SOLA: STOCHASTIC ONE-LOSS AUTOENCODERS WITH POPULATION-DERIVED LATENT GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational autoencoders (VAEs) typically obtain stochastic latent representations by balancing reconstruction against a Kullback–Leibler (KL) regularization term, introducing a problem-dependent trade-off that can lead to posterior collapse. We ask whether a stochastic generative autoencoder can retain a principled KL interpretation without optimizing a KL loss at all. We introduce the *Stochastic One-Loss Autoencoder* (SOLA), which derives both latent stochasticity and sampling geometry directly from the encoded population. SOLA combines a rank-reduced orthogonal latent representation with random latent mixing, inducing Gaussian perturbations whose covariance is determined by the geometry of each encoded batch. We show that the resulting stochastic representation admits a geometry-matched Gaussian reference for which the batch-averaged KL divergence is exactly constant for fixed bottleneck dimension and noise scale, independently of the network parameters and learned singular spectrum. Its optimization gradient is therefore identically zero. Accordingly, SOLA is trained using reconstruction alone while retaining a stochastic latent representation. After training, the learned batch geometries are aggregated into a fixed global representation that supports deterministic inference and a population-derived Gaussian sampling rule, without a learned variance head, KL weight, or KL schedule.

Across six synthetic, structured, and real-world datasets, SOLA achieves competitive reconstruction and generation using the same noise scale $\sigma = 0.1$ throughout. In contrast, both the preferred VAE KL weight β and the qualitative response to changing it vary substantially across problems. Controlled ablations separate the effects of dimensionality reduction, orthogonal latent geometry, and stochastic mixing, while empirical diagnostics assess the latent statistics underlying the Gaussian sampling approximation. Comparisons with modified VAE objectives and alternative generative autoencoders, including KL annealing, Free Bits, σ-VAE, capacity-controlled β-VAE, RAE, WAE, and VQ-VAE, further show that competitive generative performance can be obtained without explicitly optimizing a prior-matching objective or fitting a separate latent density for SOLA's native generator. The code is open-source and available in the supplementary material.

1 INTRODUCTION

Learning latent representations that support reconstruction, interpolation, and generation is a central objective of generative modeling. Variational autoencoders (VAEs) (Kingma & Welling, 2014) achieve this by combining a stochastic encoder with a generative decoder. In the conventional Gaussian formulation, the encoder predicts sample-specific means and variances, while training balances reconstruction against a Kullback–Leibler (KL) penalty toward a prescribed prior, $\mathcal{L} = \mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{KL}}$. The KL term makes the latent representation compatible with the distribution used for generation, but the resulting trade-off can be difficult to balance. Excessive prior-matching pressure may suppress information and contribute to posterior collapse (Lucas et al., 2019; He et al., 2019; Mounayer et al., 2026), while the effective value of β can vary substantially across problems.

Many approaches modify this balance. β-VAE and capacity-controlled variants alter the strength or information capacity of the KL term (Higgins et al., 2017; Burgess et al., 2018), FactorVAE, β-TCVAE, and InfoVAE introduce alternative latent regularization objectives (Kim & Mnih, 2018;

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

Chen et al., 2018; Zhao et al., 2019), and annealing or lagging-inference strategies modify its optimization (Fu et al., 2019; He et al., 2019). Other methods change the prior or the level at which distribution matching is imposed, including VampPrior (Tomczak & Welling, 2018), Wasserstein autoencoders (Tolstikhin et al., 2018), vector-quantized autoencoders (Van den Oord et al., 2017), and regularized autoencoders with post-training density estimation (Ghosh et al., 2020). Despite their differences, generation in these approaches relies on a prescribed, learned, matched, or post-hoc fitted sampling distribution.

We instead ask whether a stochastic generative autoencoder can obtain its sampling geometry directly from the encoded population, without optimizing a prior-matching loss. We introduce the *Stochastic One-Loss Autoencoder* (SOLA). SOLA combines the rank-reduced latent representation of Rank Reduction Autoencoders (RRAEs) (Mounayer et al., 2024) with random latent mixing. For each encoded batch, a truncated SVD exposes a K-dimensional orthogonal representation whose axes and scales are determined by the population itself. Randomly mixing these coordinates induces conditionally Gaussian sample-wise perturbations with a covariance shared across the batch and determined by its latent geometry. Thus, stochasticity is introduced without predicting a separate variance for every sample.

The central property of SOLA is that this stochastic representation admits a natural geometry-matched Gaussian reference. If $q_i^{\mathcal{B}}$ denotes the stochastic representation of sample i in batch $\mathcal{B}$, and $p_{\mathcal{B}}$ the corresponding reference distribution, we show that

$$\frac{1}{B}\sum_{i=1}^{B} D_{\mathrm{KL}}\left(q_i^{\mathcal{B}} \,\|\, p_{\mathcal{B}}\right) = \frac{K}{2}\left(\sigma^2 - \log(\sigma^2)\right).$$

For fixed bottleneck dimension K and perturbation scale σ, this quantity is independent of the network parameters and of the singular values learned from the batch. Its optimization gradient is therefore identically zero. SOLA can consequently be trained using reconstruction alone while still possessing a stochastic latent representation with an explicit distributional interpretation. After training, the geometries observed across batches are aggregated into a fixed global basis and spectrum, providing deterministic inference and motivating a population-derived Gaussian sampling rule.

Across six synthetic, structured, and real-world datasets, SOLA achieves competitive reconstruction and generation using the same fixed noise scale $\sigma = 0.1$ throughout. In contrast, the VAE exhibits markedly different responses to the KL weight across problems, with both the preferred value and the qualitative sensitivity trend changing between datasets. We further compare against modified VAE objectives and alternative generative autoencoders, including KL annealing, Free Bits, σ-VAE, capacity-controlled β-VAE, RAE, WAE, and VQ-VAE, and empirically assess the right-singular-coordinate statistics that motivate SOLA's post-training Gaussian generator.

Our main contributions are:

- We introduce SOLA, a stochastic generative autoencoder in which both latent perturbations and the eventual sampling geometry are derived from the encoded population, without sample-specific variance prediction or an optimized prior-matching term.
- We derive the induced stochastic distributions and prove that their batch-averaged KL divergence to a geometry-matched Gaussian reference is exactly constant for fixed K and σ. This yields reconstruction-only training. A fixed post-training generator is then constructed from the aggregated population geometry, with its Gaussian sampling approximation analyzed separately.
- We evaluate SOLA across six datasets and perform controlled component, mixing, and sensitivity studies that disentangle bottleneck dimensionality, orthogonal population geometry, and stochastic mixing. SOLA uses the same $\sigma = 0.1$ throughout, while the VAE exhibits strongly problem-dependent sensitivity to β. We further validate the right-singular-coordinate statistics, perform a post-hoc latent-density control, and compare against tuned and alternative generative autoencoders.

2 METHODOLOGY

2.1 ORTHOGONAL LATENT REPRESENTATION

SOLA builds on the latent-space construction of Rank Reduction Autoencoders (RRAEs) (Mounayer et al., 2024). Let $X \in \mathbb{R}^{F\times B}$ denote a batch of B samples and let $Y = f_\theta(X) \in \mathbb{R}^{L\times B}$ be its encoded representation. For a chosen bottleneck dimension[1] $1 \le K \le \operatorname{rank}(Y) \le \min(L, B)$, we compute the rank-$K$ truncated SVD

$$Y_K = U_K \Sigma_K V_K^\top, \qquad U_K \in \mathbb{R}^{L\times K}, \quad \Sigma_K \in \mathbb{R}^{K\times K}, \quad V_K^\top \in \mathbb{R}^{K\times B}.$$

We define the intrinsic bottleneck coordinates as

$$A_K := \Sigma_K V_K^\top = U_K^\top Y \in \mathbb{R}^{K\times B}, \qquad Y_K = U_K A_K.$$

Since $V_K^\top V_K = I_K$,

$$\boxed{A_K A_K^\top = \Sigma_K^2.}$$

Thus, the latent directions are mutually orthogonal across the sample dimension, while their scales are given by the retained singular values. Numerical aspects of the truncated SVD follow the RRAE construction (Mounayer et al., 2024). The batch-wise SVD introduces both computational and numerical considerations. A dense reduced SVD of $Y \in \mathbb{R}^{L\times B}$ has complexity $\mathcal{O}(LB\min(L, B))$, so its training-time cost can become significant for large encoded widths or batch sizes. A detailed empirical analysis of this overhead is provided in Appendix A.13. In addition, SVD backpropagation can become ill-conditioned when singular values are small or closely clustered. We follow the stabilized SVD differentiation procedure used in the RRAE framework (Mounayer et al., 2024). After training, the batch-dependent representations are consolidated into a fixed basis U_f and global geometry through a singular-value-weighted aggregation of converged batches. The complete construction, including the common-subspace result and convergence of the aggregated geometry under the stated representative-batch assumptions, is given in Appendix A.1.

2.2 SOLA: STRUCTURED STOCHASTIC BOTTLENECK

SOLA introduces stochasticity by randomly mixing the structured latent coordinates within each batch. Let $E \in \mathbb{R}^{B\times B}$ have independent standard Gaussian entries and define

$$G = I_B + \frac{\sigma}{\sqrt{B}} E, \qquad E_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1),$$

where $\sigma > 0$ controls the perturbation amplitude. The stochastic bottleneck is

$$\widetilde{A}_K = A_K G, \qquad \widehat{X} = g_\phi\left(U_K \widetilde{A}_K\right).$$

Writing $A_K = [a_1, \ldots, a_B]$, the stochastic latent vector associated with sample i satisfies

$$\boxed{\widetilde{a}_i \mid A_K \sim \mathcal{N}\left(a_i, \frac{\sigma^2}{B}\Sigma_K^2\right).}$$

Hence, conditional on the encoded batch, all samples share the same stochastic covariance, which is determined directly by the singular spectrum of the encoded population rather than predicted sample by sample. Moreover, the full $B \times B$ mixing matrix need not be instantiated. Using $V_K^\top V_K = I_K$,

$$\boxed{\widetilde{A}_K \overset{d}{=} A_K + \frac{\sigma}{\sqrt{B}}\Sigma_K \Xi, \qquad \Xi_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1).} \tag{1}$$

Thus, implementation requires only the $K \times B$ matrix Ξ. The full derivation and its relation to linear, affine, and convex mixing are given in Appendix A.2.

[1] As in RRAEs, SOLA requires $K \le B$, since the rank of the batch matrix cannot exceed the batch size.

2.3 CONSTANT GEOMETRY-MATCHED KL DIVERGENCE

For a training batch $\mathcal{B}$, define its intrinsic geometry by

$$D_{\mathcal{B}} = \frac{1}{B}\Sigma_K^2 = \frac{1}{B}A_K A_K^\top. \tag{2}$$

This is the uncentered second-moment geometry induced by the SVD coordinates, not in general their centered empirical covariance. The stochastic latent distribution of sample i is

$$q_i^{\mathcal{B}} = \mathcal{N}\left(a_i, \sigma^2 D_{\mathcal{B}}\right),$$

and we associate with the same batch the geometry-matched Gaussian reference

$$p_{\mathcal{B}} = \mathcal{N}\left(0, D_{\mathcal{B}}\right).$$

This distribution is an analytical reference rather than a prior optimized during training. Although the individual KL divergence depends on a_i, its average over the batch is exactly

$$\boxed{\frac{1}{B}\sum_{i=1}^{B} D_{\mathrm{KL}}\left(q_i^{\mathcal{B}} \,\|\, p_{\mathcal{B}}\right) = \frac{K}{2}\left(\sigma^2 - \log\sigma^2\right).} \tag{3}$$

For fixed K and σ, this quantity is independent of the network parameters and of the singular values of the current batch. Its optimization gradient is therefore exactly zero, and SOLA is trained using reconstruction alone:

$$\boxed{\mathcal{L}_{\mathrm{SOLA}} = \mathcal{L}_{\mathrm{rec}}.}$$

This result holds for every batch satisfying the rank condition and does not require Gaussianity, centering, or independence of the clean latent coordinates. The complete proof is provided in Appendix A.3. The batch geometries need not converge individually. Instead, converged batches are aggregated into a fixed global representation. If

$$W_f = [U_1\Sigma_1 \quad \cdots \quad U_M\Sigma_M] = U_f\Sigma_f R^\top,$$

and N_{rep} is the total number of represented samples, the resulting global intrinsic geometry is

$$\boxed{\widehat{D}_f = \frac{\Sigma_f^2}{N_{\mathrm{rep}}}.} \tag{4}$$

Appendix A.1 gives the exact finite-sample aggregation identity and its population-level convergence under the stated assumptions.

2.4 TRAINING, EVALUATION, AND GENERATION

The three operating regimes of SOLA are summarized in Figure 1. During training, a truncated SVD is computed for each encoded batch and the stochastic bottleneck in Equation 1 is used. Only the reconstruction loss is optimized. After training, inference is deterministic and requires no batch-wise SVD. The encoded representation is projected onto the fixed global basis,

$$Y_f = U_f U_f^\top f_\theta(X), \qquad \widehat{X} = g_\phi(Y_f).$$

For generation, the sampling geometry is derived from the learned SVD structure. Writing $v_i \in \mathbb{R}^K$ for a column of $V_K^\top$, the normalized right-singular coordinates are

$$w_i = \sqrt{B}\, v_i.$$

Their uncentered second moment is exactly the identity by SVD orthogonality. Appendix A.4 shows that a normalized right-singular coordinate drawn from the pooled batch–sample ensemble is approximately described by

$$w \overset{d}{\approx} \mathcal{N}(0, I_K). \tag{5}$$

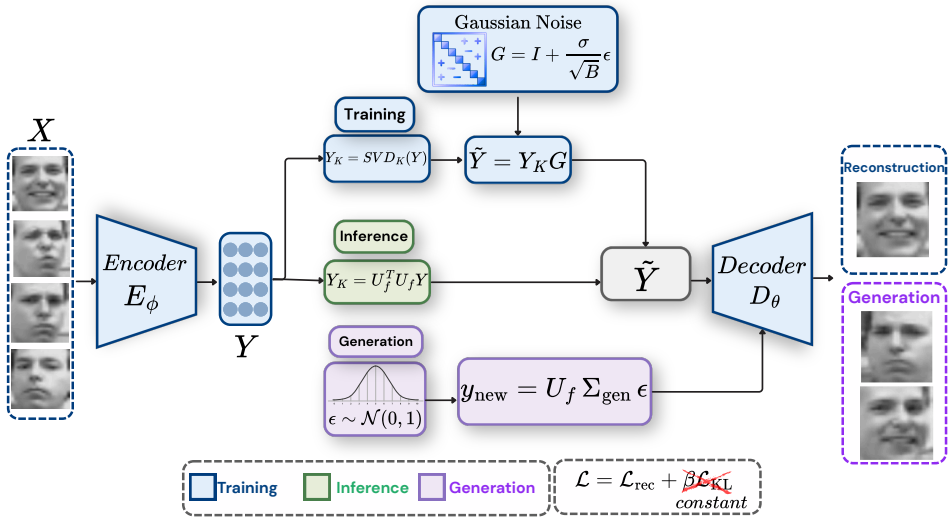


Figure 1: Overview of SOLA training, deterministic inference, and generation. During training, stochastic mixing is applied to the batch-wise orthogonal bottleneck, inference uses a fixed global basis, and generation samples new coordinates in the learned latent geometry.

This approximation, which is distinct from the exact constant-KL result, relies on a small dimension-normalized pooled mean $\|\mu_W\|_2/\sqrt{K} \ll 1$ and approximate Gaussian ensemble shape. Both properties are tested directly at finite batch sizes in Appendix A.5. Combining this Gaussian reference with the stochastic perturbation and the global geometry in Equation 4 gives the final generation scale

$$\boxed{\Sigma_{\text{gen}} = \frac{\sqrt{1+\sigma^2}}{\sqrt{N_{\text{rep}}}}\Sigma_f.}$$

The complete derivation is provided in Appendix A.6. Given M desired samples, we draw $\epsilon \in \mathbb{R}^{K\times M}$ with $\epsilon_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and generate

$$\boxed{A_{\text{gen}} = \Sigma_{\text{gen}}\epsilon, \qquad X_{\text{gen}} = g_\phi\left(U_f A_{\text{gen}}\right).}$$

Thus, the SVD provides the structured population geometry used to define the bottleneck and post-training sampler, while random mixing introduces the stochastic perturbations used during training. SOLA combines these two mechanisms while optimizing only the reconstruction loss.

