A FIXED-POINT ALGORITHM FOR CONSTRAINING PRE-TRAINED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

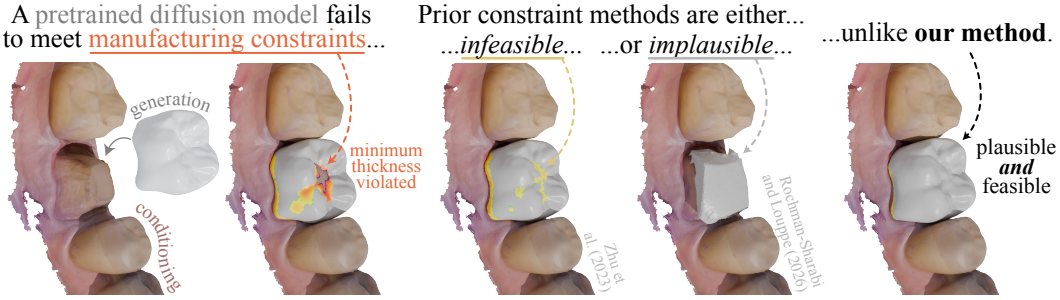


Figure 1: Inspired by patient-specific prosthetic design, we propose an algorithm (right) that imposes constraints (e.g., minimum manufacturable thickness) on the outputs of a pre-trained diffusion model (first two panels), while avoiding the artifacts of existing approaches (middle panels).

ABSTRACT

Three-dimensional diffusion models have emerged as a powerful tool to extract priors from large geometric datasets. However, exploiting these priors in generative high-precision applications (e.g., the design of medical prosthetics) requires a strategy for enforcing hard constraints on each generated shape. Unfortunately, we find that existing algorithms to impose these constraints on pre-trained diffusion models are inadequate for a large class of 3D generative tasks, as they either modify the denoised generation, risking escaping the data distribution, or rely on guiding the noisy reverse diffusion trajectory, very often failing to satisfy the constraints exactly. We introduce a *fixed-point* strategy for modifying the reverse diffusion trajectory without escaping the data distribution, improving the tradeoff between *plausibility* and *feasibility* of the generations. We showcase our method's performance on the 3D generative tasks that motivate its design, on which it outperforms every considered baseline, as well as on common 2D and multi-dimensional benchmarks.

1 INTRODUCTION

Diffusion models learn rich generative priors by progressively transforming noise into samples from a data distribution. They are emerging as a powerful tool in high-precision geometric design, where realistic outputs must also obey application-specific requirements. In settings such as medical-device design or bespoke manufacturing, these requirements are often formulated as hard geometric constraints that may vary from one generation to the next. This raises the following problem: given a trained diffusion model and some constraints, how can we generate shapes that satisfy the constraints (i.e., are *feasible*) while aligning as much as possible with the pretrained prior (i.e., are *plausible*).

Unfortunately, simply projecting an unconstrained output onto the constraint set often escapes the target data distribution, producing aberrant or meaningless generations. To avoid this, the best performing recent methods (e.g., Alkhouri et al. (2024); Rochman-Sharabi & Louppe (2026)) instead modify the entire denoising trajectory, steering each intermediate generation towards one that *will* satisfy the constraint once it is fully denoised. This is a valid strategy when a feasible configuration can be reached by a well-behaved continuous optimization starting from an unfeasible one. However, in our experiments these strategies often are insufficient for three-dimensional generative tasks with complex constraints, in which they consistently produce outputs that either do not fully satisfy the constraints or escape the data distribution (see Figure 1).

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

Metric	DPS	LGD-MC	CREPE	Trust	DSG	SITCOM	ΠGDM	PDM	DiffPIR	SNORE	PPR	ReSample	CPS	**Ours**
Kendall τ ↑	0.835	0.117	0.021	0.215	0.130	-0.008	0.127	-0.013	-0.002	0.003	0.022	0.216	0.328	**0.758**
Reordered%↓	81.7	97.6	98.3	97.3	97.2	98.2	97.3	98.5	98.5	98.4	98.0	96.5	97.0	**86.9**
Feasible%↑	8.4	14.6	21.4	35.6	66.9	95.2	97.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**

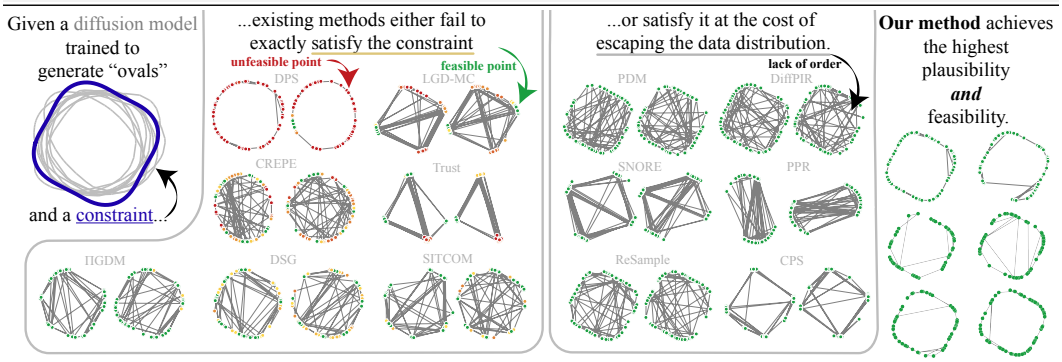


Figure 2: In this didactic example, a diffusion model was trained to generate *sorted*, two-dimensional point clouds of ovals, and a constraint (blue, left) is imposed in the generation. Some methods produce *unfeasible* outputs (points outside of the blue constraint, highligted in red), while others reorder points producing outputs considered *implausible* in our setting.

