SCALING BAYESIAN EXPERIMENTAL DESIGN TO HIGH-DIMENSIONS WITH INFORMATION-GUIDED DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

We present `DiffBED`, a Bayesian experimental design (BED) approach that scales to problems with high-dimensional design spaces. Our key insight is that current BED approaches typically cannot be scaled to real high–dimensional design problems because of the need to specify a likelihood model that remains accurate throughout the design space. We show that without this, their design optimisation procedures exploit deficiencies in the likelihood and produce implausible designs. We overcome this issue by introducing a generative prior over feasible designs using a diffusion model. By guiding this diffusion model using principled information-theoretic experimental design objectives, we are then able to generate highly informative yet realistic designs at an unprecedented scale: while previous applications of BED have been restricted to design spaces with a handful of dimensions, we show that `DiffBED` can successful scale to designing high–resolution images.

1 INTRODUCTION

Experimentation, the process by which we gather information about a phenomenon of interest, is a central task throughout science and industry. In scenarios where data collection is costly or time-consuming, such as drug discovery (Paul et al., 2010; DiMasi et al., 2016) or clinical trials (Fogel, 2018), it is natural to seek designs that yield data that is *maximally informative*. This intuition is captured by the framework of Bayesian experimental design (BED) (D. V. Lindley, 1956; Chaloner & Verdinelli, 1995; Rainforth et al., 2024; Huan et al., 2024). In BED, we specify a probabilistic model of the data gathering process, use this to derive a formal notion of the *expected information gain* (EIG) of an experiment for a target quantity of interest, then optimise this EIG to yield designs we expect to maximally reduce our uncertainty. Thanks to the coherence of Bayesian reasoning, this framework is naturally suited to adaptively gathering information across several experimental steps, utilising information from previous experiments in each sequential decision we make.

Although, in principle, BED can be applied to a wide array of tasks, successful applications have historically been limited to simple problems in which the design variables are low-dimensional (Myung et al., 2013; Vincent & Rainforth, 2017; Watson, 2017; Dushenko et al., 2020; Loredo, 2004; Vanlier et al., 2012; Shababo et al., 2013; Papadimitriou, 2004). Developing methods for high-dimensional design spaces is thus a critical open challenge (Rainforth et al., 2024; Huan et al., 2024), with the problem historically being considered mostly as one of developing scalable EIG estimators (Foster et al., 2019; Goda et al., 2022; Ao & Li, 2023; Iollo et al., 2025a; Huan et al., 2024).

In this work, we demonstrate the existence of an even more fundamental barrier: being able to specify a likelihood in high dimensions that faithfully reflects the real-world data generation process across the entirety of the design space. In other words, model misspecification becomes increasingly unavoidable as design dimensionality increases, as we must construct a likelihood model that remains accurate over an ever-growing space. In particular, while the success of modern machine learning methods relies on modelling around some data manifold, our desire to optimise with respect to the design and seek out information that is distinct from that already known inevitably relies on the ability of our likelihood to "extrapolate" to regions when our ability to predict outcomes is limited.

The upshot of this, as shown in Figure 1, is that directly optimising the EIG leads to flaws in the likelihood being exploited, and meaningless designs being produced. This is akin to reward hacking in reinforcement learning (Skalse et al., 2022). Namely, we see that even though existing stochastic gradient approaches are already effective at optimising the EIG in this very high dimensional setup,

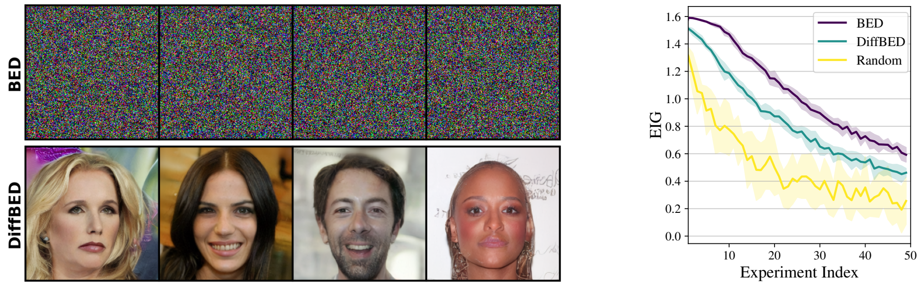


Figure 1: First iteration design sets for the *search* experiment (see Section 6) for BED (top left) and via `DiffBED` (bottom left). Also shown is incremental EIG achieved at *each* experiment iteration (right).

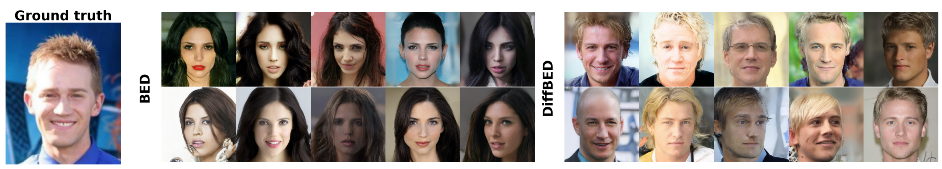


Figure 2: Posterior samples after 50 search iterations for standard BED (left) and `DiffBED` (right).

the problem instead is that the optimisation leads to unrealistic designs for which the assumed likelihood is heavily misaligned with the true data-generating process. In particular, as we show later, it seeks out designs where the model's likelihood is overconfident and the experiment outcomes will, in truth, be uninformative. Moreover, in Figure 2, we see that when sequentially applying BED to adaptively choose designs, this in turn leads to a problematic feedback loop in the posterior updates: rather than simply remaining uncertain, our posterior beliefs collapse around a distinctly incorrect θ.

To address this, we introduce `DiffBED`, a novel method for Bayesian experimental design in high-dimensional design spaces. `DiffBED` works by introducing a *prior over feasible designs*. It reframes the design optimisation as sampling from a distribution that balances prior feasibility and EIG under the model. This ensures that the designs generated stay on an admissible manifold where the likelihood is relatively well aligned, and regularises against such reward hacking behaviour.

Specifically, `DiffBED` uses a diffusion model for its design prior. Designs are then generated by a process we call *information-guided diffusion*, where designs are chosen by simulating the reverse–time SDE of this diffusion process, with guidance provided by an estimator for the score of the EIG. This estimator is itself based on a combination of Tweedie's formula (Robbins, 1956) and existing EIG gradient estimators (Rainforth et al., 2018; Foster et al., 2020). Adaptive design is performed by rerunning the diffusion process with updated EIG estimators that incorporate new observations.

As shown in Figure 1, this leads to designs that are both meaningful and informative. In turn, these designs enable effective learning about the target quantity of interest (see Figure 2). `DiffBED` therefore represents the first successful application of BED in high-dimensional design spaces: we show successful deployment of `DiffBED` to design spaces in excess of 750 000 dimensions, whereas previous BED approaches have rarely been successfully been used beyond ~ 20 dimensions.

2 Preliminaries

We begin by reviewing the key concepts underpinning the BED framework. In BED, we express our initial beliefs about the target variable of interest, θ, through a prior distribution $p(\theta)$. We also specify a likelihood $p(y \mid \theta, \xi)$, which gives the probability of possible experimental outcomes y given θ and a design ξ. If an outcome y were observed by running an experiment with design ξ, the *information gain* (IG) of such an experiment is the reduction in entropy obtained when updating from the prior $p(\theta)$ to the posterior $p(\theta \mid y, \xi)$ (Lindley, 1956), defined as, $\mathrm{IG}(y, \xi) = \mathrm{H}[p(\theta)] - \mathrm{H}[p(\theta \mid y, \xi)]$. Since the outcome y is unknown before running the experiment, we instead maximise the *expected information gain* (EIG) (Lindley, 1972; Bernardo, 1979; Sebastiani & Wynn, 2000):

$$\mathrm{EIG}(\xi) = \mathbb{E}_{p(y|\xi)}[\mathrm{IG}(y, \xi)] = \mathbb{E}_{p(\theta)\, p(y|\theta,\xi)} \left[\log p(\theta \mid y, \xi) - \log p(\theta)\right] \tag{1}$$

$$= \mathbb{E}_{p(\theta)\, p(y|\theta,\xi)} \left[\log p(y \mid \theta, \xi) - \log p(y \mid \xi)\right], \tag{2}$$

where the last equality follows from Bayes' rule. The EIG is then maximized to produce an optimal design $\xi^* = \text{argmax}_{\xi\in\Xi} \text{EIG}(\xi)$, where Ξ is the space of admissible designs.

Adaptive Design In many applications, we are interested in producing a sequence of designs $\xi_1, \xi_2, \ldots, \xi_K$ yielding data $y_1, y_2, \ldots, y_K$. While the sequence $\xi = (\xi_1, \xi_2, \ldots, \xi_K)$ could be determined statically before observing any data, a more performant approach is to select the design ξ_k adaptively depending on the history $\mathcal{D}_{k-1} = \{(\xi_i, y_i)\}_{i=1}^{k-1}$ of designs and outcomes prior to step k. In this adaptive setting, the design for the k-th experiment is typically obtained by greedily maximising the *incremental* $\text{EIG}(\xi_k \mid \mathcal{D}_{k-1}) = \mathbb{E}_{p(\theta|\mathcal{D}_{k-1})\, p(y_k|\theta,\xi_k)} \left[\log p(y_k \mid \theta, \xi_k) - \log p(y_k \mid \xi_k)\right]$. This expression is equivalent to the EIG except that the prior, $p(\theta)$, is replaced with the posterior, $p(\theta|\mathcal{D}_{k-1})$, which reflects the current beliefs given the history (Rainforth et al., 2024).

Estimating the EIG While the EIG is conceptually appealing, estimating it can be challenging due to the doubly intractable nature of the objective, with a wide variety of approaches proposed to address this, see (Rainforth et al., 2024, Section 3) for a review. However, when the outcome space $\mathcal{Y}$ is discrete, the outer expectation with respect to the likelihood can be enumerated over. In this case, the EIG is now singly-intractable, yielding a non-nested estimator (Rainforth, 2017; Gal et al., 2017)

$$\widehat{\text{EIG}}(\xi) = -\sum\nolimits_{y\in\mathcal{Y}} \hat{p}(y \mid \xi) \log \hat{p}(y \mid \xi) + \frac{1}{N}\sum\nolimits_{n=1}^{N}\sum\nolimits_{y\in\mathcal{Y}} p(y \mid \theta_n, \xi) \log p(y \mid \theta_n, \xi), \quad (3)$$

where $\theta_n \sim p(\theta)$ (or $\theta_n \sim p(\theta \mid \mathcal{D}_{k-1})$ in adaptive settings) and $\hat{p}(y \mid \xi) = \frac{1}{N}\sum_{n=1}^{N} \hat{p}(y \mid \theta_n, \xi)$.

Optimizing the EIG with Stochastic Gradients During active experimentation, we not only want to estimate the EIG, but optimize it. Stochastic gradient-based methods are scalable and effective for optimization in high-dimensional continuous design spaces and can easily be applied to estimates of the gradients of the EIG such as (3) (Huan & Marzouk, 2014; Foster et al., 2020; Goda et al., 2022).

3 Directly Optimising EIG Seeks Out Model Misspecification

Bayesian experimental design is inherently model-based, with the EIG relying on the assumed likelihood model $p(y \mid \theta, \xi)$ and a subjective prior $p(\theta)$. As such, how well the EIG reflects the true expected utility of gathering new data will depend on the accuracy of this model, in particular how well the assumed likelihood approximates the true conditional data-generating process $p_{\text{true}}(y \mid \theta, \xi)$. While the prior provides some protection against needing the likelihood to be accurate across all θ, optimising over ξ requires the likelihood to remain accurate across the entire design space.

When the designs ξ are high-dimensional, faithfully modelling $y \mid \theta, \xi$ becomes especially challenging and it is usually not realistic to construct a likelihood that is accurate across all (θ, ξ) pairs. Indeed, the assumed likelihood $p(y|\theta, \xi)$ in high-dimensional problems will often itself be a *learned* function derived from a pre-trained machine learning model. For example, in Figure 1 our likelihood utilises a fixed encoder that captures semantic content. In such cases the likelihood will only reflect the true data-generating mechanism in data regions near where the feature extracting component was trained.