3 Results

We evaluate SOLA across six synthetic, structured, and real-world datasets. Section 3.1 compares SOLA with the standard VAE while explicitly examining the dependence of VAE performance on the KL weight β. We first use the canonical $\beta = 1$ objective as a common untuned reference and then compare against the best-performing problem-specific value $\beta^\star$. This separates the behavior of a fixed reconstruction–KL balance from the best performance attained after tuning the VAE regularization. Section 3.2 broadens this comparison to several modified VAE objectives and alternative autoencoder-based generative approaches. Finally, Section 3.3 examines the contributions of bottleneck dimensionality, SVD-derived population geometry, and stochastic mixing. Further sensitivity and diagnostic experiments are reported in the appendix, including alternative constructions

of G (Appendix A.9), retained rank (Appendix A.10), batch size (Appendix A.11), noise scale (Appendix A.12), and time complexity (Appendix A.13). All needed details for reproducibility can be found in Appendix A.14 or in the code available in the supplementary material.

The image experiments are controlled comparisons under matched convolutional backbones rather than state-of-the-art image-generation benchmarks.

3.1 Standard VAE Comparison and KL-Weight Sensitivity

We compare SOLA with the standard VAE and examine how its performance depends on the KL weight β. The conventional $\beta = 1$ objective is used as a common reference before allowing β to vary for each problem. Generation quality is measured using FID (Heusel et al., 2017) for the image datasets and the direct generation error defined in Appendix A.15 for the synthetic Gaussian benchmark.

With $\beta = 1$, the VAE performs poorly on several datasets. For example, it obtains FID 326.32 ± 0.38 on Shapes3D and 468.67 ± 2.85 on CIFAR-10, compared with 47.91 ± 3.79 and 150.18 ± 3.28 for SOLA. On dSprites, the difference in mean FID is much smaller (98.03 ± 42.6 versus 92.3 ± 1.8), although the VAE is considerably more variable across seeds. These results motivate examining whether the poor performance of the common VAE objective can be recovered by adjusting β.

Figure 2 shows that tuning β is not simply a matter of selecting a different numerical value for each dataset. The shape of the sensitivity itself changes across problems. On the synthetic benchmark, increasing β is predominantly detrimental and the strongest result is obtained at $\beta = 0$. Frey Faces instead exhibits an intermediate optimum around $\beta = 10^{-3}$, with performance degrading for both smaller and larger values. dSprites shows a third behavior, where increasing β from the weakly regularized regime initially improves generation, reaching its best result around $\beta = 10^{-1}$, before deteriorating again. Thus, both the preferred value of β and the response to changing it are problem-dependent, making a common choice difficult.

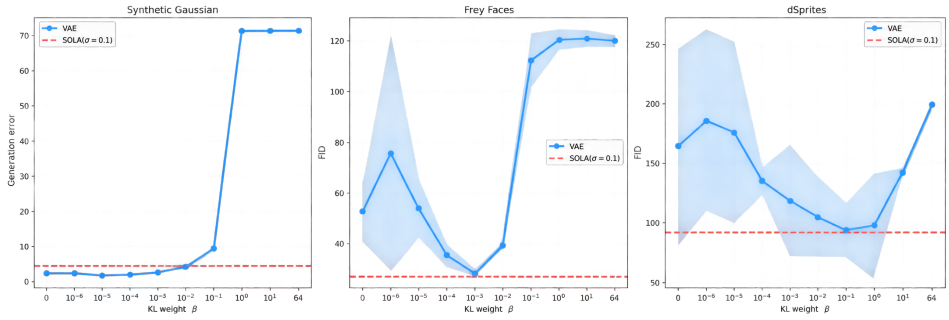


Figure 2: Sensitivity of VAE generation performance to the KL weight β on the synthetic Gaussian benchmark, Frey Faces, and dSprites. The dashed horizontal line shows SOLA performance with the fixed stochastic scale $\sigma = 0.1$.

Table 1 summarizes the common $\beta = 1$ reference and the best-performing problem-specific VAE setting $\beta^\star$. The selected values vary substantially, with $\beta^\star = 0$ on the synthetic benchmark, 10^{-3} on Shapes3D and Frey Faces, 10^{-4} on CelebA and CIFAR-10, and 10^{-1} on dSprites. SOLA, in contrast, uses the same fixed stochastic scale $\sigma = 0.1$ in all cases.

Allowing problem-specific tuning substantially improves the VAE. On the synthetic benchmark, $\beta = 0$ reduces the generation error to 1.96 ± 0.47, below SOLA at 4.27 ± 1.19. On Frey Faces, the tuned VAE reaches 28.79 ± 1.58 FID, close to SOLA at 27.42 ± 1.51, while on CelebA tuning reduces the VAE FID from 253.3 to 120.39 ± 6.58, compared with 99.88 ± 1.89 for SOLA. On CIFAR-10, the tuned VAE reaches 146.81 ± 4.60, slightly below SOLA at 150.18 ± 3.28.

Tuning does not necessarily remove run-to-run instability. This is particularly visible on Shapes3D, where the tuned VAE obtains 132.61 ± 161.38 FID compared with 47.91 ± 3.79 for SOLA, and on dSprites, where the tuned VAE remains substantially more variable than SOLA. Overall, standard VAEs can recover strong solutions when β is chosen appropriately, but both the useful regularization

Table 1: Comparison of SOLA with the standard VAE using the common $\beta = 1$ objective and the best-performing problem-specific KL weight $\beta^\star$. The generation metric is FID for the image datasets and Gaussian-fit relative error (%) for the synthetic benchmark. Values are mean ± sample standard deviation over the evaluated seeds.

Dataset	Metric	VAE ($\beta = 1$)	VAE ($\beta^\star$)	SOLA
Synthetic	Gen. error ↓	72.06 ± 2.1	$\mathbf{1.96 \pm 0.47}$ ($\beta^\star$=0)	4.27 ± 1.19
dSprites	Recon. MSE ↓	0.001 ± 0.00	$\mathbf{0.0005 \pm 0.00}$	0.002 ± 0.00
	FID ↓	98.03 ± 42.6	94.39 ± 22.3 ($\beta^\star$=0.1)	$\mathbf{92.3 \pm 1.8}$
Shapes3D	Recon. MSE ↓	0.122 ± 0.001	0.041 ± 0.07	$\mathbf{0.0002 \pm 0.00}$
	FID ↓	326.32 ± 0.38	132.61 ± 161.38 ($\beta^\star$=10^{-3})	$\mathbf{47.91 \pm 3.79}$
Frey Faces	Recon. MSE ↓	0.336 ± 0.0	0.026 ± 0.0	$\mathbf{0.02 \pm 0.0}$
	FID ↓	120.92 ± 4.46	28.79 ± 1.58 ($\beta^\star$=10^{-3})	$\mathbf{27.42 \pm 1.51}$
CelebA	Recon. MSE ↓	0.912 ± 0.001	0.103 ± 0.003	$\mathbf{0.092 \pm 0.004}$
	FID ↓	253.3 ± 1.05	120.39 ± 6.58 ($\beta^\star$=10^{-4})	$\mathbf{99.88 \pm 1.89}$
CIFAR-10	Recon. MSE ↓	0.985 ± 0.001	0.154 ± 0.002	$\mathbf{0.134 \pm 0.000}$
	FID ↓	468.67 ± 2.85	$\mathbf{146.81 \pm 4.60}$ ($\beta^\star$=10^{-4})	150.18 ± 3.28

regime and its sensitivity profile vary markedly between problems. SOLA does not introduce an additional loss-balancing coefficient and uses the same $\sigma = 0.1$ throughout. We do not claim that this value is universally optimal, but it remains competitive across datasets exhibiting substantially different VAE regularization behavior. A sensitivity analysis to the noise scale in SOLA is provided in Appendix A.12.

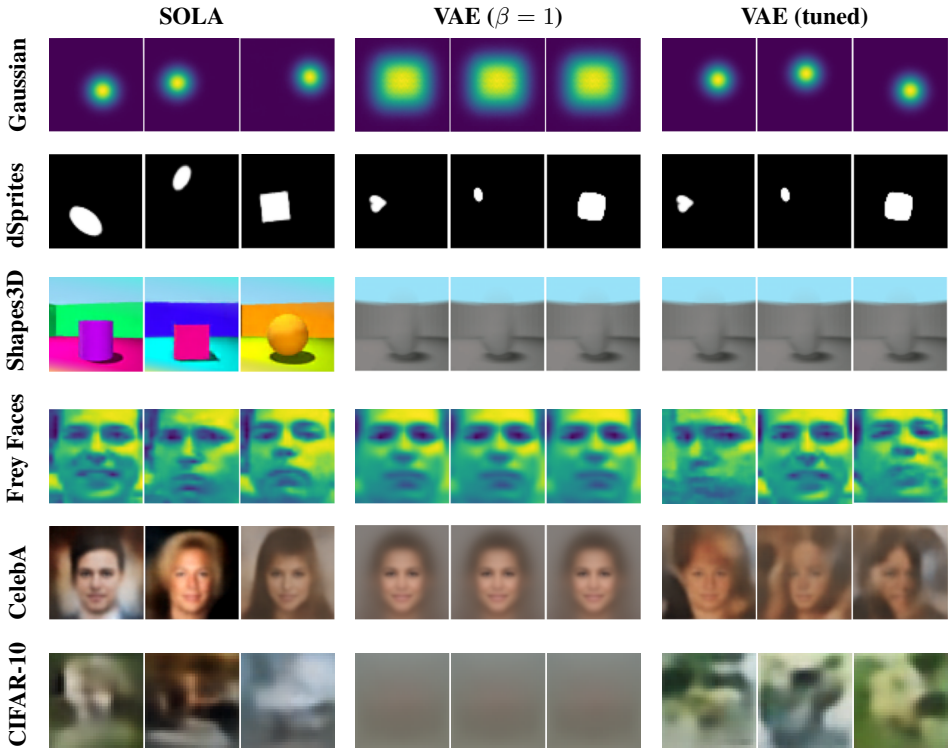


Figure 3: Representative random generations from SOLA, the common $\beta = 1$ VAE reference, and problem-specific tuned VAEs on synthetic Gaussian, dSprites, Shapes3D, Frey Faces, CelebA, and CIFAR-10. For dSprites, the tuned VAE uses $\beta^\star = 10^{-1}$. The corresponding quantitative comparison is reported in Table 1. The seed with FID closest to the mean has been chosen.

3.2 Comparison with Alternative Generative Autoencoders and VAE Variants

We next compare SOLA with several established modifications of the VAE objective, including monotonic KL annealing (Bowman et al., 2016), cyclical KL annealing (Fu et al., 2019), Free Bits (Kingma et al., 2016), σ-VAE (Rybkin et al., 2021), and capacity-controlled β-VAE (Burgess et al., 2018). We additionally compare against RAE (Ghosh et al., 2020), which uses post-training latent density estimation, WAE-MMD (Tolstikhin et al., 2018), which matches the aggregated latent distribution to a prescribed prior using maximum mean discrepancy, and VQ-VAE (Van den Oord et al., 2017), which uses a learned discrete prior for generation.

As shown in Table 2, their relative effectiveness changes substantially across datasets. On the synthetic Gaussian benchmark, σ-VAE, capacity-controlled β-VAE, and RAE obtain generation errors of 2.01 ± 0.70, 3.59 ± 0.92, and 2.9 ± 0.31, respectively, compared with 4.27 ± 1.19 for SOLA. On Frey Faces, SOLA obtains 27.42 ± 1.51 FID, compared with 32.26 ± 0.91 for RAE, 41.26 ± 2.62 for capacity-controlled β-VAE, and 54.46 ± 8.85 for σ-VAE. On CelebA, SOLA obtains 99.88 ± 1.89, compared with 103.75 ± 0.56 for RAE.

Table 2: Comparison with alternative variational training strategies and autoencoder-based generative models. Values are mean $\pm$ sample standard deviation over five seeds. Lower is better. Details of parameters for other variants can be found in Appendix A.17.

Method	Gaussian gen. error ↓	Frey Faces FID ↓	CelebA FID ↓
VAE variants			
Monotonic KL annealing	71.44 ± 0.66	122.68 ± 4.04	230.6 ± 3.1
Cyclical KL annealing	71.39 ± 0.84	120.92 ± 2.83	231.2 ± 3.5
Free Bits	6.44 ± 1.13	106.70 ± 21.03	186.5 ± 19.6
σ-VAE	$\mathbf{2.01 \pm 0.70}$	54.46 ± 8.85	114.22 ± 4.71
Capacity-controlled β-VAE	3.59 ± 0.92	41.26 ± 2.62	180.79 ± 77.75
Alternative generative autoencoders			
RAE	2.9 ± 0.31	32.26 ± 0.91	103.75 ± 0.56
WAE (MMD)	2.79 ± 0.24	66.79 ± 47.72	120.82 ± 3.43
VQ-VAE	6.81 ± 1.26	61.50 ± 4.02	129.44 ± 9.42
Proposed method			
SOLA	4.27 ± 1.19	$\mathbf{27.42 \pm 1.51}$	$\mathbf{99.88 \pm 1.89}$

No alternative considered here provides uniformly strong performance across the three datasets. In particular, methods that perform very well on the simple synthetic manifold do not necessarily retain the same advantage on Frey Faces or CelebA. RAE also shows that strong generation can be obtained through post-hoc latent density estimation, whereas SOLA remains competitive using its population-derived native sampling rule.

3.3 Component Analysis

Finally, we examine whether SOLA's behavior can be explained simply by using a smaller bottleneck, an SVD representation, or stochastic mixing in isolation. The complete ablation protocol and the corresponding synthetic experiment are reported in Appendix A.8. Here we highlight CelebA, where the representation must capture substantially more complex image variability.

We compare a wide deterministic autoencoder, wide and directly compact G-only stochastic bottlenecks, an RRAE using the SVD bottleneck without stochastic mixing, and the complete SOLA model. For variants without a native sampling geometry, generation is evaluated using the standard-normal sampling probe described in Appendix A.8. RRAE and SOLA instead use their native SVD-derived generators.

The reconstruction errors of all five models are similar, yet their direct-generation performance differs substantially. In particular, the directly compact G-only model achieves the lowest reconstruction error (0.0903 ± 0.0047) but a higher FID than SOLA (113.67 ± 6.03 versus 99.84 ± 4.12). The improvement of SOLA therefore cannot be explained by reconstruction accuracy alone.

Table 3: Component analysis on CelebA. Values are mean ± sample standard deviation over three seeds. AE and G-only variants use the standard-normal sampling probe, whereas RRAE and SOLA use their native SVD-derived generators. Lower is better.

Method	Test reconstruction MSE ↓	FID ↓
No G, No SVD (AE)	0.0923 ± 0.0028	200.96 ± 18.53
G-only (wide)	0.0950 ± 0.0019	127.63 ± 9.96
G-only (small)	$\mathbf{0.0903 \pm 0.0047}$	113.67 ± 6.03
SVD-only (RRAE)	0.0950 ± 0.0021	124.80 ± 11.83
SOLA	0.0924 ± 0.0035	$\mathbf{99.84 \pm 4.12}$

The wide G-only representation improves substantially over the deterministic wide autoencoder, showing that stochastic mixing itself can be beneficial under direct sampling. Reducing the G-only bottleneck further improves FID, confirming that latent dimensionality is also important. However, neither the compact G-only model nor the SVD-only RRAE reaches the direct-generation quality of the complete method.

This comparison measures native or prescribed sampleability rather than the intrinsic generative quality of the representations alone. A complementary control in Appendix A.8 fits a full-covariance Gaussian to the CelebA latent representations. Under this additional density-estimation step, the AE and G-only variants reach FIDs close to SOLA, whereas fitting the same type of density to SOLA gives 100.63 ± 2.88, close to its native 99.84 ± 4.12. Thus, alternative representations can also support strong generation after post-hoc density fitting, while SOLA obtains a comparable sampling geometry directly from its construction.

The synthetic component study in Appendix A.8 provides a complementary result. On the simple two-dimensional Gaussian manifold, the directly compact G-only model slightly outperforms SOLA, indicating that a compact stochastic bottleneck is sufficient when the latent geometry is simple. In contrast, on CelebA, combining stochastic mixing with the population-derived SVD geometry provides the strongest direct-generation performance among the considered variants. These results suggest that the explicit SVD geometry becomes increasingly important when native sampling must capture a more complex latent structure.

4 Conclusion

We introduced SOLA, the Stochastic One-Loss Autoencoder, which derives latent variability and sampling geometry from the encoded population rather than from sample-specific variance predictions and an explicitly weighted KL objective. By combining a rank-reduced orthogonal representation with stochastic latent mixing, SOLA induces a shared geometry-dependent covariance and yields an exactly constant batch-averaged geometry-matched KL divergence for fixed bottleneck dimension and noise scale. The KL term therefore contributes no optimization gradient, enabling reconstruction-only training while retaining a stochastic latent representation. After training, the aggregated population geometry provides the basis for a Gaussian sampling rule whose approximation is analyzed separately.

