

Contents

1 Introduction	1
2 Related Work	2
3 Preliminaries and Problem Formulation	3
3.1 Score-based Generative Models and Training-Free Guided Sampling	3
3.2 Risk-Averse Learning via Conditional Value-at-Risk	4
4 Risk-Averse Model Training via Loss-Guided Importance Samples	5
4.1 Algorithm	5
4.2 Theoretical Analysis	6
5 Experiments	7
5.1 Risk-Averse Regression over Density-Heterogeneous Gaussians	7
5.1.1 Additional Analysis	8
5.2 Risk-Averse Compression of Wireless Channel State Information	8
6 Discussion, Limitations of Work, and Future Directions	10
A Technical Results	22
A.1 Proof of Theorem 1	22
A.2 Proof of Remark 2	24
B Implementation of the Pretrained Model Loss-guided Sampling	25
B.1 SDE Discretization	25
B.2 Guidance Approximation	26
C Experiments on Gaussian Mixtures	27
C.1 Baseline Method Implementations	28
D Experiments on Wireless Communications Channel State Information	29
E Further Experimental Results	31
E.1 Cost of Importance Sampling	31
E.2 Impact of Importance Level Emphasis	31

Table 3: Notation and description

Notation	Description	Note
\mathbf{X}	Data sample	Random variable
$\mathbf{X}_{(t)}^p$	State of the diffusion at time t started from p	$\mathbf{X}_{(0)}^p \sim p$
\mathbf{x}	Realization of \mathbf{X}	$\mathbf{x} \in \mathbb{R}^{d_1}$
$p(\mathbf{x})$	Base data distribution	
$q(\mathbf{x})$	Importance sampling distribution	
θ	Parameters of task model	$\theta \in \mathbb{R}^{d_2}$
θ_0	Pretrained reference model	
$\ell(\theta; \mathbf{x})$	Loss on input \mathbf{x} for model θ	
β	CVaR confidence level	
α	Value-at-Risk (VaR) threshold	
$F_\beta(\theta, \alpha)$	Surrogate CVaR objective	
$\varphi(\cdot)$	Weighting function	Non-decreasing, $\varphi : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$
Z	Normalization constant	$Z = \mathbb{E}_p[\varphi(\ell(\theta_0; \mathbf{X}^p))]$
K	Total number of training iterations	
λ_k	Learning rate at iteration k	
ϕ_k	Joint variable (θ_k, α_k) at iteration k	
ϕ^*	Optimal joint variable	
$\hat{v}(q)$	Noise term in convergence bound	
κ	Bound on parameter norm $\ \theta_k\ $	$\kappa < \infty$
\mathcal{B}	Dataset of i.i.d. samples from $p(\mathbf{x})$	

A Technical Results

Definition 1. A differentiable function f is convex if $f(\theta') \geq f(\theta) + \langle \nabla f(\theta), \theta' - \theta \rangle, \forall \theta', \theta$.

Definition 2. For a given $L > 0$, a differentiable function f is L -smooth if $\|\nabla f(\theta') - \nabla f(\theta)\| \leq L\|\theta' - \theta\|, \forall \theta', \theta$.

A.1 Proof of Theorem 1

The proof builds upon the stochastic subgradient method analysis for CVaR minimization developed in Meng and Gower (2023), which itself extends the model-based optimization framework of stochastic convex optimization presented in Davis and Drusvyatskiy (2019).

We consider the unconstrained formulation of the CVaR minimization problem given by

$$(\theta^*, \alpha^*) = \underset{\theta \in \mathbb{R}^{d_2}, \alpha \in \mathbb{R}}{\operatorname{argmin}} F_\beta(\theta, \alpha),$$

where the objective function is represented as

$$F_\beta(\theta, \alpha) = \alpha + \frac{1}{1-\beta} \mathbb{E}_{\mathbf{x}^q \sim q} \left[\frac{p(\mathbf{X}^q)}{q(\mathbf{X}^q)} (\ell(\theta; \mathbf{X}^q) - \alpha)^+ \right],$$

with $(x)^+ = \max(x, 0)$ denoting the positive-part operator.

Consider a realization \mathbf{x} sampled from distribution q . Then, the subgradients of $F_\beta(\theta, \alpha; \mathbf{x})$, i.e., the objective value from a realization \mathbf{x} , with respect to θ and α are given by

$$\partial_\theta F_\beta(\theta, \alpha; \mathbf{x}) = \frac{1}{1-\beta} \frac{p(\mathbf{x}) \mathbf{1}_{\ell(\theta; \mathbf{x}) > \alpha} \nabla_\theta \ell(\theta; \mathbf{x})}{q(\mathbf{x})}, \quad (10)$$

$$\partial_\alpha F_\beta(\theta, \alpha; \mathbf{x}) = 1 - \frac{1}{1-\beta} \frac{p(\mathbf{x}) \mathbf{1}_{\ell(\theta; \mathbf{x}) > \alpha}}{q(\mathbf{x})}. \quad (11)$$

Accordingly, the stochastic subgradient updates at iteration k are expressed as:

$$\theta_{k+1} = \theta_k - \lambda_k \cdot \frac{1}{1-\beta} \cdot \frac{p(\mathbf{x}_k)}{q(\mathbf{x}_k)} \cdot \mathbf{1}_{\ell(\theta_k; \mathbf{x}_k) > \alpha_k} \cdot \nabla_\theta \ell(\theta_k; \mathbf{x}_k), \quad (12)$$

$$\alpha_{k+1} = \alpha_k - \lambda_k \cdot \left(1 - \frac{1}{1-\beta} \cdot \frac{p(\mathbf{x}_k)}{q(\mathbf{x}_k)} \cdot \mathbf{1}_{\ell(\theta_k; \mathbf{x}_k) > \alpha_k} \right), \quad (13)$$

where λ_k denotes the step size at iteration k . The updates in (12)-(13) correspond to the procedure `SubGradientDescent`($\partial F_\beta, \theta_k, \alpha_k$) in Algorithm 1.

To analyze convergence, we introduce a linearization model, i.e., the stochastic one-sided model, centered at the current iterate as

$$f_{\phi_k}(\phi, \mathbf{x}) = \alpha_k + \frac{1}{1-\beta} \cdot \frac{p(\mathbf{x})}{q(\mathbf{x})} \cdot (\ell(\theta_k; \mathbf{x}) - \alpha_k)^+ + \langle g_k, \phi - \phi_k \rangle \quad (14)$$

where $g_k = \partial F_\beta(\phi_k; \mathbf{x})$, $\phi_k = \begin{pmatrix} \theta_k \\ \alpha_k \end{pmatrix}$, and $\phi = \begin{pmatrix} \theta \\ \alpha \end{pmatrix}$. The update step is then equivalently expressed as

$$(\theta_{k+1}, \alpha_{k+1}) = \operatorname{argmin}_{\theta, \alpha} f_{\phi_k}(\phi, \mathbf{x}) + \frac{1}{2\lambda_k} \|\phi - \phi_k\|^2.$$

Under this formulation, the convergence behavior of the algorithm can be analyzed via the theoretical framework of model-based stochastic subgradient methods. In particular, under the following assumptions (B1)–(B4) involving sample accessibility, one-sided accuracy, weak convexity, and Lipschitz continuity, the method achieves a convergence rate of $\mathcal{O}(1/\sqrt{K})$ after K iterations (Meng and Gower, 2023; Davis and Drusvyatskiy, 2019).

