Generative Active Learning via Bayesian Acquisition for Improving the Efficiency of Synthetic Data

Jeongjun Lee
KAIST
lee.jeongjun@kaist.ac.kr

Gyeonghoon Ko
KAIST
kog@kaist.ac.kr

Juho Lee
KAIST
juholee@kaist.ac.kr

Abstract

Unlike real-world datasets, utilizing synthetic data from generative models as train dataset necessitates a rigorous consideration of the training data's informativeness. In this paper, we propose a novel framework designed to steer diffusion models based on Bayesian epistemic uncertainty to train the target model. To ensure computational tractability, we estimate the Bayesian Active Learning by Disagreement (BALD) score by applying the Laplace approximation specifically to the last-layer parameters of the downstream model. Furthermore, we incorporate Feynman-Kac Steering (FKS) for diffusion to enhance the structural diversity of the synthetic samples and mitigate the emergence of undesirable artifacts. Extensive experiments on Imagenette and ImageNet-100 for the classification task and MS-COCO for the object detection task demonstrate that our method significantly outperforms existing baselines in downstream accuracy and OOD generalization. Furthermore, our analysis of pairwise distances between inter-class and intra-class confirms that our method effectively expands the downstream model's knowledge manifold by generating diverse and informative training signals.

1 Introduction

High-quality data have long been a cornerstone of deep learning success, yet they have also persistently constrained its scalability. For many real-world tasks, obtaining well-curated and accurately labeled datasets at scale is costly, labor-intensive, and often infeasible, and the supply of such data is rapidly diminishing. This tension has become increasingly acute in recent years with the acceleration of model scaling, driving growing interest in automated labeling methods and synthetic data generation as alternative sources of supervision for training large-scale models [33, 7, 56].

Even prior to the deep learning revolution and the recent scaling trend, there existed a rich line of research aimed at improving learning efficiency under limited labeled data. Among these efforts, active learning (AL) focuses on adaptively selecting unlabeled samples to query human annotators so as to maximize learning efficiency per label [43, 16, 29]. In its classical formulation, AL assumes access to a pool of real but unlabeled data, from which an algorithm ranks candidates according to a utility or informativeness criterion. A key underlying assumption of this paradigm is that unlabeled data are readily available. However, as modern models become increasingly capable and correspondingly more data-hungry, even large reservoirs of unlabeled data are becoming scarce. As a result, traditional pool-based active learning is gradually losing relevance in contemporary large-scale learning regimes.

In recent years, image generative models have advanced to the point where they can synthesize highly realistic images that are sufficiently faithful to be used as training data for next-generation models [1, 57, 13, 52]. This capability enables a natural extension of pool-based active learning to Generative Active Learning (GAL), in which the goal is no longer to rank existing unlabeled samples by utility, but instead to generate new samples that directly maximize a prescribed utility objective.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Diffusion models [22, 26, 41], which have become the dominant paradigm for high-fidelity image generation, are particularly well suited for this setting, as they allow for explicit guidance mechanisms during the sampling process [6, 21]. Leveraging this property, Askari-Hemmat et al. [2] proposed a GAL framework based on entropy-driven guidance, where the score function of a diffusion model is steered to increase the entropy of a target classifier. Although still preliminary as a principled AL method, this approach demonstrated promising empirical performance on relatively large-scale image classification tasks.

In this paper, building on guided diffusion models within the GAL framework, we introduce two key advancements that are critical for generating both informative and diverse training images. First, rather than relying on entropy-based guidance, which can be heuristic and conflate epistemic and aleatoric uncertainty, we adopt Bayesian Active Learning by Disagreement (BALD) [23] as the guiding objective, well-known proxy that empirically targets epistemic uncertainty in the target classifier. Classical BALD estimation typically requires constructing a committee of models, for example, via deep ensembles [31] or approximate posterior samples from Bayesian neural networks (BNNs) [15]. However, maintaining such committees is computationally expensive, both in terms of model construction and guidance evaluation during generation. To address this limitation, we adapt a last-layer Bayesian approach [30], in which the feature extractor of the target model is kept deterministic while uncertainty is modeled only in the final linear layer. This formulation admits a closed-form expression for the BALD objective, substantially reducing computational overhead while retaining a highly informative epistemic uncertainty signal suitable for guidance.

Our second advancement concerns the mechanism by which guidance is applied during diffusion sampling. Instead of directly modifying the denoising dynamics, we employ Feynman–Kac Steering (FKS) [45], which constructs a population of decoding trajectories and resamples them according to the guiding criterion. This population-based steering mechanism enables the generation of a diverse set of samples that better align with the BALD. Empirically, we show that FKS is essential for maintaining diversity among informative samples. We demonstrate the effectiveness of our GAL framework on image classification and object detection. Empirical results show that our approach improves data efficiency compared to existing methods. In addition, both quantitative and qualitative analyses indicate that the generated samples consistently exhibit higher epistemic uncertainty, confirming that our method successfully targets informative regions of the data space.

2 Preliminary

2.1 Score-based Generative Models (SGM)

Let p_0 be the unknown data distribution. A Score-based Generative Model (SGM) defines stochastic processes $\{x_t\}_{t\in[0,T]}$, whose marginal distribution is denoted by p_t, that connects p_0 to an easy-to-sample distribution p_T via the forward and reverse stochastic differential equations (SDEs):

$$\text{(forward)} \qquad dx_t = f(x_t,t)\,dt + g(t)\,dw_t, \qquad x_0 \sim p_0, \tag{1}$$

$$\text{(reverse)} \qquad dx_t = \left[f(x_t,t) - g^2(t)\nabla_x \log p_t(x_t)\right]dt + g(t)\,d\bar{w}_t, \qquad x_T \sim p_T. \tag{2}$$

where w_t and $\bar{w}_t$ denote standard Wiener processes in forward and reverse time, and $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ and $g : [0,T] \to \mathbb{R}$ are the drift and diffusion functions, respectively. Common choices of f and g are the VP SDE [22] and VE SDE [48], for which the transition kernel $p_{t|0}(x_t|x_0)$ is a closed-form Gaussian with mean and variance determined by f and g. Since the score function $\nabla_x \log p_t(x_t)$ is intractable, it is approximated by a neural network (called a score network) $s_\phi \approx \nabla_x \log p_t$, trained by the denoising score matching loss [54]

$$\mathbb{E}_{t,x_0,x_t}\left[\lambda(t)\left\|s_\phi(x_t,t) - \nabla_x \log p_{t|0}(x_t|x_0)\right\|^2\right], \tag{3}$$

where $t \sim \mathrm{Unif}(0,T)$, $x_0 \sim p_0(x_0)$, $x_t \sim p_t(x_t|x_0)$ and $\lambda(t)$ is a predefined weight schedule. Once the score network s_ϕ is learned, one can sample from p_0 by integrating the reverse SDE Eq. (2) by replacing $\nabla_x \log p_t$ with s_ϕ. For more details, see e.g., Ho et al. [22], Song et al. [48].

Conditioning and steering. In practice, one often seeks to steer the sampling trajectory in Eq. (2) so that the generated samples align with user-specified rewards or objectives. The gradient-based guidance [6, 60] adds gradients of a reward function to the reverse-time drift. In contrast, Sequential

Monte Carlo (SMC)-based steering [45, 59] provides a robust population-based alternative that does not require reward gradients. Specifically, the reverse process is represented by parallel trajectories called particles, whose intermediate rewards are evaluated during the diffusion steps. These rewards serve as importance weights for resampling, removing low-utility trajectories and replicating those that better align with the objective, even when the objective is non-differentiable.

2.2 Uncertainty Quantification

Consider a supervised learning task with dataset $\mathcal{D} := \{(x^{(i)}, y^{(i)})\}_{i=1}^n$, where $(x^{(i)}, y^{(i)}) \sim p_{\text{data}}$. Bayesian neural networks (BNNs) treat the weights $\theta \in \mathbb{R}^D$ as random variables to quantify uncertainty. For the negative log-likelihood $\ell(x, y; \theta)$ and Gaussian prior $\theta \sim \mathcal{N}(0, \tau^{-1} I_D)$, the MAP estimate is obtained by solving

$$\theta_\star = \arg\min_\theta \sum_{i=1}^n \ell(x^{(i)}, y^{(i)}; \theta) + \frac{\tau}{2}\|\theta\|^2. \tag{4}$$

The *predictive uncertainty*, i.e., uncertainty in a model's prediction, is quantified by the posterior predictive distribution

$$p(y|x, \mathcal{D}) = \int p(y|x, \theta) p(\theta|\mathcal{D}) d\theta, \tag{5}$$

e.g., through the predictive entropy $\mathcal{H}[y|x, \mathcal{D}] := \mathcal{H}(p(\cdot|x, \mathcal{D}))$. Since the exact posterior is typically intractable, the Laplace approximation [38, 40] uses a Gaussian approximation $p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta_\star, H_\star^{-1})$, where $H_\star := -\nabla_\theta^2 \log p(\theta|\mathcal{D})\big|_{\theta=\theta_\star}$ is the Hessian of the negative log-posterior at $\theta_\star$.

The predictive entropy $\mathcal{H}[y|x, \mathcal{D}]$ captures the total uncertainty in the prediction, which consists of both *aleatoric* and *epistemic uncertainty*. Aleatoric uncertainty refers to the inherent ambiguity in the data, and can be measured by the expected conditional entropy $\mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[y|x, \theta]]$. Epistemic uncertainty, on the other hand, reflects uncertainty about the model parameters and can be reduced with additional informative data. From an active learning perspective, one therefore seeks samples with high epistemic uncertainty. A standard acquisition function is Bayesian Active Learning by Disagreement (BALD) [23], which measures the mutual information between the prediction and the model parameters:

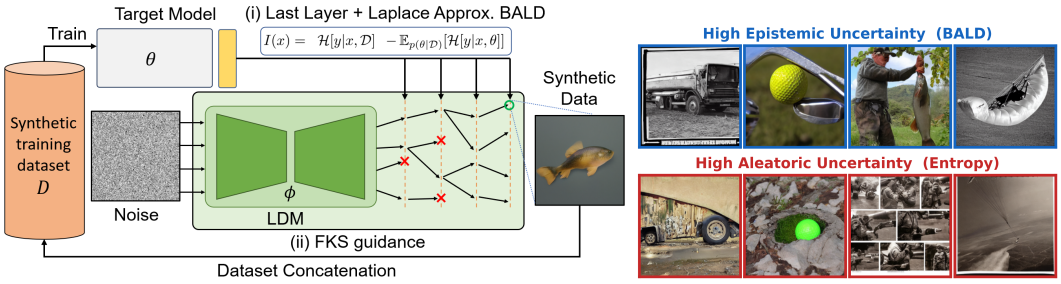
$$\mathcal{I}[y|x, \mathcal{D}] = \underbrace{\mathcal{H}[y|x, \mathcal{D}]}_{\text{Predictive entropy}} - \underbrace{\mathbb{E}_{p(\theta|\mathcal{D})}[\mathcal{H}[y|x, \theta]]}_{\text{Expected entropy}}. \tag{6}$$

BALD serves as an effective proxy to prioritize epistemic uncertainty over aleatoric noise by prioritizing samples on which posterior samples of θ disagree most about the prediction.

3 Related Work

Generative Active Learning. The initial attempt to actively generate instances for training a target model utilizes a generative adversarial network [GAN; 17]. Using GAN, Generative Adversarial Active Learning [GAAL; 62] provides a framework for collecting synthesized images whose output decision points near the decision boundary of the target learner, corresponding to images that the learner finds uncertain. Alternatively, Bayesian Generative Active Deep Learning [BGADL; 53] leverages a generative model to augment the labeled dataset by training variational autoencoder generative adversarial networks that efficiently generate both input and label data. Deliberate Practice [DP; 2] proposed a method to generate data that is challenging from the perspective of the target model using gradient-based guidance to the diffusion model to maximize the output entropy of a classification model, thereby encouraging the synthesis of difficult examples.

Guidance for Diffusion Models. Parallel to the rapid advancements in diffusion models, various methodologies have been explored to steer the inference process toward synthesizing desired samples. Classifier guidance [6] first demonstrated this potential by training an auxiliary classifier on intermediate noisy states x_t; the gradients of this classifier were then used to modulate the score function, effectively guiding the generation toward a target class. Subsequently, classifier-free guidance [21] achieved an analogous effect without the need for an external classifier by extrapolating the outputs of



(a) Pipeline of GALBA using (i) Last-layer Laplace approximation, (ii) Feynman-Kac steering of diffusion models.

(b) Visual comparison of generated data for high aleatoric and epistemic uncertainty.

a conditional diffusion model. Discriminator guidance [28] further improved post-hoc score modeling by employing a discriminator trained to distinguish between real and synthetic data at the latent states x_t. Beyond class-based steering, FreeDoM [60] introduced a more flexible approach by treating the outputs of pre-trained deep learning models as energy functions and steering the inference process to minimize these potentials. More recently, FKS [45] proposed a Sequential Monte Carlo (SMC) based sampling framework. By performing resampling among particles at intermediate states based on their respective rewards, FKS effectively steers the generative trajectories toward the synthesis of high-reward data.

4 Generative Active Learning with Bayesian Acquisition

We consider a setting in which a real-world dataset $\mathcal{R}$ for a supervised learning task is either limited in size or entirely inaccessible, and instead, we generate a synthetic dataset $\mathcal{D}$ that resembles $\mathcal{R}$ using a pretrained diffusion model s_ϕ that has been trained on a large general dataset. This setup is increasingly common in practice: while many generative models are openly released, task-dependent training datasets for supervision are often unavailable. Given a target label $y^{(i)}$, we generate a synthetic sample $x^{(i)}$ by sampling from the diffusion model s_ϕ with conditioning on the label $y^{(i)}$.