Inspired by the application of patient-specific prosthetic design (see Figure 1), we introduce an algorithm for imposing hard constraints on the generations of pre-trained diffusion models while maximizing plausibility. Like existing approaches, our method modifies the denoising trajectory to promote adherence to the constraint. Unlike prior work, however, we do not rely on smooth, gradient-based guidance to steer the generation; instead, we devise a *fixed-point algorithm* that enables us to search for feasible outputs within the pretrained prior. While we state the fixed-point algorithm in the model's denoised state, our derivation allows us to rewrite it as an iterative modification of each noisy element along the denoising trajectory.

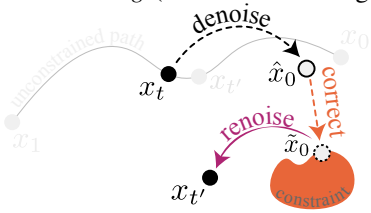
We evaluate our method on a diverse set of 3D generative tasks inspired by high-precision applications, including dental, cranial, spinal and vascular prosthetic design and generic point-cloud completion. Across these tasks, our method outperforms a dozen relevant prior approaches in balancing constraint satisfaction with output plausibility. Although developed for this class of problems, the method is general enough to be applied to other constrained diffusion settings. We therefore also evaluate it on standard image-based and other benchmarks, where it obtains competitive performance. By improving the ability of 3D diffusion models to satisfy strict constraints, our method brings 3D generative AI closer to safety-critical and highly sensitive applications.

2 Related Work

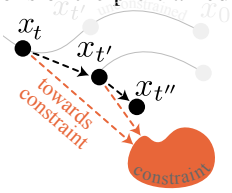
Diffusion models were originally conceived for and benchmarked on low-risk perceptual generation tasks, most prominently image generation (Ho et al., 2020; Rombach et al., 2022). As diffusion architectures have matured, they have increasingly been explored in applications where outputs represent physical objects, from medical implant design (Friedrich et al., 2023) and patient-specific digital twins (Kadry et al., 2024) to CAD geometry generation (Lee et al., 2025; Deng et al., 2025) and engineering design (Giannone et al., 2023). In these settings, perceptual realism alone is not sufficient: the generated geometry may also need to satisfy strict application-specific requirements, such as fitting a patient's anatomy or respecting manufacturing constraints (Kimmel et al., 2025).

A natural way to incorporate such requirements is to condition the generative model on them during training, along with constraint-specific losses. This approach is often unsuitable for our setting for two reasons. First, the constraint formulation may change during design, or new constraints may be introduced, leading to impractical repeated retraining or adaptation. Second, learned conditioning promotes constraint satisfaction, but does not guarantee it. While small violations may be acceptable in permissive applications, geometric design problems such as anatomical fitting or manufacturing clearance can require strict feasibility. A more promising approach is to modify the diffusion model's output distribution such as in TDS (Wu et al., 2023) and CREPE (He et al., 2026); however, these methods focus on distributional guarantees, not exact constraint satisfaction (see Figure 2). Thus, we consider strategies that directly impose the constraints at sampling time with frozen model weights.

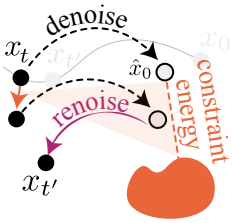
A standard (unconstrained) generative diffusion process relies on an iterative, sequential application of *denoising* and *renoising* steps, weighing the latter with a *noise scheduler* to progressively reduce noise. Given this structure, existing strategies to constraint this process differ by the moment in the sequence in which they intervene: either *after denoising* or *before denoising* (or before renoising and after renoising, respectively).



Interventions *after* denoising. Since our goal is for the final denoised sample to satisfy the constraint, a natural strategy is to add a correction layer to the denoising operation given by the pre-trained diffusion model. At each step along the diffusion trajectory x_t (see inset), these methods thus first follow the frozen denoiser, followed by a *correction* step that promotes constraint satisfaction, and a *renoising* step that returns the corrected estimate to the reverse diffusion trajectory at $x_{t'}$. If the constraint is linear (e.g., image inpainting), the correction can be performed analytically (Wang et al., 2022). If it is not, as in the anatomical and manufacturing constraints that motivate this paper, more elaborate optimization schemes and "softer" corrections are used. For example, DiffPIR (Zhu et al., 2023) solves a proximal subproblem that pulls the clean estimate toward the data while penalizing the distance moved, while CPS (Narasimhan et al., 2026) uses an annealed penalty that enforces constraint satisfaction more strongly in the later steps of the diffusion; and ReSample (Song et al., 2024) searches for a latent whose decoded output satisfies the observation, then stochastically maps that corrected latent back to the current noise level. SNORE Renaud et al. (2024) uses gradient steps to pull the current estimate toward the observed data, while a denoiser applied to a noisy copy of that estimate provides a regularizing correction. The main drawback of this type of methods, which we validate experimentally throughout this text (e.g., Figure 2), is in its plausibility, as this process can easily produce diffusion trajectories x_t that escape the marginal distributions of the pre-trained model, leading to out-of-distribution generations.



Interventions *before* denoising. Rather than editing the clean estimate and then returning it to the diffusion trajectory, another family acts directly on the current noisy state. The goal is not to make the noisy state feasible, but to modify it so that its denoised prediction better satisfies the constraint. This can be done by "nudging" each noisy estimate away from its unconstrained trajectory and towards eventual constraint satisfaction (see inset). This nudge may ignore the existence of a denoiser altogether, projecting each noisy estimate onto the constraint (e.g., PDM (Christopher et al., 2024)), or use an iterative loop that evaluates guidance on the denoised prediction (e.g., Trust Sampling (Huang et al., 2024)), or may rely on a statistical analysis that interprets the constraint itself as a noisy observation of the final (unknown) sample. This observation induces a posterior distrubtion over denoised samples given the "observed" constraint, which is intractable but can be concentrated on its mean (e.g., DPS (Chung et al., 2022)), estimated through Monte-Carlo (LDG-MC (Song et al., 2023b)), or approximated using different distributional assumptions (e.g., a Gaussian with an annealed variance (Song et al., 2023a) or a distribution of directions on a sphere (Yang et al., 2024)).