We now show that direct optimisation of the EIG is *inherently prone* to seeking out areas of the design space where the model is misspecified, specifically, it is drawn to regions where the likelihood is *overconfident*. To do this we consider the difference in using the EIG with our model's likelihood compared with a "true" EIG that uses the unknown true underlying data distribution:

$$\begin{aligned} \text{TEIG}(\xi) :&= \mathbb{E}_{p(\theta)p_{\text{true}}(y|\theta,\xi)}\left[\log p_{\text{true}}(y \mid \theta, \xi) - \log p_{\text{true}}(y \mid \xi)\right] && (4)\\ &= -\mathbb{E}_{p(\theta)}[\text{H}[p_{\text{true}}(y \mid \theta, \xi)] + \text{H}[p_{\text{true}}(y \mid \xi)] && (5)\end{aligned}$$

where $p_{\text{true}}(y \mid \xi) = \mathbb{E}_{p(\theta)}[p_{\text{true}}(y \mid \theta, \xi)]$. Equation (5) and the analogous term for $\text{EIG}(\xi)$ yields

$$\text{EIG}(\xi) = \text{TEIG}(\xi) + \underbrace{\mathbb{E}_{p(\theta)}\left[\text{H}[p_{\text{true}}(y \mid \theta, \xi)] - \text{H}[p(y \mid \theta, \xi)]\right]}_{=:\mathcal{M}(\xi)} + \text{H}[p(y \mid \xi)] - \text{H}[p_{\text{true}}(y \mid \xi)], \quad (6)$$

This decomposition provides helpful insight into how the EIG behaves when used with an approximate model likelihood. Namely, we can view $\mathcal{M}(\xi)$ as a measure on the average degree of model overconfidence across possible θ: it is zero if the likelihood matches the true data generating process and it grows as the likelihood becomes more certain than it should be. Critically, $\mathcal{M}(\xi)$ varies across designs, and its presence in the decomposition encourages designs found by directly optimizing the EIG to lie where the likelihood is overconfident.

Moreover, the remaining $\mathrm{H}[p(y \mid \xi)] - \mathrm{H}[p_{\text{true}}(y \mid \xi)]$ term typically provides little protection against this desire to move to regions of overconfident likelihoods. In particular, these marginal data distributions will inherently be more diffuse than the corresponding likelihoods, with the averaging over θ providing regularisation on their predictions. Thus, in high-dimensional spaces it will usually be easy to find designs where we are overconfident in $y|\theta, \xi$, but our uncertainty over θ ensures that $\mathrm{H}[p(y \mid \xi)]$ remains high. Thus, even when the likelihood is heavily misspecified, $p(y \mid \xi)$ and $p_{\text{true}}(y \mid \xi)$ will often still be similar for most ξ. For example, if we have a design very far away from previous designs then a misspecified (but still sensible) model will generally produce high marginal predictive uncertainty, even though it might be very confident about y when θ is known. As such, the signal from this remaining term will generally not sufficiently counteract $\mathcal{M}(\xi)$ and we can expect direct EIG optimisation to seek out designs where the likelihood is overconfident.

4 Bayesian Experimental Design via Information-Guided Diffusion

While improving the fidelity of $p(y \mid \theta, \xi)$ would reduce the misalignment in (6), simply modifying the likelihood is not a viable solution , since any residual imperfections will be exploited by the optimizer in high-dimensional spaces. Rather, we accept inevitable misalignment and instead modify the design process itself. Ideally, we would like to maximize the EIG subject to $\mathcal{M}(\xi)$ being small, but this misalignment, by definition, is often difficult, if not impossible, to quantify.

Instead, we introduce a reference prior on designs, $p^{\text{ref}}(\xi)$, that captures the manifold of feasible designs. We can then restrict our search to only those designs which have reasonable support under this reference prior. For many problems, suitable reference priors can be constructed from unlabelled auxiliary data or a separate foundation model, without requiring any task-specific data. For example, if our design is an image of a face, then $p^{\text{ref}}(\xi)$ could be instantiated as a deep generative model over faces. The manifold defined by $p^{\text{ref}}(\xi)$ also often aligns with regions where likelihood misalignment is relatively small. In particular, as we demonstrate later, it is often possible to derive the reference prior from the same data used in constructing the likelihood. For example, in Section 6 we use likelihoods that depend on an unsupervised encoder of images. Constraining design optimization to a manifold that is meaningful avoids the catastrophic collapse to incorrect θs seen in Figure 2 for traditional BED.

We now need a mechanism to produce designs that both representative samples under $p^{\text{ref}}(\xi)$ and which we expect to be highly informative under our model, i.e. that have high $\mathrm{EIG}(\xi)$. To do this, we consider the following optimisation problem over $q(\xi) \in \mathbb{P}(\Xi)$,

$$p^*(\xi) = \mathrm{argmax}_{q(\xi)\in\mathbb{P}(\Xi)} \ \mathbb{E}_{q(\xi)}[\mathrm{EIG}(\xi)] - \alpha\, \mathrm{KL}\big[q(\xi) \,\|\, p^{\text{ref}}(\xi)\big] \tag{7}$$

where $\alpha > 0$ is a hyperparameter that trades off achieving high EIG values with adherence to the reference distribution, as measured by the KL divergence. Using variational calculus, Equation (7) yields a unique solution known as the *exponential tilting* distribution (Rawlik et al., 2012)

$$p^*(\xi) \propto p^{\text{ref}}(\xi) \cdot \exp\left(\alpha^{-1}\mathrm{EIG}(\xi)\right). \tag{8}$$

Sampling $\xi \sim p^*(\xi)$ now ensures that designs are drawn from high-probability regions of $p^{\text{ref}}(\xi)$ while up-weighting those with large EIG. Notably, this approach requires no likelihood-dependent training of p^{ref} nor any modifications to the likelihood, so that a pre-trained generative model is readily applicable. While Equation (8) could potentially be maximized, we choose to sample from $p^*(\xi)$ to avoid exploiting imperfections in the approximation of the generative model, which may lead to unrealistic designs. In particular, it has been shown that the points to which deep generative models assign the highest density are often not themselves reasonable samples (Nalisnick et al., 2018).

4.1 Guiding Diffusion with EIG

Having established $p^*(\xi)$ as our target distribution for producing designs, we now introduce `DiffBED`, our proposed framework which instantiates this idea using a diffusion model (Ho et al., 2020; Song et al., 2021) as the reference generative model p^{ref} in (8). Diffusion models offer state-of-the-art generative quality and diversity across images, video, and scientific data. Unlike VAEs (Kingma & Welling, 2013) or GANs (Goodfellow et al., 2014), diffusion models learn a score function rather than a fixed decoder. This enables powerful training-free guidance methods for sampling from tilted distributions (Bansal et al., 2023; Uehara et al., 2025; Ye et al., 2024; Domingo-Enrich et al., 2024; Denker et al., 2024), making them uniquely suited to our framework by allowing sampling from (8) without retraining or latent-space optimization.

Diffusion Models Diffusion models define a forward SDE which gradually corrupts data with noise and a learn reverse process which undoes this corruption (Song et al., 2021). The forward SDE is

$$\mathrm{d}\xi_t = f(\xi_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}W_t \qquad \xi_0 \sim p_0(\xi_0) \qquad t \in [0, T] \tag{9}$$

where $p_0(\xi_0)$ is a training distribution of designs with $\xi = \xi_0$, $f(\xi_t, t)$ is a drift vector field, and $g(t)$ is a noise schedule. The functions f, g are chosen so that $p_T(\xi_T)$ is approximately Gaussian. A generative model is obtained by solving the time-reversal of Equation (9), given by

$$\mathrm{d}\xi_t = \left[f(\xi_t, t) - g(t)^2 \nabla_{\xi_t} \log p_t(\xi_t)\right]\,\mathrm{d}t + g(t)\,\mathrm{d}\overleftarrow{W_t} \qquad \xi_T \sim p_T(\xi_T) \qquad t \in [0, T] \tag{10}$$

where $\mathrm{d}\overleftarrow{W_t}$ is a time-reversed Brownian increment and $b(\xi_t, t) := f(\xi_t, t) - g(t)^2 \nabla_{\xi_t} \log p_t(\xi_t)$ is an updated drift. The intractable score $\nabla_{\xi_t} \log p_t(\xi_t)$ is approximated by a neural network $s_\phi(\xi_t, t)$ with parameters ϕ trained via denoising score matching (Hyvärinen & Dayan, 2005; Song et al., 2021). Sampling ξ_T from the Gaussian prior and integrating (10) backwards in time yields samples $\xi \sim p^{\text{ref}}(\xi)$, our generative approximation to p_0. In practice, we use pre-trained diffusion models, which can optimally be conditioned (e.g., via text prompts) to produce domain-specific designs.

Information-Guided Diffusion We aim to sample from the EIG-tilted distribution $p^*(\xi)$. This amounts to adding an extra drift term to Equation (10) of the form

$$u(\xi_t, t) = g(t)^2 \nabla_{\xi_t} \log \mathbb{E}\left[\exp(\alpha^{-1}\mathrm{EIG}(\xi_0)) \mid \xi_t\right] \approx g(t)^2 \alpha^{-1} \nabla_{\xi_t} \mathbb{E}\left[\mathrm{EIG}(\xi_0) \mid \xi_t\right] \tag{11}$$

where the approximation follows by assuming that the EIG of ξ_0 is a function of ξ_t with additive noise. This approximation is still intractable, though, as $\mathbb{E}\left[\mathrm{EIG}(\xi_0) \mid \xi_t\right]$ is unknown. Inspired by recent work on inverse problems (Chung et al., 2024), we define $\widehat{\xi_0}(\xi_t) := \mathbb{E}[\xi_0 \mid \xi_t]$ and approximate $\mathbb{E}\left[\mathrm{EIG}(\xi_0) \mid \xi_t\right] \approx \mathrm{EIG}(\widehat{\xi_0}(\xi_t))$ which follows from approximating the intractable $p(\xi_0 \mid \xi_t)$ by a delta function located at its mean. Critical to making this approximation tractable is Tweedie's formula (Robbins, 1956; Efron, 2011; Meng et al., 2021), which allows us to approximate $\widehat{\xi_0}(\xi_t)$ in terms of the score function $s_\varphi(\xi_t, t)$ without needing to simulate the SDE (10). For instance, when $f(\xi_t, t) = -\frac{1}{2}\beta(t)\xi_t$ and $g(t) = \sqrt{\beta(t)}$ (i.e., DDPM (Ho et al., 2020) or the VP-SDE (Song et al., 2021)), Tweedie's formula may be written as $\widehat{\xi_0}(\xi_t) = (\xi_t + (1 - \alpha_t)\nabla_{\xi_t} \log p_t(\xi_t))/\sqrt{\alpha_t}$, $\alpha_t = \exp(-\int_0^t \beta(s)\,\mathrm{d}s)$ enabling an efficient approximation of (11) using our pre-trained score network. Altogether, we obtain an approximate sampler for $p^*(\xi)$ by solving the SDE

$$\mathrm{d}\xi_t = \left[f(\xi_t, t) - g(t)^2\left(s_\varphi(\xi_t, t) + \alpha^{-1}\nabla_{\xi_t}\mathrm{EIG}\left(\widehat{\xi_0}(\xi_t)\right)\right)\right]\,\mathrm{d}t + g(t)\,d\overleftarrow{W}_t \tag{12}$$

backwards in time. We initialize from Gaussian noise, acknowledging a small bias from not adjusting the initial distribution (Uehara et al., 2024). In practice this simply adds a scaled EIG-gradient estimate to the pre-trained score network at each step, which we find sufficient for high-quality designs without additional Langevin corrections targeting $p^*(\xi)$. We note that the EIG gradient used during guidance is itself an approximated quantity, e.g., via (3). Importantly, this does not require reparametrization, so that our method is applicable in a broad set of contexts. If y is not discrete, alternative EIG gradient estimators can be used instead (see e.g. Rainforth et al. (2024, Section 3)).

