Supplementary Material

Unsupervised Self-debiasing of Text-to-Image Diffusion

A Training the Projector Network

To map intermediate diffusion activations to semantic space, we use CLIP's image encoder as a reference. For each timestep t, the diffusion model yields an intermediate activation h_t at the U-Net bottleneck. We train a projector network $g_\psi(h_t, t)$ to predict the CLIP embedding of the final generated image.

Each training sample consists of h_t, the timestep t, and three CLIP embeddings: one true and two perturbed positives, used to form contrastive training pairs. We optimize the projector using the NT-Xent loss and the Adam optimizer with a learning rate of 10^{-4} over 30 epochs (batch size 256). Early stopping is based on validation cosine similarity.

Figure 1 shows a UMAP projection, illustrating alignment between predicted and ground-truth CLIP embeddings.

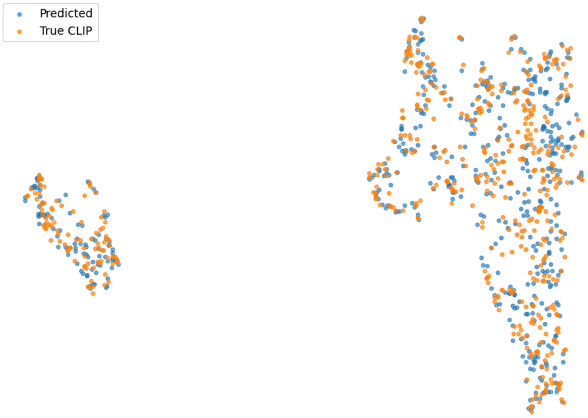


Figure 1: UMAP visualization showing alignment between original and predicted CLIP vectors for 'faces'. Cosine similarity = 0.9315.

B Imbalanced Target Distributions

Our method generalizes to arbitrary target distributions over discovered semantic clusters. This is especially useful for simulating specific environments, such as demographic skews in urban settings. Figure 2 shows results for a 70:30 male:female target.

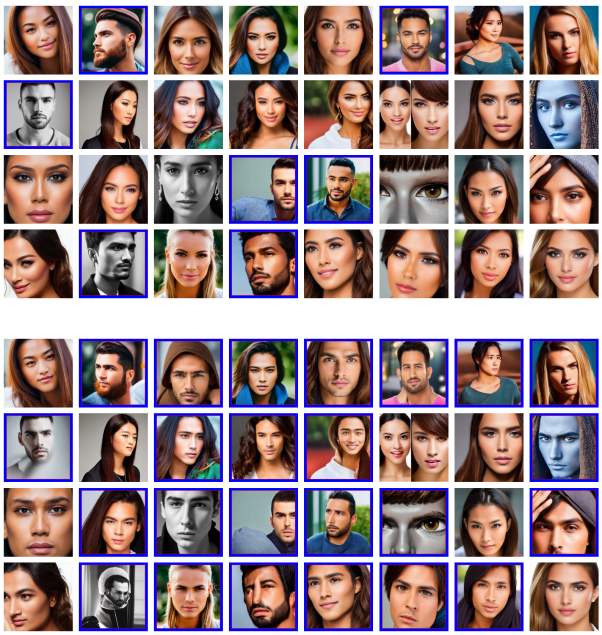


Figure 2: Imbalanced Target Debiasing. Top: Original (unbiased) generation. Bottom: Debiased output using Self-Debias steered toward 70% male subjects.

C Using OpenCLIP Image Encoder

We also evaluate performance when using OpenCLIP instead of CLIP. As shown in Figure 3, our method remains effective at debiasing, yielding better gender balance even with a different encoder.

Figure 3: OpenCLIP Debiasing. Top: Without debiasing. Bottom: Debiased output shows improved male representation (14 male-presenting subjects vs 3 originally).

D Unconditional Model Evaluation

Figure 4 shows qualitative diversity confirming the debiasing of unconditional diffusion.

Figure 4: DDIM Debiasing — Gender: Left = baseline, Right = debiased. Debiased output has equal male/female ratio (4 each) versus only 2 males in original. Nearest unsupervised baseline method, **UCE** couldn't do unconditional debiasing because of its reliance on text embeddings.

E Applicability to Transformer-based Backbones

While our main experiments focus on diffusion models with U-Net backbones, our framework is not restricted to this architecture. The only requirement is access to intermediate semantic representations during the sampling process. For transformer-based diffusion and rectified-flow models such as FLUX and HiDream, the natural analogue of our h-space is the sequence of hidden states corresponding to image tokens at denoising step t and transformer layer l.

