Meltdown: Circuits and Bifurcations in Point-Cloud-Conditioned 3D Diffusion Transformers

Anonymous authors
Paper under double-blind review

Abstract

Sparse point clouds are a common input modality for 3D surface reconstruction, including in safety-critical settings such as surgical navigation and autonomous perception. Recent point-cloud-conditioned 3D diffusion transformers achieve state-of-the-art results in this regime by leveraging learned priors. We show that these models can fail catastrophically under realistic input variation, and present a mechanistic case study of why. We identify a failure mode we call *Meltdown*: tiny on-surface perturbations to a sparse input point cloud can fracture the reconstructed output into hundreds of disconnected pieces. Adversarial search recovers Meltdown in 89.9–100% of shapes across the two open-weight state-of-the-art architectures we study (WaLa, Make-a-Shape) on real-world datasets (GSO, SimJEB) and under both DDPM and DDIM sampling. We trace Meltdown along the forward pass: it is governed by how uniformly the points are distributed on the surface, faithfully transduced through the point-cloud encoder, and committed by a single early-denoising cross-attention write in the diffusion backbone. Diffusion-trajectory ensembles exhibit symmetry-breaking near this commit step, consistent with a bifurcation of the reverse process. Through a suite of matched-magnitude controls, we show that the variable on which the model commits is directional, concentrated in a low-rank subspace of the write's perturbation drift. Motivated by this finding, we introduce `PowerRemap`, a test-time control that reshapes the singular spectrum of the localized write to suppress this drift, with rescue rates of 98.3% on WaLa and 84.6% on Make-a-Shape. Together, these results link a circuit-level cross-attention mechanism to a trajectory-level account of the failure, demonstrating how mechanistic analysis can explain and guide behavior in conditional diffusion transformers.

1 Introduction

Mechanistic interpretability has produced detailed circuit-level accounts of how transformer language models compute, with rigorous causal interventions identifying components responsible for specific behaviors (Wang et al., 2023; Conmy et al., 2023; Sharkey et al., 2025). Extending these techniques to diffusion transformers is an emerging direction (Tinaz et al., 2025; Surkov et al., 2025; Tang et al., 2022; Shabalin et al., 2025). We present a mechanistic case study on how large-scale diffusion transformers can fail unexpectedly, given the controlled task of 3D surface reconstruction from sparse point clouds.

3D surface reconstruction is a well-studied problem in computer vision and graphics (Huang et al., 2022). Many practical pipelines operate on point-cloud inputs, particularly where passive imaging cannot reliably resolve geometry (Stathopoulou & Remondino, 2023), including safety-critical settings such as surgical navigation (Liu et al., 2024) and perception for autonomous driving (Zhang et al., 2023). In such settings, these point clouds are typically *sparse* (Quan et al., 2024; Huang et al., 2024; Sulzer et al., 2024), which makes reconstruction ill-posed and motivates the use of generative priors. *Diffusion transformers*, which attain state-of-the-art results across many generative modalities (Chen et al., 2024; Sahoo et al., 2024; Jia et al., 2025; Lu et al., 2024), have recently been adapted to this task: priors learned from large-scale datasets compensate for the missing geometric information (Sanghi et al., 2024; Hui et al., 2024; Wu et al., 2024; Cao et al., 2024). Understanding when and how such priors fail under realistic input variation is therefore

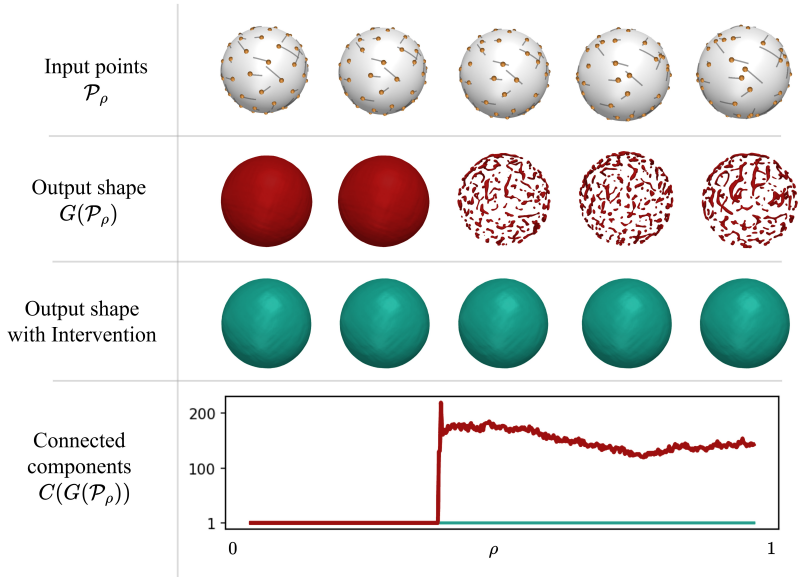


Figure 1: **Meltdown.** We investigate point-cloud-conditioned diffusion transformers on the task of 3D surface reconstruction from sparse point clouds. Tiny on-surface perturbations to a point cloud can fracture the output into many disconnected pieces — a failure we call *Meltdown*. We trace Meltdown along the forward pass and localize a single cross-attention activation early in the denoising process that commits the trajectory to the fragmented attractor. Based on this analysis, we propose `PowerRemap`, a test-time intervention that stabilizes diffusion-based surface reconstruction under sparse conditions.

a prerequisite for trustworthy deployment, and a natural target for mechanistic analysis. To the best of our knowledge, two large-scale point-cloud-conditioned diffusion transformers are currently open-weight[1]: WaLa (Sanghi et al., 2024) and Make-A-Shape (Hui et al., 2024).

We study WaLa and Make-A-Shape on the task of surface reconstruction from sparse point clouds. We observe a striking *failure mode*: a tiny on-surface perturbation to the input point cloud can fracture the output into many disconnected pieces. We call this failure mode *Meltdown* and identify it as a robustness concern for sparse-input deployment. We analyze the phenomenon through two complementary lenses: causal interventions on the network's internal circuits, and the bifurcation structure of the diffusion process itself.

First, we trace Meltdown along the forward pass: it is governed by how uniformly the input points are distributed on the surface, faithfully passed through the encoder, and committed by a single cross-attention activation early in the denoising process. Second, targeted controls show that the variable on which the model commits is directional. Motivated by this finding, we introduce `PowerRemap`, a test-time control on the localized activation that stabilizes sparse point-cloud conditioning. Third, interpreted through diffusion dynamics, the localized commit at the first denoising step sets the trajectory's position just before a symmetry-breaking bifurcation of the reverse process: small differences written into the residual stream by the cross-attention lever are amplified across a basin separatrix into the fragmented attractor over the subsequent steps.

Our contributions are summarized as follows:

1. **Interpretability case study.** We provide a worked example of how circuit-level mechanistic analysis and diffusion-dynamics theory can be combined to explain and steer behavior in conditional diffusion transformers. We link a single-cell cross-attention mechanism, isolated by activation patching with matched-magnitude directional controls, to a trajectory-level account of spontaneous symmetry breaking in the reverse process.

[1]Concurrent work extends 3D generation in different directions: point clouds as auxiliary control over image-conditioned bases (Hunyuan3D et al., 2025) and rectified-flow rather than diffusion backbones (Xia et al., 2026).

2. **Failure phenomenon: Meltdown.** We show that the state-of-the-art point-cloud-conditioned 3D diffusion transformers WaLa (Sanghi et al., 2024) and Make-a-Shape (Hui et al., 2024) perform surface reconstruction from sparse point clouds in a brittle manner: small on-surface perturbations to the input point cloud can fracture the output into multiple disconnected pieces. We call this failure phenomenon Meltdown.

3. **Test-time intervention: `PowerRemap`.** Motivated by the mechanistic analysis, we propose a test-time spectral control at the identified cross-attention site. `PowerRemap` averts Meltdown in 98.3% (84.6%) of cases on the *Google Scanned Objects (GSO)* dataset (Downs et al., 2022) and in 97.7% (83.3%) of cases on the *SimJEB* (Whalen et al., 2021) dataset for WaLa (Make-a-Shape).

We introduce the failure phenomenon, Meltdown, in Section 2. In Section 3, we analyze Meltdown from the perspective of mechanistic interpretability. Section 4 introduces our method, `PowerRemap`, and presents results on the GSO and SimJEB datasets. Finally, we link Meltdown to diffusion dynamics in Section 5 and discuss current limitations in Section 6.

2 Failure phenomenon: Meltdown

In this work, we investigate two leading open-weight point-cloud-conditioned 3D diffusion transformers, namely WaLa (Sanghi et al., 2024) and Make-a-Shape (Hui et al., 2024). Such models can generate surfaces from point clouds, thus solving the *surface reconstruction* task: given a set $\mathcal{P} = \{p_i\}_{i=1}^N \subset \mathcal{S} \subset \mathbb{R}^3$ of N points sampled from an underlying surface $\mathcal{S}$, the model G should reconstruct a surface consistent with the input and approximating the underlying surface $G(\mathcal{P}) \approx \mathcal{S}$. In many real-world scenarios (e.g., fast scene capture), N can be small, i.e. the point cloud is *sparse*.

As illustrated in Figure 1, we observe that there exist two sparse point clouds $\mathcal{P}, \mathcal{Q}$ that are close in the input space, but the corresponding outputs differ severely: $G(\mathcal{P})$ is a connected surface while $G(\mathcal{Q})$ is a fragmented "speckle" of disconnected pieces. We will refer to this sudden catastrophic fracture as *Meltdown*.

To study this failure phenomenon systematically, let us first introduce the topological quantity C that counts the connected components of the output surface and serves as a quantifiable identifier of the healthy ($C = 1$) versus unhealthy output ($C > 1$). Furthermore, let us consider a running example where the points are sampled from a simple sphere: $\mathcal{S} = \{x : \|x\|_2 = 1\}$. This allows us to perform experiments that precisely control for the distribution of the points. Specifically, we fix the random seed and first identify two point clouds of the same size $N = 400$: $\mathcal{P}_0$ which produces a sphere output $C(G(\mathcal{P}_0)) = 1$ and $\mathcal{P}_1$ which produces a speckle output $C(G(\mathcal{P}_1)) \gg 1$ (typically around 100). Using spherical interpolation (geodesics on general surfaces), we can construct a continuous family of point clouds $\mathcal{P}_\rho \subset \mathcal{S}$. We sweep $\rho \in [0, 1]$ and record $C(\rho) := C\left(G\left(\mathcal{P}_\rho\right)\right)$.

Figure 1 illustrates the outcome of this experiment. As we sweep ρ from 0 to 1, we first observe a long plateau of $C(\rho) = 1$, followed by a sudden jump to $C(\rho) \gg 1$ over a very narrow range of ρ. Refining the steps around this transition, we observe an effectively discontinuous jump in the macroscopic descriptor $C(\rho)$.

In Appendix B, we report observing Meltdown across state-of-the-art point-cloud-conditioned diffusion transformers, i.e., WaLa (Sanghi et al., 2024) and Make-a-Shape (Hui et al., 2024), real-world datasets, i.e., Google Scanned Objects (Downs et al., 2022) and SimJEB (Whalen et al., 2021), and denoising strategies, i.e., DDIM (Song et al., 2021) and DDPM (Ho et al., 2020b). Furthermore, we examine the prevalence of Meltdown depending on the sparsity of the input point-cloud in Appendix B.4.

3 Mechanistic analysis and intervention

After observing and quantifying Meltdown, we ask what causes it. We center our analysis on WaLa for clarity, while demonstrating that our insights transfer robustly to Make-a-Shape (Appendix B).

3.1 WaLa: Diffusion transformer

Before we investigate the behavior, we briefly summarize the relevant parts of the WaLa diffusion transformer. A more detailed description is available in Appendix A, the original work (Sanghi et al., 2024), and the references therein.

Transformer. WaLa is a latent diffusion model with a point-net encoder E, U-ViT-style (Hoogeboom et al., 2023) denoising backbone B, and VQ-VAE decoder (van den Oord et al., 2017) D. The U-ViT $B = B^{K-1} \circ \cdots \circ B^0$ has $K = 32$ transformer *blocks* B^k. The condition $\mathbf{C} \in \mathbb{R}^{1024\times1024}$ enters via both AdaLN modulation Esser et al. (2024) and cross-attention. Denoting by $\mathbf{Z}^k \in \mathbb{R}^{1728\times1152}$ the tokens entering the k-th block, it computes $B^k : \mathbf{Z}^k \mapsto \mathbf{Z}^{k+1}$ as a combination of multi-head self-attention SA and cross-attention CA layers (col. 2) with residual connections (col. 3):

$$\mathring{\mathbf{Z}} = \mathrm{Ad\mathring{a}LN}(\mathbf{Z}^k, \mathbf{C}), \qquad \mathring{\mathbf{Y}} = \mathrm{SA}(\mathring{\mathbf{Z}}), \qquad \mathring{\mathbf{R}} = \mathring{\mathbf{Y}} + \mathbf{Z}^k, \tag{1a}$$

$$\mathbf{Z} = \mathrm{AdaLN}(\mathring{\mathbf{R}}, \mathbf{C}), \qquad \mathbf{Y} = \mathrm{CA}(\mathbf{Z}, \mathbf{C}), \qquad \mathbf{R} = \mathbf{Y} + \mathring{\mathbf{R}}, \tag{1b}$$

$$\bar{\mathbf{Z}} = \mathrm{Ad\bar{a}LN}(\mathbf{R}, \mathbf{C}), \qquad \bar{\mathbf{Y}} = \mathrm{MLP}(\bar{\mathbf{Z}}), \qquad \mathbf{Z}^{k+1} = \bar{\mathbf{Y}} + \mathbf{R}. \tag{1c}$$

Diffusion. WaLa is trained in the standard DDPM (Ho et al., 2020b) framework. At inference, the reverse diffusion maps an initial Gaussian latent $\mathbf{Z}_T \sim \mathcal{N}(0, I)$ to $\mathbf{Z}_0$ by iterating over a fixed schedule of denoising steps $t \in \mathcal{T} = \{T, \ldots, 0\}$, where at each step the denoiser conditioned on $\mathbf{C}$ updates $\mathbf{Z}_t \to \mathbf{Z}_{t-1}$. At inference-time, we can sample using DDIM (Song et al., 2021) or DDPM (Ho et al., 2020b).

We ask whether the failure is determined by the input cloud $\mathcal{P}$ itself , by the encoder E that maps it to $\mathbf{C}$, or by the diffusion backbone B that reads $\mathbf{C}$. In Appendix E, we establish a link to the statistical properties of the input data. We find that a classical sphere-uniformity functional, the Riesz s=2 energy, is predictive of Meltdown. The ρ-path of is one trajectory through this scalar and crosses its threshold exactly where $C(\rho)$ jumps. Appendix F traces this scalar layer-by-layer through the encoder E and finds it transduced into $\mathbf{C}$ without amplification or distortion. Hence, we conjecture that the failure must commit somewhere in the diffusion backbone B, on a signal that is already present in $\mathbf{C}$. We ask where: which submodule of which block B^k at which denoising step t reads the melt-relevant content of $\mathbf{C}$ in a way that decides the trajectory's fate.

To localize this commit site, we turn to *activation patching* (Heimersheim & Nanda, 2024; Zhang & Nanda, 2024), a standard tool in mechanistic interpretability for testing the causal role of individual activations: by swapping a single activation between a healthy and an unhealthy forward pass and measuring the change in outcome, one isolates which sites carry the signal responsible for a behavior. Our setup is well-suited to this technique. The continuous ρ-path provides a controlled transition from a healthy run on $\mathcal{P}_0$ (C=1) to an unhealthy run on $\mathcal{P}_1$ ($C \gg 1$) along the Meltdown trajectory, and C provides an objective scalar outcome to score against.

3.2 Localizing Meltdown via activation patching

Activations live on a depth-time grid indexed by block $k \in \mathcal{K} = \{0, \ldots, 31\}$ and denoising step $t \in \mathcal{T} = \{7, \ldots, 0\}$. Since the perturbation enters through $\mathbf{C}$, we begin at the sites where $\mathbf{C}$ is read, i.e., AdaLN modulation and cross-attention, giving $32\times8\times2 = 512$ conditioning cells. A single cell restores connectivity when its activation is replaced by the corresponding healthy value: the cross-attention write $\mathbf{Y}_{4,7} \in \mathbb{R}^{1728\times1152}$ at block 4, first denoising step (Fig. 2). We henceforth write $\mathbf{Y} \equiv \mathbf{Y}_{4,7}$. Appendix D extends the scan to all 19 within-block activations across the full 32×8 grid (4,864 patches per seed). $\mathbf{Y}$ remains the unique site that rescues with full shape quality, and other rescuing components carry its signal downstream. Make-A-Shape exhibits the analogous early-step localization (Appendix B), and patching is robust across diffusion seeds and shape classes (Appendix B.5). The early-step location is consistent with the latent being close to noise at t=7, so the model leans most heavily on conditioning then (Liu et al., 2025).

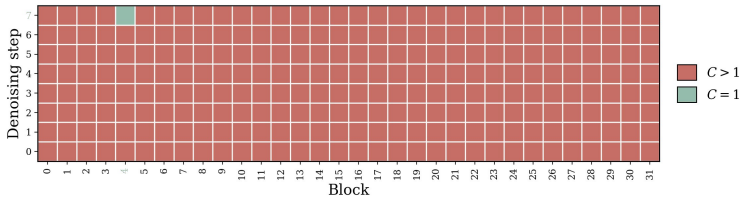


Figure 2: Our search in activation space finds that a single cross-attention write $\mathbf{Y}_{4,7}$ controls Meltdown.

3.3 Investigating the effect of patching

Having localized $\mathbf{Y}$ as the single-cell handle on Meltdown, we ask what variable inside $\mathbf{Y}$ the model commits on. We will find that the answer is directional rather than scalar: the commit is concentrated in a low-rank subspace of $\mathbf{Y}$'s perturbation drift between healthy and unhealthy runs. We then introduce a scalar probe of $\mathbf{Y}$'s spectrum that co-moves reliably with this directional commit and serves as a useful diagnostic of the transition, while not itself being the causal variable.

3.3.1 The committed variable is directional

To identify the variable on which the model commits, we ablate $\mathbf{Y}$ along the directions in which it moves with ρ: from the unhealthy $\mathbf{Y}$ we subtract its projection onto the top twenty singular directions of the drift between healthy and unhealthy runs. This rescues every diffusion seed we tested. Four magnitude-matched controls, i.e., a random subspace orthogonal to the drift, scalar attenuation, isotropic Gaussian noise, and removing $\mathbf{Y}$'s own top components, fail across thousands of runs (Appendix C). Therefore, we conjecture that the variable on which the model commits is the directional content of $\mathbf{Y}$'s perturbation drift, concentrated in a low-rank subspace. Make-a-Shape, which exposes $\mathbf{Y}$ as one of only eight cross-attention writes rather than thirty-two, admits an analogous conclusion: removing the tail of $\mathbf{Y}$'s spectrum directly rescues, while removing its top is destructive (Appendix C.3). The converse intervention is informative on the other side: transplanting the unhealthy $\mathbf{Y}$ into an otherwise-healthy run does not by itself induce Meltdown (Appendix D). Therefore, we conjecture that $\mathbf{Y}$ is a single-cell handle on the failure, and that handle is the directional content of its spectrum.

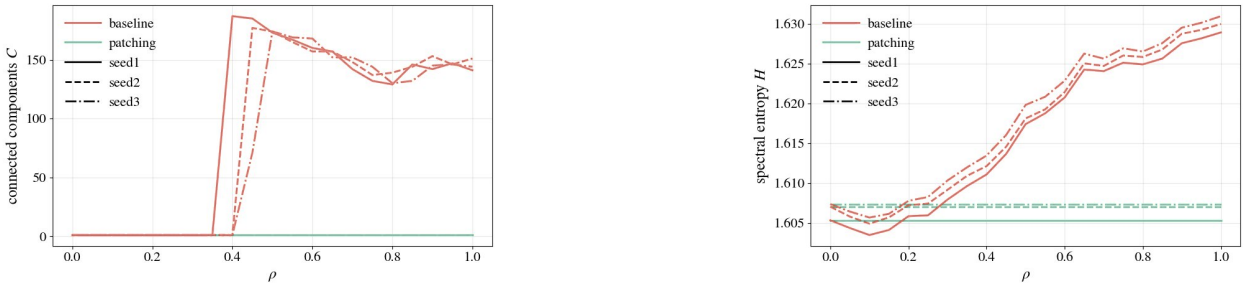
3.3.2 A scalar probe of the transition

The directional commit lives in a high-dimensional, run-specific subspace and is not directly observable at test time. We therefore ask whether there is an observable scalar summary of $\mathbf{Y}$ that co-moves with this commit and can serve as a diagnostic of the transition. We considered several natural candidates along the ρ-sweep — activation norm, the condition number, and entropy-based summaries of the singular spectrum (a comparison is in Appendix B.6). While most stay flat along ρ or vary idiosyncratically across shapes and seeds, one candidate behaves consistently: the spectral entropy (Powell & Percival, 1979)

$$H(\mathbf{Y}) \;=\; -\sum_i p_i \log p_i, \qquad p_i \;=\; \sigma_i^2 \big/ \textstyle\sum_j \sigma_j^2, \tag{2}$$

with σ_i the singular values of $\mathbf{Y}$. Along the ρ-sweep, $H(\rho)$ rises smoothly while $C(\rho)$ jumps, and patching $\mathbf{Y}$ suppresses both (Fig. 3). The pattern transfers to Make-a-Shape and holds across diverse shapes from GSO and SimJEB (Appendix B). H is a robust diagnostic: a cheap, observable scalar that reliably tracks proximity to the transition across shapes, seeds, and architectures. It is not, however, the variable on which the model commits. The matched-magnitude controls of §3.3.1 push H both above and below the rescuing value without rescuing, while the directional surgery leaves H essentially unchanged and rescues (Appendix C.7). H co-moves with the directional commit but does not cause it.

The two roles of $\mathbf{Y}$'s spectrum are now distinct: the directional content of the spectrum is the surface on which the model commits (§3.3.1), and the entropy of the spectrum is an observable scalar that co-moves with this commit (§3.3.2). The next section uses this picture to build a deployable intervention.



(a) Connected components C vs. ρ

(b) Spectral entropy H vs. ρ

Figure 3: The spectral-entropy probe H co-moves with the connectivity transition. As we move from a healthy to an unhealthy run, the baseline case shows a smooth rise in H and a sudden jump in C. Patching $\mathbf{Y}$ keeps both at healthy levels. The behavior is consistent across diffusion seeds. H is diagnostic of the transition; the causal variable is the directional drift inside $\mathbf{Y}$ (§3.3.1).

4 `PowerRemap`: a test-time intervention

The directional surgery of §3.3.1 localizes the lever but is not deployable: it requires the perturbation drift between a healthy and an unhealthy run, and therefore a healthy reference cloud which is unavailable at test time. We seek a single-pass intervention on $\mathbf{Y}$ that depresses this directional content without one. Both the lever and the probe live on the singular spectrum of $\mathbf{Y}$, so we operate on it directly: we modify the singular values while leaving the singular vectors fixed, so the feature directions written into the residual stream are preserved and only their energy distribution changes.

Concretely, let $\mathbf{Y} = U\Sigma V^\top$ with $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ and $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. `PowerRemap` replaces Σ with

$$\Sigma' = \mathrm{diag}(\sigma'_1, \ldots, \sigma'_n), \qquad \sigma'_i = \frac{\sigma_i^\gamma}{\sigma_1^{\gamma-1}}, \qquad \gamma \geq 1, \tag{3}$$

and writes $\mathbf{Y}' = U\Sigma' V^\top$ back into the residual stream at the site identified in §3.2. A sweep applying `PowerRemap` at every cross-attention and MLP site across WaLa's U-ViT corroborates that the intervention is site-specific: rescues are confined to $\mathbf{Y}_{4,7}$ and two upstream feeders ($\mathbf{Y}_{3,7}$, $\mathrm{MLP}_{0,7}$) at the first denoising step (Appendix G). The strength γ controls how sharply the tail is suppressed: $\gamma = 1$ recovers the identity, and larger γ concentrates the spectrum onto the leading singular values. The effective γ is model-dependent (WaLa: $\gamma = 100$; Make-a-Shape: $\gamma = 1.05$; Appendix B.8), reflecting the architectural redundancy gap between the two models: WaLa exposes $\mathbf{Y}$ as one of 32 cross-attention writes plus AdaLN, so $\mathbf{Y}$ tolerates aggressive compression; Make-a-Shape exposes $\mathbf{Y}$ as one of only 8 writes, where the same operation must be milder to preserve the conditioning content the model cannot spare.