Closest to our method, a recent line of work proposes modifying the denoising trajectory by solving an energy minimization problem that accounts for the need of the eventual denoised sample to satisfy the constraint (see inset). This strategy relies on the existence of an energy measuring adherence to the constraint; in most cases, this can be as simple as Euclidean distance to the constraint set. At each reverse diffusion step, the energy is minimized through a few iterations of gradient descent, repeating optimization and renoising at the same noise level (as in the case of PPR (Rochman-Sharabi & Louppe, 2026)) or regularizing the optimization toward the initial noisy state and then renoising the resulting clean estimate (as in the case of SITCOM (Alkhouri et al., 2024)). After optimization, each sample is denoised again and renoised to return to the reverse diffusion trajectory. Like the approaches in the previous paragraph, this family relies on some regularity about the diffusion process and the constraint set, encoded through the ability of the algorithm to find a feasible-yet-still-plausible trajectory via small iterative displacements from the unconstrained trajectory. Experimentally, we find these gradient-based or regularization-based strategies to be insufficient for the complex geometric tasks considered in this paper, motivating our design of a *fixed-point* optimization to modify the reverse diffusion sampling trajectory.

3 METHOD

Given an known *noise* distribution p_1 and an unknown *clean* data distribution p_0, we assume a diffusion model has been pre-trained to progressively *denoise* samples from one distribution $x_1 \sim p_1$ into the other $x_0 \sim p_0$, producing a *trajectory* of elements $x_t \in \mathbb{R}^n$. The diffusion model's training promotes that the denoised outputs x_0 indeed belong to the data distribution p_0; in other words, that they are *plausible* under the priors contained in the training data.

In particular, let us assume the pre-trained model is a variance-preserving diffusion one (Ho et al., 2020) with cumulative signal schedule $\{\bar{\alpha}_t\}_{t=1}^T$, such that the noisy state is the marginal of the forward process

$$\mathbf{x}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 \;+\; \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,, \tag{1}$$

and the trained network $\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)$ induces a denoising map

$$D_t(\mathbf{x}_t) \;=\; \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}, \tag{2}$$

whose output is the Tweedie clean estimate

$$\hat{\mathbf{x}}_0^{(t)} \;:=\; D_t(\mathbf{x}_t) \;\approx\; \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]\,. \tag{3}$$

Now, let us assume the existence of a *constraint set* $\mathcal{C} \subseteq \mathbb{R}^n$ that the generated sample $\hat{\mathbf{x}}_0$ must belong to (we say $\hat{x}_0$ is *feasible* if $\hat{\mathbf{x}}_0 \in \mathcal{C}$). The exact form of $\mathcal{C}$ is unknown during training, and may change from one generation to the next. In our motivating example, the diffusion model has been trained on a captured anatomical dataset at the population level (e.g., of human teeth), and the constraints are given by a specific patient's anatomy (e.g., a generated dental crown cannot overlap with a patient's neighboring or opposing teeth). Beyond this specific application, in what follows, we derive a general algorithm to steer the denoising trajectory $\mathbf{x}_t$ towards a clean estimate $\hat{\mathbf{x}}_0$ that is both *plausible* given the data distribution and *feasible* under the constraint set.

3.1 ENFORCING CONSTRAINTS THROUGH FIXED-POINT ITERATIONS

Let us begin by defining a *projection* operator $T : \mathbb{R}^n \to \mathcal{C}$ which converts any sample x into an element of the nonempty and closed constraint set $T(x) \in \mathcal{C}$. Our derivation will be agnostic to the specific form of this operator; for now, one may imagine it to be a closest-point projection $T(x) = \arg\min_{c \in \mathcal{C}} \|x - c\|$; for nonconvex sets, the nearest point need not be unique.

It should be noted that, strictly speaking, we desire only that the *final* denoised estimate $\hat{\mathbf{x}}_0 = \hat{\mathbf{x}}_0^{(0)}$ satisfies the constraint; as there is no need for the intermediate states $\mathbf{x}_t$ or even their intermediate denoised estimates $\mathbf{x}_0^{(t)}$ to do so. One could, therefore, use the model's standard unconstrained denoising to obtain a denoised estimate $\hat{\mathbf{x}}_0$, followed by a projection $\hat{\mathbf{x}}_0 \to T(\hat{\mathbf{x}}_0)$. Depending on the geometry of constraint set $\mathcal{C}$, however, this will often produce a final estimate $T(\hat{\mathbf{x}}_0)$ belonging to a region of very low likelihood under the data distribution p_0. In other words, this strategy would produce a feasible estimate, but not a plausible one.

A better idea is to modify the entire trajectory $\mathbf{x}_t$, using the information contained in each intermediate denoised estimate $\mathbf{x}_0^{(t)}$ to steer each $\mathbf{x}_t$ towards constraint satisfaction. For example, one could, at each denoising timestep t, backpropagate through the denoiser network and use gradient descent to move $\mathbf{x}_t$ in the direction that reduces the distance from its denoised estimate $\mathbf{x}_0^{(t)}$ to the constraint set $\mathcal{C}$ (Rochman-Sharabi & Louppe, 2026; Alkhouri et al., 2024). This strategy is valid in some generative tasks; however, it is highly sensitive to local minima. Even when it succesfully escapes these minima, the decision of how much to displace $\mathbf{x}_t$ requires one to thread the needle between not satisfying the constraint (if one does not change $\mathbf{x}_t$ enough, producing infeasible generations) and premature satisfaction (leading to implausible outputs).