These token-level hidden states contain rich semantic information, similar in spirit to the bottleneck activations in U-Net models. In principle, our Semantic Projection Module (Sec. 3.1 in the main paper) can be trained to align these hidden states with a semantic embedding space (e.g., CLIP or OpenCLIP). Subsequently, the Semantic Mode Discovery and Self-debiasing Modules (Secs. 3.2 and 3.3 in the main paper) can be applied in exactly the same manner, using the projected transformer features instead of projected U-Net features.

A few architecture-specific considerations arise:

- **Hook points:** In transformers, intermediate hidden states are available at every layer. Selecting which layer(s) to hook is an open design choice. Earlier layers may encode low-level structural information, while deeper layers capture high-level semantics.
- **Sequence length:** Transformer backbones operate over long token sequences, leading to higher memory and compute costs. Pooling strategies (e.g., mean pooling, attention pooling, or downsampling subsets of tokens) can reduce dimensionality while retaining semantic fidelity.
- **Token granularity:** Unlike U-Net activations, token embeddings correspond to discrete spatial patches or latent codes. The choice of token granularity (e.g., patch

size or latent resolution) will impact both semantic separability and computational efficiency.

While these considerations require empirical exploration, they are orthogonal to our main contribution. Our framework is fully compatible with transformer-based architectures, and extending SelfDebias to such backbones is a promising direction for future work.

F Ablation on Number of Gradient Update Steps

In our main framework, we apply a single gradient update to the h-space per denoising step (Eq. (8)) for efficiency. To study the effect of taking multiple gradient updates, we perform an ablation with 1, 2, 3, and 5 inner updates per denoising step.

# Steps	1	2	3	5
FD ↓	0.009	0.0088	0.0087	0.0087
FID ↓	70.52	70.85	77.53	88.32

Table 1: Ablation on the number of gradient updates per denoising step. FD improves marginally with more steps, while FID degrades as the number of updates increases. This is for gender attribute for faces data.

We observe that Fairness Discrepancy (FD) improves slightly as the number of updates increases, indicating marginal gains in bias mitigation. However, FID worsens substantially beyond two steps. This behavior can be explained as follows: each additional update step applies stronger corrective pressure toward the uniform cluster distribution, which reduces demographic skew but simultaneously pushes the denoising trajectory farther away from the model's natural manifold. As a result, semantic alignment improves (lower FD) but visual fidelity degrades (higher FID). Beyond two steps, this over-correction leads to overshooting and accumulation of artifacts. Therefore, our default choice of one gradient update per denoising step achieves the best trade-off between fairness and image quality, while also remaining computationally efficient.

G Runtime Analysis

We report a runtime comparison between our framework and the vanilla diffusion model in Table 2. Our approach introduces two one-time offline stages: (i) training the projector network and (ii) identifying semantic modes via clustering. These costs are incurred once per prompt family, after which the centroids can be reused across related prompts (as demonstrated in Sec. 4.3 where centroids derived from faces are reused for occupation-based prompts). The per-sample overhead during inference is modest: we apply lightweight gradient updates in h-space without backpropagating through the full U-Net.

Overall, the results show that while our method introduces a one-time offline cost, the inference-time overhead remains practical, roughly 1.87X sampling time relative to vanilla

Table 2: Runtime comparison on an RTX A5000 GPU. Preprocessing is a one-time offline cost; scope indicates whether it is per attribute or per prompt family.

Method	**One-time preprocessing (scope)**	**Sampling (per image)**
Vanilla Stable Diffusion	None	1.78 s
H-Guidance	2.93 h / attribute	2.85 s
Self-Discovery	4.25 h / attribute	3.02 s
Ours	3.63 h / prompt family	3.34 s

diffusion. Crucially, because centroids can be reused across prompt families, our framework enables real-time debiasing applications once this offline stage has been performed.

H Hyper-parameter Values and Ablation

We use a small, fixed set of hyper-parameters across all experiments, without tuning them within a prompt family.

- **Stage-1 cluster count k:** chosen via a silhouette-score sweep on projected features (typically 2–5 across prompt families).
- **Stage-2 refinement:** recursive spectral splits with a default $d_{\max} = 3$. We set $s_{\min} = 0.05N$ (i.e., a minimum leaf size of 5% of the dataset) to prevent very small, impure leaves whose centroids would be unrepresentative. Because $d_{\max}$ controls *how deep* we recurse, it is more sensitive at shallow depths (coarse vs. mid-level partitions). Beyond a certain depth, however, $s_{\min}$ blocks further splits, so increasing $d_{\max}$ has diminishing effect.
- **Temperature α:** fixed globally at $\alpha = 8$, controlling the sharpness of the soft assignment in Eq. (3).

