Conformal Reliability: A New Evaluation Metric for Conditional Generation

Anonymous authors
Paper under double-blind review

Abstract

Conditional generative models have recently achieved remarkable success in various applications. However, a suitable metric for evaluating the reliability of these models, which takes into account their inherent uncertainty, is still lacking. Existing metrics, which typically assess a single output, may fail to capture the variability or potential risks in generation. In this paper, we propose a novel evaluation metric called *reliability score* based on conformal prediction, which measures the worst-case performance within the prediction set at a pre-specified confidence level. However, computing this score is challenging due to the high-dimensional nature of the output space and the nonconvexity of both the metric function and the prediction set. To efficiently compute this score, we introduce Conformal ReLiability (CReL), a framework that can **(i)** construct the prediction set with desired coverage; and **(ii)** accurately optimize the reliability score within the constructed prediction set. We provide theoretical results on coverage and demonstrate empirically that our method produces more informative prediction sets than existing approaches. Experiments on synthetic data and on the image-to-text and text-to-image tasks further demonstrate the interpretability of our new metric, and the validity and effectiveness of our computational framework.

1 Introduction

Conditional generative models map a given input condition (*e.g.*, a textual prompt) to high-dimensional outputs (*e.g.*, images, sequences). Powered by large-scale datasets and models, this paradigm underpins breakthroughs in diverse domains, from text-to-image synthesis Reed et al. (2016) and drug discovery Bian & Xie (2021) to autonomous systems Gasparyan & Qiu (2024). Yet, despite their remarkable generative prowess, a fundamental question remains largely unanswered: *how trustworthy are these models when deployed in the real world*?

Current metrics such as the CLIP score Radford et al. (2021) typically assess a single generated output. While effective for measuring *average* capability, this paradigm fundamentally ignores the stochastic nature of generative models, masking potential risks hidden within the distribution of plausible outputs. A model may achieve a high average score by frequently generating high-quality samples, yet still harbor a non-negligible probability of producing catastrophic failures. For instance, in an image-to-text task, a model might correctly caption an image as “A man playing the guitar” in most cases, but under different sampling seeds, it could plausibly generate “A man pointing a gun” due to visual ambiguity or hallucinations. In safety-critical applications Gawlikowski et al. (2023); He et al. (2023), reliability is defined not by how good the model *can* be, but by how bad it *might* be. Therefore, single-output evaluation is insufficient; a rigorous framework must quantify the *worst-case performance* among all statistically plausible outputs.

To provide such an assessment, we propose the *reliability score*, which evaluates the worst-case value of a user-specified similarity metric ρ within a calibrated prediction set at confidence level $1-\alpha$. In classical Conformal Prediction (CP), valid models are commonly compared using the geometric size (volume) of the prediction set—a notion of *sharpness*. However, in high-dimensional generative tasks, set volume is not only computationally intractable but also misaligned with the evaluation metric ρ: geometric size provides no information about how close the set remains to the ground truth under the metric of interest. To bridge this gap, our reliability score replaces geometric volume with a *metric-aware measure of sharpness*, defined as the worst achievable performance under ρ among all statistically plausible outputs. This quantity directly reflects the model's performance *floor* at confidence level $1 - \alpha$. Yet, computing this score presents a significant challenge: directly optimizing the worst-case performance is intractable due to the high dimensionality of generative outputs, as well as the nonconvexity of both the similarity metric ρ and the prediction set serving as the constraint.

For the regression task with multi-dimensional output, one can apply directional quantile regression (DQR) Kong & Mizera (2012); Paindaveine & Šiman (2011), which can yield a convex prediction set. However, this set may be overly conservative, as it must account for extreme cases in order to guarantee coverage. Besides, the prediction set may not be informative since the true set may not be convex. Other methods Xu et al.; Javanmard et al. (2025); Feldman et al. (2023) leveraged conformal calibration, which can alleviate some of these issues. For instance, Xu et al. modeled outputs as Gaussian mixtures or projected them into lower-dimensional latent spaces. Yet, none tackle the optimization of worst-case reliability under a general similarity metric—a problem that is both computationally and statistically intractable in high-dimensional output spaces.

To address these limitations, we introduce *Conformal ReLiability* (CReL), a principled computational framework for quantifying the reliability of conditional generative models. At its core, CReL projects high-dimensional outputs into a structured latent space, wherein both conformal calibration and optimization are performed. Compared to existing approaches that calibrate in the original output space, our method enjoys better computational efficiency and optimization tractability for computing the reliability score. Theoretically, we establish that the resulting prediction set satisfies the target guarantee. Moreover, reformulating the objective over the latent-space prediction set transforms the problem into an optimization program with convex constraints, on which the projection operation can be efficiently computed using linear programming. This formulation enables the employment of projected gradient descent, endowed with provable global convergence guarantees for computing the reliability score. We demonstrate the validity and effectiveness of our framework on synthetic data and on both the image-to-text and text-to-image tasks.

To summarize, our contributions are:

- *Reliability-Centric Metric*: We introduce the reliability score to quantify the worst-case performance of conditional generative models at a specified confidence level, addressing risks overlooked by single-sample evaluations.
- *CReL Framework*: we develop Conformal ReLiability (CReL), a computational framework that can efficiently and accurately compute the reliability score.
- *Theoretical Guarantee*: We show that the prediction set generated by our method meets the coverage guarantee. Additionally, we empirically find that the prediction set given by our procedure has much smaller or comparable size to other methods, highlighting the effectiveness of our approach in delivering more informative calibration.
- *Empirical Validation*: We evaluate our methods on synthetic data and on both the image-to-text and text-to-image tasks. For synthetic data, we validate the effectiveness of our computational framework. In the image-to-text and text-to-image tasks, we demonstrate that our new metric provides more interpretable evaluations compared to traditional single-output metrics.

2 Related Works

Evaluating condition generative models. Typical metrics for evaluating conditional generative models include *Structural Similarity Index Measure* (SSIM) Wang et al. (2004), *Contrastive Language-Image Pretraining* (CLIP) Radford et al. (2021), and others. Specifically, SSIM evaluates the structural similarity between generated images and reference images, while CLIP measures the similarity between generated images and corresponding textual descriptions by projecting both modalities into a shared embedding space and calculating the cosine similarity between them. Other popular metrics, like BERT-similarity Kenton & Toutanova (2019) and *Fréchet Inception Distance* (FID) Heusel et al. (2017), also rely on embedding models to quantify how well the generated samples match the given conditions. Recent works have highlighted the limitations of these metrics. For class-conditional generation, Benny et al. (2021) proposed conditional versions (e.g., WCFID, BCFID) to measure average performance within each class and between all class averages. For the broader text-conditional setting, HEIM Lee et al. (2023) introduced a holistic benchmark to evaluate 12 aspects (e.g., alignment, robustness, bias), noting that simple automated metrics often correlate weakly with human preferences. Concurrently, other metrics like the Conditional-Vendi score Jalali et al. (2025) and Scendi score Ospanov et al. (2025) were developed to disentangle true model-induced diversity from prompt-induced diversity, using information theory or Schur complements, respectively. However, while these approaches offer richer evaluations of average-case performance or output diversity, they do not explicitly quantify reliability. Metrics like CLIP assess only single outputs, ignoring variability; meanwhile, distributional metrics (e.g., WCFID, Scendi) focus on aggregate properties (quality or diversity) rather than the risk of failure. Crucially, none of these metrics evaluate the reliability of the generative model, which is the focus of our work.

Conformal prediction for multi-dimensional data. Many works have been done on this topic recently. Kong & Mizera (2012); Boček & Šiman (2017) proposed directional quantile regression (DQR) that estimated quantile hyperplanes for multiple directions in the response space. However, this approach may remain conservative and uninformative, since the prediction set is constrained to be convex and requires estimating extreme quantiles to ensure coverage. While the vector quantile regression Carlier et al. (2016) can produce non-convex sets, it restricts the output to linearly depend on the input. Other attempts include Messoudi et al. (2022); Xu et al. that constructed the prediction set as an ellipsoidal set, Wang et al. (2022b); Johnstone & Ndiaye (2022); Gibbs et al. (2025); Plassier et al. (2024) that modeled the conditional distribution of the output, and Plassier et al. (2025) which rectifies scalar conformity scores via quantile regression to improve conditional coverage. In particular, Feldman et al. (2023) proposed to map the high-dimensional response into a lower-dimensional latent space, which can alleviate the conservativeness problem when applying DQR. While these CP methods aim to construct accurate prediction sets, our work introduces a different goal: evaluating a reliability score. This presents a new computational challenge, as calculating this score requires an optimization step. However, this step has not been considered and is not feasible by applying previous works. For instance, while Feldman et al. (2023) also mapped the original output into the latent space, it performed calibration in the output space and produced a non-convex prediction set therein. As a result, optimization becomes intractable due to the non-convexity of both the similarity metric and the prediction set. Our key novelty lies in leveraging the LGM and calibration to construct a convex latent-space set, over which we optimize the reliability score, thereby making the problem computationally tractable.

3 Methodology

We aim to evaluate the reliability of a target model $f : \mathbb{R}^p \to \mathbb{R}^d$ in conditional regression tasks, with respect to a user-defined similarity metric ρ, where higher values of ρ indicate better performance. Suppose our data has n independent and identically distributed (*i.i.d.*) samples $\{(X_i, Y_i)\}_{i=1}^n$, where $X \in \mathbb{R}^p$ represents the input condition (*e.g.*, prompt) and $Y \in \mathbb{R}^d$ denotes the ground-truth output. For each X_i, the target model f generates the output $\widehat{Y}_i := f(X_i)$.

Our goal is to assess the reliability of f for a new observation X_{n+1} with respect to ρ. Given a confidence level $\alpha \in (0,1)$, we aim to quantify the worst-case performance at confidence level $1-\alpha$:

$$\min_{\widehat{Y} \in C_{\mathcal{Y}}(X_{n+1})} \rho(\widehat{Y}, \mathrm{GT}_{n+1}), \text{ such that } \mathbb{P}\left(\widehat{Y}_{n+1} \in C_{\mathcal{Y}}(X_{n+1})\right) \geq 1-\alpha, \tag{1}$$

where GT_{n+1} denotes the ground truth response, *i.e.*, X_{n+1} (*e.g.*, CLIP similarity between the generated text and the image) or Y_{n+1} (*e.g.*, BERT similarity between the true text and the generated image). This metric provides a robust, uncertainty-aware lower bound on performance, evaluating the reliability of the method in the worst-case allowed by the confidence level $1-\alpha$.

Because $\widehat{Y}$ is normally high-dimensional, applying directional quantile regression (DQR) can result in overly conservative prediction sets, which may hinder an accurate assessment of reliability. To address this issue, we introduce *Conformal ReLiability* (CReL), a conformal framework built on the latent generative model, which allows both the calibration procedure and the optimization of $\rho(\cdot,\cdot)$ to be performed in a much lower-dimensional latent space (as illustrated in Fig. 1). This approach yields more informative prediction sets and, consequently, more accurate reliability evaluation.

The rest of this section is organized as follows: Section 3.1 first introduces our conformal procedure on the latent space. Then, we will show in Section 3.2 that such a procedure can meet the coverage

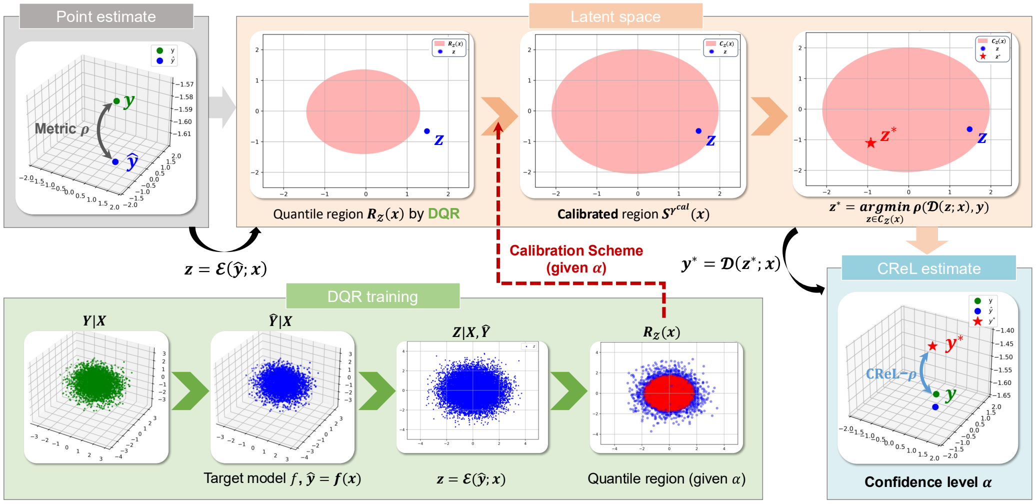

Point estimate
Metric ρ
y
$\hat{y}$
Latent space
$R_{\mathcal{Z}}(x)$
z
Quantile region $R_{\mathcal{Z}}(x)$ by DQR
$C_{\mathcal{Z}}(x)$
Calibrated region $S^{\gamma^{cal}}(x)$
z^*
$z^* = \underset{z \in C_{\mathcal{Z}}(x)}{argmin}\, \rho(\mathcal{D}(z;x), y)$
$z = \mathcal{E}(\hat{y}; x)$
Calibration Scheme (given α)
$y^* = \mathcal{D}(z^*; x)$
CReL estimate
DQR training
$Y|X$
$\hat{Y}|X$
$Z|X, \hat{Y}$
$R_{\mathcal{Z}}(x)$
Target model f, $\hat{y} = f(x)$
$z = \mathcal{E}(\hat{y}; x)$
Quantile region (given α)
y^*
CReL$-\rho$
Confidence level α

Figure 1: Illustration of our procedure. During DQR training, the latent generative model maps the target model's prediction space $\widehat{\mathcal{Y}}$ to a latent space $\mathcal{Z}$, where DQR constructs the quantile region $R_{\mathcal{Z}}(x)$ (3) using the loss (2). Then, CReL applies the calibration to adjust $R_{\mathcal{Z}}(x)$, such that the calibrated region $S^{\gamma_{\text{cal}}}(x)$ satisfies the marginal coverage at level $1 - \alpha$. The final reliability metric CReL-ρ is then computed by optimizing (8).

guarantee as long as the latent generative model is trained well. Finally, Section 3.3 introduces our optimization methods for computing the reliability score.