In some applications, our task calls for *sets* of designs. We consider several applications of this nature in Section 6. In Appendix B.3, we discuss how our techniques can be extended to the set-valued case.

4.2 DiffBED: BED with Information-Guided Diffusion

Our end-to-end procedure for `DiffBED` is similar to the standard (sequential) BED framework, except that our optimization for ξ at each experiment iteration utilises the guided diffusion technique in Section 4.1. In Appendix B, we provide a full discussion of the details needed to implement `DiffBED`, including an algorithmic description in Algorithm 1.

For sequential BED, we require a high-fidelity and fast posterior sampler. A key design choice in `DiffBED` is to perform inference in a latent space rather than directly in pixel space. This approach exploits the fact that, in many ostensibly high-dimensional tasks, the information we care about lives in a much lower-dimensional space (such as perceptual features) while other variations (background or pixel noise) can be ignored. This makes sequential BED feasible to scale while remaining compatible with a wide range of problems where the likelihood is naturally defined on top of an encoder.

Concretely, we embed θ using a trained encoder and place a simple prior on the resulting latent vector (e.g., a Gaussian or a uniform distribution on the unit sphere). Although we work in an embedding space, the latent dimension can still be moderately high (e.g., 64 dimensions), which is sufficient for expressivity but tractable for inference. Posterior inference over θ is done in this latent space using fast particle-based filtering methods (Johansen, 2009), yielding high-fidelity posterior samples at each sequential iteration. From these latent posterior samples we can also recover image-space designs – for instance, by nearest-neighbour retrieval from a large pool or by guiding a diffusion model to synthesize images whose embeddings match the posterior particles (e.g., via cosine similarity). Overall, this strategy makes `DiffBED` scalable and flexible: it retains high-quality inference while seamlessly interfacing with modern generative models. Details of our particle filtering and inverse-mapping procedures are given in Appendix C.

5 Related Work

Although the framework of BED has a long history (Lindley, 1956; Bernardo, 1979; Sebastiani & Wynn, 2000), scaling BED to realistic settings remains an open challenge (Rainforth et al., 2024; Huan et al., 2024). Viewing challenge as one of computational costs, a long list of works (Foster et al., 2019; 2020; Goda et al., 2022; Ao & Li, 2023; Iollo et al., 2025a) have looked to provide improved gradient estimators compared to simple nested Monte Carlo (Rainforth et al., 2018). Notably, Iollo et al. (2025a) also considers the use of diffusion models, but only in an attempt to improve scaling in the *target variable space*, θ, by using a diffusion model for their prior, $p(\theta)$. All of the aforementioned works are restricted to *low–dimensional design spaces*, with the 15-dimensional and 20-dimensional design spaces considered by Iollo et al. (2025b) and Ivanova et al. (2021) respectively being some of the highest dimensional applications of sequential BED. By contrast, we successfully conduct sequential design optimisation in design spaces of over 750 000 dimensions.

Recent BED work has also considered leveraging LLMs to generate natural language questions as experiment designs to be used in preference elicitation (Choudhury et al., 2025; Kobalczyk et al., 2025). A batch of candidate designs is generated by an LLM, ranked, and the design with the highest EIG is selected. Handa et al. (2024) similarly considers preference elicitation using BED and LLMs, but assumes the parameter and design are supported on an explicit and fixed low-dimensional feature space. On the other hand, `DiffBED` is explicitly focused on the setting where the design space is high-dimensional and continuous, leveraging gradient-based optimization.

Active learning (AL) (Settles, 2009) also aims to select informative data, but typically with the aim of learning a predictive model rather than learning about a specific target quantity as in BED. Much of the AL literature focuses on pool-based selection, where an existing set of unlabelled examples is available (Houlsby et al., 2011; Gal et al., 2017). Some works consider *query synthesis*, i.e., generating inputs directly, often by optimizing an acquisition function in a generative latent space to ensure plausible queries (Zhu & Bento, 2017; Schumann & Rehbein, 2019; Mayer & Timofte, 2020). Instead, we focus on the more general setting of experimental design (which subsumes active learning); active learning approaches are not generally applicable the problems considered in our experiments.

Other works have also previously studied model misspecification in BED (Forster et al., 2025; Overstall & McGree, 2022; Go & Isaac, 2022; Feng et al., 2015). However, they all look to address this by adjusting the model or the objective function itself. In addition, a few works have considered that misalignment may not be constant across the design space, and explicitly modelled this using a bias term (Sürer et al., 2024; Oliveira et al., 2024). Adjusting our model by learning an explicit bias term is not feasible in our high-dimensional, low-data setting, as this would require access to large quantities of experimental outcome data. Beyond acknowledging that model misspecification causes challenges for inference, we are the first to demonstrate that reward hacking–style behaviour is unavoidable in high-dimensional design spaces when using an imperfect, learned likelihood model. In such complex domains, adding an explicit bias term or an 'off-manifold' penalty is not feasible. Thus, `DiffBED` provides the first scalable solution by instead introducing a design prior to regularise the design optimisation, effectively addressing this failure mode.

6 Experiments

We now perform an extensive empirical evaluation of our proposed `DiffBED` method. Although `DiffBED` is not specific to any one experimental setting, our experiments are unified by the theme

Figure 3: Design sets of four images generated by `DiffBED` for CelebA Search, against two ground truths (top and bottom row), at experiment iteration 1 (centre, both rows) and 40 (right, both rows).

of human feedback elicitation, as this encapsulates a broad range of important, real-world tasks with high-dimensional designs. In particular, we focus on the setting where designs consist of one or more images. We consider a range of datasets and feedback types, which include binary rankings, rankings of a subset of images from the design set, and discrete ratings. We defer in depth experimental details, including additional results and ablations, to the Appendix.

Baselines Our primary baseline is the current standard paradigm (Huan et al., 2024; Rainforth et al., 2024) for BED, namely, direct gradient-based maximization of the EIG estimator (3) (Huan & Marzouk, 2014; Foster et al., 2020). We refer to this baseline simply as *BED*. We also compare against *Entropy*, a variant of `DiffBED` where we guide the diffusion model with the marginal predictive entropy rather than the EIG. In addition, we consider *Rank*, an ablation of `DiffBED` where we generate a set of 1000 unguided candidate designs upfront (where the set size is chosen to roughly match runtime to `DiffBED`), then select at each iteration the candidate with the highest estimated EIG. To ensure the strongest baseline possible, we take these candidate designs directly from the data originally used to train the diffusion model, rather than actually diffusing them. Finally, we compare against *Random*, a simple baseline where designs are selected uniformly at random from a discrete set of feasible designs.

Metrics To evaluate the effectiveness of the various design strategies, at each experiment iteration we compute the average cosine similarity between the ground truth, θ_{true}, and our current posterior θ samples. This measures our ability to recover a ground truth target variable. We also evaluate the incremental EIG of the chosen design at each step; we emphasise that this should not be viewed as a success metric itself but rather an insightful quantity to track. We additionally include several qualitative results (c.f. Figures 1 and 2). All numerical results on MNIST are averaged over 25 random seeds, while the higher-dimensional CelebA and Zappos runs are averaged over 10.

6.1 Information-Theoretic Search

We first evaluate `DiffBED` on an information-theoretic search task, where the goal is to recover a ground-truth image based on feedback from a user. Here, designs are sets of images and the outcomes y are rankings indicating the relative similarity of each design to the ground-truth image. As a motivating example, suppose an eyewitness of a crime is being interviewed in order to construct possible images of the suspect that we wish to be perceptually close to the true suspect. Though it is not generally possible for the eyewitness to directly generate accurate images, they likely can positively identify a photo of the true suspect if shown one and more generally provide feedback when shown images. We can therefore instead iteratively show the eyewitness a set of candidate images, $\xi_k = (\xi_k^1, \ldots, \xi_k^J)$ and have them provide feedback by ranking the images in how well they match the suspect. Such an approach is commonly deployed by UK police forces, but with software that uses low-resolution images and chooses them in a heuristic manner (VisionMetric, 2019).

To apply BED, we require a model capturing the complex relationship between the sketch and the victim's response. Since a human's perception of images and identities does not operate at a pixel-by-pixel resolution, we can assume a reasonable model is $p(y_k \mid \theta, \xi_k)$, where θ is a rich, sufficiently high-dimensional feature space encoding of an image, following Section 4.2, and our likelihood is based around the similarity of each ξ_k^j to the underlying θ. For our experiment setup, the simulated participant is given a set of $N = 4$ images and their response, y, is a ranking of the top $M = 2$ images, based on the relative perceived similarities of each image to the ground truth. Under this setup, we evaluate `DiffBED` on MNIST and CelebA (LeCun et al., 1998; Liu et al., 2015), using SimCLR

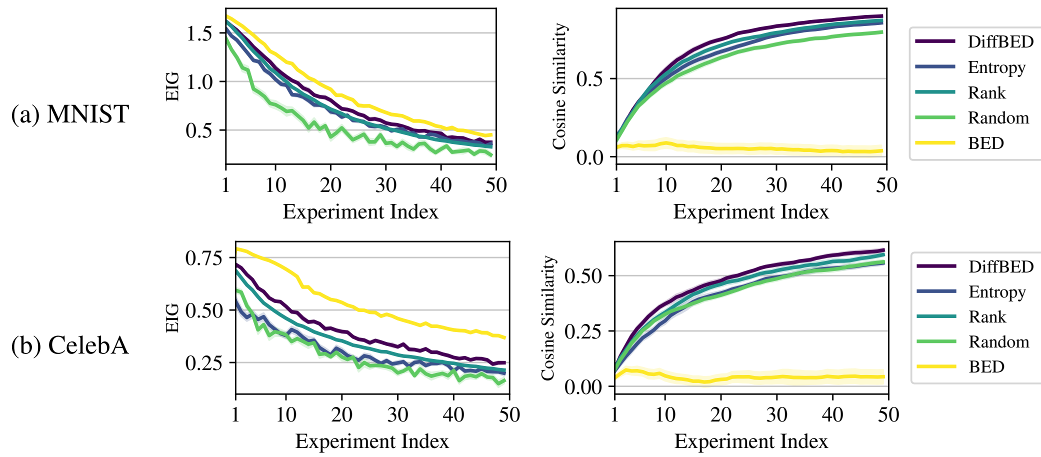


Figure 4: EIG and cosine similarities for search (mean with $\pm$ std. error shading), where designs are sets of four images and responses are the rank of the top-two candidates. `DiffBED` achieves the highest mean cosine similarity, while standard BED fails to solve the task despite achieving high EIGs.

embeddings (Susmelj et al., 2020; Chen et al., 2020) for the former and a pre-trained VGGFace2 model (Cao et al., 2018) fine-tuned using a triplet loss for the latter. For full setup details see Appendix A.

We plot the resulting EIG and cosine similarities in Figure 4. Standard BED is capable of achieving high EIG under the assumed likelihood model for both datasets. However, due to model misalignment, the designs produced are imperceptible from pure noise, and so the return responses are meaningless. Therefore, the cosine similarity between θ_{true} and posterior samples remains effectively zero throughout, indicating that standard BED has failed to solve the task.

While `DiffBED` inevitably fails to achieve as high an EIG as standard BED, by sticking to a realistic design manifold it succeeds at the underlying task, producing effective learning in the true θ as reflected in the cosine similarity plots. We also see that it outperforms the random baselines, confirming the benefit of our information-guided sampling. Finally, we see that `DiffBED` outperforms the ablations of *Entropy* and *Rank*. We highlight that the relative performance gap between `DiffBED` and Rank increases in favour of `DiffBED` when the search space for designs is increased, as seen in the search experiment on CelebA over MNIST. In higher-dimensional design spaces, `DiffBED` is more efficient, as it leverages gradient information to directly generate informative designs, whereas Rank requires many candidate designs to, by chance, identify an informative design. In the experiments above, `DiffBED` outer performs Rank despite only generating a single design set at each iteration, whereas Rank requires considers 1000 candidate design sets. It is particularly notable for this problem that predictive entropy does not in anyway encourage the set of images the user is asked to rank to be different, with the EIG needed to capture the nuance that the rankings need to be informative, not simply uncertain.