Across six datasets, SOLA achieves accurate reconstruction and competitive generation using the same fixed noise scale $\sigma = 0.1$, whereas both the preferred VAE KL weight β and the response to changing it vary substantially across problems. Comparisons with modified VAE objectives and alternative generative autoencoders further show that SOLA remains competitive without an optimized prior-matching objective or a separately fitted latent density. Controlled ablations show that a compact stochastic bottleneck can suffice when the latent geometry is simple, while on CelebA the combination of stochastic mixing and population-derived SVD geometry provides the strongest direct-generation performance among the considered variants. Together, these results suggest that population-derived latent geometry and structured stochastic mixing provide a viable alternative to explicitly balancing reconstruction and prior-matching losses in generative autoencoders.

AI USE STATEMENT

In this work, we used generative AI tools to assist with several stages of the research process. These tools were used to help write and debug code for synthetic dataset generation, develop and refine theoretical models and conceptual frameworks, formulate and check mathematical claims, suggest intermediate derivations and critical ingredients for mathematical proofs, and assist in writing and improving the presentation of proofs. Generative AI was also used to propose or refine research hypotheses, provide feedback on research methodology and experimental design, assist with the implementation and debugging of methods and experiments, clean and reformat datasets for use in the training pipeline, and support discussions concerning the interpretation of quantitative experimental results.

Additionally, generative AI tools were used during manuscript preparation to improve clarity, consistency, readability, and organization, and to assist with drafting, correcting, and redrafting sections of the paper. These tools also supported literature discovery by suggesting potentially relevant publications and helping identify related work. All AI-assisted manuscript content was reviewed and revised by the authors, and suggested references were independently verified against the original publications.

We did not use generative AI tools for translation or for qualitative or thematic data analysis. The core research questions, methodological decisions, and final theoretical and empirical conclusions were determined by the authors. AI-generated suggestions concerning theory, proofs, experiments, or interpretation were treated as suggestions rather than accepted as authoritative results.

All AI-assisted work was independently reviewed and validated by the authors. Mathematical claims, derivations, and proofs were checked for correctness and consistency with the stated assumptions. AI-assisted code was inspected, tested, and validated numerically where applicable, and synthetic-data generation and dataset-processing procedures were checked against the intended experimental protocols. Experimental suggestions and interpretations were independently assessed against the resulting measurements. Finally, AI-assisted manuscript content and literature suggestions were manually reviewed to ensure that scientific claims were appropriately supported, references were accurate and relevant, and no unattributed material was introduced.

We take full responsibility for the final content of this work, including all text, mathematical claims, implementations, datasets, experimental results, and other artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We have taken several steps to facilitate independent reproduction of the results reported in this work. The manuscript and supplementary appendices specify the SOLA construction, training and sampling procedures, computational considerations, dataset preparation, model architectures, optimization settings, evaluation protocols, and the ablation and sensitivity studies used throughout the paper. To further support reproducibility, the implementation used for the reported experiments, together with the scripts required to reproduce the main quantitative evaluations and analyses, is available in the supplementary material.

REFERENCES

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21. Association for Computational Linguistics, 2016. doi: 10.18653/v1/K16-1002. URL https://aclanthology.org/K16-1002/.

Christopher P. Burgess, Irina Higgins, Arka Pal, Loïc Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018. URL https://arxiv.org/abs/1804.03599.

Ricky T. Q. Chen, Xuechen Li, Roger B. Grosse, and David K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*,

volume 31, pp. 2610–2620, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/1ee3dfcd8a0645a25a35977997223d22-Abstract.html.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 240–250. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1021.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael J. Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g7tpEYDS.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rylDfnCqF7.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658. PMLR, 2018. URL https://proceedings.mlr.press/v80/kim18b.html.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4743–4751, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Abstract.html.

James Lucas, George Tucker, Roger B. Grosse, and Mohammad Norouzi. Understanding posterior collapse in generative latent variable models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2019. URL https://openreview.net/forum?id=r1xaVLUYuE.

Jad Mounayer, Sebastian Rodriguez, Chady Ghnatios, Charbel Farhat, and Francisco Chinesta. Rank reduction autoencoders. *arXiv preprint arXiv:2405.13980*, 2024. doi: 10.48550/arXiv.2405.13980. URL https://arxiv.org/abs/2405.13980.

Jad Mounayer, Alicia Tierz, Jérôme Tomezyk, Chady Ghnatios, and Francisco Chinesta. Variational rank reduction autoencoders. *Neurocomputing*, 700:134422, 2026. doi: 10.1016/j.neucom.2026.134422. URL https://doi.org/10.1016/j.neucom.2026.134422.

Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and effective VAE training with calibrated decoders. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9179–9189. PMLR, 2021. URL https://proceedings.mlr.press/v139/rybkin21a.html.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HkL7n1-0b.

Jakub M. Tomczak and Max Welling. VAE with a VampPrior. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pp. 1214–1223. PMLR, 2018. URL https://proceedings.mlr.press/v84/tomczak18a.html.

Aaron Van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5885–5892, 2019. doi: 10.1609/aaai.v33i01.33015885.

A Appendix

A.1 Common-Basis Representation and Convergence of the Latent Geometry

The batch-wise truncated SVD yields orthonormal bases that may differ between batches even when they represent the same latent subspace. We first recall the common-subspace convergence property established for RRAEs (Mounayer et al., 2024). Let $U^{\mathcal{B}_i} \in \mathbb{R}^{L\times K}$ denote the retained left-singular basis associated with batch $\mathcal{B}_i$. Under the RRAE convergence assumptions, the batch-wise projection operators converge toward a common rank-K projector,

$$U^{\mathcal{B}_i} U^{\mathcal{B}_i\top} \longrightarrow U_\star U_\star^\top. \tag{6}$$

Consequently, the individual orthonormal bases become equivalent up to orthogonal transformations within the retained subspace. In the limiting representation, for each batch there exists $Q_i \in \mathbb{R}^{K\times K}$, with $Q_i^\top Q_i = I_K$, such that

$$U^{\mathcal{B}_i} = U_\star Q_i. \tag{7}$$

Thus, although the SVD basis itself is not unique, the converged batch-wise representations span the same latent subspace.

In the original RRAE construction, the fixed basis is obtained by aggregating the batch-wise bases. For SOLA, we slightly modify this procedure in order to recover not only a common basis but also the corresponding global latent geometry. For each of the last N_b training batches, with truncated decomposition

$$Y_K^{\mathcal{B}_i} = U^{\mathcal{B}_i}\Sigma^{\mathcal{B}_i} V^{\mathcal{B}_i\top}, \tag{8}$$

we retain the singular-value-weighted basis $U^{\mathcal{B}_i}\Sigma^{\mathcal{B}_i}$, and form

$$W_f = \left[U^{\mathcal{B}_1}\Sigma^{\mathcal{B}_1}, \ldots, U^{\mathcal{B}_{N_b}}\Sigma^{\mathcal{B}_{N_b}}\right]. \tag{9}$$

We then compute

$$W_f = \bar{U}\,\bar{\Sigma}\,\bar{V}^\top, \tag{10}$$

and define

$$U_f = \bar{U}_{[:,1:K]}, \qquad \Sigma_f = \bar{\Sigma}_{[1:K,1:K]}. \tag{11}$$

This modification preserves the common-subspace property of RRAEs. Indeed, using Equation 7,

$$W_f = U_\star \left[Q_1\Sigma^{\mathcal{B}_1}, \ldots, Q_{N_b}\Sigma^{\mathcal{B}_{N_b}}\right] \tag{12}$$

in the limiting regime. Hence, provided the aggregated matrix has rank K, the column space of W_f is precisely the common rank-K subspace. The leading left singular vectors of W_f therefore recover a representative orthonormal basis of this common subspace. For finite training, the truncated SVD provides the best rank-K approximation of the aggregated matrix W_f.

The advantage of the weighted construction is that its singular values also summarize the geometry carried by the retained batches. Since W_f is formed by horizontal concatenation,

$$\begin{aligned} W_f W_f^\top &= \sum_{i=1}^{N_b} U^{\mathcal{B}_i}\left(\Sigma^{\mathcal{B}_i}\right)^2 U^{\mathcal{B}_i\top} \\ &= \sum_{i=1}^{N_b} Y_K^{\mathcal{B}_i} Y_K^{\mathcal{B}_i\top}, \end{aligned} \tag{13}$$

where the second equality follows from $V^{\mathcal{B}_i\top}V^{\mathcal{B}_i} = I_K$.

For a batch $\mathcal{B}_i$ of size B_i, define its rank-K geometry in the ambient latent space by

$$G_{\mathcal{B}_i} = \frac{1}{B_i} Y_K^{\mathcal{B}_i} Y_K^{\mathcal{B}_i\top} = \frac{1}{B_i} U^{\mathcal{B}_i}\left(\Sigma^{\mathcal{B}_i}\right)^2 U^{\mathcal{B}_i\top}. \tag{14}$$

Let

$$N_{\text{rep}} = \sum_{i=1}^{N_b} B_i \tag{15}$$

denote the total number of encoded samples represented by the retained batches. Equation 13 then gives the exact finite-sample identity

$$\frac{1}{N_{\text{rep}}} W_f W_f^\top = \sum_{i=1}^{N_b} \frac{B_i}{N_{\text{rep}}} G_{\mathcal{B}_i}. \tag{16}$$

Thus, the matrix $W_f W_f^\top / N_{\text{rep}}$ is the sample-size-weighted aggregate of the retained batch geometries. In the usual case of equal batch sizes $B_i = B$,

$$\frac{1}{N_{\text{rep}}} W_f W_f^\top = \frac{1}{N_b} \sum_{i=1}^{N_b} G_{\mathcal{B}_i}. \tag{17}$$

After convergence of the encoder, assume that the retained batches are representative independent draws from the same encoded population. For fixed batch size B, the matrices $G_{\mathcal{B}_i}$ are then identically distributed random matrices. Assuming

$$\mathbb{E}[\|G_{\mathcal{B}}\|] < \infty, \tag{18}$$

the law of large numbers applied entry-wise yields

$$\frac{1}{N_b} \sum_{i=1}^{N_b} G_{\mathcal{B}_i} \longrightarrow G_f := \mathbb{E}[G_{\mathcal{B}}], \qquad N_b \to \infty. \tag{19}$$

Hence, individual finite-batch geometries need not become identical: their fluctuations are instead averaged out by the aggregation over successive converged batches.

Finally, from the SVD of W_f, and in the limiting common rank-K subspace,

$$W_f W_f^\top = U_f \Sigma_f^2 U_f^\top. \tag{20}$$

Consequently, the empirical global geometry represented in the fixed coordinates U_f is

$$\widehat{D}_f = U_f^\top \left(\frac{1}{N_{\text{rep}}} W_f W_f^\top \right) U_f = \frac{1}{N_{\text{rep}}} \Sigma_f^2. \tag{21}$$

If $\lambda_1(G_f), \ldots, \lambda_K(G_f)$ denote the retained eigenvalues of the limiting geometry, ordered consistently with the SVD, define

$$D_f = \operatorname{diag}\left(\lambda_1(G_f), \ldots, \lambda_K(G_f)\right).$$

Together with Equation 19, this yields

$$\widehat{D}_f \longrightarrow D_f, \tag{22}$$

while simultaneously providing the fixed basis U_f and the representative singular-value matrix Σ_f used after training.

Therefore, SOLA does not require the singular values of individual finite batches to converge to identical values. Instead, the singular-value-weighted aggregation $W_f = [U^{\mathcal{B}_i} \Sigma^{\mathcal{B}_i}]_{i=1}^{N_b}$ provides a finite-sample estimator of the common population geometry, whose left singular vectors define U_f and whose singular values define Σ_f.

A.2 Derivation of the Structured Stochastic Bottleneck

We provide here the complete derivation of the stochastic bottleneck introduced in Section 2.2. For a training batch, let

$$A_K = \Sigma_K V_K^\top = [a_1, \ldots, a_B] \in \mathbb{R}^{K \times B} \tag{23}$$

denote the structured RRAE coordinates. SOLA introduces stochasticity through the random mixing matrix

$$G = I_B + \frac{\sigma}{\sqrt{B}} E, \qquad E_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{24}$$

and defines

$$\widetilde{A}_K = A_K G. \tag{25}$$

Expanding the product gives

$$\widetilde{A}_K = A_K + \frac{\sigma}{\sqrt{B}} A_K E. \tag{26}$$

Let $e_i \in \mathbb{R}^B$ denote the i-th column of E. The stochastic coordinate associated with sample i is therefore

$$\widetilde{a}_i = a_i + \frac{\sigma}{\sqrt{B}} A_K e_i. \tag{27}$$

Conditionally on the encoded batch A_K,

$$e_i \sim \mathcal{N}(0, I_B),$$

and hence

$$\mathbb{E}\left[\widetilde{a}_i \mid A_K\right] = a_i. \tag{28}$$

Thus, the identity component of G preserves the deterministic coordinate in expectation.

Since a linear transformation of a Gaussian vector remains Gaussian,

$$A_K e_i \mid A_K \sim \mathcal{N}\left(0, A_K A_K^\top\right). \tag{29}$$

Using the orthogonal structure

$$A_K A_K^\top = \Sigma_K^2, \tag{30}$$

we obtain

$$\mathrm{Cov}\left(\widetilde{a}_i \mid A_K\right) = \frac{\sigma^2}{B} \Sigma_K^2. \tag{31}$$

Consequently, the conditional stochastic representation is exactly

$$\boxed{\widetilde{a}_i \mid A_K \sim \mathcal{N}\left(a_i, \frac{\sigma^2}{B} \Sigma_K^2\right)}. \tag{32}$$

The perturbation covariance is therefore common to all samples in the batch and is determined entirely by the retained singular spectrum.

Equivalent reduced-noise formulation. The conceptual construction in Equation 24 involves a $B \times B$ Gaussian matrix. This matrix need not be instantiated in practice. Substituting $A_K = \Sigma_K V_K^\top$ into Equation 26 gives

$$\widetilde{A}_K = A_K + \frac{\sigma}{\sqrt{B}} \Sigma_K V_K^\top E. \tag{33}$$

Consider one column e_i of E. Since

$$e_i \sim \mathcal{N}(0, I_B)$$

and $V_K^\top V_K = I_K$,

$$V_K^\top e_i \mid V_K \sim \mathcal{N}\left(0, V_K^\top V_K\right) = \mathcal{N}(0, I_K). \tag{34}$$

Moreover, the columns of E are mutually independent, so the transformed columns $V_K^\top e_i$ remain independent. Therefore, conditionally on V_K,

$$V_K^\top E \overset{d}{=} \Xi, \qquad \Xi \in \mathbb{R}^{K \times B}, \qquad \Xi_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1). \tag{35}$$

Equation 33 can consequently be written as

$$\boxed{\widetilde{A}_K \overset{d}{=} A_K + \frac{\sigma}{\sqrt{B}} \Sigma_K \Xi}. \tag{36}$$

Thus, the original random mixing of the B encoded samples is distributionally equivalent to drawing only $K \times B$ independent standard Gaussian variables and scaling them along the singular directions by Σ_K. This is the formulation used in the implementation.

Relation to interpolation. For the i-th stochastic sample, Equation 27 may also be written as

$$\widetilde{a}_i = A_K g_i, \qquad g_i = \delta_i + \frac{\sigma}{\sqrt{B}} e_i, \tag{37}$$

where δ_i denotes the i-th canonical basis vector of $\mathbb{R}^B$. Hence,

$$\widetilde{a}_i = \sum_{j=1}^{B} (g_i)_j a_j. \tag{38}$$

The stochastic representation is therefore a linear combination of coordinates already present in the encoded batch. In contrast to convex interpolation, SOLA does not constrain the coefficients to be nonnegative or to sum to one. If the coefficients were additionally constrained to sum to one, the construction would become an affine combination, and imposing both unit sum and nonnegativity would yield a convex combination. SOLA intentionally retains the more general linear form, allowing stochastic perturbations in directions spanned by the encoded population without restricting them to its convex hull. A comparison of different types of linear mixing is provided in Appendix A.9.

A.3 Proof of the Constant Geometry-Matched KL Divergence

We provide the complete derivation of the constant batch-averaged KL divergence introduced in Section 2.3. For a training batch $\mathcal{B}$, recall the geometry matrix

$$D_{\mathcal{B}} = \frac{1}{B}\Sigma_K^2 = \frac{1}{B} A_K A_K^\top, \tag{39}$$

and the sample-wise stochastic distribution

$$q_i^{\mathcal{B}} = \mathcal{N}\left(a_i, \sigma^2 D_{\mathcal{B}}\right), \tag{40}$$

where a_i denotes the i-th column of A_K. The corresponding geometry-matched Gaussian reference is

$$p_{\mathcal{B}} = \mathcal{N}\left(0, D_{\mathcal{B}}\right). \tag{41}$$

Because $K \leq \operatorname{rank}(Y)$, all retained singular values are strictly positive. Hence $D_{\mathcal{B}}$ is positive definite and invertible.