Consider a target learning task with a loss–regularizer pair (ℓ, r) and a model f_θ being learned. The goal of Generative Active Learning (GAL) is constructing a synthetic training set $\mathcal{D}$ that is maximally informative with respect to the current training state of the target model f_θ. To achieve this, we propose a novel generative active learning framework, Generative Active Learning via Bayesian Acquisition (GALBA). GALBA integrates BALD (detailed in Section 4.1) with Feynman-Kac Steering (FKS) (detailed in Section 4.2) to steer the data synthesis process toward regions of high epistemic uncertainty. The overall architecture of the proposed framework is illustrated in Fig. 1a.

4.1 GAL via Bayesian Acquisition Function

A key challenge in using synthetic data for training is the presence of artifacts, noise, or structural imperfections, which can introduce undesirable aleatoric uncertainty into the training process [12, 63]. Thus, steering diffusion toward useful samples requires an informativeness metric that distinguishes learnable uncertainty from synthesis-induced ambiguity.

A common baseline is single-model entropy $\mathcal{H}[y|x,\theta]$, which measures *point-estimate uncertainty* because it is computed from a fixed model θ without posterior marginalization. This criterion does not emphasize epistemic uncertainty and can instead favor synthetic samples with high observation noise or artifacts. Fig. 1b highlights the importance of distinguishing between aleatoric and epistemic uncertainty. High aleatoric uncertainty often exhibits unrealistic artifacts, blurring, and noise, which can impede model training. Conversely, high epistemic uncertainty tends to enhance data informativeness while preserving underlying semantic context.

Our goal is to guide generation toward samples with high epistemic uncertainty, improving the target model rather than amplifying synthesis errors. To this end, we treat the target model as a BNN by placing a posterior $p(\theta|\mathcal{D})$ over its parameters, and use the BALD acquisition function in Eq. (6) as the diffusion guidance signal. BALD approximates epistemic uncertainty by measuring the mutual information between the model parameters and the predictive output [23].

Computing BALD exactly is generally intractable. While some Monte Carlo estimators based on MC Dropout [15, 16] or deep ensembles [31] are available, they require multiple forward passes or need additional training costs. This is impractical in our setting since BALD must be evaluated repeatedly during diffusion sampling. We therefore adopt a last-layer Laplace approximation: only the final layer is treated as stochastic, while the preceding layers define a deterministic feature extractor, and the final-layer posterior is approximated by Laplace approximation (LA) [38]. Although this approximation may underrepresent full-network uncertainty, last-layer Bayesian models provide an efficient and competitive alternative [30, 44].

Formally, for a K-way classification problem, we write the network logits as

$$f_\theta(x) = W^\top h_\psi(x), \qquad p_\theta(y|x) = \mathrm{softmax}(W^\top h_\psi(x)), \tag{7}$$

where $h_\psi : \mathcal{X} \to \mathbb{R}^d$ is a deterministic feature extractor and $W \in \mathbb{R}^{d\times K}$ is the final-layer weight matrix. We use the cross-entropy loss and place a Gaussian prior on the vectorized weights, $\mathrm{vec}(W) \sim \mathcal{N}(0, \tau^{-1} I_{Kd})$. We approximate the posterior over W by the last-layer Laplace approximation

$$q(W) = \mathcal{N}\big(\mathrm{vec}(W); \mathrm{vec}(W_\star), H_\star^{-1}\big)\,, \tag{8}$$

where $W_\star$ is the MAP estimate of the final-layer weights. This yields a tractable BALD approximation, given in Lemma 4.1, that can be efficiently evaluated during diffusion sampling.

Lemma 4.1 (BALD with LA)**.** *The BALD computed with LA for $p(W \mid \mathcal{D})$ can be approximated as,*

$$\mathcal{I}[y, W|x, \mathcal{D}] \approx \frac{1}{2}\sum_{c=1}^{K} \hat{p}_c(x)(1 - \hat{p}_c(x))\tilde{v}_c(x), \tag{9}$$

where $\hat{p}_c(x)$ is the predictive probability from Lemma A.3, and $\tilde{v}_c(x) = v_c(x)/(1 + \frac{\pi}{8} v_c(x))$ denotes the rescaled logit variance.

In practice, predictive uncertainty is sensitive to the prior precision τ. We select τ by empirical Bayes, maximizing the Laplace-approximated marginal likelihood

$$\log p(Y \mid X; \tau) = -\frac{1}{2}\log\det(H_\star) - \frac{\tau}{2} W_\star^\top W_\star + \frac{Kd}{2}\log\tau + \mathrm{const.} \tag{10}$$

The full derivation is provided in Section A.1.3.

To assess our BALD estimator in Eq. (9), we compare it with MC Dropout [15] using the same classification backbone. We report Spearman rank correlation and Top-K% overlap between the two scores. As shown in Table 1, our estimator is highly correlated with MC Dropout and yields consistent rankings, validating the use of last-layer LA for uncertainty estimation.

Table 1: Correlation of BALD across the number of samples for MC-dropout and ours.

Samples (T)	50	100	150	200
Spearman corr.	0.87	**0.88**	0.88	0.87
Top-20% overlap	0.58	**0.65**	0.60	0.59
Top-50% overlap	0.84	**0.85**	0.85	0.84

4.2 Applying Feynman-Kac Steering for Diffusion

To preserve data quality while improving the BALD acquisition, we use Feynman–Kac Steering (FKS) [45], an SMC-based steering method [10] that guides the reverse diffusion process without requiring gradients of the reward. FKS samples multiple particles within the trajectories and sequentially resamples them according to their potentials, favoring trajectories with higher reward. This choice addresses key limitations of gradient-based guidance [6, 60, 3]. Backpropagating through the target model at every diffusion step is computationally intensive, and direct score manipulation can induce mode collapse [61] or visual artifacts [8, 42]. Since GAL relies on both the fidelity and diversity of synthetic training samples, such degradations can harm target-model generalization. FKS mitigates these issues by population-based steering rather than direct score modification.

Given a reverse sampling process $\{x_t\}_{t\in[0,T]}$ from $t = T$ to $t = 0$ with marginals $\{p_t\}_{t\in[0,T]}$, FKS aims to tilt the distribution into $\hat{p}_0(x_0) \propto p_0(x_0)\exp(\lambda R(x_0))$, where $R(x_0)$ is the reward function and λ is the guidance scale. For simplicity, we use a discretized trajectory $x_{[T:0]} = (x_T, x_{T-1}, \ldots, x_0)$ with the reverse transition kernel $p_{t|t+1}(x_t|x_{t+1})$. Starting from $\hat{p}_T \propto \mathcal{M}_T p_T$ and given a tilted partial-trajectory distribution $\hat{p}_{t+1}(x_{[T:t+1]})$, FKS updates it given as

$$\hat{p}_t(x_{[T:t]}) \propto p_{t|t+1}(x_t|x_{t+1})\mathcal{M}_t(x_{[T:t]})\hat{p}_{t+1}(x_{[T:t+1]}). \tag{11}$$

where $\mathcal{M}_t$ is a potential function measuring the intermediate importance of the partial trajectory. This is achieved by using S particles initialized as $x_T^{(1)}, \ldots, x_T^{(S)} \sim \hat{p}_T$, propagated by the reverse kernel, and resampled at each step using normalized potentials $\tilde{\mathcal{M}}_t^{(i)} = \frac{\mathcal{M}_t^{(i)}}{\sum_{j=1}^{S} \mathcal{M}_t^{(j)}}$:

$$\{\hat{x}_t^{(i)}\}_{i=1}^{S} \sim \text{Multinomial}(\{x_t^{(i)}\}_{i=1}^{S}, \{\tilde{\mathcal{M}}_t^{(i)}\}_{i=1}^{S}). \tag{12}$$

For the choice of the potential, we use the maximum value of BALD along the trajectory

$$\mathcal{M}_t = \exp\left(\lambda \cdot \max_{t \le j < T} \mathcal{I}[y, W | x_j, \mathcal{D}]\right) \tag{13}$$

with $\mathcal{M}_T = 1$. The detailed algorithm is described in Algorithm 1.

Expected reward of intermediate state x_t. A practical challenge in guiding the reverse SDE is that the target model is trained on clean images, not noisy latent states x_t. Therefore, the latent states x_t cannot be directly proceeded by the target model for reward evaluation. To circumvent this problem, we use the surrogate reward value by projecting the current state x_t onto the data manifold by DDIM prediction [47]; given a noise prediction ϵ_ϕ, it is formulated as:

$$x_{0|t} = (x_t - \sqrt{1 - \alpha_t}\epsilon_\phi)/\sqrt{\alpha_t}, \tag{14}$$

where α_t denotes the predefined variance schedule. We then compute the expected reward $R(x_{0|t})$ and use it as a proxy for the reward of the intermediate state x_t. This approach is consistent with several recent works on diffusion guidance [45, 60, 3] that leverage x_0-prediction to provide guidance signals during the reverse sampling process.

Interval and Adaptive Resampling. Resampling alleviates degeneracy in SMC, but overly frequent resampling can decrease the variance of trajectory-level estimates. We therefore adopt interval and adaptive resampling to control the resampling frequency. See Section B.4 for details.

4.3 Overall Generative Active Learning Procedure

GALBA operates as an iterative loop of target-model training and guided data acquisition. In the **initial phase**, the diffusion model s_ϕ generates an initial synthetic dataset $\mathcal{D}_0$ of size N_0 via unguided sampling; we set $\mathcal{D}_{\text{train}} = \mathcal{D}_0$ and train the target model f_θ on it. In the **training and monitoring phase**, f_θ is periodically evaluated on $\mathcal{D}_{\text{val}}$ using a validation metric v, and a patience parameter γ detects stagnation: if v fails to exceed the historical best $v_{\max}$ for γ consecutive evaluations, the current training set is considered exhausted and acquisition is triggered. In the **guided data acquisition phase**, GALBA synthesizes a new batch $\mathcal{D}_{\text{new}}$ of size N using FKS (Section 4.2) guided by the last-layer Laplace BALD score (Section 4.1), steering generation toward regions of high epistemic uncertainty. The new samples are appended to the training set, $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{new}}$, and training resumes. This cycle continues until the total data budget B is reached. The full workflow is formalized in Algorithm 2 of the Appendix.

5 Experiments

We extensively validate GALBA on image classification and object detection tasks, using Imagenette and ImageNet-100 for classification and MS-COCO for detection. Across these settings, we evaluate whether GALBA improves synthetic data utility and generates informative samples.

5.1 Experiment Setup

Datasets. For classification, we use Imagenette [24] and ImageNet-100 [51], representative subsets of ImageNet [5]. Imagenette contains 10 classes with 9,469 training and 3,925 validation images, while ImageNet-100 contains 100 classes with 126,689 training and 5,000 validation images. For object detection, we use MS-COCO [35], with 118,287 training and 5,000 validation images.

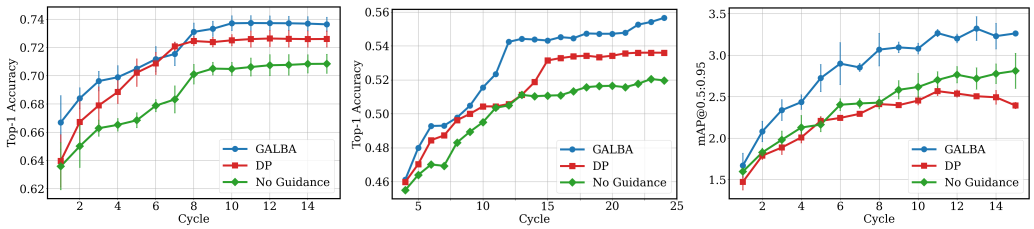


Figure 2: Performance along the GAL cycles. (Left) Top-1 accuracy on ImageNette with 1-σ error bar. (Middle) Top-1 accuracy on ImageNet-100. (Right) mAP on MS-COCO.

Target models. For image classification, we employ ViT-B [9] and evaluate performance using Top-1 validation accuracy. For object detection, we adopt YOLOv11s [27] and report the mean Average Precision (mAP). Similar to Haussmann et al. [19], we compute predictive entropy and BALD for object detection by identifying the anchor box with the highest class probability and estimating uncertainty from its corresponding feature representation. We refer the reader to Section B.1 for the formulations.

Generative models. We use Stable Diffusion v1.5 [41] for text-to-image generation in classification experiments, and GLIGEN [34] for layout-to-image generation in detection experiments. Additional details are provided in Section B.

5.2 Comparison of Effectiveness of Synthetic Data

To evaluate the efficacy of our proposed framework, we compare three distinct scenarios. **1) No Guidance**: A baseline approach where synthetic data is generated via standard diffusion inference without any steering mechanism. **2) Deliberate Practice (DP)** [2]: A gradient-based guidance method that steers the diffusion process to maximize the entropy of the target model's output distribution. Following the original study, we set the gradient guidance scale to 0.05. **3) GALBA (Ours)**: Our proposed framework, which employs FKS to guide the diffusion model toward samples to maximize the BALD score with respect to the target model.

The experiments follow the procedure in Section 4.3. Specifically, the process initiates with the generation of a baseline synthetic pool, comprising 3,200, 32,000, and 10,000 images for Imagenette, ImageNet-100 and MS-COCO, respectively. The size of the generated data for each cycles are 1,600, 16,000, and 5,000 for Imagenette, ImageNet-100, and MS-COCO, respectively. This iterative process is repeated until the total training duration reaches 25k steps for the Imagenette and 50k steps for the ImageNet-100 and MS-COCO. We conduct experiments with 5 different seeds on Imagenette and 3 seeds on MS-COCO for measuring statistical error.