3.1 Conformal Calibration

The key insight of our framework is to learn an embedding space $\mathcal{Z}$ via a latent generative model, enabling conformal calibration in such a lower-dimensional space and thereby significantly reducing the over-conservativeness present in the original output space. Within this latent space, we construct $C_{\mathcal{Z}}(X_{n+1})$ through conformal calibration following DQR models Kong & Mizera (2012), transforming the original non-convex reliability optimization problem into a computationally tractable convex-constrained optimization.

We begin by partitioning the training indices into three folds: $\mathcal{I}_{\text{lgm}}$ for training the latent generative model, $\mathcal{I}_{\text{dqr}}$ for training the DQR model, and $\mathcal{I}_{\text{cal}}$ for final conformal calibration. We denote the corresponding datasets as $\mathcal{D}_{\text{lgm}}$, $\mathcal{D}_{\text{dqr}}$, and $\mathcal{D}_{\text{cal}}$, respectively.

Step 1: Training the latent generative model. The model is composed of an encoder that transforms $\widehat{Y}|X = x$ into a latent distribution Z_x and constructs $C_{\mathcal{Z}}(X)$, followed by a decoder that gives the prediction set $C_{\mathcal{Y}}(X) := \mathcal{D}ec(C_{\mathcal{Z}}(X), X)$. It is ensured in Theorem 3.3 that $C_{\mathcal{Y}}(X)$ meets the coverage guarantee, as long as the latent generative model (LGM) can well recover the distribution $\widehat{Y}|X$. Typical choices of generative models satisfying this property include the *Variational Autoencoder* (VAE) Khemakhem et al. (2020); Kingma & Welling (2013) or the stable diffusion model (Rombach et al., 2022). To this end, we fit an LGM on $\{(X_i, \widehat{Y}_i)\}_{i \in \mathcal{I}_{\text{lgm}}}$. After training, we can obtain an encoder $\mathcal{E}(\cdot, \cdot) : \widehat{\mathcal{Y}} \times \mathcal{X} \mapsto \mathcal{Z}$ and the decoder $\mathcal{D}ec(\cdot, \cdot) : \mathcal{Z} \times \mathcal{X} \mapsto \mathcal{Y}$. To align the encoder and decoder with the similarity metric ρ, we replace the mean square loss $\|\widehat{Y} - \mathcal{D}ec(\mathcal{E}(\widehat{Y}, X), X)\|_2^2$ with $\rho(\widehat{Y}, \mathcal{D}ec(\mathcal{E}(\widehat{Y}, X), X))$ during training.

Step 2: Fitting the DQR model. After training the LGM, we use the encoder $\mathcal{E}$ to obtain (Z_i, X_i) from $(\widehat{Y}_i, X_i)$ for each $i \in \mathcal{I}_{\text{dqr}}$, where $Z_i := \mathcal{E}(\widehat{Y}_i; X_i) \in \mathbb{R}^r$. Then, we apply the DQR Kong & Mizera (2012) on $\{Z_i, X_i\}_{\mathcal{I}_{\text{dqr}}}$ to obtain an initialized region $R_{\mathcal{Z}}(x)$ for any x in the calibration dataset. Specifically, given a direction $\mathbf{u} \in \mathbb{S}^{r-1} := \{\mathbf{u} \in \mathbb{R}^r : \|\mathbf{u}\|_2 = 1\}$, DQR models the quantiles of a response vector in $\mathbf{u}$, allowing us to estimate the α-th quantile for any input X_i by projecting the response vector Z_i onto $\mathbf{u}$. Specifically, DQR minimizes the following objective (2) to estimate the α-th quantile of the projection $\mathbf{u}^\intercal Z_i$ given X_i:

$$\widehat{\beta} = \frac{1}{|\mathcal{D}_{\text{dqr}}|} \sum_{i \in \mathcal{D}_{\text{dqr}}} \zeta_\alpha \left(\mathbf{u}^\intercal Z_i, f_{\boldsymbol{\beta}}(X_i, \mathbf{u}) \right), \tag{2}$$

where the pinball loss $\zeta_\alpha(y, \hat{y})$ is defined as:

$$\zeta_\alpha(y, \hat{y}) = \begin{cases} \alpha(y - \hat{y}) & \text{if } y - \hat{y} > 0, \\ (1-\alpha)(\hat{y} - y) & \text{otherwise.} \end{cases}$$

Here, $f_{\boldsymbol{\beta}}(X_i, \mathbf{u})$ represents the regression function parameterized by $\boldsymbol{\beta}$, which predicts the value of the α-th quantile for the projected data. For each direction $\mathbf{u}$, DQR defines a convex half-space $\mathbb{H}_u^+(x) = \{z \in \mathbb{R}^r : \mathbf{u}^\intercal z \geq f_{\boldsymbol{\beta}}(x, \mathbf{u})\}$. The quantile region is then obtained by taking the intersection of all such half-spaces across all directions $\mathbf{u} \in \mathbb{S}^{r-1}$:

$$R_{\mathcal{Z}}(x) = \bigcap_{\mathbf{u} \in \mathbb{S}^{r-1}} \mathbb{H}_u^+(x), \tag{3}$$

which yields a convex region in the latent space $\mathcal{Z}$. The DQR fitting procedure is illustrated in Fig. 1, where the quantile region $R_{\mathcal{Z}}(x)$ constructed in $\mathcal{Z}$ during training is marked in red.

Step 3: Calibration. When the latent dimension $r > 1$, $R_{\mathcal{Z}}(X)$ covers strictly less than $1 - \alpha$ of the distribution, due to the intersection of the half-spaces. To address this, we perform calibration to construct $C_{\mathcal{Z}}(X)$ on $\mathcal{D}_{\text{cal}} := \{(X_i, Z_i)\}_{i \in \mathcal{I}_{\text{cal}}}$, such that $\mathbb{P}\left(Z_{n+1} \in C_{\mathcal{Z}}(X_{n+1})\right) \geq 1 - \alpha$. To this end, we first define a base region:

$$S^\gamma(x) = \left\{ z \in \mathbb{R}^r : \min_{a \in R_{\mathcal{Z}}(x)} d(a, z) \leq \gamma \right\}, \tag{4}$$

where $d(\cdot, \cdot)$ a distance function. The goal is to find a γ_{cal} such that $C_{\mathcal{Z}}(X) = S^{\gamma_{\text{cal}}}(X)$. To achieve the target coverage, we expect γ_{cal} to be the $(1-\alpha)$-quantile of the distribution function over $\min_{a \in R_{\mathcal{Z}}(X)} d(a, \cdot)$. We first obtain the coverage rate γ_{init} of the uncalibrated base regions (*i.e.*, $S^0(X_i) = R_{\mathcal{Z}}(X_i)$):

$$\gamma_{\text{init}} = \frac{1}{|\mathcal{D}_{\text{cal}}|} \left| \{Z_i : Z_i \in R_{\mathcal{Z}}(X_i), i \in \mathcal{I}_{\text{cal}} \right|, \tag{5}$$

where $Z_i = \mathcal{E}(\widehat{Y}_i; X_i)$ for each $i \in \mathcal{I}_{\text{cal}}$. Since the coverage of $R_{\mathcal{Z}}(X_i)$ is strictly less than $1 - \alpha$ when $r > 1$, we would have $\gamma_{\text{init}} \leq 1 - \alpha$ as long as the sample size of the calibration data, *i.e.*, $|\mathcal{I}_{\text{cal}}|$ is sufficiently large. That means, to achieve coverage guarantee, we should grow the base quantile region by computing γ_{cal} as

$$\begin{aligned} &E_i^+ = \min_{a \in R_{\mathcal{Z}}(X_i)} d(a, Z_i), \forall i \in \mathcal{I}_{\text{cal}}, \\ &\gamma_{\text{cal}} := \lceil (|\mathcal{D}_{\text{cal}}| + 1)(1 - \alpha) \rceil\text{-th smallest value of } \{E_i^+ : i \in \mathcal{I}_{\text{cal}}\}. \end{aligned} \tag{6}$$

Finally, the calibrated quantile region $C_{\mathcal{Z}}(X)$ is given by $S^{\gamma_{\text{cal}}}(X)$ in (4).

Remark 3.1. Unlike Feldman et al. (2023), which performs calibration in the output space $\mathcal{Y}$, we calibrate directly in the latent space $\mathcal{Z}$. This is motivated by computational and optimization considerations. Specifically, Feldman et al. (2023) calibrates on $R_{\mathcal{Y}}(X) = \mathcal{D}ec(R_{\mathcal{Z}}(X), X)$. Since $R_{\mathcal{Y}}(X)$ may be non-convex, calibration requires discretization, which can be computationally expensive, particularly in high-dimensional spaces. In contrast, $R_{\mathcal{Z}}(X)$ is convex, allowing the core E_i^+ to be computed efficiently via linear programming. Please refer to Appendix D for more details about computational complexity. Furthermore, direct optimizing (1) over $C_{\mathcal{Y}}(X_{n+1})$ can be intractable, as both $C_{\mathcal{Y}}(X_{n+1})$ and $\rho(\cdot, \cdot)$ are non-convex. In contrast, as we will show, the optimization can be reformulated into one that optimizes in the latent space $C_{\mathcal{Z}}(X_{n+1}) = S^{\gamma_{\text{cal}}}(X_{n+1})$. Because this space is convex and compact, the optimization becomes more tractable.

Step 4: Constructing $C_{\mathcal{Y}}(X_{n+1})$. Our final prediction set is given by $C_{\mathcal{Y}}(X_{n+1}) := \mathcal{D}ec(S^{\gamma_{\text{cal}}}(X_{n+1}), X_{n+1})$. Alg. 1 summarizes the overall procedure for calibration.

3.2 Theoretical guarantee

In this section, we provide the coverage guarantee for $C_{\mathcal{Y}}(X_{n+1})$. First, we show that after calibration, $C_{\mathcal{Z}}(X_{n+1}) = S^{\gamma_{\text{cal}}}(X_{n+1})$ satisfies the coverage guarantee in the latent space.

Algorithm 1 Conformal ReLiability

```
Input: Dataset $\{(X_i, Y_i)\}_{i=1}^n$, target model $f$, similarity metric $\rho$, nominal confidence level $\alpha \in (0,1)$, encoder $\mathcal{E}(\cdot,\cdot)$ and decoder $\mathcal{D}ec(\cdot,\cdot)$ in the latent generative model, DQR algorithm, a test point $X_{n+1}$.
Output: $C_{\mathcal{Y}}(X_{n+1})$.
Training time:
  1: Split $\{1,\cdots,n\}$ into three disjoint sets $\mathcal{I}_{\mathrm{lgm}}, \mathcal{I}_{\mathrm{dqr}}, \mathcal{I}_{\mathrm{cal}}$.
  2: Train a latent generative model on $\{(X_i, \widehat{Y}_i)\}_{i\in\mathcal{I}_{\mathrm{lgm}}}$, where $\widehat{Y}_i := f(X_i)$ for each $i = 1,\ldots,n$.
  3: Fit a DQR model on $\{(X_i, Z_i)\}_{i\in\mathcal{I}_{\mathrm{dqr}}}$, where $Z_i := \mathcal{E}(\widehat{Y}_i; X_i)$, and to obtain $R_{\mathcal{Z}}(x)$ (3).
Calibrating time:
  1: Compute the coverage of the uncalibrated quantile regions on $\{(X_i, Z_i)\}_{i\in\mathcal{I}_{\mathrm{cal}}}$ via (5).
  2: Compute $E_i^+$ and obtain $\gamma_{\mathrm{cal}}$ according to (6).
Test time:
  1: Obtain a base quantile region $R_{\mathcal{Z}}(X_{n+1})$ using a pre-trained DQR model.
  2: Construct the calibrated quantile region $S^{\gamma_{\mathrm{cal}}}(X_{n+1})$ according to (4).
  3: Construct $C_{\mathcal{Y}}(X_{n+1}) = \mathcal{D}ec(S^{\gamma_{\mathrm{cal}}}(X_{n+1}), X_{n+1})$.
```