For two K-dimensional Gaussian distributions

$$q = \mathcal{N}(\mu_q, \Sigma_q), \qquad p = \mathcal{N}(\mu_p, \Sigma_p),$$

their KL divergence is

$$D_{\mathrm{KL}}(q\|p) = \frac{1}{2}\left[\operatorname{tr}\left(\Sigma_p^{-1}\Sigma_q\right) + (\mu_p - \mu_q)^\top \Sigma_p^{-1}(\mu_p - \mu_q) - K + \log\frac{\det\Sigma_p}{\det\Sigma_q}\right]. \tag{42}$$

For the SOLA distributions,

$$\mu_q = a_i, \qquad \mu_p = 0, \qquad \Sigma_q = \sigma^2 D_{\mathcal{B}}, \qquad \Sigma_p = D_{\mathcal{B}}.$$

The trace term becomes

$$\operatorname{tr}\left(D_{\mathcal{B}}^{-1}\sigma^2 D_{\mathcal{B}}\right) = \sigma^2 K, \tag{43}$$

while the mean contribution is

$$a_i^\top D_{\mathcal{B}}^{-1} a_i. \tag{44}$$

Finally,

$$\begin{aligned}\log\frac{\det D_{\mathcal{B}}}{\det(\sigma^2 D_{\mathcal{B}})} &= \log\frac{\det D_{\mathcal{B}}}{\sigma^{2K}\det D_{\mathcal{B}}} \\ &= -K\log(\sigma^2).\end{aligned} \tag{45}$$

Substituting these terms into Equation 42 gives

$$D_{\mathrm{KL}}\left(q_i^{\mathcal{B}} \,\|\, p_{\mathcal{B}}\right) = \frac{1}{2}\left[a_i^\top D_{\mathcal{B}}^{-1} a_i + K\left(\sigma^2 - 1 - \log(\sigma^2)\right)\right]. \tag{46}$$

The individual divergence therefore depends on the latent coordinate a_i. However, its batch average is fixed by the orthogonal structure of A_K. Indeed,

$$\begin{aligned}\frac{1}{B}\sum_{i=1}^{B} a_i^\top D_{\mathcal{B}}^{-1} a_i &= \frac{1}{B}\operatorname{tr}\left(D_{\mathcal{B}}^{-1}\sum_{i=1}^{B} a_i a_i^\top\right)\\ &= \operatorname{tr}\left(D_{\mathcal{B}}^{-1}\frac{1}{B}A_K A_K^\top\right)\\ &= \operatorname{tr}\left(D_{\mathcal{B}}^{-1} D_{\mathcal{B}}\right)\\ &= K. \end{aligned} \tag{47}$$

Averaging Equation 46 over the batch therefore yields

$$\begin{aligned}\frac{1}{B}\sum_{i=1}^{B} D_{\mathrm{KL}}\left(q_i^{\mathcal{B}} \,\|\, p_{\mathcal{B}}\right) &= \frac{1}{2}\left[K + K\left(\sigma^2 - 1 - \log(\sigma^2)\right)\right]\\ &= \boxed{\frac{K}{2}\left(\sigma^2 - \log(\sigma^2)\right)}. \end{aligned} \tag{48}$$

This identity is exact for every batch satisfying the rank condition. In particular, it does not require the latent coordinates to be centered, Gaussian, or independently distributed. For fixed bottleneck dimension K and perturbation amplitude σ, Equation 48 contains neither the network parameters nor the singular values of the encoded batch. Therefore,

$$\nabla_{\theta,\phi}\left[\frac{1}{B}\sum_{i=1}^{B} D_{\mathrm{KL}}\left(q_i^{\mathcal{B}} \,\|\, p_{\mathcal{B}}\right)\right] = 0. \tag{49}$$

Consequently, adding the geometry-matched KL divergence to the SOLA training objective changes it only by an additive constant, and the network can be optimized using the reconstruction loss alone.

Generality of the constant-KL identity. The constancy of the batch-averaged KL divergence is not specific to the SVD. More generally, for $\sigma > 0$, let $A \in \mathbb{R}^{K\times B}$ be any full-row-rank coordinate matrix and define

$$D = \frac{1}{B}AA^\top. \tag{50}$$

Using the same random mixing construction

$$\widetilde{A} = A\left(I_B + \frac{\sigma}{\sqrt{B}}E\right) \tag{51}$$

induces

$$\widetilde{a}_i \mid A \sim \mathcal{N}(a_i, \sigma^2 D), \tag{52}$$

and the same trace argument yields

$$\frac{1}{B}\sum_{i=1}^{B} D_{\mathrm{KL}}\left(\mathcal{N}(a_i, \sigma^2 D) \,\|\, \mathcal{N}(0, D)\right) = \frac{K}{2}\left(\sigma^2 - \log\sigma^2\right). \tag{53}$$

Thus, random mixing together with a geometry-matched reference is sufficient for the constant-KL property. In SOLA, the SVD plays a distinct role: it provides an explicit rank-K bottleneck with orthogonal population geometry and supplies the structured global geometry used by the post-training sampler.

Finally, although the geometry-matched reference $p_{\mathcal{B}} = \mathcal{N}(0, D_{\mathcal{B}})$ is batch-dependent during training, this dependence does not persist in the final representation. As shown in Appendix A.1, the geometries of successive converged batches admit a common-basis representation whose aggregate converges toward a fixed global geometry D_f. This induces the single fixed clean geometry-matched reference

$$p_f = \mathcal{N}(0, D_f). \tag{54}$$

Thus, the batch-dependent references used to characterize the stochastic training representation are consolidated after training into a single clean global latent reference. The additional stochastic inflation used for generation is derived separately in Appendix A.6.

A.4 Gaussian Approximation of the Right-Singular Coordinates

The generation mechanism of SOLA does not require a Gaussian model for the ambient encoded representation or for its coordinates in the fixed global basis. The relevant quantities are instead the normalized right-singular coordinates appearing directly in the rank-K bottleneck.

Consider a randomly drawn converged batch $\mathcal{B}$ of size B, and let

$$Y_K^{\mathcal{B}} = U_K^{\mathcal{B}} \Sigma_K^{\mathcal{B}} V_K^{\mathcal{B}\top} \tag{55}$$

denote its rank-K truncated singular value decomposition, with

$$V_K^{\mathcal{B}\top} = \begin{bmatrix} v_1^{\mathcal{B}} & \cdots & v_B^{\mathcal{B}} \end{bmatrix}, \qquad v_i^{\mathcal{B}} \in \mathbb{R}^K. \tag{56}$$

The clean bottleneck coordinate associated with sample i is

$$a_i^{\mathcal{B}} = \Sigma_K^{\mathcal{B}} v_i^{\mathcal{B}}. \tag{57}$$

We introduce the normalized right-singular coordinates

$$w_i^{\mathcal{B}} := \sqrt{B}\, v_i^{\mathcal{B}}, \qquad W_K^{\mathcal{B}} := \sqrt{B}\, V_K^{\mathcal{B}\top} = \begin{bmatrix} w_1^{\mathcal{B}} & \cdots & w_B^{\mathcal{B}} \end{bmatrix}. \tag{58}$$

Exact second-moment structure. For every batch, the orthogonality relation

$$V_K^{\mathcal{B}\top} V_K^{\mathcal{B}} = I_K \tag{59}$$

immediately gives

$$\boxed{\frac{1}{B} \sum_{i=1}^{B} w_i^{\mathcal{B}} w_i^{\mathcal{B}\top} = I_K.} \tag{60}$$

Thus, independently of the distribution of the encoded samples, the normalized right-singular coordinates have exactly unit uncentered second moment in every finite batch.

For a fixed batch, define

$$\bar{w}_{\mathcal{B}} = \frac{1}{B} \sum_{i=1}^{B} w_i^{\mathcal{B}}. \tag{61}$$

The corresponding centered empirical covariance is

$$\frac{1}{B} \sum_{i=1}^{B} \left(w_i^{\mathcal{B}} - \bar{w}_{\mathcal{B}}\right) \left(w_i^{\mathcal{B}} - \bar{w}_{\mathcal{B}}\right)^\top = I_K - \bar{w}_{\mathcal{B}} \bar{w}_{\mathcal{B}}^\top. \tag{62}$$

Importantly, the mean $\bar{w}_{\mathcal{B}}$ need not be negligible for every individual finite batch. SOLA constructs a single global generator from an ensemble of converged batches, and the relevant distribution is therefore the distribution obtained by jointly sampling a batch and a sample within that batch.

To formalize this, let $\mathcal{B}$ denote a random converged batch and, conditionally on $\mathcal{B}$, let

$$I \sim \mathrm{Unif}\{1, \ldots, B\}. \tag{63}$$

We define the random normalized right-singular coordinate

$$W := w_I^{\mathcal{B}} = \sqrt{B}\, v_I^{\mathcal{B}}. \tag{64}$$

For any fixed batch, Equation 60 gives

$$\mathbb{E}_I \left[W W^\top \mid \mathcal{B} \right] = I_K. \tag{65}$$

Averaging over the random batches therefore yields the exact ensemble identity

$$\boxed{\mathbb{E}_{\mathcal{B},I} \left[W W^\top \right] = I_K.} \tag{66}$$

This identity holds regardless of the variation of the individual batch means.

Assumptions. Only two non-structural properties of the ensemble of normalized right-singular coordinates are required.

(V1) Small dimension-normalized pooled mean.

We define the ensemble mean

$$\mu_W := \mathbb{E}_{\mathcal{B},I}[W] = \mathbb{E}_{\mathcal{B}}\left[\bar{w}_{\mathcal{B}}\right]. \tag{67}$$

We assume that the pooled mean is small relative to the natural $\sqrt{K}$ scale of the normalized K-dimensional coordinates:

$$\boxed{\frac{\|\mu_W\|_2}{\sqrt{K}} \ll 1.} \tag{68}$$

Equivalently,

$$\frac{\|\mu_W\|_2^2}{K} \ll 1. \tag{69}$$

This normalized quantity has a direct second-moment interpretation. Since

$$\mathrm{tr}\left(\mu_W \mu_W^\top\right) = \|\mu_W\|_2^2 \tag{70}$$

and

$$\mathrm{tr}(I_K) = K, \tag{71}$$

assumption (V1) is equivalently

$$\boxed{\frac{\mathrm{tr}\left(\mu_W \mu_W^\top\right)}{\mathrm{tr}(I_K)} = \frac{\|\mu_W\|_2^2}{K} \ll 1.} \tag{72}$$

Thus, the contribution of the pooled mean is small relative to the total K-dimensional second-moment energy.

The same condition admits a useful directional interpretation. Let $t \in \mathbb{R}^K$ be uniformly distributed on the unit sphere. Then

$$\mathbb{E}_t\left[tt^\top\right] = \frac{1}{K} I_K, \tag{73}$$

and consequently

$$\begin{aligned}\mathbb{E}_t\left[(t^\top \mu_W)^2\right] &= \mu_W^\top \mathbb{E}_t\left[tt^\top\right] \mu_W \\ &= \frac{\|\mu_W\|_2^2}{K}.\end{aligned} \tag{74}$$

Therefore,

$$\sqrt{\mathbb{E}_t\left[(t^\top \mu_W)^2\right]} = \frac{\|\mu_W\|_2}{\sqrt{K}} \ll 1. \tag{75}$$

Assumption (V1) therefore states that the RMS pooled mean shift along a typical latent direction is small relative to the unit coordinate scale. It does not require every individual retained direction to have a negligible mean.

Combining the exact second-moment identity Equation 66 with the definition of μ_W gives

$$\begin{aligned}\mathrm{Cov}_{\mathcal{B},I}(W) &= \mathbb{E}_{\mathcal{B},I}\left[WW^\top\right] - \mu_W \mu_W^\top \\ &= I_K - \mu_W \mu_W^\top.\end{aligned} \tag{76}$$

Under (V1), the correction term is small relative to the total second-moment energy, since

$$\frac{\mathrm{tr}\left(\mu_W \mu_W^\top\right)}{\mathrm{tr}(I_K)} = \frac{\|\mu_W\|_2^2}{K} \ll 1. \tag{77}$$

Likewise, for a random unit direction t,

$$t^\top \operatorname{Cov}(W)t = 1 - (t^\top \mu_W)^2, \tag{78}$$

and hence

$$\mathbb{E}_t\left[1 - t^\top \operatorname{Cov}(W)t\right] = \frac{\|\mu_W\|_2^2}{K} \ll 1. \tag{79}$$

Thus, once (V1) is established, the covariance is approximately unit in the normalized average-direction sense relevant to the K-dimensional sampling geometry.

Assumption (V1) is directly testable by pooling the coordinates $\sqrt{B}\, V_K^{\mathcal{B}\top}$ over independently sampled batches and estimating

$$\frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}}. \tag{80}$$

This behavior is assessed empirically in Appendix A.5.

(V2) Approximate Gaussian shape of the ensemble coordinates.

We assume that the ensemble distribution of

$$W = \sqrt{B}\, v_I^{\mathcal{B}} \tag{81}$$

is adequately represented, for the purpose of post-training sampling, by a multivariate Gaussian distribution having the same first two moments,

$$\boxed{W \overset{d}{\approx} \mathcal{N}\left(\mu_W, I_K - \mu_W \mu_W^\top\right).} \tag{82}$$

Equation 82 is an approximate distributional model rather than an exact consequence of the SVD. In particular, it does not require every individual singular direction or every one-dimensional projection to follow an exact Gaussian law. Moderate direction-dependent deviations are compatible with (V2), provided the ensemble distribution retains the approximately Gaussian geometry required for post-training sampling.

While this Gaussian approximation is not an exact consequence of the SVD, there are two complementary reasons to expect it in the present setting. First, each normalized right-singular coefficient is a normalized projection of a high-dimensional encoded sample. Indeed, for the k-th retained singular direction,

$$Y^{\mathcal{B}\top} u_k^{\mathcal{B}} = \sigma_k^{\mathcal{B}} v_k^{\mathcal{B}}, \tag{83}$$

so that

$$\sqrt{B}\, v_{ki}^{\mathcal{B}} = \frac{u_k^{\mathcal{B}\top} y_i}{\sigma_k^{\mathcal{B}}/\sqrt{B}} = \frac{\sum_{\ell=1}^{L} u_{k\ell}^{\mathcal{B}} y_{\ell i}}{\sigma_k^{\mathcal{B}}/\sqrt{B}}. \tag{84}$$

When the retained left-singular directions are delocalized, so that no small number of encoded components dominates the projection, $u_k^{\mathcal{B}\top} y_i$ is a weighted sum of many contributions. Under the usual finite-moment and weak-dependence conditions associated with projection central-limit phenomena, such normalized projections are naturally expected to exhibit approximately Gaussian behavior even when the complete encoded representation is itself strongly non-Gaussian. Assumption (V2) therefore does not require Gaussianity of the encoder output.

A complementary motivation follows from the geometry of the right singular vectors. Since batch samples are randomly drawn and their indices carry no intrinsic information, no individual sample-index axis is privileged. In the absence of systematic localization, the retained right-singular vectors are therefore expected to spread their mass over many of the B sample coordinates, although collective permutation-invariant directions may still occur.

An idealized reference case is an isotropically oriented unit vector in $\mathbb{R}^B$. Its individual coordinates have characteristic magnitude $B^{-1/2}$, while the scaled coordinates

$$\sqrt{B}\, v_i \tag{85}$$

approach standard Gaussian variables as the sample-space dimension increases. The same phenomenon holds for fixed-dimensional projections of isotropically distributed orthonormal frames.

The learned singular vectors are not assumed to be exactly isotropic or Haar distributed, and no limit $B \to \infty$ is required by the method. These high-dimensional arguments provide only a motivation for the Gaussian approximation. The adequacy of the Gaussian approximation in Equation 82 is assessed empirically in Appendix A.5, where we examine both the row-wise distributions of the normalized right-singular coordinates and random one-dimensional projections of the full K-dimensional ensemble.

Proposition: standard Gaussian approximation of the normalized right-singular coordinates. Under assumptions (V1)–(V2), the normalized right-singular coordinate obtained by randomly selecting a converged batch and a sample within that batch is adequately represented, for post-training sampling, by the standard Gaussian reference

$$\boxed{W = \sqrt{B}\, v_I^{\mathcal{B}} \overset{d}{\approx} \mathcal{N}(0, I_K),} \tag{86}$$

where the approximation is understood in the normalized average-direction sense quantified by (V1).