Sensitivity of α (gender). Table 3 shows a mild trend: α=6 slightly lowers FD, but at a small cost to FID; α=8 attains the best overall balance (lower FID and stable FD), and we use it as default.

α	4	6	8	10	12	16
FD ↓	0.017	**0.014**	0.015	0.015	0.016	0.017
FID ↓	87.53	87.28	**87.08**	87.33	87.72	88.19

Table 3: Ablation on α for **gender**. While α=6 slightly lowers FD, α=8 offers the best *balance* (lower FID and stability), so we adopt α=8 as default.

Sensitivity of $d_{\max}$ (race). Table 4 highlights that deeper recursion up to $d_{\max}$=3 is beneficial (coarse → mid-level structure). Pushing deeper to $d_{\max}$=4 starts to over-fragment, slightly hurting FD/FID. From $d_{\max}$=4 to 5, changes are small; for $d_{\max} > 5$, we

observe *no* changes because $s_{\min}$ prevents further splits, so additional depth is effectively inert.

$d_{\max}$	2	3	4	5
FD ↓	0.239	**0.237**	0.258	0.253
FID ↓	87.82	**87.08**	87.63	87.85

Table 4: Ablation on $d_{\max}$ for **race**. Early increases in depth are meaningful; beyond $d_{\max}$=4, $s_{\min}$ blocks most additional partitions, so $d_{\max}$=5 yields only minor changes, and $d_{\max}$>5 has no effect. Default $d_{\max}$=3 achieves the best trade-off.

Overall, FD/FID are robust around the defaults, with $d_{\max}$ mattering primarily at shallow depths and quickly saturating once $s_{\min}$ halts further splits. This supports using a single configuration (α=8, $d_{\max}$=3) across a prompt family without per-attribute tuning.

I Two-Stage Clustering

CLIP space exhibits a few *dominant* axes with finer variation *within* those groups, so we use a two-stage procedure that matches this structure.

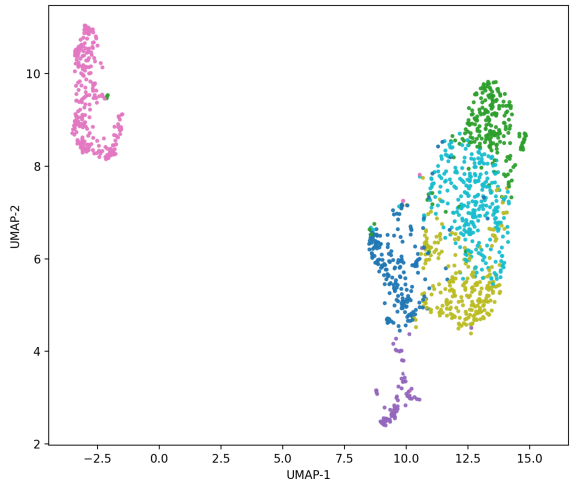
Stage-1 (coarse split). Choose k via the silhouette criterion to capture the dominant partition.

Stage-2 (local refinement). Recursively refine *within* each Stage-1 cluster by spectral clustering, gated by $(s_{\min}, d_{\max})$, to reveal fine-grained variants while preserving the coarse split. Any Stage-1 cluster with $|c| \geq s_{\min}$ is eligible for refinement.

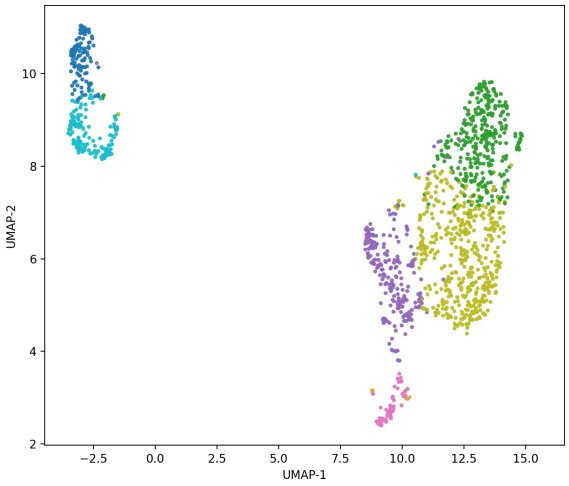
Why not a flat global k? With a single global k, one cannot control *where* splits land: even at the same leaf count (e.g., $k = 6$), the optimizer may allocate most splits to the larger coarse group and leave the smaller group unsplit. Our locally gated refinement ensures both large and smaller coarse groups are split when warranted, without retuning k. Figure 5 shows such a scenario.