Proposition 3.2. *Suppose data in $\mathcal{D}_{\mathrm{lgm}}$, $\mathcal{D}_{\mathrm{dqr}}$, and $\mathcal{D}_{\mathrm{cal}} \cup \{X_{n+1}, Y_{n+1}\}$ are independent to each other. Besides, we assume $\{X_i, Y_i\}_{i\in\mathcal{I}_{\mathrm{cal}}} \cup (X_{n+1}, Y_{n+1})$ are exchangeable. Given a nominal coverage level $\alpha \in (0,1)$, the quantile region $S^{\gamma_{\mathrm{cal}}}(X_{n+1})$ given by Alg. 1 satisfies:*

$$1 - \alpha \le \mathbb{P}\left(Z_{n+1} \in S^{\gamma_{\mathrm{cal}}}(X_{n+1})\right) \le 1 - \alpha + \frac{1}{1 + |\mathcal{D}_{\mathrm{cal}}|}.$$

Proof. The goal is to show that $\{E_i^+\}_{i\in\mathcal{I}_{\mathrm{cal}}} \cup E_{n+1}^+$ are exchangeable. First, since f has been trained and is fixed, $\{(X_i, \widehat{Y}_i)\}_{i\in\mathcal{I}_{\mathrm{cal}}\cup\{n+1\}}$ are exchangeable. Since for each $i \in \mathcal{I}_{\mathrm{cal}}$, Z_i is obtained from $\mathcal{E}(\widehat{Y}_i, X_i)$, and the encoder $\mathcal{E}(\cdot,\cdot)$ is trained on $\mathcal{D}_{\mathrm{lgm}}$ that are independent to $\mathcal{D}_{\mathrm{cal}}$, we have $\{(X_i, Z_i)\}_{i\in\mathcal{I}_{\mathrm{cal}}\cup\{n+1\}}$ are exchangeable. Since E_i^+ for each $i \in \mathcal{I}_{\mathrm{cal}}$ is determined by $\mathcal{D}_{\mathrm{dqr}}$ that are independent to $\mathcal{D}_{\mathrm{cal}} \cup \{X_{n+1}, Y_{n+1}\}$, $\{E_i^+\}_{i\in\mathcal{I}_{\mathrm{cal}}} \cup E_{n+1}^+$ are exchangeable. □

Using this property, we can further demonstrate that, provided the latent generative model accurately recovers the conditional distribution $Y|X = x$, the resulting prediction set $C_{\mathcal{Y}}(X_{n+1})$ also satisfies the desired coverage.

Theorem 3.3. *Assume conditions in proposition 3.2 hold. Besides, we assume that $\forall x \in \mathcal{X}$, $\mathcal{D}ec(\mathcal{E}(\widehat{Y}, x), x) =_d \widehat{Y}|X = x$. Given any nominal coverage level $\alpha \in (0,1)$, $C_{\mathcal{Y}}(X_{n+1}) := \mathcal{D}ec(S^{\gamma_{\mathrm{cal}}}(X_{n+1}), X_{n+1})$ given by Alg. 1 satisfies:*

$$\mathbb{P}\left(\widehat{Y}_{n+1} \in C_{\mathcal{Y}}(X_{n+1})\right) \ge 1 - \alpha.$$

Proof. First, by proposition 3.2, we have $\mathbb{P}(Z_{n+1} \in S^{\gamma_{\mathrm{cal}}}(X_{n+1})) \ge 1 - \alpha$. Since $Z_{n+1} \in S^{\gamma_{\mathrm{cal}}}(X_{n+1}) \implies \mathcal{D}ec(Z_{n+1}, X_{n+1}) \in \mathcal{D}ec(S^{\gamma_{\mathrm{cal}}}(X_{n+1}), X_{n+1})$, we have:

$$\mathbb{P}\left(\mathcal{D}ec(Z_{n+1}, X_{n+1}) \in \mathcal{D}ec(S^{\gamma_{\mathrm{cal}}}(X_{n+1}), X_{n+1})\right) \overset{(1)}{\ge} \mathbb{P}(Z_{n+1} \in S^{\gamma_{\mathrm{cal}}}(X_{n+1})) \ge 1 - \alpha. \quad (7)$$

Since $\mathcal{D}ec(\mathcal{E}(\widehat{Y}, x), x) =_d \widehat{Y}|X = x$, we further have

$$\mathbb{P}(\widehat{Y}_{n+1} \in C_{\mathcal{Y}}(X_{n+1})) = \mathbb{P}\left(\mathcal{D}ec(\mathcal{E}(\widehat{Y}_{n+1}, X_{n+1}), X_{n+1}) \in C_{\mathcal{Y}}(X_{n+1})\right) \ge 1 - \alpha.$$

This completes the proof. □

Remark 3.4. Compared to Feldman et al. (2023), which performs calibration directly in the original output space $\mathcal{Y}$, the prediction set $C_{\mathcal{Y}}(X_{n+1})$ generated by our method may be slightly more conservative in terms of coverage. This is due to the effect described in "(1)" of (7), where the decoder can expand the region. Nevertheless, this slight increase in conservativeness is a worthwhile trade-off, as it facilitates optimization when computing the reliability score. Moreover, as demonstrated empirically, the degree of overconservativeness is minor, *i.e.*, the size of the resulting region is comparable to that reported in (Feldman et al., 2023).

The assumption that LGM can well recover the conditional distribution has been similarly made in (Feldman et al., 2023). This property can hold for many types of latent generative models, including the variational autoencoder Khemakhem et al. (2020), and the stable diffusion model (Rombach et al., 2022; Li et al., 2023).

3.3 Optimization

After constructing $C_{\mathcal{Z}}(X_{n+1}) := S^{\gamma_{\mathrm{cal}}}(X_{n+1})$ and $C_{\mathcal{Y}}(X_{n+1})$, we are ready to compute the reliability (1) given a metric ρ. Since $C_{\mathcal{Y}}(X_{n+1}) := \mathcal{D}ec(C_{\mathcal{Z}}(X_{n+1}), X_{n+1})$, it is equivalent to consider the following objective:

$$\min_{z \in C_{\mathcal{Z}}(X_{n+1})} \rho\left(\mathcal{D}ec(z; X_{n+1}), \mathrm{GT}_{n+1}\right). \tag{8}$$

Compared with the original objective—where both ρ and the constraint set may be nonconvex—the feasible region $C_{\mathcal{Z}}(X_{n+1})$ is convex and compact, as shown below. As such, objective (8) falls into the category of nonconvex optimization over a convex set, as studied in (Lacoste-Julien, 2016).

Proposition 3.5. *If $R_{\mathcal{Z}}(x)$ is convex and $d(\cdot,\cdot)$ is jointly convex, $S^{\gamma}(x)$ is a convex and compact set for any γ.*

Remark 3.6. The joint convexity can hold for any norm-induced distance (*i.e.*, $d(a,b) := \|a-b\|$), Bregman divergence, or f-divergence. In this paper, we choose $d(a,b)$ to be the Euclidean distance.

Moreover, by (6), it is easy to see that the projection onto $S^{\gamma}(x)$ can be efficiently solved via a linear programming algorithm. Specifically, suppose $d(x,y) := \|x-y\|_2$. To compute $\Pi_{S^{\gamma_{\mathrm{cal}}}(x)}(y) := \arg\min_{z \in S^{\gamma_{\mathrm{cal}}}(x)} \|y-z\|_2$ given a new point y to be projected, we can see that:

$$\Pi_{S^{\gamma_{\mathrm{cal}}}(x)}(y) = \begin{cases} y & \text{if } y \in S^{\gamma_{\mathrm{cal}}}(x) \\ y^* + \gamma_{\mathrm{cal}} \frac{y^*-y}{\|y^*-y\|_2} & \text{otherwise,} \end{cases}$$

where $y^* := \arg\min_{y_1 \in R_{\mathcal{Z}}(x)} \|y_1 - y\|_2$. It is then sufficient to compute $\Pi_{R_{\mathcal{Z}}(x)}(y)$, which can be formulated as the linear programming problem as follows:

$$\min_{y_1} \|y_1 - y\|_2^2, \text{ subject to } \mathbf{u}_k^\top y_1 \geq f_{\widehat{\beta}}(x, \mathbf{u}_k) \text{ for each } k = 1, \ldots, K,$$

where $\widehat{\beta}$ is obtained after DQR training, and $R_{\mathcal{Z}}(x)$ is constructed using K directions $\mathbf{u}_1, \ldots, \mathbf{u}_K$. As the projection can be efficiently computed, we can implement projected gradient descent Ghadimi et al. (2016); Ghadimi & Lan (2016), whose global convergence property has been established (Ghadimi & Lan, 2016). To find the global optimal, we use random search to pick several starting points and then apply the projected gradient descent to the initial that has the smallest ρ.

4 Experiments

We evaluate our method on synthetic data and the benchmark MS-COCO 2014 Lin et al. (2014) for the image-to-text and text-to-image tasks.

4.1 Experiments on Synthetic Datasets

Setups. We consider the *nonlinear* synthetic setting (see Appendix B.1 for generation details). We generate $50,000$ samples and set $p = 38$, $d = 2$, and $\epsilon = 0.3$. The dataset is split as follows: 60% for training the latent generative model, 24% for training the DQR, 8% for calibration, and 8% for testing. By default, we adopt the Mean Squared Error (MSE) as the similarity metric ρ, consistent with the standard VAE reconstruction loss. We report the average coverage ratios of $C_{\mathcal{Z}}(X)$ and $C_{\mathcal{Y}}(X)$, as well as the area (defined as the number of grid points falling into the region) of the calibration region $C_{\mathcal{Y}}(X)$. For comparison, we also report the coverage ratios and areas of the calibration regions for the method of Feldman et al. (2023) and the standard DQR method. Additionally, to demonstrate that our coverage guarantees are robust to the choice of metric, we provide results using the Mean Absolute Error (MAE) in Appendix C.3.

Implementation details. We set the latent space dimension to $r = 2$. For the latent generative model, we choose VAE and set the KL regularization hyperparameter $\beta = 0.001$ (see Appendix C for the ablation study on the choice of r and β). For DQR in our method and in Feldman's method, the input size is $p + d$, and each gradient step uses $1,024$ directions with $\alpha = 0.1$. All data are L_2-normalized before training. Since setting the quantile level to α, DQR does not achieve the target coverage, we decrease the quantile level until the target coverage is met, resulting in the quantile levels of 0.01 and 0.001 to achieve coverage rates of $1-\alpha = 0.9$ and 0.98, respectively. For simplicity, we denote them as DQR-0.01 and DQR-0.001. More details can be found in Appendix B.2.

Calibration result. As shown in Tab. 1, all methods achieve the target coverage, and the prediction set region of our method is much smaller than that of DQR. We also note that the region is slightly larger than that of Feldman, which may be due to the expansion caused by the decoder (7).

Table 1: Coverage ratio and area on the *nonlinear* synthetic dataset with different nominal levels α.

α	Coverage					Area in $\mathcal{Y}$		
	Ours-$\mathcal{Z}$	Ours-$\mathcal{Y}$	Feldman-$\mathcal{Y}$	DQR-$\mathcal{Z}$	DQR-$\mathcal{Y}$	Ours	Feldman	DQR
0.02	0.9770	0.9760	0.9718	0.9818	0.9872	398.5	377.8	749.1
0.10	0.8953	0.8915	0.8940	0.8823	0.9145	232.7	234.5	287.4

Visualization. We visualize the region for two different values of X, where $R_{\mathcal{Z}}$ represents the region before calibration (4), and $R_{\mathcal{Y}}$ denotes the decoded region of $R_{\mathcal{Z}}$, *i.e.*, $R_{\mathcal{Y}} = \mathcal{D}ec(R_{\mathcal{Z}}, x)$. As shown in Fig. 2, the pre-calibrated region (in red) initially excludes the outcome (in green), but after calibration, the outcome is successfully included. Compared to DQR, the regions produced by our method and Feldman's approach are smaller. Additionally, the regions have different shapes across the two cases, demonstrating the adaptiveness of our calibration procedure.

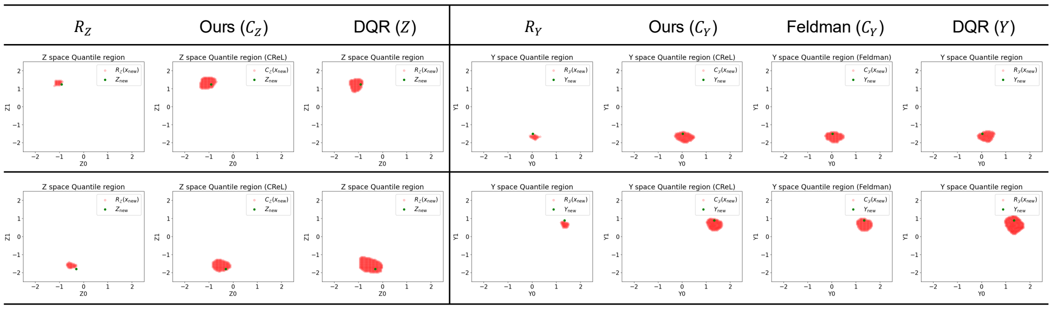


Figure 2: Visualization of regions produced by various methods ($\alpha = 0.1$). Each row represents a case, *i.e.*, a fixed x. **Left:** region in $\mathcal{Z}$; **Right:** region in $\mathcal{Y}$. Calibrated regions are marked in red, the test sample (Z_{new} or Y_{new}) is marked in green.