Justification. From the exact ensemble identity Equation 66,

$$\mathbb{E}_{\mathcal{B},I}\left[WW^\top\right] = I_K. \tag{87}$$

Assumption (V2) models the ensemble distribution using the Gaussian reference with the same first two moments,

$$W \overset{d}{\approx} \mathcal{N}\left(\mu_W, I_K - \mu_W \mu_W^\top\right). \tag{88}$$

Assumption (V1) gives

$$\frac{\|\mu_W\|_2}{\sqrt{K}} \ll 1. \tag{89}$$

Therefore, along a uniformly sampled unit direction t, the RMS mean displacement satisfies

$$\sqrt{\mathbb{E}_t\left[(t^\top \mu_W)^2\right]} = \frac{\|\mu_W\|_2}{\sqrt{K}} \ll 1. \tag{90}$$

Similarly, from

$$\mathrm{Cov}(W) = I_K - \mu_W \mu_W^\top, \tag{91}$$

the mean variance correction along a random direction is

$$\mathbb{E}_t\left[1 - t^\top \mathrm{Cov}(W) t\right] = \frac{\|\mu_W\|_2^2}{K} \ll 1. \tag{92}$$

Hence, along a typical direction, the Gaussian reference

$$\mathcal{N}\left(\mu_W, I_K - \mu_W \mu_W^\top\right) \tag{93}$$

has mean close to zero and variance close to one. This motivates the standard Gaussian approximation

$$W = \sqrt{B}\, v_I^{\mathcal{B}} \overset{d}{\approx} \mathcal{N}(0, I_K) \tag{94}$$

used for post-training generation.

Moreover, the standard Gaussian reference preserves exactly the uncentered second moment of the normalized right-singular ensemble. If

$$z \sim \mathcal{N}(0, I_K), \tag{95}$$

then

$$\mathbb{E}\left[zz^\top\right] = I_K = \mathbb{E}_{\mathcal{B},I}\left[WW^\top\right]. \tag{96}$$

Thus, the standard Gaussian reference both preserves the exact uncentered second-moment geometry and differs from the mean-aware Gaussian model only by a small normalized mean and covariance correction under assumption (V1).

Gaussian reference for the bottleneck coordinates. For a given converged batch, the clean bottleneck coordinate satisfies

$$a_i^{\mathcal{B}} = \frac{1}{\sqrt{B}} \Sigma_K^{\mathcal{B}} w_i^{\mathcal{B}}. \tag{97}$$

Equation 86 motivates replacing the normalized right-singular coordinate in the post-training sampling model by a standard Gaussian reference

$$z \sim \mathcal{N}(0, I_K). \tag{98}$$

For a batch spectrum $\Sigma_K^{\mathcal{B}}$, this gives the corresponding Gaussian reference coordinate

$$a_{\mathrm{ref}}^{\mathcal{B}} = \frac{1}{\sqrt{B}} \Sigma_K^{\mathcal{B}} z, \tag{99}$$

with

$$\boxed{a_{\mathrm{ref}}^{\mathcal{B}} \sim \mathcal{N}\left(0, \frac{1}{B}\left(\Sigma_K^{\mathcal{B}}\right)^2\right).} \tag{100}$$

Equation 100 should be interpreted as the Gaussian reference induced by the empirically observed ensemble geometry of the normalized right-singular coordinates, rather than as a claim that every individual finite batch or every individual retained singular direction is exactly centered.

The key point is that the standard Gaussian reference preserves the exact uncentered second moment of the normalized right-singular coordinates, while assumption (V1) ensures that the pooled mean contribution is small relative to the total K-dimensional second-moment scale.

In Appendix A.6, this reference is combined with the stochastic perturbation derived in Appendix A.2 and with the global geometry constructed in Appendix A.1 to obtain the final post-training sampling rule.

A.5 Empirical Verification of the Gaussian Right-Singular Geometry

Appendix A.4 showed that the SOLA generation rule is motivated by two properties of the ensemble of normalized right-singular coordinates

$$W = \sqrt{B}\, v_I^{\mathcal{B}}, \tag{101}$$

obtained by randomly selecting a converged batch $\mathcal{B}$ and a sample I within that batch. These properties are the small dimension-normalized pooled mean assumption (V1) and the approximate Gaussian ensemble model (V2).

In this section, we examine these assumptions directly on the learned right-singular coordinates using trained SOLA models on Frey Faces and CelebA.

For Frey Faces, we use the seed-0 trained SOLA model with $K = 10$, $L = 200$, $\sigma = 0.1$, and training batch size $B = 32$. The encoder is kept fixed after training, and the diagnostics are computed by repeatedly drawing batches from the training split. To examine the finite-batch behavior, the right-singular coordinates are evaluated over independently drawn batches of sizes

$$B \in \{16, 32, 64, 128\}. \tag{102}$$

For each batch size, 150 random batches are used. The actual batch size used during training is $B = 32$.

For CelebA, we use the trained SOLA model with $K = 186$, $L = 512$, $\sigma = 0.1$, and the actual training batch size $B = 576$. We again draw 150 random batches from the training split, giving a total of 86400 pooled normalized right-singular coordinate vectors.

Empirical verification of (V1): small dimension-normalized pooled mean. Assumption (V1) concerns the ensemble mean

$$\mu_W = \mathbb{E}_{\mathcal{B},I}\left[W\right], \tag{103}$$

rather than the empirical mean of each individual finite batch.

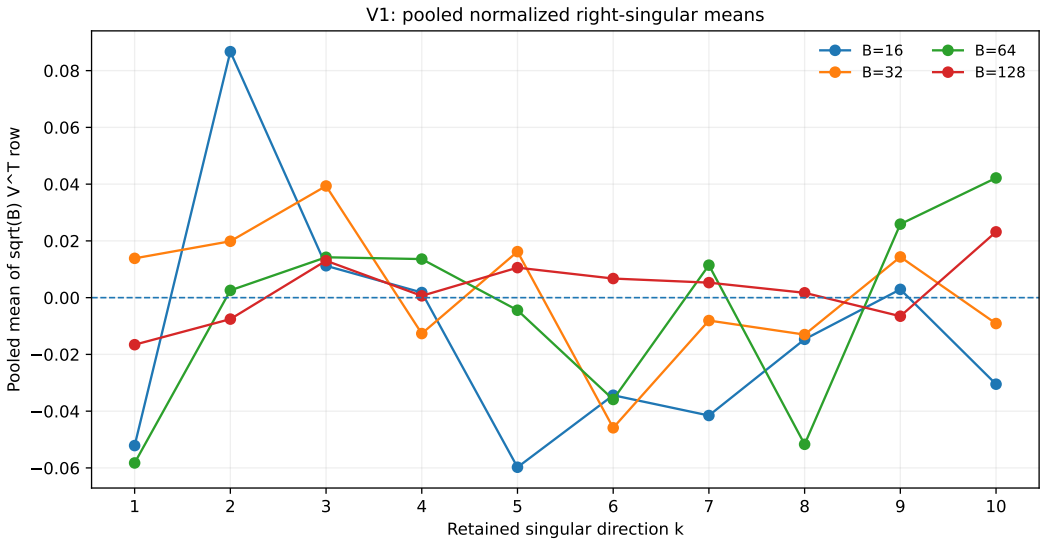


Figure 4: Pooled mean components of the normalized right-singular coordinates $\sqrt{B}\,V_K^\top$ on Frey Faces across independently drawn batches and sample indices. Results are shown for several finite batch sizes around the actual training batch size $B = 32$.

Table 4: Pooled-mean diagnostics on Frey Faces as a function of batch size. The Haar reference is obtained from isotropically distributed orthonormal frames with matched B and K.

B	$\|\widehat{\mu}_W\|_2$	$\max_k \|\widehat{\mu}_{W,k}\|$	Haar $\|\widehat{\mu}_W\|_2$
16	0.1341	0.0867	0.0812
32	0.0720	0.0459	0.0585
64	0.1017	0.0583	0.0375
128	0.0356	0.0232	0.0272

For a fixed batch size B, let

$$W_{\mathcal{B}_b} = \sqrt{B}\,V_K^{\mathcal{B}_b\top} = \left[w_1^{\mathcal{B}_b} \quad \cdots \quad w_B^{\mathcal{B}_b}\right], \qquad b = 1, \ldots, M, \tag{104}$$

where M independently drawn batches are considered. We estimate the ensemble mean by pooling both the batch and sample indices,

$$\widehat{\mu}_W = \frac{1}{MB}\sum_{b=1}^{M}\sum_{i=1}^{B} w_i^{\mathcal{B}_b}. \tag{105}$$

Assumption (V1) is evaluated through the dimension-normalized quantity

$$\widehat{\epsilon}_\mu = \frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}}, \tag{106}$$

which measures the RMS pooled mean per retained coordinate. As shown in Appendix A.4, it also equals the RMS mean shift along a uniformly sampled unit direction.

Figure 4 reports the individual components of $\widehat{\mu}_W$ for Frey Faces for all retained singular directions and all evaluated batch sizes.

Table 4 summarizes the Euclidean norm of the pooled mean and its largest component magnitude for the Frey Faces finite-B study. For comparison, we also report the corresponding pooled mean obtained from isotropically distributed Haar orthonormal frames; the construction of this reference is detailed below in the analysis of (V2).

At the actual Frey Faces training batch size $B = 32$,

$$\|\widehat{\mu}_W\|_2 = 0.0720, \qquad \frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}} = 0.0228, \tag{107}$$

Table 5: Empirical verification of assumption (V1) at the actual SOLA training configurations. $N_{\mathrm{pool}} = MB$ denotes the total number of pooled normalized right-singular coordinate vectors.

Dataset	K	B	N_{pool}	$\|\widehat{\mu}_W\|_2$	$\|\widehat{\mu}_W\|_2/\sqrt{K}$	$\max_k \|\widehat{\mu}_{W,k}\|$
Frey Faces	10	32	4800	0.0720	0.0228	0.0459
CelebA	186	576	86400	0.8532	0.0626	0.8479

with

$$\max_k |\widehat{\mu}_{W,k}| = 0.0459. \tag{108}$$

The normalized pooled mean is therefore approximately 2.3% of the unit coordinate scale.

To test the same assumption in a substantially higher-dimensional representation, Table 5 compares Frey Faces and CelebA at their respective actual SOLA training batch sizes.

For CelebA,

$$\frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}} = 0.0626. \tag{109}$$

Thus, although the unnormalized Euclidean norm of the pooled mean is larger because the retained space has $K = 186$ dimensions, its RMS magnitude per retained coordinate remains approximately 6.3% of the unit normalized-coordinate scale.

Equivalently, the relative mean-energy contribution appearing in (V1) is

$$\frac{\|\widehat{\mu}_W\|_2^2}{K} = 0.00052 \tag{110}$$

for Frey Faces and

$$\frac{\|\widehat{\mu}_W\|_2^2}{K} = 0.00391 \tag{111}$$

for CelebA. Hence, the pooled-mean contribution represents approximately 0.052% and 0.391%, respectively, of the total normalized second-moment energy.

The CelebA maximum component,

$$\max_k |\widehat{\mu}_{W,k}| = 0.8479, \tag{112}$$

shows that the pooled mean is not uniformly small in every individual singular direction. This does not contradict (V1), which is a dimension-normalized condition and therefore concerns the mean contribution relative to the complete K-dimensional geometry rather than a worst-direction bound.

Taken together, the two datasets support the empirical condition

$$\boxed{\frac{\|\mu_W\|_2}{\sqrt{K}} \ll 1} \tag{113}$$

used in (V1).

Empirical verification of (V2): Gaussian ensemble geometry. Assumption (V2) models the ensemble distribution of W using the Gaussian reference

$$W \overset{d}{\approx} \mathcal{N}\left(\mu_W, I_K - \mu_W \mu_W^\top\right). \tag{114}$$

The quality of this approximation is examined using both individual coordinate distributions and random projections of the complete K-dimensional ensemble.

First, we pool the values

$$\sqrt{B}\, v_{ki}^{\mathcal{B}} \tag{115}$$

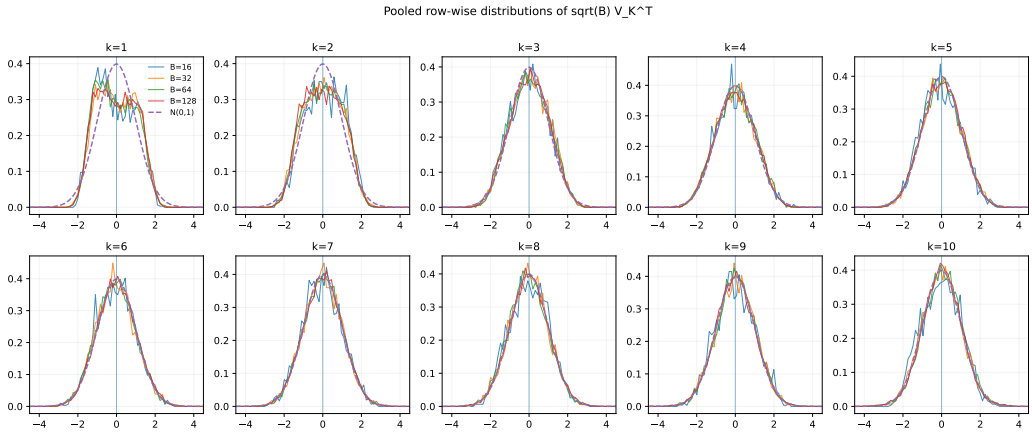


Figure 5: Pooled row-wise distributions of the normalized right-singular coordinates $\sqrt{B}\,V_K^\top$ on Frey Faces for several finite batch sizes. Most retained directions exhibit close agreement with the Gaussian reference, while some singular directions show moderate direction-specific departures from an exact Gaussian law.

over independently sampled batches and sample indices for every retained singular direction k. Figure 5 compares these empirical distributions with the Gaussian reference for the Frey Faces finite-B study.

At the actual Frey Faces training batch size $B = 32$, the median row-wise normal Q–Q error over the $K = 10$ retained directions is

$$0.0317, \tag{116}$$

while the largest value is

$$0.1423. \tag{117}$$

For CelebA, with $K = 186$ and $B = 576$, the median row-wise normal Q–Q error is

$$0.0371, \tag{118}$$

while the largest direction-specific value is

$$0.4084. \tag{119}$$

Thus, the median row-wise errors are similar on the two datasets, although both contain individual singular directions that depart more strongly from exact Gaussianity. This is consistent with (V2) being an ensemble approximation rather than a requirement of exact Gaussianity along every retained singular direction.

Since Gaussian marginal distributions alone do not characterize a multivariate distribution, we additionally examine random one-dimensional projections of the full ensemble.

For a unit vector

$$t \in \mathbb{R}^K, \qquad \|t\|_2 = 1, \tag{120}$$

we define

$$R_t = \frac{t^\top (W - \widehat{\mu}_W)}{\sqrt{t^\top \left(I_K - \widehat{\mu}_W \widehat{\mu}_W^\top\right) t}}. \tag{121}$$

Under the Gaussian reference in (V2), these projected variables should be approximately standard Gaussian. We evaluate Equation 121 for multiple randomly sampled directions t.

To account for finite-B effects produced purely by orthogonality, we also construct a matched isotropic reference. A random $B \times K$ Gaussian matrix is orthogonalized to obtain a Haar orthonormal frame Q, from which

$$W_{\text{Haar}} = \sqrt{B}\,Q^\top \tag{122}$$

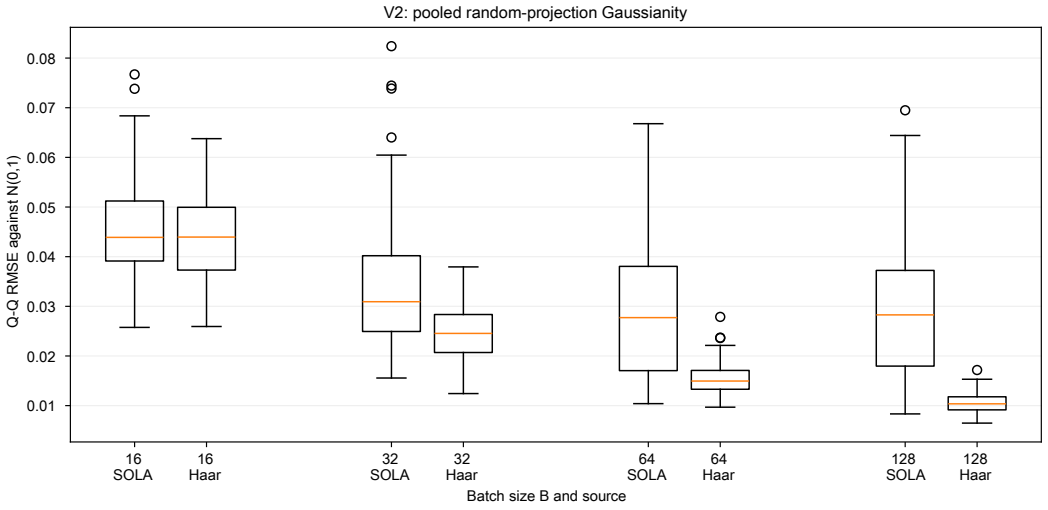


Figure 6: Normal Q–Q error of random one-dimensional projections of the pooled Frey Faces normalized right-singular coordinates, compared with matched Haar-distributed orthonormal frames. The SOLA ensemble exhibits near-Gaussian projection behavior over the evaluated finite batch sizes.