4.0.1 Evaluation at scale

We assess whether Meltdown and the effectiveness of `PowerRemap` generalize across diverse input geometries on two datasets, *Google Scanned Objects (GSO)* (Downs et al., 2022) and *SimJEB* (Whalen et al., 2021), neither of which was used to train WaLa or Make-a-Shape. The evaluation protocol is detailed in Appendix B.3.

For WaLa we evaluate every shape with a single global γ=100. For Make-a-Shape, the milder γ regime identified in §4 requires per-shape calibration: we use category-stratified subsets (130 GSO, 30 SimJEB shapes) and select γ per shape by grid search using output connectivity C=1 as the criterion. This requires no ground-truth surface and is therefore deployable at test time. Effective γ values cluster tightly (median 1.10, σ=0.13 on GSO; median 1.05, σ=0.063 on SimJEB; Appendix B.8), so the grid can be small in practice. Subset sizes for Make-a-Shape reflect its 100-step DDIM schedule (vs. 8 for WaLa), which puts full-corpus γ-search outside our compute budget.

GSO. GSO (Downs et al., 2022) is a diverse corpus of 1,030 scanned household objects. We identify Meltdown in 89.9% of WaLa runs and 100% on the Make-a-Shape subset. `PowerRemap` stabilizes 98.3% of WaLa failures and 84.6% on the Make-a-Shape subset (Table 1, left; qualitative examples in Fig. 4).

SimJEB. SimJEB (Whalen et al., 2021) is a benchmark of 381 3D jet-engine bracket CAD models. We identify Meltdown in 92.4% of WaLa runs and 100% on the Make-a-Shape subset. `PowerRemap` stabilizes 97.7% of WaLa failures and 83.3% on the Make-a-Shape subset (Table 1, right).

Additional experiments show that representative alternative interventions such input-cloud uniformization and noise injection do not rescue Meltdown, while `PowerRemap` does (Appendix H). Further experiments show that Meltdown and the efficacy of `PowerRemap` generalize to multi-object inputs (Appendix B.7).

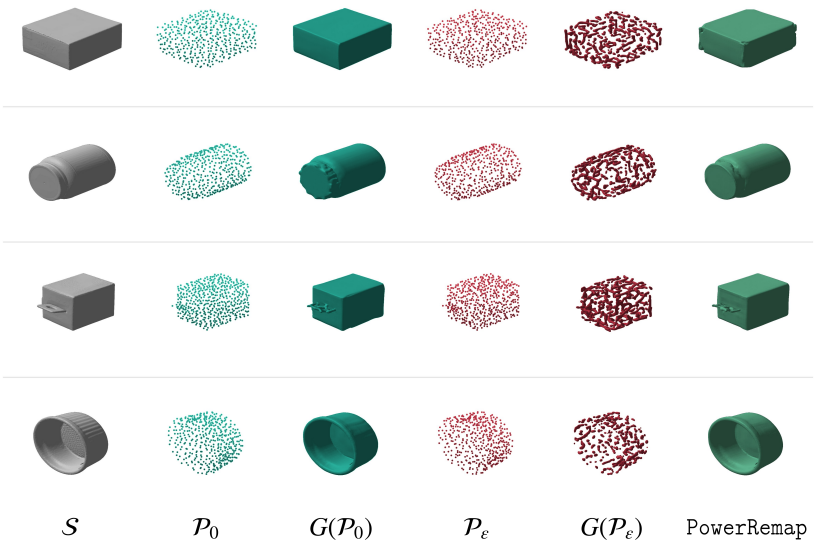


Figure 4: **Qualitative WaLa results on *Google Scanned Objects***, one shape per row. For each shape, the ground-truth surface $\mathcal{S}$ is sampled to produce a sparse healthy point cloud $\mathcal{P}_0$, from which WaLa reconstructs a connected baseline $G(\mathcal{P}_0)$ with $C{=}1$. A tiny on-surface perturbation yields $\mathcal{P}_\varepsilon$, yet the corresponding output $G(\mathcal{P}_\varepsilon)$ fractures into hundreds of disconnected pieces ($C{\gg}1$). We call this phenomenon Meltdown. `PowerRemap`, applied to the same perturbed input, restores connectivity. Adversarial search (Alg. 1) finds Meltdown in 89.9% of WaLa runs on GSO, of which `PowerRemap` rescues 98.3% (Table 1).

5 Diffusion dynamics

Diffusion dynamics refers to a collection of ideas describing the generative diffusion process using established theory from statistical physics (Raya & Ambrogioni, 2023; Biroli et al., 2024; Yu & Huang, 2025; Ambrogioni, 2025), information theory (Ambrogioni, 2025), information geometry (Chen et al., 2023; Ventura et al., 2025), random-matrix theory (Ventura et al., 2025), and dynamical systems (Ambrogioni, 2025). Key concepts from diffusion dynamics allow us to frame both the observed failure phenomenon and the intervention, ultimately connecting the mechanistic analysis to a theoretically established interpretation of the generative diffusion process.

5.1 Preliminaries

We introduce key ideas of diffusion dynamics adapted from Raya & Ambrogioni (2023); Biroli et al. (2024); Ambrogioni (2025). The reverse-time diffusion can be viewed as a noisy gradient flow in a time-dependent potential $u(\cdot, s)$:

$$\mathrm{d}\mathbf{X}_t = -\nabla_{\mathbf{x}} u(\mathbf{X}_t, s)\mathrm{d}t + g(s)\mathrm{d}\mathbf{W}_t, \qquad u(\mathbf{x}, s) = -g^2(s)\log p(\mathbf{x}, s) + \Phi(\mathbf{x}, s), \tag{4}$$

Table 1: Category-wise evaluation of `PowerRemap`. *Left:* WaLa on the full GSO and SimJEB datasets — stabilization rate 98.3% (GSO) and 97.7% (SimJEB). *Right:* Make-A-Shape on category-representative subsets — 84.6% (GSO) and 83.3% (SimJEB).

WaLa			
Category	**Shapes**	**Meltdown** [%]	**Rescue** [%]
GSO			
Shoe	254	97.2	99.6
Consumer goods	248	97.6	99.2
Unknown	216	88.4	95.8
Other	112	92.9	99.0
Total	**1030**	**89.9**	**98.3**
SimJEB			
Arch	37	89.2	100.0
Beam	46	100.0	100.0
Block	99	87.9	97.7
Butterfly	43	93.0	95.0
Flat	147	93.2	97.8
Other	9	100.0	88.9
Total	**381**	**92.4**	**97.7**

Make-A-Shape			
Category	**Shapes**	**Meltdown** [%]	**Rescue** [%]
GSO (subset)			
Consumer goods	60	100.0	90.0
Bottles/cans/cups	23	100.0	95.7
Unknown	18	100.0	83.3
Other	29	100.0	65.5
Total	**130**	**100.0**	**84.6**
SimJEB (subset)			
Arch	2	100.0	50.0
Beam	4	100.0	75.0
Block	9	100.0	100.0
Butterfly	3	100.0	66.7
Flat	11	100.0	90.9
Other	1	100.0	0.0
Total	**30**	**100.0**	**83.3**

where $p(\cdot, s)$ is the forward marginal, g is the noise scale, and $\Phi(\mathbf{x}, s) = \int_0^{\mathbf{x}} f(\mathbf{z})\mathrm{d}\mathbf{z}$ integrates the forward drift f. The potential u is essentially a scaled and shifted marginal. The critical points x^* of this potential $\nabla u(x^*, s) = 0$ are the *attractors* of the dynamics. Early in the generation ($t \approx 0$), there is a global symmetric basin with a stable central fixed point, and the trajectories exhibit mean-reverting fluctuations around it. As noise decreases, the energy landscape deforms and, at a critical time τ^*, the fixed point loses stability and the landscape *bifurcates* into two basins. Such bifurcations repeat until at $t \approx T$ the potential has many fixed points aligning with the data modes (i.e., the data points under an exact score assumption). These bifurcation times τ^* can be interpreted as *decision* times where the sample trajectory is committed to a future attractor basin.

Around the degenerate critical point $x^*(\tau^*)$, two paths that are nearby for $t < \tau^*$ may diverge exponentially for $t > \tau*$ due to the Lyapunov exponent becoming positive (the smallest eigenvalue of $\nabla^2 u$ obtained from linearizing the reverse dynamics around the critical point). This can amplify tiny input differences and is the mechanism behind sending trajectories to different attractors.

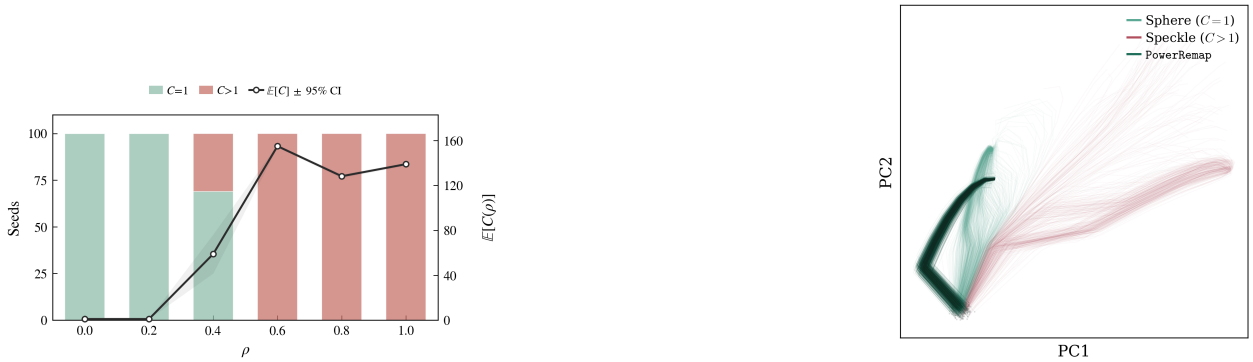
This selection of one among many symmetry-equivalent states is called *spontaneous symmetry breaking*. A canonical example is a ferromagnet: at high temperature ($t \approx T$) spins are disordered, while as $t \to 0$ they align. Any magnetization direction is a priori equivalent, yet each realization picks one. The underlying symmetry is visible only in the ensemble over many realizations.

5.2 Application to Meltdown and intervention

To test the diffusion dynamic perspective of the Meltdown phenomenon and the intervention, we perform several experiments predicted by this view. However, we must first introduce conditioning in the above diffusion dynamics view. For a fixed condition $\mathbf{C}$, this extension is trivial: simply modify the marginal $p(\cdot, s) = p(\cdot, s|\mathbf{C})$. However, a family of conditions, like the univariate interpolation $\{\mathbf{C}(\rho)|\rho \in [0, 1]\}$, introduces an additional dependence in the above formalism, and it is not obvious how to analyze the evident bifurcation around $\mathbf{C}^*$ instead of τ^*. Fortunately, since the symmetry breaking originates *locally* around the bifurcation time τ^* and point x^*, a small change in the condition $\mathrm{d}\mathbf{C}$ can be related to a small change in the initial condition $\mathrm{d}x_T$ through the total differential of the reverse path $\gamma : (t, x_T, \mathbf{C}) \mapsto x_t$. Qualitatively, this allows us to consider different x_T for a fixed $\mathbf{C}$ in place of different $\mathbf{C}$ for a fixed x_T.

Ensemble. Spontaneous symmetry breaking suggests that even though a single trajectory commits to a single attractor (sphere versus speckle), both "symmetric" configurations are visited over an ensemble of random trajectories. We record the trajectories for 100 initial conditions $x_T \sim \mathcal{N}(0, I)$ over the $\rho \in [0, 1]$ range and plot the resulting shape connected component distribution in Figure 5a. The extremes $\rho = 0, 1$

are far from a critical condition, and all trajectories converge to the respective attractors. However, at the intermediate conditions, the ensemble of trajectories visits both attractors, with the ratio of fractured shapes increasing steadily with ρ. In expectation, the component curve $\mathbb{E}_{x_T}[C(\rho)]$ exhibits a smooth behavior, relaxing the discrete jump in $C(\rho)$ for a single x_T.



(a) (b)

Figure 5: A collection of diffusion trajectories reveals additional insights about the Meltdown phenomenon. (a) In expectation over the initial noise, both the sphere and speckle shapes are produced at intermediate conditions, relaxing the sharp Meltdown behavior for a fixed initial noise. (b) Latent diffusion trajectories projected onto a 2D linear subspace spanned by the first two principal components of the final distribution of the baseline. The `PowerRemap` trajectories in green form a tight bundle following a different path that converges to a minor mode of the baseline distribution.

Trajectories. At intermediate ρ=0.4 we visualize 1000 trajectories projected onto the first two principal components of $p(\cdot, 0)$ (Fig. 5b). The first denoising step is mean-reverting (Biroli et al., 2024; Ventura et al., 2025). The second step marks the symmetry breaking. We confirm this with Hartigan's dip test (Hartigan & Hartigan, 1985) on the projection onto $\hat{\mathbf{u}} = (\mathbf{c}_{\text{sphere}} - \mathbf{c}_{\text{speckle}})/\|\cdot\|$: unimodality is not rejected at t=T or after step 1 (Holm-adjusted p=1.0), but is rejected from step 2 onward (p=4.5×10^{-3} at t=5, p<10^{-4} thereafter). The intervention alters the first step, after which trajectories flow to a tight minor mode of the baseline.

Potential. We calculate the potential similar to the procedure introduced by Raya & Ambrogioni (2023). We select a pair of representative trajectories from each attractor and interpolate between them along a variance-preserving curve $x_t(\alpha) = \cos(\alpha)x_t^{\text{sphere}} + \sin(\alpha)x_t^{\text{speckle}}$ for $\alpha \in [-0.2\pi, 1.2\pi]$. Figure 6 reveals the two diffusion stages separated by the bifurcation time $\tau^* \approx 5$, where the single potential well flattens and splits into the two attractor basins.

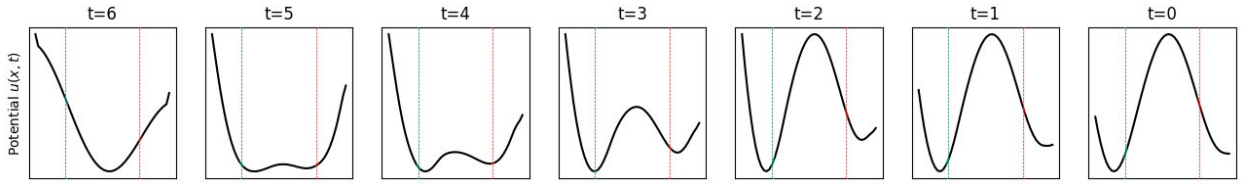


Figure 6: The potential u (related to the marginal probability via Eq. (4)) reveals the two diffusion stages separated by the bifurcation time $\tau^* \approx 5$, where the single potential well flattens and splits into the two attractor basins. The particle's location just before this early bifurcation commits it to the final attractor and ultimately determines the generated shape. Small perturbations around this time become amplified, giving the appearance of discrete jumps that characterize the observed Meltdown.

We note that the mechanistic localization in §3.2 and the bifurcation analysis below describe two distinct moments of the reverse trajectory. The cross-attention lever $\mathbf{Y}_{4,7}$ writes into the residual stream at the

first denoising step (t=7); the potential bifurcates around $\tau^* \approx 5$ (Fig. 6), and Hartigan's dip test rejects unimodality from t=5 onward. The lever therefore acts *upstream* of the basin split: it sets the trajectory's position at the moment the basins form, after which exponential separation under a positive Lyapunov exponent amplifies the difference across the separatrix. This sequencing also explains why the patching scan of Appendix D finds no within-block rescue site at $t \in \{0, 1\}$: by then the trajectory lies on one side of the separatrix and single-cell interventions cannot transport it across.

6 Discussion and Limitations

What the analysis demonstrates. Three threads come together in this work. (i) A specific failure pattern—catastrophic fragmentation of reconstructions from sparse on-surface inputs—is identified, quantified, and shown to generalize across two open-weight architectures, two real-world datasets and two samplers. (ii) Targeted causal interventions isolate this failure to a single cross-attention write in the first denoising step, and matched-magnitude controls demonstrate that the committed variable is the directional content of the perturbation drift of the activation in a low-rank subspace. (iii) Interpreted through diffusion-dynamics theory, the localized commit sits immediately upstream of a symmetry-breaking bifurcation of the reverse-time potential, after which exponential separation under a positive Lyapunov exponent transports the trajectory across a basin separatrix. The two layers of description, i.e., a circuit-level lever and a trajectory-level bifurcation, are internally consistent, and motivate `PowerRemap`, a test-time spectral intervention that rescues 84.6–98.3% of failures across architectures and datasets.

Practical implications. Sparse point-cloud reconstruction is non-trivial because the model must infer geometry the input does not directly resolve. Our results show that, under realistic input variation, the generative prior can move catastrophically between attractors. For safety-critical pipelines such as surgical navigation (Liu et al., 2024) or perception for autonomous driving (Zhang et al., 2023), a model that occasionally produces hundreds of disconnected pieces from a near-identical sparse input is unsuitable regardless of average-case fidelity. Our work provides two practical takeaways. First, scalar correlates of the transition, such as the spectral entropy, provide a cheap online diagnostic that can be computed at test time without ground-truth surfaces. Second, single-site spectral interventions can stabilize a deployed model without retraining, with hyperparameters that cluster tightly across shapes (Appendix B.8).

Why this particular signature? A natural follow-up question is why the particular cross-attention write in the first denoising step carries a directional commit for this specific failure. The first denoising step is when the latent is closest to noise and the model leans most heavily on conditioning (Liu et al., 2025). The early cross-attention writes are accordingly the highest-bandwidth conduits for conditioning information. Why the directional content of the cross-attention write concentrated in a low-rank subspace becomes primarily responsible for topology specifically is open. We conjecture this reflects an inductive bias acquired during training. The general pattern of diffusion transformers committing to coarse, high-level structure before resolving fine-grained detail has been observed in text-to-image models (Tinaz et al., 2025). Our finding is consistent with a 3D analog in which the coarse structure being committed early is topology, and the commitment is encoded as a low-rank direction in conditioning space.

Limitations.

Architectural and methodological scope. Our analysis covers two open-weight point-cloud-conditioned DDPM-based diffusion transformers, WaLa (Sanghi et al., 2024) and Make-A-Shape (Hui et al., 2024). Transfer to flow-matching backbones (Hunyuan3D et al., 2025; Xia et al., 2026) is unverified.

Distributed signatures cannot be ruled out. Single-cell activation patching identifies a sufficient lever for rescue, but the causal-asymmetry result in Appendix D (noising scan) shows that transplanting an unhealthy $\mathbf{Y}_{4,7}$ into an otherwise-healthy run does not by itself induce Meltdown. The committed signal is carried cumulatively by the residual stream, and the cross-attention site is a single-cell handle, not a sufficient cause in isolation. Distributed commit signatures across multiple submodules are therefore not excluded by our analysis.

Trajectory-level claims rest on representative paths. The bifurcation analysis in Section 5 uses representative trajectories from a sphere example, and the unimodality tests use a fixed 2D projection. The qualitative picture transfers across shapes and seeds (Appendix B.5), but we do not claim a quantitative bifurcation theorem for arbitrary shape spaces.

Per-shape calibration for Make-A-Shape. On WaLa, a single global $\gamma = 100$ suffices. On Make-A-Shape, per-shape selection over a small grid is required, using connectivity C=1 as the criterion. While this criterion is deployable at test time (no ground truth required), it adds inference cost. The tight clustering of effective γ around 1.05–1.10 (Appendix B.8) suggests this could be ameliorated by a learned shape-category default.

Adversarial nature of the failure. Meltdown is identified by adversarial search (Algorithm 1). We show that arbitrary on-surface samplings *can* induce the failure, that the failure is frequent (89.9–100%) under adversarial search across two corpora, and that it concentrates in the low-areal-density regime (Appendix B.4) characteristic of real sparse-capture pipelines.

Future directions. Several extensions follow naturally. Extending the analysis to flow-matching and rectified-flow 3D backbones would test whether the same combined mechanistic–dynamical picture holds beyond DDPM and DDIM. Probing whether the directional signature is detectable at training time could enable preventive interventions during model development rather than test-time rescue. More broadly, the framework demonstrated here, i.e., activation patching with matched-magnitude directional controls, paired with a diffusion-dynamics interpretation of the resulting commit, appears to us applicable to other conditional generative settings where catastrophic, hard-to-anticipate mode commitments must be diagnosed and mitigated.

7 Conclusion

We have presented a mechanistic case study of a catastrophic failure mode in two state-of-the-art point-cloud-conditioned 3D diffusion transformers. Tiny on-surface perturbations to a sparse input cloud can fracture the reconstructed output into hundreds of disconnected pieces—a failure we call *Meltdown*—with prevalence 89.9–100% under adversarial search across two open-weight architectures (WaLa, Make-A-Shape), two real-world datasets (GSO, SimJEB), and both DDPM and DDIM samplers. Activation patching localizes the failure to a single cross-attention write at the fourth block and first denoising step, and matched-magnitude directional controls demonstrate that the committed variable is the directional content of that activation concentrated in a low-rank subspace of its perturbation drift. Embedded in diffusion-dynamics theory, this localized commit sits immediately upstream of a symmetry-breaking bifurcation of the reverse-time potential. Motivated by this combined picture, we introduced `PowerRemap`, a test-time spectral control on the localized activation that stabilizes 84.6–98.3% of failures across architectures and datasets.

Beyond this specific result, our work offers a template for combining mechanistic interpretability with diffusion-dynamics theory to study conditional diffusion transformers. Linking a single-cell cross-attention mechanism to a trajectory-level account of spontaneous symmetry breaking provides a level of explanatory coverage that, in our experience, neither approach achieves alone: the circuit identifies where and with what the model commits, while the dynamics explain why a tiny local perturbation has a discontinuous global effect. We expect this combined methodology to be useful for other conditional generative models and other failure modes, particularly in safety-critical deployment regimes where catastrophic and hard-to-anticipate behaviors must be diagnosed and mitigated before they reach end users.

References

Luca Ambrogioni. The information dynamics of generative diffusion. *arXiv preprint arXiv:2508.19897*, 2025.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 22669–22679. IEEE, 2023. doi: 10.1109/CVPR52729.2023.02171. URL https://doi.org/10.1109/CVPR52729.2023.02171.

Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1), November 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-54281-3. URL http://dx.doi.org/10.1038/s41467-024-54281-3.

Johann S. Brauchart and Peter J. Grabner. Distributing many points on spheres: Minimal energy and designs. *Journal of Complexity*, 31(3):293–326, June 2015. ISSN 0885-064X. doi: 10.1016/j.jco.2015.02.003. URL http://dx.doi.org/10.1016/j.jco.2015.02.003.

Wei Cao, Chang Luo, Biao Zhang, Matthias Nießner, and Jiapeng Tang. Motion2vecsets: 4d latent vector set diffusion for non-rigid shape reconstruction and tracking, 2024.

Defang Chen, Zhenyu Zhou, Jian-Ping Mei, Chunhua Shen, Chun Chen, and Can Wang. A geometric perspective on diffusion models. *arXiv preprint arXiv:2305.19947*, 2023.

Xinwang Chen, Ning Liu, Yichen Zhu, Feifei Feng, and Jian Tang. Edt: An efficient diffusion transformer framework inspired by human-like sketching. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 134075–134106. Curran Associates, Inc., 2024. URL https://doi.org/10.52202/079017-4261.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 16318–16352. Curran Associates, Inc., 2023. URL https://doi.org/10.48550/arXiv.2304.14997.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. URL https://arxiv.org/abs/2204.11918.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2403.03206.