6.2 Active Preference Elicitation

We now consider the setting of *active preference elicitation*. Motivating examples for this are wide spread, including recommender systems that are tailored to an individuals preferences. For example, a dating site might wish to recommend profiles that a user is likely to engage with. To do this, the company may suppose that a users preferences can be distilled into distinct interpretable features, indicating the presence of, for example, glasses, or a smile in the image. To infer a user's preferences, we can learn from a user's preference over two potential profiles.

Concretely, for this problem we setup θ as a unit-normalised vector in which each element is a preference weighting for binary CelebA attributes. We use a Bradley-Terry (BT) response model, in which designs are pairs of images. We parametrise the latent reward as being proportional to the dot product between the preference vector and a vector of classifier probabilities that each attribute is present for each image in the pair. See Appendix A for additional experimental details, and Appendix C for details about the inference.

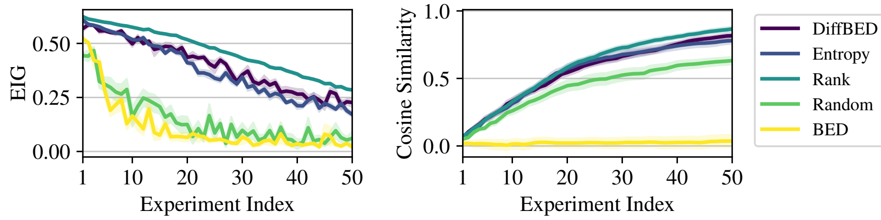


Figure 5: EIG and cosine similarity for preference elicitation on CelebA. Designs consist of image pairs with binary preference responses. Standard BED yields EIG values comparable to random selection and struggles to identify the true preference vector (low cosine similarity), while `DiffBED` maintains higher cosine similarity and more informative designs.

Figure 6: Example high resolution designs produced by `DiffBED` on the Zappos dataset.

As shown in Figure 5, `DiffBED` achieves consistently high EIG designs and substantially outperforms both the standard BED and random baselines by producing much higher cosine similarities. Interestingly, the EIG performance of the standard BED baseline is itself much lower than `DiffBED`, especially after the first few iterations, even though it still produces designs that look like pure noise. The likely reason that this is happening is that the EIG optimisation is struggling more than in information-theoretic search, noting that as there are only binary images and not a single ground truth target image, the signal for choosing good designs is weaker here. By contrast, the reference prior in `DiffBED` also helps in guiding the optimisation process towards sensible designs, so that it actually assists in finding regions of the design space with high EIG when the signal for the latter is weak or noisy.

While `DiffBED` again outperforms *Entropy* on this problem, *Rank* now slightly outperforms it. This may again in part be down to the difficulty of the EIG optimisation, but it is also likely because we are only looking at pairs of images and the space of suitable image pairs is easier to search through a guess–and–check strategy than it is to choose good sets of four images. We thus see that *Rank* provides a useful variant on our `DiffBED` approach for this simpler setup, but is less useful when simple sampling from $p^{\text{ref}}(\xi)$ is not sufficient for generating good candidate designs.

6.3 Text-to-Image Foundation Models

We now scale our application of `DiffBED` even further by leveraging text-to-image foundation models as p^{ref}, focusing on the problem of preference elicitation over e-commerce products. Specifically, we consider a setup where the user is shown a single image of a shoe and asked to give a rating of 1 to 5, and we then use this to try and hone in on the users notion of an "ideal shoe" that we can use for making recommendiations.

For the reference diffusion model, we use Stable Diffusion v1.5 (Rombach et al., 2022), a 1B parameter foundation model, fine-tuned on the Zappos (Yu & Grauman, 2014) dataset, which provides high resolution (512×512) images of shoes. To model a user's feedback, we use the Ordinal Logit Model which assumes the discrete score is correlated with an underlying reward proportional to the similarity of the design image to the image of the reference shoe. See Appendix A.3.3. Figure 6 shows that `DiffBED` is able to produce effective qualitative designs that highly realistic. Figure 7 provides quantitative results and confirms these designs are informative: `DiffBED` and *Rank* perform similarly, while outperforming all other methods. Standard BED again fails to learn anything meaningful.

7 Conclusion

We present `DiffBED`, a technique which enables us to scale gradient-based Bayesian experimental design to high-dimensional, continuous design spaces. Our approach is based on guiding a diffusion model pre-trained on feasible designs with a principled information-theoretic acquisition function, allowing us to produce designs which are simultaneously realistic and highly informative. We

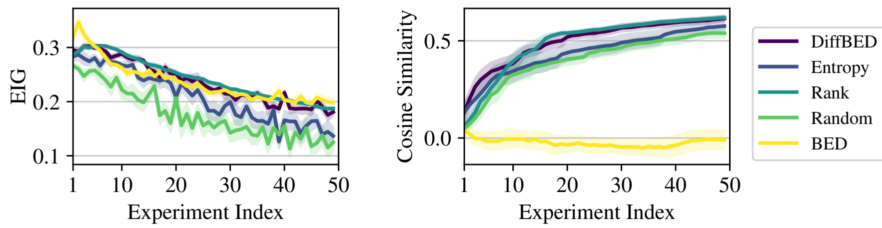


Figure 7: Search on the large-scale Zappos dataset with high-resolution (512×512) images. Designs are single images with discrete ratings as responses. Even at this scale, `DiffBED` remains effective.

showcase `DiffBED` on a suite of preference learning tasks which demonstrate the first successful application of BED with image-scale designs. Taken together, these results highlight the potential of `DiffBED` as a general framework for bringing BED to complex, real-world domains.

References

Ziqiao Ao and Jinglai Li. On estimating the gradient of the expected information gain in bayesian experimental design, 2023.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models, 2023.

José M Bernardo. Expected information as expected utility. *the Annals of Statistics*, pp. 686–690, 1979.

Tom Blau, Edwin V Bonilla, Iadine Chades, and Amir Dezfouli. Optimizing sequential experimental design with deep reinforcement learning. In *International conference on machine learning*, pp. 2107–2128. PMLR, 2022.

Ralph Bradley and Milton E Allan. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324–345, 1952.

Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74. IEEE, 2018.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, August 1995.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

Deepro Choudhury, Sinead Williamson, Adam Goliński, Ning Miao, Freddie Bickford Smith, Michael Kirchhof, Yizhe Zhang, and Tom Rainforth. Bed-llm: Intelligent information gathering with llms and bayesian experimental design, 2025.

Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2024.

D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, December 1956.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexander Denker, Francisco Vargas, Shreyas Padhy, Kieran Didi, Simon Mathis, Riccardo Barbano, Vincent Dutordoir, Emile Mathieu, Urszula Julia Komorowska, and Pietro Lio. Deft: Efficient fine-tuning of diffusion models by learning the generalised h-transform. *Advances in Neural Information Processing Systems*, 37:19636–19682, 2024.

Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. Innovation in the pharmaceutical industry: New estimates of R&D costs. *J. Health Econ.*, 47:20–33, May 2016.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. *An introduction to sequential Monte Carlo methods*. Springer, New York, NY; New York, 2001.

Sergey Dushenko, Kapildeb Ambal, and Robert D McMichael. Sequential bayesian experiment design for optically detected magnetic resonance of nitrogen-vacancy centers. *Physical review applied*, 14(5):054036, 2020.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Chi Feng et al. *Optimal Bayesian experimental design in the presence of model error*. PhD thesis, Massachusetts Institute of Technology, 2015.

David B Fogel. Factors associated with clinical trials that fail and opportunities for improving the likelihood of success: A review. *Contemp. Clin. Trials Commun.*, 11:156–164, September 2018.

Alex Forster, Desi R Ivanova, and Tom Rainforth. Improving robustness to model misspecification in bayesian experimental design. In *7th Symposium on Advances in Approximate Bayesian Inference Workshop Track*, 2025.

Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. *Advances in neural information processing systems*, 32, 2019.

Adam Foster, Martin Jankowiak, Matthew O'Meara, Yee Whye Teh, and Tom Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pp. 2959–2969. PMLR, 2020.

Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International conference on machine learning*, pp. 3384–3395. PMLR, 2021.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data, 2017.

M Girolami and B Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011.

Jinwoo Go and Tobin Isaac. Robust expected information gain for optimal bayesian experimental design using ambiguity sets. In *Uncertainty in Artificial Intelligence*, pp. 728–737. PMLR, 2022.

Takashi Goda, Tomohiko Hironaka, Wataru Kitade, and Adam Foster. Unbiased mlmc stochastic gradient-based optimization of bayesian experimental designs. *SIAM Journal on Scientific Computing*, 44(1):A286–A311, 2022.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Kunal Handa, Yarin Gal, Ellie Pavlick, Noah Goodman, Jacob Andreas, Alex Tamkin, and Belinda Z Li. Bayesian preference elicitation with language models. *arXiv preprint arXiv:2403.05534*, 2024.

Marcel Hedman, Desi R Ivanova, Cong Guan, and Tom Rainforth. Step-DAD: Semi-amortized policy-based Bayesian experimental design. In *International Conference on Machine Learning (ICML)*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning, 2011.

Xun Huan and Youssef M Marzouk. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6), 2014.

Xun Huan, Jayanth Jagalur, and Youssef Marzouk. Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840, 2024.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Jacopo Iollo, Christophe Heinkelé, Pierre Alliez, and Florence Forbes. Bayesian experimental design via contrastive diffusions. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Jacopo Iollo, Geoffroy Oudoumanessah, Carole Lartizien, Michel Dojat, and Florence Forbes. Active mri acquisition with diffusion guided bayesian experimental design. *arXiv preprint arXiv:2506.16237*, 2025b.

Desi R Ivanova, Adam Foster, Steven Kleinegesse, Michael U Gutmann, and Thomas Rainforth. Implicit deep adaptive design: Policy-based experimental design without likelihoods. *Advances in neural information processing systems*, 34:25785–25798, 2021.

Adam Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. 2009.

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

Katarzyna Kobalczyk, Nicolas Astorga, Tennison Liu, and Mihaela van der Schaar. Active task disambiguation with llms. *arXiv preprint arXiv:2502.04485*, 2025.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Dennis Victor Lindley. *Bayesian statistics: A review*. SIAM, 1972.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 2951–2959, 2015.

Thomas J Loredo. Bayesian adaptive exploration. In *AIP Conference Proceedings*, volume 707, pp. 330–346. American Institute of Physics, 2004.

R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3071–3079, 2020.

P Mccullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society: Series B (Methodological)*, 42(2):109–127, 1980.

Chenlin Meng, Yang Song, Wenzhe Li, and Stefano Ermon. Estimating high order gradients of the data distribution by denoising. *Advances in Neural Information Processing Systems*, 34: 25359–25369, 2021.

Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt. A tutorial on adaptive design optimization. *Journal of mathematical psychology*, 57(3-4):53–67, 2013.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2018.

Rafael Oliveira, Dino Sejdinovic, David Howard, and Edwin V Bonilla. Bayesian adaptive calibration and optimal design. *Advances in Neural Information Processing Systems*, 37:56526–56551, 2024.

Antony Overstall and James McGree. Bayesian decision-theoretic design of experiments under an alternative model. *Bayesian Analysis*, 17(4):1021–1041, 2022.

Costas Papadimitriou. Optimal sensor placement methodology for parametric identification of structural systems. *Journal of sound and vibration*, 278(4-5):923–947, 2004.

S Patterson and Y W Teh. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in neural information processing systems*. 2013.

Steven M Paul, Daniel S Mytelka, Christopher T Dunwiddie, Charles C Persinger, Bernard H Munos, Stacy R Lindborg, and Aaron L Schacht. How to improve R&D productivity: the pharmaceutical industry's grand challenge. *Nat. Rev. Drug Discov.*, 9(3):203–214, March 2010.

R L Plackett. The analysis of permutations. *J. R. Stat. Soc. Ser. C. Appl. Stat.*, 24(2):193, 1975.

Tom Rainforth. *Automating inference, learning, and design using probabilistic programming*. PhD thesis, University of Oxford, 2017.

Tom Rainforth, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pp. 4267–4276. PMLR, 2018.

Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024.

K Rawlik, M Toussaint, and S Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *Proceedings of Robotics: Science and Systems VIII*. 2012.

Herbert Robbins. An empirical Bayes approach to statistics. In *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, 1954–1955, Vol. I*, pp. 157–163, Berkeley and Los Angeles, 1956. University of California Press.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Raphael Schumann and Ines Rehbein. Active learning via membership query synthesis for semi-supervised sentence classification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pp. 472–481, 2019.

Paola Sebastiani and Henry P Wynn. Maximum entropy sampling and optimal bayesian experimental design. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(1):145–157, 2000.

Burr Settles. Active learning literature survey. 2009.

Ben Shababo, Brooks Paige, Ari Pakman, and Liam Paninski. Bayesian inference and online experimental design for mapping neural microcircuits. *Advances in Neural Information Processing Systems*, 26, 2013.

Yifei Shen, Xinyang Jiang, Yezhen Wang, Yifan Yang, Dongqi Han, and Dongsheng Li. Understanding and improving training-free loss-based diffusion guidance, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

Özge Sürer, Matthew Plumlee, and Stefan M Wild. Sequential bayesian experimental design for calibration of expensive simulation models. *Technometrics*, 66(2):157–171, 2024.

Igor Susmelj, Matthias Heller, Philipp Wirth, Jeremy Prescott, Malte Ebner, et al. Lightly. https://github.com/lightly-ai/lightly, 2020.

Masatoshi Uehara, Yulai Zhao, Kevin Black, Ehsan Hajiramezanali, Gabriele Scalia, Nathaniel Lee Diamant, Alex M Tseng, Tommaso Biancalani, and Sergey Levine. Fine-tuning of continuous-time diffusion models as entropy-regularized control. *arXiv preprint arXiv:2402.15194*, 2024.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review, 2025.

Joep Vanlier, Christian A Tiemann, Peter AJ Hilbers, and Natal AW van Riel. A bayesian approach to targeted experiment design. *Bioinformatics*, 28(8):1136–1142, 2012.

Benjamin T Vincent and Tom Rainforth. The darc toolbox: automated, flexible, and efficient delayed and risky choice experiments using bayesian adaptive design. *PsyArXiv. October*, 20, 2017.

VisionMetric. Deep learning strategies for accurate identification from facial composite images (e2id), 2019.

Andrew B Watson. Quest+: A general multidimensional bayesian adaptive psychometric method. *Journal of Vision*, 17(3):10–10, 2017.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. Tfg: Unified training-free guidance for diffusion models, 2024.

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

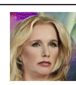
Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

A LIKELIHOOD MODELS

This section provides details for all likelihood models studied in our experiments (Section 6).

When picking an experiment paradigm, practitioners must consider the trade-off between the information conveyed in a given observation (e.g. binary preference, partial ranking), and the cost of running said experiment. For example, a full-ranking contains high amounts of information per observation, but may not be feasible to collect. We summarize the paradigms studied in our experiments in Table 1).

Table 1: Common paradigms for eliciting human feedback.

Paradigm	**Example design**	**Example feedback**	**Example Likelihood**
Binary Ranking	 	"Image 1 $\succ$ Image 2"	Bradley–Terry (BT) (Bradley & Allan, 1952)
Ranking k designs from n	   	"Image 1 $\succ$ Image 2 $\succ$ Image 3"	Plackett–Luce (PL) (Plackett, 1975; Luce, 1959)
Discrete rating		"4 out of 5 stars"	Ordinal Logit (OL) (Mccullagh, 1980)

A.1 LIKELIHOOD PMFS

We now provide the functional form of the likelihood models considered in our experiments. Note that all of the models leverage a latent reward model, $r_\theta(\xi)$, parametrized by the quantity of interest, θ, which assigns a score $r_\theta(\xi) \in \mathbb{R}$ to the design/each element in the design set.

Binary Preferences: Bradley-Terry (Bradley & Allan, 1952)

Let $\xi = \{\xi_1, \xi_2\}$ be a design set of two items. The Binary Bradley-Terry model assumes,

$$p(y = \xi_i \succ \xi_j \mid \theta, \xi) = \frac{\exp(r_\theta(\xi_i))}{\exp(r_\theta(\xi_i)) + \exp(r_\theta(\xi_j))}.$$

Partial Rankings: Plackett-Luce (Plackett, 1975)
Let $\xi = \{\xi_1, \xi_2, \dots, \xi_S\}$ be a design set of S items. Suppose we observe a partial ranking of the form

$$\xi_{\sigma_1} \succ \xi_{\sigma_2} \succ \cdots \succ \xi_{\sigma_M}, \quad \text{with } M \leq S,$$

where $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_M)$ is an ordered list of distinct indices indicating the ranked items, and let $C_M := \xi \setminus \{\xi_{\sigma_1}, \xi_{\sigma_2}, \dots, \xi_{\sigma_M}\}$. Then the Plackett–Luce likelihood of the observed partial ranking is

$$p(y = \xi_{\sigma_1} \succ \xi_{\sigma_2} \succ \cdots \succ \xi_{\sigma_M} \mid \theta, \xi) = \prod_{j=1}^{M} \frac{\exp\big(r_\theta(\xi_{\sigma_j})\big)}{\sum\limits_{\xi \in C_j} \exp(r_\theta(\xi))},$$

where C_j is the set of items available at stage j, with $C_1 = \xi$ and $C_{j+1} = C_j \setminus \{\xi_{\sigma_j}\}$.

Discrete Ratings: Ordinal Logit (Mccullagh, 1980) Let ξ being a single item, unlike the Bradley-Terry and Plackett-Luce models, which operate on design sets. Under the Ordinal Logit model, observations are one of K ordered, discrete categories, modelled under the following PMF:

$$p(y = k \mid \xi, \theta) = \sigma\left(\tfrac{b_k - r_\theta(\xi)}{\tau}\right) - \sigma\left(\tfrac{b_{k-1} - r_\theta(\xi)}{\tau}\right), \qquad k = 1, \dots, K,$$

with $\sigma(x) = (1 + e^{-x})^{-1}$.

A.2 Parametrisation of Latent Reward Model

As reiterated in the main body of the paper, in the high-dimensional, challenging settings considered in the paper, defining a likelihood, $p(y|\theta, \xi)$, that mimics human behaviour for all $\{\theta, \xi\}$ is extremely challenging. Further, we note that $p(y|\theta, \xi)$ can often not be directly trained in a supervised manner, and practitioners must leverage domain-specific insight. This is especially true in cases in which the θ of interest is inherently latent, e.g. user preferences. As such, we must often assume the structure of the canonical models. However, in all of the models considered above, we must still define a latent reward model, $r_\theta(\xi)$.

In many problems, e.g. preference-elicitation, the interpretable features that are most pertinent can be identified. In such settings, the following parametrization can be considered: $r_\theta(\xi) = \theta^T d(\xi) = \sum_{k=1}^{K} \theta_k d_k(\xi)$, of K interpretable features, e.g. alignment to a style. In this parametrisation, θ can be viewed as interpretable preference weights for the K different aspects. In other problems, where such interpretable features can not be identified, one can opt for the following parametrisation: $r_\theta(\xi) = \text{cos\_sim}(\theta, d(\xi))$ where $d(.)$ is an encoder of ξ trained in an unsupervised fashion.

A further practical benefit of these parametrisations is that they are data-efficient, arising from the incorporation of task-specific knowledge through the fixed, pre-trained feature extractors, $d(.)$, which should support faster learning. However, we stress that `DiffBED` is agnostic to the parametrisation of the latent reward model, $r_\theta(.)$, and more generally, the response paradigm and likelihood model used.

Temperature hyper-parameter, τ In line with common practice, we restrict the range of the latent reward, $r(\xi)$, to a pre-defined range $[-1, 1]$, for example through normalisation or by computing the reward as cosine similarities, before introducing a multiplicative scaling parameter, τ, to expand the range to $\left[-\tau^{-1}, \tau^{-1}\right]$. This enables explicit control over the sharpness of the assumed likelihood model's response distribution. As $\tau^{-1} \to 0$, the distribution over potential observations for any design across all the models introduced tends to a uniform distribution. On the other hand, increasing τ^{-1} results in peakier likelihoods, and faster shrinkage of the posterior. However, as it assigns a stronger belief on the data generating mechanism, if this does not reflect reality, the degree of misalignment with the true data generating process is also increased.

A.3 Details of Latent Reward Models

In this section, we present the specific details of the latent reward models used in our experiments. In the search experiments, we leverage unsupervised encoders trained through contrastive losses, to encourage learning general representations of the data. In the preference elicitation experiment, we leverage interpretable feature extractors.

A.3.1 MNIST

For search, we train an encoder using the SimCLR loss (Chen et al., 2020). We leverage a simple CNN architecture, with two convolutional layers and fully-connected layer. The embedding dimensionality is $K = 32$, with a projection-head of size $P = 512$, a hyper-parameter for computing SimCLR contrastive losses. The encoder is trained for 50 000 steps, with a batch size of 1024. The underlying reward function used in our Bradley-Terry model is $r_\theta(\xi) = \tau^{-1}\text{cosine\_sim}(\text{simclr}(\xi), \theta)$, where simclr is our encoder. We take τ^{-1} to be 25 in Figure 4.

To explicitly incorporate model misspecification into the responses, we leverage a pre-trained discriminator that detects out-of-distribution (OOD) MNIST images. For in-distribution images, the simulated observation y is from a re-normalised PMF of the rankings that don't include any OOD images, and in cases where all images are classed as OOD, the ranking observations are generated uniformly at random.

A.3.2 CelebA

Search We train a CelebA encoder on the CelebA train set. We take a pre-trained VGGFace2 model as our backbone (Cao et al., 2018), and fine-tune it. We remove the original last layer and replace it with a 64 dimensional linear last layer, hence θ is, 64 dimensional. We freeze the backbone, training only the last layer weights, before unfreezing the final block before the last layer, and fine tuning these weights alongside the last layer's weights. We train using a triplet loss, since

CelebA includes identity labels. The underlying reward function used in our Placket-Luce model is $r_\theta(\xi) = \tau^{-1}\text{cosine\_sim}(\text{vgg}(\xi), \theta)$, where vgg is our fine-tuned encoder. We take τ^{-1} to be 25. We run the experiment for 50 iterations.

Preference Elicitation We train a supervised feature extractor on the CelebA dataset by fitting a multi-label attribute classifier to predict the following $C = 23$ characteristics that have binary labels in the dataset:

```
Bald, Wavy_Hair, Straight_Hair, Receding_Hairline,
Bangs, Sideburns, Black_Hair, Gray_Hair, Blond_Hair,
Brown_Hair, No_Beard, 5_o_Clock_Shadow, Mustache, Goatee,
Big_Lips, Big_Nose, Eyeglasses, Smiling, Heavy_Makeup,
Wearing_Lipstick, Wearing_Necklace, Wearing_Earrings.
```

We leverage a ResNet50 model initialised with ImageNet weights, with the final layer removed and replaced with a linear layer that maps to $C = 23$. The model is trained for 25 epochs, with a batch size of 128. We use binary cross-entropy with logits, assigning per-attribute positive weights equal to the negative-to-positive sample ratio to correct for class imbalance, and Adam as the optimizer.

We use a Bradley-Terry model, with underlying reward being $r_\theta(\xi) = \tau^{-1}\sum_{i=1}^{N} d_i(\xi)\cdot\theta_i$, where $d(\xi) \in \mathbb{R}^{|C|}$ is a vector with element $d_i(\xi)$ being the ResNet50 classifier probability that attribute i is present in a design image.

A.3.3 ZAPPOS

We train a Zappos encoder using the SimCLR loss, with $K = 64$ and $P = 512$. We leverage a ResNet50-based model and initialized with torchvision's retrained ImageNet-1K (Deng et al., 2009) weights[1], with the final fully connected (FC) layer removed and replaced by a linear projection to a K embedding space. The model is trained 50,000 steps, with a batch size of 256, a learning-rate of 1, and with SGD as the optimizer. We utilise an Ordinal-Logit model with an underlying reward $r_\theta(\xi) = \tau^{-1}\cos\_\text{sim}(\text{simclr}(\xi), \theta)$, where simclr is our simclr encoder, and θ is the simclr embedding of the ground truth reference image.