Table 6: Median normal Q–Q error of random ensemble projections on Frey Faces for SOLA and the matched Haar reference.

B	SOLA	Haar reference
16	0.0439	0.0440
32	0.0309	0.0245
64	0.0277	0.0150
128	0.0283	0.0104

is formed. This reference obeys exactly the same second-moment constraint

$$\frac{1}{B} W_{\mathrm{Haar}} W_{\mathrm{Haar}}^{\top} = I_K \tag{123}$$

as the learned normalized right-singular coordinates, while providing the idealized isotropic geometry motivating (V2).

Figure 6 compares the normal Q–Q errors of the Frey Faces SOLA projections with this Haar reference.

The corresponding median Q–Q errors are reported in Table 6.

At the Frey Faces training batch size $B = 32$, the median random-projection Q–Q error is 0.0309, compared with 0.0245 for the isotropic Haar reference. Representative Q–Q plots are shown in Figure 7.

Table 7 compares the complete Gaussian diagnostics for Frey Faces and CelebA at their respective actual SOLA training configurations.

Despite the substantial difference in dimensionality and batch size, the median random-projection errors of the learned ensembles are very similar:

$$0.0309 \qquad \text{for Frey Faces,} \tag{124}$$

and

$$0.0315 \qquad \text{for CelebA.} \tag{125}$$

The matched Haar reference is closer to an exact Gaussian projection on CelebA, with median Q–Q error 0.0049, showing that the learned right-singular geometry is not identical to an ideal

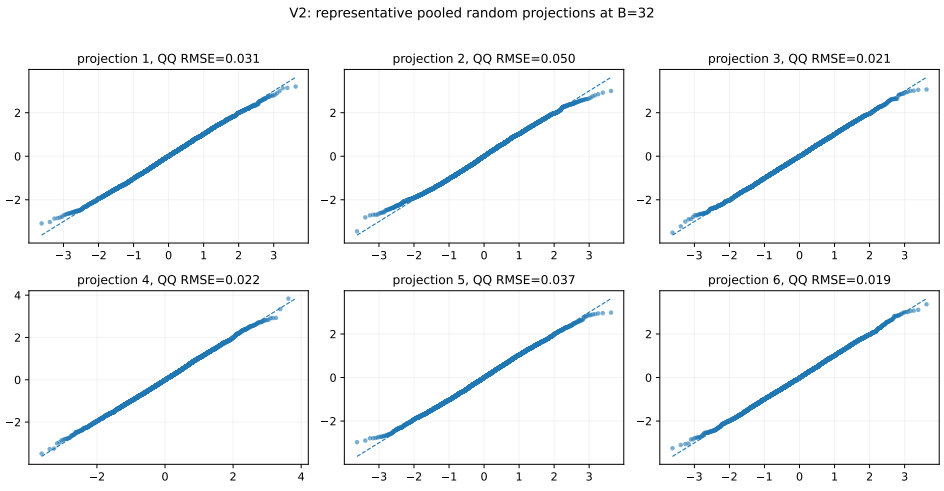


Figure 7: Representative random-projection Q–Q plots of the pooled normalized right-singular coordinates on Frey Faces at the training batch size $B = 32$. The projected ensemble coordinates closely follow the standard Gaussian reference after centering and variance normalization.

Table 7: Normal Q–Q diagnostics for the normalized right-singular coordinates at the actual SOLA training batch sizes. Projection statistics are medians over randomly sampled unit directions. The last column reports the corresponding matched Haar reference.

Dataset	K	B	Row median	Row max	Projection median	Haar projection median
Frey Faces	10	32	0.0317	0.1423	0.0309	0.0245
CelebA	186	576	0.0371	0.4084	0.0315	0.0049

isotropic Haar frame. Nevertheless, the small median projection error indicates that typical centered projections of the learned ensemble remain well approximated by Gaussian variables.

Taken together, the pooled-mean and Gaussian-shape diagnostics support the two empirical properties introduced in Appendix A.4. In particular, the normalized pooled mean satisfies

$$\frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}} = 0.0228 \tag{126}$$

for Frey Faces and

$$\frac{\|\widehat{\mu}_W\|_2}{\sqrt{K}} = 0.0626 \tag{127}$$

for CelebA, corresponding to relative mean-energy contributions of only 0.052% and 0.391%, respectively.

The random-projection diagnostics further show that the ensemble distribution is close to the Gaussian model in typical latent directions. These observations empirically support the standard Gaussian reference

$$\boxed{W = \sqrt{B}\, v_I^{\mathcal{B}} \overset{d}{\approx} \mathcal{N}(0, I_K),} \tag{128}$$

in the dimension-normalized, average-direction sense introduced in Appendix A.4.

The row-wise diagnostics additionally show that this approximation should not be interpreted as an exact Gaussian law or an exact zero-mean condition for every retained singular direction. Rather, the native SOLA generation rule uses the global Gaussian reference supported by the small normalized pooled-mean energy, the exact second-moment identity, and the observed near-Gaussian behavior of typical projections.

A.6 Derivation of the Global Gaussian Generator

Appendix A.4 motivates the Gaussian reference for the normalized right-singular coordinates, whose empirical adequacy is verified in Appendix A.5. In this section, we combine the stochastic perturbation derived in Appendix A.2 with the global geometry construction of Appendix A.1 to obtain the final post-training sampling rule. The resulting generator is a Gaussian model for the ensemble of normalized right-singular coordinates, rather than for the complete encoder output.

For a converged batch $\mathcal{B}$, the clean bottleneck coordinate is

$$a_i^{\mathcal{B}} = \Sigma_K^{\mathcal{B}} v_i^{\mathcal{B}} = \frac{1}{\sqrt{B}} \Sigma_K^{\mathcal{B}} w_i^{\mathcal{B}}. \tag{129}$$

Appendix A.4 therefore motivates replacing the normalized right-singular coordinate in the post-training reference model by

$$z \sim \mathcal{N}(0, I_K). \tag{130}$$

The corresponding clean Gaussian reference is

$$a_{\mathrm{ref}}^{\mathcal{B}} = \frac{1}{\sqrt{B}} \Sigma_K^{\mathcal{B}} z. \tag{131}$$

Appendix A.2 showed that the SOLA perturbation is generated by an independent Gaussian variable

$$\xi \sim \mathcal{N}(0, I_K) \tag{132}$$

through the term

$$\frac{\sigma}{\sqrt{B}} \Sigma_K^{\mathcal{B}} \xi. \tag{133}$$

The stochastic Gaussian reference for the bottleneck is therefore

$$\widetilde{a}_{\mathrm{ref}}^{\mathcal{B}} = \frac{1}{\sqrt{B}} \Sigma_K^{\mathcal{B}} \left(z + \sigma \xi \right), \qquad z, \xi \overset{\mathrm{ind}}{\sim} \mathcal{N}(0, I_K). \tag{134}$$

Since

$$z + \sigma \xi \sim \mathcal{N}\left(0, (1+\sigma^2) I_K\right), \tag{135}$$

we obtain

$$\boxed{\widetilde{a}_{\mathrm{ref}}^{\mathcal{B}} \sim \mathcal{N}\left(0, \frac{1+\sigma^2}{B} \left(\Sigma_K^{\mathcal{B}}\right)^2\right).} \tag{136}$$

Equation 136 is the stochastic Gaussian reference used for post-training generation. It should not be interpreted as requiring the empirical coordinates of every individual finite batch to follow an exact Gaussian distribution.

For a retained converged batch $\mathcal{B}_b$ of size B_b, the stochastic reference covariance in its local basis is

$$\frac{1+\sigma^2}{B_b} \left(\Sigma_b\right)^2. \tag{137}$$

After representation through the corresponding left-singular basis, the associated geometry is

$$\frac{1+\sigma^2}{B_b} U_b \left(\Sigma_b\right)^2 U_b^\top. \tag{138}$$

Appendix A.1 showed that the batch-dependent geometries are consolidated by defining

$$W_f = \left[U_1 \Sigma_1 \quad \cdots \quad U_{N_b} \Sigma_{N_b} \right], \tag{139}$$

with singular value decomposition

$$W_f = \bar{U} \, \bar{\Sigma} \, \bar{V}^\top, \tag{140}$$

from which U_f and Σ_f denote the leading K left-singular vectors and singular values.

Let

$$N_{\text{rep}} = \sum_{b=1}^{N_b} B_b. \tag{141}$$

Then the exact aggregated geometry is

$$\frac{1}{N_{\text{rep}}} W_f W_f^\top = \frac{1}{N_{\text{rep}}} \sum_{b=1}^{N_b} U_b \left(\Sigma_b\right)^2 U_b^\top. \tag{142}$$

Equivalently,

$$\frac{1}{N_{\text{rep}}} W_f W_f^\top = \sum_{b=1}^{N_b} \frac{B_b}{N_{\text{rep}}} \left(\frac{1}{B_b} U_b \left(\Sigma_b\right)^2 U_b^\top \right), \tag{143}$$

so the aggregated geometry is the sample-count-weighted average of the retained batch geometries.

In the common rank-K limit described in Appendix A.1,

$$\frac{1}{N_{\text{rep}}} W_f W_f^\top = U_f \frac{\left(\Sigma_f\right)^2}{N_{\text{rep}}} U_f^\top. \tag{144}$$

For finite training,

$$U_f \frac{\left(\Sigma_f\right)^2}{N_{\text{rep}}} U_f^\top \tag{145}$$

is the corresponding rank-K approximation used by SOLA.

Applying the stochastic inflation factor $1+\sigma^2$ from Equation 136 gives the rank-K global generation geometry

$$(1+\sigma^2) U_f \frac{\left(\Sigma_f\right)^2}{N_{\text{rep}}} U_f^\top. \tag{146}$$

This geometry can be sampled directly in the fixed rank-K basis. Let

$$z_{\text{gen}} \sim \mathcal{N}(0, I_K), \tag{147}$$

and define

$$\boxed{\Sigma_{\text{gen}} = \frac{\sqrt{1+\sigma^2}}{\sqrt{N_{\text{rep}}}} \Sigma_f.} \tag{148}$$

The generated bottleneck coordinate is

$$\boxed{a_{\text{gen}} = \Sigma_{\text{gen}} z_{\text{gen}}, \qquad z_{\text{gen}} \sim \mathcal{N}(0, I_K),} \tag{149}$$

and generated samples are finally obtained through

$$\boxed{x_{\text{gen}} = g_\phi \left(U_f \Sigma_{\text{gen}} z_{\text{gen}} \right).} \tag{150}$$

The standard Gaussian variable z_{gen} therefore represents the empirically supported Gaussian reference for the ensemble of normalized right-singular coordinates. It is not introduced as a Gaussian model for the complete encoder output.

A.7 Construction of the Synthetic Gaussian Dataset

The synthetic benchmark is designed to provide a simple generative setting in which the structure of valid samples is known analytically. Each sample is a two-dimensional Gaussian field defined on a regular 64×64 grid over the square domain $[-1, 1]^2$. More precisely, letting $(x, y) \in [-1, 1]^2$, each image is generated as

$$I(x, y; x_0, y_0) = \exp\left(-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma_g^2} \right), \qquad \sigma_g = 0.2, \tag{151}$$

where (x_0, y_0) denotes the center of the Gaussian bump. The amplitude is fixed to one for all samples, so the variability of the dataset is entirely determined by the location of the center.

The training and test sets differ only in the way these centers are sampled. For a training set of size N_{tr}, the centers are placed on a regular grid covering the square $[-0.5, 0.5]^2$. In practice, if

$$n_g = \lfloor \sqrt{N_{\mathrm{tr}}} \rfloor + 1,$$

we form two one-dimensional grids of length n_g in $[-0.5, 0.5]$, build their Cartesian product, and retain the first N_{tr} grid points as training centers. This yields a structured sampling of the center space. In contrast, the test centers are drawn independently and uniformly from $[-0.5, 0.5]^2$, which provides out-of-grid but in-distribution Gaussian samples for evaluation.

After generation, the training-set mean and standard deviation are computed and used to normalize both the training and test sets. If I_{raw} denotes a raw image, the normalized image is

$$I_{\mathrm{norm}} = \frac{I_{\mathrm{raw}} - \mu_{\mathrm{tr}}}{\sigma_{\mathrm{tr}}}, \tag{152}$$

where μ_{tr} and σ_{tr} denote the empirical mean and standard deviation of the training set. The normalized images are then provided to the models as single-channel tensors of shape $1 \times 64 \times 64$.

Representative examples of the resulting dataset are shown in Figure 8.

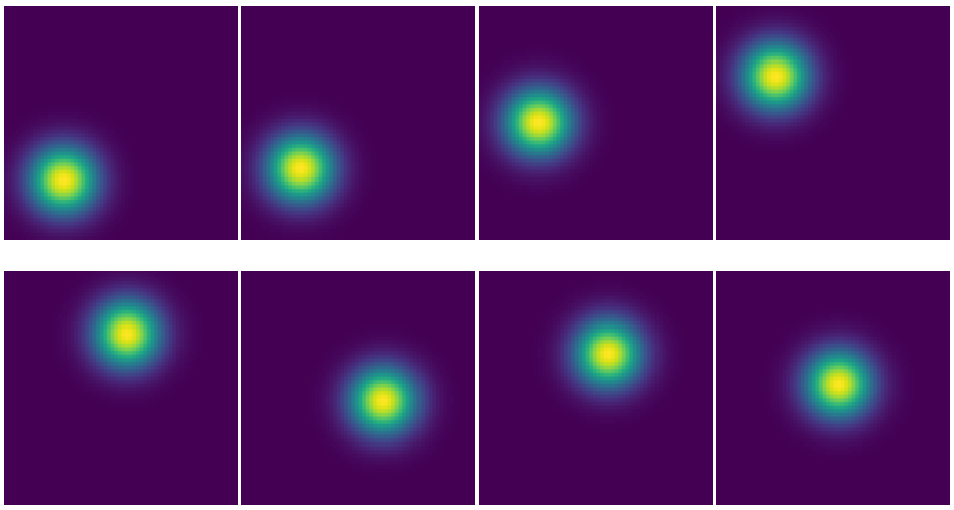


Figure 8: Representative samples from the synthetic Gaussian dataset. The top row shows training examples generated from Gaussian centers placed on a regular grid in $[-0.5, 0.5]^2$, while the bottom row shows test examples whose centers are sampled uniformly at random from the same square.

A.8 Component Ablation: Bottleneck Dimensionality, SVD Geometry, and Stochastic Mixing

We next isolate the respective contributions of bottleneck dimensionality, SVD-based population geometry, and stochastic mixing. A direct comparison between SOLA and an unstructured stochastic autoencoder is potentially confounded by the fact that SOLA also reduces a wide latent representation to rank K. We therefore consider five variants:

- **AE (wide):** a deterministic autoencoder with the same wide latent dimension used before the SOLA bottleneck, with neither SVD nor stochastic mixing.
- **G-only (wide):** stochastic mixing is applied directly to the wide latent representation, without an SVD.
- **G-only (small):** stochastic mixing is applied to a directly parameterized K-dimensional bottleneck, without an SVD. This controls for the possibility that the benefit of the SVD-based model is explained primarily by dimensionality reduction.

- **RRAE / SVD-only:** the wide representation is reduced to rank K using the RRAE bottleneck, but no stochastic mixing is applied.
- **SOLA:** the complete method, combining the rank-K SVD representation with stochastic mixing.

All variants use the same encoder–decoder family and optimization protocol within each dataset. The wide latent dimension is $L = 200$ and the bottleneck dimension is $K = 2$ for the synthetic Gaussian benchmark. For CelebA we use $L = 512$ and $K = 186$. The stochastic variants use $\sigma = 0.1$.

An important distinction concerns generation. The AE and G-only variants do not define a native post-training sampling distribution. In the primary component ablation, we therefore use the deliberately simple *standard-normal sampling probe*

$$z \sim \mathcal{N}(0, I_d), \tag{153}$$

where d is the actual bottleneck dimension, and decode z directly. These results measure whether the learned representation itself is compatible with a simple prescribed sampling law, and they should not be interpreted as an estimate of the best generation performance obtainable after fitting an additional density model to the latent codes.