The progression of model performance across successive GAL cycles is illustrated in Fig. 2. Both throughout the training process and at the point of saturation, the target model trained with GALBA-generated data consistently outperforms those trained via the unguided baseline or DP. Moreover, GALBA not only achieves a higher final accuracy but also demonstrates a more efficient learning trajectory, reaching competitive performance levels with fewer synthetic samples.

5.3 Compatibility with Real Dataset

To demonstrate that our methods provide informative signals that complement existing real world data, we conduct experiments on the Imagenette dataset where each cycle involved adding 1,600 synthetic images with a limited budget of 100 randomly sampled real images to the training pool for all GAL scenario. As baselines, we train a model by adding only the 100 real images per cycle, with and without initial synthetic dataset (*Real Data Acq.* and *Pure Real only*, respectively).

We evaluate the models on the Imagenette validation set and three OOD benchmarks: ImageNet-v2 (IN-v2) [39], ImageNet-Sketch (IN-Sk) [55], and ImageNet-A (IN-A) [20]. For these OOD benchmarks, we restrict the evaluation to the shared classes between Imagenette and the validation datasets.

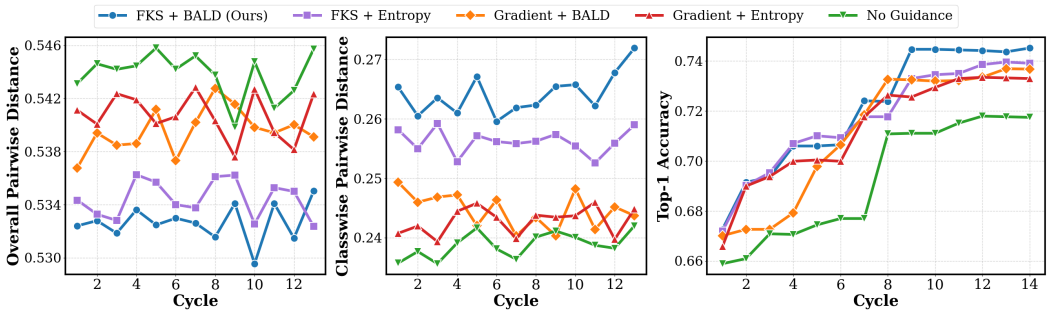


Figure 4: Analysis of (Left) Inter-class and (Middle) intra-class MPD of synthetic data across GAL cycles. (Right) Top-1 performance of the target model on Imagenette.

The resulting dataset sizes and performance metrics at the saturation point are summarized in Table 2. Specifically, GALBA consistently outperforms all baselines, achieving a $+1.5\%$ gain on Imagenette while No Guidance and DP show negligible or negative impact. Notably, GALBA exhibits superior OOD robustness, improving accuracy by $+16.5\%$ on ImageNet-Sketch and $+8.7\%$ on ImageNet-A. These gains confirm that GALBA synthesizes difficult edge cases that effectively expand the model's knowledge manifold and enhance robustness.

Table 2: Top-1 Accuracy on Imagenette and OOD benchmarks using both of synthetic and real data.

Method	**Real**	**Syn.**	**IN-Val**	**IN-v2**	**IN-Sk**	**IN-A**
Pure Real only	1.5k	-	71.8	61.0	32.2	39.8
Real Data Acq.	1.5k	3.2k[1]	78.8	66.0	28.9	47.8
No Guidance	0.9k	19k	78.5	68.0	44.4	43.5
DP	0.9k	19k	79.2	68.0	42.4	39.1
GALBA	0.9k	19k	**80.3**	**69.0**	**45.4**	**56.5**

[1] Initial size N_0 of synthetic train dataset.

6 Discussion

6.1 Analysis of Model-Perceived Uncertainty

To verify whether the target model f_θ perceives the synthesized data as genuinely informative, we evaluate the Negative Log Likelihood (NLL) and the BALD score of the generated samples along the GAL cycles. A high NLL indicates that the model finds the data challenging or identifies it as being outside its current high-confidence regions. Moreover, the BALD score prioritizes epistemic uncertainty from a Bayesian perspective, which we aim to maximize to select the most informative samples.

The results are visualized in Fig. 3. The samples synthesized via GALBA exhibit consistently higher NLL and BALD scores compared to other baselines. Importantly, it indicates that the perceived difficulty from GALBA is not an artifact, but it reflects the presence of informative features targeting the model's knowledge gaps. By synthesizing samples that reside in the regions of high epistemic uncertainty, GALBA provides the target model with informative training signals, which explains the performance gains observed in our primary experiments.

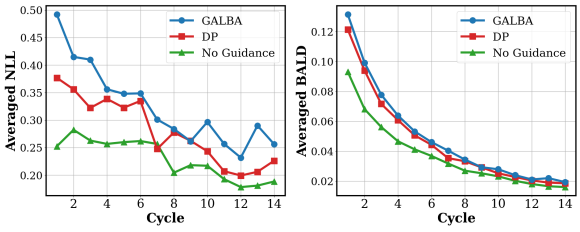


Figure 3: Comparison of model-perceived uncertainty across cycles. (Left) NLL. (Right) BALD.

6.2 Analysis of Discriminability and Diversity

To quantitatively assess the structural diversity and informativeness of the samples synthesized by the GAL framework, we monitor the Mean Pairwise Distances (MPD) of the features of the synthetic data at two levels: inter-class (overall) and intra-class (class-wise). Specifically, a lower inter-class MPD indicates that the generative process effectively targets the decision boundaries between different

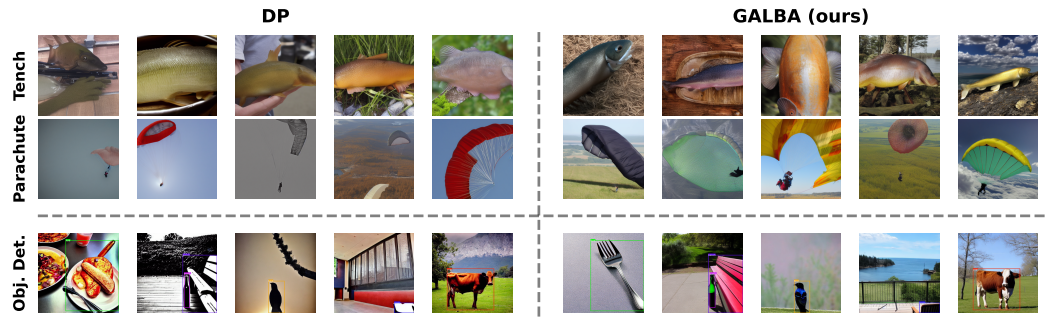


Figure 5: Qualitative comparison of samples produced by DP and GALBA. (Top) Data from Stable Diffusion v1.5 for classification. (Bottom) Data from GLIGEN for object detection.

categories, producing ambiguous and difficult samples that challenge the target model. Conversely, a higher intra-class MPD implies that the model generates a structurally diverse set of samples within each category, ensuring comprehensive coverage of the class-specific data manifold. The detailed formulation of MPD is provided in Section B.5.

To disentangle the contributions of our core components to this improved informativeness, we conduct an ablation study isolating the effects of the steering mechanism (FKS vs. Gradient-based) and the reward metric (BALD vs. Entropy). As illustrated in the left and middle panels of Fig. 4, steering the diffusion process via FKS significantly outperforms gradient-based guidance. The FKS (blue and purple lines) consistently maintain a lower inter-class MPD and a substantially higher intra-class MPD. This demonstrates that FKS explores diverse data manifolds without falling into the mode convergence typically observed in gradient-based methods [61].

Furthermore, comparing the reward metrics within the FKS framework reveals the superiority of our proposed objective. The FKS + BALD configuration (GALBA) further improves upon the FKS + Entropy baseline, achieving the lowest inter-class MPD and the highest intra-class MPD across most active learning cycles. This confirms that the BALD score serves as a more precise acquisition metric than the entropy for identifying high-utility samples that are simultaneously diverse and located near complex decision boundaries, creating a strong synergy with the FKS steering mechanism.

The benefits of generating samples with low inter-class discriminability and high intra-class diversity are empirically validated by the performance of the target model. As depicted in the right panel of Fig. 4, the Top-1 accuracy on the Imagenette dataset is proportional to the quality of the MPD metrics. Consequently, our full method, GALBA, provides the most informative training signals, resulting in the highest performance gains throughout the GAL cycles.

Additionally, while our framework adopts an efficient last-layer Laplace approximation to minimize computational overhead, one might wonder if more standard, yet expensive, methods would yield better guidance. Our extended evaluation in Section C.8 addresses this by demonstrating that our approximation not only maintains comparable uncertainty profiles to these computationally heavy estimators but also achieves superior stability and performance in the GAL pipeline.

These observations are substantiated by the qualitative results presented in Fig. 5. In the upper part, which illustrates the synthetic data for classification tasks, samples synthesized by DP exhibit limited intra-class variety and remain easily discriminable across categories. In contrast, GALBA generates highly diverse samples within each class that are visually challenging to distinguish among classes, corroborating our claim that the framework steers generation toward informative regions near decision boundaries. Furthermore, in the lower part of Fig. 5, the synthetic data for detection tasks, while DP often introduces unrealistic visual artifacts and over-saturation, GALBA mitigates these issues to produce more coherent data. Additional experimental details are provided in Section C.

7 Conclusion

In this paper, we introduced GALBA, a novel GAL framework that steers the diffusion inference process toward generating informative samples. By integrating FKS with an efficient last-layer

Bayesian acquisition function, GALBA effectively overcomes the computational bottlenecks of traditional generative active learning while bypassing the fidelity-diversity trade-offs inherent in gradient-based guidance. Our extensive experiments on Imagenette, ImageNet-100 and MS-COCO datasets confirm that GALBA is a highly effective tool for generating high-quality, diverse, and informative synthetic data. This work highlights the potential of particle-based diffusion steering as a principled approach to autonomous data acquisition from pre-trained generative models.

Limitations and Future Directions. Despite its empirical effectiveness, the scalability of the Hessian approximation remains a bottleneck in high-dimensional settings. Although more memory-efficient than full-network Bayesian alternatives, storing block-diagonal Hessian matrices incurs a complexity of $O(K \cdot D^2)$, where K and D denote the number of classes and feature dimensions, respectively. This quadratic dependency on D can pose substantial memory constraints when dealing with exceptionally high-dimensional features. Furthermore, an overly aggressive guidance scale λ can induce particle degeneracy by prioritizing higher rewards at the expense of sample diversity, as evidenced in Table 13b (Appendix). Lastly, the current formulation of the approximated BALD score in object detection neglects spatial dependencies across image regions, leaving room for further optimization. Consequently, promising avenues for future research include reducing the asymptotic memory complexity, enhancing robustness against the guidance scale, and formulating unified acquisition metrics that jointly account for both classification and regression objectives in object detection.

Acknowledgement

This work was partly supported by Institute for Information & communications Technology Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST); (RS-2024-00469482 & RS-2024-00509279), Global AI Frontier Lab; No.RS2022-II220713, Meta-learning Applicable to Real-world Problems); Cloud TPUs from Google's TPU Research Cloud (TRC).

References

[1] A. Antoniou, A. Storkey, and H. Edwards. Augmenting image classifiers using data augmentation generative adversarial networks. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27*. Springer, 2018.

[2] R. Askari-Hemmat, M. Pezeshki, E. Dohmatob, F. Bordes, P. Astolfi, M. Hall, J. Verbeek, M. Drozdzal, and A. Romero-Soriano. Improving the scaling laws of synthetic data with deliberate practice. In *Proceedings of The 42nd International Conference on Machine Learning (ICML 2025)*, 2025.

[3] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 843–852, June 2023.

[4] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems 35 (NeurIPS 2021)*, 2021.

[7] N. Ding, Y. Chen, B. Xu, Y. Qin, S. Hu, Z. Liu, M. Sun, and B. Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[8] A.-D. Dinh, D. Liu, and C. Xu. Rethinking conditional diffusion sampling with progressive guidance. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 37 (NeurIPS 2023)*, 2023.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

[10] A. Doucet, N. De Freitas, and N. Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*. Springer, 2001.

[11] L. Dunlap, A. Umino, H. Zhang, J. Yang, J. E. Gonzalez, and T. Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2023)*, 2023.

[12] L. Fan, K. Chen, D. Krishnan, D. Katabi, P. Isola, and Y. Tian. Scaling laws of synthetic images for model training... for now. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] H. Fang, B. Han, S. Zhang, S. Zhou, C. Hu, and W.-M. Ye. Data augmentation for object detection via controllable diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1257–1266, 2024.

[14] Y. Fang, Q. Sun, X. Wang, T. Huang, X. Wang, and Y. Cao. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149:105171, 2024.

[15] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning (ICML 2016)*, 2016.

[16] Y. Gal, R. Islam, and Z. Ghahramani. Deep Bayesian active learning with image data. In *Proceedings of The 34th International Conference on Machine Learning (ICML 2017)*, 2017.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 28 (NIPS 2014)*, 2014.

[18] J. Harrison, J. Willes, and J. Snoek. Variational bayesian last layers, 2024. URL `https://arxiv.org/abs/2404.11599`.

[19] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez. Scalable active learning for object detection. In *2020 IEEE intelligent vehicles symposium (iv)*, 2020.

[20] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[22] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2020)*, 2020.