Our analysis uses this framework in the CVaR minimization setting with the fixed importance sampling distribution. Specifically, we consider a sampling distribution $q(\mathbf{x})$ that is constructed a priori based on an initial loss evaluation and a task-dependent importance weight function. This extension allows the optimization to benefit from variance reduction while preserving the convergence guarantees of stochastic model-based methods. We next verify assumptions (B1)–(B4).

(B1) Sampling. It is possible to generate i.i.d. realizations $\mathbf{x}_1, \mathbf{x}_2, \dots \sim q$. This condition is satisfied by the underlying assumption of Theorem 1.

(B2) One-sided accuracy. There exists $\zeta \in \mathbb{R}$ and there is an open convex set U containing the domain and a measurable function $(\phi_k, \phi, \mathbf{x}) \mapsto f_{\phi_k}(\phi; \mathbf{x})$, defined on $U \times U \times \Omega$, satisfying

$$\mathbb{E}_{\mathbf{x}^q \sim q}[f_{\phi_k}(\phi_k; \mathbf{X}^q)] = F_\beta(\theta_k, \alpha_k) \quad \forall \phi_k \in U, \quad (15)$$

and

$$\mathbb{E}_{\mathbf{x}^q \sim q}[f_{\phi_k}(\phi; \mathbf{X}^q) - F_\beta(\theta, \alpha)] \leq \frac{\zeta}{2} \|\phi - \phi_k\|^2, \quad (16)$$

where Ω is the sample space.

The equality (15) holds due to the definition of $f_{\phi_k}(\phi, \mathbf{x})$. Moreover, $\mathbb{E}_{\mathbf{x}^q \sim q}[f_{\phi_k}(\phi; \mathbf{X}^q) - F_\beta(\theta, \alpha)] = F_\beta(\theta_k, \alpha_k) - F_\beta(\theta, \alpha) + \mathbb{E}_{\mathbf{x}^q \sim q}[\langle g_k, \phi - \phi_k \rangle] \leq 0$ by the convexity of $F_\beta(\theta, \alpha)$ with respect to ϕ , indicating $\zeta = 0$.

(B3) Weak convexity. $f_{\phi_k}(\phi; \mathbf{x})$ is convex for all ϕ_k , a.e. $\mathbf{x} \in \Omega$. This holds by the linearization model definition in (14).

(B4) Lipschitz property. There exist $V \in \mathbb{R}$ and a measurable function $v : \Omega \rightarrow \mathbb{R}_+$ satisfying $\sqrt{\mathbb{E}_{\mathbf{x}}[v(\mathbf{X})^2]} \leq V$ such that

$$f_{\phi_k}(\phi_k; \mathbf{x}) - f_{\phi_k}(\phi; \mathbf{x}) \leq v(\mathbf{x}) \|\phi_k - \phi\| \quad (17)$$

$\forall \phi_k, \phi \in U$ and a.e. $\mathbf{x} \sim q$.

To show this, we examine the one-sided model gap as follows.

$$f_{\phi_k}(\phi_k, \mathbf{x}) - f_{\phi_k}(\phi, \mathbf{x}) = \langle g_k, \phi_k - \phi \rangle - \langle g_k, \phi - \phi_k \rangle \leq \|g_k\| \|\phi_k - \phi\| \quad (18)$$

where g_k is the subgradient of the estimated object. The norm of the subgradient is given as follows.

$$\|g_k\|^2 = \left\| \frac{1}{1-\beta} \frac{p(\mathbf{x}_k) u_k \nabla_\theta \ell(\theta_k; \mathbf{x}_k)}{q(\mathbf{x}_k)} \right\|^2 + \left\| 1 - \frac{1}{1-\beta} \frac{p(\mathbf{x}_k) u_k}{q(\mathbf{x}_k)} \right\|^2 \quad (19)$$

$$\leq \frac{1}{(1-\beta)^2} \left\| \frac{p(\mathbf{x}_k) \nabla_\theta \ell(\theta_k; \mathbf{x}_k)}{q(\mathbf{x}_k)} \right\|^2 + 1 + \frac{p(\mathbf{x}_k)^2}{q(\mathbf{x}_k)^2 (1-\beta)^2} \quad (20)$$

where $u_k = \mathbf{1}_{\ell(\theta_k; \mathbf{x}_k) > \alpha_k}$ and the inequality holds by the subadditivity of the norm. We denote the square root of the upper bound as v as

$$v(\mathbf{x}) := \sqrt{\frac{1}{(1-\beta)^2} \left\| \frac{p(\mathbf{x}) \nabla_{\theta} \ell(\theta_k; \mathbf{x})}{q(\mathbf{x})} \right\|^2 + 1 + \frac{p(\mathbf{x})^2}{q(\mathbf{x})^2 (1-\beta)^2}}. \quad (21)$$

This function $v(\mathbf{x})$ satisfies the pointwise Lipschitz condition [\(17\)](#). Furthermore, we use upper bounds on $v(\mathbf{x})$ to show the connection between the gradient of the loss and its value.

Consider the expected value of the stochastic noise as follows.

$$\sqrt{\mathbb{E}_{\mathbf{X}^q \sim q}[v(\mathbf{X}^q)^2]} = \sqrt{\mathbb{E}_{\mathbf{X}^q \sim q} \left[\frac{1}{(1-\beta)^2} \left\| \frac{p(\mathbf{X}^q) \nabla_{\theta} \ell(\theta_k; \mathbf{X}^q)}{q(\mathbf{X}^q)} \right\|^2 + 1 + \frac{p(\mathbf{X}^q)^2}{q(\mathbf{X}^q)^2 (1-\beta)^2} \right]}. \quad (22)$$