Kyoungmin Han, Kyujin Jung, Jaeho Yoon, and Minsik Lee. Point cloud resampling by simulating electric charges on metallic surfaces. *Sensors*, 21(22), 2021. ISSN 1424-8220. doi: 10.3390/s21227768. URL https://www.mdpi.com/1424-8220/21/22/7768.

Douglas P. Hardin and Edward B. Saff. Discretizing manifolds via minimum energy points. *Notices of the American Mathematical Society*, 51(10):1186–1194, 2004.

John A. Hartigan and Pamela M. Hartigan. The dip test of unimodality. *Annals of Statistics*, 13:70–84, 1985. URL https://api.semanticscholar.org/CorpusID:119847480.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

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., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020b. URL https://arxiv.org/abs/2006.11239.

Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. Simple diffusion: End-to-end diffusion for high resolution images, 2023. URL https://arxiv.org/abs/2301.11093.

Hui Huang, Dan Li, Hao Zhang, Uri Ascher, and Daniel Cohen-Or. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.*, 28(5):1–7, December 2009. ISSN 0730-0301. doi: 10.1145/1618452.1618522. URL https://doi.org/10.1145/1618452.1618522.

Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, and Kui Jia. Surface reconstruction from point clouds: A survey and a benchmark, 2022. URL https://arxiv.org/abs/2205.02413.

Zhangjin Huang, Yuxin Wen, Zihao Wang, Jinjuan Ren, and Kui Jia. Surface reconstruction from point clouds: A survey and a benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46 (12):9727–9748, 2024. doi: 10.1109/TPAMI.2024.3429209.

Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. Make-a-shape: a ten-million-scale 3d shape model, 2024. URL https://arxiv.org/abs/2401.11067.

Team Hunyuan3D, :, Bowen Zhang, Chunchao Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jingwei Huang, Junlin Yu, Kunhong Li, Linus, Penghao Wang, Qingxiang Lin, Sicong Liu, Xianghui Yang, Yixuan Tang, Yunfei Zhao, Zeqiang Lai, Zhihao Liang, and Zibo Zhao. Hunyuan3d-omni: A unified framework for controllable generation of 3d assets, 2025. URL https://arxiv.org/abs/2509.21245.

Dongya Jia, Zhuo Chen, Jiawei Chen, Chenpeng Du, Jian Wu, Jian Cong, Xiaobin Zhuang, Chumin Li, Zhen Wei, Yuping Wang, and Yuxuan Wang. Ditar: Diffusion transformer autoregressive modeling for speech generation. *CoRR*, abs/2502.03930, 2025. doi: 10.48550/ARXIV.2502.03930. URL https://doi.org/10.48550/arXiv.2502.03930.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models, 2022a. URL https://arxiv.org/abs/2206.00364.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577. Curran Associates, Inc., 2022b. URL https://doi.org/10.48550/arXiv.2206.00364.

Yaron Lipman, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer. Parameterization-free projection for geometry reconstruction. In *ACM SIGGRAPH 2007 Papers*, SIGGRAPH '07, pp. 22–es, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781450378369. doi: 10.1145/1275808.1276405. URL https://doi.org/10.1145/1275808.1276405.

Haozhe Liu, Wentian Zhang, Jinheng Xie, Francesco Faccio, Mengmeng Xu, Tao Xiang, Mike Zheng Shou, Juan-Manuel Perez-Rua, and Jürgen Schmidhuber. Faster diffusion via temporal attention decomposition. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=xXs2GKXPnH.

Zifeng Liu, Zhiyong Yang, Shan Jiang, and Zeyang Zhou. A spatial registration method based on point cloud and deep learning for augmented reality neurosurgical navigation. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 20(6), December 2024. ISSN 1478-596X. doi: 10.1002/rcs.70030. URL http://dx.doi.org/10.1002/rcs.70030.

Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: general-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Un0rgm9f04.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps, 2025. URL https://arxiv.org/abs/2501.09732.

G. E. Powell and I. C. Percival. A spectral entropy method for distinguishing regular and irregular motion of hamiltonian systems. *Journal of Physics A: Mathematical and General*, 12(11):2053–2071, 1979. doi: 10.1088/0305-4470/12/11/017.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation, 2017. URL https://arxiv.org/abs/1612.00593.

Siwen Quan, Junhao Yu, Ziming Nie, Muze Wang, Sijia Feng, Pei An, and Jiaqi Yang. Deep learning for 3d point cloud enhancement: A survey. *arXiv preprint arXiv:2411.00857*, 2024.

Marie-Julie Rakotosaona, Vittorio La Barbera, Paul Guerrero, Niloy J. Mitra, and Maks Ovsjanikov. POINTCLEANNET: learning to denoise and remove outliers from dense point clouds. *CoRR*, abs/1901.01060, 2019. URL http://arxiv.org/abs/1901.01060.

Gabriel Raya and Luca Ambrogioni. Spontaneous symmetry breaking in generative diffusion models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=lxGFGMMSVl.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. URL https://arxiv.org/abs/2112.10752.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 130136–130184. Curran Associates, Inc., 2024. URL https://doi.org/10.52202/079017-4135.

Aditya Sanghi, Aliasghar Khani, Pradyumna Reddy, Arianna Rampini, Derek Cheung, Kamal Rahimi Malekshan, Kanika Madan, and Hooman Shayani. Wavelet latent diffusion (wala): Billion-parameter 3d generative model with compact wavelet encodings, 2024. URL https://arxiv.org/abs/2411.08017.

Stepan Shabalin, Ayush Panda, Dmitrii Kharlapenko, Abdur Raheem Ali, Yixiong Hao, and Arthur Conmy. Interpreting large text-to-image diffusion models with dictionary learning, 2025. URL https://arxiv.org/abs/2505.24360.

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025. URL https://arxiv.org/abs/2501.16496.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Elisavet Konstantina Stathopoulou and Fabio Remondino. A survey on conventional and learning-based methods for multi-view stereo. *The Photogrammetric Record*, 38(183):374–407, 2023. doi: 10.1111/phor.12456.

Patrick Stotko, Michael Weinmann, and Reinhard Klein. Incomplete gamma kernels: Generalizing locally optimal projection operators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(6):4075–4089, 2024. doi: 10.1109/TPAMI.2024.3349967.

Raphael Sulzer, Renaud Marlet, Bruno Vallet, and Loic Landrieu. A survey and benchmark of automatic surface reconstruction from point clouds. *arXiv preprint arXiv:2301.13656*, 2024.

Viacheslav Surkov, Chris Wendler, Antonio Mari, Mikhail Terekhov, Justin Deschenaux, Robert West, Caglar Gulcehre, and David Bau. One-step is enough: Sparse autoencoders for text-to-image diffusion models, 2025. URL https://arxiv.org/abs/2410.22366.

Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention, 2022. URL https://arxiv.org/abs/2210.04885.

Berk Tinaz, Zalan Fabian, and Mahdi Soltanolkotabi. Emergence and evolution of interpretable concepts in diffusion models, 2025. URL https://arxiv.org/abs/2504.15473.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Enrico Ventura, Beatrice Achilli, Gianluigi Silvestri, Carlo Lucibello, and Luca Ambrogioni. Manifolds, random matrices and spectral gaps: The geometric phases of generative diffusion. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

E. Whalen, A. Beyene, and C. Mueller. Simjeb: Simulated jet engine bracket dataset. *Computer Graphics Forum*, 40(5):9–17, August 2021. ISSN 1467-8659. doi: 10.1111/cgf.14353. URL http://dx.doi.org/10.1111/cgf.14353.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 121859–121881. Curran Associates, Inc., 2024. URL https://doi.org/10.52202/079017-3873.

Jiatong Xia, Zicheng Duan, Anton van den Hengel, and Lingqiao Liu. Points-to-3d: Structure-aware 3d generation with point cloud priors, 2026. URL https://arxiv.org/abs/2603.18782.

Zhendong Yu and Haiping Huang. Nonequilbrium physics of generative diffusion models. *Phys. Rev. E*, 111:014111, Jan 2025. doi: 10.1103/PhysRevE.111.014111. URL https://link.aps.org/doi/10.1103/PhysRevE.111.014111.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.

Yuxiao Zhang, Alexander Carballo, Hanting Yang, and Kazuya Takeda. Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177, 2023. doi: 10.1016/j.isprsjprs.2022.12.021.

A Background

In this section, we provide background information on the diffusion transformers WaLa (Sanghi et al., 2024) and Make-A-Shape (Hui et al., 2024) along the dimensions *diffusion* (Appendix A.1) and *transformer* (Appendix A.2).

A.1 Diffusion

Forward transition. Diffusion generative models synthesize data by inverting a Markov chain that gradually corrupts an observation $\mathbf{x}_0 \sim p_{\text{data}}$ with Gaussian noise over T discrete timesteps (Sohl-Dickstein et al., 2015; Ho et al., 2020a). The forward (noising) transition is

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\ \beta_t \mathbf{I}\big), \quad t = 1, \ldots, T, \tag{5}$$

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\big(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0,\ (1-\bar{\alpha}_t)\mathbf{I}\big), \quad \text{with} \quad \bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s), \tag{6}$$

where the variance schedule $\{\beta_t\}_{t=1}^{T} \subset (0,1)$ is chosen so that $\mathbf{x}_T$ is nearly i.i.d. $\mathcal{N}(0, I)$. Both architectures are associated with a cosine variance schedule.

Noise prediction objective. Instead of directly regressing $\mathbf{x}_0$, the denoising neural networks ϵ_θ of WaLa and Make-A-Shape have been trained to predict the added noise:

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\big\|\boldsymbol{\epsilon} - \epsilon_\theta(\underbrace{\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}}_{\mathbf{x}_t}, t)\big\|_2^2\Big], \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I). \tag{7}$$

This "ϵ-parameterization" empirically stabilizes the training and is adopted by nearly all modern models (Ho et al., 2020a).

Denoising (DDPM). Given a trained ϵ_θ, the original Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020a) samples via the stochastic reverse transition

$$p_\theta(\mathbf{x}_{t-1} \,|\, \mathbf{x}_t) = \mathcal{N}\big(\mathbf{x}_{t-1}; \underbrace{\frac{1}{\sqrt{1-\beta_t}}\Big(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(\mathbf{x}_t, t)\Big)}_{\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)}, \sigma_t^2 \mathbf{I}\big), \qquad \sigma_t^2 = \tilde{\beta}_t, \tag{8}$$

where $\tilde{\beta}_t = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}$. Iterating Eq. equation 8 from $t{=}T$ to 1 produces $\mathbf{x}_0$ in T noisy steps.

Denoising (DDIM). Song et al. (2021) showed that the same model admits a *deterministic* implicit sampler (DDIM) obtained by setting the variance term to zero:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\underbrace{\Big(\frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}\Big)}_{\hat{\mathbf{x}}_0} + \sqrt{1-\bar{\alpha}_{t-1}}\,\epsilon_\theta(\mathbf{x}_t, t). \tag{9}$$

Eq. equation 9 preserves the marginal $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)$, enabling user-specified inference schedules (e.g., $t = T, \ldots, 1$ with $T \gg 1$ for high fidelity or sparse subsets for speed) without retraining. Crucially, Eqs. equation 8–equation 9 share the same ϵ_θ trained via Eq. equation 7. Hence one can *train* with the log-likelihood–consistent DDPM objective but sample using DDPM or DDIM.

Classifier-free guidance. Both architectures employ classifier-free guidance (CFG) (Ho & Salimans, 2022). CFG biases the denoising direction toward a user condition without requiring an external classifier. For a current latent $\mathbf{x}_t$ and the shared noise predictor ϵ_θ, we obtain two estimates at the same step t:

the unconditional prediction ϵ_{uncond} (with the condition omitted) and the conditional prediction ϵ_{cond} (under the desired condition). We then form a guided estimate

$$\tilde{\epsilon} = \epsilon_{\text{uncond}} + s\big(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}\big), \qquad s \geq 0,$$

and substitute $\tilde{\epsilon}$ in place of $\epsilon_\theta(\mathbf{x}_t, t)$ in the DDPM/DDIM updates (Eqs. equation 8–equation 9). Setting $s = 1$ at inference-time ignores updates from the unconditional stream, i.e., $\tilde{\epsilon} = \epsilon_{\text{cond}}$.

A.2 Transformer

The noise–prediction objective equation 7 only specifies what to learn but leaves open how the denoiser ϵ_θ is parameterised. Classical DDPMs adopt a convolutional U-Net encoder–decoder (Ronneberger et al., 2015; Ho et al., 2020a), whereas modern large-scale models (Rombach et al., 2022; Bao et al., 2023; Karras et al., 2022b) replace the convolutional blocks with *transformer* layers, yielding the U-ViT (U-shaped Vision Transformer) backbone. Both architectures, WaLa and Make-A-Shape, implement a diffusion generative model via transformer layers.

A.2.1 Overview

Both methods adopt a wavelet–latent diffusion pipeline in which 3D shapes are represented as multiscale wavelet coefficients and a U-ViT-style denoising backbone (Hoogeboom et al., 2023) is trained in the DDPM (Ho et al., 2020b) framework. The key difference lies in how the wavelet data are fed to the diffusion core.

1. WALA first compresses the full wavelet tree with a convolutional VQ-VAE (stage 1), mapping the diffusible wavelet tree to a latent grid. The latent grid is then modeled by a 32-layer U-ViT (stage 2), where each transformer layer runs self-attention and cross-attention, totaling 32 cross-attention calls.

2. Make-A-Shape skips the auto-encoder and instead packs selected wavelet coefficients into a compact grid. The U-ViT backbone then downsamples this tensor to a bottleneck volume. The bottleneck is traversed by a 16-layer U-ViT core—8 self-attention layers immediately followed by 8 cross-attention layers— before up-sampling restores the packed grid.

A.2.2 Conditioning pathway (point-cloud)

In general, both Make-A-Shape and WaLa share a common pipeline for conditioning on point clouds: a PointNet (Qi et al., 2017) encoding followed by aggregation and injecting the resulting latent vectors into the U-ViT generator via *(i)* affine modulation of normalization layers and *(ii)* cross-attention.

In particular, Make-A-Shape injects the conditioning latent vectors into the U-ViT generator at three stages: (1) *concatenation*: the latent vectors are aggregated and concatenated as additional channels of the input noise coefficients, (2) *affine modulation*: the latent vectors are aggregated and subsequently utilized to condition the convolution (down-sampling) and de-convolution (up-sampling) layers via modulating the affine parameters of the group normalization layers, (3) *cross-attention*: each condition latent vector is augmented with an element-wise positional encoding and then fed into a cross-attention module alongside the bottleneck volume.

WaLa injects the conditioning latent vectors into the U-ViT generator at two stages: (1) *affine modulation*: the latent vectors are linearly projected via a global projection network and used to modulate the scale and bias parameters of GroupNorm layers in both the ResNet and attention blocks (AdaGN) (Esser et al., 2024), (2) *cross-attention*: each latent vector, augmented with an element-wise positional encoding, is employed as the key and value in cross-attention modules interleaved within each transformer block.

B Experiments

This section shows that the observations and insights gained through studying the diffusion transformer WaLa (Sanghi et al., 2024) under DDIM sampling on spheres robustly transfer (i) to the diffusion transformer Make-A-Shape (Hui et al., 2024) (ii) sampling under DDPM and (iii) other shapes (GSO and SimJEB). Additionally, this section details the experimental setup to reproduce our results and empirically investigates further variables. In particular:

1. **General** (B.1): This section provides an overview on our experimental setup.
2. **Sphere Experiments** (B.2):
   (a) **General** (B.2.1: This section provides an overview on the setup for the sphere experiments.
   (b) **WaLa, DDIM** (B.2.2): This section reports the experimental setup for the sphere experiments in the main text.
   (c) **WaLa, DDPM** (B.2.3): This section reports additional results for WaLa under DDPM sampling.
   (d) **Make-A-Shape, DDIM** (B.2.4): This section provides results for Make-A-Shape under DDIM sampling.
   (e) **Make-A-Shape, DDPM** B.2.5: This section provides results for Make-A-Shape under DDPM sampling.
3. **Datasets** (B.3): This section details our experiments on GSO (Downs et al., 2022) and SimJEB (Whalen et al., 2021).
4. **Density Study** (B.4): This section examines how the prevalence of Meltdown depends on the sparsity of the input point cloud.
5. **Extended Activation Patching** (D): This section provides activation-patching results on additional components beyond cross-attention.
6. **More Datapoints** (B.5): This section provides further evidence that the patterns observed in Section 3.2-3.3 generalize when evaluated on more data points and random seeds.
7. **Additional Spectral Metrics** (B.6): This section assesses additional spectral metrics as potential indicators of Meltdown.
8. **Multiple Objects** (B.7): This section examines whether the Meltdown phenomenon and the effectiveness of `PowerRemap` extend beyond single-object inputs.
9. **Examining `PowerRemap` strength** (B.8): This section empirically investigates the influence of the `PowerRemap` Strength γ on reconstruction connectivity.
10. **`PowerRemap` on Non-Meltdown Cases** (B.9): This section empirically verifies that `PowerRemap` does not interfere with non-Meltdown cases.

B.1 General

This section provides a general overview on the experimental setup for all results reported in this work.

Restrict analysis to conditional stream. As the failure behavior, Meltdown, is independent of the unconditional stream, we exclusively investigate the conditional prediction stream. That is, we set the CFG scale $s = 1.0$ and restrict our mechanistic analysis (e.g., activations) and diffusion dynamics analysis (e.g., latents) to the conditional stream.

Seeding. Randomness regarding a diffusion trajectory is controlled globally by seeding Python, NumPy, and PyTorch (`torch.backends.cudnn.deterministic=True`, `benchmark=False`) so that every evaluation at a given ρ starts from the same terminal noise x_T.

B.2 Sphere Experiments

This section (i) provides a detailed account on our setup for the sphere experiments and (ii) reports additional results for Meltdown on Make-A-Shape and DDPM sampling.

B.2.1 Setup

We detail the minimal, fully reproducible setup used to produce the sphere experiments for WaLa and Make-A-Shape. Throughout, the control parameter is $\rho \in [0, 1]$, and the phenomenological order parameter is the number of connected components $C(\rho)$ in the generated mesh.

Conditioning clouds on the sphere. We work on $\mathcal{S} = \{x : \|x\|_2 = 1\}$ and fix $N = 400$ points for WaLa and $N = 1200$ for Make-A-Shape. The base cloud $\mathcal{P}(0)$ uses a golden-angle (Fibonacci) sphere distribution:

$$g = \pi(3 - \sqrt{5}), \quad i \in \{0, \dots, N-1\}, \quad y_i = 1 - \frac{2(i+0.5)}{N}, \quad r_i = \sqrt{1 - y_i^2}, \quad \theta_i = g\, i,$$

$$p_i(0) = \big(r_i \cos\theta_i,\ y_i,\ r_i \sin\theta_i\big).$$

A second target cloud $\mathcal{P}(1)$ is produced by jittering each $p_i(0)$ with i.i.d. Gaussian noise $n_i \sim \mathcal{N}(0, 0.1^2 \mathbf{I}_3)$ and renormalizing to the unit sphere:

$$\tilde{p}_i = \frac{p_i(0) + n_i}{\|p_i(0) + n_i\|_2}, \qquad \mathcal{P}(1) = \{\tilde{p}_i\}_{i=1}^N.$$

We then move each point *along* the surface via per-point spherical linear interpolation (SLERP) between corresponding pairs:

$$p_i(\rho) = \mathrm{slerp}\big(p_i(0), \tilde{p}_i; \rho\big) = \frac{\sin((1-\rho)\omega_i)}{\sin\omega_i}\, p_i(0) + \frac{\sin(\rho\,\omega_i)}{\sin\omega_i}\, \tilde{p}_i, \quad \omega_i = \arccos\big(\langle p_i(0), \tilde{p}_i\rangle\big).$$

This yields the cloud path $\mathcal{P}(\rho) = \{p_i(\rho)\}_{i=1}^N$ used throughout.

Decoding and component counting. Given $\mathcal{P}(\rho)$, we compute a conditioning code via the model's encoder and sample a latent with the diffusion sampler to yield $G(\mathcal{P}(\rho))$, i.e. a `mesh`. We report

$$C(\rho) = \texttt{len}(\texttt{trimesh.split(mesh)}),$$

i.e., the number of connected components in `trimesh`.

Grid over the control parameter. We sweep a uniform grid of ρ values, i.e., $\rho \in \{0, 0.05, \dots, 1.0\}$.

Connectivity curve $C(\rho)$. We evaluate $C(\rho)$ on the uniform ρ grid. For each ρ, we reseed the RNGs to reproduce the identical terminal noise x_T. The curve reported is the set

$$\{(\rho,\, C(\rho))\}_{\rho \in \{0, 0.05, \dots, 1.0\}},$$

from which the observed plateau at $C(\rho) = 1$ and the subsequent jump to $C(\rho) > 1$ over a narrow ρ-interval (a connectivity bifurcation) are directly obtained.

Spectral entropy curve $H(\rho)$. We evaluate the spectral entropy of the localized cross–attention write on the same uniform control grid $\rho \in \{0, 0.05, \dots, 1.0\}$ and with identical terminal noise across ρ.

For each ρ, we encode the cloud $\mathcal{P}(\rho)$, run a single sampling trace, and read out the token-wise cross-attention write at the chosen site, $\mathbf{Y}(\rho)$. Let $\{\sigma_i(\rho)\}_i$ be the singular values of $\mathbf{Y}(\rho)$ (SVD of the matrix with shape tokens $\times$ features). We form normalized directional energies

$$p_i(\rho) \;=\; \frac{\sigma_i(\rho)^2}{\sum_j \sigma_j(\rho)^2}, \qquad H(\rho) \;=\; -\sum_i p_i(\rho)\, \log p_i(\rho),$$

using the natural logarithm. The reported curve is the set

$$\{ (\rho,\ H(\rho)) \}_{\rho \in \{0, 0.05, \ldots, 1.0\}}.$$

B.2.2 WaLa, DDIM

Key hyperparameters.

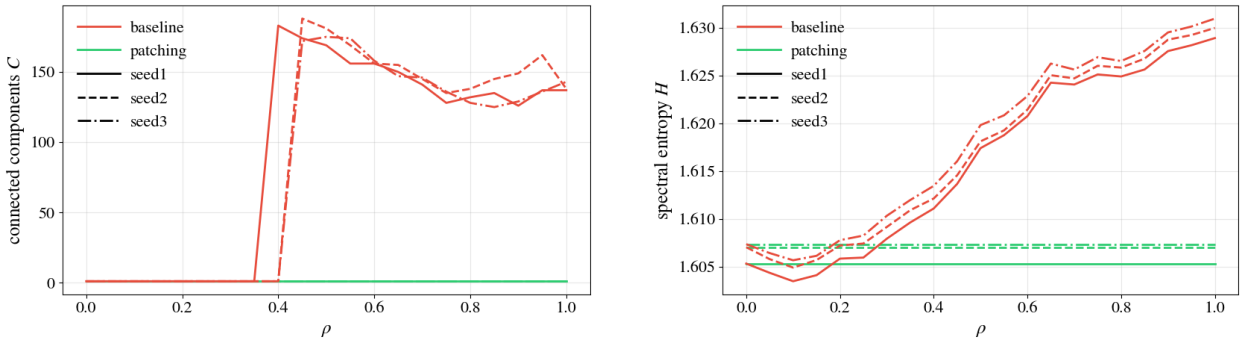
Model	`ADSKAILab/WaLa-PC-1B`
Sampler	DDIM ($\eta = 0$)
Diffusion rescale	8 steps (`diffusion_rescale_timestep=8`), i.e., default
CFG weight	1.0 (`scale=1.0`), i.e., we consider only conditional stream
Points per cloud	$N = 400$
Cloud source	Unit sphere, golden-angle placement
Target cloud	Gaussian jitter $\sigma = 0.1$ on $\mathbb{R}^3$, renormalize to $\mathbb{S}^2$
Interpolation	Per-point SLERP, control $\rho \in [0, 1]$
ρ grid	21 values: $0, 0.05, \ldots, 1.0$
Seeds	$0 - 1000$ for all RNG calls
Order parameter	$C(\rho) = \#$ connected components (`trimesh.split`)
Device	`cuda` (CPU is functionally equivalent but slower)

B.2.3 WaLa, DDPM

The activation-patching grid for WaLa under DDPM is equivalent to Figure 2, i.e., WaLa under DDIM. The corresponding curves can be found in Figure 7.