[23] N. Houlsby, F. Huzár, Z. Ghahramani, and M. Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[24] J. Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. URL `https://github.com/fastai/imagenette`.

[25] T. Huang, J. Liu, S. You, and C. Xu. Active generation for image classification, 2024. URL `https://arxiv.org/abs/2403.06517`.

[26] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2022)*, 2022.

[27] R. Khanam and M. Hussain. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*, 2024.

[28] D. Kim, Y. Kim, W. Kang, and I.-C. Moon. Refining generative process with discriminator guidance in score-based diffusion models. In *Proceedings of The 40th International Conference on Machine Learning (ICML 2023)*, 2023.

[29] A. Kirsch, J. van Amersfoort, and Y. Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2019)*, 2019.

[30] A. Kristiadi, M. Hein, and P. Hennig. Being bayesian, even just a bit, fixes overconfidence in ReLU networks. In *Proceedings of The 37th International Conference on Machine Learning (ICML 2020)*, 2020.

[31] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31 (NIPS 2017)*, 2017.

[32] H. Lee, S. Lee, and J. Lee. Variational bayesian pseudo-coreset. In *International Conference on Learning Representations (ICLR)*, 2025.

[33] S. Lee, G. Nam, M. Choi, H. Lee, and J. Lee. PANGEA: Projection-based augmentation with non-relevant general data for enhanced domain adaptation in LLMs. In *Advances in Neural Information Processing Systems 39 (NeurIPS 2025)*, 2025.

[34] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. GLIGEN: Open-Set Grounded Text-to-Image Generation, Apr. 2023. URL `http://arxiv.org/abs/2301.07093`. arXiv:2301.07093 [cs].

[35] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick. Microsoft coco: Common objects in context. In *The European Conference on Computer Vision (ECCV)*, 2014.

[36] I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019.

[37] Z. Lu, E. Ie, and F. Sha. Mean-field approximation to gaussian-softmax integral with application to uncertainty estimation. *arXiv preprint arXiv:2006.07584*, 2020.

[38] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[39] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[40] H. Ritter, A. Botev, and D. Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[41] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[42] S. Sadat, O. Hilliges, and R. M. Weber. Eliminating oversaturation and artifacts of high guidance scales in diffusion models. In *International Conference on Learning Representations (ICLR)*, 2025.

[43] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*, 2018.

[44] M. Sharma, S. Farquhar, E. Nalisnick, and T. Rainforth. Do Bayesian neural networks need to be fully stochastic? In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023*, 2023.

[45] R. Singhal, Z. Horvitz, R. Teehan, M. Ren, Z. Yu, K. McKeown, and R. Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Proceedings of The 42nd International Conference on Machine Learning (ICML 2025)*, 2025.

[46] S. Sinha, S. Ebrahimi, and T. Darrell. Variational adversarial active learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[47] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[48] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

[49] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of The 30th International Conference on Machine Learning (ICML 2013)*, 2013.

[50] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[51] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding, 2020. URL `https://arxiv.org/abs/1906.05849`.

[52] B. Trabucco, K. Doherty, M. A. Gurinas, and R. Salakhutdinov. Effective data augmentation with diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

[53] T. Tran, T.-T. Do, I. D. Reid, and G. Carneiro. Bayesian generative active deep learning. In *Proceedings of The 36th International Conference on Machine Learning (ICML 2019)*, 2019.

[54] P. Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 2011.

[55] H. Wang, S. Ge, Z. Lipton, and E. P. Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2019)*, pages 10506–10518, 2019.

[56] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[57] Z. Wang, L. Wei, T. Wang, H. Chen, Y. Hao, X. Wang, X. He, and Q. Tian. Enhance image classification via inter-class image mixup with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17223–17233, 2024.

[58] K. Wen, Z. Li, J. S. Wang, D. L. W. Hall, P. Liang, and T. Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape view. In *International Conference on Learning Representations (ICLR)*, 2025.

[59] L. Wu, B. L. Trippe, C. A. Naesseth, J. P. Cunningham, and D. Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2023)*, 2023.

[60] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[61] X. Zhao and A. G. Schwing. Studying classifier(-free) guidance from a classifier-centric perspective, 2026.

[62] J.-J. Zhu and J. Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

[63] M. Zhu, C. Fan, H. Chen, Y. Liu, W. Mao, X. Xu, and C. Shen. Generative active learning for long-tailed instance segmentation. In *Proceedings of The 41st International Conference on Machine Learning (ICML 2024)*, 2024.

Algorithm 1 Feynman-Kac Steering for Diffusion

Require: Transition kernel $p_{t|t+1}(x_t|x_{t+1})$, number of particles S, reward function R, guidance scale λ, resampling intervals $\mathcal{T}_{\text{resam}}$, noise scheduler α_t.

Draw initial particles $\{x_T^i\}_{i=1}^S \sim p(x_T)$.
for $t \in \{T, \dots, 1\}$ **do**
  Sample $x_{t-1}^{(i)} \sim p_\phi(x_{t-1}|x_t^{(i)})$ for each $i \in [S]$.
  Estimate $x_{0|t-1}^{(i)}$ using Eq. (14).
  Calculate $\mathcal{M}_{t-1}^i$ using Eq. (13).
  Resample $\{\hat{x}_{t-1}^{(i)}\}_{i=1}^S$ using Eq. (12).
end for
Return $x_0^{(i)}$ with the highest reward $R(x_0^{(i)})$ among the final particles.

Algorithm 2 Overall Generative Active Learning Pipeline

Require: Target model f_θ, pre-trained diffusion model s_ϕ, initial size N_0, step size N, validation metric $V(\cdot)$, patience threshold γ, total budget B.

Initial Phase:
$\mathcal{D}^{\text{train}} \leftarrow$ Generate N_0 samples from s_ϕ without guidance.
Initialize $v_{\text{max}} \leftarrow 0$, $C \leftarrow 0$, $N_{\text{total}} \leftarrow N_0$.
while $N_{\text{total}} \leq B$ **do**
  Training Phase:
  Update θ by minimizing loss on $\mathcal{D}_{\text{train}}$.
  $v \leftarrow V(f_\theta)$ (Evaluate on $\mathcal{D}_{\text{val}}$)
  if $v > v_{\text{max}}$ **then**
    $v_{\text{max}} \leftarrow v$, $C \leftarrow 0$ (Improvement found)
  else
    $C \leftarrow C + 1$ (Increment patience counter)
  end if
  if $C \geq \gamma$ **then**
    Data Acquisition Phase:
    $\mathcal{D}_{\text{new}} \leftarrow$ Generate N samples using Algorithm 1.
    $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{new}}$.
    $N_{\text{total}} \leftarrow N_{\text{total}} + N$.
    Reset $C \leftarrow 0$.
  end if
end while

A Various Proofs and Derivations

A.1 Uncertainty Quantification for Last-Layer Parameters

We consider a K-way classification framework f_θ with parameters $\theta = (\psi, W)$, decomposed by a fixed feature extractor $h_\psi : \mathcal{X} \to \mathbb{R}^d$ followed by a linear output layer with weights $W \in \mathbb{R}^{d\times K}$, where d denotes the feature dimension and K represents the number of classes. For a given input x, the logits and their corresponding softmax probabilities are defined as $f_\theta(x) = W^\top h_\psi(x) \in \mathbb{R}^K$ and $p_\theta(x) = \text{softmax}(f_\theta(x)) \in \mathbb{R}^K$, respectively.

By assuming a Gaussian prior on the weights, $W \sim \mathcal{N}(0, \tau^{-1} I_{Kd})$, the model's objective function aligns with the Negative Log-Likelihood (NLL) incorporating ℓ_2 regularization:

$$L(W) = -\sum_{i=1}^{n} y^{(i)} \log p_\theta(y^{(i)}|x^{(i)}; W) + \frac{\tau}{2}\|W\|_2^2, \tag{15}$$

where $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ denotes the training dataset and $y^{(i)}$ is the one-hot encoded ground-truth label.

Upon obtaining the Maximum A Posteriori (MAP) estimate $W_\star = \arg\min_W L(W)$, we quantify the predictive uncertainty via the Laplace approximation. This method approximates the posterior distribution as a Gaussian, $W|D \sim \mathcal{N}(W_\star, \Sigma_\star)$, centered at the MAP estimate. The covariance matrix $\Sigma_\star$ is determined by the inverse of the Hessian of the objective function $L(W)$ evaluated at $W_\star$ (i.e., $\Sigma_\star = H_\star^{-1}$, where $H = \nabla^2 L(W)$). Hereafter, we suppress the deterministic feature-extractor parameters ψ from the notation for simplicity.

Lemma A.1 (Covariance of the last-layer parameters under Laplace approximation)**.** *Under the Laplace approximation, the precision matrix (the inverse of the covariance matrix) of the last-layer weights corresponds to the Hessian of the objective function $L(W)$. This Hessian can be approximated by a block-diagonal matrix:*

$$\Sigma_\star^{-1} = H_\star \approx \bigoplus_{c=1}^{K} H_c, \quad \textit{where} \quad H_c = \sum_{i=1}^{n} p_{ic}(1-p_{ic}) h(x^{(i)}) h(x^{(i)})^\top + \tau I_d, \tag{16}$$

where p_{ic} denotes the predicted probability of the c-th class for input x^i, $\oplus$ represents the direct sum (block-diagonal concatenation) of matrices, and $h(x^{(i)})$ is the feature vector.

Proof. The Jacobian of the cross-entropy loss $L(W)$ with respect to the weights W is derived via the chain rule. Let $f_\theta \in \mathbb{R}^K$ be the logits and $p = \text{softmax}(f_\theta)$ be the predictive probabilities. The gradient with respect to W is:

$$\frac{\partial L(W)}{\partial W} = \frac{\partial L(W)}{\partial f} \cdot \frac{\partial f}{\partial W}. \tag{17}$$

The derivative of the loss with respect to the logits is $\frac{\partial L}{\partial f} = (p - y)^\top$. Since the logit function $f_\theta(x) = W^\top h(x)$ is affine with respect to W, the Jacobian of the logits with respect to the weights for each class c is simply the feature vector $h(x)$. Specifically, for the vectorized weights, $\frac{\partial f_\theta}{\partial \text{vec}(W)} = I_K \otimes h(x)^\top$.

To obtain the Hessian, we apply the second-order chain rule:

$$H_\star = \sum_{i=1}^{n} \left(\frac{\partial f_\theta(x^{(i)})}{\partial W} \right)^\top \nabla_f^2 L(W) \left(\frac{\partial f_\theta(x^{(i)})}{\partial W} \right) + \tau I_{dK}. \tag{18}$$

Since the second derivative of the loss with respect to the logits is the negative of the softmax Jacobian, $\Omega = \nabla_f^2 L(W) = \text{diag}(p) - pp^\top$, we have:

$$H_\star = \sum_{i=1}^{n} (I_K \otimes h(x^{(i)})) \Omega_i (I_K \otimes h(x^{(i)})^\top) + \tau I_{dK} = \sum_{i=1}^{n} (\Omega_i \otimes h(x^{(i)}) h(x^{(i)})^\top) + \tau I_{dK}. \tag{19}$$

For computational tractability and to focus on class-specific uncertainty, we adopt a block-diagonal approximation. This assumes that the weights associated with different classes are independent in the posterior distribution (i.e., we ignore the off-diagonal blocks of the Hessian where $c \neq c'$). Under this assumption, the Hessian becomes the direct sum of class-specific matrices H_c. The c-th diagonal element of Ω_i is $p_{ic}(1 - p_{ic})$, leading to:

$$H_c = \sum_{i=1}^{n} p_{ic}(1-p_{ic}) h(x^{(i)}) h(x^{(i)})^\top + \tau I_d. \tag{20}$$

This completes the proof.

□

Lemma A.2 (Predictive Distribution of Logits)**.** *Let $f_\theta(x) \in \mathbb{R}^K$ denote the logit vector for a given input x, where $f_c(x)$ is the c-th element corresponding to class $c \in \{1, \ldots, K\}$. Under the Laplace approximation of the posterior $p(W|\mathcal{D})$, the predictive distribution of the logits is a multivariate Gaussian distribution:*

$$f_\theta(x) \sim \mathcal{N}\left(\mu_f(x), \Sigma_f(x)\right). \tag{21}$$

Furthermore, under the block-diagonal approximation of the Hessian, the marginal distribution for each class c is given by:

$$f_{\theta,c}(x) := f_c(x) \sim \mathcal{N}\left(\mu_c(x), v_c(x)\right), \tag{22}$$

where $\mu_c(x) = w_{\star,c}^\top h(x)$, $w_{\star,c}$ is a vectorized last-layer weight vector for class c, and $v_c(x) = h(x)^\top H_c^{-1} h(x)$.