We then have

$$\mathbb{E}_{\mathbf{X}^q \sim q} \left[\frac{1}{(1-\beta)^2} \left\| \frac{p(\mathbf{X}^q) \nabla_{\theta} \ell(\theta_k; \mathbf{X}^q)}{q(\mathbf{X}^q)} \right\|^2 + 1 + \frac{p(\mathbf{X}^q)^2}{q(\mathbf{X}^q)^2 (1-\beta)^2} \right] \quad (23)$$

$$\leq \mathbb{E}_{\mathbf{X}^q \sim q} \left[\frac{1}{(1-\beta)^2} \left\| \frac{p(\mathbf{X}^q) (\|\nabla_{\theta} \ell(\theta_0; \mathbf{X}^q)\| + 2L_2 \kappa)}{q(\mathbf{X}^q)} \right\|^2 + 1 + \frac{p(\mathbf{X}^q)^2}{q(\mathbf{X}^q)^2 (1-\beta)^2} \right] \quad (24)$$

$$= \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{1}{(1-\beta)^2} \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p)} (\|\nabla_{\theta} \ell(\theta_0; \mathbf{X}^p)\| + 2L_2 \kappa)^2 + 1 + \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p) (1-\beta)^2} \right] \quad (25)$$

where the first inequality holds by the assumption that the gradient of the loss function satisfies

$$\|\nabla \ell(\theta; \mathbf{x})\| - \|\nabla \ell(\theta'; \mathbf{x})\| \leq L_2 \|\theta - \theta'\| \quad \forall \theta, \theta' \in \mathbb{R}^{d_2}, \mathbf{x} \in \mathbb{R}^{d_1}, \quad (26)$$

and the norm of the model parameter θ has a bounded value κ with $L_2 \|\theta_k - \theta_0\| \leq 2L_2 \kappa$. A corresponding bound also holds with L_1 in place of L_2 by the reverse triangle inequality, $\|\|\nabla \ell(\theta_k; \mathbf{x})\| - \|\nabla \ell(\theta_0; \mathbf{x})\|\| \leq \|\nabla \ell(\theta_k; \mathbf{x}) - \nabla \ell(\theta_0; \mathbf{x})\|$.

Moreover, L_1 -smoothness and convexity yield

$$\|\nabla_{\theta} \ell(\theta_0; \mathbf{x})\|^2 \leq 2L_1 (\ell(\theta_0; \mathbf{x}) - \ell^*) \leq 2L_1 \ell(\theta_0; \mathbf{x}). \quad (27)$$

Combining this, we define $\hat{v}(q)$ such that

$$\hat{v}(q) := \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{1}{(1-\beta)^2} \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p)} \left(\sqrt{2L_1 \ell(\theta_0; \mathbf{X}^p)} + 2L_2 \kappa \right)^2 + 1 + \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p) (1-\beta)^2} \right], \quad (28)$$

which satisfies $\sqrt{\mathbb{E}_{\mathbf{X}}[v(\mathbf{X})^2]} \leq \sqrt{\hat{v}(q)}$ and we set $V = \sqrt{\hat{v}(q)}$.

To simplify the term, we introduce $w^*(\mathbf{x}) = \sqrt{\left(\sqrt{2L_1 \ell(\theta_0; \mathbf{x})} + 2L_2 \kappa \right)^2 + 1}$ which gives us

$$\hat{v}(q) = \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{w^*(\mathbf{X}^p)^2}{(1-\beta)^2} \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p)} + 1 \right].$$

The imposed conditions **(B1–B4)** allow us to directly apply the standard model-based stochastic gradient convergence analysis, as established in Theorem 4.4 of [Davis and Drusvyatskiy \(2019\)](#) and Theorem 5.2 of [Meng and Gower \(2023\)](#). This yields the following convergence bound

$$\mathbb{E} \left[F_{\beta} \left(\frac{1}{K+1} \sum_{k=1}^{K+1} \phi_k \right) - F_{\beta}(\phi^*) \right] \leq \frac{\|(\theta_0, \alpha_0)^{\top} - \phi^*\|^2}{2\lambda \sqrt{K+1}} + \frac{\lambda \hat{v}(q)}{\sqrt{K+1}}. \quad (29)$$

A.2 Proof of Remark [2](#)

Recall the definition of the stochastic noise $\hat{v}(q) = \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{w^*(\mathbf{X}^p)^2}{(1-\beta)^2} \frac{p(\mathbf{X}^p)}{q(\mathbf{X}^p)} + 1 \right]$. Since the scalar constant $1/(1-\beta)^2$ and the additive term $+1$ are independent of the choice of the sampling distribution q . Consider the importance sampling distribution q such that $q(\mathbf{x}) \propto \varphi(\ell(\theta_0; \mathbf{x}))p(\mathbf{x})$.

By definition, we have

$$\hat{v}(p) = \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{w^*(\mathbf{X}^p)^2}{(1-\beta)^2} + 1 \right] = \frac{1}{(1-\beta)^2} \mathbb{E}_{\mathbf{X}^p \sim p} [w^*(\mathbf{X}^p)^2] + 1$$

and

$$\hat{v}(q) = \frac{1}{(1-\beta)^2} \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{w^*(\mathbf{X}^p)^2}{\varphi(\ell(\theta_0; \mathbf{X}^p))} \right] \mathbb{E}_{\mathbf{X}^p \sim p} [\varphi(\ell(\theta_0; \mathbf{X}^p))] + 1,$$

which directly gives us the equivalent condition

$$\mathbb{E}[w^*(\mathbf{X}^p)^2] \geq \mathbb{E} \left[\frac{w^*(\mathbf{X}^p)^2}{\varphi(\ell(\theta_0; \mathbf{X}^p))} \right] \cdot \mathbb{E}[\varphi(\ell(\theta_0; \mathbf{X}^p))]. \quad (30)$$

Similarly, consider the per-iteration CVaR objective as

$$F_\beta(\theta_k, \alpha_k) = \alpha_k + \frac{1}{1-\beta} \mathbb{E}_{\mathbf{X}^q \sim q} \left[\frac{p(\mathbf{X}^q)}{q(\mathbf{X}^q)} (\ell(\theta_k; \mathbf{X}^q) - \alpha_k)^+ \right]. \quad (31)$$

Then, the variance of the MC estimator under q is smaller than that under the base distribution p if and only if

$$\mathbb{E}_{\mathbf{X}^p \sim p} [((\ell(\theta_k; \mathbf{X}^p) - \alpha_k)^+)^2] \geq \mathbb{E}_{\mathbf{X}^q \sim q} \left[\frac{Z}{\varphi(\ell(\theta_0; \mathbf{X}^q))} ((\ell(\theta_k; \mathbf{X}^q) - \alpha_k)^+)^2 \right]. \quad (32)$$

Intuitively, this condition can be satisfied when high-loss examples under the reference model θ_0 tend to remain high-loss during fine-tuning, so that $((\ell(\theta_k; \mathbf{x}) - \alpha_k)^+)^2$ and $\varphi(\ell(\theta_0; \mathbf{x}))$ are positively correlated. Such an assumption can be realistic in settings where high-loss inputs often persist across training iterations and require multiple optimization steps to mitigate their contribution to risk.