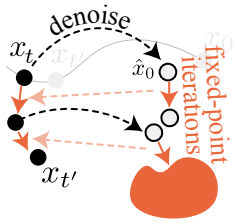
Our method is inspired by these prior approaches, but instead notes that if T is a true projection operator, then an intermediate denoised estimate $\hat{\mathbf{x}}_0^{(t)}$ belongs to $\mathcal{C}$ if and only if

$$T\left(\hat{\mathbf{x}}_0^{(t)}\right) = \hat{\mathbf{x}}_0^{(t)}\,. \tag{4}$$

To move $\hat{\mathbf{x}}_0^{(t)}$ towards constraint satisfaction, one could then follow a fixed-point scheme of the form

$$\hat{\mathbf{x}}_0^{(t,k+1)} = T\left(\hat{\mathbf{x}}_0^{(t,k)}\right)\,, \tag{5}$$

converging once the difference between succesive iterates approaches zero. Of course, $\hat{\mathbf{x}}_0^{(t)}$ is not a variable on its own, but a function of x_t, meaning that the iterations above must actually be performed in the space of intermediate noisy elements $\mathbf{x}_t^k$. Making this dependence explicit, one can write the fixed-point iteration as



$$D_t\left(\mathbf{x}_t^{k+1}\right) = T\left(D_t\left(\mathbf{x}_t^k\right)\right) . \tag{6}$$

This expression is no longer a trivial fixed-point iteration, but an equation on $\mathbf{x}_t^{k+1}$ whose solution requires knowledge about the specific denoising process. To solve it, let us parametrize $\mathbf{x}_t^{k+1}$ through its displacement from $\mathbf{x}_t^k$, weighted by the size of the noise scheduler

$$\mathbf{x}_t^{k+1} = \mathbf{x}_t^k + \sqrt{\bar{\alpha}_t}\, u^{k+1} , \tag{7}$$

turning Eq. 6 into an equation that we must solve for u^{k+1}

$$D_t\left(\mathbf{x}_t^k + \sqrt{\bar{\alpha}_t}\, u^{k+1}\right) - D_t(\mathbf{x}_t^k) = T\left(D_t\left(\mathbf{x}_t^k\right)\right) - D_t\left(\mathbf{x}_t^k\right) . \tag{8}$$

Fortunately, the following lemma, whose proof relies on Eq. 2 (see Supplemental A.1) allows us to relate displacements in the denoised space to displacements in the noisy space:

Lemma 1 (Response decomposition)**.** *For any* $\mathbf{x}_t$*,* $\mathbf{u}$*, and* t *with* $0 < \bar{\alpha}_t < 1$*,*

$$D_t(\mathbf{x}_t + \sqrt{\bar{\alpha}_t}\mathbf{u}) - D_t(\mathbf{x}_t) = \mathbf{u} - \frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\left[\epsilon_\theta(\mathbf{x}_t + \sqrt{\bar{\alpha}_t}\mathbf{u}, t) - \epsilon_\theta(\mathbf{x}_t, t)\right] . \tag{9}$$

Applied to the left-hand side of Equation 8, the above lemma expresses the change in the denoised estimate as the clean-space displacement $\mathbf{u}^{k+1}$ minus a scaled difference in the noise prediction. If the network has a locally Lipschitz Jacobian, a Taylor expansion gives

$$D_t(\mathbf{x}_t + \sqrt{\bar{\alpha}_t}\mathbf{u}) - D_t(\mathbf{x}_t) = \left[I - \sqrt{1-\bar{\alpha}_t} J_{\epsilon_\theta}(\mathbf{x}_t, t)\right]\mathbf{u} + O(\|\mathbf{u}\|^2). \tag{10}$$

The matrix multiplying $\mathbf{u}$ describes how the denoiser responds locally to the correction. In the ideal case where $D_t(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_0 \mid \mathbf{x}_t]$, the relation between the denoiser's Jacobian and the posterior covariance (Manor & Michaeli, 2024) gives

$$I - \sqrt{1-\bar{\alpha}_t} J_{\epsilon_\theta}(\mathbf{x}_t, t) = \sqrt{\bar{\alpha}_t} J_{D_t}(\mathbf{x}_t) = \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\,\mathrm{Cov}(\mathbf{x}_0 \mid \mathbf{x}_t). \tag{11}$$

Thus, this response depends on the uncertainty about the clean sample given the current noisy state. Solving the linearized form of Equation 8 would require choosing $\mathbf{u}^{k+1}$ so that multiplication by this response matrix yields the desired clean-space correction. Also, reducing the constraint violation does not require realizing the prescribed correction exactly. If the discrepancy between the actual denoised displacement and the prescribed correction is smaller than the current distance to the constraint set, the updated estimate is closer to feasibility. This motivates the explicit iterative scheme below, which applies the projection residual directly and recomputes it after each denoising operation

$$\mathbf{u}^{k+1} = T\left(D_t\left(\mathbf{x}_t^k\right)\right) - D_t\left(\mathbf{x}_t^k\right) , \tag{12}$$

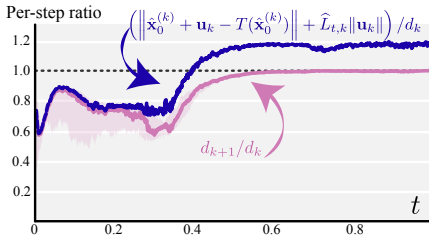
$$\mathbf{x}_t^{k+1} = \mathbf{x}_t^k + \sqrt{\bar{\alpha}_t}\,\mathbf{u}^{k+1} . \tag{13}$$

Intuitively, this fixed-point scheme performs corrections on the intermediate noisy sample $\mathbf{x}_t$ until its denoised estimate $\tilde{\mathbf{x}}_0^{(t)} = \lim_k \hat{\mathbf{x}}_0^{(t)}\left(\mathbf{x}_t^k\right)$ satisfies the constraint (i.e., until it is feasible) while relying on repeated applications of the denoiser to encourage the sample to inherit the prior contained in the pre-trained model (i.e., remains plausible). To avoid computing an asymptotic limit, in practice, we compute a fixed number of iterations followed by a final projection $\tilde{\mathbf{x}}_0^{(t)} \leftarrow T(\tilde{\mathbf{x}}_0^{(t)})$.

