathbb{R}^d$ before an uncertain outcome $Y \in \mathcal{Y} \subseteq \mathbb{R}^p$ is revealed. The decision belongs to a polyhedral feasible set $\mathcal{Z} = \{z : Az = b,\ Gz \leq h\}$, which may depend on x, and incurs the realized cost $f(y, z) = r(z) + \phi(y, z)$. We assume that r and $\phi(y, \cdot)$ are convex and suppress the dependence on x when clear from context.

Let P denote the true conditional distribution of Y and Q a predictive distribution. The stochastic problem induced by Q is

$$F_Q(z) = r(z) + \mathbb{E}_{Y\sim Q}[\phi(Y, z)], \qquad z^*_Q \in \arg\min_{z\in\mathcal{Z}} F_Q(z). \tag{1}$$

We evaluate Q through the decision it induces under the true distribution,

$$R_P(Q) = F_P(z^*_Q) - F_P(z^*_P). \tag{2}$$

For a parameterized predictive model $Q_\theta(\cdot \mid x)$, minimizing expected regret is equivalent to minimizing $\mathbb{E}[f(Y, z^*_{Q_\theta(X)})]$, since the optimal value under P is independent of θ.

Our local analysis considers a regular solution z^*_Q near which F_Q is twice continuously differentiable. Let $\mathcal{I}_Q$ denote the active inequality constraints, $C_Q = [A^\top, G^\top_{\mathcal{I}_Q}]^\top$, and $H_Q = \nabla^2_z F_Q(z^*_Q)$. We assume LICQ, strict complementarity, and positive definiteness of the reduced Hessian on $\ker C_Q$. If U_Q is an orthonormal basis of $\ker C_Q$, define

$$S_Q = U_Q(U_Q^\top H_Q U_Q)^{-1} U_Q^\top.$$

Under these standard conditions, S_Q is the local solution sensitivity operator determined by the active constraints and local curvature (Fiacco, 1976; Robinson, 1980; Bonnans & Shapiro, 2000).

3.2 DISTRIBUTIONAL OPTIMALITY FIELD

A predictive distribution may vary in ways that do not affect stochastic optimality. Let $\psi(y, z) = \nabla_z \phi(y, z)$, defined wherever $\phi(y, \cdot)$ is differentiable, denote the contribution of one outcome to first-order optimality. We define the *distributional optimality field*

$$\Phi_Q(z) = \mathbb{E}_{Y\sim Q}[\psi(Y, z)]. \tag{3}$$

Since $\nabla_z F_Q(z) = \nabla r(z) + \Phi_Q(z)$, the predictive distribution enters first-order optimality through Φ_Q. For each fixed z, the map $Q \mapsto \Phi_Q(z)$ is a distributional functional, and the field collects these functionals over the decision space.

If $\phi(y, z) = y^\top z$, then $\Phi_Q(z) = \mathbb{E}_Q[Y]$ and only the mean enters first-order optimality. For nonlinear costs, the field can vary with z and may involve distribution functions, tail quantities, or more general nonlinear expectations. If $\Phi_Q(z) = \Phi_P(z)$ for every $z \in \mathcal{Z}$, then $F_Q - F_P$ is constant over the convex feasible region, so the two distributions induce the same optimization problem up to an additive constant. More locally, if $\Phi_Q(z^*_P) = \Phi_P(z^*_P)$, then z^*_P is also optimal under Q, and $z^*_Q = z^*_P$ when F_Q is strictly convex. The converse need not hold, since two distributions may share the same optimal solution while inducing different fields elsewhere. Thus preserving the field is sufficient for the induced problem, and matching the full predictive distribution is unnecessary.

3.3 DISTRIBUTIONAL SENSITIVITY

Consider a differentiable path $Q_t = Q + t\nu + o(t)$, where ν is a signed measure with $\nu(\mathcal{Y}) = 0$. The resulting field perturbation at a fixed decision is $\int \psi(y, z)\, d\nu(y)$, the directional derivative of the functional $Q \mapsto \Phi_Q(z)$ along ν. Evaluated at the current optimizer, we write

$$\mathcal{B}_Q \nu = \int_{\mathcal{Y}} \psi(y, z^*_Q)\, d\nu(y).$$

Differentiating the KKT conditions along this path gives the standard sensitivity system, here specialized to perturbations of the predictive distribution (Shapiro, 1990; Dupačová, 1990),

$$\begin{bmatrix} H_Q & C_Q^\top \\ C_Q & 0 \end{bmatrix} \begin{bmatrix} Dz_Q^*[\nu] \\ D\eta_Q[\nu] \end{bmatrix} = - \begin{bmatrix} \mathcal{B}_Q \nu \\ 0 \end{bmatrix}.$$

Restricting the system to $\ker C_Q$, on which $U_Q^\top H_Q U_Q$ is invertible, gives the following result.

Proposition 1 (Distributional decision and regret sensitivity)**.** *Under the regularity conditions of Section 3.1, the decision and the regret satisfy*

$$Dz_Q^*[\nu] = -S_Q \mathcal{B}_Q \nu, \qquad DR_P(Q)[\nu] = \Delta_{Q,P}^\top S_Q \mathcal{B}_Q \nu,$$

where $\Delta_{Q,P} = \Phi_Q(z_Q^*) - \Phi_P(z_Q^*)$. Moreover, for a perturbation around the true distribution,

$$R_P(P + t\nu) = \frac{t^2}{2} (\mathcal{B}_P \nu)^\top S_P (\mathcal{B}_P \nu) + o(t^2).$$

The regret derivative follows from the decision derivative. Predictive stationarity gives $\nabla F_Q(z_Q^*) = -C_Q^\top \eta_Q$, which is orthogonal to $Dz_Q^*[\nu] \in \ker C_Q$, and $\nabla F_Q - \nabla F_P = \Phi_Q - \Phi_P$, so

$$DR_P(Q)[\nu] = \nabla F_P(z_Q^*)^\top Dz_Q^*[\nu] = -\Delta_{Q,P}^\top Dz_Q^*[\nu] = \Delta_{Q,P}^\top S_Q \mathcal{B}_Q \nu.$$

At $Q = P$ the first-order term vanishes, and the quadratic term follows from $S_P H_P S_P = S_P$. Proposition 1 separates the effect of the stochastic objective from the local geometry of the constrained problem. A distributional perturbation first induces an optimality perturbation through $\mathcal{B}_Q$, which is then mapped to a solution perturbation by S_Q. The proof is provided in Appendix A.1.

Two distributional perturbations have the same first-order effect on the decision exactly when $U_Q^\top \mathcal{B}_Q(\nu_1 - \nu_2) = 0$. A perturbation is locally invisible when $U_Q^\top \mathcal{B}_Q \nu = 0$, either because it leaves stochastic optimality unchanged at the current decision or because its effect lies in directions absorbed by the active constraints.

The optimality field also gives a nonlocal relation. Let $\varepsilon_{Q,P}(z) = \Phi_Q(z) - \Phi_P(z)$. If F_P is m-strongly convex on $\mathcal{Z}$, then $\|z_Q^* - z_P^*\| \le \|\varepsilon_{Q,P}(z_Q^*)\|/m$ and $R_P(Q) \le \|\varepsilon_{Q,P}(z_Q^*)\|^2/(2m)$. Thus the field error at the induced decision controls both decision error and regret under strong convexity. A proof is provided in Appendix A.2.

4 The Regret Gradient through the Optimality Field

We characterize the regret gradient of a predictive distribution through the optimality field and local solution sensitivity, yielding an exact representation that propagates the field residual at the induced solution through the local optimization geometry.

4.1 Exact Gradient Representation

We consider training Q_θ by differentiating the realized cost $f(y, z_{Q_\theta}^*)$ through the stochastic decision, following the differentiable stochastic optimization setting of Donti et al. (2017). The expectation of this sample gradient is the regret gradient in Eq. 2. Let $z_\theta^* = z_{Q_\theta}^*$ and $S_\theta = S_{Q_\theta}$. For a fixed decision z, let $J_\theta(z) = \partial_\theta \Phi_{Q_\theta}(z)$ denote the parameter Jacobian of the optimality field. Proposition 1 gives

$$D_\theta z_\theta^* = -S_\theta J_\theta(z_\theta^*).$$

For an observation y, define the field residual

$$e_\theta(y, z) = \Phi_{Q_\theta}(z) - \psi(y, z). \tag{4}$$

Since $\mathbb{E}_P[\psi(Y, z)] = \Phi_P(z)$, the expected residual is $\Phi_{Q_\theta}(z) - \Phi_P(z)$. Let $\bar{z}_\theta$ and $\bar{S}_\theta$ denote the current values of z_θ^* and S_θ, held fixed under differentiation, and define

$$\ell_{\mathrm{reg}}(\theta; y) = \frac{1}{2} e_\theta(y, \bar{z}_\theta)^\top \bar{S}_\theta e_\theta(y, \bar{z}_\theta). \tag{5}$$

Theorem 2 (Exact field representation of the regret gradient)**.** *Under the regularity conditions of Section 3.1, for every observation y such that $\phi(y,\cdot)$ is differentiable at z^*_θ,*

$$\nabla_\theta f(y, z^*_\theta) = J_\theta(z^*_\theta)^\top S_\theta e_\theta(y, z^*_\theta) = \nabla_\theta \ell_{\mathrm{reg}}(\theta; y).$$

If these gradients are integrable under P and differentiation and expectation commute,

$$\mathbb{E}_P[\nabla_\theta \ell_{\mathrm{reg}}(\theta; Y)] = \nabla_\theta R_P(Q_\theta) = J_\theta(z^*_\theta)^\top S_\theta \big[\Phi_{Q_\theta}(z^*_\theta) - \Phi_P(z^*_\theta)\big].$$

By the chain rule with $D_\theta z^*_\theta = -S_\theta J_\theta(z^*_\theta)$,

$$\nabla_\theta f(y, z^*_\theta) = -J_\theta(z^*_\theta)^\top S_\theta \big(\nabla r(z^*_\theta) + \psi(y, z^*_\theta)\big).$$

At the induced solution, stationarity and $S_\theta C_\theta^\top = 0$ imply $S_\theta \nabla r(z^*_\theta) = -S_\theta \Phi_{Q_\theta}(z^*_\theta)$, yielding the field residual in Theorem 2. The equality is sample-wise, and the equivalence with ℓ_{reg} holds for the parameter gradient rather than the loss value. Appendix A.3 gives the full proof.