B Implementation of the Pretrained Model Loss-guided Sampling

This section outlines the practical implementation of loss-guided sampling using a pretrained score-based generative model, characterized by its score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

Following standard practice (Ho et al., 2020; Lugmayr et al., 2022; Nichol and Dhariwal, 2021; Choi et al., 2021), we adopt $\mathbf{f}(\mathbf{x}, t) = -\frac{1}{2}\beta(t)\mathbf{x}$, $\sigma(t) = \sqrt{\beta(t)}$, where $\beta(t)$ is a non-negative scalar-valued function satisfying $0 \leq \beta(t) \leq 1$.²

B.1 SDE Discretization

Given the approximated score function of the importance sampling density, $\nabla_{\mathbf{x}} \log q_t(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \tilde{g}(\mathbf{x}, t)$ where $\tilde{g}(\mathbf{x}, t)$ is given in (41), we follow the reverse-time SDE formulation to simulate the generative process. Recall that the continuous-time reverse SDE governed by the time-dependent score function $\nabla_{\mathbf{x}} \log q_t(\mathbf{x})$ is given as

$$d\mathbf{X}_{(t)}^q = \left(-\frac{1}{2}\beta(t)\mathbf{X}_{(t)}^q - \beta(t)\nabla_{\mathbf{x}} \log q_t(\mathbf{X}_{(t)}^q) \right) dt + \sqrt{\beta(t)} d\tilde{\mathbf{W}}_{(t)}.$$

This formulation is equivalent to the reverse-time dynamics derived in Song et al. (2021) under the variance-preserving setting.

To implement the sampling process in discrete time, we adopt the DDPM-style discretization. In this paragraph, we slightly abuse notation by using t to denote discrete timesteps $t \in \{0, \dots, T-1\}$. We define discrete-time variance schedulers as

$$\alpha_t := 1 - \beta_t, \quad \bar{\alpha}_t := \prod_{s=0}^t \alpha_s = \prod_{s=0}^t (1 - \beta_s). \quad (33)$$

²We follow conventional score-based generative model notation while avoiding clashes with CVaR parameters. CVaR: (α, β) ; generative model: variance-schedule parameters (α, β) . Thus, (α, β) are used only for diffusion schedules, whereas (α, β) are used for denoting the VaR value and confidence level.

We use the cosine schedule (Nichol and Dhariwal 2021) in its discrete form as

$$\nu_t = \cos^2\left(\frac{t/T + \varepsilon_\beta}{1 + \varepsilon_\beta} \cdot \frac{\pi}{2}\right), \quad \beta_t = 1 - \frac{\nu_{t+1}}{\nu_t}, \quad \alpha_t = 1 - \beta_t, \quad (34)$$

with $\varepsilon_\beta = 0.008$ for numerical stability. Then the update formula is given as

$$\mathbf{x}_{(t-1)} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_{(t)} + \beta_t \nabla_{\mathbf{x}_{(t)}} \log q_t(\mathbf{x}_{(t)}) \right) + \sqrt{\tilde{\beta}_t} \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (35)$$

where $\tilde{\beta}_t = \beta_t \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}$.

For baseline methods that do not use importance samples, we utilize samples generated from the pretrained base score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ through the reverse process.

B.2 Guidance Approximation

The noise-perturbed score function of the importance sampling density q can be represented as a summation of the base score function and a guidance term g as follows.

$$\nabla_{\mathbf{x}} \log q_t(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + g(\mathbf{x}, t), \quad (36)$$

where the guidance term $g(\mathbf{x}, t)$ is defined by

$$g(\mathbf{x}, t) := \nabla_{\mathbf{x}} \log \mathbb{E}_{\mathbf{X}_{(0)}^p \sim p_{\mathbf{X}_{(0)}^p | \mathbf{X}_{(t)}^p}(\cdot | \mathbf{x})} \left[w(\mathbf{X}_{(0)}^p) \right], \quad (37)$$

with $w(\mathbf{X}_{(0)}^p)$ denoting the weight function as $q(\mathbf{x}) \propto w(\mathbf{x})p(\mathbf{x})$. In our setting $w(\mathbf{x}) = \varphi(\ell(\theta_0; \mathbf{x}))$; for brevity we write w throughout this section.

Since this conditional expectation in (37) is intractable in general, a first-order Taylor expansion of $w(\mathbf{X}_{(0)}^p)$ around the conditional mean can be considered. Let

$$\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t} := \mathbb{E}[\mathbf{X}_{(0)}^p | \mathbf{X}_{(t)}^p = \mathbf{x}]$$

denote the conditional mean of $\mathbf{X}_{(0)}^p$ given $\mathbf{X}_{(t)}^p = \mathbf{x}$. Linearizing the loss function around this point yields

$$w(\mathbf{X}_{(0)}^p) \approx w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}) + \nabla w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t})^\top (\mathbf{X}_{(0)}^p - \bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}).$$

Taking the expectation over $p_{\mathbf{X}_{(0)}^p | \mathbf{X}_{(t)}^p}$ eliminates the second term due to the zero-mean residual, giving the following approximation

$$g(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}). \quad (38)$$

The conditional mean $\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}$ is represented via Tweedie's formula (Chung et al. 2023; Yu et al. 2023), which connects the posterior mean to the score function of the marginal at time t ,

$$\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t} = \frac{1}{\sqrt{\bar{\alpha}(t)}} (\mathbf{x} + (1 - \bar{\alpha}(t)) \nabla_{\mathbf{x}} \log p_t(\mathbf{x})), \quad (39)$$

where $\bar{\alpha}(t) = \exp\left(-\int_0^t \beta(s) ds\right)$.

We further simplify (38) by using the finite difference approximation of the Hessian (Kim et al. 2025a). Specifically, for a small step size $\epsilon > 0$, the following directional approximation holds:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \epsilon H_{p_t}(\mathbf{x}) \nabla_{\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x} + \epsilon \nabla_{\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t})), \quad (40)$$

which gives us

$$\begin{aligned} g(\mathbf{x}, t) &\approx \tilde{g}(\mathbf{x}, t) := \frac{1}{\sqrt{\bar{\alpha}(t)}} \nabla_{\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}) \\ &+ \frac{1}{\epsilon(1 - \bar{\alpha}(t))^{-1} \sqrt{\bar{\alpha}(t)}} \left(\nabla_{\mathbf{x}} \log p_t(\mathbf{x} + \epsilon \nabla_{\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t})) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right). \end{aligned} \quad (41)$$

In our applications, the approximation in (41) can yield a guidance term for $w(\mathbf{x}) = \varphi(\ell(\theta_0; \mathbf{x}))$; sampling proceeds by replacing the base score with $\nabla_{\mathbf{x}} \log q_t(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \tilde{g}(\mathbf{x}, t)$ in the reverse-time updates in (35).

C Experiments on Gaussian Mixtures

This experiment is designed to evaluate the performance of the proposed RAMIS framework and baseline methods in a controlled setting where the data distribution contains low-density (i.e., rare) regions that are often underrepresented in standard training regimes. Specifically, we construct a synthetic mixture-of-Gaussians distribution in which certain components contribute small probability mass compared to high-density regions. These rare components are configured to induce high loss under models trained with ERM, thereby creating a challenging testbed for assessing risk-aware generative sampling.

To evaluate the sampling behavior of the proposed method, particularly the ability to capture rare, high-loss regions, we leverage the closed-form expression of the ground-truth score function for the mixture of Gaussian distributions. This allows us to perform both accurate generative modeling and precise evaluation of coverage in the tail of the data distribution.