Proof. The logit vector $s(x)$ is defined as an affine transformation of the last-layer weights W. By let $w = \text{vec}(W) \in \mathbb{R}^{dK}$ be the vectorized weight parameters, the logit vector can be expressed as:

$$f_\theta(x) = W^\top h(x) = (I_K \otimes h(x)^\top) w. \tag{23}$$

Recall from the Laplace approximation that the posterior distribution of the weights is Gaussian, $w \sim \mathcal{N}(\text{vec}(W_\star), \Sigma_\star)$, where $\Sigma_\star = H_\star^{-1}$. Since any linear transformation of a Gaussian random variable remains Gaussian, the logit vector $f_\theta(x)$ also follows a Gaussian distribution. The mean and covariance of $f_\theta(x)$ are derived as follows:

$$\mu_f(x) = \mathbb{E}[W^\top h(x)] = W_\star^\top h(x), \tag{24}$$

$$\Sigma_f(x) = (I_K \otimes h(x)^\top) \Sigma_\star (I_K \otimes h(x)). \tag{25}$$

Following the block-diagonal approximation of the Hessian $H_\star \approx \bigoplus_{c=1}^K H_c$ introduced in Lemma A.1, the covariance matrix $\Sigma_\star = H_\star^{-1}$ also adopts a block-diagonal structure, $\Sigma_\star \approx \bigoplus_{c=1}^K H_c^{-1}$. Substituting this into the expression for $\Sigma_s(x)$, we obtain:

$$\Sigma_f(x) \approx (I_K \otimes h(x)^\top) \left(\bigoplus_{c=1}^K H_c^{-1} \right) (I_K \otimes h(x)) = \bigoplus_{c=1}^K \left(h(x)^\top H_c^{-1} h(x) \right). \tag{26}$$

The resulting logit covariance matrix $\Sigma_f(x)$ is diagonal, implying that the logits for different classes are independent. Therefore, the distribution for the c-th class logit simplifies to:

$$f_c(x) \sim \mathcal{N}\left(w_{\star,c}^\top h(x), h(x)^\top H_c^{-1} h(x) \right) = \mathcal{N}\left(\mu_c(x), v_c(x)\right). \tag{27}$$

This completes the proof.

□

A.1.1 Predictive Entropy for Last-layer Uncertainty

Lemma A.3. *For a given input x, the predictive entropy under the Laplace approximation is given by:*

$$\mathcal{H}[y|x, \mathcal{D}] \approx -\sum_{c=1}^K \hat{p}_c(x) \log \hat{p}_c(x) \tag{28}$$

where $\hat{p}_c(x)$ is the predictive probability of class c, computed via a softmax function over the scaled logits:

$$\hat{f}_c(x) = \frac{\mu_c(x)}{\sqrt{1 + \frac{\pi}{8} v_c(x)}}. \tag{29}$$

Here, $\mu_c(x) = w_{\star,c}^\top h(x)$ and $v_c(x) = h(x)^\top H_c^{-1} h(x)$ represent the mean and variance of the c-th logit, respectively, as derived from Lemma A.2.

Proof. The predictive distribution is obtained by marginalizing the softmax likelihood over the Gaussian posterior of the weights:

$$p(y = c|x, \mathcal{D}) = \mathbb{E}_{f\sim\mathcal{N}(\mu_f,\Sigma_f)}[p(y = c|f)] \tag{30}$$

$$= \int \frac{\exp(f_c(x))}{\sum_{i=1}^{K}\exp(f_i(x))}\mathcal{N}(f(x); \mu_f, \Sigma_f)df. \tag{31}$$

Since the integral in Eq. (31) is analytically intractable due to the softmax nonlinearity, we utilize the probit approximation [4]. First, we rewrite the softmax function for class c in terms of sigmoids:

$$\mathbb{E}_{W|\mathcal{D}}\left[\frac{\exp(f_c(x))}{\sum_{i=1}^{K}\exp(f_i(x))}\right] = \int \frac{\exp(f_c(x)))}{\sum_{i=1}^{K}\exp(f_i(x))}\mathcal{N}(\mu_f, \Sigma_f)df \tag{32}$$

$$= \int \frac{1}{1 + \sum_{i\neq c}\exp(-(f_c(x) - f_i(x)))}\mathcal{N}(\mu_f, \Sigma_f)df \tag{33}$$

$$= \int \left(1 + \sum_{i\neq c}\exp(-(f_c(x) - f_i(x)))\right)^{-1} \mathcal{N}(\mu_f, \Sigma_f)df \tag{34}$$

$$= \int \left(2 - K + \sum_{i\neq c}(1 + \exp(-(f_c(x) - f_i(x)))\right)^{-1} \mathcal{N}(\mu_f, \Sigma_f)df \tag{35}$$

$$= \int \left(2 - K + \sum_{i\neq c}\frac{1}{\sigma(f_c(x) - f_i(x))}\right)^{-1} \mathcal{N}(\mu_f, \Sigma_f)df \tag{36}$$

Here, σ is sigmoid function. We can apply mean field approximation [37, 32] and rewrite the formula as follows:

$$\int \left(2 - K + \sum_{i\neq c}\frac{1}{\sigma(f_c(x) - f_i(x))}\right)^{-1} \mathcal{N}(\mu_f, \Sigma_f)df \tag{37}$$

$$\approx \left(2 - K + \sum_{i\neq c}\frac{1}{\mathbb{E}[\sigma(f_c(x) - f_i(x))]}\right)^{-1} \tag{38}$$

$$= \left(2 - K + \sum_{i\neq c}\frac{1}{\sigma\left(\frac{\mu_c}{\sqrt{1+\frac{\pi}{8}v_c}} - \frac{\mu_i}{\sqrt{1+\frac{\pi}{8}v_i}}\right)}\right)^{-1} \tag{39}$$

$$= \left(1 + \sum_{i\neq c}\exp\left(\frac{\mu_c}{\sqrt{1+\frac{\pi}{8}v_c}} - \frac{\mu_i}{\sqrt{1+\frac{\pi}{8}v_i}}\right)\right)^{-1} \tag{40}$$

$$= \frac{\exp(\mu_c/\sqrt{1+\frac{\pi}{8}v_c})}{\sum_{i=1}^{K}\exp(\mu_i/\sqrt{1+\frac{\pi}{8}v_i})} \tag{41}$$

Following this approach, the predictive probability $p(y = c|x, \mathcal{D})$ is simplified to a softmax function over the scaled logits $\hat{s}_c(x)$:

$$p(y = c|x, \mathcal{D}) \approx \frac{\exp(\hat{f}_c(x))}{\sum_{i=1}^{K}\exp(\hat{f}_i(x))} = \hat{p}_c(x), \tag{42}$$

where $\hat{f}_c(x) = \mu_c(x)/\sqrt{1 + \frac{\pi}{8} v_c(x)}$. Finally, the predictive entropy is calculated by substituting these approximated probabilities into the entropy definition:

$$\mathcal{H}[y|x, \mathcal{D}] = -\sum_{c=1}^{K} p(y = c|x, \mathcal{D}) \log p(y = c|x, \mathcal{D}) \tag{43}$$

$$\approx -\sum_{c=1}^{K} \hat{p}_c(x) \log \hat{p}_c(x). \tag{44}$$

This completes the proof.

□

A.1.2 BALD for Last-layer parameters

Lemma A.4 (BALD for Last-layer parameters)**.** *For an input x, the BALD score can be approximated under the Laplace approximation as follows:*

$$\mathcal{I}[y, W|x, \mathcal{D}] \approx \frac{1}{2} \sum_{c=1}^{K} \hat{p}_c(x)(1 - \hat{p}_c(x))\tilde{v}_c(x), \tag{45}$$

where $\hat{p}_c(x)$ is the predictive probability from Lemma A.3, and $\tilde{v}_c(x) = v_c(x)/(1 + \frac{\pi}{8} v_c(x))$ denotes the rescaled logit variance.

Proof. The BALD score is defined as the difference between the entropy of the predictive distribution and the expected entropy of the model's output:

$$\mathcal{I}[y, W|x, \mathcal{D}] = \mathcal{H}[p(y|x, \mathcal{D})] - \mathbb{E}_{p(W|\mathcal{D})}[\mathcal{H}[p(y|x, W)]]. \tag{46}$$

The first term is the predictive entropy derived in Lemma A.3. To approximate the second term, we consider the entropy $\mathcal{H}(p)$ as a function of the probability vector $p = \text{softmax}(s)$. Applying a second-order Taylor expansion to the expected entropy around the predictive mean $\hat{p} = \mathbb{E}[p]$, we obtain:

$$\mathbb{E}_{p(W|\mathcal{D})}[\mathcal{H}(p)] \approx \mathcal{H}(\hat{p}) + \frac{1}{2} \text{tr}\left(\nabla_p^2 \mathcal{H}(\hat{p}) \Sigma_p\right), \tag{47}$$

where Σ_p is the covariance matrix of the probability vector p. The Hessian of the entropy with respect to the probabilities is given by $\nabla_p^2 \mathcal{H}(p) = -\text{diag}(1/p)$.

To estimate Σ_p, we can use the 2nd Taylor approximation of variance. Let J be the Jacobian of the softmax function with respect to the scaled logits $\hat{s}$. The covariance Σ_p can be approximated as:

$$\Sigma_p \approx J \tilde{\Sigma}_f J^\top, \tag{48}$$

where $\tilde{\Sigma}_f = \text{diag}(\tilde{v}_1, \ldots, \tilde{v}_K)$ is the diagonal covariance matrix of the effective logits, with $\tilde{v}_c(x) = v_c(x)/(1 + \frac{\pi}{8} v_c(x))$. The Jacobian of the softmax function is $J = \text{diag}(\hat{p}) - \hat{p}\hat{p}^\top$. Substituting these into the trace term:

$$\text{tr}\left(\nabla_p^2 \mathcal{H}(\hat{p}) \Sigma_p\right) = \text{tr}\left(-\text{diag}(1/\hat{p}) J \tilde{\Sigma}_f J^\top\right) \tag{49}$$

$$= -\text{tr}\left(J^\top \text{diag}(1/\hat{p}) J \tilde{\Sigma}_f\right). \tag{50}$$

Expanding the matrix product $J^\top \text{diag}(1/\hat{p}) J$:

$$J^\top \text{diag}(1/\hat{p}) J = (\text{diag}(\hat{p}) - \hat{p}\hat{p}^\top)\text{diag}(1/\hat{p})(\text{diag}(\hat{p}) - \hat{p}\hat{p}^\top) \tag{51}$$

$$= \text{diag}(\hat{p}) - \hat{p}\hat{p}^\top. \tag{52}$$

Thus, the trace becomes:

$$-\text{tr}\left((\text{diag}(\hat{p}) - \hat{p}\hat{p}^\top)\tilde{\Sigma}_f\right) = -\sum_{c=1}^{K} \hat{p}_c(1 - \hat{p}_c)\tilde{v}_c(x). \tag{53}$$

Finally, substituting this back into the BALD expression:

$$\mathcal{I}[y, W|x, \mathcal{D}] \approx \mathcal{H}(\hat{p}) - \left(\mathcal{H}(\hat{p}) - \frac{1}{2} \sum_{c=1}^{K} \hat{p}_c(1 - \hat{p}_c)\tilde{v}_c(x) \right) \tag{54}$$

$$= \frac{1}{2} \sum_{c=1}^{K} \hat{p}_c(1 - \hat{p}_c)\tilde{v}_c(x). \tag{55}$$

This completes the proof.

□

A.1.3 Hyperparameter Optimization via Marginal Likelihood

To successfully derive the predictive distribution via Laplace approximation, it is crucial to appropriately configure the Gaussian prior precision τ of the last layer. In this study, we optimize τ by maximizing the marginal likelihood (i.e., the evidence). This strategy ensures that the prior is well-calibrated to the observed data, thereby enhancing the reliability of the resulting posterior approximation.

Lemma A.5 (Log-Marginal Likelihood for the Last Layer)**.** *The log-marginal likelihood* $\log p(Y|X;\tau)$*, which represents the evidence of the hyperparameter* τ *given the training data, can be approximated as follows:*

$$\log p(Y|X;\tau) \approx -\frac{\tau}{2} W_\star^\top W_\star - \frac{1}{2} \log\det(H_\star) + \frac{dK}{2} \log\tau + C, \tag{56}$$

where $W_\star$ *is the MAP estimate,* $H_\star = \nabla^2 L(W_\star)$ *is the Hessian matrix and* C *is a constant independent of* τ*.*

Proof. The marginal likelihood is obtained by integrating the product of the likelihood and the prior over the parameter space:

$$p(Y|X;\tau) = \int p(Y|X, W)p(W|\tau)dW. \tag{57}$$

Let $E(W) = -\log p(Y|X, W) - \log p(W|\tau)$ be the negative log-posterior (equivalent to our objective function $L(W)$ excluding constant terms). Using a second-order Taylor expansion of $E(W)$ around the MAP estimate $\hat{W}$, we have:

$$E(W) \approx E(\hat{W}) + \frac{1}{2}(W - W_\star)^\top H(W - W_\star), \tag{58}$$

where H is the Hessian evaluated at $\hat{W}$. Substituting this expansion into the integral:

$$p(Y|X;\tau) \approx \exp(-E(W_\star)) \int \exp\left(-\frac{1}{2}(W - W_\star)^\top H(W - W_\star) \right) dW \tag{59}$$

$$= \exp(-E(W_\star))(2\pi)^{dK/2} \det(H_\star)^{-1/2}. \tag{60}$$

Taking the logarithm, we obtain:

$$\log p(Y|X;\tau) \approx -E(W_\star) - \frac{1}{2} \log\det(H_\star) + \frac{Kd}{2} \log(2\pi). \tag{61}$$

Recalling the definition of $E(W_\star)$ and the Gaussian prior $p(W|\tau) = (\frac{\tau}{2\pi})^{Kd/2} \exp(-\frac{\tau}{2}\|W\|^2)$, we have:

$$-E(\hat{W}) = \log p(Y|X, W_\star) - \frac{\tau}{2} W_\star^\top W_\star + \frac{Kd}{2} \log\tau - \frac{Kd}{2} \log(2\pi). \tag{62}$$

Substituting this back into the log-marginal likelihood expression. With a fixed last-layer parameters $\hat{W}$, the log likelihood $\log p(Y|X;\tau)$ become constant. Gathering the constants into C:

$$\log p(Y|X;\tau) \approx -\frac{\tau}{2} W_\star^\top W_\star - \frac{1}{2} \log\det(H_\star) + \frac{Kd}{2} \log\tau + C. \tag{63}$$

This completes the proof.