For $Q_\theta = \delta_{\hat{y}_\theta}$ and $\phi(y, z) = y^\top z$, the residual becomes $\hat{y}_\theta - y$, recovering the prediction-error form of Lee et al. (2026). For predictive distributions, the prediction error is replaced by the field residual, while the same solution sensitivity appears. When Φ_{Q_θ} is available analytically, Eq. 5 gives the exact sample gradient from one downstream solve and its local sensitivity, without scenario sampling or an optimization layer.

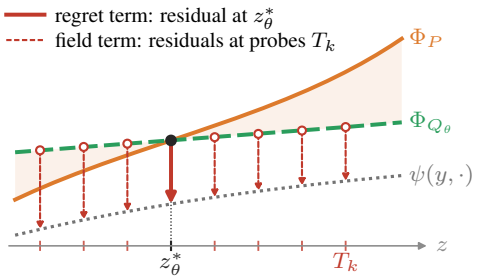


Figure 2: One observation supervises the field at many decisions. The regret gradient uses the residual between the predicted field Φ_{Q_θ} and the realized cost gradient $\psi(y,\cdot)$ at the induced decision z^*_θ only (solid arrow). Functional-DFL keeps this term and adds the residuals of the same observation at probes T_k (dashed arrows). The expectation of $\psi(Y,\cdot)$ is the true field Φ_P, and the shaded gap is the field error.

4.2 Local Field Matching

Theorem 2 gives a field-matching interpretation of the regret gradient. The residual compares the predictive field with the realized optimality contribution at z^*_θ, S_θ maps it to the first-order change in the solution, and $J_\theta(z^*_\theta)^\top$ transfers it to the parameters, so the regret gradient performs local field matching through the solution sensitivity.

This matching is local because the residual, field Jacobian, and solution sensitivity are evaluated at the induced decision. If $S_\theta J_\theta(z^*_\theta)v = 0$ for a parameter direction v, the current gradient has no component along v even when $J_\theta(T)v \neq 0$ at another feasible decision T. In Figure 2, this is the single solid arrow at z^*_θ. Shared parameters and changes in z^*_θ can still alter the field elsewhere during training, but the identity characterizes only the field evaluations that enter the current gradient.

For any fixed feasible decision z, however, $\mathbb{E}_P[\psi(Y, z)] = \Phi_P(z)$, so a single observation provides an unbiased estimate of the true field at every feasible decision, not only at the induced one.

5 Functional Decision-Focused Learning

By Theorem 2, regret is learned through the optimality field, and the regret gradient already matches the field at the induced decision. The field being learned, however, also determines that decision and its sensitivity, so the gradient constrains the field only where the current prediction places the decision, and the decision moves as the prediction changes. Functional-DFL keeps the exact regret gradient and, using the same observation, matches the same residual at additional decisions, so that regret remains controlled as the induced decision moves.

5.1 Field Supervision Beyond the Local Regret Geometry

Let κ be a distribution over decisions where the field is defined, typically feasible ones, and let $T \sim \kappa$ be conditionally independent of the observed outcome given the context. We call these

draws field probes and evaluate the residual $e_\theta(y, T)$ of Section 4 at them. Define the population field error

$$D_\kappa^2(Q, P) = \mathbb{E}_{T\sim\kappa}\left[\|\Phi_Q(T) - \Phi_P(T)\|^2\right]. \tag{6}$$

The implementation averages the squared residual over probes and the d field coordinates, and at the population level,

$$\mathbb{E}_{Y,T}\left[\frac{1}{2d}\|e_\theta(Y, T)\|^2\right] = \frac{1}{2d}D_\kappa^2(Q_\theta, P) + C(P, \kappa), \tag{7}$$

where $C(P, \kappa) = \frac{1}{2d}\mathbb{E}_T[\operatorname{tr}\operatorname{Cov}_P(\psi(Y, T))]$ is independent of θ. The additional term therefore matches the same optimality field that appears in the regret gradient rather than fitting the full predictive distribution, as the dashed arrows in Figure 2 show. The proof is provided in Appendix A.4.

Proposition 3 (Field error controls regret)**.** *Suppose F_P is m-strongly convex on $\mathcal{Z}$.*

(i) *If the field error $\varepsilon_{Q,P}$ of Section 3.3 is L-Lipschitz, $z_Q^* \in \mathcal{Z}_0$, and every $z \in \mathcal{Z}_0$ has κ-mass at least κ_τ within distance τ, then*

$$R_P(Q) \le \frac{1}{m}\left(\frac{D_\kappa^2(Q, P)}{\kappa_\tau} + L^2\tau^2\right).$$

(ii) *If $\mathcal{Z} = \operatorname{conv}\{t_1, \dots, t_K\}$, $\varepsilon_{Q,P}$ is affine on $\mathcal{Z}$, and $\kappa(\{t_k\}) \ge \pi_{\min} > 0$ for every k, then*

$$R_P(Q) \le \frac{D_\kappa^2(Q, P)}{2m\pi_{\min}}.$$

Part (i) uses local coverage of the induced decision, while part (ii) uses exact interpolation of an affine field from the vertices. In either case, controlling field error at the selected decisions controls regret without requiring the predictive distribution itself to be correct. The field term therefore supplements rather than replaces the regret gradient, which already matches the field at the induced decision, and keeps regret under this bound as that decision moves during training. The proof and task-specific conditions are given in Appendix A.5.

5.2 Functional Learning Objective

For K field probes $T_1, \dots, T_K \sim \kappa$, Functional-DFL uses

$$\mathcal{L}_{\mathrm{FDFL}}(\theta; y) = \ell_{\mathrm{reg}}(\theta; y) + \frac{\lambda}{2Kd}\sum_{k=1}^{K}\|e_\theta(y, T_k)\|^2, \qquad \lambda \ge 0. \tag{8}$$

The first term gives the exact regret gradient of Theorem 2 and the second uses the same residual at additional decisions, so $\lambda = 0$ recovers the exact regret update and $\lambda > 0$ adds auxiliary field supervision. For each context, one solve of the stochastic problem gives z_θ^* and, through the local KKT system, S_θ, while the probes require only $\Phi_{Q_\theta}(T_k)$ and $\psi(y, T_k)$ with no further solve or sensitivity computation. Task-specific forms and probe choices are given in Appendices B and C.

Under the same integrability conditions, combining Theorem 2 with Eq. 7 shows that, for κ independent of θ, the expected update is $\nabla_\theta[R_P(Q_\theta) + \frac{\lambda}{2d}D_\kappa^2(Q_\theta, P)]$. Restoring the context dependence, let $P_x = P(\cdot \mid x)$, $Q_{\theta,x} = Q_\theta(\cdot \mid x)$, and $T \mid X = x \sim \kappa_x$. The corresponding population objective is $\mathcal{L}_\lambda(\theta) = \mathbb{E}_X[R_{P_X}(Q_{\theta,X}) + \frac{\lambda}{2d}D_{\kappa_X}^2(Q_{\theta,X}, P_X)]$.

Suppose there is a parameter θ^* such that $\Phi_{Q_{\theta^*,x}} = \Phi_{P_x}$ on $\mathcal{Z}(x)$ and κ_x-almost everywhere, for almost every x. Then $\mathcal{L}_\lambda(\theta^*) = 0$ for every $\lambda \ge 0$, so the additional field term preserves this population minimizer while adding local curvature. Write $z_{P,x}^*$, $S_{P,x}$, $C_{P,x}$, and $J_{\theta^*,x}(z) = \partial_\theta\Phi_{Q_{\theta,x}}(z)\big|_{\theta=\theta^*}$ for the true solution, its sensitivity operator and active constraint matrix, and the field Jacobian. Assume the regularity conditions of Proposition 1 hold for almost every x, and that $\mathcal{L}_\lambda$ is twice differentiable at θ^*, with differentiation and expectation over X interchangeable.

Proposition 4 (Local curvature of the functional objective)**.** *At θ^*,*

$$\nabla_\theta^2\mathcal{L}_\lambda(\theta^*) = \mathbb{E}_X\left[J_{\theta^*,X}(z_{P,X}^*)^\top S_{P,X} J_{\theta^*,X}(z_{P,X}^*)\right] + \frac{\lambda}{d}\,\mathbb{E}_{X,T}\left[J_{\theta^*,X}(T)^\top J_{\theta^*,X}(T)\right]. \tag{9}$$

Moreover, if $S_{P,X} J_{\theta^*,X}(z^*_{P,X})v = 0$ *almost surely, then* $v^\top \nabla^2_\theta \mathcal{L}_\lambda(\theta^*)v = \frac{\lambda}{d}\,\mathbb{E}_{X,T}[\|J_{\theta^*,X}(T)v\|^2]$. *If* $\Gamma_\kappa = \mathbb{E}_{X,T}[J_{\theta^*,X}(T)^\top J_{\theta^*,X}(T)]$ *is positive definite, then* $\nabla^2_\theta \mathcal{L}_\lambda(\theta^*) \succeq \frac{\lambda}{d}\lambda_{\min}(\Gamma_\kappa) I$, *and* θ^* *is a strict local minimizer of* $\mathcal{L}_\lambda$ *for every* $\lambda > 0$.