Key hyperparameters.

Model	`ADSKAILab/WaLa-PC-1B`
Sampler	DDPM
Diffusion rescale	8 steps (`diffusion_rescale_timestep=8`)
CFG weight	1.0 (`scale=1.0`) (we consider only conditional stream)
Points per cloud	$N = 400$
Cloud source	Unit sphere, golden-angle placement
Target cloud	Gaussian jitter $\sigma = 0.1$ on $\mathbb{R}^3$, renormalize to $\mathbb{S}^2$
Interpolation	Per-point SLERP, control $\rho \in [0, 1]$
ρ grid	21 values: $0, 0.05, \ldots, 1.0$
Seeds	0 for all RNG calls
Order parameter	$C(\rho) = \#$ connected components (`trimesh.split`)
Device	`cuda` (CPU is functionally equivalent but slower)



(a) Connected components C vs. ρ

(b) Spectral entropy H vs. ρ

Figure 7: [WaLa, DDPM]. Our results from WaLa under DDIM sampling transfer to WaLa under DDPM sampling.

B.2.4 Make-A-Shape, DDIM

We report the result for the activation search procedure for Make-A-Shape under DDIM in Figure 8. The corresponding curves are depicted in Figure 9.

Key hyperparameters.

Model	`ADSKAILab/Make-A-Shape-point-cloud-20m`
Sampler	DDIM
Diffusion rescale	100 steps (`diffusion_rescale_timestep=8`), i.e., default
CFG weight	1.0 (`scale=1.0`), i.e., we consider only conditional stream
Points per cloud	$N = 1200$
Cloud source	Unit sphere, golden-angle placement
Target cloud	Gaussian jitter $\sigma = 0.1$ on $\mathbb{R}^3$, renormalize to $\mathbb{S}^2$
Interpolation	Per-point SLERP, control $\rho \in [0, 1]$
ρ grid	21 values: $0, 0.05, \ldots, 1.0$
Seeds	0 for all RNG calls
Order parameter	$C(\rho) = \#$ connected components (`trimesh.split`)
Device	`cuda` (CPU is functionally equivalent but slower)

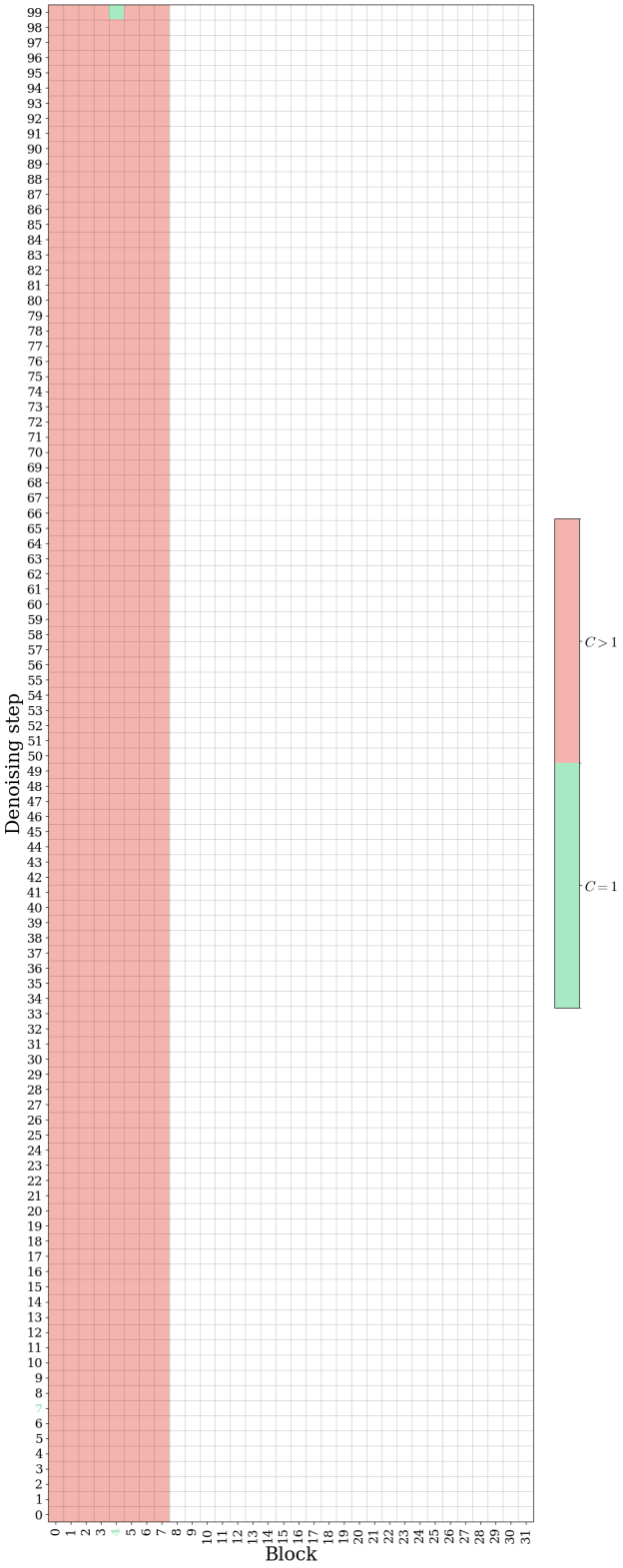
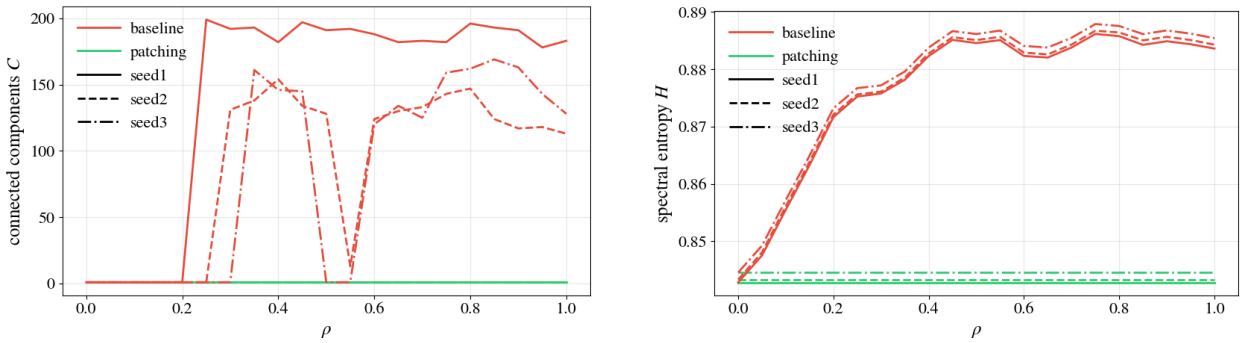


Figure 8: Activation-patching result for Make-A-Shape. Analogous to our result for WaLa, we find an early denoising cross-attention activation that controls Meltdown behavior.



(a) Connected components C vs. ρ

(b) Spectral entropy H vs. ρ

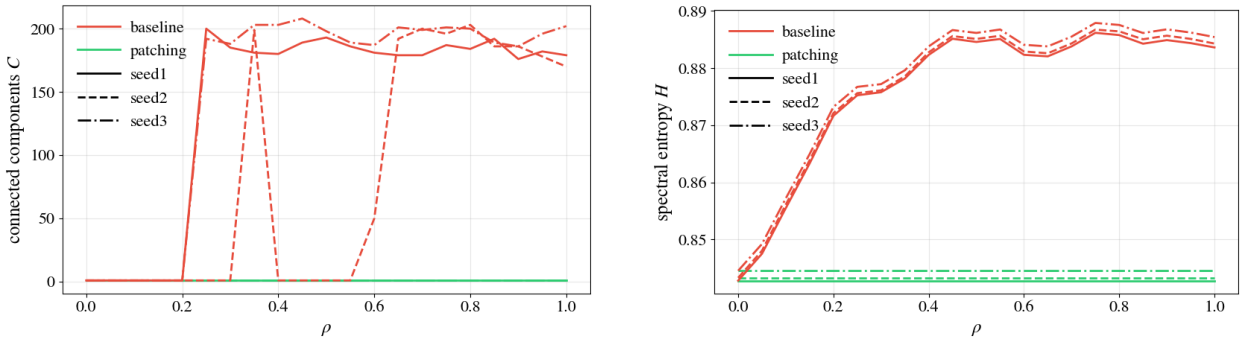
Figure 9: [Make-A-Shape, DDIM]. Our results from WaLa transfer to Make-A-Shape: As we move from a healthy to an unhealthy run, we observe that the baseline case shows a smooth rise in spectral entropy and a sudden jump in connectivity. Patching our $\mathbf{Y}$ keeps the spectral entropy at healthy levels and preserves connectivity. This behavior is consistent across diffusion seeds.

B.2.5 Make-A-Shape, DDPM

The activation-patching grid for Make-A-Shape under DDPM is equivalent to Figure 8, i.e., Make-A-Shape under DDIM. The corresponding curves can be found in Figure 10.

Key hyperparameters.

Model	`ADSKAILab/Make-A-Shape-point-cloud-20m`
Sampler	DDPM
Diffusion rescale	100 steps (`diffusion_rescale_timestep=8`), i.e., default
CFG weight	1.0 (`scale=1.0`) (we consider only conditional stream)
Points per cloud	$N = 1200$
Cloud source	Unit sphere, golden-angle placement
Target cloud	Gaussian jitter $\sigma = 0.1$ on $\mathbb{R}^3$, renormalize to $\mathbb{S}^2$
Interpolation	Per-point SLERP, control $\rho \in [0, 1]$
ρ grid	21 values: $0, 0.05, \ldots, 1.0$
Seeds	0 for all RNG calls
Order parameter	$C(\rho) = \#$ connected components (`trimesh.split`)
Device	`cuda` (CPU is functionally equivalent but slower)



(a) Connected components C vs. ρ

(b) Spectral entropy H vs. ρ

Figure 10: [Make-A-Shape, DDPM]. Our results from [Make-A-Shape under DDIM sampling transfer to WaLa under DDPM sampling.

B.3 Dataset Evaluation

This section provides (i) evidence that Meltdown exists across a variety of shapes, i.e., across the GSO (Downs et al., 2022) and SimJEB (Whalen et al., 2021) corpora, and diffusion transformers, i.e., WaLa and Make-A-Shape. Furthermore it details our setup to evaluate our method `PowerRemap` on GSO and SimJEB as well as the results of these evaluations.

General. We evaluate `PowerRemap` on the WaLa and Make-A-Shape architectures, using DDIM sampling (Song et al., 2021). For each object, we load the corresponding mesh as the ground-truth surface $\mathcal{S} \subset \mathbb{R}^3$, center it and scale it to the unit cube.

Find Meltdown. We reuse the notation of §2. Given a mesh $\mathcal{S}$ and generator G, we first determine a sparse point budget by searching the smallest N over a grid for which a Poisson-disk sample $A = \{a_i\}_{i=1}^N \subset \mathcal{S}$ yields a healthy output $C(G(A)) = 1$. We then define a surface-constrained Meltdown path by jittering A and projecting back to $\mathcal{S}$ to obtain $B = \{b_i\}_{i=1}^N$, and interpolate on-manifold

$$\mathcal{P}_\rho \;=\; \Pi_{\mathcal{S}}\big((1-\rho)A + \rho B\big), \quad \rho \in [0,1],$$

where $\Pi_{\mathcal{S}}$ is nearest-point projection. With a fixed random seed (reseeded before every inference), we sweep ρ on a geometric grid to bracket a jump in connectivity, then refine by bisection to the smallest ε such that $C(G(\mathcal{P}_\varepsilon)) \gg 1$.

Algorithm 1 Adversarial Meltdown Search

Require: surface $\mathcal{S}$, generator G, component counter $C(\cdot)$, seed $s = 0$
1: normalize $\mathcal{S}$; find smallest N s.t. $A \sim \text{Poisson}(\mathcal{S}, N)$ gives $C(G(A))=1$ (reseed s)
2: $B \leftarrow \Pi_{\mathcal{S}}(A + \xi)$ ▷ jitter & project
3: sweep ρ on a geometric grid; find $\rho_{\text{lo}} < \rho_{\text{hi}}$ with $C(\rho_{\text{lo}})=1$, $C(\rho_{\text{hi}})>1$
4: $\varepsilon \leftarrow \text{bisection}(\rho_{\text{lo}}, \rho_{\text{hi}})$ with reseeding to s
5: **Return** $(N, \varepsilon,\ C_0{=}C(G(A)),\ C_\varepsilon{=}C(G(\mathcal{P}_\varepsilon)))$

Evaluate `PowerRemap`. The task of reconstructing a global surface from a sparse point cloud has only two possible outcomes: success or failure. Thus, we assess the effectiveness `PowerRemap` by counting the number of times it succeeded in reducing C_ε to 1, i.e., turning a speckle into a shape. Hence, we treat each shape as a Bernoulli trial under our adversarial search (Algorithm 1). Each trial has an outcome $p \in \{0, 1\}$, where $p = 1$ iff the reconstruction meets the criterion $C_\varepsilon = 1$; otherwise $p = 0$. We first identify baseline failures as those with $C_{0,\text{baseline}} = 1$ and $C_{\varepsilon,\text{baseline}} > 1$. We then apply `PowerRemap` only to these failures and count a remedy when $C_{\varepsilon,\texttt{PowerRemap}} = 1$.

B.3.1 GSO

This section provides a quantitative evaluation of the Meltdown phenomenon on the GSO dataset (Downs et al., 2022) and assesses the effectiveness of `PowerRemap` as a mitigation strategy. GSO (Downs et al., 2022) is a diverse corpus of 1,030 scanned household objects and was *not* used to train either WaLa or Make-a-Shape. All results in this section are obtained by applying the protocol described in Appendix B.3 to the GSO dataset.

Evaluate `PowerRemap`. For WaLa, we found Meltdown in 926/1,030 (89.9%) shapes. Our method `PowerRemap` remedies failure in 910/926 (98.3%) for $\gamma = 100$. Table 1 (top) depicts the performance of our method across all shape categories. For Make-A-Shape, we consider a category-representative subset of 130 GSO shapes and find Meltdown in 130/130 (100%) shapes. We evaluate `PowerRemap` over a γ-grid for the given subset, where

$$\gamma \in \{1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.5, 2\}.$$

For 110 out of 130 shapes (84.6%), we find at least one value of γ in this range that successfully remedies Meltdown. Among the rescued cases, the median effective value of γ is 1.10 with a standard deviation of 0.13. Table 1 (top) depicts the performance of our method across the representative subset. In Appendix B.8, we empirically study the influence of the `PowerRemap` strength γ on reconstruction connectivity, concluding that the optimal hyperparameter is model-dependent.

B.3.2 SimJEB

This section provides a quantitative evaluation of the Meltdown phenomenon on the SimJEB dataset (Whalen et al., 2021) and assesses the effectiveness of `PowerRemap` as a mitigation strategy. SimJEB is a curated benchmark of 381 3D jet-engine bracket CAD models that was *not* included in the training data of either WaLa or Make-a-Shape. All results in this section are obtained by applying the protocol described in Appendix B.3 to the SimJEB dataset.

Evaluate `PowerRemap` For WaLa, we found Meltdown in 352 out of 381 shapes (92.4%). Our method `PowerRemap` remedies failure in 344/352 (97.7%) for $\gamma = 100$. Table 1 (bottom) depicts the performance of our method across all shape categories. For Make-A-Shape, we consider a category-representative subset of 30 SimJEB shapes and find Meltdown in 30/30 (100%) shapes. We evaluate `PowerRemap` over a γ-grid for the given subset, where

$$\gamma \in \{1.05, 1.1, 1.15, 1.2, 1.25, 1.3, 1.35, 1.4, 1.5, 2\}.$$

For 25 out of 30 shapes (83.3%), we find at least one value of γ in this range that successfully remedies Meltdown. Across the rescued cases, the median effective γ is 1.05 with a standard deviation 0.063. Table 1 (bottom) depicts the performance of our method across the representative subset. In Appendix B.8, we empirically study the influence of the `PowerRemap` strength γ on reconstruction connectivity, concluding that the optimal hyperparameter is model-dependent.

B.4 Density

In this section, we examine how the prevalence of Meltdown depends on the sparsity of the input point cloud. We quantify the sparsity of an input point cloud of size N by its *areal density*

$$\eta = \frac{N}{A_{\mathcal{S}}}, \tag{10}$$

where $A_{\mathcal{S}}$ denotes the surface area of the underlying surface $\mathcal{S}$. Higher values of η correspond to denser samplings of $\mathcal{S}$.

In Figure 11, we examine how the prevalence of Meltdown depends on η for SimJEB shape 492. For each target areal density, we run Algorithm 1 (parameterized by η) for 10 independent trials and record how often a Meltdown configuration is identified. We observe that Meltdown is particularly frequent in the low-η regime.

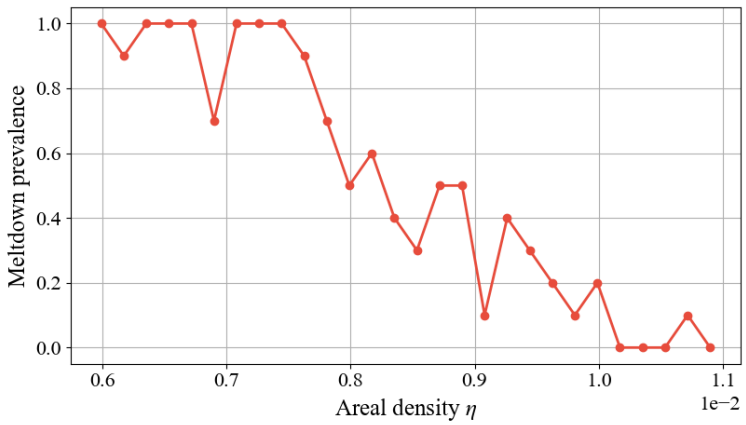
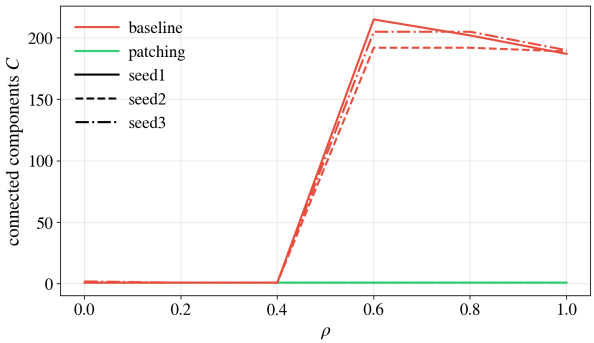
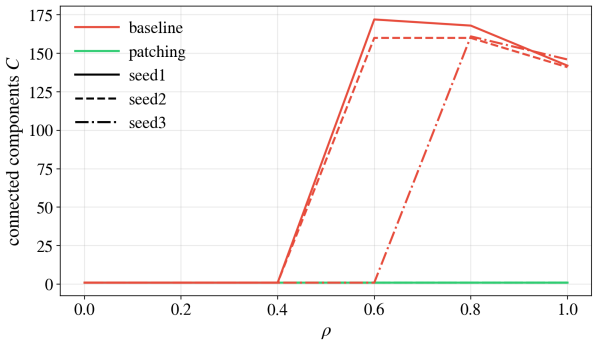


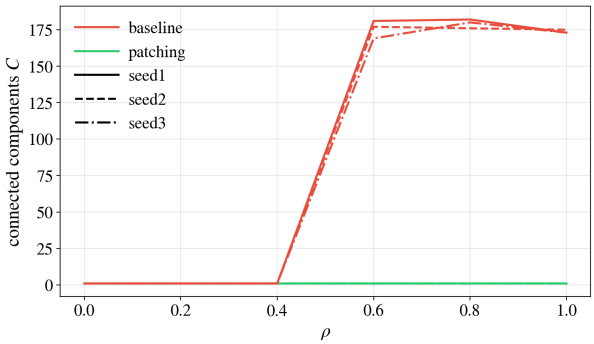
Figure 11: Incidence of Meltdown on SimJEB shape 492 as a function of areal density η. Meltdown events are especially common for low areal densities, underscoring the difficulty of robust surface reconstruction from sparse point clouds.



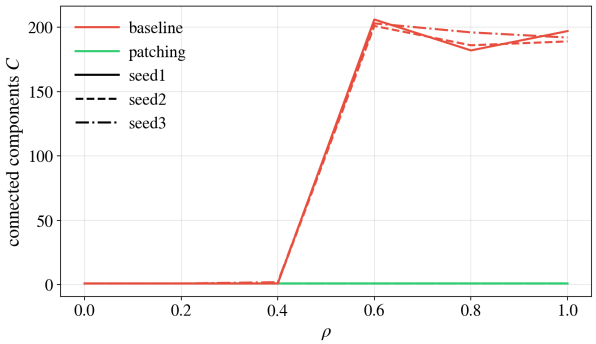
(a) Shape 1 of Figure 4 (GSO)



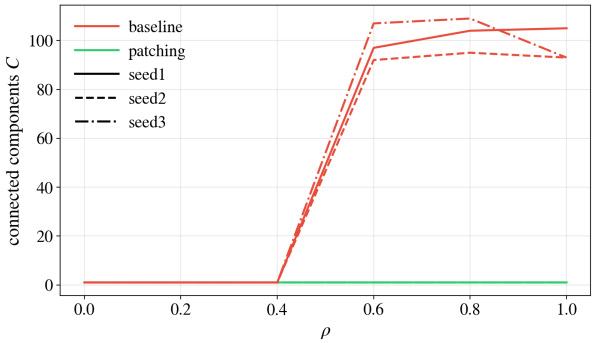
(b) Shape 2 of Figure 4 (GSO)



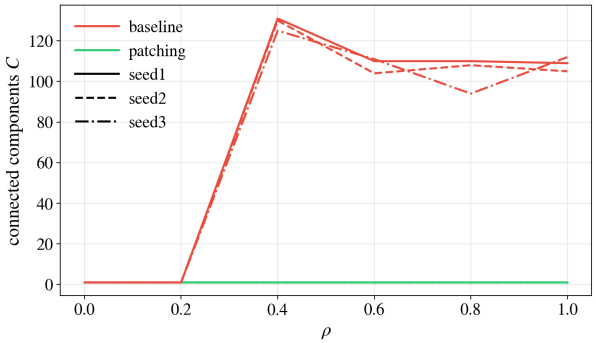
(c) Shape 3 of Figure 4 (GSO)



(d) Shape 4 of Figure 4 (GSO)



(e) Shape 492 (SimJEB)

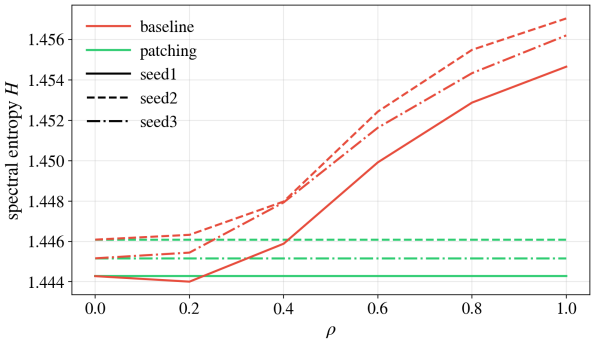


(f) Shape 525 (SimJEB)

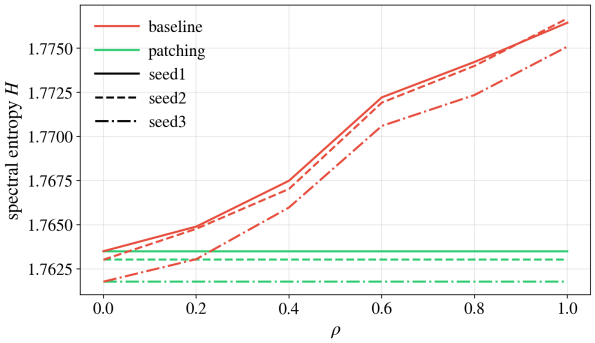
Figure 12: Connected components C vs. ρ.