CReL scales efficiently for high-dimensional calibration. We evaluate the total calibration runtime across latent dimensions and find that our method scales efficiently with r, while the grid-based approach Feldman et al. (2023) incurs exponential growth and becomes infeasible in high dimensions (see Appendix E). This confirms CReL's practicality for modern large-scale systems.

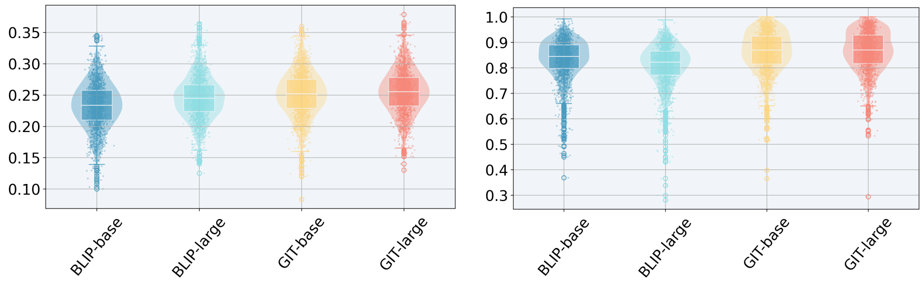
4.2 Experiments on Image-to-Text Task

Dataset and preprocessing. We use the MS-COCO 2014 validation set Lin et al. (2014) ($40{,}504$ image-caption pairs), and split it into 75% for VAE training, 15% for DQR, 5% for calibration, and 5% for testing. We evaluate four models: BLIP (base and large) Li et al. (2022) and GIT (base and large) Wang et al. (2022a), all at image size 224×224. We measure CLIP cosine similarity (CLIP-SIM) between the condition image and generated caption, *i.e.*, $\rho(\widehat{Y}, \text{GT}) = \text{Cos-Sim}(\widehat{Y}, X)$, where X denotes the image feature and $\widehat{Y}$ denotes the generated caption. We also measure the BERT cosine similarity (BERT-SIM) between the ground truth caption and generated caption, *i.e.*, $\rho(\widehat{Y}, \text{GT}) = \text{Cos-Sim}(\widehat{Y}, Y)$, which first extracts features from $\widehat{Y}$ and Y and then computes the cosine similarity between these feature representations.

Implementation details. Image features are extracted using CLIP ViT-L/14 Radford et al. (2021), and caption features are obtained using BERT-base Kenton & Toutanova (2019), where we use the [CLS] token that serves as a summary feature of the entire caption. Both feature types have dimension $p = d = 768$. The maximum text length is set to 50. For VAE, we set $r = 50$ and use $\beta = 0.001$ for KL regularization (see Appendix C for ablation study). For DQR, the input size is $p + d$, and each gradient step uses $1{,}024$ directions with $\alpha = 0.1$. All data are L_2-normalized before training. During optimization, we initialize the procedure with 50 starting points for BERT and CLIP. More details can be found in Appendix C.5.

Table 2: Quantitative results of the image-to-text generation task at $\alpha = 0.1$, with differences between CReL-ρ and ρ (Δ) highlighted in blue. Superscripts indicate the performance rank.

Model	CLIP-SIM		BERT-SIM	
	CLIP	CReL-CLIP	BERT	CReL-BERT
BLIP-base	0.2330^{4}	0.0070^{1} (-0.2260)	0.8349^{3}	0.6335^{3} (-0.2014)
BLIP-large	0.2453^{3}	-0.0074^{4} (-0.2527)	0.8106^{4}	0.5631^{4} (-0.2475)
GIT-base	0.2511^{2}	-0.0021^{2} (-0.2532)	0.8620^{2}	0.6474^{1} (-0.2146)
GIT-large	0.2550^{1}	-0.0043^{3} (-0.2593)	0.8649^{1}	0.6459^{2} (-0.2190)

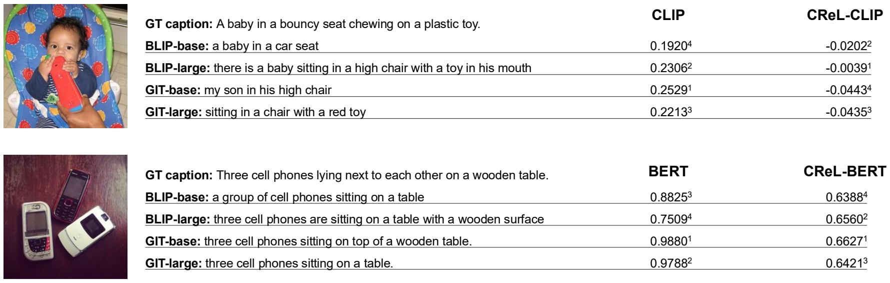


(a) CLIP (b) BERT

Figure 3: Distribution of ρ values for different models on the image-to-text generation task.

Quantitative comparison. We compare two large-scale caption generation models (BLIP and GIT) in both base and large versions, using two similarity metrics: CLIP-SIM and BERT-SIM. Results at $\alpha = 0.1$ is shown in Tab. 2. For CLIP-SIM, the rankings among BLIP-base, BLIP-large, and GIT-large vary after calibration. Notably, BLIP-base ranks last in the original score but first in our reliability score. This can be explained by Fig. 3(a), where the distribution of BLIP-base's scores is more concentrated around the central region compared to GIT-large, resulting in a higher worst-case score after calibration. For BERT-SIM, we observe that the gap between BLIP-base and BLIP-large is enlarged after calibration. Similarly, this can be explained by the more concentrated score distribution of BLIP-base relative to BLIP-large. In addition, our results also indicate that BLIP-base is the most reliable one in CLIP-SIM, while the GIT-base/large achieves the highest reliability in BERT-similarity. This may be because CLIP-SIM captures high-level semantic similarity between generated image and text, making it more suited to lightweight models like BLIP-base that avoid overfitting to irrelevant features. In contrast, BERT-SIM focuses on deeper and subtler contextual similarity. As a result, the GIT model, with its larger capacity and ability to process more intricate relationships, performs better in this task.

Qualitative results: CReL effectively identifies misalignments. Figure 4 presents examples illustrating that our calibrated metric better reflects generation quality compared to the original uncalibrated metric. Specifically, we take examples from CReL-CLIP (image–caption) and CReL-BERT (caption–caption). In the example for CReL-CLIP (*i.e.*, image-caption), the ground truth image shows "a baby in a seat playing a toy", but the GIT-base overlooks the information of "playing a toy". Despite this omission, CLIP assigns higher similarity scores to the GIT-base than the BLIP-large, which correctly identifies this semantics and is accurately ranked first by our reliability metric. In the example for CReL-BERT (*i.e.*, caption-caption), both BLIP-base ("a group of cell phones on a table") and GIT-large ("three phones sitting on a table") fail to specify the number of phones or mention that the table surface is wooden; however, they are ranked higher than BLIP-large, whose caption accurately captures both pieces of information. These results demonstrate that CReL effectively detects visual and semantic discrepancies that standard metrics miss, quantifying model reliability without sacrificing predictive performance. More examples can be found in Appendix G.1.



GT caption: A baby in a bouncy seat chewing on a plastic toy.	CLIP	CReL-CLIP
BLIP-base: a baby in a car seat	0.1920^4	-0.0202^2
BLIP-large: there is a baby sitting in a high chair with a toy in his mouth	0.2306^2	-0.0039^1
GIT-base: my son in his high chair	0.2529^1	-0.0443^4
GIT-large: sitting in a chair with a red toy	0.2213^3	-0.0435^3

GT caption: Three cell phones lying next to each other on a wooden table.	BERT	CReL-BERT
BLIP-base: a group of cell phones sitting on a table	0.8825^3	0.6388^4
BLIP-large: three cell phones are sitting on a table with a wooden surface	0.7509^4	0.6560^2
GIT-base: three cell phones sitting on top of a wooden table.	0.9880^1	0.6627^1
GIT-large: three cell phones sitting on a table.	0.9788^2	0.6421^3

Figure 4: Qualitative results of image-to-text models ($\alpha = 0.1$). Superscripts denote rank.

4.3 Experiments on Text-to-Image Task

Dataset and preprocessing. We use the MS-COCO 2014 validation set Lin et al. (2014) ($40,504$ image-caption pairs), and split it into 75% for VAE training, 15% for DQR, 5% for calibration, and 5% for testing. We evaluate three models: SD3-M Esser et al. (2024), SD3.5-L Esser et al. (2024), FLUX.1-dev Labs (2024), all at image size 512×512. We measure CLIP cosine similarity (CLIP-SIM) between the condition caption and generated image, *i.e.*, $\rho(\widehat{Y}, \mathrm{GT}) = \mathrm{Cos\text{-}Sim}(\widehat{Y}, X)$, where X denotes the text feature and $\widehat{Y}$ denotes the generated image. Please refer to Appendix G.2 for implementation details.

Table 3: Quantitative results of the text-to-image generation task at $\alpha = 0.1$, with differences between CReL-CLIP and CLIP (Δ) highlighted in blue. Superscripts indicate the performance rank.

Model	CLIP-SIM	
	CLIP	CReL-CLIP
SD3-M	0.2590^2	0.0134^1 (-0.2456)
SD3.5-L	0.2596^1	0.0116^2 (-0.2480)
FLUX.1-dev	0.2509^3	0.0056^3 (-0.2453)

Quantitative comparison. We compare three state-of-the-art text-to-image models: SD3-M, SD3.5-L, and FLUX.1-dev, using CLIP-SIM as the base metric. Results at $\alpha = 0.1$ are shown in Tab. 3. Notably, we observe a re-ranking phenomenon between SD3-M and SD3.5-L, highlighting the value of our reliability metric. Although SD3.5-L achieves the highest CLIP score ((0.2596)) and ranks first, it drops to second place in our CReL metric ((0.0116)). In contrast, SD3-M, with a slightly lower CLIP score ((0.2590)), achieves the highest reliability score ((0.0134)). This rank swap can be explained by the score distributions in Fig. 11. The standard CLIP score reflects average performance, where SD3.5-L benefits from high-scoring samples that boost the mean. However, CReL evaluates worst-case performance within the high-probability region $1 - \alpha$. As shown in the violin plots, SD3-M has a more uniform score distribution, indicating consistent performance, which ensures robustness at the $1 - \alpha$ confidence boundary. In contrast, SD3.5-L's higher mean is offset by its distribution characteristics, leading to a lower value at this critical threshold. This demonstrates that while SD3.5-L excels in average quality, SD3-M offers better consistency, making it more reliable for safety-critical applications requiring predictable performance. FLUX.1-dev consistently ranks third, indicating lower alignment with the MS-COCO prompts under this metric.

5 Conclusion and Future Works

We introduce a worst-case reliability metric based on conformal calibration to evaluate the conditional generative models, which provides a more interpretable assessment of model trustworthiness than traditional metrics that only consider a single output. A computational framework called Conformal ReLiability (CReL) was proposed to compute the reliability. CReL is highly flexible, accommodating any under most common or 'bespoke' similarity metrics. **In the future**, we will extend our framework to more complex scenarios, such as video generation, 3D reconstruction, or embodied robotics. Unlike one-to-one conditional generation, these tasks involve one-to-many, many-to-one, or many-to-many mappings, and will likely require novel joint latent representations and calibration techniques to ensure robust guarantees.

Reproducibility statement

The proof of proposition 3.2 and theorem 3.3 is provided in Section 3, while the proof of proposition 3.5 is provided in Appendix A.4. The generation details of synthetic data are provided in Appendix B.1, and the implementation details of the latent generative model are introduced in Appendix B.2. The code will be released once the paper is accepted [1].

References

Yaniv Benny, Tomer Galanti, Sagie Benaim, and Lior Wolf. Evaluation metrics for conditional image generation. *International Journal of Computer Vision*, 129(5):1712–1731, 2021.

Yuemin Bian and Xiang-Qun Xie. Generative chemistry: drug discovery with deep learning generative models. *Journal of Molecular Modeling*, 27(3):71, 2021.

Pavel Boček and Miroslav Šiman. Directional quantile regression in r. *Kybernetika*, 53(3):480–492, 2017.

Guillaume Carlier, Victor Chernozhukov, and Alfred Galichon. Vector quantile regression: an optimal transport approach. 2016.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Shai Feldman, Stephen Bates, and Yaniv Romano. Calibrated multiple-output quantile regression with representation learning. *Journal of Machine Learning Research*, 24(24):1–48, 2023.

Artur Gasparyan and Ruiqi Qiu. Diffusion models for novel view synthesis in autonomous driving. 2024.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1):59–99, 2016.

Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. qkaf008, 2025.

Wenchong He, Zhe Jiang, Tingsong Xiao, Zelin Xu, and Yukun Li. A survey on uncertainty quantification methods for deep learning. *arXiv preprint arXiv:2302.13425*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Mohammad Jalali, Azim Ospanov, Amin Gohari, and Farzan Farnia. An information-theoretic approach to diversity evaluation of prompt-based generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7634–7643, 2025.

Adel Javanmard, Simeng Shao, and Jacob Bien. Prediction sets for high-dimensional mixture of experts models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. qkae117, 2025.

[1] According to the ICLR policy, this section is excluded from the page limit.