□

B Experimental Details

B.1 Extending LLLA-BALD to Object Detection

To adapt last-layer Laplace BALD (LLLA-BALD) to object detection architectures, we apply the Laplace approximation exclusively to the classification head while maintaining deterministic bounding box regression heads. Specifically, given the set of all predicted bounding box candidates $\mathcal{B}$, we first compute their classification probabilities across all classes $\mathcal{C}$. To represent the overall predictive uncertainty of the image, we identify the most confident candidate $b^* \in \mathcal{B}$ with the maximum predicted class probability:

$$b^* = \arg\max_{b \in \mathcal{B}} \left(\max_{c \in \mathcal{C}} p_{b,c} \right). \tag{64}$$

Conditioned on the latent classification feature associated with this representative proposal b^*, we evaluate its last-layer BALD score $I(b^*)$ via Theorem 4.1:

$$I(b^*) = \frac{1}{2} \sum_{c \in \mathcal{C}} \hat{p}_{b^*,c}(1 - p_{b^*,c}) \tilde{v}_{b^*,c}. \tag{65}$$

Finally, we assign this instance-level uncertainty $I(b^*)$ as the overall informativeness reward for the input image.

It is worth noting that this representative selection exclusively reflects semantic uncertainty from the classification head, whereas bounding box regression is equally critical for comprehensive detection performance. Consequently, incorporating spatial uncertainty or localization-aware metrics into the image-level guidance reward represents a promising avenue for further performance improvements.

B.2 Precomputing and Caching Class-wise Hessians

To optimize the computational efficiency of the guided data acquisition phase, we implement a precomputation strategy for the class-wise Hessians of the target model's last-layer parameters. Since the Hessian matrix, derived from the Laplace approximation, depends solely on the feature representations of the existing synthetic training set and the current optimized state of the target model's final layer, it remains static throughout the acquisition process within a single GAL cycle.

Specifically, Immediately following the training phase and prior to the guided synthesis of new samples, we perform a single forward pass over the synthetic training corpus to retrieve the feature vectors and their corresponding probability vectors. Utilizing these values, we compute the class-wise Hessians as formulated in Lemma A.1.

In practice, the BALD score calculation requires the inverse of the Hessian matrix to compute the posterior variance. To avoid the high computational cost and numerical instability associated with direct matrix inversion, we store the Cholesky factors of each class-wise Hessian. During the subsequent acquisition phase, the product of the newly evaluated input's logit vector and the inverse Hessian is efficiently computed via Cholesky solving. This caching mechanism significantly reduces the per-step latency of the FKS-guided diffusion process, enabling the framework to scale effectively even with high-dimensional last-layer parameters.

B.3 Adaptive Validation Patience

In the implementation of GALBA (Section 4.3), we observe that employing a fixed validation patience threshold, the number of consecutive evaluation intervals permitted without performance improvement, can lead to premature convergence. In a GAL framework, the training set expands at each cycle with increasingly synthetic samples. If the patience threshold remains static, the target model may trigger early stopping before it has sufficiently assimilated the information from the newly incorporated data.

To mitigate this, we employ a heuristic for adaptive validation patience, where the threshold γ scales linearly with the progression of GAL cycles. Formally, the patience threshold for the c-th cycle is

defined as:

$$\gamma_c = c \cdot \gamma_0 \tag{66}$$

where γ_0 represents the initial patience threshold, and c is the number of GAL cycles proceeding. This adaptive scaling ensures that the target model is afforded adequate training iterations to incorporate the expanding and increasingly complex data manifold, thereby preventing premature termination of the learning process as the dataset grows.

B.4 Interval and Adaptive Resampling

In Feynman-Kac Steering (mentioned in Section 4.2), frequent resampling in SMC can lead to particle deprivation, where diversity is lost as particles collapse into a few high-likelihood trajectories. To mitigate this, we adopt two strategies:

1. **Interval Resampling:** Resampling is performed only at predefined time steps $\mathcal{T}_{\text{resam}}$ rather than at every iteration.
2. **Adaptive Resampling:** Resampling is triggered only when the Effective Sample Size (ESS) falls below a certain threshold, ensuring that resampling is conducted only when the particle weights are sufficiently skewed. Formally, the ESS can be represented as: $\text{ESS}(\{M_t^i\}) = 1/\sum_{i=1}^{P}\left(\bar{\mathcal{M}}_t^i\right)$ where $\bar{\mathcal{M}}_t^i$ is the normalized potential by their mean. If $\text{ESS}(\{M_t^i\}) < P/2$, then the resampling will be skipped.

B.5 Calculation of Pairwise Distances

The dispersion between the latent features of data is often used as the indicative proxy metrics of structural diversity and coverage of the dataset [46, 43]. Therefore, we use the mean pairwise distance (MPD) for measuring the discriminability and diversity of the data instances.

Specifically, we extract the latent feature vector $z_i \in \mathbb{R}^d$ for each sample using a pretrained Inception-V3 [50], normalized such that $\|z_i\|_2 = 1$. Let $\mathcal{P}_c = \{(i,j) \mid y_i = y_j = c, i < j\}$ denote the set of all unique pairs of synthetic samples belonging to class c. The intra-class MPD is computed as the average pairwise cosine distance within each class and averaged across all K classes:

$$\text{MPD}_{\text{intra}} = \frac{1}{K}\sum_{c=1}^{K}\frac{1}{|\mathcal{P}_c|}\sum_{(i,j)\in\mathcal{P}_c}(1 - z_i^{\intercal} z_j). \tag{67}$$

Similarly, letting $\mathcal{P}_{\text{all}} = \{(i,j) \mid i < j\}$ be the set of all unique pairs across the entire synthetic dataset regardless of class labels, the inter-class MPD is defined as:

$$\text{MPD}_{\text{inter}} = \frac{1}{|\mathcal{P}_{\text{all}}|}\sum_{(i,j)\in\mathcal{P}_{\text{all}}}(1 - z_i^{\intercal} z_j). \tag{68}$$

A lower $\text{MPD}_{\text{inter}}$ indicates that the generated samples cluster near decision boundaries, making classification more challenging and informative for the downstream model.

As depicted in Section 6.2, GALBA consistently achieves the highest intra-class MPD and lowest inter-class MPD, quantitatively confirming its ability to synthesize diverse yet boundary-challenging data.

B.6 Implementation Details

Framework and Hardware. All experiments were implemented using the PyTorch framework. For experiments on Imagenette and MS-COCO, we utilized 4 NVIDIA RTX 3090 GPUs. For the larger-scale ImageNet-100 experiments, we employed a cluster of eight TPU VM nodes, each equipped with four TPU v5e chips, to handle the increased computational demand.

Training Configurations. We utilized the ViT-B [9] architecture for classification tasks and YOLOv11 [27] for object detection tasks as our target model. The specific training hyperparameters for each dataset are summarized as follows:

- Imagenette: The ViT-b was trained for 25k iterations with batch size of 128 and a learning rate of 1×10^{-4}. The initial validation patience threshold γ_0 was set to 10.
- ImageNet-100: We trained the Vit-b for 50k iterations with a learning rate of 1×10^{-3}. Due to the faster convergence observed on this dataset, the initial validation patience threshold γ_0 was set to 6.
- MS-COCO: The YOLOv11 was trained for 50k iterations with batch size of 64 and learning rate of 1×10^{-2}. The initial validation patience threshold γ_0 was 4.

Data Generation Configurations. We use 50 inference steps using Euler ancestral scheduler. For the input prompts to generate the synthetic data, we use the its class name only. As we mentioned, we adapt FKS, with the interval steps {10, 20, 30, 40, 49} to conduct the resampling. We use the number of particles $S = 4$ as balanced configuration; however, this can be increased to further enhance performance at the cost of inference speed. We set the FKS guidance value λ as 16 on Imagenette and MS-COCO, and 24 on Imagenet-100, respectively.

Optimization and Scheduling. On Imagenette and ImageNet-100 settings, we adopted the adaptive validation patience strategy described in Section B.3. For learning rate management, we employed the WSD (Warmup-Stable-Decay) scheduler [58], allocating 10% of the total iterations for the warmup stage and 20% for the final decay stage. To enhance regularization, we applied label smoothing [50] with a factor of 0.11. We use AdamW [36] to optimize the target model during training phase. On MS-COCO, we use nesterov SGD [49] with 0.937 of momentum and 0.0005 of the weight decay.

Data Augmentation. For classification tasks, we applied a consistent set of augmentations across all experiments, including random resized cropping ($p = 0.5$) and random horizontal flipping ($p = 0.5$). For the ImageNet-100 dataset, we further applied the augmentation by incorporating Mixup ($p = 0.5$) and CutMix ($p = 1.0$). For object detection tasks, we do not apply additional data augmentations.

Laplace Prior Optimization. In order to calculate the BALD score contained in GALBA properly, finding optimized Gaussian prior precision τ for the Laplace approximation with respect to the last layer parameter is necessary. we optimized τ via SGD with a learning rate of 1×10^{-5}.

Baseline Implementation (DP). Due to the unavailability of the original training code for the Deliberate Practice (DP) method, we re-implemented the DP guidance mechanism by referencing the following repository provided by the authors: https://github.com/facebookresearch/Feedback-guided-Data-Synthesis.

C More Experiments

C.1 Additional Generative Oracle

C.2 Comparison with Diffusion-based Augmentation Baselines

To further evaluate the generality and competitive advantage of GALBA, we benchmark our framework against representative diffusion-based data augmentation methodologies: ActGen [25], DA-Fusion [52], and ALIA [11] . While these existing paradigms primarily aim to augment existing real-world images, GALBA is fundamentally designed to generate maximally informative data from scratch via active epistemic steering. For a rigorous and fair evaluation, we assess all methods under two distinct operational regimes: (1) settings with partial access to real training data (following the setup in Section 5.3), and (2) fully synthetic regimes with zero access to real data (following Section 5.2).

C.2.1 Evaluation with access to real data

In this setting, the downstream model is trained using a combination of a restricted real-world dataset and synthetic data generated by each respective method.

Table 3: Downstream Top-1 accuracy comparison on Imagenette with access to real data.

Method	No guidance	ActGen	DA-fusion	ALIA	GALBA
Accuracy	78.5	78.9	79.2	79.0	**80.3**

As shown in Table 3, existing diffusion augmentation methods offer modest gains over unguided generation. However, GALBA consistently outperforms all baselines. This demonstrates that actively targeting the classifier's epistemic knowledge gaps supplies more synergistic and complementary inductive bias than conventional semantic perturbations of real images.

C.2.2 Evaluation in Fully Synthetic Regimes

A crucial objective of Generative Active Learning (GAL) is constructing an informative dataset from scratch when domain-specific real training dataset are scarce or entirely inaccessible. To evaluate baseline adaptability in this regime, we configure ActGen, DA-Fusion, and ALIA to augment purely synthetic seed samples rather than real images.

Table 4: Downstream Top-1 accuracy comparison on Imagenette under fully synthetic regimes (without real data).

Method	No guidance	ActGen	DA-fusion	ALIA	GALBA
Accuracy	70.8	62.5	64.9	64.6	**74.2**

The results are depicted in Table 4. Under this setting, the empirical behavior of standard augmentation methods diverges drastically. Existing baselines experience significant performance drops, falling well below the unguided baseline. Because these methods implicitly condition on the fidelity and manifold consistency of real instances, applying iterative transformations to synthetic images compounds generative noise, visual artifacts, and unnatural deformations.

Conversely, GALBA does not rely on real-world reference anchors. By steering the diffusion trajectories directly toward high epistemic uncertainty via FKS and our last-layer Laplace BALD objective, GALBA synthesizes structurally sound yet decision-challenging examples from scratch.

These findings confirm that while existing diffusion augmentation pipelines are confined to perturbing existing real data distributions, GALBA provides a principled and self-contained framework capable of generating high-utility training distributions independently.

C.3 Comparison with Post-Hoc Filtering Baselines

A natural alternative to online trajectory steering is a post-hoc filtering paradigm, where one synthesizes an oversized candidate pool via standard unguided diffusion and subsequently filters samples using an acquisition criterion. To investigate whether active sampling guidance during diffusion is necessary, or if the observed benefits stem merely from ranking and filtering synthetic samples after generation, we evaluate this decoupled strategy across multiple acquisition metrics.

Following the Imagenette benchmark setup described in Section 5.2, at each acquisition cycle we generate an unguided candidate pool of size $4 \times N$ (6,400 samples, four times the cycle acquisition budget $N = 1{,}600$). We then score all candidates under the current target model using representative utility criteria:

- Least Confidence: Prioritizes samples with the lowest maximum predicted class probability.
- Entropy: Ranks samples by single-model entropy $\mathcal{H}[y|x]$.
- BALD: Selects samples maximizing our Last-Layer Laplace BALD objective $\mathcal{I}[y, W|x, \mathcal{D}]$.

The top 1,600 highest-utility samples from each criterion are retained and appended to the downstream training set $\mathcal{D}_{\text{train}}$, discarding the remaining 4,800 candidates. All procedures are matched to GALBA in total training steps, batch sizes, and data acquisition budgets.

Table 5: Comparison between post-hoc filtering and GALBA on Imagenette.