B.5 More Datapoints

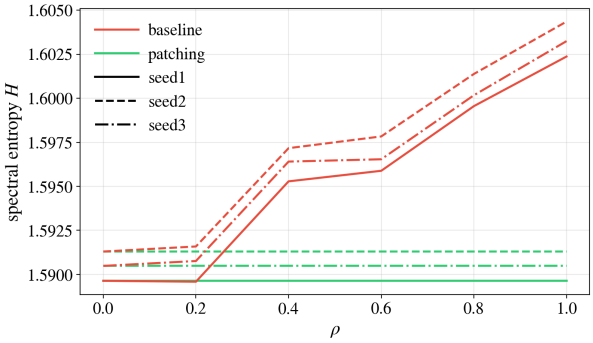
In this section, we provide additional evidence that the patterns observed in Section 3.2-3.3 generalize when evaluated on more data points and random seeds. Figure 12 and Figure 13 show that the behavior transfers to diverse shapes from the GSO (Downs et al., 2022) and SimJEB (Whalen et al., 2021) corpora as well as diffusion seeds for the WaLa model. Figure 14 shows that the average behavior over a population of 150 diffusion seeds for SimJEB shape 492 is consistent with the observations reported in Section 3.3 for the WaLa model.



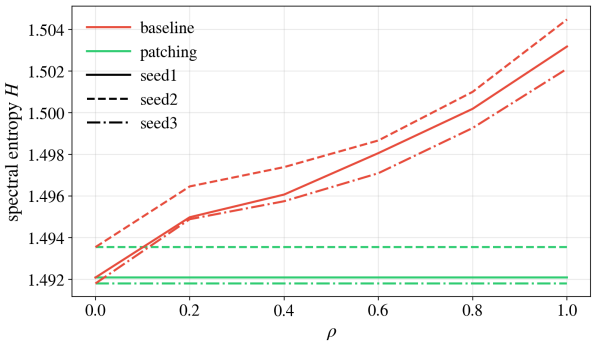
(a) Shape 1 of Figure 4 (GSO)



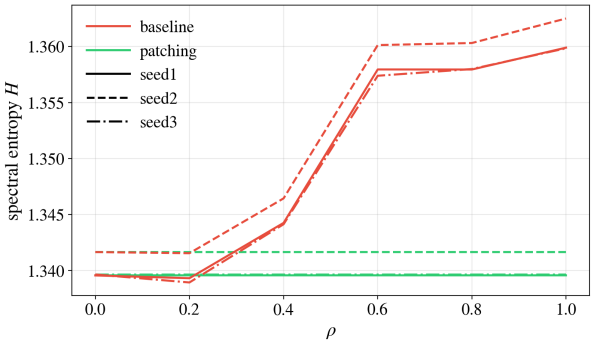
(b) Shape 2 of Figure 4 (GSO)



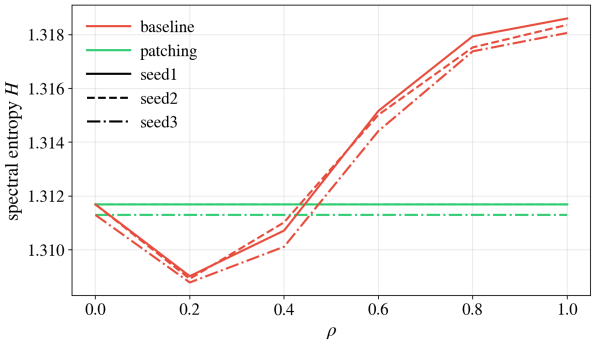
(c) Shape 3 of Figure 4 (GSO)



(d) Shape 4 of Figure 4 (GSO)

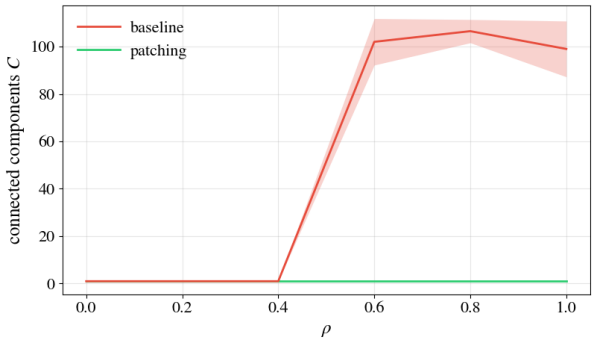


(e) Shape 492 (SimJEB)

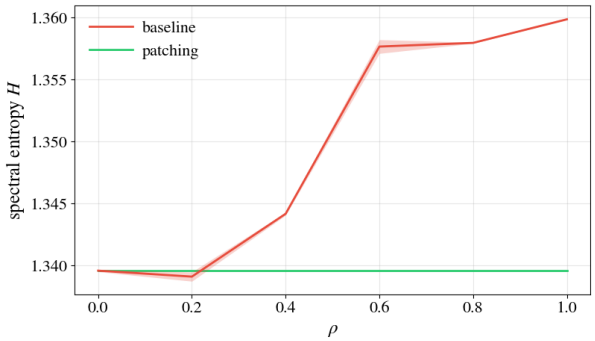


(f) Shape 525 (SimJEB)

Figure 13: Spectral entropy H vs. ρ.

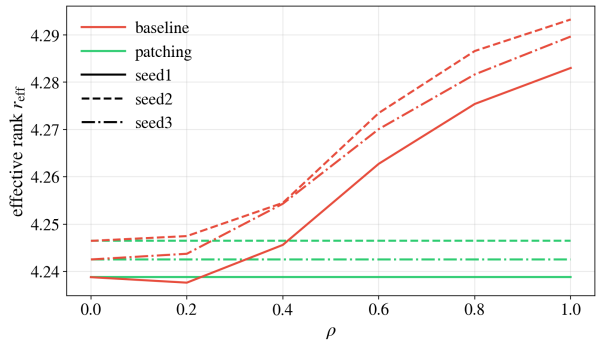


(a) Connected components C vs. ρ

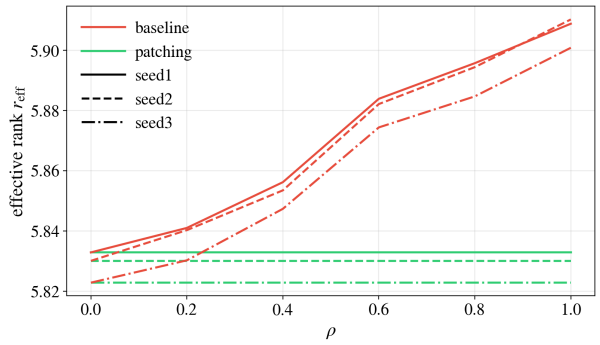


(b) Spectral entropy H vs. ρ

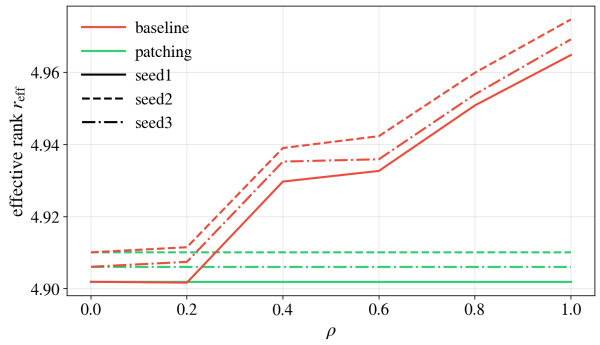
Figure 14: Patterns at population level for SimJEB shape 492, using 150 diffusion seeds.



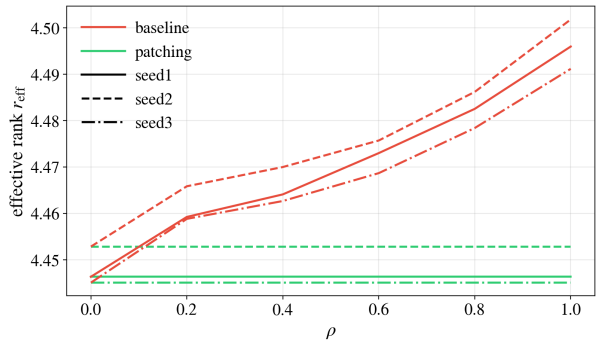
(a) Shape 1 of Figure 4 (GSO)



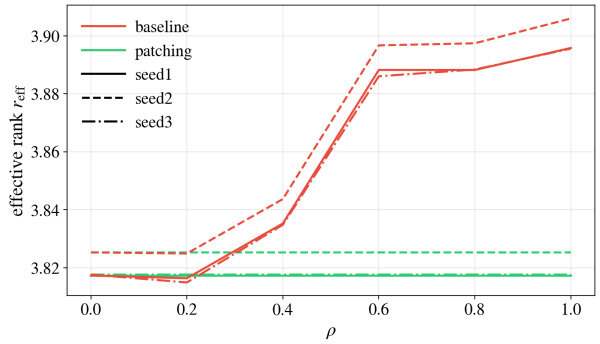
(b) Shape 2 of Figure 4 (GSO)



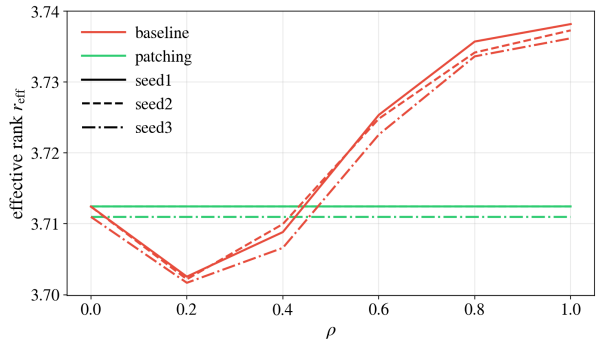
(c) Shape 3 of Figure 4 (GSO)



(d) Shape 4 of Figure 4 (GSO)



(e) Shape 492 (SimJEB)

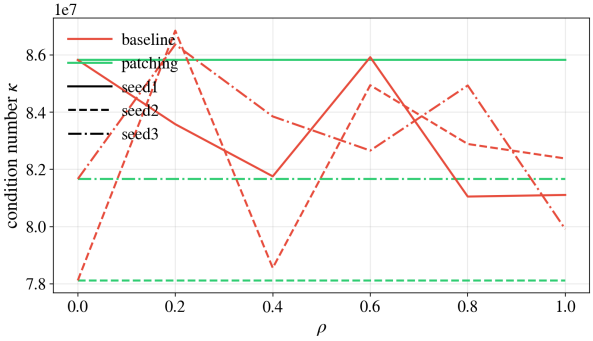


(f) Shape 525 (SimJEB)

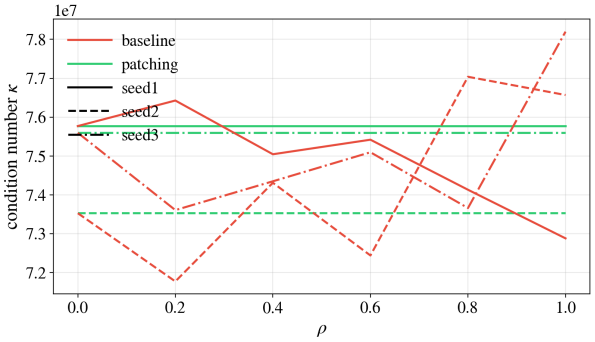
Figure 15: Effective rank r_{eff} vs. ρ.

B.6 Additional Spectral Metrics

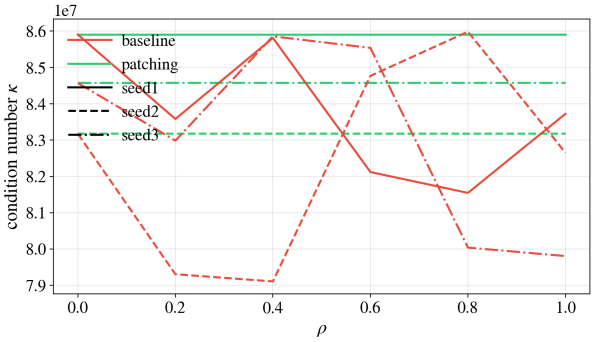
In this section, we analyze additional spectral metrics to assess their suitability as indicators of Meltdown for the WaLa model. In particular, Figure 15 reports the effective rank $r_{\text{eff}} = \exp(H)$ and Figure 16 the condition number $\kappa = \sigma_{\max}/\sigma_{\min}$ as alternatives to spectral entropy for a diverse set of shapes. We observe that the effective rank—which is a monotonic transformation of spectral entropy—provides an equally informative indicator of Meltdown. By contrast, the condition number exhibits no apparent correlation with the failure phenomenon, suggesting that it is not a suitable diagnostic metric in this setting.



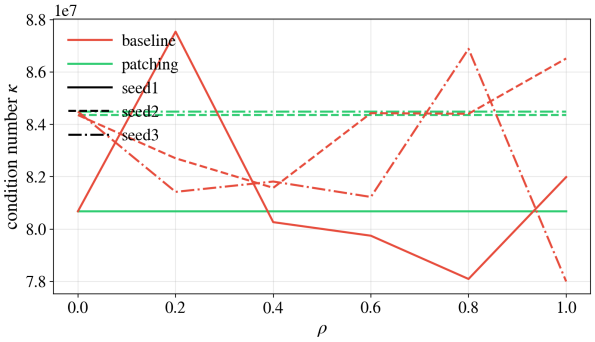
(a) Shape 1 of Figure 4 (GSO)



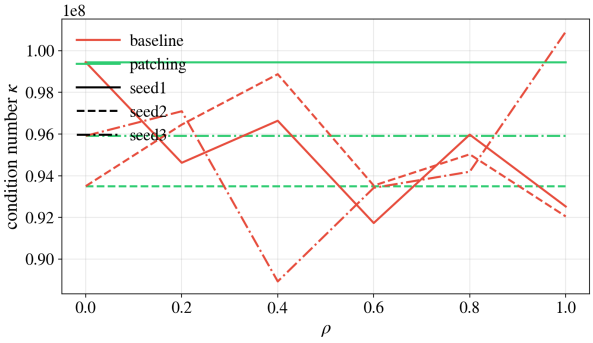
(b) Shape 2 of Figure 4 (GSO)



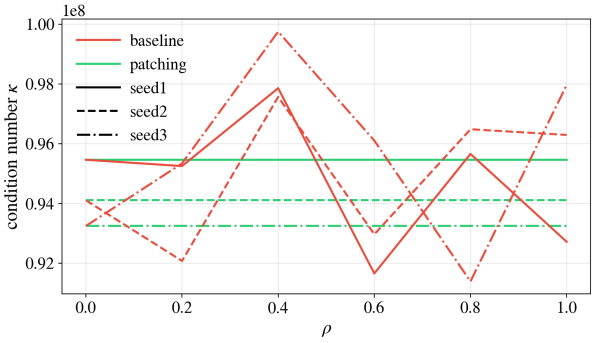
(c) Shape 3 of Figure 4 (GSO)



(d) Shape 4 of Figure 4 (GSO)



(e) Shape 492 (SimJEB)



(f) Shape 525 (SimJEB)

Figure 16: Condition number κ vs. ρ.

B.7 Multiple Objects

We further assess whether the Meltdown phenomenon and the effectiveness of `PowerRemap` extend beyond single-object inputs. Figure 17 provides a qualitative evaluation on a scene containing multiple objects for the WaLa model. We observe that Meltdown still occurs in this multi-object setting, while using `PowerRemap` reliably suppresses the failure and preserves a plausible reconstruction of all objects in the scene.

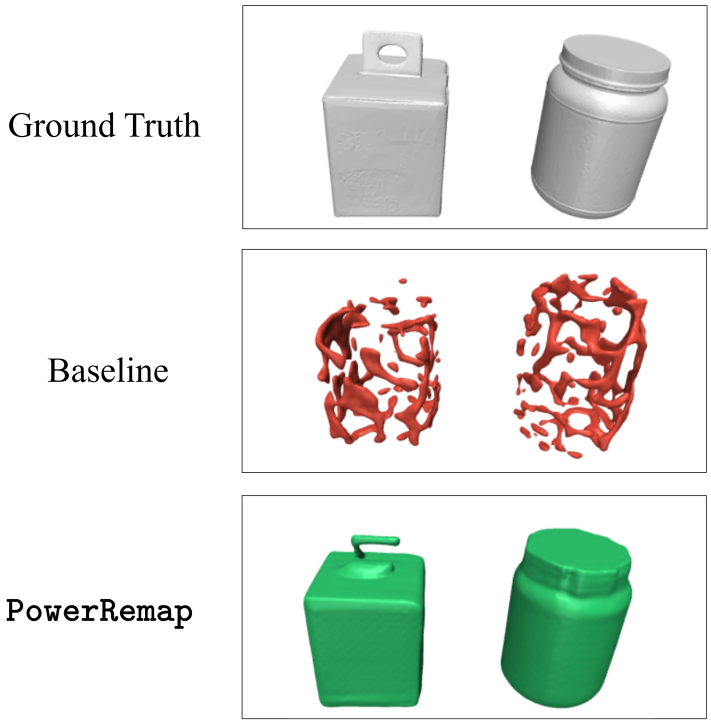


Figure 17: Qualitative evaluation of multi-object inputs. Meltdown persists in scenes with multiple objects, leading to severe degradation of the reconstruction, whereas `PowerRemap` effectively prevents this failure mode and yields a stable reconstruction of all objects.

B.8 Examining `PowerRemap` strength γ on Reconstruction Connectivity

We empirically investigate the influence of the `PowerRemap` strength γ on reconstruction connectivity. Overall, we conclude that the optimal hyperparameter γ is model-dependent.

WaLa We run Algorithm 1 on SimJEB (WaLa) shape 492 for 10 independent random seeds and select the Meltdown configuration with $C_\varepsilon > 1$. We then apply `PowerRemap` to this configuration over the hyperparameter grid $\gamma \in \{1, 2, 5, 10, 100\}$, where $\gamma = 1$ denotes the identity mapping. As can be seen in Figure 18, our `PowerRemap` method achieves a high success rate for $\gamma > 2$.

Make-a-Shape For Make-a-Shape, we investigate the distribution of `PowerRemap` strengths γ for the 130-shape subset of GSO as discussed in Table 1 (top). We find that γ values around 1.05 are effective to remedy Meltdown, as illustrated in Figure 19.

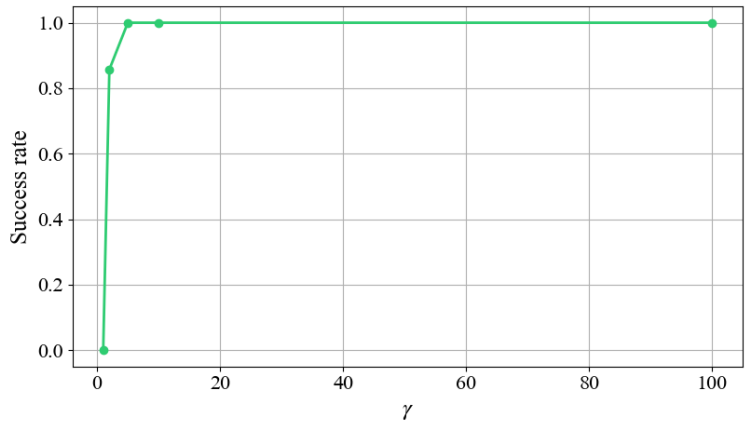


Figure 18: `WaLa`. We find that a `PowerRemap` strength of $\gamma > 2$ remedies Meltdown.

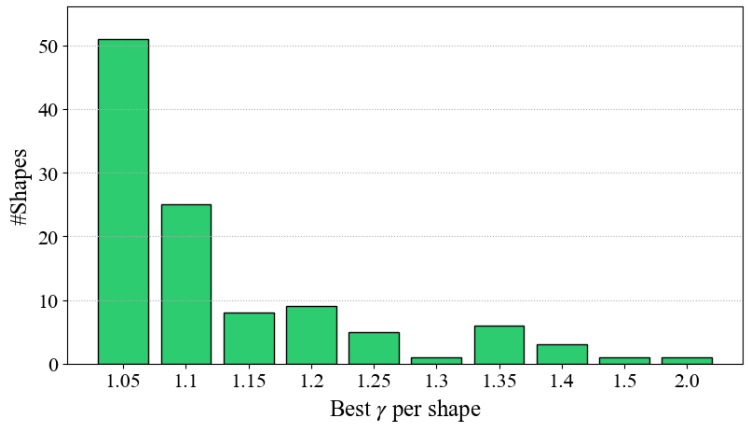


Figure 19: Make-a-Shape. For the representative subset in Table 1 (top), we find that γ values around 1.05 are effective to remedy Meltdown.

B.9 Examining `PowerRemap` on Non-Meltdown Cases

We empirically verify that `PowerRemap` does not interfere with non-Meltdown runs. Specifically, we run Algorithm 1 on SimJEB (WaLa) shape 492 for 10 independent random seeds and select a configuration with $C_0 = 1$ (a single connected component). We then apply `PowerRemap` to this configuration over the hyperparameter grid $\gamma \in \{2, 5, 10, 100\}$. As can be seen in Table 2, the reconstructed surface retains $C_0 = 1$ in all cases and we do not observe any change in the final topology. The results in Table 2 indicate that `PowerRemap` is effectively topologically neutral on this non-Meltdown instance.

Table 2: Evaluation of `PowerRemap` on a non-Meltdown SimJEB shape (shape 492). Across 10 random seeds and $\gamma \in \{2, 5, 10, 100\}$, `PowerRemap` preserves the original topology ($C_0 = 1$) in all cases.

Shapes	**Seeds**	**Meltdown occurs** [%]	**Topology preserved** [%]
SimJEB 492	10	0.0	100.0
Total	**10**	**0.0**	**100.0**

C Spectral analysis of the cross-attention write $\mathbf{Y}$

This appendix details the protocols and full numerical results behind the Make-a-Shape and WaLa paragraphs in the main text. The two models are analyzed with the same target (the block-4, first-denoising-step cross-attention write $\mathbf{Y}$), but their architectural redundancy differs, and the ablations that cleanly localize Meltdown differ accordingly. We first describe the shared protocol (§C.1), then establish the architectural contrast (§C.2), then present the Make-a-Shape experiment where direct spectral ablation is informative (§C.3), and finally the WaLa experiments where direct ablation is confounded and a drift-targeted protocol is required (§C.6).

C.1 Shared protocol

For each model, we identify the cross-attention write at block 4 on the first denoising step — $\mathbf{Y} \equiv \mathbf{Y}_{4,7}$ for WaLa (T=8 DDIM steps) and $\mathbf{Y} \equiv \mathbf{Y}_{4,99}$ for Make-a-Shape (T=100 DDIM steps), following the localization in the main text. All interventions are single-forward-pass: we register a forward hook on the corresponding sub-module (`unet.blocks[4].cross_attn.proj` in WaLa; `unet.middle_blocks[4][0].att` in Make-a-Shape) that overwrites the output at the target denoising step and lets the rest of the reverse process run unmodified.

Each intervention is evaluated on 100 diffusion seeds ($\{0, \ldots, 99\}$) at the model-specific phase-transition point: ρ_{melt}=0.4 for WaLa and ρ_{melt}=0.5 for Make-a-Shape. Rescue is declared when the output mesh has a single connected component (C=1); for WaLa, we additionally require the chamfer distance to the healthy reference to be at least 0.01 below the unintervened baseline, to distinguish genuine sphere recovery from trivial rank collapse. The baselines (unintervened at ρ_{melt}, averaged over the 100 seeds) are: WaLa: $\bar{C}$=143.9, $\overline{\text{chamfer}}$=0.1122, rescue rate 0/100; Make-a-Shape: $\bar{C}$=124.9, rescue rate 25/100 (stochasticity at the bifurcation is larger for MAS; see main text §5 on ensemble behavior).