The first term in Eq. 9 is the local regret geometry from Proposition 1, expressed in parameter space through the field Jacobian and averaged over contexts. For each context it has rank at most $\dim \ker C_{P,x}$, so the regret objective can be locally flat along parameter directions that do not move the current solution. The second term adds curvature along directions that change the field at the probes, including directions outside the local regret geometry. Under misspecification, the two terms need not share a minimizer, and λ controls their trade-off. The proof is provided in Appendix A.6.

6 Experiments

We evaluate Functional-DFL, which uses conditional Gaussian-mixture predictors with directly evaluable fields, on two synthetic and two real-world stochastic optimization problems against the following baselines, reporting mean and standard deviation over five seeds.

- **Two-stage** trains a point predictor by squared error and plugs it into the problem.
- **Gaussian Two-stage** trains a Gaussian predictive distribution by likelihood.
- **Point-DFL** trains a point predictor through a differentiable optimization layer (Agrawal et al., 2019), using the prediction in place of the uncertain outcome.
- **DF**^{2} (Kong et al., 2025) learns the expected objective over context and decision and optimizes the learned surface.
- **Diffusion-DFL** (Zhao et al., 2026b) learns a conditional diffusion model of the outcome and trains it end-to-end through the sample-average problem over its scenarios.

6.1 Stochastic Optimization Problems

We report test regret, using normalized regret in percent for the two portfolio tasks, and realized cost for Power Scheduling as in Diffusion-DFL. Regret is measured against a hindsight oracle for Allocation and Mean–Variance Portfolio and under the known conditional law for Mean–CVaR Portfolio. Task-specific fields and experimental protocols are given in Appendices B and C.

Allocation. Adapting the synthetic problem of Diffusion-DFL (Zhao et al., 2026b), the outcome $Y \in \mathbb{R}^{10}$ has a bimodal law and the decision is a bounded allocation solving

$$\min_{0 \le z \le C\mathbf{1}} \ \mathbb{E}[\exp(-Y^\top z) \mid X = x].$$

Its field $\mathbb{E}[-Y \exp(-Y^\top z) \mid X = x]$ is a nonlinear expectation, so the conditional mean alone does not determine the decision.

Mean–CVaR Portfolio. Five contextual features generate the returns of 20 assets through the nonlinear generator of PyEPO (Tang & Khalil, 2024), whose conditional law is known, and the decision is a long-only portfolio solving

$$\min_{w \ge 0,\ \mathbf{1}^\top w = 1} \ -(1-\beta)\,\mathbb{E}[Y^\top w \mid X = x] + \beta\ \mathrm{CVaR}_{0.95}(-Y^\top w \mid X = x),$$

with $\beta = 0.5$. The field depends on the mean return, the tail probability of the portfolio loss, and the return vector averaged over the tail.

Power Scheduling. In the PJM task of Donti et al. (2017), the context holds the previous day's load and temperature, the forecast temperature, and calendar features, and the outcome is the next-day demand $Y = (Y_1, \dots, Y_H)$ over $H = 24$ hours. A schedule pays $\gamma_{\text{under}} = 50$ per unit of unmet demand and $\gamma_{\text{over}} = 0.5$ per unit of excess under a ramp limit,

$$\min_z \ \mathbb{E}\Big[\sum_{h=1}^{H}\big(\gamma_{\text{under}}(Y_h - z_h)_+ + \gamma_{\text{over}}(z_h - Y_h)_+ + \tfrac{1}{2}(z_h - Y_h)^2\big) \ \Big|\ X = x\Big] \ \text{ s.t. } \ |z_{h+1} - z_h| \le c_{\text{ramp}}.$$

The field depends on the marginal means and distribution functions, while the ramp constraints couple the decisions across hours.

Table 1: Decision quality over five seeds. Lower is better, best in bold.

	Synthetic		Real-World	
Method	Allocation	CVaR Portfolio	Power Scheduling	MV Portfolio
Two-stage	2.69 ±1.64	202.81 ±17.67	33.97 ±3.07	92.8 ±1.1
Gaussian Two-stage	0.827 ±0.055	84.61 ±4.01	4.25 ±0.05	87.8 ±3.9
Point-DFL	0.950 ±0.069	64.00 ±3.79	5.77 ±4.16	88.6 ±9.0
DF^2	0.803 ±0.629	27.03 ±1.92	8.57 ±0.14	90.7 ±5.2
Diffusion-DFL	0.372 ±0.052	29.99 ±2.20	3.59 ±0.21	87.8 ±6.3
Functional-DFL (Ours)	**0.325 ±0.005**	**23.61 ±0.11**	**3.27 ±0.09**	**82.7 ±3.7**

Table 2: Training time (s).

Task	**DF^2**	**Diff.-DFL**	**Ours**
Allocation	17.8 ±5.2	519 ±147	**5.7 ±3.8**
CVaR	91.6 ±29.6	447 ±124	**74.0 ±5.1**
Power	328 ±62	1106 ±432	**47.5 ±5.7**
MV	42.7 ±18.5	6030 ±3216	**42.1 ±4.1**

Table 3: Ablation of the training signal.

Task	**Regret only**	**+ NLL**	**+ Field (Ours)**
Allocation	**0.325 ±0.002**	0.772 ±0.271	**0.325 ±0.005**
CVaR	24.51 ±0.08	**23.06 ±0.27**	23.61 ±0.11
Power	**3.26 ±0.11**	3.29 ±0.11	3.27 ±0.09
MV	88.1 ±4.2	90.1 ±2.2	**82.7 ±3.7**

Mean–Variance Portfolio. Following the S&P 100 protocol of prior decision-focused portfolio work (Lee et al., 2026), the context is the past 63 daily returns of the 93 constituents with complete data, the outcome is the next 21-day return window, and the decision is a long-only portfolio solving

$$\min_{w \geq 0,\ \mathbf{1}^\top w = 1} \ \frac{\rho}{2} w^\top \Sigma(x) w - \mu(x)^\top w,$$

where $\mu(x)$ and $\Sigma(x)$ are the predicted mean and expected covariance of the window and $\rho = 2$, so the field is determined by these two statistics.

6.2 Main Results

Functional-DFL has the lowest mean decision loss on all four tasks in Table 1. The point-based methods fall behind wherever the decision depends on more than the mean. On Mean–CVaR, whose objective depends on the lower tail, Two-stage and Point-DFL are far from every distributional method. Allocation is constructed so that the context carries no information and the decision depends on the bimodal law through a nonlinear expectation, and there Point-DFL is worse than Gaussian Two-stage, although the latter is trained without any decision signal. Training a point predictor for decisions does not recover what a single outcome value cannot represent. The distributional baselines trade places across tasks, with DF^2 strongest on Mean–CVaR, Diffusion-DFL on Allocation and Power Scheduling, and Gaussian Two-stage competitive on Mean–Variance Portfolio, whereas Functional-DFL is best on each task and, on Allocation and Mean–CVaR, also varies least across seeds, with a standard deviation an order of magnitude below that of Diffusion-DFL, whose gradient resamples scenarios at every step while ours evaluates the field exactly. Its margin over the strongest baseline is largest on Allocation and Mean–CVaR, where the field is a nonlinear expectation or a tail quantity, and smallest on Mean–Variance Portfolio, where the field is affine in the decision.

Table 2 shows that this structure is inexpensive to exploit. With one downstream solve per context and no solves at the probes, Functional-DFL is comparable to DF^2 on Mean–Variance Portfolio, faster on the other three tasks, and substantially faster than Diffusion-DFL, which solves the sample-average problem over its scenarios at every step.

6.3 Effect of Additional Field Supervision

We train the same predictor and solver under three objectives. Regret only uses the exact regret gradient of Theorem 2, + NLL replaces the field term with joint negative log-likelihood at the same weight, and + Field is Functional-DFL.

Regret only is already competitive with the strongest baselines in Table 1, as Theorem 2 suggests: its gradient matches the field at the induced decision, so the field is learned even without the field term. Adding the field term improves Mean–CVaR and Mean–Variance Portfolio and matches Regret only

on the other two tasks, which is the behavior Proposition 3 describes, since the term keeps the field matched, and regret bounded, at decisions the regret gradient has not visited. Appendix C.6 confirms this mechanism: at the largest perturbation from the induced decision, the held-out field error of Functional-DFL is lower than that of Regret only on every task, by a factor of 2.6 on Power Scheduling. The likelihood variant behaves differently. + NLL is best on Mean–CVaR, whose returns are conditionally Gaussian and close to the mixture class, but more than doubles the loss on Allocation, whose law has 2^{10} modes that eight components cannot represent, and is worse on Mean–Variance Portfolio. The likelihood pulls the model toward the full distribution whether or not it is representable, and under misspecification this pull can work against the regret gradient, whereas the field term targets the object the regret gradient already uses and its guarantee does not require a well-specified model. The improvement on these two tasks holds at every field weight and probe count in the sweeps of Appendix D.1.