Chancellor Johnstone and Eugene Ndiaye. Exact and approximate conformal inference in multiple dimensions. *arXiv preprint arXiv:2210.17405*, 2022.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, pp. 2. Minneapolis, Minnesota, 2019.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Linglong Kong and Ivan Mizera. Quantile tomography: using quantiles with multivariate data. *Statistica Sinica*, pp. 1589–1610, 2012.

Black Forest Labs. Flux.1, 2024. URL `https://blackforestlabs.ai/announcing-black-forest-labs/`.

Simon Lacoste-Julien. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345*, 2016.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36:69981–70011, 2023.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Puheng Li, Zhong Li, Huishuai Zhang, and Jiang Bian. On the generalization properties of diffusion models. *Advances in Neural Information Processing Systems*, 36:2097–2127, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Soundouss Messoudi, Sébastien Destercke, and Sylvain Rousseau. Ellipsoidal conformal inference for multi-target regression. In *Conformal and Probabilistic Prediction with Applications*, pp. 294–306. PMLR, 2022.

Azim Ospanov, Mohammad Jalali, and Farzan Farnia. Scendi score: Prompt-aware diversity evaluation via schur complement of clip embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16927–16937, 2025.

Davy Paindaveine and Miroslav Šiman. On directional multiple-output quantile regression. *Journal of Multivariate Analysis*, 102(2):193–212, 2011.

Vincent Plassier, Alexander Fishkov, Mohsen Guizani, Maxim Panov, and Eric Moulines. Probabilistic conformal prediction with approximate conditional validity. *arXiv preprint arXiv:2407.01794*, 2024.

Vincent Plassier, Alexander Fishkov, Victor Dheur, Mohsen Guizani, Souhaib Ben Taieb, Maxim Panov, and Eric Moulines. Rectifying conformity scores for better conditional coverage. *arXiv preprint arXiv:2502.16336*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pp. 1060–1069. Pmlr, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022a.

Zhendong Wang, Ruijiang Gao, Mingzhang Yin, Mingyuan Zhou, and David M Blei. Probabilistic conformal prediction using conditional random samples. *arXiv preprint arXiv:2206.06584*, 2022b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Chen Xu, Hanyang Jiang, and Yao Xie. Conformal prediction for multi-dimensional time series by ellipsoidal sets. In *Forty-first International Conference on Machine Learning*.

A Algorithm Details

In this section, we introduce our implementation of the VAE and the Stable Diffusion model.

A.1 Variational Auto Encoder

The variational lower bound of the model is written as follows Sohn et al. (2015):

$$\log p_\theta(\mathbf{Y}) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{Y})}\left[\log p_\theta(\mathbf{Y}|\mathbf{Z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{Z}|\mathbf{Y})\|p(\mathbf{Z})\right)}_{\mathcal{L}_{\mathrm{ELBO}}}, \tag{9}$$

where $\mathcal{L}_{\mathrm{ELBO}}$ denotes the ELBO objective to be maximized. Here:

- $\mathbf{Y}$: response variables (target)
- $\mathbf{Z}$: latent variables
- $q_\phi(\mathbf{Z}|\mathbf{Y})$: inference model (encoder)
- $p_\theta(\mathbf{Y}|\mathbf{Z})$: generative model (decoder)

The latent prior is fixed as $p(\mathbf{Z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$. The decoder outputs are denoted as $\widehat{\mathbf{Y}}$.

To incorporate task-specific metric $\rho(\mathbf{Y}, \widehat{\mathbf{Y}})$ (where higher values indicate better performance), we reformulate the likelihood as an energy-based model:

$$p_\theta(\mathbf{Y}|\mathbf{Z}) \propto \exp\left(\lambda \cdot \rho(\mathbf{Y}, \widehat{\mathbf{Y}})\right), \tag{10}$$

where $\lambda > 0$ is a temperature parameter. The intractable normalization constant:

$$C(\mathbf{Z}) = \int \exp\left(\lambda \cdot \rho(\mathbf{Y}, \widehat{\mathbf{Y}})\right) d\mathbf{Y} \tag{11}$$

is omitted during optimization following energy-based modeling conventions LeCun et al. (2006).

Substituting into the ELBO definition gives:

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\lambda \cdot \rho(\mathbf{Y}, \widehat{\mathbf{Y}}) - \log C(\mathbf{Z})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{Z}|\mathbf{Y})\|p(\mathbf{Z})\right). \tag{12}$$

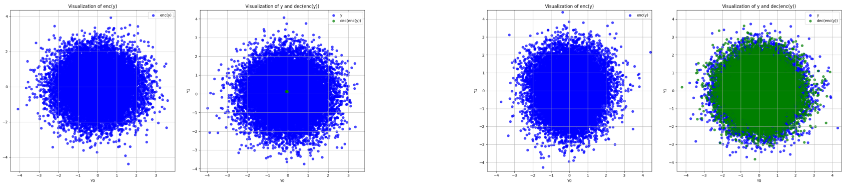
Approximating $\log C(\mathbf{X}, \mathbf{Z})$ as constant for gradient-based optimization yields:

$$\mathcal{L}_{\mathrm{ELBO}} \approx \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\lambda \cdot \rho(\mathbf{Y}, \widehat{\mathbf{Y}})\right] - D_{\mathrm{KL}}\left(q_\phi(\mathbf{Z}|\mathbf{Y})\|p(\mathbf{Z}|)\right). \tag{13}$$

The final loss function $\mathcal{L} = -\mathcal{L}_{\mathrm{ELBO}}$ is composed of two parts:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_\rho + \beta \cdot \mathcal{L}_{\mathrm{KL}}, \tag{14}$$

where $\mathcal{L}_\rho = -\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{Y})}\left[\rho(\mathbf{Y}, \widehat{\mathbf{Y}})\right]$ is the metric-driven reconstruction term with λ setting to 1, and $\mathcal{L}_{\mathrm{KL}} = D_{\mathrm{KL}}\left(q_\phi(\mathbf{Z}|\mathbf{Y})\|p(\mathbf{Z})\right)$ is the KL regularization term with β controlling its strength.



(a) CVAE

(b) VAE

Figure 5: The reconstruction region of CVAE (subfigure (a)) and the VAE (subfigure (b)) given a fixed x. In each subfigure, the left image visualizes the encoded region $\{\mathcal{E}(Y_i^x, x))\}_{i \leq N}$ in the Z's space; the right image visualizes the decoded region $\{\mathcal{D}ec(\mathcal{E}(Y_i^x, x))\}_{i \leq N}$ in the Y's space.

VAE vs. CVAE. In Feldman et al. (2023), the authors used the conditional variational auto-encoder Sohn et al. (2015), where the inference model $q_\phi(\mathbf{Z}|\mathbf{Y})$ and the generative model $p_\theta(\mathbf{Y}|\mathbf{Z})$ are respectively replaced with $q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{Y})$ and $p_\theta(\mathbf{Y}|\mathbf{X},\mathbf{Z})$. However, the CVAE can be very sensitive to the input condition $\mathbf{X}$, making it easy to collapse when conditioning on a fixed x. Therefore, we turn to use VAE, which can well reconstruct the output even when conditioning on a fixed x.

To illustrate, we train both a CVAE and a VAE on the dataset $\{X_i, Y_i\}$ in the nonlinear setting. At test time, we fix $X = x$ and generate samples $\{Y_1^x, \ldots, Y_N^x\}$ from the conditional model $Y \mid X = x$. We then visualize the reconstruction regions: for the CVAE, $\{\mathcal{D}ec_{\text{CVAE}}(\mathcal{E}_{\text{CVAE}}(Y_i^x, x))\}_{i\le N}$, and for the VAE, $\{\mathcal{D}ec_{\text{VAE}}(\mathcal{E}_{\text{VAE}}(Y_i^x, x))\}_{i\le N}$. As shown in Fig. 5, the VAE can reconstruct outputs faithfully when conditioned on x, whereas the CVAE reconstructions collapse into a much smaller region.

A.2 Stable diffusion model

We begin by training the VAE and use its encoder $\mathcal{E}(Y_i)$ to obtain the low-dimensional latent representation $Z_i^0 = \mathcal{E}(Y_i)$ for each i. We then apply the diffusion process in the latent space to evolve Z_i^0 into Z_i^t, and finally reconstruct $\widehat{Y}_i$ through the decoder $\mathcal{D}ec_\phi(Z_i^t)$.

First, we consider the forward noising process:

$$q(z_t \mid z_{t-1}) = \mathcal{N}\Big(z_t \,\Big|\, \sqrt{1-\beta_t}\, z_{t-1},\, \beta_t I\Big),$$

which admits the closed-form

$$q(z_t \mid z_0) = \mathcal{N}(z_t \mid \sqrt{\alpha_t}\, z_0,\, (1-\alpha_t) I), \quad \alpha_t = \prod_{s=1}^{t}(1-\beta_s).$$

Thus, at step T, we have $z_T \sim \mathcal{N}(0, I)$. The reverse process is parameterized by a neural network ϵ_θ (where we choose MLP on synthetic data), which predicts the noise component of z_t. Conditioning on x, the model learns to approximate

$$\epsilon_\theta(z_t, t, x) \approx \epsilon,\ \epsilon \sim \mathcal{N}(0, I).$$

The denoising distribution is then given by:

$$p_\theta(z_{t-1}|z_t, x) = \mathcal{N}(\mu_\theta(z_t, t, x), \Sigma_t I),$$

with the mean parameter

$$\mu_\theta(z_t, t, x) = \frac{1}{\sqrt{1-\beta_t}}\Big(z_t - \beta_t\, \epsilon_\theta(z_t, t, x)\Big).$$

The training is based on denoising score matching. Given (x, y) sampled from the dataset, we encode $z_0 = \mathcal{E}_\phi(y)$, draw $t \sim \text{Unif}(\{1, \ldots, T\})$, and generate z_t via the forward process. The objective is

$$\mathcal{L}(\theta) = \mathbb{E}_{z_0, x, t, \epsilon}\left[\|\epsilon - \epsilon_\theta(z_t, t, x)\|_2^2\right].$$

At inference, one begins with Gaussian noise $z_T \sim \mathcal{N}(0, I)$ and applies the learned reverse process to obtain a latent z_0. The decoder then maps z_0 back to image space:

$$\widehat{Y} = \mathcal{D}ec_\phi(z_0).$$

A.3 Computing Score

To compute the score E_i^+ in (6), we view this as a quadratic programming problem bounded by multiple inequalities:

$$\min_a \|a - Z_i\|^2 \text{ s.t.} \mathbf{u}_k^T a \ge f_\beta(X_i, \mathbf{u}_k), \|\mathbf{u}_k\| = 1,\ \forall k = 1, ..., K$$

Now our goal is to find a point $a^\star$ that minimizes the distance to the target point Z_i and satisfies all the half-space constraints. The score takes the value of the Euclidean distance from Z_i to $a^\star$.

A.4 Proofs of Section 3

Proof of Proposition 3.5. We now show that for any γ, $S^{\gamma}(x)$ is convex and compact. For any $z_1, z_2 \in S^{\gamma_{\text{cal}}}(x)$, we have $a_1 \in R_{\mathcal{Z}}(x), a_2 \in R_{\mathcal{Z}}(x)$ respectively such that $d(a_1, z_1) \le \gamma$ and $d(a_2, z_2) \le \gamma$. Besides, as $R_{\mathcal{Z}}(x)$ is convex, we have $\alpha a_1 + (1-\alpha)a_2 \in R_{\mathcal{Z}}(x)$. Then for any $0 < \alpha < 1$, we have:

$$
\begin{aligned}
\min_{a \in R_{\mathcal{Z}}(x)} d(a, \alpha z_1 + (1-\alpha)z_2) &= d(\alpha a + (1-\alpha)a, \alpha z_1 + (1-\alpha)z_2) \\
&\le d(\alpha a_1 + (1-\alpha)a_2, \alpha z_1 + (1-\alpha)z_2) \\
&\overset{(1)}{\le} \alpha \|a_1 - z_1\|_2 + (1-\alpha)\|a_2 - z_2\|_2 \le \gamma,
\end{aligned}
$$

where "(1)" is due to the jointly convexity of $d(\cdot,\cdot)$. The compactness follows from the compactness of $R_{\mathcal{Z}}(X)$ and that γ_{cal} is bounded. □

B Experimental Details

B.1 Synthetic Data Details

Linear data generation. The generation of the condition vector X and the response variable Y in the *linear* version of the synthetic data is defined as follows:

$$
\begin{aligned}
&X \sim \text{Uniform}(0.8, 3.2)^p, \\
&A \sim \mathcal{N}(0,1)^{p\times d}, \\
&\epsilon \sim \mathcal{N}(0,\sigma^2)^d, \\
&Y = XA + \epsilon,
\end{aligned}
\tag{15}
$$

where Uniform(a, b) is a uniform distribution on the interval (a, b), $X \in \mathbb{R}^p$ is the condition vector, $A \in \mathbb{R}^{p\times d}$ is the coefficient matrix, $\epsilon \in \mathbb{R}^d$ is Gaussian noise with variance σ^2, and $Y \in \mathbb{R}^d$ is the response variable.