In these experiments, for each random seed, we generate new samples from the diffusion model and evaluate the performance of the baseline methods on these samples.

Score Function for Mixture of Gaussians. We consider a synthetic mixture-of-Gaussians (MoG) prior at $t = 0$:

$$p_0(\mathbf{x}) = \sum_{i=1}^N \pi_i \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{0,i}, \boldsymbol{\Sigma}_{0,i}),$$

where $\pi_i \geq 0$, $\sum_{i=1}^N \pi_i = 1$, and $\boldsymbol{\mu}_{t,i}$ and $\boldsymbol{\Sigma}_{t,i}$ denote the mean and covariance matrix of i -th component at t , respectively.

Under the forward diffusion process, the marginal distribution at time t evolves with

$$p_t(\mathbf{x}) = \sum_{i=1}^N \pi_i p_{t,i}(\mathbf{x}), \quad p_{t,i}(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i}),$$

where $\boldsymbol{\mu}_{t,i} = \sqrt{\bar{\alpha}_t} \boldsymbol{\mu}_{0,i}$ and $\boldsymbol{\Sigma}_{t,i} = \bar{\alpha}_t \boldsymbol{\Sigma}_{0,i} + (1 - \bar{\alpha}_t) \mathbf{I}$.

The score function of the i -th component and the posterior responsibility are given as

$$\nabla_{\mathbf{x}} \log p_{t,i}(\mathbf{x}) = -\boldsymbol{\Sigma}_{t,i}^{-1} (\mathbf{x} - \boldsymbol{\mu}_{t,i}), \quad \rho_{t,i}(\mathbf{x}) = \frac{\pi_i p_{t,i}(\mathbf{x})}{\sum_{j=1}^N \pi_j p_{t,j}(\mathbf{x})}.$$

Then, the score function of the mixture distribution is

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) = \sum_{i=1}^N \rho_{t,i}(\mathbf{x}) \nabla_{\mathbf{x}} \log p_{t,i}(\mathbf{x}).$$

We set $T = 100$ and use the discretization method and the variance scheduling presented in Appendix [B.1](#)

Task Model. For the regression task on the Gaussian Mixture distribution, we consider a simple nonlinear model that maps the input vector $\mathbf{x} \in \mathbb{R}^2$ to a scalar prediction. The model operates on the first coordinate of the input, applying a quadratic transformation. The target label is defined as the second coordinate of the input. The loss function is the mean squared error between the model prediction and the label as follows.

$$\ell(\theta; \mathbf{x}) = \left(\theta_{[1]} \mathbf{x}_{[1]}^2 + \theta_{[2]} \mathbf{x}_{[1]} + \theta_{[3]} - \mathbf{x}_{[2]} \right)^2.$$

Pretrained Model Training. The regression model is first trained using ERM on a dataset containing 100 samples drawn from the predefined generative model. The dataset is evenly split into training and validation sets. Training is conducted for 1000 epochs using the Adam optimizer with a learning rate of 0.1 and a batch size of 100.

Table 4: CVaR (mean \pm std) across quantile levels β when training with the Extremile objective.

Method $\backslash \beta$	0.99	0.95	0.90	0.80	0.50
RAMIS + Extremile	0.187 \pm 0.144	0.064 \pm 0.021	0.044 \pm 0.015	0.0288 \pm 0.0062	0.0158 \pm 0.0015
Extremile	0.339 \pm 0.207	0.090 \pm 0.044	0.048 \pm 0.021	0.0289 \pm 0.0041	0.0179 \pm 0.0025

Fine-Tuning. Each method, including the proposed RAMIS framework and all baselines, performs 1000 epochs of fine-tuning to adapt to newly generated samples. For RAMIS, fine-tuning is conducted using samples drawn via importance sampling from the generative model, where the importance weight function is chosen as $\varphi(x) = x^{1/2}$.

Following common practice in guided diffusion, we stabilize sampling by gradually increasing the strength of the additive guidance over reverse steps [Kim et al., (2025c)]. Specifically, at reverse time t we multiply $\tilde{g}(\mathbf{x}, t)$ by $((1 + \gamma)^t - 1)/((1 + \gamma)^{T-1} - 1)$ for $t = T - 1, \dots, 0$, so the guidance weight increases over the reverse diffusion trajectory. We use $\gamma = 0.001$.

Baseline methods are fine-tuned using the same number of newly generated samples, but drawn uniformly from the base generative model. All fine-tuning procedures use the same optimizer and training configuration as the initial ERM phase.

Applications with Other Risk Measures. This subsection examines whether the proposed framework based on the score-based generative models and loss-guided sampling is also helpful when training with different risk objectives. As an example, we consider *Extremiles* [Daouia et al., 2022].

Let $\mathbf{L} = \ell(\theta; \mathbf{X})$ denote the per-sample loss and let \tilde{F} be the CDF of \mathbf{L} . Following [Daouia et al., 2022, 2019], define $\tilde{K}_\tau(z) = z^{r(\tau)}$ for $z \in [0, 1]$ and $0.5 \leq \tau < 1$, where $r(\tau) = \log(1/2)/\log(\tau)$, and set $J_\tau(z) = \frac{d}{dz} \tilde{K}_\tau(z)$. The probability-weighted-moment form of the τ extremile is given as

$$\mathbb{E}_{\mathbf{X}^p \sim p}[\mathbf{L} J_\tau(\tilde{F}(\mathbf{L}))].$$

In our implementation, \tilde{F} is estimated on each minibatch using a weighted empirical CDF based on the importance weighted samples from the pretrained generative model. We evaluate $\beta \in \{0.99, 0.95, 0.90, 0.80, 0.50\}$ and set $\tau = \beta$ to align tail emphasis with the CVaR quantile.

Table 4 indicates that RAMIS+Extremile achieves lower tail risk than optimizing the Extremile objective alone on the evaluated setups without importance samples. This empirical evidence supports the usefulness of importance-weighted samples when training with a different risk objective. Moreover, designing the importance sampling distribution to reflect the chosen risk measure or target objective may further improve performance.

C.1 Baseline Method Implementations

We evaluate the proposed RAMIS framework against strong baselines that capture risk-sensitive training paradigms. All methods are implemented under a unified training pipeline, sharing the same network architecture, initialization, optimizer, and sample budget, in order to ensure a fair and controlled comparison.

Empirical Risk Minimization (ERM). ERM serves as the canonical risk-neutral baseline. It is trained directly using samples from the original data distribution $p(\mathbf{x})$ without reweighting or sample selection. The model is optimized to minimize the expected loss $\mathcal{L}_{\text{ERM}} = \mathbb{E}_{\mathbf{X}^p \sim p}[\ell(\theta; \mathbf{X}^p)]$. This objective corresponds to uniform sampling from p followed by standard stochastic gradient descent.