6.4 Comparison with Differentiable Optimization Layers

Functional-DFL and Mixture DFL in Figure 3 share the Gaussian-mixture backbone and differ in how the regret gradient is computed, the field term leaving Power Scheduling unchanged in Table 3. Mixture DFL differentiates through the stochastic optimization layer of Donti et al. (2017), an SQP built on a differentiable quadratic program layer (Amos & Kolter, 2017). By Theorem 2, the gradient this layer targets and the gradient of the regret term in Eq. 8 are the same mathematical object, but their numerical computation is not. The layer differentiates through the approximate solution its own solver returns, so the gradient is only as accurate as that solution and inherits the active-set misidentification and the ill-conditioning under weakly active constraints documented for such layers (Magoon et al., 2025; Lee et al., 2026), whereas the regret term applies one KKT sensitivity at the solution.

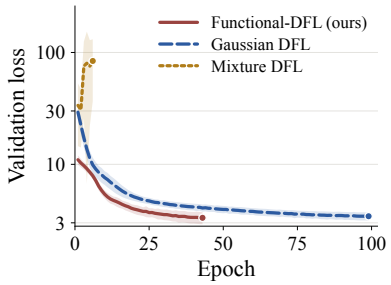


Figure 3: Validation loss on Power Scheduling over five seeds. Mixture DFL shares the backbone of Functional-DFL and the layer of Gaussian DFL.

Power Scheduling makes this difference visible. Gaussian DFL, which is the original setting of Donti et al. (2017) with a single Gaussian predictor, trains without difficulty and reaches a validation loss comparable to Functional-DFL after more than twice as many epochs. With the mixture backbone under otherwise identical settings, the same SQP fails to converge on 99.9% of training instances, and the validation loss diverges within a few epochs. As Appendix C.7 shows, the undamped SQP converges for unimodal Gaussians even at 0.3 times the residual scale, whereas the well-separated components of the mixture leave near-flat regions of the expected cost where Newton steps overshoot. The expressive predictive distribution therefore breaks the layer, which ties the gradient to one solver, whereas Functional-DFL needs only a forward solution and its local sensitivity.

7 Conclusion

We studied which aspects of a predictive distribution matter to a stochastic optimization problem. The optimality field identifies the distributional functionals entering stochastic optimality, while constrained solution sensitivity determines how errors in this field affect decisions and regret. This yields an exact field representation of the regret gradient at the induced decision and carries the structure of point-based DFL to predictive distributions. Functional-DFL retains this gradient and uses the same observation to supervise the field at additional decisions without additional downstream solves. Across two synthetic and two real-world tasks, it achieves the lowest mean decision loss among the compared baselines at low computational cost, while the benefit of additional field supervision is task dependent. More broadly, which part of a distribution a decision needs depends on the objective, suggesting that learning the distributional functionals used by downstream optimality can be more direct than fitting the full predictive distribution.

AI USE STATEMENT

In this work, we used generative AI tools to understand and debug the public implementations of prior methods used as baselines, and to obtain feedback on the experimental code. Additionally, we used generative AI tools to edit the manuscript for readability. We have reviewed all AI-assisted work, and all code was reviewed and tested by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in Neural Information Processing Systems*, 32, 2019.

Brandon Amos and J Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pp. 136–145. PMLR, 2017.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. *Advances in Neural Information Processing Systems*, 33:9508–9519, 2020.

J. Frédéric Bonnans and Alexander Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2000. doi: 10.1007/978-1-4612-1394-9.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5484–5494, 2017.

Jitka Dupačová. Stability and sensitivity-analysis for stochastic programming. *Annals of Operations Research*, 27(1):115–142, 1990.

Adam N Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68 (1):9–26, 2022.

Adam N Elmachtoub, Henry Lam, Haofeng Zhang, and Yunfan Zhao. Estimate-then-optimize versus integrated-estimation-optimization versus sample average approximation: a stochastic dominance perspective. *arXiv preprint arXiv:2304.06833*, 2023.

Anthony V Fiacco. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming*, 10(1):287–311, 1976.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Lingkai Kong, Wenhao Mu, Jiaming Cui, Yuchen Zhuang, B. Aditya Prakash, Bo Dai, and Chao Zhang. DF^2: Distribution-free decision-focused learning. In *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, pp. 2269–2290. PMLR, 2025.

Junhyeong Lee, Sangjin Jin, and Yongjae Lee. Decision-focused learning via tangent-space projection of prediction error. In *International Conference on Machine Learning*, 2026.

Suhan Liu and Mo Liu. Decision-focused optimal transport. *arXiv preprint arXiv:2602.02800*, 2026.

Connor Magoon, Fengyu Yang, Noam Aigerman, and Shahar Kovalsky. Differentiation through black-box quadratic programming solvers. In *Advances in Neural Information Processing Systems*, volume 38, pp. 108486–108517, 2025.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. doi: 10.1613/jair.1.15320.

Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2020.

Meng Qi, Paul Grigas, and Zuo-Jun Shen. Integrated conditional estimation-optimization. *Operations Research*, 74(3):1604–1625, 2026.

Stephen M Robinson. Strongly regular generalized equations. *Mathematics of Operations Research*, 5(1):43–62, 1980.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *The Journal of Risk*, 2(3):21–41, 2000. doi: 10.21314/JOR.2000.038.

Werner Römisch and Rüdiger Schultz. Distribution sensitivity in stochastic programming. *Mathematical Programming*, 50(1–3):197–226, 1991.

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, 2025.

Alexander Shapiro. On differential stability in stochastic programming. *Mathematical Programming*, 47(1–3):107–116, 1990.

Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. doi: 10.1007/s12532-020-00179-2.

Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *Mathematical Programming Computation*, 16(3):297–335, 2024. doi: 10.1007/s12532-024-00255-x.

Prince Zizhuang Wang, Shuyi Chen, Jinhao Liang, Ferdinando Fioretto, and Shixiang Zhu. Gen-DFL: Decision-focused generative learning for robust decision making. In *International Conference on Learning Representations*, 2026.

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, pp. 1658–1665, 2019.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023. doi: 10.1609/aaai.v37i9.26317.

Zihao Zhao, Kai-Chia Mo, Shing-Hei Ho, Brandon Amos, and Kai Wang. A fully first-order layer for differentiable optimization. In *International Conference on Machine Learning*, 2026a.

Zihao Zhao, Christopher Yeh, Lingkai Kong, and Kai Wang. Diffusion-DFL: Decision-focused diffusion models for stochastic optimization. In *International Conference on Learning Representations*, 2026b.

A PROOFS

A.1 PROOF OF PROPOSITION 1

Proof. Let $z_t = z^*_{Q_t}$ for $Q_t = Q + t\nu + o(t)$. Under LICQ, strict complementarity, and positive definiteness of the reduced Hessian, the implicit function theorem applied to the KKT system of the active constraints gives a differentiable solution path $t \mapsto (z_t, \eta_t)$ along which the active set is fixed, so differentiating the KKT system at $t = 0$ gives

$$\begin{bmatrix} H_Q & C_Q^\top \\ C_Q & 0 \end{bmatrix} \begin{bmatrix} Dz^*_Q[\nu] \\ D\eta_Q[\nu] \end{bmatrix} = - \begin{bmatrix} \mathcal{B}_Q \nu \\ 0 \end{bmatrix}.$$

Since $Dz^*_Q[\nu] \in \ker C_Q$, write $Dz^*_Q[\nu] = U_Q a$. Projecting the first block row onto $\ker C_Q$ gives

$$(U_Q^\top H_Q U_Q)a = -U_Q^\top \mathcal{B}_Q \nu,$$

and hence

$$Dz^*_Q[\nu] = -U_Q(U_Q^\top H_Q U_Q)^{-1} U_Q^\top \mathcal{B}_Q \nu = -S_Q \mathcal{B}_Q \nu.$$

Because F_P is fixed with respect to Q,

$$DR_P(Q)[\nu] = \nabla F_P(z^*_Q)^\top Dz^*_Q[\nu].$$

Predictive stationarity gives $\nabla F_Q(z^*_Q) = -C_Q^\top \eta_Q$, so $\nabla F_Q(z^*_Q)^\top Dz^*_Q[\nu] = 0$. Therefore

$$DR_P(Q)[\nu] = -\Delta_{Q,P}^\top Dz^*_Q[\nu] = \Delta_{Q,P}^\top S_Q \mathcal{B}_Q \nu.$$

At $Q = P$, $\Delta_{P,P} = 0$ and the first-order term vanishes. Let $b = \mathcal{B}_P \nu$ and $d = -S_P b$, so that $z^*_{P+t\nu} = z^*_P + td + o(t)$. Since the active face is locally unchanged, $C_P(z^*_{P+t\nu} - z^*_P) = 0$, and stationarity gives $\nabla F_P(z^*_P)^\top (z^*_{P+t\nu} - z^*_P) = -\eta_P^\top C_P(z^*_{P+t\nu} - z^*_P) = 0$. A second-order expansion of F_P around z^*_P therefore gives

$$R_P(P + t\nu) = \frac{t^2}{2} d^\top H_P d + o(t^2).$$

Finally, $S_P H_P S_P = S_P$, so $d^\top H_P d = b^\top S_P b$, proving the expansion. □