The ability of our iterative scheme to converge can be established by the discrepancy between the prescribed correction and the actual displacement, which we quantify in terms of the *relative defect*

$$\widehat{L}_{t,k} \;=\; \frac{\left\|\hat{\mathbf{x}}_0^{(k+1)} - \hat{\mathbf{x}}_0^{(k)} - \mathbf{u}^{k+1}\right\|}{\|\mathbf{u}^{k+1}\|} . \tag{14}$$

The relation between Eqs 12 and 13 and our original fixed-point iterative problem in Equation 6 is formalized in the following result.



Theorem 1. *If the relative defects defined in Equation 14 satisfy $\widehat{L}_{t,k} \leq L < 1$ for every inner iteration, then iterations defined in Eqs. 12 and 13 satisfy $d_{k+1} \leq L d_k$, where d_k quantifies the sample's feasibility by measuring the distance between $\hat{\mathbf{x}}_0^{(k)}$ and its Euclidean nearest-point projection $T(\hat{\mathbf{x}}_0^{(k)})$. Moreover, $\mathbf{x}_t^k$ converges to a limit whose denoised estimate belongs to C. The proof is provided in Supplemental A.2.*

Applying the triangle inequality to Equation 13, we have

$$d_{k+1} \leq \left\|\hat{\mathbf{x}}_0^{(k)} + \mathbf{u}^{k+1} - T(\hat{\mathbf{x}}_0^{(k)})\right\| + \widehat{L}_{t,k}\|\mathbf{u}^{k+1}\|. \tag{15}$$

Intuitively, this inequality tells us that the performance of the fixed-point method parametrized by u^{k+1} is bounded by the performance of the original fixed-point algorithm in Equation 6, plus a term that quantifies the quality of the approximation carried out in Equation 10. The method will converge if the quotient d_{k+1}/d_k is consistently below one; an example of this (for the didactic generation in Figure 2) is shown in the inset.

Damping A surprising fact about the fixed-point algorithm presented in this section is that its derivation does not rely on T being a *true* algebraic projection operator (i.e., it does not necessitate that $T(x) \in \mathcal{C}$ for all $x \in \mathbb{R}^n$). Indeed, strictly speaking, all that's needed is the fixed-point property in Equation 6; i.e., that $T(x) = x \iff x \in \mathcal{C}$. While the traditional closest point projection $T(x) = \arg\min_{c \in \mathcal{C}} \|x - c\|$ satisfies this property; in practice, we find that it harm robustness and create overly noisy sampling trajectories. To avoid this, we instead use a *damped* projection $T_d(x) = \frac{1}{2}(T(x) + x)$, which improves performance without sacrificing theoretical properties enumerated in this section, see details in Supplemental A.2 and A.3.

3.2 Reverse diffusion step

Once the corrected $\tilde{\mathbf{x}}_0^{(t)}$ has been obtained, one *could* perform the next reverse diffusion step for timestep t' as

$$\mathbf{x}_{t'} = \sqrt{\bar{\alpha}_{t'}}\,\tilde{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t'}}\mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,. \tag{16}$$

Doing so, however, would make every constraint correction also reset the stochastic direction of the current trajectory. Instead, we cache the noise residual predicted by the denoiser before the fixed-point corrections:

$$\hat{\boldsymbol{\varepsilon}} = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\;\hat{\mathbf{x}}_0^{(t)}\left(\mathbf{x}_t^{k=0}\right)}{\sqrt{1 - \bar{\alpha}_t}}\,, \tag{17}$$

and combine it with fresh noise to obtain our actual next reverse diffusion step

$$\mathbf{x}_{t'} = \sqrt{\bar{\alpha}_{t'}}\,\tilde{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t'}}\left(\sqrt{1 - \eta^2}\,\hat{\boldsymbol{\varepsilon}} + \eta\,\mathbf{z}\right), \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{18}$$

where η is a hyperparameter that controls the stochasticity of the reverse step: $\sqrt{1 - \eta^2}$ determines how much of $\hat{\boldsymbol{\varepsilon}}$ is retained, while η determines how much fresh Gaussian noise is introduced. The optimal value of η varies with each application; interestingly, we find that it can be estimated once through properties of the *unconstrained*, pre-trained diffusion alone, without depending on or relying on properties of the imposed constraints. We elaborate on this estimation in Supplemental A.5.

Thus, our final method is deceptively simple: at each diffusion timestep t with current noisy estimate x_t we first use the pre-trained denoiser to obtain an unconstrained $\hat{\mathbf{x}}_0^{(t)}$ and cache the initial predicted noise residual $\hat{\boldsymbol{\varepsilon}}$ using Eq 17. Then, we perform the fixed-point iterations stated in Eqs 12 and 13 to convert $\hat{\mathbf{x}}_0^{(t)}$ into a $\tilde{\mathbf{x}}_0^{(t)}$ that satisfies the constraint, and use Eq 18 to move backwards towards a less noisy estimate $\mathbf{x}_{t'}$. We dedicate the rest of the paper to showcasing how this method can outperform existing strategies on most of our motivating applications.

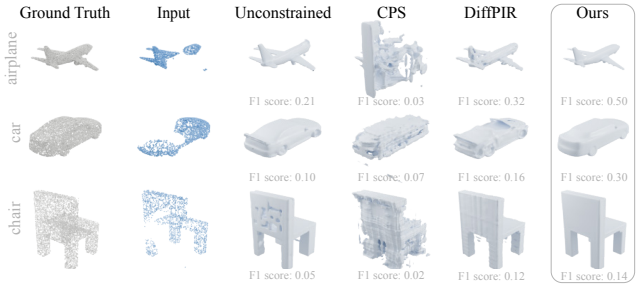


Figure 3: A pre-trained diffusion model is constrained to agree with an input point cloud. Our method completes the geometry, staying faithful to it while avoiding the artifacts of the best-performing baselines.

Table 1: Our method obtains the best feasibility ("fit") while excelling in most plausibility proxies. Other baselines obtain higher "coverage" by producing aberrations spanning large volumetric regions (see figure).