B DIFFBED DETAILS

This section contains the algorithmic details needed to implement `DiffBED` in practice. In Algorithm 1, we provide pseucodode which describes how `DiffBED` is applied end-to-end for (sequential) BED problems.

B.1 REFERENCE MODELS

For each experiment, we require a reference diffusion model $p^{\text{ref}}(\xi)$ whose samples produce reasonable designs for the problem at hand. We detail our specific choices for each dataset here.

MNIST We use the training script provided in the PyTorch codebase provided by Song et al. (2021)[2] to train an unconditional MNIST diffusion model. We use an NCSN++ UNet (64 base channels, one residual block per level) with a continuous VP-SDE noise schedule and EMA (0.999). We train for 500k iterations with a batch size of 256, using the Adam optimizer ($lr = 0.0002$, with gradient clipping at norm$= 1.0$).

CelebA and Zappos For the higher dimensional datasets, CelebA (Liu et al., 2015) and Zappos (Yu & Grauman, 2014), we leverage the Hugging Face `diffusers` library[3], which provides standardized pipelines for training, inference, and sampling of diffusion and latent diffusion models.

[1] https://pytorch.org/vision/stable/models.html
[2] https://github.com/yang-song/score_sde_pytorch
[3] https://huggingface.co/docs/diffusers

Algorithm 1 `DiffBED`: BED with Information Guided Diffusion

Input: BED setup: prior $p(\theta)$; likelihood $p(y \mid \theta, \xi)$; experiment steps K
Input: Diffusion: reference score model s_φ^{ref}; SDE steps T; guidance scale α
Input: Inference: particle count N; particle filter;
Output: Designs $\xi_{1:K}$, observations $y_{1:K}$, particles $\{\theta_n^{(K)}\}_{n=1}^N$;

1: **Initialize:** draw $\{\theta_n^{(0)}\}_{n=1}^N \sim p(\theta)$; set $\xi_{1:0} \leftarrow \emptyset$, $y_{1:0} \leftarrow \emptyset$.
2: **for** $k = 1, \ldots, K$ **do** ▷ Sequentially design and run each experiment
— Diffusion-based design sampling —
3: $\xi^{(T)} \sim \mathcal{N}(0; I)$ ▷ Initialize at noise
4: **for** $t = T, T-1, \ldots, 1$ **do**
5: $g_t \leftarrow \nabla_{\xi_t} \widehat{\text{EIG}}\big(\widehat{\xi_0}(\xi_t)\big)$ ▷ Estimate EIG gradient using $\{\theta_n^{(k-1)}\}$
6: $\xi_{t-1} \leftarrow \text{SDEstep}(\xi_t, s_\varphi^{ref}(\xi_t, t), g_t, \alpha)$ ▷ SDESolver step
7: **end for**
8: Set design $\xi_k \leftarrow \xi_0$
— Run experiment and update posterior —
9: $y_k \leftarrow y \sim p(y \mid \theta^\star, \xi_k)$
10: $\xi_{1:k} \leftarrow \xi_{1:k-1} \cup \{\xi_k\}$, $y_{1:k} \leftarrow y_{1:k-1} \cup \{y_k\}$
11: $\{\theta_n^{(k)}\} \leftarrow \text{ParticleFilter}\big(\{\theta_n^{(k-1)}\}, \xi_{1:k}, y_{1:k}\big)$
12: **end for**
13: **return** $\xi_{1:K}, y_{1:K}, \{\theta_n^{(K)}\}$

For CelebA, we use a pre-trained latent diffusion model (Rombach et al., 2022) checkpoint.[4] For Zappos, we use a fine-tuned version of Stable Diffusion v1.5 (SDv1.5). [5]

B.2 Sampling

We now turn to the details involved in sampling from $p^*(\xi)$ using Equation (12). In particular, Shen et al. (2024) considers the shortfalls of training-free guidance of diffusion models and leverage ideas from optimization literature to mitigate them. We find that Polyak step-size parametrisation of the guidance scale is beneficial in finding a favourable trade-off between the informativeness and realism of designs. Namely, this considers a time-dependent guidance scale as a multiplier of the EIG gradient estimator, which we denote as γ_t for brevity, during the reverse-process:

$$\alpha^{-1}(t) = \eta \cdot \frac{\|\epsilon_\varphi(\xi_t, t)\|}{\|\gamma_t\|_2^2}. \tag{13}$$

Across all experiments, we use the Euler–Maruyama discretization of the reverse SDE, equivalent to the ancestral/DDPM sampler. We provide further dataset-specific sampling hyper-parameters below. We use uniform time-steps on all datasets, with 500/250/100 steps on MNIST/CelebA/Zappos, respectively. The choice of the η is an empirical one, determined by the robustness of the reference diffusion model (Ye et al., 2024). We set this parameter by visually inspecting a small set of samples for increasing values of η, setting the maximal value that still consistently produces high-fidelity images. We use $\eta = 0.0375, 0.10$ for CelebA/Zappos experiments, and present example samples produced at this guidance scale in Figure 10 and Figure 6.

As SDv1.5 is a text-conditioned model, we use the following prompt to capture the data-distribution of interest for the Zappos task: `‘‘Studio product photo of a footwear, isolated on white background, high detail’’`. To avoid artefacts, we also use the following negative prompt: `‘‘blurry, low resolution, watermark, deformed’’`.

[4] https://huggingface.co/CompVis/ldm-celebahq-256
[5] https://huggingface.co/benisonjac/finetune-of-stable-diffuson-on-Zappos-shoe-dataset

B.3 Diffusion on Sets

In some applications, we may have access to a diffusion model producing single designs, but our task calls for *sets* of designs. We consider several applications of this nature in Section 6. Our `DiffBED` framework developed in Section 4 can be extended to this setting. In particular, designs are now sets $\xi = \{\xi_1, \ldots, \xi_J\}$ of J elements. We extend p^{ref} to be a set-valued distribution by assuming independence of the elements, i.e., $p^{\text{ref}}(\xi) \propto \prod_{j=1}^{J} p^{\text{ref}}(\xi_j)$.

During the reverse diffusion process, we now maintain a noisy set $\xi_t = \{\xi_{1,t}, \ldots, \xi_{J,t}\}$. We can then write the reverse diffusion process for each individual element $\xi_{j,t}$ as

$$d\xi_{j,t} = \left[f(\xi_t, t) - g(t)^2 \left(s_\varphi(\xi_{j,t}, t) + \alpha^{-1} \nabla_{\xi_{j,t}} \text{EIG}(\widehat{\xi}_0(\xi_t)) \right) \right] \mathrm{d}t + g(t)\, \mathrm{d}\overleftarrow{W}_{j,t}, \qquad j = 1, \ldots, J,$$

where $\widehat{\xi}_0(\xi_t) = \{\widehat{\xi}_0(\xi_{1,t}), \ldots, \widehat{\xi}_0(\xi_{J,t})\}$ is the set of element-wise conditional means and $\mathrm{d}\overleftarrow{W}_{j,t}$ are independent Brownian increments.

Intuitively, our information-guided set diffusion acts as an interacting particle system. The diffusion prior contributes an independent score update for each element, ensuring realism, while the EIG gradient term $\nabla_{\xi_j,t} \text{EIG}(\xi_t)$ introduces cross-element coupling, ensuring informativeness of the entire set as a design.

B.4 Sampling from Tilted Distributions

As discussed in Section 4, we aim to sample from the tilted distribution

$$p^*(\xi) \propto p^{\text{ref}}(\xi) \cdot \exp\left(\alpha^{-1} \text{EIG}(\xi)\right). \tag{14}$$

In this section, we discuss how we achieve this through guided diffusion. Recall that we assume a forward SDE of the form

$$\mathrm{d}\xi_t = f(\xi_t, t)\, \mathrm{d}t + g(t)\, \mathrm{d}W_t \qquad \xi_0 \sim p_0(\xi_0) \qquad t \in [0, T] \tag{15}$$

whose time-reversal is

$$\mathrm{d}\xi_t = \left[f(\xi_t, t) - g(t)^2 \nabla_{\xi_t} \log p_t(\xi_t) \right] \mathrm{d}t + g(t)\, \mathrm{d}\overleftarrow{W_t} \qquad \xi_T \sim p_T(\xi_T) \qquad t \in [0, T]. \tag{16}$$

Using techniques from stochastic optimal control, Uehara et al. (2024) prove the following result. Intuitively, we may obtain an SDE which samples from the tilted distribution $p^*(\xi)$ by adding an additional drift term to our reverse-time SDE and adjusting its initial (i.e., at $t = T$) distribution.

Proposition 1.
Consider the distribution

$$q(\xi) \propto p_T(\xi) \cdot \mathbb{E}_{\xi_0 \sim p^{\text{ref}}} \left[\exp\left(\alpha^{-1} \text{EIG}(\xi_0)\right) \mid \xi_T = \xi \right] \tag{17}$$

and vector field

$$u(\xi, t) = g(t)^2 \nabla_\xi \left(\log \mathbb{E}_{\xi_0 \sim p^{\text{ref}}} \left[\exp\left(\alpha^{-1} \text{EIG}(\xi_0) \mid \xi_t = \xi\right) \right] \right). \tag{18}$$

The SDE

$$\mathrm{d}\xi_t = \left[f(\xi_t, t) - g(t)^2 \nabla_{\xi_t} \log p_t(\xi_t) + u(\xi_t, t) \right] \mathrm{d}t + g(t)\, \mathrm{d}\overleftarrow{W}_t \qquad \xi_T \sim q(\xi_t) \qquad t \in [0, T] \tag{19}$$

is such that solutions at $t = 0$ satisfy $\xi_0 \sim p^*$.

Sampling using (19) requires us to find $q(\xi_T)$ and $u(\xi_t, t)$. In our work, we choose to sample $\xi_T \sim p_T(\xi_T)$ in practice as our initial condition in order to avoid estimating $q(\xi_T)$, acknowledging a small resulting bias.

On the other hand, estimating $u(\xi_t, t)$ is crucial. Inspired by the derivation by Uehara et al. (2024), we assume that $\text{EIG}(\xi_0) = k(\xi_t) + \epsilon_t$ can be written in terms of a fixed function k and time-dependent noise ϵ_t. We further assume ϵ_t is mean zero and independent of ξ_t.

Under these assumptions, we have

$$u(\xi, t) = g(t)^2 \nabla_\xi \left(\log \mathbb{E}_{\xi_0 \sim p^{\text{ref}}} \left[\exp\left(\alpha^{-1} \text{EIG}(\xi_0) \mid \xi_t = \xi\right)\right]\right) \tag{20}$$

$$= g(t)^2 \nabla_\xi \left(\alpha^{-1} k(\xi) + \log \mathbb{E}_{p^{\text{ref}}} \left[\exp\left(\alpha^{-1} \epsilon_t\right) \mid \xi_t = \xi\right]\right) \tag{21}$$

$$= g(t)^2 \alpha^{-1} \nabla_\xi k(\xi) \tag{22}$$

where we use the independence of ϵ_t and ξ in the last step.