A.2 PROOF OF THE STRONG-CONVEXITY FIELD BOUND IN SECTION 3.3

Proof. Let $d = z^*_Q - z^*_P$ and $\varepsilon = \varepsilon_{Q,P}(z^*_Q)$. Optimality over the convex feasible set gives

$$\nabla F_Q(z^*_Q)^\top d \le 0, \qquad \nabla F_P(z^*_P)^\top d \ge 0.$$

Strong convexity of F_P implies

$$\big(\nabla F_P(z^*_Q) - \nabla F_P(z^*_P)\big)^\top d \ge m\|d\|^2.$$

Using $\nabla F_P(z^*_Q) = \nabla F_Q(z^*_Q) - \varepsilon$,

$$m\|d\|^2 \le -\varepsilon^\top d \le \|\varepsilon\|\,\|d\|,$$

which gives

$$\|z_Q^* - z_P^*\| \le \frac{\|\varepsilon_{Q,P}(z_Q^*)\|}{m}.$$

Strong convexity along the segment from z_Q^* to z_P^* gives $F_P(z_P^*) \ge F_P(z_Q^*) - \nabla F_P(z_Q^*)^\top d + \frac{m}{2}\|d\|^2$, and with $\nabla F_P(z_Q^*)^\top d = \nabla F_Q(z_Q^*)^\top d - \varepsilon^\top d \le -\varepsilon^\top d$ this yields

$$R_P(Q) \le -\varepsilon^\top d - \frac{m}{2}\|d\|^2 \le \|\varepsilon\|\,\|d\| - \frac{m}{2}\|d\|^2.$$

Maximizing the last quadratic over $\|d\| \ge 0$ yields $R_P(Q) \le \|\varepsilon\|^2/(2m)$. □

A.3 Proof of Theorem 2

Proof. Because $\bar{z}_\theta$ and $\bar{S}_\theta$ are held fixed in ℓ_{reg}, its gradient at the current parameter value is

$$\nabla_\theta \ell_{\mathrm{reg}}(\theta; y) = J_\theta(\bar{z}_\theta)^\top \bar{S}_\theta e_\theta(y, \bar{z}_\theta) = J_\theta(z_\theta^*)^\top S_\theta e_\theta(y, z_\theta^*).$$

From Proposition 1, $D_\theta z_\theta^* = -S_\theta J_\theta(z_\theta^*)$, so

$$\nabla_\theta f(y, z_\theta^*) = -J_\theta(z_\theta^*)^\top S_\theta\big(\nabla r(z_\theta^*) + \psi(y, z_\theta^*)\big).$$

Predictive stationarity gives $\nabla r(z_\theta^*) + \Phi_{Q_\theta}(z_\theta^*) + C_\theta^\top \eta_\theta = 0$. Since $S_\theta C_\theta^\top = 0$,

$$\nabla_\theta f(y, z_\theta^*) = J_\theta(z_\theta^*)^\top S_\theta\big(\Phi_{Q_\theta}(z_\theta^*) - \psi(y, z_\theta^*)\big) = \nabla_\theta \ell_{\mathrm{reg}}(\theta; y).$$

Taking expectation and using $\mathbb{E}_P[e_\theta(Y, z_\theta^*)] = \Phi_{Q_\theta}(z_\theta^*) - \Phi_P(z_\theta^*)$ gives the expected update. When differentiation and expectation may be interchanged, this is also $\nabla_\theta R_P(Q_\theta)$, since $F_P(z_P^*)$ is independent of θ. □

A.4 Proof of the field-matching identity in Section 5.1

Proof. Fix a context x and condition on $T = t$, with $P = P(\cdot \mid x)$ throughout this proof. The conditional independence of T and Y given x lets us write

$$e_\theta(Y, t) = \big(\Phi_{Q_\theta}(t) - \Phi_P(t)\big) + \big(\Phi_P(t) - \psi(Y, t)\big).$$

The second term has zero mean under P, so

$$\mathbb{E}_P\|e_\theta(Y, t)\|^2 = \|\Phi_{Q_\theta}(t) - \Phi_P(t)\|^2 + \operatorname{tr}\operatorname{Cov}_P(\psi(Y, t)).$$

Averaging over $T \sim \kappa$ and multiplying by $1/(2d)$ gives Eq. 7, and the covariance term does not depend on θ. □

A.5 Proof of Proposition 3

Proof. Write $\varepsilon = \varepsilon_{Q,P}$ and $z^* = z_Q^*$. By the strong-convexity bound in Section 3.3, it suffices to bound $\|\varepsilon(z^*)\|^2$, since $R_P(Q) \le \|\varepsilon(z^*)\|^2/(2m)$. The proof of that bound uses strong convexity of F_P only on the segment between z_Q^* and z_P^*, so it suffices that F_P be m-strongly convex on a convex set containing both decisions.

For part (i), let $B = \{t \in \mathcal{Z} : \|t - z^*\| \le \tau\}$. Since $z^* \in \mathcal{Z}_0$, $\kappa(B) \ge \kappa_\tau$. For every $t \in B$, the Lipschitz condition gives $\|\varepsilon(z^*)\| \le \|\varepsilon(t)\| + L\tau$, and hence $\|\varepsilon(z^*)\|^2 \le 2\|\varepsilon(t)\|^2 + 2L^2\tau^2$. Integrating over B with respect to κ,

$$\kappa(B)\,\|\varepsilon(z^*)\|^2 \le 2\int_B \|\varepsilon(t)\|^2\, d\kappa(t) + 2L^2\tau^2\kappa(B) \le 2D_\kappa^2(Q, P) + 2L^2\tau^2\kappa(B).$$

Dividing by $\kappa(B) \geq \kappa_\tau$ gives $\|\varepsilon(z^*)\|^2 \leq 2D_\kappa^2(Q,P)/\kappa_\tau + 2L^2\tau^2$, which proves part (i).

For part (ii), write $z^* = \sum_k \alpha_k t_k$ with $\alpha_k \geq 0$ and $\sum_k \alpha_k = 1$. Since ε is affine on $\mathcal{Z}$, $\varepsilon(z^*) = \sum_k \alpha_k \varepsilon(t_k)$, and convexity of the squared norm gives

$$\|\varepsilon(z^*)\|^2 \leq \sum_k \alpha_k \|\varepsilon(t_k)\|^2 \leq \max_k \|\varepsilon(t_k)\|^2 \leq \frac{1}{\pi_{\min}} \sum_k \kappa(\{t_k\}) \|\varepsilon(t_k)\|^2 \leq \frac{D_\kappa^2(Q,P)}{\pi_{\min}}.$$

This proves part (ii). □

The argument of part (i) also applies coordinatewise. Suppose the field is separable, $\varepsilon(z) = (\varepsilon_1(z_1), \ldots, \varepsilon_d(z_d))$, with each ε_h being L_h-Lipschitz on $\mathbb{R}$, and suppose $\kappa = \bigotimes_h \kappa_h$ with $\kappa_h([z_h - \tau_h, z_h + \tau_h]) \geq \kappa_\tau$ for every $z \in \mathcal{Z}_0$ and every h. Then $D_\kappa^2(Q,P) = \sum_h \mathbb{E}_{T_h \sim \kappa_h}[\varepsilon_h(T_h)^2]$, and applying the scalar argument to each coordinate and summing gives

$$R_P(Q) \leq \frac{1}{m} \left(\frac{D_\kappa^2(Q,P)}{\kappa_\tau} + \sum_{h=1}^{d} L_h^2 \tau_h^2 \right).$$

Here the probes need not satisfy the coupling constraints of $\mathcal{Z}$, since each coordinate of the field is defined on $\mathbb{R}$. For Power Scheduling, constant schedules are feasible at every level, so equality of the predictive and true fields on $\mathcal{Z}$ implies equality of the separable fields on $\mathbb{R}^{24}$, and Proposition 4 is unaffected by probes outside $\mathcal{Z}$.

We now record the conditions in each task. Throughout, predictive component covariances are positive definite, as in Appendix C.

For Allocation, the true law of Y is a mixture of 2^{10} nondegenerate Gaussians, so $\nabla^2 F_P(z) = \mathbb{E}_P[YY^\top e^{-Y^\top z}]$ is positive definite and continuous in z, and F_P is strongly convex on the compact box. The fields have bounded Jacobians on the box, so ε is Lipschitz there, and part (i) holds with $\mathcal{Z}_0 = \mathcal{Z}$, $\kappa_\tau = 1/32$, and τ equal to the fill distance of the 32 probe points in $[0,2]^{10}$. With 32 probes in ten dimensions this fill distance is comparable to the side of the box, so for Allocation the bound is qualitative, and its discretization term decreases only as the probes become denser or as the field error becomes flatter.

For Power Scheduling, the quadratic term makes F_P strongly convex with $m = 1$. The field is separable, and $\varepsilon_h(z_h) = \mathbb{E}_P[Y_h] - \mathbb{E}_Q[Y_h] + (\gamma_{\text{under}} + \gamma_{\text{over}})(F_{Q,h}(z_h) - F_{P,h}(z_h))$, which is Lipschitz with $L_h = (\gamma_{\text{under}} + \gamma_{\text{over}}) \sup_s |f_{Q,h}(s) - f_{P,h}(s)|$ whenever the marginal densities $f_{Q,h}$ and $f_{P,h}$ are bounded. The coordinatewise form holds with $\kappa_\tau = 1/21$, τ_h equal to half the grid spacing of hour h, and $\mathcal{Z}_0$ the schedules whose coordinates lie within the threshold grids.

For Mean–Variance Portfolio, $\widehat{\Sigma}(Y) \succeq \epsilon I$ by construction, so F_P is strongly convex with $m = \rho\epsilon$. Both Φ_Q and Φ_P are affine in w, the probes contain the simplex vertices, and part (ii) holds with $\pi_{\min} = 1/94$.