Metric	Prior	CPS	DiffPIR	**Ours**
fit↓	0.0164	0.0170	0.0093	**0.0043**
<.01↑	0.41	0.35	0.59	**0.86**
DCD↓	0.760	0.864	0.732	**0.668**
F1↑	0.119	0.050	0.157	**0.269**
cov↓	0.069	**0.051**	0.071	0.099
parts↓	5.5	58.8	6.5	**2.2**

4 Experiments & Applications

Didactic example. Our method is motivated by geometric problems in which one generates a point cloud that must adhere to some constraint. We explore this type of problem in the didactic example of Figure 2. In it, a DDPM diffusion model is trained to produce 2D point clouds with 64 points in the shape of a distorted oval. Additionally, we draw a rounded square and constraint all generations to lay on it (feasibility is measured in terms of the distance to it). On the other hand, for a given generation seed, a *plausible* constrained output will match the unconstrained generation as much as possible, but for a small displacement in the normal direction to acommodate the constraint. Thus, one can measure plausibility by quantifying the extent to which the ordering of the points in the unconstrained generation is preserved in the constrained one (visualized in Figure 2 as a lack of grey lines crossing the origin). The quantitative evaluation of this experiment is at the top of Figure 2, which exemplifies the dychotomy common to constrained generative modeling: while some methods (in the table, shaded grey; e.g., DPS (Chung et al., 2022)) produce highly plausible results, they do so at a heavy cost to feasibility. Among the methods producing feasible outputs, ours obtains by far the largest Kendall τ score and reorders the smallest percentage of points.

Point cloud completion. Inspired by the success of our method on the didactic task described above, we benchmark it on a more common 3D generative task: completing a point cloud from a partial observation. We use the diffusion model proposed in 3DShape2Vec Zhang et al. (2023), in which shapes from five ShapeNet Chang et al. (2015) categories are encoded and generation is performed on their 512×8-size latent space, from which a continuous occupancy field can be decoded. Given a partial observation, we constrain the generation so that the value of the decoded occupancy is near the decision boundary (i.e., between -1 and 1) at the voxels contining input points. Our projection T is performed through gradient steps on the latent space until the constraint is satisfied. Our results are presented in Table 1, where feasibility is measured in terms of matching the input point cloud and plausibility is measured as the overlap with the groundtruth and the number of separate connected components. Experimentally, we find that our method matches the input data best (see "fit" column) while also producing the best F1 score and the lowest number of spurious components; in other words, it outputs feasible generations that are also plausible. A gallery of results is presented in Figure 3, and more experimental details are provided in Section A.6.3.

4.1 Patient-specific prosthetic design

Our method is motivated by the emerging problem of using pre-trained diffusion models for the design of dental crowns conditioned on a patient's dental scan. We show a result of this task in Figure 1; however, we stress that we do so only for motivation, as the lack of a publicly available dental crown generation dataset preclude us from using it as a true benchmarked result. Because of this, we instead benchmark our method on three existing open-source anatomical tasks: reconstructing missing aortic segments for graft design, vertebra reconstruction and cranial implant generation.

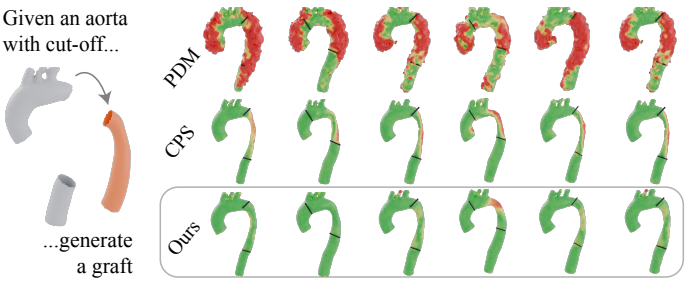


Figure 4: In a fictional aortic graft design task, our method produces geometries that align best with the patient's remaining anatomy and best match the unseen, groundtruth geometry.

Table 2: Our method quantitatively outperforms even the best-performing baselines while remaining feasible. See Table 7 for full results, including all 17 considered baselines.

Method	Feas.↑	Chamfer↓	F-score↑
Trust	0.00	4.16	0.538
PDM	0.47	7.25	0.479
CPS	0.78	4.38	0.711
Ours	**1.00**	**3.10**	**0.777**

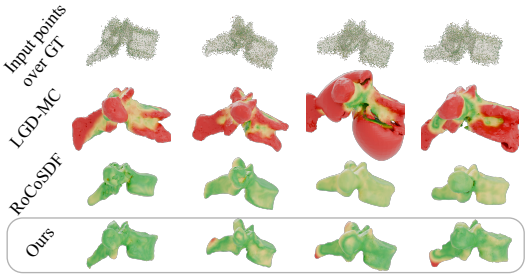


Figure 5: Our algorithm excels at completing vertebral geometries from noisy observations, in this example inspired by spinal surgery.

Table 3: Among methods that produce feasible outputs, our method outputs the lowest Chamfer distance. The few that outperform ours in "Coverage" do so by producing large artifacts (e.g., LGD-MC, see figure, second row).

Method	Feas.↑	Cov.↑	Chamfer↓
DiffPIR	0.19	0.849	1.64
RoCoSDF	0.58	0.925	1.31
CPS	**1.00**	0.996	4.95
LGD-MC	**1.00**	**0.999**	4.35
Ours	**1.00**	0.984	**1.16**

Aortic grafts Aortic pathologies such as an aneurysm or dissection disrupt the normal blood flow and often demand a replacement of a damaged segment with a graft (Zhai et al., 2026). We study this setting using SynthAorta (Bošnjak et al., 2025), a collection of 30,000 synthetic physiological aortas with three supra-aortic branches. We extract surface point clouds from the aorta's meshes and train an unconditional PVCNN Friedrich et al. (2023) DDPM to generate healthy aortas as point clouds. For testing, we remove a segment of the main aortic vessel while retaining the supra-aortic branches as landmarks. In a similar setting to the point cloud completion task from Figure 3, we then generate full aortic geometries constrained to matching the supra-aortic anatomy. For this, we employ an operator T which includes a displacement along the normal direction (see Section A.6.2 for details). Results in Figure 4 and Table 2 (which include only the best-performing baselines, see Table 7 for full results) show that that our method is the only evaluated sampler to be exactly feasible while generating a plausible completed aorta.