Stochastic SubGradient Method (SSGM). We also evaluate the stochastic model-based optimization method presented in [Meng and Gower, 2023], which ours extends. This is recovered by disabling importance sampling, i.e., by setting the weighting function φ to a constant. In this setting, samples are generated from a pretrained score-based generative model approximating $p(\mathbf{x})$, and the model parameters are updated using stochastic subgradients of the loss.

Distributionally Robust CVaR (DORO). To implement the CVaR-DORO method (Zhai et al. 2021), given a minibatch of \hat{B} samples and associated loss vector $\mathbf{l} \in \mathbb{R}^{\hat{B}}$, we sort losses in descending order and compute the following loss function

$$\mathcal{L}_{\text{CVaR-DORO}} = \frac{1}{(1 - \beta)(\hat{B} - n_2)} \sum_{i=n_2}^{\hat{B}} \mathbf{l}_{(i)},$$

where $\mathbf{l}_{(i)}$ denotes the i -th largest loss, and the selection range $[n_2, \hat{B}]$ is determined by quantile truncation parameters $n_1 = \lfloor (1 - \beta)\hat{B} \rfloor$ and $n_2 = 0$. The loss vector is obtained via computing $\ell(\theta; \mathbf{x})$ where \mathbf{x} is a realization of the minibatch. Although DORO was designed for an outlier-aware setting with $n_2 \geq 0$, in this section, we set $n_2 = 0$ and use it simply as a within-minibatch sorter that selects the top $(1 - \beta)$ fraction of highest-loss samples.

χ^2 -Divergence Robust Optimization (χ^2 -DRO). We implement a divergence-constrained robust optimization baseline based on χ^2 -divergence risk envelopes (Namkoong and Duchi 2016). The robust objective is given by

$$\mathcal{L}_{\chi^2} = \inf_{\eta \in [0, L_{\max}]} \left\{ C \cdot \sqrt{\mathbb{E}_{\mathbf{X}^p \sim p} \left[((\ell(\mathbf{X}^p) - \eta)^+)^2 \right]} + \eta \right\},$$

where $C = \sqrt{1 + \left(\frac{1}{1-\beta} - 1\right)^2}$ is a divergence-induced robustness factor and $L_{\max} = 10$ is a fixed upper bound for η . The inner minimization over η is solved numerically using Brent’s method.

D Experiments on Wireless Communications Channel State Information

Background. Accurate downlink channel state information (CSI) feedback is essential for high throughput and effective interference coordination in 5G and beyond. The challenge intensifies in massive MIMO deployments that span hundreds to thousands of subcarriers (Dahlman et al. 2013; Zaidi et al. 2018), producing high-dimensional CSI that must be returned to the base station (BS) from the user equipment (UE). Conventional feedback schemes scale poorly in this regime due to the substantial signaling overhead required.

To address this bottleneck, recent work has focused on deep learning-based CSI compression (Wen et al. 2018; Guo et al. 2022). These methods often use autoencoders: the UE compresses CSI into a compact representation, and the BS decodes it. Trained on environment-specific data, such models learn channel priors and typically surpass codebook-based and compressed-sensing approaches. Early efforts like CsiNet (Wen et al. 2018) introduced convolutional architectures that outperformed classical baselines, followed by extensions that exploit temporal (Wang et al. 2018; Liu and Simeone 2021) and spatial (Lu et al. 2020; Cai et al. 2019) structures in CSI. This data-driven direction has attracted considerable interest in both research and standardization, e.g., 3GPP Release 18 (Lin 2022).

More recently, score-based generative models have emerged as powerful tools for modeling and synthesizing wireless channels (Lee et al. 2025). Their ability to generate realistic channel realizations has enabled applications in joint source–channel coding (Ankireddy et al. 2025) and neural CSI compression (Kim et al. 2025b), where generative priors capture complex channel distributions and facilitate robust reconstruction from highly compressed representations.

Objective. Building on these developments, we investigate the effectiveness of loss-guided channel generation for improving neural CSI compression. Specifically, we fine-tune a pretrained neural CSI compressor by augmenting its training set with high-loss channel realizations generated via a score-based channel generative model.

Dataset. We evaluate our framework on a wireless CSI compression task using datasets generated with the QuaDRiGa channel simulator (QUAsi Deterministic RadIo Channel GenerAtor) (Jaekel et al. 2014, 2021). We consider carrier frequencies of 0.8 GHz and 2.4 GHz, corresponding to urban macro-cell and urban micro-cell deployment scenarios, respectively. For each deployment, we generate both line-of-sight (LOS) and non-line-of-sight (NLOS) channel realizations. The LOS

channels contain 5–10 dominant signal clusters (Jaeckel et al. 2021, Sec. 3), while the NLOS channels exhibit richer scattering with 40–50. This setup captures wireless environments with varying propagation geometry and scattering complexity.

Macro-cell base stations (BSs) are placed at a height of 25 m, while micro-cell BSs are positioned at 10 m. User equipment (UE) employs omnidirectional antennas and is randomly distributed within a circular region of radius 30 m centered at the BS. The BS antennas follow the 3GPP-3D antenna model.

To reduce training complexity, we operate on a compact angular–delay representation of the channel. Specifically, each CSI realization is transformed into the delay–angle domain using a two-dimensional inverse fast Fourier transform (2D-IFFT), after which the high-delay region, whose energy is negligible, is cropped. The resulting representation is a 32×32 complex-valued tensor per sample.

Generative Model. The denoising network used in our diffusion model is based on a modified UNet architecture. The network is implemented using the UNet2DModel class provided by diffusers library (von Platen et al. 2022), with an input resolution of 32×32 and two input/output channels corresponding to the real and imaginary parts of the CSI tensor. Each block in the UNet contains two residual convolutional layers (layers_per_block=2). The encoder path comprises six downsampling stages with output channel sizes of (128, 128, 256, 256, 512, 512), where the fifth block includes a spatial self-attention mechanism via the AttnDownBlock2D layer. The decoder mirrors this structure, employing six upsampling blocks with corresponding channel sizes in reverse order, and includes an attention layer (AttnUpBlock2D) in the second stage of the decoder.

The discretization of the SDE based on this score model follows the method in Appendix B.1 including the variance scheduling and $T = 10^2$.

Task Model. We adopt a vector-quantized autoencoder architecture for compressing and reconstructing CSI matrices. Specifically, we employ VQModel in diffusers (von Platen et al. 2022) and customize it to operate on two-channel inputs (e.g., representing real and imaginary components) and outputs reconstructions of the same dimensionality.

The encoder network comprises three 2D downsampling blocks, each with increasing channel capacity (64, 128, 256) and two convolutional layers per block, enabling the model to effectively compress spatially correlated CSI features. Symmetrically, the decoder is composed of three upsampling blocks mirroring the encoder’s configuration. The architecture utilizes the SiLU activation function and group normalization with 32 groups. The latent representation has 128 channels and is discretized by using four vector quantization learnable embeddings of dimension 128. During forward propagation, the output includes both the reconstructed CSI sample and an auxiliary vector quantization commitment loss.