Now, from our assumption $\text{EIG}(\xi_0) = k(\xi_t) + \epsilon_t$, we see

$$\mathbb{E}_{\xi_0 \sim p^{\text{ref}}} [\text{EIG}(\xi_0) \mid \xi_t = \xi] = k(\xi) + \mathbb{E}[\epsilon_t \mid \xi_t = \xi] \tag{23}$$

$$= k(\xi) \tag{24}$$

where the last line follows from independence between ϵ_t, ξ_t and the mean-zero assumption. Overall, this yields

$$u(\xi, t) = \alpha^{-1} \sigma^2(t) \nabla_\xi \mathbb{E}_{\xi_0 \sim p^{\text{ref}}} [\text{EIG}(\xi_0) \mid \xi_t = \xi] \,. \tag{25}$$

This is still intractable, though, as it depends on an expectation over the diffusion process. While Uehara et al. (2024) propose to estimate this expectation via regression, this approach can be expensive and non-trivial to successfully carry out in practice. We instead follow a simpler approach inspired by inverse problems (Chung et al., 2024). In particular, we define $\widehat{\xi_0}(\xi) := \mathbb{E}_{p^{\text{ref}}}[\xi_0 \mid \xi_t = \xi]$ to be the conditional mean of noise-free designs given a noisy design $\xi_t = \xi$ at time t. As discussed in Section 4, this conditional expectation can be easily obtained via Tweedie's formula for many diffusion processes of interest. We then approximate

$$\mathbb{E}_{\xi_0 \sim p^{\text{ref}}} [\text{EIG}(\xi_0) \mid \xi_t = \xi] \approx \text{EIG}(\widehat{\xi_0}(\xi)) \tag{26}$$

which follows from approximating this conditional distribution by a Dirac delta at $\widehat{\xi_0}(\xi)$.

Finally, the true score $\nabla_\xi \log p_t(\xi_t)$ is approximated by a pre-trained diffusion model $s_\varphi(\xi_t, t)$. Altogether, we obtain an approximate sampler for $p^*(\xi)$ by solving the SDE

$$\text{d}\xi_t = \left[f(\xi_t, t) - g(t)^2 \left(s_\varphi(\xi_t, t) + \alpha^{-1} \nabla_{\xi_t} \text{EIG}\left(\widehat{\xi_0}(\xi_t)\right)\right)\right] \text{d}t + g(t)\, d\overleftarrow{W}_t \qquad \xi_T \sim q(\xi_T) \tag{27}$$

backwards in time on $t \in [0, T]$.

C Posterior Inference

We now present details on how we conduct inference on the embedding space during active experimentation. In search problems, as the encoders are trained with a cosine-similarity objective, both our likelihoods and the geometry of the problem depend only on the *direction* of θ, not its scale. Similarly, in preference elicitation settings, we often want to learn normalized preference weights [6], in order to allow for explicit control and regularisation over the sharpness of the preference distributions, for example through a scaling hyper-parameter.

In such settings, although θ lives in $\mathbb{R}^D$, the unit-length constraint removes one degree of freedom, leaving an intrinsic dimension of $D - 1$. Accordingly we place a *uniform prior* on the unit sphere $\mathbb{S}^{D-1} = \left\{\theta \in \mathbb{R}^D : \|\theta\|_2 = 1\right\}$ the $(D-1)$-dimensional manifold of D-dimensional vectors of length one. We then explicitly approximate the posterior on the unit sphere.

Particle-Based Inference At the start of the experimentation we draw an i.i.d. set of particles $\{\theta_i^{(0)}\}_{i=1}^N \sim \text{Unif}(\mathbb{S}^{D-1})$ to represent this prior. At each experiment index k, after observing an outcome y_k at design ξ_k, we update particle weights according to the likelihood

$$\log w_i^{(k)} = \log w_i^{(k-1)} + \log p(y_k \mid \theta_i^{(k-1)}, \xi_k).$$

Weights are normalized and particles are resampled via multinomial resampling to prevent degeneracy (Doucet et al., 2001), yielding an empirical posterior approximation $\{\theta_i^{(t)}\}$.

[6]Another alternative is to learn weights that sum to one.

To maintain diversity and ensure that the particle cloud accurately tracks the true posterior on the sphere, we rejuvenate the particles by applying a *projected unadjusted Langevin algorithm* (ULA) on $\mathbb{S}^{D-1}$:

$$\theta \leftarrow \text{Normalize}\Big(\theta + \frac{\epsilon}{2} P_\theta \nabla_\theta \log \pi_k(\theta) + \sqrt{\epsilon}\, P_\theta \eta\Big),$$

where $P_\theta = I - \theta\theta^\top$ projects gradients and noise onto the tangent space $T_\theta\mathbb{S}^{D-1}$ and $\eta \sim \mathcal{N}(0, I)$. This step can be viewed as an unadjusted discretization of the Riemannian Langevin diffusion on $\mathbb{S}^{D-1}$ (Girolami & Calderhead, 2011; Patterson & Teh, 2013), where the drift term is the gradient of the *full current posterior* $\pi_k(\theta) \propto \pi_0(\theta) \prod_{s=1}^{k} p(y_s \mid \theta, \xi_s)$, so that each move incorporates all experimental observations up and including experiment index k.

Computational Cost As the log-likelihoods are parametric functions of simple vector products, i.e. cosine-similarities, between θ and encodings of previous designs, both resampling and Langevin steps are very efficient.[7] Crucially, resampling and Langevin steps do not require expensive neural network evaluations per θ particle. For all experiments, we leverage $N = 10^6$ particles, take 100 Langevin steps, and have step-size $\epsilon = 10^{-4}$. Note that the cost of storing $N = 1000000$ particles of dimensionality $D = 32, 64$ has costs $\approx 122, 244$ MiB of memory, which is a negligible overhead on modern GPUs.

D Location Finding

In this section, we present source location finding, a synthetic problem widely considered in the BED literature (Ivanova et al., 2021; Blau et al., 2022; Iollo et al., 2025a; Hedman et al., 2025). In this problem, the goal is to infer the location of N sources, $\theta_1, \theta_2, ..., \theta_N \in \mathbb{R}^D$. Each source emits a signal which decays according to an inverse square law. At each experiment iteration, a sensor is placed at a location $d \in \mathbb{R}^D$ which records a noisy measurement $y \in \mathbb{R}$ of the signal intensity at the sensor location. Concretely, the (noiseless) signal strength is modelled using$\mu(\boldsymbol{\theta}, \boldsymbol{\xi}) = b + \sum_{c=1}^{C} \frac{\alpha_c}{m + \|\boldsymbol{\theta}_c - \boldsymbol{\xi}\|_2^2}$, where α_c, b, and m are known constants. The observed signal follows a log-normal distribution $(\log y \mid \boldsymbol{\theta}, \boldsymbol{\xi}) \sim \mathcal{N}(\log \mu(\boldsymbol{\theta}, \boldsymbol{\xi}), \sigma)$, with σ representing the standard deviation. A standard Gaussian prior is assumed for each source location, $\boldsymbol{\theta}_c \sim \mathcal{N}(0, I_D)$, and the observation noise is modelled as a log-normal. Following Foster et al. (2021), we set $D = 2$, $C = 2$, $\alpha_1 = \alpha_2 = 1$, $m = 10^{-4}$, $b = 10^{-1}$, $\sigma = 0.25$, and design $K = 30$ sequential design optimizations.

In the BED literature, methods are typically evaluated using data generated from the *assumed* likelihood model. However, the fundamental issue tackled by `DiffBED` is the varying misalignment between the assumed and data-generating likelihood across the design space. To evaluate methods under this setting, we introduce model misalignment that is a function of the design variable to the location finding problem. Concretely, we assume the aforementioned form of the likelihood, denoted as p_{model}, during design optimisation but generate samples from the following latent mixture model:

$$y \sim p_{\text{true}}(y|\theta, \xi) = z(\xi) p_{\text{model}}(y|\theta, \xi) + (1 - z(\xi)) p_{\text{ood}}(y|\theta, \xi)$$

where $z(\xi)$ is a discrete latent variable:

$$z(\xi) = \begin{cases} 1, & \text{if } \xi \in \Xi, \\ 0, & \text{if } \xi \notin \Xi. \end{cases}$$

In many real-world source-finding tasks, such as localizing radio beacons, chemical or radiation leaks, or acoustic emitters, the assumed physical model (e.g., an inverse-square decay) is only accurate in regions where the environment is simple and well-behaved. In such areas, such as open or unobstructed areas, the measurements should typically follow the expected model, p_{model}. However, when the sensor moves into more complex environments, such as areas with obstacles, turbulent airflow, or reflecting surfaces, the underlying physics changes, producing systematically different observations. Our mixture formulation captures this spatially dependent model fidelity by allowing designs inside certain regions to follow the assumed model, while those outside follow an alternative, unknown likelihood, p_{ood}. Effectively modelling the physics outside of the simple case where p_{model} is accurate may not be feasible. If this is the case, we may wish to find informative design, whilst

[7] Note that embeddings of designs/design sets can be cached, and as such, just needs to be computed once. For the rest of the roll-out, it can be shared across all particles for resampling and Langevin-step computation.

encouraging adherence to a design prior, where we can trust our ability to model the data-generating process.

Misalignment In this synthetic experiment, we define the well-aligned region, Ξ, to be eight circles with radii equal to 0.4, and each circle centred on the circumference of the unit circle (such that the spacing between the circles is even). This geometry is inspired by the eight Gaussian dataset, commonly used for demonstrating generative models on simple datasets. For p_{ood}, we use the same form and parameters as the aforementioned p_{model} but with $\alpha_1 = \alpha_2 = 0.5$. As such, in misaligned regions, the signal mean, $\mu(\boldsymbol{\theta}, \boldsymbol{\xi})$ is systematically lower. We stress that in real-world scenarios, we have very limited knowledge of p_{ood}, and hence cannot write down an explicit $p_{\text{true}}(y|\theta, \xi)$.

Design Strategies: For an implicit design prior, a diffusion model trained to sample a Gaussian mixture model (GMM) with eight Gaussians with evenly-spaced means on the unit circle and diagonal covariance matrices ($\Sigma = \sigma^2 I_2; \sigma = 0.1$) could be used. Note that this design prior assigns high mass to regions in Ξ and thus reflects our preferences towards areas of design space that exhibit lower misalignment. For a design prior that is a GMM, the time-dependent score $\nabla_{\xi_t} \log p_t(\xi_t)$ is available analytically [8]. As such, we simply use the analytic time-dependent score in the reverse process with `DiffBED`. For Rank ($N = 1000$) and Random, we also sample candidate designs from this GMM, to ensure that designs from these strategies are not penalised under the misalignment formulation above. BED, as in the main body of the paper, is unconstrained, gradient-based optimisation of designs. In this experiment, the observation space, y, is continuous. As such, we use the prior contrastive estimator (PCE) (Foster et al., 2020) for estimating the EIG, and it's gradients.

Inference: For all strategies, we employ the particle-based inference technique described in Appendix C, with $N = 10^6$ particles, as in the other experiments.

Evaluation: We compute the sequential prior contrastive estimation (sPCE) bound presented in Foster et al. (2021), at each iteration of the experimentation, using $L = 10^7$ samples. The model exhibits non-identifiability with respect to the ordering of the two sources, θ_1 and θ_2 during posterior sampling. As such, the correspondence between estimated and true sources may be swapped between samples. To address this when computing the L_2, we evaluate both orderings of each posterior sample and use the ordering that yields the lower error. We average over 25 roll-outs with different random seeds and present our results in Figure 8.

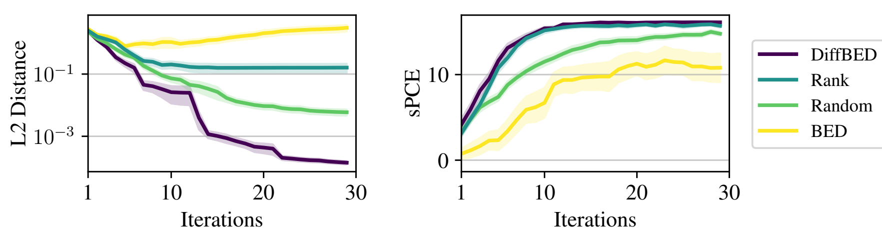


Figure 8: Average L_2 distance between 10^6 posterior samples and the ground truth locations summing over the two sources (left), and the EIG estimated using $sPCE$ at each experiment iteration (right). DiffBED is competitive with rank, needing only 1 sample of the guided diffusion to match Rank which ranks 1000 candidate designs. Traditional BED performs poorly due to its unconstrained optimization.