Vertebra reconstruction Some medical interventions like spinal surgery rely on a detailed, accurate map of the patient's vertebra to avoid severing critical nerves with catastrophic consequences. Ultrasounds are a way to capture this map without radiation; however, tracking errors and misalignments mean the produced vertebral point clouds are noisy and may even miss entire regions. This has motivated others (most recently, RocoSDF (Chen et al., 2024)) to work on the task of producing clean and accurate vertebral geometries from imperfect observations. We train a DDPM (Ho et al., 2020) on 1,606 vertebrae segmented from the VerSe dataset (Sekuboyina et al., 2021) and generate 8,192-point clouds conditioned on the vertebral level. Following the methodology proposed by Chen et al. (2024), we simulate observations that lie ≈ 1.0mm from the true bone surface, and impose this margin of error as a constraint during generation (see Section A.6.2 for more details). As shown in Figure 5 and Table 3, which report results for 24 held-out subjects, our method is feasible on all samples and has the lowest Chamfer distance to the patient's real vertebra. The only metric in which our method does not outperform all baselines is *coverage*; however, it should be noted that this is due to some methods producing artifact-laden, noisy geometries that happen to cover large regions of space (see LGD-MC, second row in Figure 4). Indeed, our method even outperforms RoCoSDF, an algorithm built specifically for this task and which gives a close plausibility but meets the coverage criterion on only 58% of cases.

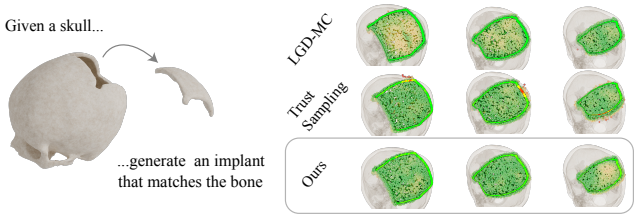


Figure 6: In a preexisting benchmark, one aims to fill in a missing region of a skull. Our method best adheres to the rim of the missing region, while at the same time being closest to the groundtruth skull's shape.

Table 4: Our outputs match the rim of the skull's missing region ("Feas."), while minimizing distance to the groundtruth. See Table 8 for full results with all considered baselines.

Method	Feas.↑	Chamfer↓	F-score↑
Trust	0.76	3.21	0.828
DiffPIR	0.90	3.05	0.848
CPS	**1.00**	3.35	0.802
DSG	**1.00**	3.23	0.826
LGD-MC	**1.00**	3.12	0.820
Ours	**1.00**	**2.80**	**0.888**

Cranial implants. For reasons from accidents to diseases, a section of a patient's skull may need to be removed surgically and replaced with an implant. This is the task considered by the *SkullFix* dataset Kodym et al. (2021), generated synthetically as part of the AutoImplant challenge Li et al. (2021). From each binary volume, we extract two point clouds (sizes $27,648$ and $3,072$) representing the skull and the implant. We train a PVCNN model Friedrich et al. (2023) to generate implant shapes conditioned on skull geometry. Importantly, we train the model for two thirds of the official implementation's epochs to provide a more challenging task to both our method and the baselines. Once the model is trained, we generate a new implant for each patient's anatomy, constraining it to adhere to the boundary (the rim) of the missing region in the cranium. Our projection operator T pushes the k nearest points of the implant towards the uncovered points of the rim. Our results for this experiment are presented in Table 4 and exemplified in Figure 6. Plausibility is measured in terms of the Chamfer distance and F-score with respect to the ground truth implant, and we call a sample feasible when its rim-gap does not exceed that of the patient's own implant. Most baselines fail to produce feasible outputs, while the few that do generate geometries that do not match the anatomy of the original cranium as closely as our method's.

4.2 Other Benchmarks

While we designed our method for 3D generative tasks like the ones in the previous sections, we place our work in the broader context of constrained diffusion models by testing on other commonly used benchmarks. In particular, we measure our method's accuracy on the same datasets, models and metrics considered by Trust Sampling (Huang et al., 2024), the non-linear image-based tasks developed by (Tran et al., 2021) analogous to (Chung et al., 2022) and the synthetic Data2D example proposed by (Rochman-Sharabi & Louppe, 2026). A full review of results is presented in Section A.6.4. In summary, across all tasks, we find that our method often produces outputs that perfectly or close-to-perfectly adhere to the constraints, with plausibility metrics that are competitive with prior work. Our algorithm shows strong performance on tasks like constrained motion generation (see Figure 8 and Table 19) and , while it is outperformed on image-based tasks by purpose-based methods designed specifically for them (e.g., SITCOM (Alkhouri et al., 2024)).

5 Limitations and Conclusions

Inspired by the application of patient-prosthetic design, we have presented a novel constrained diffusion generation method. Across an array of 3D geometric tasks, our method outperforms prior work along the plausibility-feasibility tradeoff; however, it does not erase the tradeoff: the fundamental contradiction between these two remains the main theoretical challenge in the field of constrained generation. We have extensively explored and reported our method's performance; at the same time, our convergence result in Theorem 1 is only valid under an assumption that we have only validated experimentally. To obtain this improvement, our method enforces constraints through fixed-point iterations, which need a *projection* operator T that may be difficult to obtain in more complex scenarios. Despite these limitations, it is our hope that work like ours helps bridge the gap between generative modeling and high-precision applications like dentistry and medicine. Implementing our work in clinical settings (like dental prosthetics) will require modeling the more nuanced preferences of each physician as well as handling competition between mutually exclusive constraints.