Generative Model Training Configuration. The score-based generative model is trained for 10^3 epochs using a dataset consisting of 8×10^4 samples. We use the Adam optimizer with an initial learning rate of 10^{-4} . We adopt the one-cycle learning rate scheduler, which increases the learning rate linearly to a peak value during the first 25% of training steps and then anneals it following a cosine decay schedule. The scheduler is configured with total steps set to the product of the number of epochs and the number of batches per epoch.

Pretrained Model Training Configuration. The pretrained model is trained using ERM with the mean squared error loss over 10^2 epochs with a training set of size 10^5 . We use the Adam optimizer with a learning rate of 10^{-4} and a batch size of 10^2 .

Fine-Tuning. The fine-tuning learning rate is set to 10^{-5} and optimization is conducted using Adam. For RAMIS, the importance weight function φ is chosen as the squared loss function (i.e., $\varphi(x) = x^2$). For baseline methods, fine-tuning is performed using 10^5 newly generated samples from the generative model, without the importance reweighting. For RAMIS and SSGM, the positive part operator $(x)^+ = \max(x, 0)$ is implemented using the SoftPlus surrogate $0.1 \log(1 + \exp(x/0.1))$ (Soma and Yoshida 2020). Per-sample importance weights are stabilized by adding a constant of 1, and normalized to sum to one per mini-batch. For DORO, we set $n_1 = \lfloor (1 - \beta(1 - 10^{-2})) \times \hat{B} \rfloor$ and $n_2 = \hat{B} \times 10^{-2}$, considering the worst 1% of samples as outliers.

Table 5: Cross-quantile evaluation of models trained with $\beta=0.99$. Lower is better.

β	RAMIS (ours)	SSGM	DORO	χ^2 -DRO	ERM
0.99	2.2604 \pm 0.0419	2.3413 \pm 0.0346	2.4987 \pm 0.0744	2.5594 \pm 0.0341	2.3442 \pm 0.0346
0.95	1.4740 \pm 0.0261	1.5248 \pm 0.0197	1.6559 \pm 0.0423	1.6888 \pm 0.0243	1.5273 \pm 0.0198
0.90	1.1279 \pm 0.0193	1.1655 \pm 0.0129	1.2785 \pm 0.0249	1.2985 \pm 0.0201	1.1670 \pm 0.0126
0.80	0.7889 \pm 0.0127	0.8109 \pm 0.0070	0.9043 \pm 0.0085	0.9093 \pm 0.0193	0.8119 \pm 0.0067
0.50	0.3864 \pm 0.0056	0.3949 \pm 0.0026	0.4552 \pm 0.0030	0.4503 \pm 0.0214	0.3953 \pm 0.0024
0.00	0.2026 \pm 0.0028	0.2059 \pm 0.0015	0.2418 \pm 0.0017	0.2406 \pm 0.0221	0.2061 \pm 0.0014

The remaining details, including the baseline methods implementation and generative model discretization, follow the setups described in Appendix C

We evaluate performance across 10 trials, consisting of five independent training runs conducted on each of two distinct sets of fixed generated data, base and importance samples.

Additional Results. In Table 5 we study whether emphasizing tail risk induces trade-offs with average-case performance. We fix the training target to $\beta = 0.99$ (i.e., the worst 1% tail) and evaluate each trained model across a sweep of non-target quantiles $\beta \in \{0.99, 0.95, 0.90, 0.80, 0.50, 0.00\}$.

RAMIS delivers the strongest performance at the intended target $\beta = 0.99$ and across the high-risk tail, outperforming alternatives at $\beta \in \{0.95, 0.90\}$ with a notable gain. While a tail-average trade-off is observed in individual trials, the aggregated results across multiple trials show that the proposed method maintains good performance on average.

E Further Experimental Results

E.1 Cost of Importance Sampling

Compared to the base sampling from the given generative model, the additional computational cost of importance sampling primarily arises from the need to evaluate a composed gradient of the importance weight function $w(\mathbf{x})$, which in our case is $\varphi(\ell(\theta_0, \mathbf{x}))$, alongside the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. Specifically, the guided score used in our method is approximated as $g(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log w(\bar{\mathbf{x}}'_0 |_{\mathbf{x}, t})$, as explored in (Chung et al., 2023). This gradient computation must be performed at each reverse diffusion step, making it more costly than standard sampling from the base model. For example, in Section 5.1 the sampling time increases by approximately $2.3\times$, relative to the base generative model. This includes the overhead of computing two evaluations of the score function per step, as well as a single backward pass to differentiate the importance weighting function.

This added cost may be acceptable for scenarios where the goal is to capture rare samples. In such regimes, conventional generative models often fail to sample from the critical low-density regions, which dominate tail-risk measures.

To quantify the cost-benefit trade-off, we report empirical results in Figure 4. Each plot shows the CVaR performance (vertical axis) as a function of the number of samples used (horizontal axis) for different values of β .

As shown in Figure 4, when $\beta = 0.99$, the proposed method achieves better CVaR performance even with only $1/8$ the number of samples compared to baseline methods. Similar trends are observed for $\beta = 0.95$, underscoring the importance of sampling from rare, high-risk regions. As β decreases (e.g., $\beta = 0.80$ or $\beta = 0.50$), the importance of extreme loss values diminishes, and a larger sample budget becomes necessary to achieve better performance over baselines.

E.2 Impact of Importance Level Emphasis

We investigate how the choice of the importance weighting function φ affects performance.

Square-root emphasis. Consider the optimization-noise term

$$\hat{v}(q) = \mathbb{E}_{\mathbf{X}^p \sim p} \left[\frac{w^*(\mathbf{X}^p)^2 p(\mathbf{X}^p)}{(1 - \beta)^2 q(\mathbf{X}^p)} + 1 \right].$$

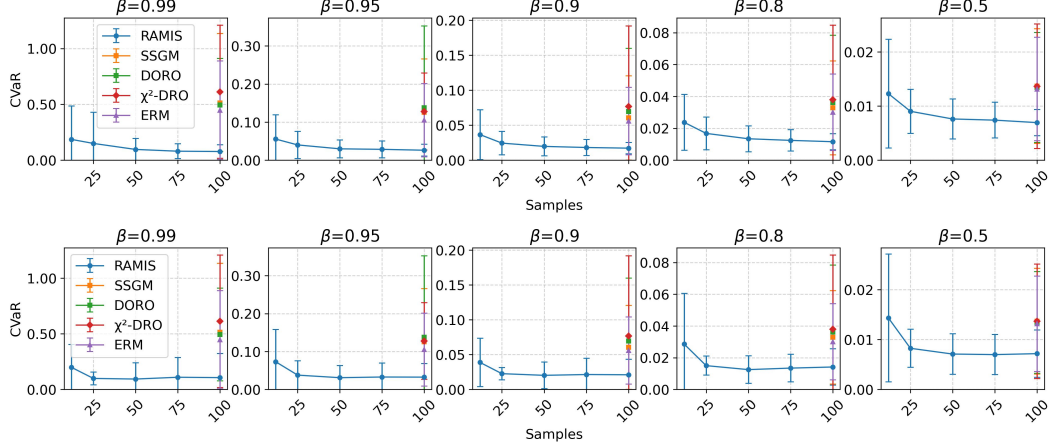


Figure 4: CVaR performance versus sample size for different β . Top: $\varphi(x) = x^{1/2}$. Bottom: $\varphi(x) = x$. The proposed method outperforms baselines in the high- β regime (e.g., $\beta = 0.99, 0.95$) with fewer than half the samples.