Methods	No guidance	Least confidence	Entropy	BALD	GALBA
Accuracy	70.8	73.0	72.9	73.1	**74.2**

The results are depiced in Table 5. Post-hoc filtering with uncertainty metrics consistently outperforms standard unguided generation, confirming that selecting challenging instances enhances downstream training utility over uniform sampling. However, despite searching over a significantly larger candidate pool ($4\times$ the target size), all generate-then-select variants underperform GALBA by a clear margin.

This gap arises because unguided generation draws samples strictly according to the base generator prior $p_0(x_0)$, where highly informative, boundary-adjacent instances reside in the extreme tails of the distribution. Passive filtering is fundamentally bounded by the stochastic draw of this prior; it cannot create informative samples if none were generated in the candidate batch.

In contrast, GALBA actively modulates the intermediate reverse diffusion path using FKS. By resampling partial trajectories according to BALD potentials throughout the denoising process, FKS tilts the sampling distribution directly toward high-epistemic-uncertainty modes that would otherwise have near-zero probability under unguided sampling. Furthermore, because FKS steers trajectories via population-based importance weights rather than direct gradient manipulation, it preserves structural diversity without introducing artifacts.

These results validate that online steering during the diffusion process is fundamentally more effective and data-efficient than decoupled post-hoc filtering.

C.4 Robustness Across Diffusion Generative Backbones

To evaluate whether the effectiveness of GALBA is tied to a specific generative architecture, we investigate the sensitivity of our framework to the underlying generative backbone. While our primary classification experiments employ Stable Diffusion v1.5 (SD v1.5), we conduct an additional ablation study on Imagenette using Stable Diffusion v2.1 (SD v2.1) as the base generative model. All downstream training pipelines, active learning cycle budgets, and target model configurations (ViT-B) strictly follow the protocol outlined in Section 5.2.

The results are depicted in Fig. 6. Under the SD v2.1 backbone, the entropy-driven gradient baseline (DP) yields a negative improvement relative to unguided generation. We presume that the entropy guidance exacerbates high-frequency noise and synthesis artifacts, demonstrating sensitivity to generator-specific failure modes.

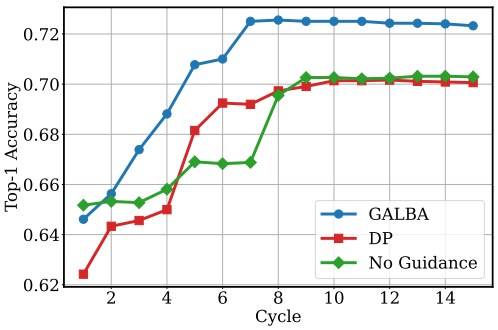


Figure 6: Top-1 accuracy progression on Imagenette synthesized by Stable Diffusion v2.1

In contrast, GALBA consistently achieves substantial downstream gains across both architectures, outperforming the unguided baseline by +2.19% on SD v2.1. By concentrating on epistemic uncertainty than aleatoric synthesis artifacts via the Last-Layer Laplace BALD objective and relying on derivative-free particle steering (FKS), GALBA steers trajectories reliably regardless of the underlying diffusion architecture. These findings confirm that GALBA is not narrowly tuned to a specific generator, demonstrating robust applicability across varying diffusion model families.

C.5 Rigorous Disentanglement of Epistemic and Aleatoric Uncertainty

To evaluate whether guided generation disproportionately amplifies irreducible aleatoric ambiguity (e.g., synthesis noise, blur, or malformed artifacts) versus true epistemic informativeness, we monitor both metrics and their relative concentration across all GAL cycles on Imagenette. Table 13 summa-

rizes the cycle-averaged total uncertainty $\mathcal{H}[y \mid x, \mathcal{D}]$, BALD score $\mathcal{I}[y, W \mid x, \mathcal{D}]$, and the epistemic concentration ratio, $\mathcal{I}[y, W \mid x, \mathcal{D}]/\mathcal{H}[y \mid x, \mathcal{D}]$.

Table 6: Quantitative comparison of cycle-averaged predictive uncertainty components on Imagenette.

Method	**Total Predictive Uncertainty** ($\mathcal{H}$)	**BALD** ($\mathcal{I}$)	**Ratio** ($\mathcal{I}/\mathcal{H}$)
No Guidance	0.8582	0.0307	0.0348
Deliberate Practice (DP)	0.9707 (+13.1%)	0.0383 (+24.6%)	0.0374 (+7.5%)
GALBA (Ours)	**0.9976** (+16.2%)	**0.0398** (+29.6%)	**0.0381** (+9.5%)

The results are depicted in Table 6. Although total predictive uncertainty increases alongside guidance, GALBA consistently achieves the highest ratio between BALD and total uncertainty. This quantitative shift demonstrates that the samples generated by GALBA contain a significantly higher proportion of reducible model uncertainty, rather than merely exhibiting high aleatoric/predictive entropy induced by synthesis artifacts.

C.6 Decision Boundary Analysis via Margin Distance

To provide direct quantitative verification of boundary proximity and complement the MPD analysis, we evaluate the margin distance across the synthesized datasets. For a given synthesized instance x and a K-way target classifier producing probabilities $p(y \mid x) = (p_1(x), \ldots, p_K(x))$, let $c_{(1)} = \arg\max_c p_c(x)$ and $c_{(2)} = \arg\max_{c \neq c_{(1)}} p_c(x)$ denote the classes with the highest and second-highest predicted confidence, respectively. The margin distance is defined as $p_{c_{(1)}}(x) - p_{c_{(2)}}(x)$. A smaller margin distance directly indicates that the classifier struggles to separate the primary category from the closest competing alternative, providing an established metric for decision-boundary proximity.

Table 7: Margin distance of synthetic samples with respect to the target classifier.

Method	No Guidance	DP	GALBA (Ours)
Margin Distance $\downarrow$	0.7845	0.7435	**0.7250**

The results are depicted in Table 7. Specifically, GALBA achieves the lowest average margin distance, demonstrating a substantial reduction compared to both unguided diffusion and DP. This confirms that steering via our Last-Layer Laplace BALD objective actively populates the low-confidence regions near the target model's decision boundaries.

C.7 Correlation Analysis among various BALD estimator

Table 8: Comparison of Spearman correlation and overlap metrics between LLLA and MC-dropout. MC-D stands for MC-dropout, and T is the number of samples to calcualte MC-dropout BALD.

Metric	LLLA (ours)	MC-D ($T = 10$)	MC-D ($T = 50$)	MC-D ($T = 100$)
Spearman corr.	0.66	0.64 (-0.02)	0.69 ($+0.03$)	0.69 ($+0.03$)
Top 20% overlap	0.46	0.44 (-0.02)	0.47 ($+0.01$)	0.46 (0.00)
Top 50% overlap	0.72	0.72 (0.00)	0.73 ($+0.01$)	0.74 ($+0.02$)

To further validate the reliability of our uncertainty estimates, we compare our proposed Last-layer Laplace approximated BALD estimator (LLLA) and MC-Dropout against a 5-model Deep Ensemble [31], which is often regarded as a gold standard for uncertainty quantification. As summarized in Table 8, LLLA exhibits a decent Spearman correlation and Top-K% overlap with the Deep Ensemble's rankings, and the score differences between the LLLA and MC-dropout methods are negligible. These results demonstrate that LLLA effectively captures uncertainty profiles comparable to those of computationally expensive deep ensembles and MC-dropout, while maintaining significantly lower overhead. Please refer Section C.9 for detailed analysis of time consumptions.

C.8 Various BALD Evaluation Method for Guiding Diffusion

While Deep Ensembles provide a robust framework for quantifying epistemic uncertainty, their computational cost, requiring the full training of multiple model instances, is often prohibitive in resource-constrained settings. Because our objective is to maximize performance within a fixed computational budget, we prioritize efficient uncertainty estimation over full ensemble-based approaches. Notably, our deterministic approximation produces BALD scores highly comparable to those derived from ensembles, without the necessity of training multiple models.

To evaluate our approach against other practical Bayesian estimators, we compare GALBA with two prominent alternatives: MC-dropout and subspace ensembles. For the MC-dropout baseline, we utilize 100 stochastic forward passes. For the subspace ensemble, we construct a 5 model ensemble by freezing all but the last two layers of the target model, thereby reducing the training burden compared to full deep ensembles.

As illustrated in Fig. 7, neither alternative outperforms GALBA. Despite the significant computational overhead of 100 forward passes, MC-dropout based guidance falls short, an observation that aligns with recent literature suggesting that last-layer approximations can offer superior uncertainty calibration in specific regimes [18]. Furthermore, while subspace ensembles are more efficient than full-parameter ensembles, they still introduce non-trivial computational overhead and structural complexity to the GAL pipeline, such as the need to independently train and manage multiple network branches.

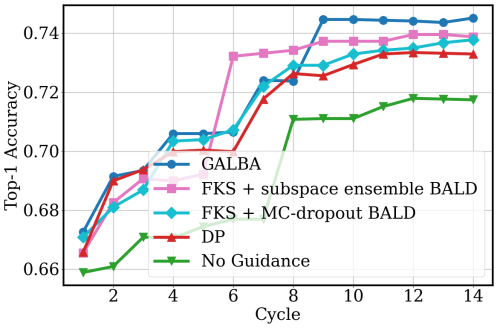


Figure 7: Top-1 accuracy progression on Imagenette for ablation study, comparing MC-dropout and subspace deep ensemble.

These findings confirm that GALBA provides a highly effective solution, making it a more appropriate approach for steering diffusion models toward informative data distributions without relying on expensive Bayesian computations.

C.9 Time and Memory

Table 9: Comparison of precomputation time, filtering time, total generation time, and peak memory across different methods.

Metric	No guidance	DP (Grad + Ent)	FKS + MC-dropout (T=100)	Post-hoc filtering (LLLA BALD)	GALBA (Ours)
Precomputation time (s)	–	–	–	16.56	17.38
Filtering time (s)	–	–	–	2.68	–
Total generation time (s)	**311.43**	1255.09	1652.04	1249.31	1087.44*
Peak memory (GB)	**2.87**	7.81	5.90	3.67	5.94

*Including precomputation time.

To evaluate the practical viability of guidance frameworks, we conduct a comprehensive efficiency analysis benchmarking wall-clock runtime and peak GPU memory footprint across candidate methods. We compare our approach, GALBA, against unguided generation (No Guidance) and DP, alongside two representative baselines: (1) **FKS + MC-dropout**, which evaluates BALD scores via $T = 100$ stochastic forward passes, serving as a non-deterministic benchmark against our closed-form approximation (Section C.8); and (2) **Post-hoc filtering with LLLA-BALD**, which generates an unguided candidate pool and subsequently filters out samples with low informativeness scores (Section C.3). All benchmarks are conducted on a single NVIDIA RTX 3090 GPU with a batch size of 1 to synthesize 100 samples.

As summarized in Table 9, GALBA requires a one-time pre-computation step to construct class-wise Hessian matrices by iterating over the training dataset. Despite this initial overhead, the results show that GALBA is the most efficient among all guided generation methods. Compared to DP, GALBA achieves a 13% reduction in total generation time and a 24% reduction in peak GPU memory usage

during inference. Compared to the MC-dropout baseline, GALBA reduces total time consumption by approximately 34% while maintaining a comparable memory consumption.

In addition, compared to the post-hoc baseline, GALBA achieves about 13% reduction in total generation time. We presume that, in GALBA, intermediate guidance and BALD evaluations occur entirely in GPU VRAM as accelerated tensor operations. In contrast, the post-hoc method generates 4 times candidate images that incur substantial storage disk read/write overhead during the candidate saving, loading, and filtering stages, creating a practical system bottleneck that scales worse as data volume increases.

In practical scenarios, samples are synthesized in parallel using larger batch size to optimize throughput. To demonstrate the scalability of GALBA compared to gradient-based frameworks, we measured the wall-clock time required to synthesize 100 data instances across varying batch sizes. As shown in Table 10, the DP method suffers from out-of-memory constraints even at a modest batch size of 4. This is because gradient-based guidance necessitates backpropagation through the target model at every inference step, which significantly increases the memory usage due to the storage of intermediate activations. In contrast, by utilizing the target model only for forward-pass evaluations to compute rewards, GALBA significantly reduces the peak memory. This efficiency allows GALBA to support larger batch sizes and achieve faster total wall-clock time for data synthesis, highlighting practicality of our method with constrained computing resources.

Table 10: Comparison of wall-clock time and memory efficiency.

Methods	Batch	Time (s)
No Guidance	2	250.21
	10	212.08
DP	2	1136.58
	4	OOM
GALBA (ours)	2	<u>1013.89</u>
	10	<u>967.09</u>

C.10 Consistency Analysis Between Data and Label

Table 11: Performance comparison on 1,600 synthetic images. Oracle results (left) confirm semantic preservation, while target accuracy (right) indicates sample difficulty.

Metric	No Guidance	DP	GALBA
Top-1 Acc. (%)	76.56	74.88	74.31
Top-5 Acc. (%)	99.25	98.88	98.94

(a) Oracle (EVA-02 Large) performance.

Metric	No Guidance	DP	GALBA
Accuracy (%)	95.81	92.38	88.31

(b) Target model accuracy.

To verify that GALBA consistently aligns between the synthetic data and the labels given to the diffusion models, we conduct a dual-validation analysis. We assess both the semantic fidelity of the generated samples and their actual difficulty for the learner by comparing the predictions of an oracle model with those of the target model.