For Mean–CVaR Portfolio, the curvature of the expected hinge vanishes as the auxiliary threshold η moves away from the portfolio loss distribution, so F_P is not strongly convex on the whole feasible set. By the regularity argument in Appendix B and continuity of the Hessian, F_P is strongly convex on every compact set $\Delta \times [\eta_-, \eta_+]$, where Δ is the simplex. The proof of the strong-convexity bound uses strong convexity only on the segment between z_Q^* and z_P^*, so part (i) applies with a range-dependent modulus when both decisions lie in such a set. The fields are Lipschitz there, since $w^\top \Sigma w$ is bounded away from zero on Δ for every positive definite component covariance, and the probes give $\kappa_\tau = 1/32$ with τ equal to their fill distance in $\Delta \times [-0.2, 0]$. As for Allocation, the resulting bound is qualitative at this probe density.

A.6 Proof of Proposition 4

Proof. For fixed x, define

$$\Delta_\theta = \Phi_{Q_{\theta,x}}(z^*_{\theta,x}) - \Phi_{P_x}(z^*_{\theta,x}).$$

By Theorem 2,

$$\nabla_\theta R_{P_x}(Q_{\theta,x}) = J_{\theta,x}(z^*_{\theta,x})^\top S_{\theta,x}\Delta_\theta.$$

At θ^*, the assumed field equality gives $\Delta_{\theta^*} = 0$, and since $F_{Q_{\theta^*,x}} - F_{P_x}$ is then constant on the feasible set, $z^*_{\theta^*,x} = z^*_{P,x}$. Hence all product-rule terms containing derivatives of $J_{\theta,x}$ or $S_{\theta,x}$ are multiplied by $\Delta_{\theta^*} = 0$ and vanish. The remaining term is $D_\theta\Delta_\theta = J_{\theta,x}(z^*_{\theta,x}) + \big(\nabla_z\Phi_{Q_{\theta,x}} - \nabla_z\Phi_{P_x}\big)(z^*_{\theta,x})\, D_\theta z^*_{\theta,x}$. At θ^* the second part vanishes, because $D_\theta z^*_{\theta,x}$ lies in $\ker C_{P,x}$, strict complementarity keeps small moves in these directions inside the feasible set, and $\Phi_{Q_{\theta^*,x}} - \Phi_{P_x}$ is identically zero there. Thus

$$\nabla^2_\theta R_{P_x}(Q_{\theta^*,x}) = J_{\theta^*,x}(z^*_{P,x})^\top S_{P,x} J_{\theta^*,x}(z^*_{P,x}).$$

For $g_\theta(t) = \Phi_{Q_{\theta,x}}(t) - \Phi_{P_x}(t)$, $g_{\theta^*}(t) = 0$, so

$$\nabla^2_\theta \frac{\lambda}{2d}\|g_\theta(t)\|^2\Big|_{\theta=\theta^*} = \frac{\lambda}{d} J_{\theta^*,x}(t)^\top J_{\theta^*,x}(t).$$

Taking expectations gives Eq. 9. If $S_{P,X}J_{\theta^*,X}(z^*_{P,X})v = 0$ almost surely, the regret-curvature term vanishes along v, leaving $\frac{\lambda}{d}\,\mathbb{E}_{X,T}\|J_{\theta^*,X}(T)v\|^2$.

Since the reduced Hessian is positive definite, $S_{P,x} = U(U^\top H_P U)^{-1}U^\top$ is positive semidefinite, so the first term of Eq. 9 is positive semidefinite and $\nabla^2_\theta\mathcal{L}_\lambda(\theta^*) \succeq \frac{\lambda}{d}\Gamma_\kappa \succeq \frac{\lambda}{d}\lambda_{\min}(\Gamma_\kappa)I$. Because $\mathcal{L}_\lambda \geq 0 = \mathcal{L}_\lambda(\theta^*)$, θ^* is a minimizer and $\nabla_\theta\mathcal{L}_\lambda(\theta^*) = 0$. A positive definite Hessian then makes θ^* a strict local minimizer by the second-order sufficient condition. Finally, for each context, $J_{\theta^*,x}(z^*_{P,x})^\top S_{P,x}J_{\theta^*,x}(z^*_{P,x})$ has rank at most $\operatorname{rank} S_{P,x} = \dim\ker C_{P,x}$. □

B Task-Specific Instantiations

We record the realized-cost gradients used as field targets and the corresponding predictive fields. Allocation adapts the synthetic problem of Diffusion-DFL (Zhao et al., 2026b), and Power Scheduling follows the task and data protocol of Donti et al. (2017). The synthetic return generator for Mean–CVaR Portfolio follows PyEPO (Tang & Khalil, 2024), and the long-only mean–variance setting follows prior decision-focused portfolio work (Lee et al., 2026). The expressions below follow from the objectives in Section 6.1.

Allocation. For $\phi(y,z) = \exp(-y^\top z)$ on $0 \leq z \leq C\mathbf{1}$,

$$\psi(y,z) = -ye^{-y^\top z}, \qquad \Phi_Q(z) = -\mathbb{E}_Q[Ye^{-Y^\top z}].$$

For a Gaussian mixture $Q = \sum_j \pi_j\mathcal{N}(\mu_j,\Sigma_j)$, this field can be evaluated without predictive sampling:

$$\Phi_Q(z) = \sum_j \pi_j \exp\big(-\mu_j^\top z + \tfrac{1}{2}z^\top\Sigma_j z\big)\,(\Sigma_j z - \mu_j).$$

The local sensitivity uses the free coordinates of the box-constrained solution.

Mean–CVaR Portfolio. For portfolio loss $L(y,w) = -y^\top w$, we use the auxiliary-variable representation of CVaR at level $\alpha = 0.95$ (Rockafellar & Uryasev, 2000). With augmented decision (w,η), the realized objective is

$$\phi(y,w,\eta) = -(1-\beta)y^\top w + \beta\left[\eta + \frac{(L(y,w)-\eta)_+}{1-\alpha}\right], \qquad w \geq 0, \quad \mathbf{1}^\top w = 1.$$

Away from the hinge, its gradient is

$$\psi(y, w, \eta) = \begin{bmatrix} -(1-\beta)y - \dfrac{\beta}{1-\alpha}\mathbf{1}\{-y^\top w > \eta\}y \\ \beta\left(1 - \dfrac{\mathbf{1}\{-y^\top w > \eta\}}{1-\alpha}\right) \end{bmatrix}.$$

Thus Φ_Q depends on the mean return, the tail probability of the portfolio loss, and the tail-weighted return vector $\mathbb{E}_Q[Y\,\mathbf{1}\{-w^\top Y > \eta\}]$. Under a Gaussian-mixture component, $L = -w^\top Y$ is normal with mean $m_j = -w^\top \mu_j$ and variance $s_j^2 = w^\top \Sigma_j w$. Writing $a_j = (m_j - \eta)/s_j$, its expected hinge is

$$\mathbb{E}_j[(L-\eta)_+] = (m_j - \eta)\Phi_{\mathrm{N}}(a_j) + s_j \varphi_{\mathrm{N}}(a_j),$$

where Φ_{N} and φ_{N} are the standard normal CDF and density. Mixture-weighted sums of these terms give the objective, and differentiating them gives the field and local Hessian without predictive sampling. The stochastic decision and sensitivity include η as well as the portfolio weights.

Power Scheduling. For each time coordinate, the realized gradient of the asymmetric penalty and quadratic cost is

$$\psi_h(y, z) = z_h - y_h - \gamma_{\mathrm{under}}\mathbf{1}\{y_h > z_h\} + \gamma_{\mathrm{over}}\mathbf{1}\{y_h < z_h\},$$

and hence

$$[\Phi_Q(z)]_h = z_h - \mathbb{E}_Q[Y_h] - \gamma_{\mathrm{under}}\big(1 - F_{Q,h}(z_h)\big) + \gamma_{\mathrm{over}} F_{Q,h}(z_h).$$

Only the marginal mean and CDF at each hour are needed to evaluate this field, while ramp constraints couple hours in the downstream solution and its sensitivity. For a Gaussian mixture, the marginal CDF and density are analytic. Field supervision evaluates individual coordinates on hour-specific threshold grids. Since the objective is separable across hours before imposing the ramp constraints, each squared coordinate residual gives a sample-level squared loss for that field component.

Mean–Variance Portfolio. Here $Y = (Y_1, \ldots, Y_H)$ denotes a future return window and the history of length M is fixed by the context. Let $R_1, \ldots, R_{M+H}$ denote the concatenated history and future returns. The observed statistics are

$$\widehat{\mu}(Y) = \frac{1}{H}\sum_{h=1}^{H} Y_h, \qquad \widehat{\Sigma}(Y) = \frac{1}{M+H-1}\sum_{s=1}^{M+H}(R_s - \bar{R})(R_s - \bar{R})^\top + c(Y)I,$$

where $\bar{R}$ is the average of $R_1, \ldots, R_{M+H}$ and $c(Y) \geq 0$ is the smallest shift that makes the minimum eigenvalue of $\widehat{\Sigma}(Y)$ at least $\epsilon = 10^{-4}$. The realized cost is $\frac{\rho}{2} w^\top \widehat{\Sigma}(Y) w - \widehat{\mu}(Y)^\top w$, and therefore

$$\psi(Y, w) = \rho\,\widehat{\Sigma}(Y)w - \widehat{\mu}(Y), \qquad \Phi_Q(w) = \rho\,\mathbb{E}_Q[\widehat{\Sigma}(Y)]w - \mathbb{E}_Q[\widehat{\mu}(Y)].$$

The predictive statistics are expectations of these window statistics under the conditional model of future returns. In particular, the expected window covariance is distinct from the covariance of a single future return. Here $M + H = 84 < 93$, so the unregularized window covariance is rank deficient and the eigenvalue correction is exactly ϵI. Its expectation is therefore available from the mixture component means and variances. The decision is a long-only simplex portfolio, and its local sensitivity is obtained from the quadratic objective on the active face.