In contrast, RRAE and SOLA use their native SVD-derived generators. RRAE samples using the clean global SVD scale $\Sigma_f/\sqrt{N_{\mathrm{rep}}}$, while SOLA uses the stochastic scale $\sqrt{1+\sigma^2}\,\Sigma_f/\sqrt{N_{\mathrm{rep}}}$ derived in Appendix A.6. No post-hoc latent density estimation is used for any variant in these first two ablation tables.

Table 8: Component ablation on the synthetic Gaussian benchmark. The generation metric is the Gaussian-fit relative error (%, lower is better). AE and G-only variants use a standard-normal sampling probe, whereas RRAE and SOLA use their native SVD-derived generators. Values are mean $\pm$ sample standard deviation over five seeds.

Method	Test reconstruction MSE ↓	Generation error (%) ↓
AE (wide)	$\mathbf{0.00149 \pm 0.00060}$	90.85 ± 5.14
G-only (wide)	0.00371 ± 0.00084	85.82 ± 12.10
G-only (small)	0.01003 ± 0.00302	$\mathbf{4.45 \pm 1.30}$
RRAE / SVD-only	0.01362 ± 0.01187	8.60 ± 3.41
SOLA	0.02116 ± 0.00862	5.90 ± 1.36

The synthetic experiment reveals that dimensionality reduction itself is an important part of the generative behavior. Both wide models reconstruct the data accurately but perform poorly under the standard-normal sampling probe: the AE and wide G-only model obtain generation errors of 90.85% and 85.82%, respectively. Thus, stochastic mixing by itself does not make an arbitrary wide latent representation compatible with standard-normal sampling.

In contrast, imposing the correct low-dimensional bottleneck directly causes the G-only model to improve dramatically, reaching $4.45 \pm 1.30\%$, slightly outperforming SOLA on this benchmark. Consequently, the strong synthetic performance of SOLA cannot be attributed uniquely to the SVD: for this simple problem, whose intrinsic variability is explicitly two-dimensional, a directly parameterized compact stochastic bottleneck is already sufficient to learn a representation that aligns well with the standard-normal sampling probe.

The SVD-only model also achieves strong generation ($8.60 \pm 3.41\%$), but exhibits greater variability across seeds than SOLA. Adding stochastic mixing reduces the generation error to $5.90 \pm 1.36\%$. These results indicate that both compactness and the stochastic perturbation can contribute to generation, while also showing that the relative importance of the SVD cannot be established from the simple two-dimensional benchmark alone.

We therefore repeat the same ablation on CelebA, where the representation must capture substantially more complex image variability.

The CelebA results separate reconstruction quality from direct generative sampleability more clearly. All five variants achieve similar reconstruction errors, between approximately 0.09 and

Table 9: Component ablation on CelebA under direct generation without post-hoc density fitting. AE and G-only variants use the standard-normal sampling probe, while RRAE and SOLA use their native SVD-derived generators. Values are mean $\pm$ sample standard deviation over three seeds. Lower is better.

Method	Test reconstruction MSE ↓	FID ↓
AE (wide)	0.0923 ± 0.0028	200.96 ± 18.53
G-only (wide)	0.0950 ± 0.0019	127.63 ± 9.96
G-only (small)	$\mathbf{0.0903 \pm 0.0047}$	113.67 ± 6.03
RRAE / SVD-only	0.0950 ± 0.0021	124.80 ± 11.83
SOLA	0.0924 ± 0.0035	$\mathbf{99.84 \pm 4.12}$

0.095, yet their direct generation performance differs substantially. In particular, the G-only small model attains the lowest reconstruction error but a higher FID than SOLA (113.67 ± 6.03 versus 99.84 ± 4.12). The difference in direct generation therefore cannot be explained simply by more accurate reconstruction.

Reducing the G-only bottleneck from the wide representation to $K = 186$ again provides a substantial improvement, reducing FID from 127.63 ± 9.96 to 113.67 ± 6.03. Compactness therefore remains important on CelebA. Unlike on the two-dimensional synthetic problem, however, the directly parameterized compact bottleneck does not reach the performance obtained with the population-derived SVD geometry and its native generator. SOLA achieves the lowest FID among the five variants, while the SVD-only model reaches 124.80 ± 11.83. The improvement from RRAE to SOLA further indicates that stochastic mixing contributes beyond the low-rank SVD representation alone.

These direct-generation results address whether the learned representation comes with, or happens to align with, a useful sampling law. They do not establish that the alternative representations are intrinsically unsuitable for generation. In particular, a deterministic autoencoder may learn a representation that reconstructs well while occupying a latent distribution that is substantially different from $\mathcal{N}(0, I)$. To distinguish latent-distribution mismatch from the quality of the learned representation and decoder, we perform an additional control experiment on CelebA in which a density is fitted explicitly after training.

Post-hoc latent-density control. For each of the AE and G-only variants, we encode the training set and fit a full-covariance Gaussian to the resulting latent codes. Generation is then performed by sampling from this fitted Gaussian and decoding the sampled latent vectors. This procedure estimates the first- and second-order structure of each learned latent distribution rather than requiring it to match a standard normal distribution.

We apply the same control to SOLA in its intrinsic K-dimensional coordinates. This is not required by the SOLA generation procedure, but provides a direct test of whether fitting an additional Gaussian density materially improves over its native population-derived generator.

Table 10: CelebA generation after post-hoc full-covariance Gaussian fitting. For AE and G-only variants, a Gaussian is fitted to the corresponding latent codes after training. For SOLA, the Gaussian is fitted to the intrinsic SVD coordinates. Values are mean $\pm$ sample standard deviation over three seeds. Lower is better.

Method	Post-hoc Gaussian FID ↓
AE (wide)	102.84 ± 6.44
G-only (wide)	102.64 ± 4.36
G-only (small)	102.22 ± 4.30
SOLA (fitted)	$\mathbf{100.63 \pm 2.88}$

The post-hoc control substantially changes the interpretation of the direct sampling results. Once an explicit full-covariance Gaussian is fitted, the three non-SVD ablations attain very similar generation performance, with FIDs between 102.22 and 102.84. These values are also close to the

performance of SOLA. Thus, the large differences observed under direct sampling do not imply that the alternative autoencoders have learned unusable representations or decoders.

This control is particularly informative for the wide deterministic autoencoder. Its direct-sampling FID is 200.96 ± 18.53, whereas fitting its latent mean and full covariance reduces the FID to 102.84 ± 6.44. The representation therefore contains sufficient information for substantially better generation once an additional density model is supplied. A similar, although smaller, effect is observed for the G-only variants.

Importantly, applying the same post-hoc treatment to SOLA produces little change relative to its native generator: the fitted intrinsic Gaussian obtains 100.63 ± 2.88, compared with 99.84 ± 4.12 for native SOLA in Table 9. The additional fitted density therefore does not provide a meaningful improvement in this experiment. This is consistent with the purpose of the SVD-derived SOLA generator: the population geometry required for sampling is already incorporated into the method rather than being estimated as a separate post-training model.

The full-covariance control is intentionally expressive. For a d-dimensional latent space, fitting its mean and covariance requires estimating

$$d + \frac{d(d+1)}{2} \tag{154}$$

distributional parameters, in addition to the parameters of the trained autoencoder. The resulting experiment should therefore be interpreted as a control for latent-density mismatch rather than as the native generation procedure of the ablated models.

Taken together, the two CelebA experiments answer complementary questions. The direct-sampling ablation evaluates whether the representation itself provides, or naturally aligns with, a usable sampling geometry. Under this criterion, the SVD-derived geometry and stochastic mixing of SOLA provide a clear advantage over the tested ablations. The post-hoc experiment instead asks whether the alternative representations can support good generation if an additional latent-density estimation stage is introduced. Under this stronger post-processing procedure, the performance gap largely disappears.

The role of SOLA should therefore not be interpreted as making post-hoc density modelling ineffective for conventional autoencoders. Rather, SOLA incorporates a useful population-derived sampling geometry directly into the learned representation and provides a native generator without requiring a separate density-fitting stage. The ablation further indicates that compactness, SVD-derived geometry, and stochastic mixing play distinct roles: compactness can be sufficient on a simple known low-dimensional manifold, while on CelebA the complete method combines a structured population geometry with stochastic training and remains directly sampleable without an additional latent model.

A.9 Mixing Constraints and Design of G

We next investigate whether the stochastic mixing matrix G should be constrained to represent an interpolation of the encoded samples, or whether the unconstrained signed construction used by SOLA is preferable. Recall that the proposed formulation is

$$G = I_B + \frac{\sigma}{\sqrt{B}} E, \qquad E_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \tag{155}$$

so that, for the intrinsic coordinates A_K,

$$\widetilde{A}_K = A_K G. \tag{156}$$

For a given column g_i of G, the corresponding stochastic coordinate can therefore be written as

$$\widetilde{a}_i = A_K g_i = \sum_{j=1}^{B} (g_i)_j a_j. \tag{157}$$

The constraints placed on g_i consequently determine whether this operation is a general linear combination, a positive combination, or a strict convex interpolation of the coordinates already present in the batch.

We compare four constructions. The proposed *signed-noise* formulation is the unconstrained construction in Equation 155. Its coefficients may be positive or negative and are not constrained to sum to one. It therefore permits general linear combinations and can move the stochastic coordinates outside the convex hull of the current batch. More importantly, the random component is zero-mean,

$$\mathbb{E}[G] = I_B, \tag{158}$$

and hence

$$\mathbb{E}[\widetilde{A}_K \mid A_K] = A_K. \tag{159}$$

The stochastic transformation is therefore centered on the clean rank-reduced representation. This is precisely the property used in the derivation of the SOLA stochastic bottleneck.

The *positive-noise* variant removes negative mixing coefficients by replacing E with $|E|$,

$$G = I_B + \frac{\sigma}{\sqrt{B}}|E|. \tag{160}$$

The resulting coefficients are nonnegative, but their columns are not normalized and therefore do not define convex combinations. In particular, this construction no longer gives a zero-mean perturbation: $\mathbb{E}[|E|] \neq 0$. Consequently, the stochastic representation is systematically shifted rather than being centered on A_K.

The *softmax-mixing* variant explicitly enforces interpolation. Starting from the same random matrix, we define

$$G = \mathrm{softmax}_{\mathrm{col}}\left(\frac{I_B + \frac{\sigma}{\sqrt{B}}E}{T}\right), \tag{161}$$

where the softmax is applied independently to each column and T is a temperature parameter. Every column then satisfies

$$(g_i)_j > 0, \qquad \sum_{j=1}^{B}(g_i)_j = 1, \tag{162}$$

so that every transformed coordinate is a strict convex combination of the coordinates in the current batch. This prevents the stochastic representation from leaving their convex hull.

Finally, the *Gaussian-kernel* construction removes the random matrix entirely and determines the mixing coefficients from pairwise distances between the intrinsic coordinates. For coordinates a_i and a_j, we set

$$(G)_{ji} \propto \exp\left(-\frac{\|a_j - a_i\|_2^2}{2\tau^2}\right), \tag{163}$$

followed by column normalization. This again gives nonnegative coefficients summing to one, but now produces a deterministic, data-dependent local averaging operation rather than a random perturbation.

The four constructions therefore impose progressively different restrictions on the latent mixing:

Design	Signed coefficients	Column sum $= 1$	Random
Signed noise	✓	×	✓
Positive noise	×	×	✓
Softmax mixing	×	✓	✓
Gaussian kernel	×	✓	×

Thus, the ablation directly tests whether restricting the stochastic transformation toward positive or convex interpolation improves the learned generative representation.

All four variants are applied after the same truncated SVD and use the same rank, network architecture, optimizer, training budget, and training seeds. The stochastic variants use the same noise scale $\sigma = 0.1$. For the softmax construction we use $T = 0.2$, and for the Gaussian kernel we use the fixed bandwidth $\tau = 0.3$. To isolate the effect of the training-time mixing rule, all variants are evaluated using the same SVD-based post-training sampling procedure. On the synthetic Gaussian benchmark, generation quality is measured using the Gaussian-fit relative error, whereas Frey Faces is evaluated using FID.

Table 11: Ablation of the design and constraints imposed on the latent mixing matrix G. Lower values are better. Results are mean $\pm$ sample standard deviation over five seeds.

G design	Gaussian-fit error ↓	Frey FID ↓
Signed noise (ours)	$\mathbf{5.90 \pm 1.36}$	$\mathbf{27.63 \pm 1.34}$
Positive noise	12.64 ± 4.35	43.91 ± 3.58
Softmax mixing	17.41 ± 12.38	67.06 ± 9.64
Gaussian kernel	32.06 ± 42.16	95.09 ± 5.69

The results can be seen in Table 11. Unconstrained signed construction gives the best generation performance on both datasets and also exhibits the lowest variability among the tested designs. Removing the negative coefficients while retaining an unnormalized random mixture already degrades performance, indicating that restricting the direction of the perturbation is detrimental. Enforcing a strict convex combination through the softmax degrades generation further, despite ensuring that every stochastic coordinate lies inside the convex hull of the current batch. The deterministic Gaussian-kernel interpolation performs worst overall and is particularly unstable on the synthetic benchmark.

These results indicate that the role of G is not simply to interpolate between existing latent samples. In the proposed construction, G instead provides a zero-mean stochastic perturbation around each clean coordinate, with directions and amplitudes governed by the population geometry exposed by the SVD. Allowing signed coefficients permits these perturbations to explore directions both inside and outside the convex hull of the current mini-batch, while preserving the clean representation in expectation. Constraining the coefficients to be positive or to form convex combinations removes this property and leads to consistently poorer generation in the present experiments.

A.10 Sensitivity to the Retained Rank

We examine the sensitivity of SOLA to the retained rank K on Frey Faces and Shapes3D. In SOLA, K is the dimension of the effective latent bottleneck. This is not an additional method-specific regularization hyperparameter: every autoencoder requires a choice of latent bottleneck dimension, and SOLA expresses this choice through the retained SVD rank.

In practice, K can be treated like the latent dimensionality of a conventional autoencoder. When prior knowledge is available, a natural value can be selected directly. For example, the synthetic Gaussian benchmark has two underlying degrees of freedom, motivating $K = 2$. When such prior knowledge is unavailable, SOLA can also inherit the adaptive rank-selection strategy introduced for RRAEs (Mounayer et al., 2024), which determines the effective retained rank automatically during training. We do not use this adaptive scheme in the experiments reported here because fixed bottleneck dimensions were sufficient for the present study. The sweep below is included to illustrate the effect of this standard architectural choice, rather than to introduce an additional SOLA-specific regularization hyperparameter.

The wide latent dimension is fixed at $L = 200$, while all other settings, including the signed mixing matrix G, noise scale $\sigma = 0.1$, optimizer, and training budget, are held fixed. Each value of K is evaluated over five seeds.

Table 12 shows that increasing the retained rank improves reconstruction substantially on both datasets. On Frey Faces, reconstruction MSE decreases from 0.111 ± 0.003 at $K = 2$ to 0.0174 ± 0.001 at $K = 20$. The largest improvements occur at the lower ranks, with the reconstruction error already reduced to 0.022 ± 0.001 at $K = 10$. Generation follows a similar trend: FID decreases from 48.02 ± 2.61 at $K = 2$ to 28.2 ± 1.32 at $K = 10$, with only a modest further improvement to 27.12 ± 0.34 at $K = 20$. We therefore use $K = 10$ in the main Frey Faces experiments as a compact bottleneck that captures most of the reconstruction and generation benefit obtained at the larger rank.

Shapes3D exhibits a clearer distinction between reconstruction and generation. Reconstruction improves strongly as K increases, decreasing from 0.02273 at $K = 2$ to 0.00025 at $K = 10$ and 0.00018 at $K = 20$. Generation quality, however, is non-monotonic. FID decreases from 167.4 at $K = 2$ to 57.3 at $K = 5$ and reaches its lowest value in the evaluated range at $K = 10$, with an FID

of 47.8. Increasing the rank further to $K = 20$ improves reconstruction slightly but increases the FID to 113.0.

Thus, increasing the bottleneck dimension does not necessarily improve the sampling geometry. On Frey Faces, both reconstruction and generation improve with rank but exhibit diminishing returns beyond $K = 10$. On Shapes3D, reconstruction continues to improve at the larger rank while generation degrades after $K = 10$. These results illustrate the usual architectural trade-off associated with selecting a latent bottleneck dimension and provide practical guidance for choosing K in SOLA. The retained rank should therefore be viewed as the latent dimensionality of the model rather than as an additional regularization parameter introduced by SOLA.

Table 12: Sensitivity of SOLA to the bottleneck dimension, expressed as the retained rank K, on Frey Faces and Shapes3D. The wide latent dimension is fixed at $L = 200$, and each configuration is evaluated over five seeds. Lower values are better.