First, we evaluate the semantic quality of 1,600 synthesized samples using a high-performance oracle classifier, EVA-02 Large [14] pretrained on ImageNet-1K. As shown in Table 11a, while guided methods (DP and GALBA) exhibit a minor decrease in Top-1 accuracy compared to unguided generation, their Top-5 accuracy remains remarkably high ($\sim$ 99%). This near-perfect Top-5 performance indicates that the generated images successfully preserve the core semantics of the target classes. We attribute the slight reduction in Top-1 accuracy to increased scene complexity (e.g., placing a "dog" on a "sofa"), which introduces benign ambiguity rather than semantic noise.

Second, we evaluate the target model (from the previous cycle) on these identical samples. As reported in Table 11b, the target model's accuracy on GALBA-generated data drops significantly to 88.31%, compared to 95.81% for unguided samples.

The contrast between the robust recognition by the oracle model and the degraded performance of the target model provides strong evidence that GALBA synthesizes samples that are semantically valid yet strategically challenging for the current learner. This confirms that the perceived difficulty

of GALBA samples is not an artifact of synthesis errors, but a direct result of effective epistemic targeting.

C.11 Efficacy of GALBA over Strong Real Data Baselines

Table 12: Performance gains when augmenting a strong real-data baseline (ViT-B trained on 50% real train data of Imagenette) with 4,800 synthetic samples.

Method	Accuracy (%)	Improvement (Δ)
Real data only (Baseline)	78.23	-
+ No guidance	79.79	+1.56
+ DP	79.30	+1.07
+ GALBA (Ours)	**81.77**	**+3.54**

To investigate the scalability and effectiveness of GALBA in scenarios where the target model has already been exposed to substantial amounts of real-world data, we evaluate our method over a strong baseline. Specifically, we pre-train a ViT-B model on 4,731 real images (representing 50% of the Imagenette dataset) and subsequently augment the training set with 4,800 synthetic samples generated by various methods.

The results, presented in Table 12, demonstrate that GALBA consistently outperforms other generation strategies. Notably, GALBA achieves a significant performance gain of +3.54%, which is more than double the improvement obtained from unguided generation (+1.56%). In contrast, DP yields a marginal gain of +1.07%, underperforming even the unguided baseline.

We attribute this behavior to the distinct handling of uncertainty. As a model is trained on extensive real data, its epistemic uncertainty naturally decreases. Consequently, entropy-based guidance mechanisms, such as DP, become overly sensitive to aleatoric uncertainty This often leads the diffusion model to prioritize the synthesis of artifacts or noisy patterns rather than semantically informative features. By leveraging the BALD estimator, GALBA explicitly isolates epistemic uncertainty, ensuring that the guidance process robustly targets regions where the model lacks knowledge, rather than exploiting synthesis errors.

Table 13: Ablation studies on Imagenette using FKS.

S	2	4	6	8
Top-1 Acc (%)	71.7	73.1	**73.7**	**73.7**

(a) Ablation on particles (S)

λ	8.0	16.0	24.0	32.0
Top-1 Acc (%)	72.8	72.7	**72.9**	72.4

(b) Ablation on guidance scale (λ)

C.12 Ablation studies for the Number of Particles

Prior research indicates that the ability of FKS-based diffusion inference to achieve high rewards is strongly dependent on the number of particles utilized during the inference stage [45]. Generally, a larger particle ensemble during steering allows for a more effective exploration of the state space, leading to higher rewards in the diffusion output. To verify whether our GAL framework exhibits a similar trend, we conducted an ablation study on the particle count for FKS steering using the Imagenette dataset.

The experimental setup followed the primary Imagenette configuration, with the exception that the simulations were executed in a TPU environment. The results in Table 13a indicating the saturated Top-1 accuracy after all GAL cycles are done, demonstrate a clear positive correlation between the number of particles and the target model's Top-1 accuracy. Notably, the performance improves significantly as the particle count increases from 2 to 6, reaching a plateau at 8. This confirms that our GAL framework effectively leverages increased particle counts to generate more informative synthetic samples, consistent with the theoretical behavior of FKS.

C.13 Ablation Studies for FKS Guidance Scale

The guidance scale λ is a critical hyperparameter in the FKS framework, as it modulates the selection pressure during the resampling process. Formally, λ scales the reward values to define the potential landscape in Eq. (13). A higher value of λ amplifies the relative potential differences between particles; specifically, it increases the probability that particles with higher rewards will be selected during the weighted resampling stage. Thus, λ effectively controls the trade-off between adhering to the prior diffusion distribution and steering toward the guidance objective. To evaluate the sensitivity of our framework to this parameter, we conducted a series of experiments on the Imagenette dataset. The experimental configuration remained consistent with our primary setup, with simulations performed in a TPU-based environment to handle parallel particle trajectories efficiently.

The results are summarized in Table 13b. We observe that the target model's performance remains relatively robust within the range of $\lambda = 8.0$ to 24.0, with the peak accuracy of 72.87% achieved at $\lambda = 24.0$. However, a noticeable decline in Top-1 accuracy occurs when the scale is increased to 32.0 (72.4%). This performance drop suggests that an excessively high guidance scale leads to over-steering, where the resampling process becomes too concentrated on a small subset of high-reward particles. This phenomenon, often referred to as particle degeneracy, reduces the structural diversity of the synthesized dataset, thereby providing the target model with redundant information and hindering its ability to generalize. These findings indicate that a moderate guidance scale is optimal for maintaining the necessary diversity while ensuring the "hardness" of the generated samples.

D Broader Impacts

This research introduces GALBA, a framework that enhances generative active learning efficiency by reducing the total volume of samples required to reach model saturation. By minimizing the data acquisition overhead, GALBA optimizes energy consumption and mitigates the environmental footprint associated with training large-scale deep learning models. Despite these efficiency gains, ethical challenges persist, as the framework may inherit or amplify societal biases embedded within underlying pretrained generative models like Stable Diffusion. Therefore, the use of GALBA-generated synthetic datasets necessitates rigorous auditing and bias-mitigation strategies to ensure fair and responsible AI deployment.

E Examples of generated data

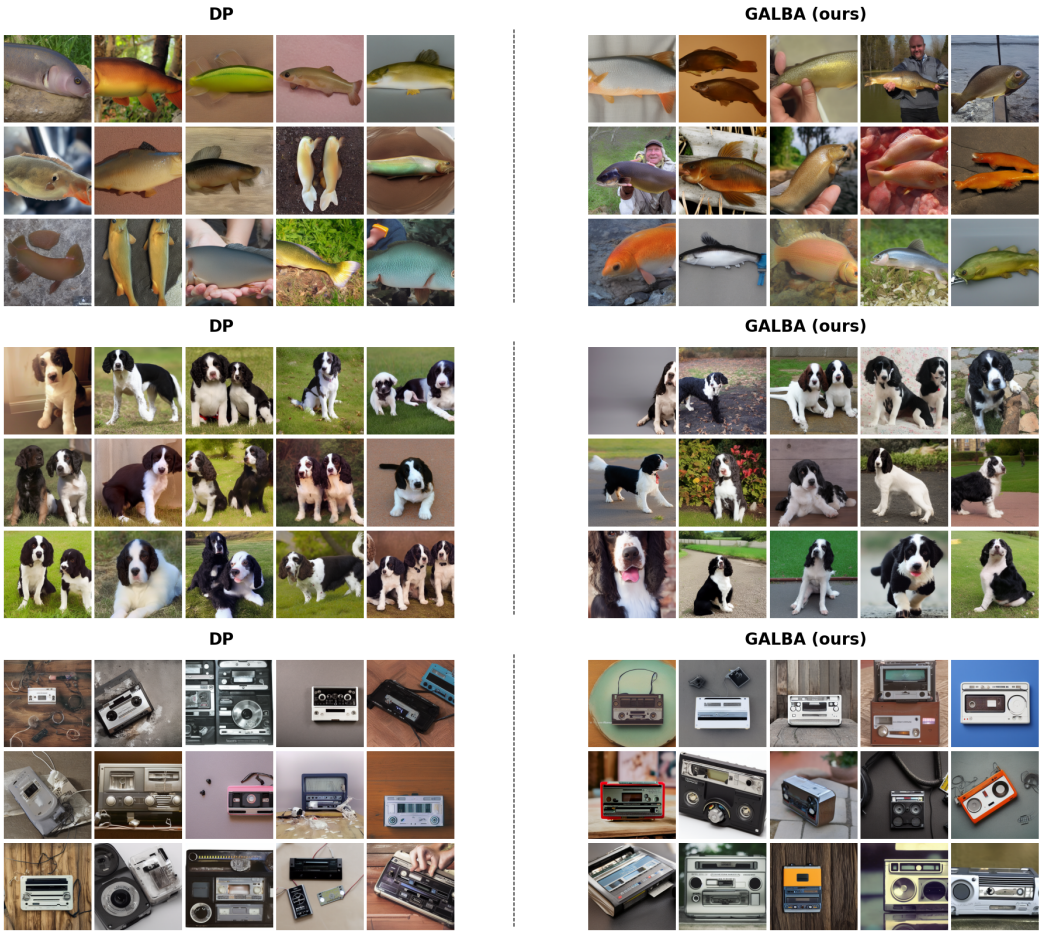


Figure 8: Comparison of examples of synthetic data between DP and ours on Imagenette. (Top) tench, (Middle) English springer, (Bottom) casette player.

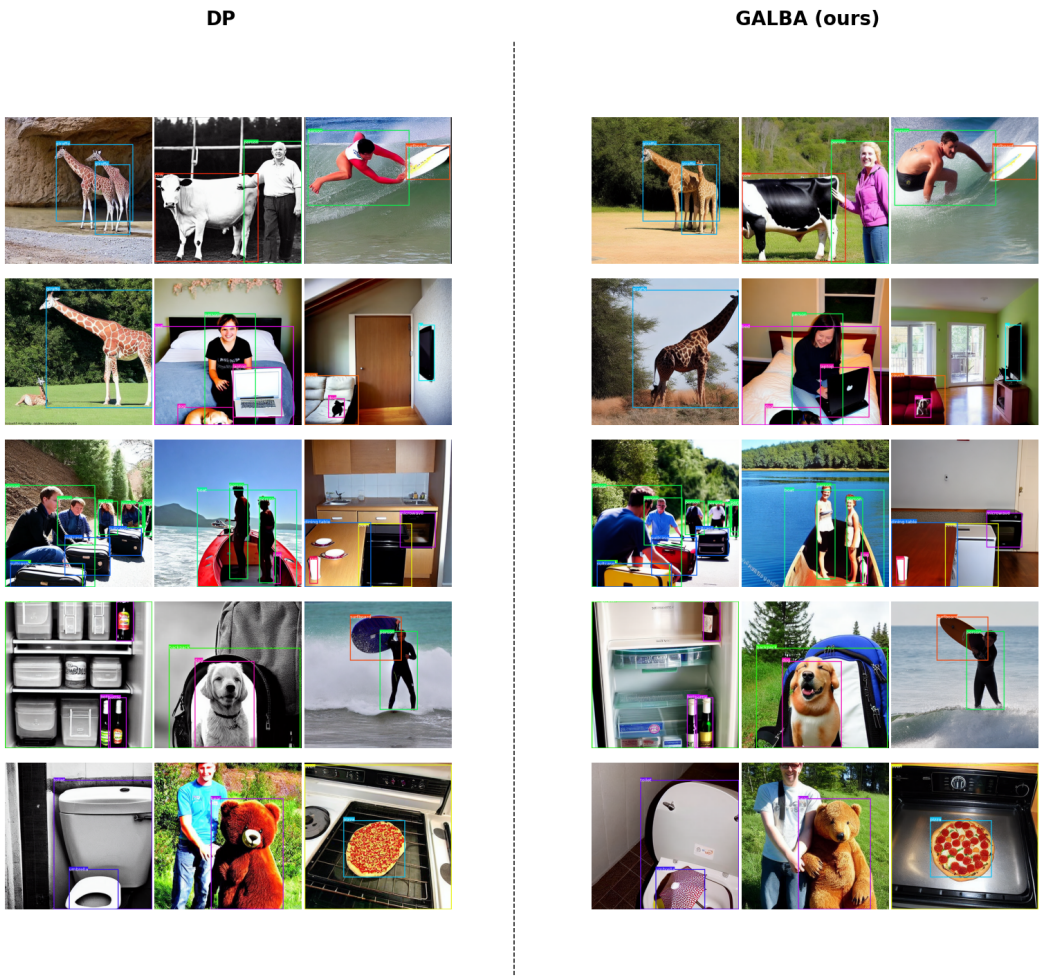


Figure 9: Comparison of examples of synthetic data between DP and ours on MS-COCO.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our paper includes the contribution and scopes in the abstract and the introduction well.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We state the limitations of our research in conclusion section.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Our theoretical results, depicted in Section 4.1, is explained by the derivation in Section A.1.2 of the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In this paper, we provide the implementation details in Section 5 and Section B.6.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide the experiment code on the supplemental materials.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental details, including hyperparameters, optimizer, are provided in Section 5 and Section B.6.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We conduct the experiments with 3 different seeds to calculate standard derivation on Imagenette for main claims Fig. 2. Because of computational constraints, error bars were only measured for the small-scale dataset experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We state our compute resources for conducted experiments in Section B.6.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research in the paper satisfies the NeurIPS Code of Ethis.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We state the broader impacts of our works, including potential positive and negative social impacts, in Section D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: Our research focus on the guidance of diffusion model for enhancing the training utility of the generated data with respect to the target model. As our research does not release new pretrained weight of generative model or new datasets, we state that our research has no risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cited of the dataset and the code we used in Section 5 and Section B.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide asset, including descriptions of the experiments, and codes, in Section 5 and supplemental materials.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our research does not include crowdsourcing experiments nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subject.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We only use LLMs for editing and polishing the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.