Then the desired importance weight is $q(\mathbf{x}) \propto w^*(\mathbf{x})p(\mathbf{x})$; substituting this choice yields a Jensen-type inequality as

$$\hat{v}(p) = \frac{1}{(1-\beta)^2} \mathbb{E}_{\mathbf{X}^p \sim p} [w^*(\mathbf{X}^p)^2] + 1 \geq \frac{1}{(1-\beta)^2} \mathbb{E}_{\mathbf{X}^p \sim p} [w^*(\mathbf{X}^p)] \mathbb{E}_{\mathbf{X}^p \sim p} [w^*(\mathbf{X}^p)] + 1. \quad (42)$$

Based on this, we consider $\varphi(x) = \sqrt{x} + c$, where $c \geq 0$ is a scalar hyperparameter that adjusts the relative emphasis on high-loss regions. Intuitively, smaller values of c amplify the contrast between low- and high-loss samples, focusing the generative model more aggressively on rare, high-risk regions. In contrast, larger values of c can flatten the weight distribution, thereby reducing the selectivity of the sampling process. To empirically examine this effect, we evaluate the performance trend across various c from 0.001 to 10, denoted “RAMIS+ c ” in the left column of Figure 5

Figure 5 shows the CVaR performance under various β . For $\beta \in \{0.99, 0.95\}$, all φ variants significantly outperform the risk minimization and robust optimization baselines. Notably, configurations with $c \in \{0.001, 0.01, 0.1\}$ achieve the best results, improving CVaR by over 30% relative to the strongest baseline. Larger offsets, $c \geq 1$, degrade performance by reducing the dynamic range of the resulting importance weights.

Linear emphasis. We also consider the following linear choice to mimic the behavior of w^* as $\varphi(x) = x + c$. As shown in the right column of Fig. 5, the best performance is achieved for small constants $c \leq 0.1$. In this regime, the proposed method outperforms baselines that do not use importance samples across all settings. In contrast, setting $c > 0.1$ diminishes the emphasis on high-loss samples and results in degraded performance, approaching that of SSGM and the other baselines without importance sampling.

These results indicate that the additive constant c in φ can serve as a simple hyperparameter controlling the strength of importance sampling. Smaller c sharpens focus on high-risk inputs (beneficial for tail-sensitive objectives), whereas overly large c flattens the weights.

Limitations. While these experiments illustrate performance trends with respect to the choice of the weighting function, one limitation is that constructing the importance distribution via guided sampling may introduce sampling-approximation error in the score-based generative model. In addition, the analysis controls stochastic noise through an upper-bound surrogate. Exploring guidance procedures with reduced approximation error and alternative weight functions φ informed by tighter bounds may improve the quality of the importance sampling and its empirical performance.

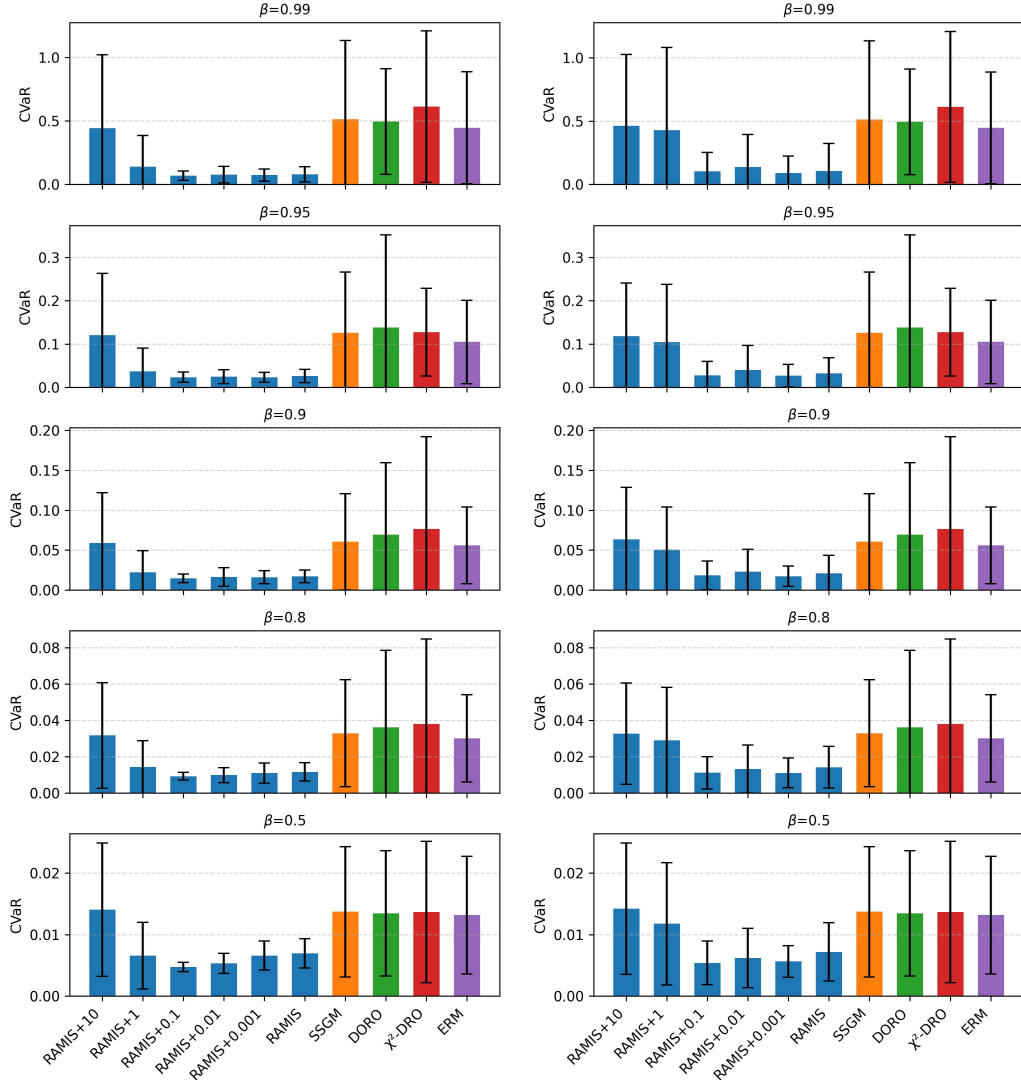


Figure 5: CVaR performance for different values of c . For RAMIS, left: $\varphi(x) = x^{1/2} + c$; right: $\varphi(x) = x + c$.