Nonlinear data generation. The *nonlinear* version of the synthetic data is generated as follows:

$$
\begin{aligned}
&X \sim \text{Uniform}(0.8, 3.2)^p, \\
&A \sim \mathcal{N}(0,1)^{p\times d}, \\
&B \sim \mathcal{N}(0,1)^{p\times d}, \\
&\epsilon \sim \mathcal{N}(0,\sigma^2)^d, \\
&Y = XA + X^2B + \epsilon,
\end{aligned}
\tag{16}
$$

where $X \in \mathbb{R}^p$, $A \in \mathbb{R}^{p\times d}$, $B \in \mathbb{R}^{p\times d}$, $\epsilon \in \mathbb{R}^d$, and $Y \in \mathbb{R}^d$. The term X^2 denotes element-wise squaring of X.

B.2 Implementation Details

Network architectures.

- *VAE Encoder/Decoder Hidden Dimensions:*
  - Synthetic data: [64, 128, 256, 256, 128, 64]
  - Image-to-text task: [128, 256, 512, 512, 256, 128]
- *Stable Diffusion Denoiser:*
  - MLP hidden dimensions: [128, 256, 128]
  - Time embedding dimension: 128
- *DQR Network:*
  - Hidden dimensions: [8, 16, 8]

- Dropout (rate 0.1) and batch normalization are applied for the image-to-text dataset.

Training hyperparameters.

- *VAE:*
  - Learning rate: 1×10^{-3}
  - Activation: leaky ReLU (slope 0.2)
- *Stable Diffusion:*
  - Learning rate: 1×10^{-4}
  - Number of diffusion timesteps (training): 1000

DQR directions.

- Each gradient step uses 1024 distinct directions, sampled from a fixed set of 2048 directions generated before training.

Discretization.

- Number of grid points to decode region in $\mathcal{Z}$ space: 2×10^4
- Feldman grid in $\mathcal{Z}$ space: 2×10^4
- Feldman grid in $\mathcal{Y}$ space: 2×10^4
- Number of grid points for area calculation in $\mathcal{Y}$ space: 2×10^4

Hardware.

- Synthetic data simulations: NVIDIA RTX A6000 GPU (48GB VRAM)
- Image-to-text task: NVIDIA H100 GPU (80GB HBM3)

C ABLATION STUDY

C.1 EFFECT OF THE KL REGULARIZATION WEIGHT IN VAE

To investigate the effect of the KL regularization weight (β) in the VAE training loss, we conduct an ablation study on the *nonlinear* synthetic data, following the same setup as Section 4.1. As shown in Table 4, when $\beta = 0.001$, our method achieves both the target coverage ($\alpha = 0.1$) and a compact informative region. Therefore, we set $\beta = 0.001$ for all experiments.

Table 4: Ablation study on the effect of the KL regularization weight β in the VAE loss. The table reports the coverage ratios and the area of the region $C_{\mathcal{Y}}$ on the *nonlinear* synthetic dataset for different values of β. The target nominal level is $\alpha = 0.1$.

Metric	$\beta = 0.1$	$\beta = 0.01$	$\beta = 0.001$	$\beta = 0.0001$	$\beta = 0.00001$
coverage of $R_{\mathcal{Z}}$	0.5570	0.5525	0.4465	0.4203	0.4910
coverage of $C_{\mathcal{Z}}$	0.8883	0.8995	0.8953	0.8908	0.8945
coverage of $R_{\mathcal{Y}}$	0.9200	0.7280	0.5387	0.5060	0.5730
coverage of $C_{\mathcal{Y}}$	0.9945	0.9525	0.8915	0.8832	0.8895
area of $C_{\mathcal{Y}}$	1044.54	320.58	232.73	213.26	249.41

C.2 CHOICE OF LATENT GENERATIVE MODEL

To explore the effect of different latent generative models in our framework, we compare the Variational Autoencoder (VAE) and Stable Diffusion (SD) models on the *nonlinear* synthetic dataset, using the same experimental setup as Section 4.1. For the VAE, the KL regularization weight is set to $\beta = 0.001$. For the SD model, we use an MLP network as the denoiser; for implementation details regarding SD, please refer to Section B.2.

Table 5: Ablation study of the SD conditional denoiser: coverage and area for different inference steps T on the *nonlinear* synthetic dataset ($\alpha = 0.1$).

Metric	$T = 10$	$T = 20$	$T = 30$	$T = 40$	$T = 50$
coverage of $R_{\mathcal{Z}}$	0.4452	0.4725	0.4412	0.5503	0.5570
coverage of $C_{\mathcal{Z}}$	0.8968	0.9025	0.8952	0.9000	0.8998
coverage of $R_{\mathcal{Y}}$	0.5595	0.6218	0.6338	0.8055	0.8475
coverage of $C_{\mathcal{Y}}$	0.9065	0.9405	0.9675	0.9773	0.9883
area of $C_{\mathcal{Y}}$	239.05	285.93	367.64	405.89	525.12

Table 6: Ablation study of the SD unconditional denoiser: coverage and area for different inference steps T on the *nonlinear* synthetic dataset ($\alpha = 0.1$).

Metric	$T = 10$	$T = 20$	$T = 30$	$T = 40$	$T = 50$
coverage of $R_{\mathcal{Z}}$	0.4460	0.5320	0.5333	0.5507	0.5620
coverage of $C_{\mathcal{Z}}$	0.9008	0.8900	0.8988	0.8960	0.9058
coverage of $R_{\mathcal{Y}}$	0.5668	0.6893	0.7418	0.8085	0.8613
coverage of $C_{\mathcal{Y}}$	0.9203	0.9468	0.9638	0.9790	0.9923
area of $C_{\mathcal{Y}}$	271.26	309.56	370.03	415.60	542.00

Ablation study: SD denoiser architecture and inference steps. We first conduct an ablation study on the SD model to investigate the effect of (a) whether the denoiser is conditioned on input, and (b) the number of inference steps T (ranging from 10 to 50). We set the target nominal level to $\alpha = 0.1$. As shown in Tables 5 and 6, all values of T achieve the target coverage, with $T = 10$ providing the tightest coverage and, consequently, the least conservative region Therefore, we use the conditional denoiser with $T = 10$ in all subsequent SD experiments.

Comparison between SD and VAE as latent generative models. Finally, we compare the performance of SD (with conditional denoiser, $T = 10$) and VAE as the latent generative model in our CReL framework, evaluating both the coverage and the area on the *nonlinear* synthetic dataset for two nominal levels ($\alpha = 0.02, 0.10$), as summarized in Table 7. Both models achieve the target coverage, but the VAE consistently produces a more compact (informative) covered region in $\mathcal{Y}$. Based on these results, we use the VAE as the default latent generative model in all main experiments, due to its greater informativeness while maintaining desired coverage.

C.3 Robustness of Coverage Guarantees to Similarity Metric

CReL is designed to be metric-agnostic, successfully quantifying reliability with respect to a user's specific concern. To demonstrate that our coverage guarantees are independent of the metric choice, we conducted a new simulation experiment using the Mean Absolute Error (MAE) as the metric ρ. As shown in Tab. 8, the empirical coverage remains valid and closely tracks the nominal levels, just as it did for MSE in the main text.

C.4 Effect of Latent Space Dimensionality in VAE

To determine the appropriate dimensionality of the VAE latent space for the image-to-text task, we conduct an ablation study by varying the latent dimension r and evaluating its impact on the loss value (BERT-SIM, BLIP-large). The results, shown in Table 9, indicate that the loss remains rela-

Table 7: Comparison of VAE and SD as latent generative models in the CReL framework on the *nonlinear* synthetic dataset. Coverage and area metrics are reported for different α.

α	Coverage				Area in $\mathcal{Y}$	
	VAE-$\mathcal{Z}$	VAE-$\mathcal{Y}$	SD-$\mathcal{Z}$	SD-$\mathcal{Y}$	VAE	SD
0.02	0.9770	0.9760	0.9810	0.9843	398.51	432.99
0.10	0.8953	0.8915	0.8968	0.9065	232.73	239.05

Table 8: Coverage ratios and areas on the nonlinear synthetic dataset using MAE as the similarity metric.

α	Coverage in $\mathcal{Z}$	Coverage in $\mathcal{Y}$	Area in $\mathcal{Y}$
0.02	0.9743	0.9735	407.0
0.04	0.9563	0.9540	353.8
0.06	0.9403	0.9405	295.5
0.08	0.9100	0.9147	267.0
0.10	0.8905	0.8938	243.1

Table 9: Ablation study on the dimensionality of the VAE latent space in the image-to-text task. The table reports the loss value $\mathcal{L}$ for different choices of the latent dimension r.

r	10	20	50	100	200	300
$\mathcal{L}$	0.0442	0.0418	0.0418	0.0442	0.0422	0.0422

tively stable across a wide range of latent dimensions. Based on these observations, we empirically set $r = 50$ for all image-to-text experiments.

C.5 Effect of Initial Points

We analyze how the number of initial points (z_0) affects the accuracy of reliability estimates. As described in Section 3, the optimization of 8 combines the projected gradient descent and random search, where num$_{z_0}$ is a key hyperparameter. A larger num$_{z_0}$ improves estimation accuracy but increases computational cost.

To study its impact, we evaluate four generative models on the image-to-text task using CReL-CLIP and CReL-BERT at $\alpha = 0.1$, and three generative models on the text-to-image task using CReL-CLIP at $\alpha = 0.1$. As shown in Fig. 6 and Fig. 7, increasing num$_{z_0}$ achieves smaller reliability score. Besides, it can be shown that when the error stays stable num$_{z_0}$ achieves 50. To balance reliability and efficiency, we set num$_{z_0}$ = 50 for CReL-CLIP and CReL-BERT in all experiments.

D Computational Complexity Comparison

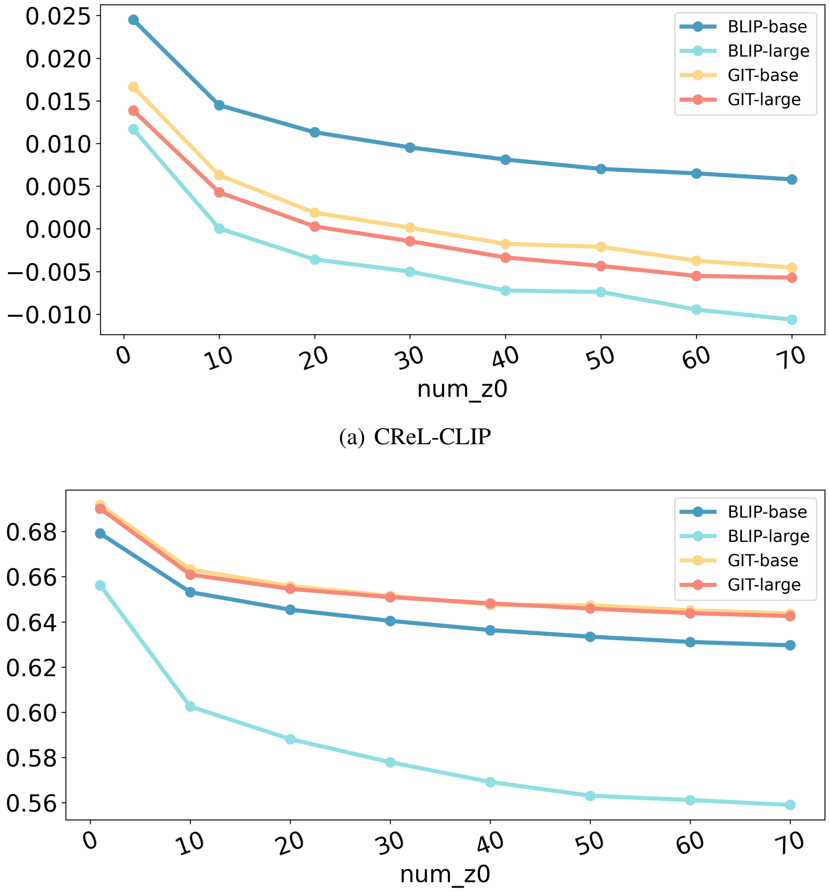
We compare the computational complexity of our calibration scheme with the grid-based discretization method of Feldman *et al.* (Feldman et al., 2023). While their approach incurs exponential costs in both the latent and original data spaces, our method operates directly in the lower-dimensional embedding and leverages DQR for initialization, yielding a significant reduction in computational cost. Here, we denote $n_{\text{cal}} := |\mathcal{I}_{\text{cal}}|$ as the sample size in the calibration set.

Feldman *et al.* (grid-based). They discretize both the r-dimensional latent space and the d-dimensional original space using a uniform grid of size m per axis:

- Latent space discretization: $O(m^r)$.
- Original space discretization: $O(m^d)$.
- Quantile initialization: computing the 90th percentile of all pairwise distances to obtain γ_{init} requires $O\big(n_{\text{cal}} \cdot m^{2d} \cdot d \cdot \log m\big)$.

Our calibration scheme. Our method avoids costly discretization and initialization by:

- Latent-space region: directly constructing the quantile region in the r-dimensional embedding space, bypassing any m-grid.
- DQR initialization: using the region obtained from DQR as the calibration starting point, eliminating the $O(n_{\text{cal}} \cdot m^{2d} \cdot d \cdot \log m)$ step.



(a) CReL-CLIP

(b) CReL-BERT

Figure 6: Effect of the number of initial points (z_0) on CReL-ρ for the image-to-text task.