C.2 Architectural contrast

The two models expose the encoded condition $\mathbf{C}$ to the denoising backbone through different numbers of sites.

WaLa. The denoiser is a `Latent_UVIT` (`network._orig_mod.unet`) whose main body is a `ModuleList` of 32 sequential `Cross_DiTBlock`s. Each block contains, in order of the residual stream update:

- a self-attention layer (`attn`) whose pre-norm (`LayerNorm(1152, elementwise_affine=False)`) is AdaLN-modulated;
- an MLP ($1152 \to 4608 \to 1152$) whose pre-norm is also AdaLN-modulated;
- a cross-attention layer (`cross_attn`, the site we study) reading $\mathbf{C}$ as keys/values.

The `adaLN_modulation` submodule in each block is `Linear(1152,6912)`, producing 6×1152 scale/shift/gate parameters — scale, shift, and gate for both the attention and the MLP norms — from $\mathbf{C}$-derived features (main text §3.1). Across all 32 blocks, the condition therefore enters the residual stream through 32 cross-attention writes *and* 32 AdaLN-modulation sites. A single `cross_attn` output is one of these 64 depth-wise injection points.

Make-a-Shape. The denoiser is a `Condition_UVIT` (`network.unet`). Its bottleneck is structured as two parallel module lists of length 8: `self_middle_blocks` (self-attention) and `middle_blocks` (cross-attention to $\mathbf{C}$). Each `Transformer_Block` in `middle_blocks` contains an attention layer (reading $\mathbf{C}$ as keys/values) and an MLP with FiLM-style scale and shift produced by two `Linear(256,512)` submodules from a 256-dimensional embedding. The condition therefore enters the backbone explicitly through the 8 cross-attention layers of `middle_blocks`; no per-block $\mathbf{C}$-dependent AdaLN is present.

Implication. Zeroing the output of a single `cross_attn` removes 1/32 of the direct cross-attention pathways in WaLa and 1/8 in Make-a-Shape, with the former further buffered by 32 independent AdaLN-modulation sites. This 4× (or larger, counting AdaLN) redundancy asymmetry is the structural reason a single-write ablation is informative in Make-a-Shape but confounded in WaLa, as the experiments below confirm.

C.3 Make-a-Shape: spectral ablation of $\mathbf{Y}$

Protocol. For each seed we SVD-decompose $\mathbf{Y}{=}U\Sigma V^\top$ at the target site and replace $\mathbf{Y}$ with one of two rank-modified versions for $m \in \{1, 5, 10, 20, 50, 100\}$:

- **keep-top-m**: retain only the m largest components, $\mathbf{Y}^{\text{keep}}_{(m)}{=}\sum_{i\le m}\sigma_i u_i v_i^\top$ (zeros the tail);
- **zero-top-m**: retain only the tail, $\mathbf{Y}^{\text{zero}}_{(m)}{=}\sum_{i>m}\sigma_i u_i v_i^\top$ (zeros the top).

These are run at every m across all 100 seeds (1,200 runs plus 100 unintervened baseline).

Results. Rescue counts out of 100 seeds (baseline: 25/100):

Table 3: Make-a-Shape spectral ablation at $\rho_{\text{melt}}{=}0.5$; 100 seeds. Mean connected-component count $\bar{C}$ in parentheses.

ablation \ m	1	5	10	20	50	100
keep-top-m (zero tail)	26 (127.5)	**99** (2.75)	**99** (2.67)	61 (64.1)	25 (124.8)	25 (125.0)
zero-top-m (zero top)	8 (156.8)	4 (165.3)	5 (165.5)	8 (158.9)	8 (158.9)	8 (158.7)

Three observations. First, keep-top-m is strongly non-monotonic: at $m{=}1$ rescue (26/100) matches baseline (25/100) — the rank-1 write carries too little to condition the denoiser; at $m \in \{5, 10\}$ rescue jumps to 99/100 with $\bar{C}{\approx}2.7$ (essentially a clean sphere); at $m \in \{50, 100\}$ rescue returns *exactly* to baseline with $\bar{C}$ matching the unintervened distribution, meaning the intervention has no effect at these ranks. This last fact implies the effective rank of $\mathbf{Y}$ at this site is $\le$50: retaining the top-50 of the spectrum is equivalent to retaining the entire spectrum.

Second, zero-top-m is uniformly destructive: rescue is 4–8%, strictly below baseline, with $\bar{C} \in [156.8, 165.5]$ exceeding even the unintervened $\bar{C}{=}124.9$. Removing the top of the spectrum makes outcomes worse than doing nothing.

Third, combining these: because keep-top-50 already reproduces baseline, zero-top-m at $m{\ge}50$ effectively sets $\mathbf{Y}{=}\mathbf{0}$, and yields only 8/100 rescues. In Make-a-Shape, zeroing $\mathbf{Y}$ is actively harmful. This is consistent with the architectural analysis in §C.2: $\mathbf{Y}$ is one of only 8 cross-attention writes into the backbone and has no redundant AdaLN fallback.

Interpretation. The two ablations mirror each other on opposite halves of the spectrum. Removing the tail while keeping an informative top rescues Meltdown; removing the top (with or without the tail) is uniformly worse than doing nothing. The causal asymmetry is clean: the *tail* of $\mathbf{Y}$ carries the Meltdown-inducing signal, while the *dominant* directions carry the conditioning information the model cannot spare.

C.4 WaLa: direct spectral ablation is non-informative

Keep-top-m on Y's own basis. Using the same truncation protocol as §C.3 for $m \in \{1, 2, 3, 5, 10, 15, 20, 25, 30, 40, 50, 75, 100, 150, 200\}$:

At every $m{\ge}3$, rescue is 0/100 and both $\bar{C}$ and mean chamfer are indistinguishable from the unintervened baseline. The cases $m{\in}\{1, 2\}$ rescue at 62/100 and 73/100, but these are degenerate: at $m{=}1$ the mean

Table 4: WaLa keep-top-m on $\mathbf{Y}$'s own basis at ρ_{melt}=0.4; 100 seeds. Baseline $\bar{C}$=143.9, $\overline{\text{chamfer}}$=0.1122.

m	1	2	3	5	10	20	30	50	100	200
rescue	62	73	0	0	0	0	0	0	0	0
$\bar{C}$	1.06	25.7	142.8	143.6	144.0	144.0	143.6	143.7	143.9	144.0
$\overline{\text{cham}}$	.100	.100	.113	.112	.112	.112	.112	.112	.112	.112

chamfer is 0.0996, closer to the $\mathbf{Y}$=0 output than to the healthy reference (0.0887); a rank-1 write carries essentially no content and the behaviour reduces to the full $\mathbf{Y}$=0 ablation. The Make-a-Shape-style tail-removal protocol therefore does not transfer.

Zeroing Y entirely. The fully-ablated condition ($\mathbf{Y}$=$\mathbf{0}$) *does* rescue in WaLa. As argued in §C.2, this rescue follows from architectural redundancy — the other 31 cross-attention writes plus the 32 AdaLN-modulation sites suffice to propagate the condition — rather than from any property of Meltdown. The direct empirical contrast with Make-a-Shape, where the same intervention is destructive (§C.3), makes this redundancy interpretation concrete.

C.5 WaLa: targeted drift surgery

To localize past the redundancy confound we target not $\mathbf{Y}$ itself but the *drift* induced by ρ. Let $\mathbf{Y}_c = \mathbf{Y}(\rho_{\text{melt}})$, $\mathbf{Y}_0 = \mathbf{Y}(0)$, and $d\mathbf{Y} = \mathbf{Y}_c - \mathbf{Y}_0$. Let $V_m^{d\mathbf{Y}}$ denote the top-m right-singular subspace of $d\mathbf{Y}$. The targeted surgery subtracts the portion of the drift that lives in its own top-m subspace:

$$\mathbf{Y}_{\text{post}} \;=\; \mathbf{Y}_c \;-\; \Pi_{V_m^{d\mathbf{Y}}}(\mathbf{Y}_c - \mathbf{Y}_0).$$

Rescue exhibits a sharp rank threshold:

m	5	10	20	50
rescue count	0/100	76/100	100/100	100/100
$\bar{C}$	143.6	24.0	1.00	1.00
$\overline{\text{chamfer}}$	0.1122	0.1008	0.0887	0.0887

At $m \geq 20$, the rescued chamfer matches the healthy reference (0.0887) to four decimals.

C.6 WaLa: magnitude-matched direction controls

The drift surgery at m=20 perturbs $\mathbf{Y}$ with Frobenius magnitude $\Delta_{20} \approx 1.385 \times 10^6$. A concern is that any perturbation of comparable magnitude might rescue by pushing the latent past an expendability threshold, independent of direction. To test this, we construct four controls that match or compare against this magnitude but differ in directional content.

Controls. Let $d\mathbf{Y} = U_d \Sigma_d V_d^\top$.

C1 (drift-orthogonal random subspace). $\mathbf{Y}_{\text{post}} = \mathbf{Y}_c - U_{d,1:m}\,\Sigma_{d,1:m}\,R^\top$, where $R \in \mathbb{R}^{d\times m}$ is an orthonormal basis of a randomly sampled m-dimensional subspace of the orthogonal complement of $V_{d,1:m}$, produced by QR on a Gaussian matrix projected off $V_{d,1:m}$. This preserves the left factors U_d, the singular values Σ_d, and Δ_m exactly; only the feature-space directions V are rotated. The orthogonality residual $\|V_{d,1:m}^\top R\|$ across all 1,200 runs has mean 5.6×10^{-7} and maximum 1.6×10^{-6}. Three independent R draws per (seed, m).

C2 (scalar attenuation). $\mathbf{Y}_{\text{post}} = \alpha \mathbf{Y}_c$ with $\alpha = \max\{0,\, 1 - \Delta_m/\|\mathbf{Y}_c\|_F\}$. Preserves the *shape* of $\mathbf{Y}_c$'s spectrum exactly (every singular value is scaled by α); represents the "shrink uniformly toward zero" direction. One run per (seed, m).

C3 (isotropic Gaussian noise). $\mathbf{Y}_{\text{post}} = \mathbf{Y}_c + \epsilon_m G$, $G \sim \mathcal{N}(0, I)$ i.i.d. entrywise, ϵ_m chosen so $\|\epsilon_m G\|_F = \Delta_m$. No directional structure. Three independent G draws per (seed, m).

C4 (keep-top-m on Y's own basis). The intervention of §C.4, repeated here for completeness. Its Frobenius perturbation $\|\mathbf{Y}_c - \mathbf{Y}_{(m)}^{\text{keep}}\|_F$ is determined by $\mathbf{Y}_c$'s own tail energy and is not explicitly matched to Δ_m; we include C4 because it is the architecture-aligned structural ablation, not because it is magnitude-matched.

For C1–C3, the Frobenius match is exact by construction up to the orthogonality residual of C1: targeted, C2, and C3 all share $\Delta_m = 1.38465 \times 10^6$ at m=20 to machine precision, while C1 deviates by <1 part in 10^5.

Table 5: WaLa rescue counts out of 100 seeds. Randomized controls (C1, C3) are aggregated over 3 replicates per (seed, m). All controls fail at every rank tested.

variant \ m	5	10	20	50
targeted drift surgery	0	76	**100**	**100**
C1: drift-orth. random subspace ($n = 300$/rank)	0	0	0	0
C2: scalar attenuation ($n = 100$/rank)	0	0	0	0
C3: Gaussian noise ($n = 300$/rank)	0	0	0	0
C4: keep-top-m on $\mathbf{Y}$ ($n = 100$/rank)	0	0	0	0

Results. Aggregated across the four ranks: C1 rescues 0/1200, C2 rescues 0/400, C3 rescues 0/1200, C4 rescues 0/400; across the entire control suite 0/3200 runs produce a sphere, while the targeted surgery reaches 100/100 at $m \geq 20$. Rescue at $\mathbf{Y}$ is therefore direction-specific, not magnitude-driven. C2 in particular rules out the "$\mathbf{Y}$=0 just shrinks the write" hypothesis: at matched magnitude, the uniform-shrink direction never rescues.

C.7 WaLa: H is a correlate, not a cause

Main-paper Section 3.3 identifies the spectral entropy $H(\mathbf{Y})$ as a scalar that rises smoothly with ρ alongside the discontinuous jump in C. The matched-magnitude controls allow us to test whether H is the causal variable. Post-intervention values of $H(\mathbf{Y}_{\text{post}})$ at m=20 (mean over 100 seeds; standard deviations all $< 5 \times 10^{-3}$):

variant	H at m=20	rescue
targeted drift surgery	1.613	100/100
scalar attenuation (C2)	1.639	0/100
random subspace (C1)	1.779	0/100
Gaussian noise (C3)	1.884	0/100
keep-top-20 on $\mathbf{Y}$ (C4)	1.582	0/100

Because C2 preserves the shape of the spectrum exactly, its H-value (1.639) is the corrupt-baseline entropy $H(\mathbf{Y}_c)$; this is stable to four decimals across m as expected. The targeted surgery reduces H by only 0.026 and rescues. C4 drives H *below* the rescuing value (1.582 < 1.613) yet fails in all 100 seeds. C1 and C3 push H well *above* the corrupt baseline (1.779, 1.884) and also fail. Matched-magnitude interventions therefore move H in both directions without rescuing, while the successful intervention leaves H essentially unchanged. H cannot be the causal variable: it tracks proximity to the transition, but the directional content of $\mathbf{Y}$ decides which basin the trajectory commits to.

C.8 Joint interpretation

Both models localize Meltdown to the cross-attention write at (block 4, step T), and in both, *spectral* structure of $\mathbf{Y}$ carries the causal signal. The protocols differ because the architectures differ.

In Make-a-Shape, $\mathbf{Y}$ is one of only 8 cross-attention writes and has no $\mathbf{C}$-dependent AdaLN fallback. Zeroing the write is actively destructive. Directly splitting the spectrum of $\mathbf{Y}$ with SVD reveals that its *tail* carries the Meltdown-inducing component while its *top* carries the indispensable conditioning signal; this is the MAS result.

In WaLa, $\mathbf{Y}$ is one of 32 cross-attention writes with a further 32 AdaLN-modulation pathways in parallel; the write is dispensable as a whole, so the direct analog of the MAS ablation does not localize anything. Targeting the *drift* $d\mathbf{Y} = \mathbf{Y}(\rho_{\text{melt}}) - \mathbf{Y}(0)$ instead of $\mathbf{Y}$ itself, and rigorously controlling for Frobenius magnitude with four complementary controls, shows that a specific low-rank ($m \geq 20$) directional correction to the write rescues all 100 seeds, while no matched-magnitude control rescues any seed in 3,200 runs. Spectral entropy H, which rises with ρ alongside the C discontinuity, cannot be the causal variable: the controls push H both above and below the rescuing value without rescuing, while the successful surgery leaves H unchanged.

Across both models, the overarching conclusion is the same — the cross-attention write at block 4 of the first denoising step contains a direction-specific Meltdown signature that can be surgically neutralized.

D Exhaustive Within-Block Causal Scan

This appendix gives the activation-patching procedure used throughout the paper in full detail, and reports an exhaustive within-block scan that tests whether the cross-attention restriction adopted in Section 3.2 is consistent with the network's causal map. The same algorithm underlies both analyses: in the main text it is instantiated for the cross-attention output projection $\mathbf{Y}$ alone; in this appendix it is run for every accessible intermediate activation of every DiT block at every denoising step.

TL;DR. The scan covers 32 blocks $\times$ 8 denoising steps $\times$ 19 intermediate activations per block, for 4,864 patches per diffusion seed. Coverage includes three diffusion seeds. Three findings:

- Of the $32 \cdot 8 = 256$ depth-time cells, 8 admit a within-block (non-residual-stream) rescue. Exactly one of these cells is the canonical (k=4, t=7) identified in the main text. At that cell, the cross-attention output projection ($\mathbf{Y}$, observed at the three equivalent sites `ca_preproj`, `Y_kt`, `ca_module_out`) is the unique within-block rescue: the other 14 sites at the cell – comprising the eight self-attention-branch sites, the AdaLN modulation, the four CA-pre-write sites, and the two MLP sites – all fail. The remaining 7 within-block rescues across the grid lie at (k=0, $t \in \{2, \ldots, 6\}$) and (k=31, $t \in \{1, 2\}$) and involve only the post-CA-MLP path, never cross-attention.
- Self-attention-branch patches and AdaLN-modulation patches produce 0 rescues at any cell of the grid. Cross-attention pre-write patches (`ca_module_in`, `ca_q_postnorm`, `ca_k_postnorm`) likewise produce 0 rescues at any cell. The cross-attention output projection is the only cross-attention-branch site that ever rescues, and only at (k=4, t=7).
- At t=7, the depth axis partitions into three contiguous zones: a pre-commit zone ($k \in \{0, 1, 2, 3\}$) where no patch of any kind rescues, including replacement of the residual stream entering block 3; a commit point ($k = 4$) where the cross-attention output projection rescues; and a propagation zone ($k \in \{5, \ldots, 31\}$) where only residual-stream patches rescue. The CA-write rescue at the commit point and the residual-stream rescues throughout the propagation zone produce reconstructions of equivalent sphere quality (chamfer 0.0886 vs 0.0896 ± 0.0014, both marginally below the 0.0916 clean baseline; radial standard deviation 0.006; sphericity proxy 1.000), with no depth-dependent drift across the 28 blocks of the propagation zone. The seven off-canonical within-block rescues exhibit degraded geometry (chamfer 0.097–0.098, sphericity proxy 1.02–1.05).

The findings identify (k=4, t=7) as the unique cell on the grid at which a single submodule – the cross-attention output projection $\mathbf{Y}$ – causally controls Meltdown with full sphere quality, and identify SA, AdaLN, MLP, and CA pre-write activations as either causally inert (SA, AdaLN, CA pre-write) or causally peripheral (MLP, with 7 off-canonical, geometrically inferior rescues). The cross-attention restriction in Section 3.2 targets the only cell on the grid that meets these conditions. The remainder of this appendix supplies the procedure, the site enumeration, and the per-cell tables that support these claims.

D.1 Protocol

The patching procedure is parameterized by a single *site* s, an intermediate activation of a DiT block. Algorithm 2 specifies the procedure for an arbitrary site s: it caches the activation at s from a forward pass on the healthy point cloud at every depth-time cell (k, t), runs an unhealthy forward pass for each cell with the cached value reinjected at that single cell, and records the resulting connected-component count C. In the main text (Section 3.2) the procedure is instantiated with $s = \mathbf{Y}$, the cross-attention output projection. In the present appendix the same algorithm is run for each of the 19 within-block activations listed in Table 6, producing the 4,864-patch rescue map per seed summarized above.

A patch is judged a rescue when $C = 1$ and the mesh has at least 200 faces. This is the same connectivity criterion as Section 3.2; geometric quality (chamfer, mean radius, radial standard deviation, sphericity proxy) is recorded as auxiliary information and used in Section D.5 to stratify rescues. Conditions: WaLa, $\rho_{\text{crit}} = 0.4$, $N_{\text{points}} = 400$, DDIM sampling with $T = 8$ denoising steps. The patch acts only on the conditional half of the classifier-free batch (Heimersheim & Nanda, 2024).

Algorithm 2 Localizing Meltdown via activation patching at site s.

Require: Encoder E; latent diffusion transformer B with K blocks; decoder D; healthy point-cloud $\mathcal{P}$; unhealthy point-cloud $\mathcal{Q}$; site s (any one of the within-block activations of Table 6)

$\mathbf{Z}_T^0 \sim \mathcal{N}(0, I)$ ▷ sample initial noise
Record healthy activations:
$\mathbf{C}_{\mathcal{P}} \leftarrow E(\mathcal{P})$
for $t = T : 1$ **do**
  for $k = 0 : K-1$ **do**
    $\mathbf{Z}_t^{k+1} \leftarrow B^k(\mathbf{Z}_t^k, \mathbf{C}_{\mathcal{P}})$ and record activation at site s as $X_{k,t,s}^{\text{healthy}}$
  end for
  $\mathbf{Z}_{t-1}^0 \leftarrow \text{DDIM}(\mathbf{Z}_t^{K-1})$ ▷ discrete denoising update
end for
Patch unhealthy activations:
$\mathbf{C}_{\mathcal{Q}} \leftarrow E(\mathcal{Q})$
for $t' = T : 1$ **do** ▷ denoising substitution loop
  for $k' = 0 : K-1$ **do** ▷ block substitution loop
    for $t = T : 1$ **do**
      for $k = 0 : K-1$ **do**
        $\mathbf{Z}_t^{k+1} \leftarrow B^k(\mathbf{Z}_t^k, \mathbf{C}_{\mathcal{Q}})$, overwriting site s with $X_{k,t,s}^{\text{healthy}}$ **if** $t = t'$ **and** $k = k'$
      end for
      $\mathbf{Z}_{t-1}^0 \leftarrow \text{DDIM}(\mathbf{Z}_t^{K-1})$
    end for
    $C_{k',t',s} \leftarrow C(D(\mathbf{Z}_0^{K-1}))$ ▷ decode and count components after patch
  end for
end for
return repair map $\{C_{k,t,s}\}_{k=0:K-1,\, t=1:T}$

D.2 Site enumeration

A DiT block in WaLa (Eq. 1a–1c) exposes the 19 intermediate activations listed in Table 6. Sites are grouped by site class and ordered along the dataflow direction. Three groups of nominally distinct sites correspond to single dataflow paths: the block boundary $\texttt{block_out}_k = \texttt{x_res}_{k+1}$; the cross-attention output projection ($\texttt{ca_preproj}, \texttt{Y_kt}, \texttt{ca_module_out}$), which is the same physical tensor (the projection's input, output, and the CA module's output coincide once the projection is the module's last operation, as verified by post-patch chamfer values agreeing to six decimal places); and the post-CA MLP path ($\texttt{norm2_out}, \texttt{mlp_in}, \texttt{mlp_out}$), three adjacent sites whose patches converge to numerically near-identical post-patch reconstructions where they rescue.

D.3 Spatial structure at t=7

Holding the denoising step at the canonical $t = 7$, the per-block rescue map partitions the depth axis into three contiguous zones. Table 7 lists, for every block $k \in \{0, \ldots, 31\}$, the subset of the 19 within-block sites whose patching rescues.

Pre-commit zone ($k \in \{0, 1, 2, 3\}$). Patching $\texttt{x_res}_k$, $\texttt{block_out}_k$, or any of the 17 within-block sites at any of these blocks fails to rescue, with the single exception of $\texttt{block_out}_3 = \texttt{x_res}_4$. Of $4 \cdot 19 = 76$ unique site-patches in this zone, 1 rescues, and that one is the residual stream entering block 4. The Meltdown signal is not localizable to any submodule of blocks 0–3 at $t = 7$ and is not yet committed to the residual stream upstream of block 4.

Commit point ($k = 4$). Five sites rescue at (k=4, t=7): the residual-stream sites $\texttt{x_res}_4$ and $\texttt{block_out}_4$, and the cross-attention output-projection triple ($\texttt{ca_preproj}, \texttt{Y_kt}, \texttt{ca_module_out}$). The remaining 14 within-block sites at this cell do not rescue: the seven SA-branch sites (`norm1_out`, `sa_module_in`,

Site	Class	Hook	Dataflow position
`x_res`	block input	pre	block input residual $\mathbf{Z}^k$
`ada`	AdaLN params	post	AdaLN modulation parameters
`norm1_out`	SA branch	post	first LayerNorm output (pre SA-AdaLN)
`sa_module_in`	SA branch	pre	input to SA module ($\hat{\mathbf{Z}}$)
`sa_q_postnorm`	SA branch	post	SA query post-norm
`sa_k_postnorm`	SA branch	post	SA key post-norm
`sa_preproj`	SA branch	pre	input to SA output projection
`sa_proj_out`	SA branch	post	output of SA output projection
`sa_module_out`	SA branch	post	output of SA module ($\hat{\mathbf{Y}}$)
`norm2_out`	CA pre-write	post	second LayerNorm output (pre CA-AdaLN)
`ca_module_in`	CA pre-write	pre	input to CA module ($\mathbf{Z}$)
`ca_q_postnorm`	CA pre-write	post	CA query post-norm
`ca_k_postnorm`	CA pre-write	post	CA key post-norm
`ca_preproj`	CA write	pre	input to CA output projection
`Y_kt`	CA write	post	output of CA output projection ($\mathbf{Y}$)
`ca_module_out`	CA write	post	output of CA module ($\mathbf{Y}$)
`mlp_in`	MLP branch	pre	input to MLP module
`mlp_out`	MLP branch	post	output of MLP module ($\bar{\mathbf{Y}}$)
`block_out`	block output	post	block output residual $\mathbf{Z}^{k+1}$

Table 6: The 19 within-block sites covered by the scan and admitted by Algorithm 2 as the parameter s. Hooks marked *pre* replace the input to the named module; hooks marked *post* replace its output. Dataflow position references the corresponding tensor in Eq. (1a)–(1c) where applicable. The main-text instantiation (Section 3.2) uses $s =$ `Y_kt`.