REFERENCES

Ismail Alkhouri, Shijun Liang, Cheng-Han Huang, Jimmy Dai, Qing Qu, Saiprasad Ravishankar, and Rongrong Wang. Sitcom: Step-wise triple-consistent diffusion sampling for inverse problems. *arXiv preprint arXiv:2410.04479*, 2024.

Domagoj Bošnjak, Gian Marco Melito, Richard Schussnig, Katrin Ellermann, and Thomas-Peter Fries. Synthaorta: A 3d mesh dataset of parametrized physiological healthy aortas. *IEEE Transactions on Medical Imaging*, 2025.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Hongbo Chen, Yuchong Gao, Shuhang Zhang, Jiangjie Wu, Yuexin Ma, and Rui Zheng. Rocosdf: row-column scanned neural signed distance fields for freehand 3d ultrasound imaging shape reconstruction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 721–731. Springer, 2024.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Liyuan Deng, Yunpeng Bai, Yongkang Dai, Xiaoshui Huang, Hongping Gan, Dongshuo Huang, Hao Jiacheng, and Yilei Shi. Mamtiff-cad: Multi-scale latent diffusion with mamba+ for complex parametric sequence. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10517–10526. IEEE, 2025.

Paul Friedrich, Julia Wolleb, Florentin Bieder, Florian M. Thieringer, and Philippe C. Cattin. Point cloud diffusion models for automatic implant generation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pp. 112–122, 2023.

Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation. *Advances in neural information processing systems*, 36:51830–51861, 2023.

Jiajun He, Paul Jeha, Peter Potaptchik, Leo Zhang, José Miguel Hernández Lobato, Yuanqi Du, Saifuddin Syed, and Francisco Vargas. Crepe: Controlling diffusion with replica exchange. In *International Conference on Learning Representations*, volume 2026, pp. 99464–99494, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

William Huang, Yifeng Jiang, Tom Van Wouwe, and C Karen Liu. Constrained diffusion with trust sampling. *Advances in Neural Information Processing Systems*, 37:93849–93873, 2024.

Karim Kadry, Shreya Gupta, Farhad R Nezami, and Elazer R Edelman. Probing the limits and capabilities of diffusion models for the anatomic editing of digital twins. *npj Digital Medicine*, 7 (1):354, 2024.

MA Kimmel, Mueed Ur Rehman, Yonatan Bisk, and Gary K Fedder. Position: You can't manufacture a nerf. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

Oldřich Kodym, Jianning Li, Antonio Pepe, Christina Gsaxner, Sasank Chilamkurthy, Jan Egger, and Michal Španěl. Skullbreak/skullfix–dataset for automatic cranial implant design and a benchmark for volumetric shape learning tasks. *Data in Brief*, 35:106902, 2021.

Mingi Lee, Dongsu Zhang, Clément Jambon, and Young Min Kim. Brepdiff: Single-stage b-rep diffusion model. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, SIGGRAPH Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400715402. doi: 10.1145/3721238.3730698. URL https://doi.org/10.1145/3721238.3730698.

Jianning Li, Pedro Pimentel, Angelika Szengel, Moritz Ehlke, Hans Lamecker, Stefan Zachow, Laura Estacio, Christian Doenitz, Heiko Ramm, Haochen Shi, et al. Autoimplant 2020-first miccai challenge on automatic cranial implant design. *IEEE transactions on medical imaging*, 40(9): 2329–2342, 2021.

Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *International Conference on Learning Representations*, volume 2024, pp. 49233–49263, 2024.

Sai Shankar Narasimhan, Shubhankar Agarwal, Litu Rout, Sanjay Shakkottai, and Sandeep Chinchali. Constrained posterior sampling: Time series generation with hard constraints. *Advances in Neural Information Processing Systems*, 38:107923–107963, 2026.

Marien Renaud, Jean Prost, Arthur Leclaire, and Nicolas Papadakis. Plug-and-play image restoration with stochastic denoising regularization. In *Forty-first International Conference on Machine Learning*, 2024.

Omer Rochman-Sharabi and Gilles Louppe. Predict-project-renoise: Sampling diffusion models under hard constraints. *arXiv preprint arXiv:2601.21033*, 2026.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

Anjany Sekuboyina, Malek E Husseini, Amirhossein Bayat, Maximilian Löffler, Hans Liebl, Hongwei Li, Giles Tetteh, Jan Kukačka, Christian Payer, Darko Štern, et al. Verse: a vertebrae labelling and segmentation benchmark for multi-detector ct images. *Medical image analysis*, 73:102166, 2021.

Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. In *International Conference on Learning Representations*, volume 2024, pp. 7624–7654, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International conference on learning representations*, 2023a.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pp. 32483–32498. PMLR, 2023b.

Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11956–11965, 2021.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. *arXiv preprint arXiv:2212.00490*, 2022.

Luhuan Wu, Brian Trippe, Christian Naesseth, David Blei, and John P Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021.

Lingxiao Yang, Shutong Ding, Yifan Cai, Jingyi Yu, Jingya Wang, and Ye Shi. Guidance with spherical gaussian constraint for conditional diffusion. *arXiv preprint arXiv:2402.03201*, 2024.

Hannah Zhai, Yurui Chen, Yuichiro Kitada, Hiroo Takayama, and Vijay Vedula. Hemodynamic analysis of a repaired ascending aorta with preserved aortic root. *bioRxiv*, pp. 2026–01, 2026.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1219–1229. IEEE, 2023.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish writing, for retrieval and discovery of related work, and for research ideation and execution. We did not use generative AI tools to draft sections of the paper or to generate synthetic datasets; other disclosure categories not selected in the submission form are not applicable to this work.

All AI-assisted outputs were reviewed by the authors before being incorporated into the work. Suggested references and related work were checked against the original sources, research ideas and methodological suggestions were independently evaluated by the authors, and AI-assisted writing was edited and verified for accuracy and consistency with the underlying results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.