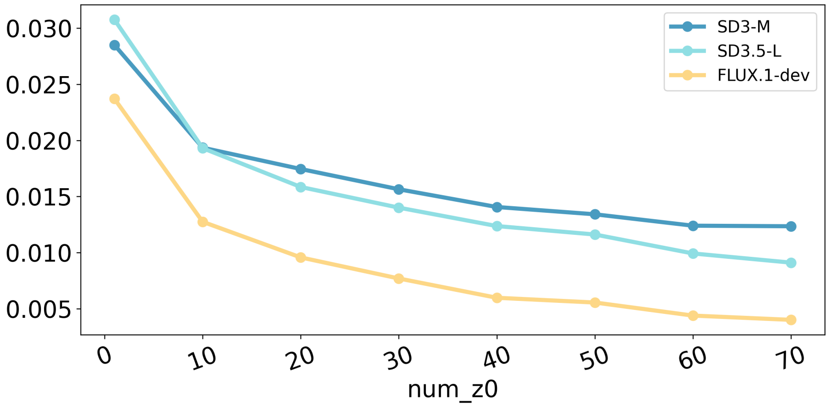


Figure 7: Effect of the number of initial points (z_0) on CReL-CLIP for the text-to-image task.

- Score computation: performing quadratic-programming updates in $O\big(n_{\mathrm{cal}} \cdot T \cdot q \cdot r\big)$, where T is the number of iterations and q is the number of calibration directions.

Overall Improvement. By operating in the lower-dimensional latent space ($r \ll d$) and leveraging our calibration scheme, we reduce the computational complexity of the calibration step from $O\big(n_{\mathrm{cal}} \cdot m^{2d} \cdot d \cdot \log m\big)$ to $O\big(n_{\mathrm{cal}} \cdot T \cdot q \cdot r\big)$.

For empirical runtime comparisons, see Appendix E.

E Calibration Scheme Runtime Comparison

E.1 Setup

Simulations on synthetic data. We evaluate the runtime of our calibration scheme against the grid-based discretization method of Feldman *et al.* (Feldman et al., 2023) using a *linear* synthetic dataset (generation details in Appendix B.1). We generate n = 50,000 samples p = 50 and d = 20, and split them as follows:

VAE training set:	60%	(30,000 samples)
DQR training set:	24%	(12,000 samples)
Calibration set:	8%	(4,000 samples)
Test set:	8%	(4,000 samples)

The calibration set is used to measure the runtime of both schemes.

Implementation details. For the VAE, we vary the latent dimension r from 2 to 12 and fix the loss weight β = 0.01. In the grid-based scheme Feldman et al. (2023), we use a uniform grid of size m = 5 per latent axis and fix the total number of grid points in the original space to 300,000 to control memory usage. For DQR, the input size is $p + d$, and each gradient step uses 1024 directions with α = 0.1. All data undergo L_2 normalization before training. Further details are provided in Appendix B.2.

E.2 Results

We report the total calibration runtime (in seconds) for both our calibration scheme and the grid-based discretization method of Feldman *et al.* across different latent dimensions. Several key observations emerge:

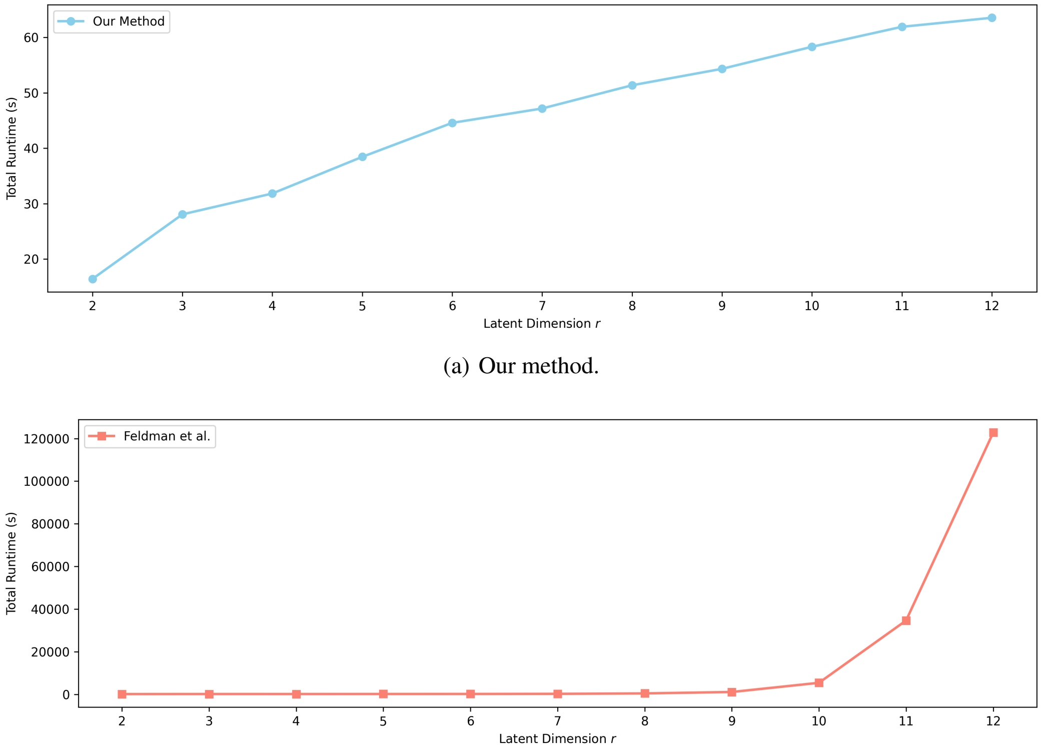
Our method exhibits near-linear growth in runtime with respect to r. Starting at approximately 16.4 s for r = 2, the total runtime increases modestly to about 63.5 s at r = 12, corresponding to an average incremental cost of under 5 s per additional latent dimension. This behavior is consistent with our theoretical complexity (Appendix D), in which r enters only linearly.

In contrast, the grid-based approach exhibits exponential growth: its calibration time increases from 116.6 s at r = 2 to 433.3 s at r = 8 and finally to 122,795.0 s at r = 12. Correspondingly, our method is over 8× faster at r = 8 (51.37 s vs. 433.3 s) and nearly $1,930$× faster at r = 12 (63.54 s vs. 122,795.0 s), rendering the grid-based scheme infeasible for moderate-to-high latent dimensions.

These empirical results corroborate our theoretical complexity reduction and demonstrate that by operating directly in the lower-dimensional embedding space, our calibration scheme remains computationally feasible even as r grows large. This efficiency gain is critical for scaling conformal calibration to high-dimensional prediction tasks.

F The Use of Large Language Models (LLMs)

We declare that in this work, LLMs were used solely for grammatical correction in writing.



(a) Our method.

(b) Feldman *et al.* Feldman et al. (2023).

Figure 8: Comparison of total calibration runtime (in seconds) across different latent dimensions r. Our method exhibits favorable scalability, whereas the grid-based approach of Feldman et al. (2023) incurs significantly higher computational cost as r increases.

G ADDITIONAL RESULTS

G.1 QUALITATIVE RESULTS OF IMAGE-TO-TEXT TASK

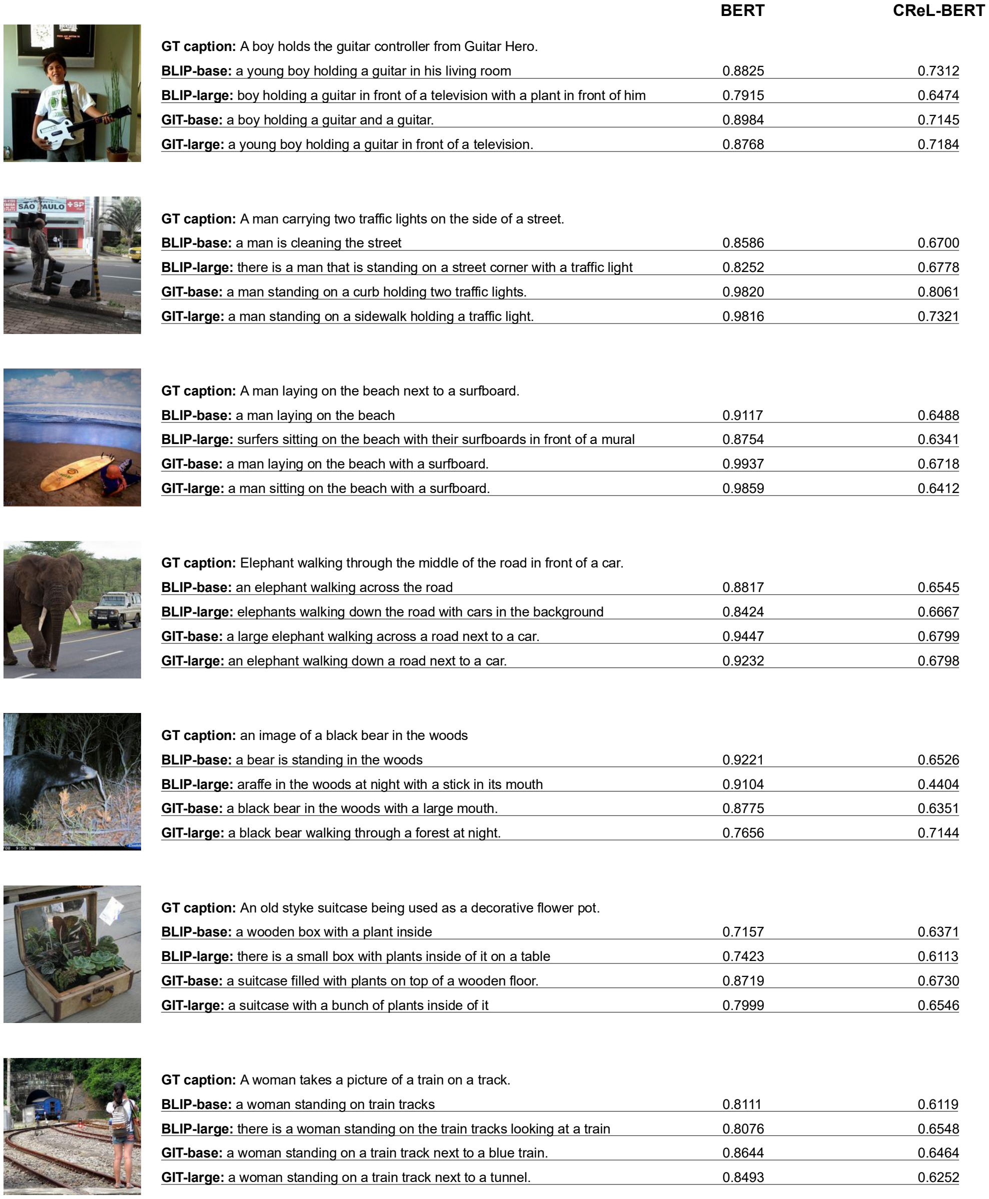
	CLIP	CReL-CLIP
GT caption: A cat sitting on top of a pile of books in a city.		
BLIP-base: a cat sitting on a pile of books	0.2698	0.0097
BLIP-large: araffe cat sitting on top of a pile of books on a sidewalk	0.2919	-0.0026
GIT-base: a cat sitting on books in a cafe	0.2932	-0.0105
GIT-large: a cat sitting on top of a book on a table.	0.2737	-0.0313
GT caption: a woman using a white laptop on the bed		
BLIP-base: a boy laying on a bed	0.1917	0.0134
BLIP-large: arafed woman laying on bed using laptop computer with pink sheets	0.2082	0.0385
GIT-base: a woman laying on a bed using a laptop.	0.2246	0.0358
GIT-large: a young man laying on a bed looking at a laptop.	0.2769	0.0114
GT caption: The man is laying out in the sand at the beach		
BLIP-base: a man laying on the beach	0.2317	0.0224
BLIP-large: there is a man laying on a beach with a surfboard	0.2468	-0.0118
GIT-base: a man laying on the beach in the sand	0.2412	0.0010
GIT-large: a man laying on the beach with his arms stretched out.	0.2574	0.0315
GT caption: A horse that is in the middle of a patch of flowers.		
BLIP-base: a flower garden with many different flowers	0.2517	0.0226
BLIP-large: arafed flower garden with a dog in the middle of it	0.2818	0.0057
GIT-base: a dog in a flower bed	0.2799	0.0104
GIT-large: a flower garden with a horse in the middle.	0.2922	0.0241
GT caption: a double decked bus parked by a stadium		
BLIP-base: a red bus parked in front of a building	0.2546	0.0017
BLIP-large: arafed bus parked in front of a large tent on a hill	0.2568	-0.0192
GIT-base: a red bus in the parking lot	0.2250	-0.0227
GIT-large: a red double decker bus parked in a parking lot.	0.2218	0.0059
GT caption: A little girl sitting at the end of a bed looking at a teddy bear.		
BLIP-base: a little girl sitting on a bed with a teddy bear	0.2958	0.0137
BLIP-large: there is a little girl sitting on a bed with a teddy bear	0.2742	0.0115
GIT-base: a little boy sitting on a bed with a stuffed animal.	0.3096	-0.0280
GIT-large: a child sitting on a bed next to a teddy bear.	0.3176	0.0037
GT caption: A man in a suit and tie standing in the desert.		
BLIP-base: a man in a suit and tie standing in a field	0.2758	-0.0032
BLIP-large: arafed man in suit and tie standing in front of a beach	0.2864	0.0313
GIT-base: a man standing on a beach with a suit and tie.	0.2743	0.0386
GIT-large: a man in a suit and tie standing on the beach.	0.2700	0.0386

Figure 9: Qualitative results of image-to-text models ($\alpha = 0.1$).