Blocks	Zone	Rescuing sites at $t = 7$
$k \in \{0, 1, 2\}$	pre-commit	none of the 19 sites
$k = 3$	pre-commit	`block_out` only
$k = 4$	commit	`x_res`, `ca_preproj`, `Y_kt`, `ca_module_out`, `block_out`
$k \in \{5, \ldots, 31\}$	propagation	`x_res`, `block_out`

Table 7: Sites whose patches rescue at t=7, by block. Identical across all available seeds. The asymmetry at $k = 3$ is mechanical: $\texttt{block_out}_3 = \texttt{x_res}_4$ as a single physical tensor, so the residual-stream cut between blocks 3 and 4 is realized by a successful patch from either side.

`sa_q_postnorm`, `sa_k_postnorm`, `sa_preproj`, `sa_proj_out`, `sa_module_out`); the AdaLN parameter site `ada`; the four CA-pre-write sites (`norm2_out`, `ca_module_in`, `ca_q_postnorm`, `ca_k_postnorm`); and the two MLP sites (`mlp_in`, `mlp_out`). The post-patch chamfer is identical to six decimal places across the three CA-write sites in each available seed (e.g. 0.088541 in seed 0; 0.088708 in seed 1), confirming that they observe a single physical tensor.

Propagation zone ($k \in \{5, \ldots, 31\}$). At every block in this range, only the residual-stream sites $\texttt{x_res}_k$ and $\texttt{block_out}_k$ rescue. Of the $27 \cdot 17 = 459$ within-block (non-residual) site-patches in this zone, 0 rescue. In particular, $\texttt{Y_kt}_{k,7}$ for $k \in \{5, \ldots, 31\}$ never rescues. The Meltdown signal, once committed at block 4, is carried by the residual stream rather than by any internal computation of the downstream blocks; cleaning a single submodule's output downstream of the commit point does not undo it.

D.4 Temporal structure at $k = 4$

Holding the block at the canonical $k = 4$, the rescue pattern across denoising steps is given in Table 8.

Step	Regime	Rescuing sites at $k = 4$
$t \in \{0, 1\}$	closed	none of the 19 sites
$t \in \{2, \ldots, 6\}$	open	`x_res`, `block_out`
$t = 7$	commit	`x_res`, `ca_preproj`, `Y_kt`, `ca_module_out`, `block_out`

Table 8: Sites whose patches rescue at $k = 4$, by denoising step. $\texttt{Y_kt}_{4,t}$ rescues only at $t = 7$. The combination of the spatial localization to $k = 4$ at $t = 7$ (Section D.3) and the temporal localization to $t = 7$ at $k = 4$ identifies (k=4, t=7) as the only cell on the 32×8 grid at which a cross-attention write rescues.

The temporal pattern at $k = 4$ generalises: across the entire grid, no within-block site rescues at $t = 0$, and only the off-canonical (k=31, t=1) MLP-path pocket rescues at $t = 1$ (Section D.6). The trajectory thus appears closed at the last two denoising steps and progressively more localized in preceding steps.

D.5 Geometric quality of rescues

The connectivity criterion ($C = 1$, ≥ 200 faces) is a topological test. A rescued mesh can have $C = 1$ while differing from the clean output in size, shape, or surface uniformity. Table 9 reports the four geometric statistics across the rescue categories identified in Sections D.3–D.4.

Category	Description	n	chamfer	mean radius	radial std
A	CA-write trio at (k=4, t=7)	6	0.0886 ± 0.0001	0.9135 ± 0.0001	0.0058
B	residual stream at (k=4, t=7)	4	0.0887 ± 0.0001	0.9133 ± 0.0000	0.0058
C	residual stream at (k=3, t=7)	2	0.0887 ± 0.0000	0.9134 ± 0.0000	0.0059
D	residual stream, propagation zone ($k \geq 5$, t=7)	72	0.0896 ± 0.0014	0.9124 ± 0.0015	0.0062
E	residual stream, open window ($t \in \{2, \ldots, 6\}$)	469	0.0977 ± 0.0030	0.9037 ± 0.0035	0.0119
F	MLP-path pocket at (k=0, $t \in \{2, \ldots, 6\}$)	30	0.0981 ± 0.0034	0.9031 ± 0.0041	0.0126
G	MLP-path pocket at (k=31, $t \in \{1, 2\}$)	6	0.0967 ± 0.0101	0.9037 ± 0.0116	0.0157
	clean baseline ($\rho = 0$)	–	0.0916	0.9104	0.0058
	corrupt baseline ($\rho = 0.4$)	–	0.1118	–	–

Table 9: Geometric statistics of rescued meshes by category. Reported as mean ± standard deviation, aggregated over all rescue rows in each category and over the available seeds. Sphericity proxy (surface area divided by $4\pi r^2$) is 1.000 ± 0.000 for categories A–D and 1.015–1.05 for categories E–G; omitted from the table for compactness. Categories A–D have chamfer slightly below the clean baseline; this reflects the chamfer being measured against an ideal 10,000-point unit sphere, with respect to which the patched runs at t=7 produce reconstructions marginally closer to ideal than the unperturbed clean run.

The categories partition into two regimes. Categories A–D, which share the canonical denoising step t=7, have chamfer 0.089 ± 0.002, mean radius 0.913 ± 0.002, radial standard deviation 0.006 ± 0.001, and sphericity proxy 1.000 ± 0.000 – matching or marginally exceeding the clean baseline on every statistic. Categories E–G, which share the property of patching at non-canonical denoising steps, have chamfer 0.097 ± 0.003, mean radius 0.903 ± 0.004, radial standard deviation 0.013 ± 0.005, and sphericity proxy 1.02 ± 0.02 – recovering connectivity but producing a slightly smaller, less uniform, less spherical reconstruction than the clean baseline.

Two consequences for the cross-attention restriction. First, within the canonical step, replacing only the cross-attention output projection at the commit point (category A, n=6, mean chamfer 0.0886) is geometrically equivalent to replacing the entire block input or output residual at the same cell (category B, n=4, mean

chamfer 0.0887): the chamfer means differ by 10^{-4}. The cross-attention write therefore carries the full structural content of the residual stream at the commit point; the SA, AdaLN, and MLP contributions to block 4's residual update at t=7 are not geometrically informative once $\mathbf{Y}_{4,7}$ is correct. Second, the propagation-zone rescues (category D) reproduce the canonical rescue's chamfer to within 10^{-3} across all 28 blocks, with no detectable depth-dependent drift; the residual stream downstream of the commit point transports the committed signal without modification. Both observations sharpen the interpretation of the cross-attention output projection at (k=4, t=7) as the unique commit lever, with downstream blocks acting as transport rather than as additional sources of structural information.

D.6 Off-canonical MLP-path pockets

Two pockets of within-block rescue lie outside the canonical Meltdown circuit. Both involve the post-CA MLP path (`norm2_out`, `mlp_in`, `mlp_out`), with all three sites rescuing together:

- (k=0, $t \in \{2, 3, 4, 5, 6\}$): 5 cells at the first DiT block, across the open temporal window.
- (k=31, $t \in \{1, 2\}$): 2 cells at the last DiT block, at the boundary of the closed and open windows.

The pockets are spatially disjoint from the canonical cell ($k \in \{0, 31\}$ versus $k = 4$), temporally disjoint ($t \in \{1, \ldots, 6\}$ versus $t = 7$), modulewise disjoint (MLP path versus cross-attention output projection), and geometrically inferior (categories F and G in Table 9). They are not part of the Meltdown circuit localized in Section 3.2.

D.7 Converse direction: noising scan

Sections D.3–D.6 report the *rescue* scan: at every cell on the depth-time grid, replace the unhealthy activation with the cached healthy one, and ask whether connectivity is restored. We now report the converse: at every cell, replace the healthy activation with the cached unhealthy one, and ask whether the run fragments. The protocol is otherwise identical to Algorithm 2, with the roles of $\mathcal{P}$ and $\mathcal{Q}$ swapped. The verdict has two tiers, mirroring the rescue verdict: *lax noise* requires the output to be fragmented ($C > 1$, ≥ 200 faces). Coverage: the full $32 \cdot 8 \cdot 19 = 4{,}864$ patches at three diffusion seeds, $\rho_{\text{crit}} = 0.4$, $N_{\text{points}} = 400$, DDIM with $T = 8$.

TL;DR. Three findings, all of which mirror or sharpen claims made in the rescue direction.

- *The commit lever is asymmetric.* At the canonical cell (k=4, t=7), the cross-attention output projection (`ca_preproj`, `Y_kt`, `ca_module_out`) does not noise: replacing the healthy $\mathbf{Y}_{4,7}$ with the cached unhealthy value produces a clean sphere ($C = 1$, chamfer 0.0893, sphericity 1.000), indistinguishable from the unintervened clean run. The same holds across the entire grid: 0/256 lax-noise events for $\mathbf{Y}_{k,t}$ at any (k, t), and 0/768 across all three CA-write sites. Y is a sufficient lever in the rescue direction (Section D.3) but not in the noise direction.
- *SA, AdaLN, and CA pre-write are inert in both directions.* Self-attention-branch patches yield 0/1,792 noise events; AdaLN-modulation patches yield 0/256; CA pre-write patches yield 0/1,024 at the canonical cell and 8/1,024 overall, all confined to (k=31) on `norm2_out` (see fourth bullet). These site classes produce no rescues anywhere on the grid (Section D.3) and produce no noise events at any cell that lies on the canonical Meltdown circuit.
- *Residual-stream noise propagation mirrors the rescue propagation zone.* At t=7, $\texttt{x_res}_k$ and $\texttt{block_out}_k$ chamfer-noise for $k \geq 4$ on `block_out` and $k \geq 5$ on `x_res` (the offset is mechanical: $\texttt{block_out}_k = \texttt{x_res}_{k+1}$), and produce a clean sphere for $k \leq 3$. The rescue scan's propagation zone ($k \geq 5$, residual stream rescues; $k = 4$, $\texttt{x_res}_4$ and $\texttt{block_out}_4$ rescue) and the rescue scan's pre-commit zone ($k \leq 3$, no within-block patch rescues except $\texttt{block_out}_3 = \texttt{x_res}_4$) appear in the noising scan as the noise-propagation zone and the noise-inert zone respectively. Both directions therefore agree that block 4 at t=7 is where the corrupt signal first appears in the residual stream.
- *Off-canonical MLP-path pockets mirror the rescue pockets.* Non-residual chamfer-noise events occur only on the post-CA MLP path (`norm2_out`, `mlp_in`, `mlp_out`) and only at the boundary blocks

$k \in \{0, 31\}$: at (k=0, $t \in \{2, 3, 4, 6\}$) and at (k=31, $t \in \{0, 1, 7\}$). The rescue scan's MLP-path pockets at (k=0, $t \in \{2, \ldots, 6\}$) and (k=31, $t \in \{1, 2\}$) cover almost the same set of cells. The boundary-block MLP path is therefore bidirectionally pluripotent — it can both rescue and noise — but, like its rescue counterpart, it is spatially, temporally, and modulewise disjoint from the canonical commit cell.

The asymmetry at the commit cell. At (k=4, t=7), the rescue and noising verdicts disagree on which within-block sites are causally active. Table 10 lists every site at the cell alongside its rescue-direction verdict from Section D.3. The two directions agree on three sites: the SA branch, AdaLN, and CA pre-write are inert in both. They disagree on two: the CA-write triple is rescue-active but noise-inert, and $\texttt{block_out}_4$ is the only site that both rescues and noises at this cell. (The block-input residual $\texttt{x_res}_4$ rescues but does not noise: it carries the clean signal forward into block 4 in the rescue direction, but injecting the corrupt signal at the entry to block 4 is too early — the subsequent steps re-anchor the trajectory toward sphere.)

Site	**Class**	**Rescue**	**Noise (chamfer)**
`x_res`	block input	✓	—
`block_out`	block output	✓	✓
`ada`	AdaLN params	—	—
SA-branch sites (×7)	SA branch	—	—
CA pre-write sites (×4)	CA pre-write	—	—
`ca_preproj`	CA write	✓	—
`Y_kt`	CA write	✓	—
`ca_module_out`	CA write	✓	—
MLP-branch sites (×2)	MLP branch	—	—

Table 10: Every within-block site at the canonical commit cell (k=4, t=7), with its rescue-direction verdict (Section D.3) and its noise-direction verdict (this section). The CA-write triple rescues but does not noise; only the block-output residual stream is causally active in both directions. The block-output residual at k=4 coincides with the block-input residual at k=5, so this entry is also the start of the noise-propagation zone described in the third TL;DR bullet.

Interpretation. A symmetric one-cell-isolates-the-commit picture would predict that patching $\mathbf{Y}_{4,7}$ in either direction transports the run between attractors. The rescue scan confirms the forward half of this prediction; the noising scan refutes the backward half. The asymmetry is consistent with the diffusion-dynamics view of §5: the trajectory must be in the basin of the speckle attractor to commit to fragmentation, and the basin boundary is crossed by a low-rank *drift* in $\mathbf{Y}$ (the direction-specific surgery of §3.3.1) rather than by any single value of $\mathbf{Y}$ in isolation. Replacing $\mathbf{Y}_{4,7}$ with the cached unhealthy value transplants the endpoint of this drift but not the cumulative state of the rest of the denoising trajectory; with the remaining seven steps conditioned on the healthy $\mathbf{C}$ and starting from a healthy latent, the trajectory re-anchors to the sphere basin. Conversely, the cumulative state *is* carried by the residual stream, which is why $\texttt{block_out}_4$ is the one within-block site at the commit cell that is causally active in both directions and why all $k \geq 5$ residual-stream cells at t=7 noise (and rescue) the run. $\mathbf{Y}_{4,7}$ is thus the *site* of the commit but not, in isolation, a *sufficient cause* of it; the surgery in §3.3.1 succeeds because it modifies the trajectory's drift, not because it sets a single activation to a single rescuing value.

E Statistical Properties of the Input Cloud

Cloud ensemble. We construct an ensemble of perturbed sphere clouds by displacing the Fibonacci reference $\mathcal{P}_0$ along random tangent fields built from real spherical harmonics. For each wavenumber $\ell \in \{2, 3, 4, 6, 8, 12, 16, 24, 32\}$ we draw five independent Gaussian coefficient vectors over the $2\ell+1$ basis functions Y_ℓ^m, take the surface gradient of the resulting eigenfunction, and apply the geodesic exponential map at amplitude ε from a 10-point grid in $[0.001, 0.5]$ radians, normalized so that ε is the maximum geodesic displacement of any point. Together with the slerp path of Section 2 (14 values of ρ, two random target

clouds), and across three sample sizes $N \in \{400, 500, 600\}$, this yields 1,437 unique input clouds; each is decoded under five independent diffusion seeds, for 7,185 forward passes total.

Riesz s=2 energy. For a finite point set $\mathcal{P} = \{p_i\}_{i=1}^N \subset \mathbb{S}^2$, the Riesz s=2 energy is

$$E_2(\mathcal{P}) \;=\; \sum_{i \neq j} \|p_i - p_j\|^{-2}, \tag{11}$$

with $\|\cdot\|$ the Euclidean distance in $\mathbb{R}^3$. The Riesz s-energy is a classical sphere-uniformity functional whose minimizers approach the uniform measure on $\mathbb{S}^2$ as $N \to \infty$ (Hardin & Saff, 2004; Brauchart & Grabner, 2015); the Fibonacci sample $\mathcal{P}_0$ is a quasi-optimal minimizer at finite N. We summarize departures from this reference by the dimensionless *Riesz excess*

$$\Delta E(\mathcal{P}) \;=\; E_2(\mathcal{P})/E_2(\mathcal{P}_0) - 1, \tag{12}$$

which is approximately zero on $\mathcal{P}_0$ and grows as $\mathcal{P}$ becomes more crowded than uniform.

Result. For each N we bin the 2,395 trials into 14 quantile bins on $\log_{10} \Delta E$ and report the binomial probability of Meltdown with 95% Jeffreys credible intervals over the diffusion-seed dimension. Figure 20 shows that across all three sample sizes, ΔE separates a Meltdown-free regime from a saturation regime through a single-decade transition. For N=400, no cloud melts in the 675 trials with $\Delta E < 10^{-2}$, 99.6% (762/765) of the trials with $\Delta E > 0.20$ melt, and the 50% probability crossing sits at $\Delta E^\star \approx 0.07$. The slerp path of Section 2 crosses $\Delta E^\star$ in the same ρ-interval over which $C(\rho)$ jumps: at ρ=0 the slerp cloud coincides with $\mathcal{P}_0$ ($\Delta E = 0$, C=1), and at the smallest non-trivial grid sample ρ=0.05 we already have $\Delta E \approx 0.18 \gg \Delta E^\star$ and $C \approx 73$.

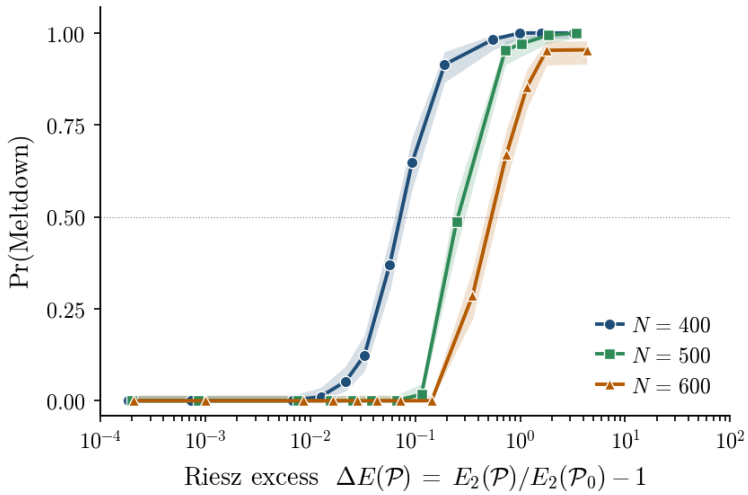


Figure 20: Probability of Meltdown as a function of the Riesz excess ΔE across the 7,185-trial ensemble, broken down by sample size $N \in \{400, 500, 600\}$. Markers are per-bin binomial means; shaded bands are 95% Jeffreys intervals over five diffusion seeds per cloud. The dotted line marks the 50% rate.

F Encoder Propagation of Input Non-uniformity

Appendix E established that Meltdown is governed by a sharp threshold on the Riesz s=2 energy of the input cloud. We now ask how this scalar reaches the diffusion backbone: which encoder stage carries it, and whether the encoder adds, removes, or distorts the melt-relevant signal en route to the conditioning $\mathbf{C}$.

Protocol. We reuse the cloud ensemble of Appendix E, restricted to a single diffusion seed per cloud so that melt outcome is a per-cloud binary. This yields 1,434 perturbed clouds across the three sample sizes (479 at each N), of which 484 melt and 950 do not. For each cloud we run the WaLa PointNet encoder once and capture post-LayerNorm activations at the seven sequential stages (`ln1`–`ln4`, the MAB pool, `ln_f1`, `ln_f2`) plus the encoder output $\mathbf{C}$. Letting X^L denote stage L and X_0^L the same stage on the Fibonacci reference $\mathcal{P}_0$, we report the relative Frobenius drift $\delta^L = \|X^L - X_0^L\|_F / \|X_0^L\|_F$, averaged separately over melted and healthy clouds at each N.

Layer-wise drift profile. Table 11 reports δ^L for melted and healthy clouds and their ratio. Three observations hold uniformly across N. First, the four per-point linear stages amplify the melted-vs-healthy drift ratio by 5.6–6.8×. Second, the MAB pool compresses this to 2.1–2.4×, a factor that is then preserved through `ln_f1`, `ln_f2`, and into $\mathbf{C}$. Third, while the absolute relative drifts shift with N (lower N has smaller healthy drift, so ratios are larger), the qualitative profile — per-point amplification, MAB compression, downstream preservation — is identical. The MAB is therefore the encoder stage at which per-point displacement is converted into a token-level conditioning signal, and the gap between $\mathbf{Y}_{\text{melt}}$ and $\mathbf{Y}_{\text{heal}}$ that the backbone subsequently reads is already established at the encoder's output.

	N=400			N=500			N=600		
stage	heal	melt	ratio	heal	melt	ratio	heal	melt	ratio
`ln1`	0.020	0.135	6.8×	0.027	0.163	6.2×	0.030	0.178	5.9×
`ln2`	0.024	0.162	6.7×	0.032	0.195	6.1×	0.037	0.213	5.8×
`ln3`	0.027	0.182	6.6×	0.037	0.219	6.0×	0.042	0.238	5.7×
`ln4`	0.026	0.171	6.6×	0.035	0.206	5.9×	0.040	0.223	5.6×
MAB pool	0.091	0.219	2.4×	0.096	0.208	2.2×	0.091	0.193	2.1×
`ln_f1`	0.065	0.156	2.4×	0.068	0.148	2.2×	0.065	0.138	2.1×
`ln_f2`	0.013	0.030	2.4×	0.013	0.029	2.2×	0.013	0.027	2.1×
$\mathbf{C}$	0.131	0.310	2.4×	0.138	0.294	2.1×	0.131	0.274	2.1×

Table 11: Relative Frobenius drift δ^L at each encoder stage, averaged over melted and healthy clouds at each sample size. The per-point stages (`ln1`–`ln4`) carry a ∼6× melted-vs-healthy gap; the MAB pool compresses this to ∼2× which is then preserved to the encoder output $\mathbf{C}$.

Faithful transduction. A simple consistency check is whether melt-prediction accuracy is lost or gained as the signal traverses the encoder. We score melt outcome with each scalar feature individually, computing the area under the ROC curve at each N. The Riesz excess ΔE achieves AUC $\in [0.981, 0.995]$ across the three N; the relative drift of $\mathbf{C}$ achieves AUC $\in [0.974, 0.982]$; no intermediate stage is more or less melt-predictive than its neighbours by more than 0.01. The encoder neither manufactures nor discards melt-relevant information: it transduces the input statistic from a property of the cloud into a property of the conditioning $\mathbf{C}$ that is read by the diffusion backbone.

Structural, not scalar. The encoder is melt-faithful but not a linear ΔE transducer. Reducing $\mathbf{C}$ to its top-64 PCA components (explaining $\geq 99\%$ of the variance across the ensemble) and fitting a 5-fold cross-validated ridge regression PCA($\mathbf{C}$) $\rightarrow \Delta E$ yields $R^2 \in [-0.18, 0.09]$ across the three N: the input scalar is not recoverable as a linear direction in $\mathbf{C}$. Yet a logistic probe on the same features predicts melt at AUC $\in [0.974, 0.981]$, and after residualizing each PCA dimension against ΔE before the probe the melt AUC remains $\in [0.864, 0.937]$. Melt-relevant content is therefore written into $\mathbf{C}$ structurally rather than along any single scalar axis. This is consistent with the activation-patching result of §3.2: the cross-attention write at (k=4, t=7) commits the trajectory by reading directional structure in the encoded condition.