In each task, S_Q is obtained from the reduced Hessian and the active constraint Jacobian as defined in Section 3.1.

Regularity conditions. LICQ holds for the box constraints. It holds for the simplex, since at most $d-1$ nonnegativity constraints can be active together with $\mathbf{1}^\top w = 1$, and for the ramp constraints, since at most one constraint of each pair is active and the active rows $\pm(e_{h+1} - e_h)$ are linearly independent. Under a Gaussian mixture with positive definite component covariances, $\nabla^2 F_Q$ is positive definite for Allocation, satisfies $\nabla^2 F_Q \succeq I$ for Power Scheduling, and satisfies $\nabla^2 F_Q \succeq \rho\epsilon I$ for Mean–Variance Portfolio. For Mean–CVaR Portfolio, the mean term is linear. Along a direction (a, b) in (w, η), the second derivative of $\mathbb{E}_Q[(-w^\top Y - \eta)_+]$ equals the density of $-w^\top Y$ at η times $\mathbb{E}_Q[(a^\top Y + b)^2 \mid -w^\top Y = \eta]$. Under each component, the conditional law of Y given $-w^\top Y = \eta$ has full support on this hyperplane, so the second derivative vanishes only if (a, b) is proportional to (w, η). Feasible directions satisfy $\mathbf{1}^\top a = 0$ while $\mathbf{1}^\top w = 1$, so the reduced Hessian is positive definite. The same argument applies to the true Gaussian law of this task. Strict complementarity is assumed as in Section 3.1.

C Experimental Details

C.1 Data and Task Protocols

Allocation. We adapt the synthetic allocation problem of Diffusion-DFL (Zhao et al., 2026b). The context $X \in \mathbb{R}^{20}$ is standard Gaussian and carries no information about the ten-dimensional outcome in this instance. Independently over coordinates,

$$Y_i = \tfrac{1}{2}\big[V_i - 3(1 - V_i) + \varepsilon_i\big], \qquad V_i \sim \text{Bernoulli}(0.9), \quad \varepsilon_i \sim \mathcal{N}(0, 0.15^2).$$

The allocation is bounded by $0 \le z_i \le 2$. We generate 1,000 examples and split them into 640 training, 160 validation, and 200 test cases. Test regret is the mean realized exponential cost minus the mean of the best cost attainable with each outcome known.

Mean–CVaR Portfolio. We use the contextual return generator adapted from PyEPO (Tang & Khalil, 2024) with five features and twenty assets. For $B_{ij} \sim \text{Bernoulli}(1/2)$ and $\Lambda_{ij} \sim \text{Uniform}(-0.0075, 0.0075)$, the conditional mean and observed return are

$$\mu_i(x) = \left(\frac{0.05}{\sqrt{5}} B_i^\top x + 0.1^{1/8}\right)^8, \qquad Y = \mu(x) + \Lambda\xi + 0.03\zeta,$$

where $\xi \sim \mathcal{N}(0, I_5)$ and $\zeta \sim \mathcal{N}(0, I_{20})$ are independent. Thus the conditional covariance is $\Lambda\Lambda^\top + 0.03^2 I_{20}$. The data size and split are the same as for Allocation. Since the conditional law is known, each test decision and its stochastic oracle are evaluated under that law. Normalized regret divides the sum of population regrets by the sum of absolute oracle costs.

Power Scheduling. We follow the PJM protocol of the released code of Donti et al. (2017), which Diffusion-DFL (Zhao et al., 2026b) also uses, with load and temperature records for 2008–2011 and 2012–2016. Each context has 149 features, consisting of six 24-hour sequences formed from preceding-day demand and temperature and forecast-day temperature, followed by five calendar features. The outcome is the next 24-hour demand trajectory. We use $\gamma_{\text{under}} = 50$, $\gamma_{\text{over}} = 0.5$, and ramp limit $c_{\text{ramp}} = 0.4$. The final 20% of days are reserved for testing, and the earlier days are split into training and validation subsets at an 80/20 ratio for each seed.

Mean–Variance Portfolio. We use the long-only portfolio protocol of prior decision-focused work (Lee et al., 2026) with daily returns of the 93 S&P 100 constituents that have data over the full period from January 2010 to December 2024. Each context contains 63 past trading days and predicts a 21-day future window. The data are split chronologically into training, validation, and test periods with proportions $70/10/20$. The risk coefficient is $\rho = 2$. Test windows are evaluated at nonoverlapping rebalance dates. For each window, the oracle solves the same simplex-constrained quadratic program using the observed window statistics. Normalized regret is the sum of utility gaps divided by the sum of absolute oracle utilities.

C.2 Baselines

Two-stage and Point-DFL use the same task-specific backbones as Functional-DFL with a point-prediction output. Two-stage is trained with the squared prediction error, and its prediction is

then plugged into the downstream problem. Point-DFL plugs the point prediction into the downstream problem and differentiates the resulting decision through a differentiable convex optimization layer (Agrawal et al., 2019). For Mean–CVaR Portfolio, a point prediction is a single scenario, so the CVaR term reduces to the loss of that scenario. For Mean–Variance Portfolio, the predicted return path replaces the future window in the window statistics of Appendix B. Gaussian Two-stage uses the corresponding distributional backbone with one diagonal Gaussian component and is trained only by joint negative log-likelihood, with learning rate and checkpoint selected by validation likelihood. DF^2 is implemented following Kong et al. (2025), and Diffusion-DFL follows Zhao et al. (2026b) with task-specific adapters to our evaluation protocol. For Mean–Variance Portfolio, our adapter forms the risk term of Diffusion-DFL from a weighted combination of the historical covariance and the covariance of its generated returns, with weights 0.75 and 0.25, rather than the analytic expected-window statistics used by Functional-DFL. All baselines are rerun under the same data protocol and on our hardware, with model selection based only on validation data, so absolute numbers can differ from those reported in the original papers.

C.3 Functional-DFL Predictors

Conditional distribution. Our predictor outputs a Gaussian mixture with context-dependent weights, means, and covariance parameters,

$$Q_\theta(\cdot \mid x) = \sum_{j=1}^{J} \pi_{\theta,j}(x)\mathcal{N}\big(\mu_{\theta,j}(x), \mathrm{diag}(v_{\theta,j}(x)) + A_{\theta,j}(x)A_{\theta,j}(x)^\top\big)\,.$$

The predictor is a standard multi-head network. A backbone feeds linear heads that output the component weights, means, diagonal variances, and low-rank covariance factors for each context, with a softmax over the weights and a softplus parameterization with a small floor for the diagonal variances. No part of the architecture is specific to Functional-DFL, and the same predictor is used by every variant in Table 3. The field Φ_{Q_θ} is computed from the mixture parameters using the task-specific expressions in Appendix B, so the Functional-DFL update involves no predictive scenario sampling at any stage.

Task-specific backbones. The two synthetic tasks use two-hidden-layer MLPs with width 1024, SiLU activations, and layer normalization. Their mixture components have diagonal-plus-rank-two covariance. Power Scheduling uses a DLinear-style decomposition of six hourly feature channels with a calendar projection and a shared four-component mixture over the 24-hour outcome with diagonal component covariances. Mean–Variance Portfolio uses a DLinear backbone with reversible instance normalization (Zeng et al., 2023; Kim et al., 2022) to predict a joint four-component distribution over the 21-day return path. Its component covariance is diagonal over the path coordinates, and the component identity is shared across times and assets, allowing the expected window covariance in Appendix B to retain dependence induced by the mixture.

C.4 Training Settings

Table 4 gives the task-specific settings of Functional-DFL. J is the number of mixture components, rank is the rank of the low-rank covariance term, and K is the number of probes. One epoch of NLL warm-up precedes decision-focused training. We use Adam (Kingma & Ba, 2015) and select checkpoints by validation decision quality.

Table 4: Functional-DFL training settings. The last column gives the maximum number of decision-focused epochs and the early-stopping patience.

Task	J/rank	K	Adam LR	Batch	Warm-up	Epochs/patience
Allocation	$8/2$	32	10^{-5}	32	1	$100/5$
Mean–CVaR Portfolio	$4/2$	32	10^{-5}	32	1	$100/5$
Power Scheduling	$4/0$	21 per hour	10^{-3}	64	1	$100/5$
Mean–Variance Portfolio	$4/0$	94	10^{-3}	16	1	$100/5$

The field weight in Eq. 8 is $\lambda = 10$ for Allocation, $\lambda = 0.1$ for Power Scheduling, and $\lambda = 1$ for the two portfolio tasks. The normalization by Kd matches the implementation, which averages the

squared field residual over probes and field coordinates. In the + NLL ablation, the field term is replaced by joint negative log-likelihood normalized by outcome dimension, with the learning rate selected by validation decision quality. Gradient norms are clipped at one for Mean–CVaR Portfolio and are not clipped for the other tasks. All experiments run on a single NVIDIA RTX A5000 GPU with an Intel Xeon Silver 4210R CPU, with neural networks on the GPU and downstream solvers on the CPU.