G.2 Experiments on Text-to-Image Task

Dataset and preprocessing. We use the MS-COCO 2014 validation set Lin et al. (2014) (40,504 image-caption pairs), and split it into 75% for VAE training, 15% for DQR, 5% for calibration, and 5% for testing. We evaluate three models: SD3-M Esser et al. (2024), SD3.5-L Esser et al. (2024), FLUX.1-dev Labs (2024), all at image size 512 × 512. We measure CLIP cosine similarity (CLIP-SIM) between the condition caption and generated image, *i.e.*, $\rho(\widehat{Y}, \mathrm{GT}) = \mathrm{Cos\text{-}Sim}(\widehat{Y}, X)$, where X denotes the text feature and $\widehat{Y}$ denotes the generated image.

Implementation details. Image and text features are extracted using CLIP ViT-L/14 Radford et al. (2021). Both feature types have dimension $p = d = 768$. All generation models use 50 inference steps and a guidance scale of 7.0. For VAE, we set $r = 50$ and use $\beta = 0.001$ for KL regularization. For DQR, the input size is $p + d$, and each gradient step uses 1,024 directions with $\alpha = 0.1$. All data



	BERT	CReL-BERT
GT caption: A boy holds the guitar controller from Guitar Hero.		
BLIP-base: a young boy holding a guitar in his living room	0.8825	0.7312
BLIP-large: boy holding a guitar in front of a television with a plant in front of him	0.7915	0.6474
GIT-base: a boy holding a guitar and a guitar.	0.8984	0.7145
GIT-large: a young boy holding a guitar in front of a television.	0.8768	0.7184
GT caption: A man carrying two traffic lights on the side of a street.		
BLIP-base: a man is cleaning the street	0.8586	0.6700
BLIP-large: there is a man that is standing on a street corner with a traffic light	0.8252	0.6778
GIT-base: a man standing on a curb holding two traffic lights.	0.9820	0.8061
GIT-large: a man standing on a sidewalk holding a traffic light.	0.9816	0.7321
GT caption: A man laying on the beach next to a surfboard.		
BLIP-base: a man laying on the beach	0.9117	0.6488
BLIP-large: surfers sitting on the beach with their surfboards in front of a mural	0.8754	0.6341
GIT-base: a man laying on the beach with a surfboard.	0.9937	0.6718
GIT-large: a man sitting on the beach with a surfboard.	0.9859	0.6412
GT caption: Elephant walking through the middle of the road in front of a car.		
BLIP-base: an elephant walking across the road	0.8817	0.6545
BLIP-large: elephants walking down the road with cars in the background	0.8424	0.6667
GIT-base: a large elephant walking across a road next to a car.	0.9447	0.6799
GIT-large: an elephant walking down a road next to a car.	0.9232	0.6798
GT caption: an image of a black bear in the woods		
BLIP-base: a bear is standing in the woods	0.9221	0.6526
BLIP-large: araffe in the woods at night with a stick in its mouth	0.9104	0.4404
GIT-base: a black bear in the woods with a large mouth.	0.8775	0.6351
GIT-large: a black bear walking through a forest at night.	0.7656	0.7144
GT caption: An old styke suitcase being used as a decorative flower pot.		
BLIP-base: a wooden box with a plant inside	0.7157	0.6371
BLIP-large: there is a small box with plants inside of it on a table	0.7423	0.6113
GIT-base: a suitcase filled with plants on top of a wooden floor.	0.8719	0.6730
GIT-large: a suitcase with a bunch of plants inside of it	0.7999	0.6546
GT caption: A woman takes a picture of a train on a track.		
BLIP-base: a woman standing on train tracks	0.8111	0.6119
BLIP-large: there is a woman standing on the train tracks looking at a train	0.8076	0.6548
GIT-base: a woman standing on a train track next to a blue train.	0.8644	0.6464
GIT-large: a woman standing on a train track next to a tunnel.	0.8493	0.6252

Figure 10: Qualitative results of image-to-text models ($\alpha = 0.1$).

are L_2-normalized before training. During optimization, we initialize the procedure with 50 starting points for CLIP. More details can be found in Appendix C.5.

Quantitative comparison. We compare three state-of-the-art text-to-image models: SD3-M, SD3.5-L, and FLUX.1-dev, utilizing CLIP-SIM as the base metric. Results at $\alpha = 0.1$ are presented in Tab. 3. Crucially, we observe a re-ranking phenomenon between the SD3-M and SD3.5-L models, which highlights the unique value of our reliability metric. While the larger model, SD3.5-L, achieves the highest standard CLIP score (0.2596) ranking first, it falls to second place in our CReL metric (0.0116). Conversely, SD3-M, despite having a slightly lower average performance (0.2590), achieves the highest reliability score (0.0134). This rank swap can be explained by analyzing the score distributions visualized in Fig. 11. Standard CLIP score reflects the *average* performance, where SD3.5-L benefits from generating high-scoring samples that push up the mean. However, CReL assesses the *worst-case* performance within the high-probability region $(1 - \alpha)$. As shown in

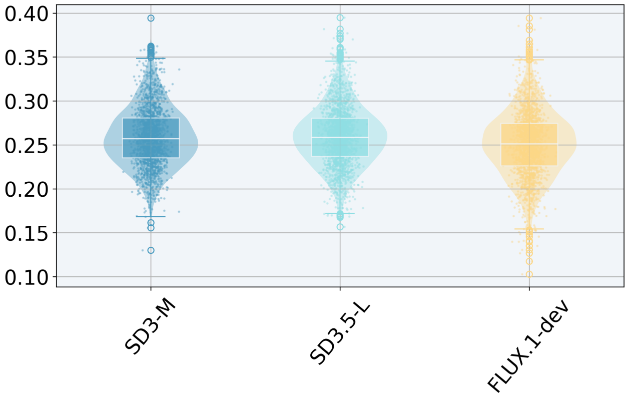
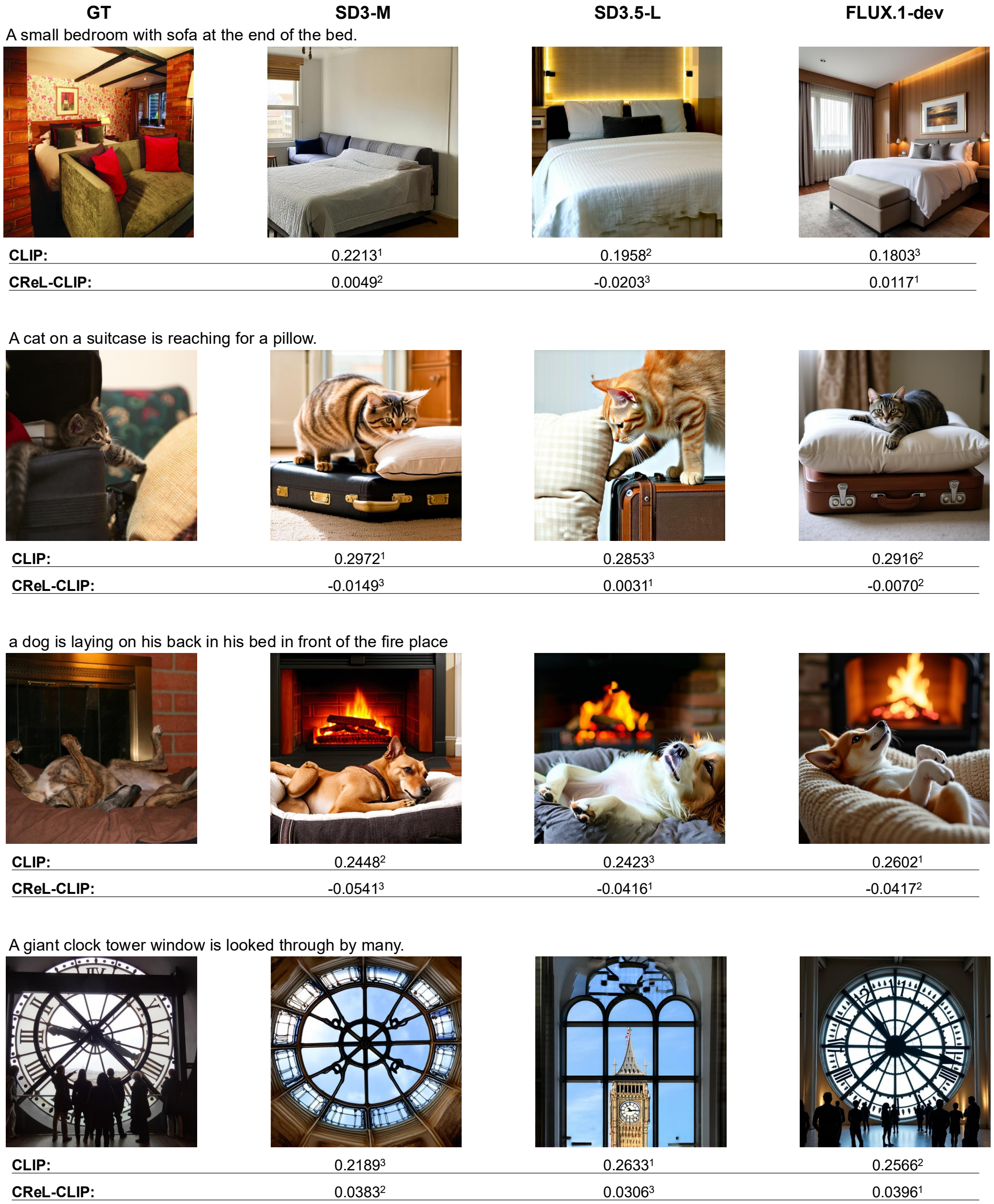


Figure 11: Distribution of CLIP values for different models on the text-to-image generation task.

the violin plots (Fig. 11), SD3-M exhibits a more uniform score distribution. This uniformity indicates a more consistent performance profile, ensuring that the metric value at the $1 - \alpha$ confidence boundary remains robust. In contrast, although SD3.5-L has a higher mean, its distribution characteristics result in a lower value at this critical reliability threshold. This demonstrates that while SD3.5-L optimizes for average quality, SD3-M offers better consistency, making it a more reliable choice for safety-critical applications where predictable performance is paramount. FLUX.1-dev consistently ranks third, indicating a general lower alignment with the MS-COCO prompts under this specific metric.

Qualitative results: CReL effectively identifies misalignments. Figure 12 provides qualitative examples illustrating that our calibrated metric better reflects generation reliability compared to the uncalibrated single-sample metric. Standard metrics like CLIP often assign high scores to images that capture the general theme but miss crucial semantic details specified in the prompt. For instance, given the prompt “A small bedroom with sofa at the end of the bed”, only the FLUX.1-dev model correctly generates the specific spatial arrangement, yet standard CLIP ranks it lower than the failing models; in contrast, CReL correctly identifies it as the most reliable model with the highest score. Similarly, for “A cat on a suitcase is reaching for a pillow”, SD3.5-L successfully depicts the “reaching” action while others fail, but CLIP ranks it last; CReL provides a more nuanced evaluation that better aligns with this semantic fulfillment. Conversely, CReL effectively penalizes hallucinations or failures that CLIP misses: in the case of “a dog is laying on his back...”, SD3-M fails to generate the correct pose but receives a high CLIP score, whereas CReL’s reliability assessment reflects the risk of this semantic failure. Finally, for “A giant clock tower window is looked through by many”, where SD3.5-L (CLIP rank 1) generates inconsistent content and SD3-M misses key semantics, FLUX.1-dev perfectly captures the scene and is accurately ranked first by CReL. These results demonstrate that CReL effectively detects fine-grained semantic discrepancies that standard metrics miss, quantifying model reliability without solely relying on average performance.



A small bedroom with sofa at the end of the bed.

	SD3-M	SD3.5-L	FLUX.1-dev
CLIP:	0.2213^{1}	0.1958^{2}	0.1803^{3}
CReL-CLIP:	0.0049^{2}	-0.0203^{3}	0.0117^{1}

A cat on a suitcase is reaching for a pillow.

	SD3-M	SD3.5-L	FLUX.1-dev
CLIP:	0.2972^{1}	0.2853^{3}	0.2916^{2}
CReL-CLIP:	-0.0149^{3}	0.0031^{1}	-0.0070^{2}

a dog is laying on his back in his bed in front of the fire place

	SD3-M	SD3.5-L	FLUX.1-dev
CLIP:	0.2448^{2}	0.2423^{3}	0.2602^{1}
CReL-CLIP:	-0.0541^{3}	-0.0416^{1}	-0.0417^{2}

A giant clock tower window is looked through by many.

	SD3-M	SD3.5-L	FLUX.1-dev
CLIP:	0.2189^{3}	0.2633^{1}	0.2566^{2}
CReL-CLIP:	0.0383^{2}	0.0306^{3}	0.0396^{1}

Figure 12: Qualitative results of text-to-image models ($\alpha = 0.1$). Superscripts denote rank.