G PowerRemap Site Sweep

The patching scan of Appendix D localizes the Meltdown commit to the cross-attention output projection $\mathbf{Y}_{4,7}$. `PowerRemap` (§4) is derived from this localization, but its operation differs from patching: it does not import a healthy activation, but reshapes the singular spectrum of whatever activation is present at the targeted site. We therefore verify directly that `PowerRemap` inherits the patching localization, by sweeping the intervention across every cross-attention and MLP cell of WaLa's U-ViT and asking which sites, if any, rescue Meltdown. Self-attention is excluded: a prior sweep over all 256 self-attention cells produced 0 rescues, and the patching scan likewise finds 0 self-attention rescues across the grid (Appendix D).

Protocol. For each (component, k, t) cell with component $\in \{$`cross_attn`, `mlp`$\}$, $k \in \{0, \ldots, 31\}$, and $t \in \{7, \ldots, 0\}$, we register a forward hook on the corresponding sub-module that applies `PowerRemap` ($\gamma = 100$) to the sub-module's output at step t and lets the rest of the reverse process run unmodified. This gives $2 \cdot 32 \cdot 8 = 512$ tested sites (768 counting the excluded self-attention grid). Conditions match the patching scan: $\rho_{\text{melt}} = 0.4$, $N_{\text{points}} = 400$, DDIM with $T = 8$, baseline $C = 140$. A site is recorded as a *rescue* when the output mesh has a single connected component ($C = 1$). Because $C = 1$ alone is necessary but not sufficient, we additionally validate each rescue by sphere-fit residual ($\sigma_{\|\cdot\|}/\bar{R}$, where R is the mean vertex-to-centroid distance) and the isoperimetric sphericity proxy $\pi^{1/3}(6V)^{2/3}/A$. A rescue is declared *valid* when sphere-fit residual < 0.25 and sphericity > 0.4 — thresholds chosen generously; clean baseline outputs have residual ≈ 0.001 and sphericity ≈ 1.000.

Results. Of the 512 sites tested, four produce a valid rescue. Table 12 lists them. No other site, in either component, at any block, at any step, recovers connectivity: the remaining 508 sites leave C statistically indistinguishable from the baseline $C = 140$. Counting against the full 768-cell grid including the excluded self-attention sweep, valid-rescue specificity is $4/768 = 0.5\%$.

	site	block	step	sphericity	sphere-fit residual	role
A	`cross_attn`	4	7	0.9998	0.0014	canonical commit ($\mathbf{Y}_{4,7}$)
B	`cross_attn`	3	7	0.9998	0.0014	upstream feeder at t=7
C	`mlp`	0	7	0.9998	0.0014	upstream feeder at t=7
D	`mlp`	0	0	0.9296	0.0880	off-canonical MLP-path pocket

Table 12: The four sites at which `PowerRemap` ($\gamma = 100$) rescues Meltdown out of 512 swept (or 0.5% of the full 768-cell grid including excluded self-attention). Sites A–C are at the canonical denoising step t=7, lie on the residual-stream pathway upstream of $\mathbf{Y}_{4,7}$, and produce sphere reconstructions indistinguishable in quality from the canonical rescue (sphericity agreement to four decimals). Site D is at the boundary block k=0 at t=0 and produces a geometrically inferior reconstruction (sphericity 0.93, residual $\sim 60\times$ larger), matching the off-canonical MLP-path pocket pattern documented in Appendix D.

Interpretation. The three t=7 rescues (A–C) form a contiguous residual-stream pathway: $\text{MLP}_{0,7}$ writes into the residual stream at block 0, that signal flows forward through blocks 1–3 where the cross-attention output projection $\mathbf{Y}_{3,7}$ contributes additional structure, and the canonical $\mathbf{Y}_{4,7}$ is the final commit. Compressing the spectrum at any of these three points heads off the corruption before it is written into the residual stream that feeds block 4's commit. Patches *at* a site downstream of $\mathbf{Y}_{4,7}$ at t=7 do not rescue, consistent with the patching scan: once the residual stream carries the committed signal, single-submodule interventions at later blocks cannot undo it.

The off-canonical site D ($\text{MLP}_{0,0}$) reproduces a pattern already isolated in the patching scan: spurious connectivity rescues on the post-CA MLP path at the boundary blocks $k \in \{0, 31\}$ at non-canonical denoising steps (Appendix D, §D.6). These pockets are spatially, temporally, and modulewise disjoint from the canonical Meltdown circuit, and produce geometrically inferior reconstructions: site D's sphericity (0.93) and sphere-fit residual (0.088) are inferior to the canonical rescue's by orders of magnitude on the residual axis, and match the quality of the off-canonical patching rescues reported in Table 9 (categories F and G).

The two scans, run on disjoint experimental designs, recover the same circuit and the same boundary-block side-channel.

Across both methods of intervention — single-cell patching with a healthy activation, and single-cell spectral compression of the existing activation — the locus of causal control over Meltdown is the same: the cross-attention pathway feeding $\mathbf{Y}_{4,7}$ at the first denoising step. `PowerRemap` inherits this localization.

H Simpler Baselines

This appendix reports the protocol and full results behind the "simpler interventions" paragraph in §4.0.1. We test three deployment-fair alternatives to `PowerRemap`, one per stage of the causal chain we identified in §3: input-cloud uniformization, diffusion-trajectory noise injection, and initial-noise resampling. Each baseline is the canonical SOTA representative of its category and operates without surface knowledge or quality oracles, matching the deployment regime considered in §1. None rescues at meaningful rates.

TL;DR. On the canonical sphere setup (§B.2.2) at ρ_{crit}=0.4 with 6 diffusion seeds:

- **Stage 1: input uniformization (WLOP).** 0/48 sphere rescues across iteration counts $\{1, 2, 5, 15\}$. At small counts (it. 1, 2) the operator partially uniformizes the cloud (Voronoi-area COV: 0.517 → 0.422, an 18% reduction) without crossing the basin separatrix; at larger counts the cloud destabilizes (collapse at it. 15, minimum pairwise arc → 10^{-4}).
- **Stage 1 oracle ablation (WLOP with $\mathbb{S}^2$ projection).** 0/18 sphere rescues across iteration counts $\{2, 5, 15\}$. Surface knowledge does not change the verdict for local-repulsion uniformizers: the on-manifold variant exhibits the same partial-uniformization plateau and the same long-iteration destabilization.
- **Stage 2: noise injection at every denoising step.** 0/96 sphere rescues across 8 steps × 2 strengths ($s \in \{0.3, 0.6\}$) × 6 seeds. Aggressive injection at the earliest steps meaningfully disturbs the trajectory ($\bar{C}$: 145.3 → 71.2 at t=T, s=0.6) but never produces a sphere; injection at the bifurcation step $\tau^\star \approx 5$ identified in §5 leaves the trajectory essentially unmoved.
- **Stage 3: best-of-K initial noise.** 0/24 sphere rescues across K=1, . . . ,4. Minimum C over 24 independent $\mathbf{x}_T$ draws is 134.
- `PowerRemap` (γ=100, anchor on the same seeds): 6/6 sphere rescues, mean chamfer 0.0887, mean sphericity proxy 1.0001.

H.1 Deployment regime and fair-baseline criteria

The paper considers reconstruction from sparse point clouds of geometry that the surface-recovery model is itself responsible for inferring (§1, §2). A baseline is *deployment-fair* if it can be run by a practitioner who has only the input cloud $\mathcal{P}$ and the diffusion model G=$D \circ B \circ E$ at hand. Concretely, a fair baseline must satisfy three conditions: (i) it uses only $\mathcal{P}$ and G — no ground-truth surface, no clean reference cloud, no surface-quality metric; (ii) it preserves the input size N, since the user has the points the sensor produced; (iii) selection criteria, where applicable, depend only on output-side observables computable from the mesh (e.g. topological connectivity), not on properties of an underlying surface. These conditions match the criterion that `PowerRemap`'s γ-grid for Make-a-Shape satisfies in §4.0.1: connectivity C=1 as the selection target requires no ground-truth surface and is therefore deployable at test time. They also exclude several classical alternatives. Tangent-plane-based redistribution and Voronoi-cell methods on a parametric surface presuppose the surface; learned point-cloud denoisers (e.g. Rakotosaona et al. (2019)) introduce a separately trained model, with its own training distribution, on top of the diffusion pipeline.

H.2 Choice of representative baselines

We choose one representative per stage of the causal chain identified in §3, taking the canonical SOTA reference at each.

Stage 1 — input cloud. The corruption $\mathcal{P}_\rho$ in §2 is on-surface non-uniformity rather than off-surface noise: each $p_i(\rho)$ lies exactly on $\mathbb{S}^2$ by construction of the per-point SLERP. The appropriate point-cloud operator is therefore a *redistribution* operator at fixed N, not a denoiser. Bilateral filters and their variants are designed for off-surface noise and project points along estimated normals toward local tangent planes; they would be approximately a no-op on our perturbation, since every input is already on its local tangent plane to numerical precision.

We therefore use Locally Optimal Projection (LOP) (Lipman et al., 2007), a parameterization-free, surface-agnostic redistribution operator that iteratively moves each particle toward an L1-median data target while a repulsion term spreads particles apart. Specifically we run its density-weighted variant WLOP (Huang et al., 2009), which is the canonical extension to non-uniform inputs — our setting. WLOP is parameter-light (repulsion weight μ and support radius h) and requires no normal estimation, making it the strictest deployment-fair representative of the surface-agnostic uniformization literature.

Stage 2 — diffusion sampling. Stage 2 baselines must operate on the reverse-time trajectory using only the trained denoiser. The canonical SOTA representative of this category is variance-preserving noise injection at intermediate denoising steps, formalized as "Langevin churn" in Karras et al. (2022a). This operator is also the natural test of the bifurcation hypothesis in §5: if the reverse trajectory is poised on a basin separatrix at $\tau^\star$, noise added near $\tau^\star$ should sometimes flip basin allocation. We sweep injection at every denoising step at two strengths to make the test exhaustive.

Stage 3 — alternative outside the activation pathway. Best-of-K initial-noise selection (Ma et al., 2025) is the simplest non-activation-based intervention: vary the seed $\mathbf{x}_T$ and let the model reconcile. The connectivity criterion C=1 used for selection is computable from the output mesh alone without surface knowledge, so this baseline is deployment-fair. It is the SOTA representative of inference-time scaling that does not modify the model.

H.3 Shared protocol

All experiments use the WaLa sphere setup of Appendix B.2.2: N=400 Fibonacci points, target cloud generated by Gaussian jitter (σ=0.1) and re-projection to $\mathbb{S}^2$, per-point SLERP with control parameter ρ, and ρ_{crit}=0.4 throughout. We use 6 diffusion seeds and report rescue under three connectivity-based verdicts of increasing stringency: *lax* (C=1 and $\geq$ 200 faces), *chamfer-rescue* (lax plus chamfer within 0.020 of the clean baseline), and *strict* (chamfer-rescue plus mean radius, radial std, and sphericity proxy within tolerances of clean). All headline rates below report chamfer-rescue. The clean baseline (6 seeds) has C=1, mean chamfer 0.0917, mean sphericity proxy 1.0006, and $\sim$ 513,000 faces; the corrupt baseline (6 seeds) has mean C=145.3, mean chamfer 0.1122, mean sphericity proxy 0.2575, and $\sim$ 129,500 faces. `PowerRemap` (γ=100) on the same seeds rescues 6/6 with mean chamfer 0.0887 and mean sphericity proxy 1.0001. We use these reference values throughout.

We additionally report two input-cloud uniformity statistics for Stage 1: the Riesz s=2 energy $E_2(\mathcal{P}) = \sum_{i\neq j} \|p_i - p_j\|^{-2}$ (Appendix E), with reference values E_2=113,461 on clean and E_2=237,309 on corrupt; and the spherical-Voronoi area coefficient of variation COV_V, with reference values COV_V=0.010 on clean and COV_V=0.517 on corrupt. These statistics are linked to Meltdown by the input-cloud analysis of Appendix E: E_2 is the scalar that determines whether $\mathcal{P}$ falls into the Meltdown regime.

H.4 Stage 1: input-cloud uniformization (LOP / WLOP)

The WLOP iteration moves each particle q_i to $q_i^{\text{new}} = d_i + \mu\, r_i$, where d_i is a density-weighted L1-median data target computed from $\mathcal{P}$ and r_i is a repulsion gradient with respect to all other particles $\{q_j\}_{j\neq i}$. We use the parameter values recommended in Huang et al. (2009): repulsion weight μ=0.45, support radius h set automatically to four times the median nearest-neighbor distance, and a step-size cap of $0.3h$ for numerical stability. Particles are initialized to the corrupt cloud $\mathcal{P}_\rho$ for in-place consolidation at fixed N=400. We run iteration counts $\{1, 2, 5, 15\}$ and apply the resulting consolidated cloud as the diffusion input.

H.4.1 LOP without surface knowledge (deployment-fair)

Table 13 reports rescue rates and input-statistics changes after WLOP. At small iteration counts (it. 1, 2), the operator measurably uniformizes the cloud: COV_V drops from 0.517 on the corrupt input to 0.436 (16% reduction) at it. 1 and to 0.422 (18% reduction) at it. 2, and the Riesz energy drops from 237,309 to $\sim$ 158,000. Neither produces a sphere rescue. At it. 5, partial uniformization continues (COV_V=0.406) but the cloud begins to drift off $\mathbb{S}^2$ (Riesz energy increases to 179,673 and the minimum pairwise arc proxy collapses to 0.008). At it. 15, the operator destabilizes catastrophically: the minimum pairwise arc proxy collapses to 10^{-4}, indicating particle coincidence, and the Riesz energy rises by three orders of magnitude. This is the documented off-manifold drift pathology of fixed-N WLOP applied without explicit surface projection (Han et al., 2021; Stotko et al., 2024): the L1-median data target of an on-manifold cloud lies inside the manifold, so particles drift inward iteration by iteration, and the inverse-square repulsion gradient diverges as inter-particle distances shrink.

Variant	**Iters**	**rescue**	$\bar{C}$	$\overline{\mathrm{COV}_V}$	$\overline{E_2}$	$\overline{\mathrm{arc}_{\min}}$
WLOP	1	0/6	163.0	0.436	1.58×10^5	0.017
WLOP	2	0/6	147.8	0.422	1.58×10^5	0.020
WLOP	5	0/6	164.7	0.406	1.80×10^5	0.008
WLOP	15	0/6	172.5	0.457	1.46×10^8	1×10^{-4}
corrupt baseline		0/6	145.3	0.517	2.37×10^5	—
clean baseline		6/6	1.0	0.010	1.13×10^5	—
`PowerRemap`		6/6	1.0	$0.517^\dagger$	$2.37\times10^{5\,\dagger}$	—

Table 13: Stage 1, deployment-fair: WLOP (Huang et al., 2009) without surface projection. Iteration counts $\{1, 2\}$ partially uniformize the cloud without rescuing; counts $\{5, 15\}$ destabilize. † `PowerRemap` acts on the activation $\mathbf{Y}_{4,7}$ and leaves the input cloud unchanged, so its input statistics equal those of the corrupt cloud.

H.4.2 LOP with $\mathbb{S}^2$ projection (oracle ablation)

To separate the failure into "surface-agnostic does not work" versus "local repulsion at fixed N does not work," we run an oracle ablation: identical WLOP iteration with a final radial re-projection to $\mathbb{S}^2$ after each step, $q_i \leftarrow q_i/\|q_i\|$. This injects the surface as a hard constraint and is therefore *not* deployment-fair; we use it only to calibrate the failure. Table 14 reports the result.

Variant	**Iters**	**rescue**	$\bar{C}$	$\overline{\mathrm{COV}_V}$	$\overline{E_2}$	$\overline{\mathrm{cham}}$
WLOP $+\Pi_{\mathbb{S}^2}$	2	0/6	144.5	0.422	1.58×10^5	0.107
WLOP $+\Pi_{\mathbb{S}^2}$	5	0/6	166.5	0.406	1.83×10^5	0.107
WLOP $+\Pi_{\mathbb{S}^2}$	15	0/6	169.7	0.455	1.50×10^8	0.110

Table 14: Stage 1, oracle ablation: WLOP with $\mathbb{S}^2$ projection after each iteration. Despite explicit surface knowledge, no iteration count produces a rescue. Surface projection is not sufficient to push WLOP past the basin transition at fixed N, because radial re-projection does not prevent particle coincidence: two particles whose updates point in similar directions land at the same location on $\mathbb{S}^2$, reproducing the long-iteration collapse.

The oracle variant exhibits the same qualitative pattern as the deployment-fair variant: at it. 2 both achieve COV_V=0.422; at it. 15 both collapse with E_2 exceeding 10^8. Surface projection therefore does not change the verdict at any iteration count: WLOP-style local-repulsion uniformization at fixed N, with or without $\mathbb{S}^2$ projection, fails to cross the basin separatrix.

H.4.3 Joint interpretation of Stage 1

The deployment-fair and oracle variants together support a stronger conclusion than either alone. The deployment-fair variant fails for two distinct reasons depending on iteration count: at low counts the cloud

remains insufficiently uniformized to cross the basin transition ($\mathrm{COV}_V \in [0.42, 0.44]$ versus the clean reference COV_V=0.010); at high counts the cloud destabilizes off the manifold. A natural concern is that the high-iteration failure is specifically the off-manifold drift, and that surface knowledge would fix it. The oracle ablation rules this out: even with explicit $\mathbb{S}^2$ projection, the WLOP iteration does not reach a uniformity level that crosses the rescue threshold, and at high iteration counts particles still coincide. The conclusion is that local-repulsion uniformizers, regardless of surface knowledge, do not produce sufficiently uniform fixed-N point sets on $\mathbb{S}^2$ to flip the diffusion trajectory back to the sphere basin. Together, Stage 1 delivers 0/48 rescues across iteration counts and surface conditions.

H.5 Stage 2: noise injection at every denoising step

We test EDM-style variance-preserving noise injection at every denoising step. At step t_{inj}, we replace the latent $\mathbf{x}_t$ with $\mathbf{x}'_t = \sqrt{1-s^2}\,\mathbf{x}_t + s\,\boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, on the conditional slot of the CFG batch (the unconditional slot is left untouched, consistent with the rest of the analysis being restricted to the conditional stream; §B.1). The remainder of reverse diffusion runs unmodified. We sweep $t_{\mathrm{inj}} \in \{0, 1, \ldots, 7\}$ across the entire DDIM schedule and two strengths $s \in \{0.3, 0.6\}$, giving $8 \times 2 \times 6 = 96$ trials. Strength s=0.6 is aggressive: the latent retains $\sqrt{1-0.6^2}$=0.8 of its norm and is mixed with a fresh Gaussian of magnitude 0.6 relative to unit variance. Sweeping the entire schedule preempts the question of whether a different choice of t_{inj} would have rescued.

Table 15 reports the per-cell mean connected-component count $\bar{C}$. No setting produces a sphere: 0/96 rescues, with C ranging from 55 to 153. The finest-grained pattern is monotone with t_{inj} at fixed s: injection at the earliest steps disturbs the trajectory most ($\bar{C}$=71.2 at (t_{inj}=0, s=0.6), down from 143.0 at (t_{inj}=7, s=0.6)), reflecting that an early perturbation has more reverse-diffusion steps over which to propagate. None of these disturbances flips basin allocation: even where $\bar{C}$ drops to 71, the resulting mesh is fragmented rather than spherical (mean chamfer at (t_{inj}=0, s=0.6) is 0.112, identical to the corrupt baseline).

	injection step t_{inj}								
strength s	0	1	2	3	4	5 ($\tau^\star$)	6	7	**rescue**
0.3	115.8	130.8	140.3	141.7	145.2	145.7	146.3	147.0	0/48
0.6	71.2	102.3	129.7	137.8	139.3	143.2	142.0	143.0	0/48
corrupt baseline (no intervention): $\bar{C}$=145.3									—

Table 15: Stage 2: per-cell mean connected-component count $\bar{C}$ for variance-preserving noise injection at every denoising step, 6 seeds per cell. Step 5 corresponds to the bifurcation step $\tau^\star \approx 5$ identified by the dip-test analysis in §5. No cell rescues at either strength. The strict-rescue (C=1 and chamfer below clean +0.020) verdict is 0/96.

The result at the bifurcation step deserves explicit comment. §5 identifies $\tau^\star \approx 5$ as the step at which the reverse-time potential bifurcates and the trajectory commits to a basin (Figs. 5b, 6; Hartigan dip test rejects unimodality from t=5 onward). A naive reading would predict that noise added at $\tau^\star$ should be the most effective intervention, since trajectories there are poised on the basin separatrix. The data refute that prediction sharply: at t_{inj}=5, even the aggressive s=0.6 injection moves $\bar{C}$ only from 145.3 (corrupt baseline) to 143.2, and produces 0/6 rescues. The bifurcation-theoretic interpretation is consistent: the basin separatrix is sharp, and a trajectory that has accumulated bias from earlier denoising steps (the directional drift in $\mathbf{Y}_{4,7}$ documented in §C.5) cannot be transported back across the separatrix by symmetric Gaussian noise. The Stage 2 result therefore both fails as a Meltdown rescue and supports the diffusion-dynamics account.

H.6 Stage 3: best-of-K initial-noise selection

For each of the 6 seeds, we run K=4 independent reverse-diffusion trajectories from different initial noises $\mathbf{x}_T$ and the identical input cloud $\mathcal{P}_\rho$, recording the per-attempt mesh and its component count. Table 16 reports

the resulting best-of-K chamfer-rescue rate as a function of K. The minimum C across all 24 independent draws is 134. At ρ_{crit}=0.4, the speckle attractor's basin of attraction is dominant enough that all 24 initial noises tested land in it, and best-of-K selection at $K \leq 4$ does not rescue.

Best-of-K rate	K=1	K=2	K=3	K=4
sphere rescue (chamfer)	0/6	0/6	0/6	0/6

Table 16: Stage 3: best-of-K initial-noise selection (Ma et al., 2025). C=1 is the selection criterion and is computable from the output mesh alone. Across $K \in \{1, 2, 3, 4\}$ on 6 seeds, the rescue rate is 0. Total 24 independent $\mathbf{x}_T$ draws, minimum C=134.

H.7 Joint summary

Across the three stages of the causal chain, the canonical SOTA deployment-fair baselines together yield 0/168 sphere rescues at ρ_{crit}, while `PowerRemap`, applied at the activation site identified by the localization analysis of §3.2, rescues 6/6 on the same seeds with output quality matching the clean baseline. Table 17 consolidates the headline numbers.

Stage	**Baseline (representative)**	**Trials**	**Sphere rescue**
1 (input)	WLOP (Huang et al., 2009), it. $\in \{1, 2, 5, 15\}$	48	0/48
1 (oracle)	WLOP $+\Pi_{\mathbb{S}^2}$, it. $\in \{2, 5, 15\}$	18	0/18
2 (diffusion)	EDM-style noise inj. (Karras et al., 2022a), all 8 steps $\times$ 2 strengths	96	0/96
3 (alternative)	Best-of-K (Ma et al., 2025), $K \in \{1, 2, 3, 4\}$	24	0/24
—	`PowerRemap` (γ=100, $\mathbf{Y}_{4,7}$)	6	**6/6**

Table 17: Headline rescue rates. The Stage 1 oracle row is included for calibration only and is not deployment-fair. `PowerRemap` is evaluated on the same 6 seeds as the baselines.

The three deployment-fair baselines fail for distinct, mechanistically informative reasons. Stage 1 (WLOP) fails because local-repulsion uniformization at fixed N does not reach a uniformity level that crosses the basin separatrix — a conclusion confirmed by the oracle ablation. Stage 2 (noise injection) fails because the basin separatrix is sharp and the trajectory's accumulated directional bias in $\mathbf{Y}_{4,7}$ cannot be transported across by symmetric Gaussian noise — a conclusion consistent with the diffusion-dynamics analysis of §5 and the directional-drift result of §3.3.1. Stage 3 (best-of-K) fails because the speckle attractor's basin is dominant at ρ_{crit}, so seed retry does not transport the trajectory.

The unified picture is that no single intervention upstream of the identified commit site — in input space, in mid-trajectory noise, or in initial-noise resampling — rescues at the operating point we study. `PowerRemap` succeeds because it acts at the commit site itself, on the directional content of the activation that the mechanistic analysis identifies as the lever (§3.2, §3.3.1).