The BED baseline provides no mechanism for constraining the optimization trajectory to less misaligned regions, as captured by an implicit prior over designs. As such, when using standard BED, inference is pathologically hampered by making Bayesian updates on data sampled from p_{ood}. This results in BED performing **worse** than random design from samples from the design prior, as seen in the large L_2 discrepancy between the ground-truth locations and posterior samples in Figure 8. On the other hand, `DiffBED` and Rank are much more effective than random design, with `DiffBED` outperforming rank, highlighting that both these methods adhere to the design prior **and** effectively

[8] As our design prior is a GMM, the time-dependent score $\nabla_{\xi_t} \log p_t(\xi_t)$ under the variance-preserving diffusion process is available in closed form. Specifically, the forward process is $\xi_t = \sqrt{\alpha_t}\,\xi_0 + \sqrt{1-\alpha_t}\,\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I)$ and the marginal density at time t is a Gaussian-convolution of the GMM, $p_t(\xi_t) = \sum_{k=1}^{8} \pi_k \mathcal{N}(\xi_t;\, \sqrt{\alpha_t}\mu_k,\, \Sigma + (1-\alpha_t)I)$, which remains a mixture of Gaussians with inflated covariance. Hence the score has a trivial, analytic expression.

optimise the EIG. Note that Rank considers 1000 candidate designs, and `DiffBED` only diffuses a single guided diffusion design sample at each experiment iteration. We present an aggregation of the experimental designs, across rollouts, for each design strategy in Figure 9.

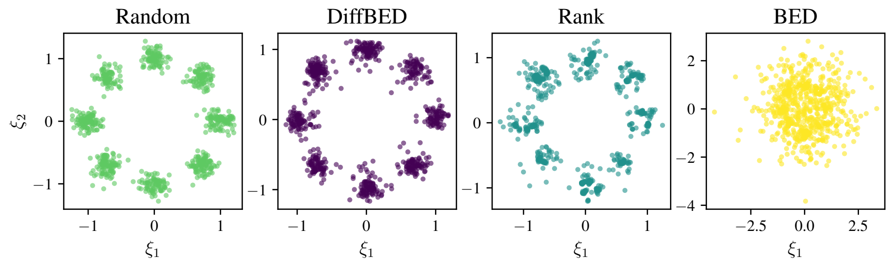


Figure 9: Designs aggregated across all iterations and experimental rollouts (25 seeds $\times$ 30 iterations = 750 designs) for 2D location finding. `DiffBED`, Random and Rank all leverage the implicit design prior in different ways to produce designs from regions where there is lower model misalignment. On the other hand, BED produces designs by *only* maximising the EIG, producing designs that lie in regions where there is greater misalignment.

E Additional Experiments

This section contains additional experiments and ablations not discussed in the main paper.

E.1 CelebA

Example DiffBED Designs: CelebA Figure 10 shows 100 example images generated by `DiffBED` and Figure 11 shows the same for Entropy. This serves as a qualitative evaluation of our designs. Both are similarly high-quality images as they leverage the same underlying diffusion model, although with different guidances. On the other hand, the designs produced by standard BED (Figure 12) are imperceptible from pure noise, despite their high EIG values.

Figure 10: Example `DiffBED` designs for the CelebA search problem.

Ablations In Figure 13 we present results across the two extra values of the likelihood temperature parameter $\tau^{-1} = \{25, 50\}$, in addition to the value of $\tau^{-1} = 10$ used for CelebA in the main body of the paper. Larger values of τ^{-1} indicate less noise in the observation process, i.e., more informative

Figure 11: Example *Entropy* designs for the CelebA search problem.

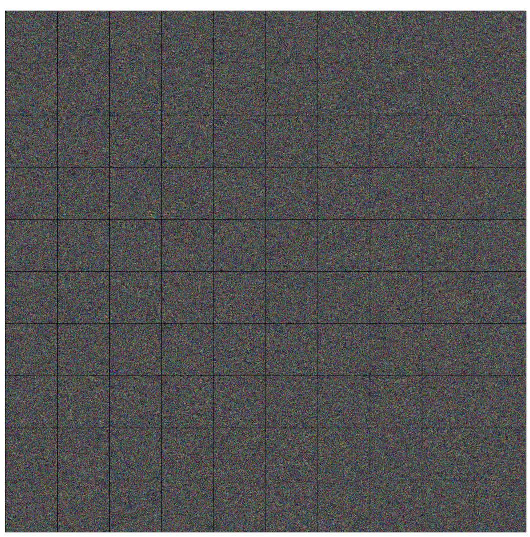
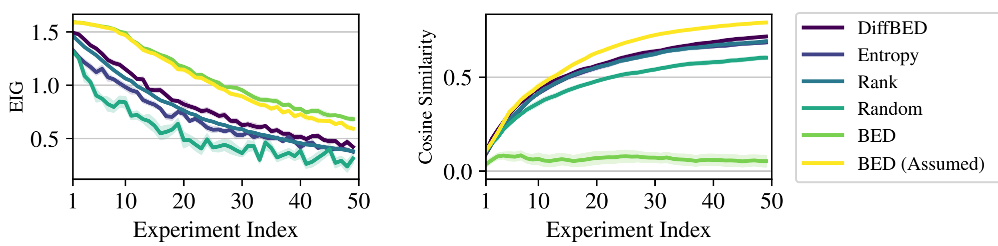


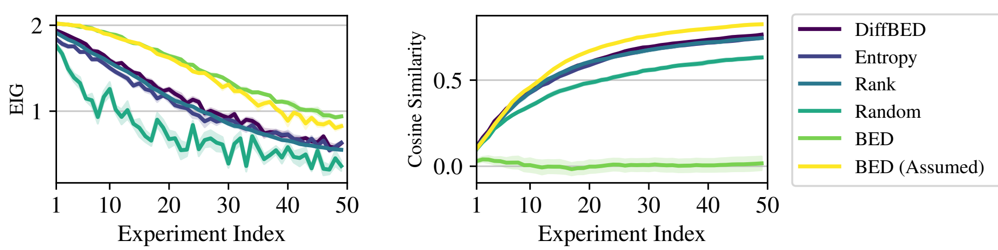
Figure 12: Example standard BED designs for the CelebA search problem.

outcomes, We see that larger τ^{-1} indeed leads to larger EIG values and a larger gap to the random baseline.

Here, we also present the performance of the naive BED baseline under data simulated from the assumed model which we call BED (assumed). That is, we sample $y \sim p(y \mid \theta, \xi)$ from the misspecified model likelihood, with no adjustments to the fact that the resulting designs may be meaningless. We see that BED (sssumed) achieves both high EIG scores and cosine similarities in this case. However, the designs produced by BED in this case are imperceptible from pure noise (Figure 12) and could not be used in a real-world experiment. As soon as we sample data y from a more realistic likelihood which returns uninformative data when the designs are pure noise, the cosine similarity of the naive approach (BED) drops to near zero, indicating a failure to determine the correct θ.



(a) $\tau^{-1} = 25$



(b) $\tau^{-1} = 50$

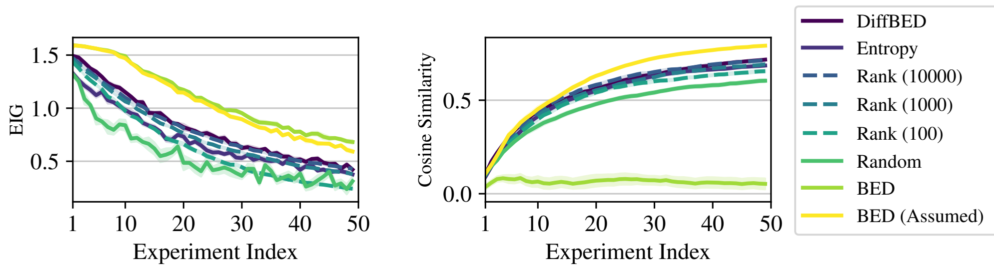
Figure 13: EIG and cosine similarity values for CelebA on the search problem, where outcomes are top-2 ranks of the images in the set of four design images. We now vary the value of τ used in the likelihood. Means are plotted with shaded regions indicating one standard error.

In a second ablation (at $\tau^{-1} = 25$) we study the effect of the number of the candidate designs considered in the *Rank* method. We plot the performance with $N = 100$, and $N = 10000$ candidate designs, alongside the $N = 1000$ set-up used in the main body of the paper. See Figure 14. Unsurprisingly, the performance of this baseline is a monotonic function of the pool size. However, increasing the pool size also comes with additional costs, especially when the pool is itself generated from a model, in which case $N = 10000$ would be significantly more time consuming than `DiffBED` while not exceeding its performance.

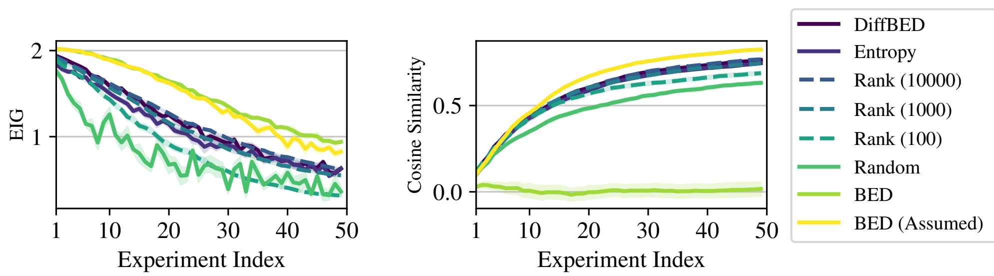
E.2 Zappos

Example DiffBED Designs: Zappos We present 100 example designs produced by `DiffBED` and standard BED on the Zappos discrete rating problem. See Figure 15 and 16.

Ablations We additionally present in Figure 17 results for $N = 10$ levels of discrete ratings, rather than $N = 5$ used in Section 6.3. Intuitively, we might expect that a more fine-grained rating would be more informative. Indeed, comparing against Figure 7, using $N = 10$ yields slightly higher cosine similarities.



(a) $\tau^{-1} = 25$



(b) $\tau^{-1} = 50$

Figure 14: EIG and cosine similarities as we the number of candidate designs consider by the *Rank* baseline in the CelebA search problem. Means are plotted with shaded regions indicating one standard error.

Figure 15: Example `DiffBED` designs for the Zappos discrete rating problem.

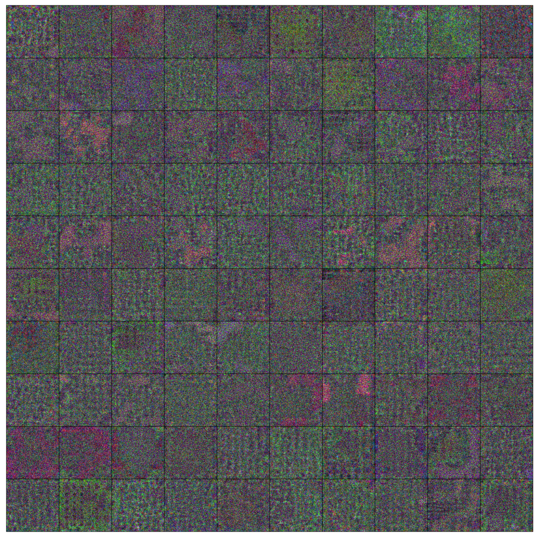


Figure 16: Example standard BED designs for the Zappos discrete rating problem.

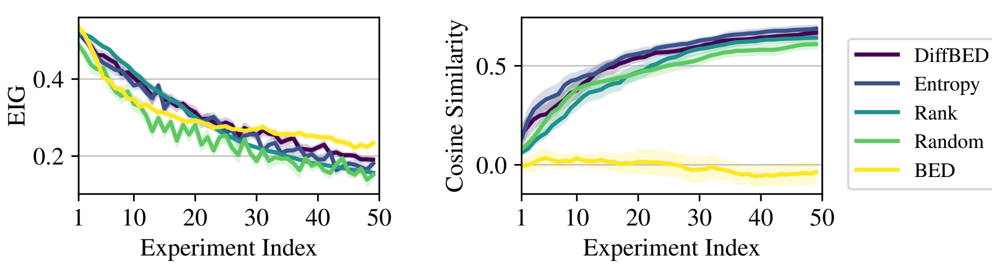


Figure 17: Quantitative results on the Zappos dataset, where designs are single images and outcomes are discrete ratings now on a scale from 1-10. Means are plotted with shaded regions indicating one standard error.