C.5 Solvers and Field Probes

Allocation. A box-constrained Newton solver uses the analytic mixture objective, gradient, and Hessian, and its free-coordinate Hessian gives the local sensitivity. The additional field term uses 32 Sobol points in $[0, 2]^{10}$, including the origin.

Mean–CVaR Portfolio. The mixture mean–CVaR problem is solved in the portfolio weights and the auxiliary threshold by a feasible active-set Newton method, with a sequential quadratic programming solver as a fallback. We use 32 simplex portfolio probes paired with auxiliary thresholds between -0.2 and 0.

Power Scheduling. Each hourly decision is first solved without the ramp constraints, and schedules that violate them are re-solved with the constraints. The local sensitivity uses the active ramp constraints. The field term uses 21 thresholds per hour, evenly spaced from four training standard deviations below to six above the training mean of the demand at that hour.

Mean–Variance Portfolio. The simplex-constrained quadratic program with the predicted mean and expected window covariance is solved by a batched active-set method, with OSQP (Stellato et al., 2020) as a fallback, and its active face determines the sensitivity. Field probes are the 93 simplex vertices and the equal-weight portfolio. Across tasks, these probes are held fixed during decision-focused training.

C.6 Field Evaluation

We evaluate the held-out field error of Regret only and Functional-DFL on the test set around the decision induced by each trained model. For each test context and perturbation scale r, we draw eight random queries. For Allocation, a query adds r times a uniformly random unit direction to the induced decision and clips the result to $[0, 2]^{10}$. For Power Scheduling, each hourly coordinate is shifted by r training standard deviations of the demand at that hour, with independent random signs. The shifted schedules are not projected onto the ramp constraints, since the field is evaluated coordinatewise. For the two portfolio tasks, the portfolio query is $(1 - r)w + r\tilde{w}$, where w is the induced portfolio and $\tilde{w}$ is drawn uniformly from the simplex, so $r = 1$ gives a uniformly random long-only portfolio. For Mean–CVaR Portfolio, the auxiliary threshold is also shifted by $0.1r$ times a standard normal draw.

The field error E_{field} is the mean squared Euclidean distance between the predicted field Φ_{Q_θ} and a reference field over these queries. For the two synthetic tasks, the reference is the true field under the known data-generating law. For the two real-world tasks, the reference is the realized gradient $\psi(y, T)$ of the held-out outcome, whose expectation is the true field. On the real-world tasks the reported quantity is therefore a realized field-residual error, which includes the observation noise of the target, and on every task the queries are centered at the decision induced by each model, so the two models are evaluated at different locations. Table 5 reports the ratio of the two field errors, each averaged over the five seeds, at scales $r \in \{0, 0.25, 0.5, 1\}$, which are defined for every task.

C.7 Comparison with a Differentiable Optimization Layer

The layer baseline of Section 6.4 follows Donti et al. (2017). For each context, the stochastic problem induced by the Gaussian-mixture predictor is solved by sequential quadratic programming, where each iteration solves a quadratic program with the differentiable layer of Amos & Kolter (2017) and the gradient is taken through the final quadratic program. We keep the solver settings of Donti et al. (2017), including the undamped Newton steps, and use this SQP as the layer's forward solver throughout. Gaussian DFL in Figure 3 is this layer with the single-Gaussian predictor

Table 5: Ratio of the held-out field error of Regret only to that of Functional-DFL across perturbation scales. Values above one favor Functional-DFL. Queries are centered at each model's own induced decision, and the real-world tasks use the realized residual as reference.

Task	$r = 0$	$r = 0.25$	$r = 0.5$	$r = 1$
Allocation	1.47	1.44	1.38	1.29
Mean–CVaR Portfolio	1.26	1.14	1.13	1.07
Power Scheduling	0.92	0.73	1.03	2.57
Mean–Variance Portfolio	1.31	1.31	1.31	1.31

of Donti et al. (2017), and Mixture DFL replaces this predictor by the Gaussian-mixture backbone used by Functional-DFL.

To isolate why the SQP fails on Power Scheduling, we fix the data and the SQP and vary only the predictive distribution over ten training batches. Table 6 reports the median component standard deviation, the median distance between component means, the number of batches on which the SQP does not meet its convergence tolerance within the iteration cap of Donti et al. (2017), and the error of the returned solution, measured as the largest coordinate-wise deviation from the solution of our analytic solver for the same predictive distribution and reported as the median over the batches. A single Gaussian with the residual standard deviation, as in Donti et al. (2017), converges on every batch, and so does a single Gaussian whose scale is reduced to 0.3 times that value, so a narrow predictive distribution is not by itself a problem. Our mixture as predicted fails on every batch. Its components are about six standard deviations apart, so the density between them is close to zero and the Hessian of the expected cost drops to the quadratic term alone, and the resulting Newton steps are about 75 times the scale of the observations and oscillate between modes. Doubling the component standard deviations, which fills the gaps, or collapsing the component means to one point, which makes the mixture unimodal, restores convergence on every batch. These controlled variants indicate that the separated mixture components drive the SQP failure in this setting.

Table 6: Convergence of the layer's SQP on Power Scheduling under different predictive distributions, with the data and the solver fixed, over ten training batches.

Predictive distribution	**Comp. std**	**Mean gap**	**Failures**	**Solution error**
Single Gaussian, residual scale	0.190	–	0/10	2×10^{-14}
Single Gaussian, $0.3\times$ residual scale	0.057	–	0/10	7×10^{-13}
Our mixture as predicted	0.077	0.487	10/10	1.1
Our mixture, component std $\times 2$	0.154	0.487	0/10	8×10^{-14}
Our mixture, means collapsed	0.077	0	0/10	2×10^{-12}

D Additional Experiments

D.1 Ablation Studies

We vary the field weight λ, the number of probes K, and the number of mixture components J separately, keeping the remaining settings of Appendix C fixed. All tables use the test metrics of Table 1 and report mean $\pm$ standard deviation over five seeds. The settings of Table 1 are $\lambda = 10$ for Allocation, 0.1 for Power Scheduling, and 1 for the portfolio tasks, with $K = 32$, 32, 21, and 94 and $J = 8$, 4, 4, and 4 for Allocation, Mean–CVaR Portfolio, Power Scheduling, and Mean–Variance Portfolio in that order.

Field weight. Table 7 varies λ over the same grid for all four tasks with K fixed at its main-results value. These values were fixed before the sweeps and not tuned on any split. On Mean–CVaR Portfolio and Mean–Variance Portfolio, where the field term improves on Regret only in Table 3, the improvement holds at every value of λ in this grid.

Table 7: Decision quality across λ. Lower is better.

Task	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 10$
Allocation	0.398 ± 0.012	0.358 ± 0.031	0.353 ± 0.029	0.325 ± 0.005
Mean–CVaR Portfolio	24.42 ± 0.15	23.97 ± 0.14	23.61 ± 0.11	23.22 ± 0.07
Power Scheduling	3.27 ± 0.09	3.12 ± 0.04	3.06 ± 0.02	3.26 ± 0.05
Mean–Variance Portfolio	86.6 ± 7.0	86.5 ± 2.1	82.7 ± 3.7	86.4 ± 6.8

Number of probes. Table 8 varies K with λ fixed. For Mean–Variance Portfolio, the field is affine in w, so its values at the 93 simplex vertices determine it on the whole simplex, and the main results use these vertices together with the equal-weight portfolio, which gives $K = 94$. For Power Scheduling, K counts thresholds per hour, and the main results use 21 thresholds that cover the range of the demand. For the two synthetic tasks, we use $K = 32$. Decision quality is relatively stable across the reported probe counts, and every setting remains below the strongest baseline in Table 1. Fewer probes are sometimes better, so the main results do not depend on dense probing.

Table 8: Decision quality across probe counts K. Lower is better.

Allocation		Mean–CVaR		Power Scheduling		Mean–Variance	
K	Test	K	Test	K	Test	K	Test
4	0.329 ± 0.002	8	23.82 ± 0.17	5	3.21 ± 0.10	24	85.3 ± 4.6
16	0.330 ± 0.002	16	23.74 ± 0.15	11	3.20 ± 0.10	47	84.6 ± 3.6
32	0.325 ± 0.005	32	23.61 ± 0.11	21	3.27 ± 0.09	94	82.7 ± 3.7
64	0.359 ± 0.026	64	23.72 ± 0.10	41	3.29 ± 0.17	188	82.7 ± 3.7

Mixture components. Table 9 varies J over $\{1, 2, 4, 8\}$ while retaining the field loss and the task-specific covariance rank. Mean–CVaR Portfolio improves from $J = 1$ to $J = 2$ and then changes little. Allocation is also relatively insensitive to J, whereas Power Scheduling has a higher cost with one component and its lowest test mean at $J = 2$. On Mean–Variance Portfolio, the main setting of $J = 4$ has the lowest mean. Performance is therefore not monotone in J.

Table 9: Decision quality across mixture components J. Lower is better.

Task	$J = 1$	$J = 2$	$J = 4$	$J = 8$
Allocation	0.337 ± 0.004	0.342 ± 0.003	0.334 ± 0.001	0.325 ± 0.005
Mean–CVaR Portfolio	24.38 ± 0.19	23.71 ± 0.12	23.61 ± 0.11	23.77 ± 0.10
Power Scheduling	3.74 ± 0.11	3.02 ± 0.05	3.27 ± 0.09	3.24 ± 0.12
Mean–Variance Portfolio	83.9 ± 2.6	85.8 ± 3.8	82.7 ± 3.7	87.1 ± 1.8