J Debiasing Non Societal Biases

Our method works not only for societal biases but also for biases while generating niche or abstract concepts as shown in Sec. 4.4 of the main paper. We here provide one more example of a non-societal bias where we debias images generated when the SD model is prompted with the prompt *'a photo of food on a table'*. On seeing the clusters formed, it is evident that larger cluster belongs to images containing only veggies and smaller cluster corresponds to images with burgers. We have shown the clusters in Fig. 6. We have debiased using the centroids found using the SelfDebias method and the result is shown in Fig. 8. We have also shown in Fig. 7, representative images from each cluster, randomly sampled. Since the number of available images in the smaller cluster is limited, and SelfDebias primarily shifts images near the decision boundary to the opposite cluster, we



(a) Flat clustering (k=6). All splits land in the larger coarse group; the smaller group remains unsplit.



(b) Two-stage (ours, 6 leaves). Stage-1 isolates coarse groups; Stage-2 refines within groups ($|c| \geq s_{\min}$).

(c) Sample images closest to the centroid of the pink cluster in (a); clearly show a mixture of races.

(d) Sample images closest to the centroids of the blue and sky-blue clusters in (b) (top: blue, bottom: sky-blue); clearly show separation by race.

Figure 5: Human face clustering and race-based selections in projected CLIP space (UMAP) and zero-shot filtering. (a,b) compare flat vs. hierarchical clustering with the same total leaves (k=6), showing control over *where* splits occur. (c) shows a single mixed row across races; (d) shows two separated rows by race, with non-overlapping images.

were unable to achieve a truly uniform distribution without compromising image quality. But we are able to decrease the bias as shown in the Fig. 8.

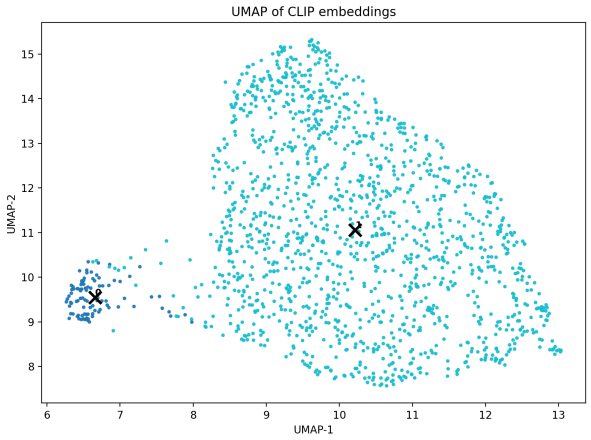


Figure 6: CLIP clusters formed when prompted with *'a photo of food on a table'*.

Figure 7: Representative images from each of the CLIP clusters, each row corresponds to a cluster.

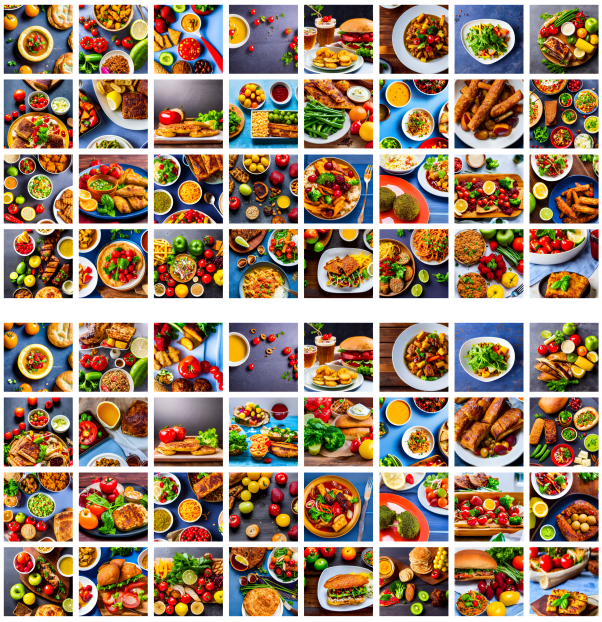


Figure 8: Top: Generated images using vanilla Stable Diffusion 1.5. Bottom: Generated images using SelfDebias. The debiased set contains 5 burgers, as opposed to only 1 using the original model.