Retained rank K	Frey recon. ↓	Frey FID ↓	Shapes3D recon. ↓	Shapes3D FID ↓
2	0.111 ± 0.003	48.02 ± 2.61	0.02241 ± 0.00002	170.34 ± 0.24
5	0.045 ± 0.000	34.04 ± 1.11	0.00239 ± 0.00009	59.26 ± 4.81
10	$\mathbf{0.021 \pm 0.000}$	$\mathbf{27.42 \pm 1.51}$	0.00025 ± 0.00001	$\mathbf{47.91 \pm 3.79}$
20	0.0374 ± 0.001	29.12 ± 0.34	$\mathbf{0.00018 \pm 0.00001}$	138.70 ± 14.21

A.11 Batch-Size Ablation

Because both the truncated SVD and the stochastic mixing matrix G are constructed from the current mini-batch, SOLA has an explicit dependence on the batch size B. We therefore examine whether generation quality is sensitive to this choice.

A direct comparison at a fixed number of epochs is potentially misleading. Increasing B reduces the number of optimizer updates performed per epoch, so an apparent batch-size effect may instead reflect a different optimization budget. For example, on Frey Faces, 200 epochs correspond to approximately 20,800 parameter updates for $B = 16$, but only approximately 2,600 updates for $B = 128$. We therefore compare

$$B \in \{16, 32, 64, 128\}$$

while adjusting the number of epochs so that the total number of optimizer updates is approximately matched across batch sizes. All other settings, including L, K, $\sigma = 0.1$, the signed mixing construction, optimizer, and learning rate, are kept unchanged. Each configuration is evaluated over three seeds.

These values are not intended to reproduce the main experimental results, since the number of training epochs is deliberately varied here to match the number of optimizer updates across batch sizes.

Frey Faces. For Frey Faces, we match the optimization budget at approximately 20,800 parameter updates. This corresponds to 200, 400, 800, and 1600 epochs for batch sizes 16, 32, 64, and 128, respectively.

Table 13: Batch-size sensitivity on Frey Faces under a matched optimization budget of approximately 20,800 parameter updates. Values are mean ± sample standard deviation over three seeds.

Batch size B	Epochs	Updates	Test recon. MSE ↓	FID ↓
16	200	20800	0.0236 ± 0.0012	22.96 ± 1.04
32	400	20800	0.0226 ± 0.0004	20.35 ± 1.03
64	800	20800	0.0228 ± 0.0013	18.96 ± 0.42
128	1600	20800	0.0227 ± 0.0012	17.99 ± 1.07

Reconstruction is effectively unchanged across the entire range of batch sizes, remaining between 0.0226 and 0.0236. Generation also remains stable and, if anything, improves moderately as the batch size increases: FID decreases from 22.96 ± 1.04 at $B = 16$ to 17.99 ± 1.07 at $B = 128$. Thus,

there is no evidence that using a larger batch degrades the latent geometry or the resulting generator once the optimization budget is controlled.

Synthetic Gaussian benchmark. We repeat the experiment on the synthetic Gaussian benchmark with $N_{\mathrm{tr}} = 400$ and $K = 2$, matching the training budget at approximately $5{,}000$ optimizer updates. Because the number of updates per epoch is not an exact divisor of the target for every B, the resulting budgets range only from $4{,}998$ to $5{,}005$ updates.

Table 14: Batch-size sensitivity on the synthetic Gaussian benchmark under an approximately matched $5{,}000$-update optimization budget. Generation is evaluated using the Gaussian-fit relative error (%). Values are mean $\pm$ sample standard deviation over three seeds.

Batch size B	Epochs	Updates	Test recon. MSE $\downarrow$	Gen. error (%) $\downarrow$
16	200	5000	0.0125 ± 0.0023	4.15 ± 0.89
32	385	5005	0.0083 ± 0.0018	3.21 ± 0.25
64	714	4998	0.0107 ± 0.0027	3.61 ± 0.12
128	1250	5000	0.0064 ± 0.0013	3.17 ± 0.28

The synthetic experiment shows the same qualitative behavior. Across an eight-fold change in batch size, the generation error remains within a narrow range, from 3.17% to 4.15%, with no systematic degradation as B increases. Reconstruction error is similarly small for all configurations.

As an additional control, we fix the batch size at $B = 128$ and vary only the duration of training. On Frey Faces, increasing the optimization budget from approximately $2{,}600$ to $20{,}800$ updates reduces the FID from approximately 61.6 to 18.0. The latter is consistent with the performance obtained by the smaller-batch configurations when trained for a comparable number of updates. This confirms that the poor performance observed for large batches under a fixed epoch budget is primarily an optimization budget effect rather than a failure of the batch-dependent SVD construction.

Taken together, these experiments indicate that SOLA is not critically dependent on a particular mini-batch size over the range considered here. Although changing B changes the finite-sample geometry from which the SVD and G are constructed, comparable generative performance is obtained once the number of parameter updates is controlled. In particular, increasing the batch size does not impair generation in either benchmark and can moderately improve it on Frey Faces.

A.12 Sensitivity to the Noise Scale σ

All main SOLA experiments in this paper use the same fixed noise scale $\sigma = 0.1$. In particular, σ is not selected separately for different datasets, and none of the reported main SOLA results uses a dataset-specific value. The purpose of this section is instead to characterize the sensitivity of SOLA to this fixed choice by examining how reconstruction and generation change when σ is deliberately varied.

Recall that σ controls the magnitude of the stochastic perturbation introduced by the mixing matrix,

$$G = I_B + \frac{\sigma}{\sqrt{B}} E. \tag{164}$$

Conditionally on the encoded batch, the resulting stochastic bottleneck has covariance proportional to σ^2,

$$\widetilde{a}_i \mid A_K \sim \mathcal{N}\left(a_i, \frac{\sigma^2}{B}\Sigma_K^2\right). \tag{165}$$

Thus, increasing σ directly increases the strength of the latent perturbation seen by the decoder during training. It also enters the post-training SOLA generator through the factor $\sqrt{1+\sigma^2}$, as derived in Appendix A.6.

To study this effect, we sweep

$$\sigma \in \{0.05, 0.1, 0.2, 0.5, 1.0\}$$

using the signed mixing construction, while keeping the remaining training settings unchanged. We consider the synthetic Gaussian benchmark with $N_{\mathrm{tr}} = 400$ and $K = 2$, and Frey Faces

with $K = 10$. Each configuration is evaluated over five seeds. We report test reconstruction error together with the corresponding generation metric, using the Gaussian-fit relative error for the synthetic benchmark and FID for Frey Faces.

Table 15: Reconstruction sensitivity to the stochastic noise scale σ. Values are mean $\pm$ sample standard deviation over five seeds. The reported values summarize the operating point used in the main experiments and the endpoint behavior of the sweep. Lower is better.

Dataset	σ	Test reconstruction MSE $\downarrow$
Synthetic Gaussian	0.05	0.0157 ± 0.0153
Synthetic Gaussian	1.0	0.3907 ± 0.0368
Frey Faces	0.05	0.0189 ± 0.0007
Frey Faces	0.1	0.0213 ± 0.0001
Frey Faces	1.0	0.1203 ± 0.0121

Table 15 shows a clear effect of increasing perturbation strength. On both datasets, reconstruction error increases approximately monotonically with σ. This behavior is expected: larger values of σ expose the decoder to progressively stronger perturbations of the clean latent representation.

Table 16: Generation sensitivity to the stochastic noise scale σ. Generation is evaluated using the Gaussian-fit relative error (%) for the synthetic benchmark and FID for Frey Faces. Values are mean $\pm$ sample standard deviation over five seeds. Lower is better.

Dataset	σ	Generation metric $\downarrow$
Synthetic Gaussian	0.1	$4.27 \pm 1.19\%$
Synthetic Gaussian	1.0	$48.13 \pm 2.38\%$
Frey Faces	0.1	27.42 ± 1.51
Frey Faces	1.0	56.96 ± 3.83

Generation quality is not monotonic in the same way. On the synthetic benchmark, the lowest generation errors occur in the small-noise regime $\sigma = 0.05$–0.1, where the observed errors are approximately 3.95%–4.71%. At the common operating value $\sigma = 0.1$, SOLA obtains the $4.27 \pm 1.19\%$ generation error reported in the main experiments. For larger perturbations, generation degrades progressively, reaching $48.13 \pm 2.38\%$ at $\sigma = 1.0$.

Frey Faces exhibits a similar intermediate-noise regime. At the common operating value $\sigma = 0.1$, SOLA obtains a reconstruction MSE of 0.02 ± 0.0 and an FID of 27.42 ± 1.51, matching the values reported in the main paper. Generation quality subsequently deteriorates as σ increases, reaching 56.96 ± 3.83 at $\sigma = 1.0$.

These results illustrate a trade-off in the strength of stochastic mixing. Small perturbations are compatible with strong generative behavior without substantially disrupting reconstruction, whereas large values of σ increasingly perturb the latent representation presented to the decoder and eventually degrade both reconstruction and generation. Importantly, the sensitivity study should not be interpreted as a dataset-specific tuning procedure: $\sigma = 0.1$ was kept fixed throughout all main SOLA experiments in the paper. The sweep is reported only to characterize the behavior of the method around this common operating value.

The results also show that the same fixed value lies in a favorable regime for both problems considered here. On Frey Faces, $\sigma = 0.1$ gives the lowest FID in the sweep, while on the synthetic problem it lies within the small-noise regime associated with the lowest observed generation errors. This provides empirical evidence that the fixed choice used throughout the paper is not unusually sensitive, while avoiding the problem-dependent loss-weight tuning required by the VAE comparisons.

A.13 Computational Cost and Training-Time Overhead

SOLA introduces a batch-wise SVD during training relative to autoencoder variants that operate directly on their encoded representation. We therefore consider computational cost from two complementary perspectives. First, we compare SOLA with the matched G-only (wide) model in order to

isolate the additional cost associated with the SVD-based geometry. Second, following the standard-VAE comparison used throughout the paper, we report the total training time of SOLA relative to the standard VAE on CelebA. The latter comparison captures the practical end-to-end cost difference between the two methods, including their different latent architectures.

Importantly, the batch-wise decomposition is required only during training. After training, deterministic inference and generation use the fixed global quantities obtained from the retained training batches, and no new batch-wise SVD is required.

For an encoded batch

$$Y \in \mathbb{R}^{L\times B},$$

a dense reduced SVD has computational complexity

$$\mathcal{O}\left(LB\min(L,B)\right). \tag{166}$$

The cost of the decomposition therefore grows with both the encoded width L and the batch size B. After rank truncation, operations involving the retained intrinsic representation additionally scale with the retained rank K. This scaling constitutes a practical limitation of the method when both the encoded representation and training batch are large.

Isolated overhead of the SVD-based geometry. To isolate the overhead associated with the SVD-based representation as directly as possible, we compare SOLA against the G-only (wide) variant from the component ablation. The two models use the same wide encoder and decoder, the same stochastic mixing mechanism, optimizer, training protocol, and number of trainable parameters. The principal architectural difference is therefore the construction and use of the rank-K SVD representation in SOLA.

Table 17 reports the measured training times for this controlled comparison. These timings include the common operations performed by the corresponding experimental pipeline and should therefore be interpreted as practical training-time differences rather than as isolated timings of the SVD kernel itself.

Table 17: Measured training-time overhead of SOLA relative to the matched G-only (wide) control. Values are mean $\pm$ sample standard deviation over five seeds for the synthetic benchmark and three seeds for CelebA. The overhead is computed from the corresponding mean training times.

Dataset	L	B	K	G-only (s)	SOLA (s)	Overhead
Synthetic Gaussian	200	32	2	64.58 ± 2.72	69.00 ± 2.82	6.8%
CelebA	512	576	186	327.69 ± 0.35	448.34 ± 2.60	36.8%

On the synthetic benchmark, introducing the SVD-based representation increases the mean training time from 64.58 to 69.00 seconds, corresponding to an overhead of 6.8%. In this setting, the encoded batch matrices are relatively small, with $L = 200$ and $B = 32$, and only $K = 2$ directions are retained.

CelebA represents the most computationally demanding SOLA configuration used in this work. The encoded representation is substantially wider ($L = 512$), the batch size is much larger ($B = 576$), and the retained bottleneck dimension is $K = 186$. In the matched component experiment, SOLA requires 448.34 ± 2.60 seconds compared with 327.69 ± 0.35 seconds for the G-only control, corresponding to a 36.8% increase in training time.

Total training time relative to the standard VAE. The preceding comparison intentionally matches the wide representation and therefore isolates the cost of introducing the SVD. It does not measure the full practical difference between SOLA and the standard VAE used in the main experiments. In particular, the CelebA VAE operates directly with a $K = 186$-dimensional latent representation, whereas SOLA first produces a wider $L = 512$-dimensional representation before applying rank reduction.

Table 18 reports the total training times measured in the main CelebA experiment. Both methods are evaluated over the same three seeds and training protocol. The reported times correspond to model training and are distinct from the matched G-only experiment above.

Table 18: Total training-time comparison between the standard VAE and SOLA on CelebA. The VAE uses latent dimension $K = 186$, while SOLA uses a wide encoded dimension $L = 512$ followed by a rank-$K = 186$ bottleneck. Mean training times are reported over three seeds.

Method	Mean training time (s)	Relative to VAE
VAE ($\beta = 1$)	287.06	–
SOLA	424.42	+47.9%

Thus, in the main CelebA experiment, SOLA requires approximately 47.9% more training time than the standard $\beta = 1$ VAE. This difference is larger than the isolated SVD-overhead comparison because it includes both the batch-wise decomposition and the wider pre-reduction latent representation used by SOLA. The two comparisons therefore answer different questions: the G-only comparison estimates the incremental cost of the SVD-based geometry under matched architecture, while the VAE comparison reflects the total training cost of the methods as used in the main experiments.

The CelebA results should be viewed as a demanding case within the experiments of this paper rather than as representative of all SOLA configurations. The polynomial dependence of the SVD cost on L and B can become significant for substantially wider representations or larger training batches. This limits the scalability of the present construction to very large latent widths and batch sizes and motivates more efficient or approximate low-rank decompositions as a possible direction for future work.

Finally, the additional decomposition cost is restricted to model fitting. During deterministic inference, an input is encoded and projected directly using the fixed global basis,

$$Y_f = U_f U_f^\top f_\theta(X), \tag{167}$$

without recomputing a batch-wise SVD. Random generation similarly uses the fixed post-training quantities U_f and Σ_{gen}. Consequently, the batch-wise SVD cost affects training but does not impose the same decomposition overhead during deployment-time inference or generation.

A.14 Experimental Setup and Network Architectures

This section provides the implementation details used in the main experiments. Within each dataset, SOLA and the corresponding VAE baselines use the same input preprocessing, convolutional encoder and decoder backbone, optimizer, training budget, and mini-batch protocol. The models differ only in the latent construction required by each method. The VAE predicts the mean and log-variance of a K-dimensional latent variable, whereas SOLA first produces a wider L-dimensional representation and subsequently applies the rank-K stochastic bottleneck described in Section 2.

Unless otherwise stated, all SOLA experiments use the signed stochastic mixing

$$G = I_B + \frac{\sigma}{\sqrt{B}} E, \qquad E_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1), \tag{168}$$

with the same fixed value

$$\sigma = 0.1$$

for every dataset. The sensitivity analysis in Appendix A.12 varies this quantity separately and does not affect the configurations used for the main results.

Shared 64×64 architecture. The synthetic Gaussian, dSprites, Shapes3D, and Frey Faces experiments use the same convolutional architecture. Only the number of input and output channels depends on the dataset: $C = 1$ for the synthetic Gaussian, dSprites, and Frey Faces, and $C = 3$ for Shapes3D.

The encoder consists of four convolutional layers,

$$C \to 32 \to 64 \to 128 \to 256, \tag{169}$$

each using a 4×4 kernel, stride 2, and padding 1, followed by a ReLU activation. Starting from a 64×64 input, the spatial resolution evolves as

$$64 \to 32 \to 16 \to 8 \to 4.$$

The resulting $256 \times 4 \times 4$ tensor is flattened and passed through the shared fully connected transformation

$$4096 \to 128,$$

followed by a ReLU.

For the VAE, the resulting 128-dimensional representation is mapped to two separate K-dimensional heads predicting the posterior mean and log-variance. A stochastic latent vector is then obtained using the standard reparameterization trick.

For SOLA, the same 128-dimensional representation is instead mapped to a wide latent representation

$$Y \in \mathbb{R}^L, \qquad L = 200,$$

before the rank-K SVD bottleneck and stochastic mixing are applied. The rank-reduced coordinates are subsequently mapped back to the L-dimensional ambient latent space before decoding.

The decoder mirrors the encoder. Its fully connected part is

$$d_{\text{lat}} \to 128 \to 4096,$$

with ReLU activations, where

$$d_{\text{lat}} = \begin{cases} K, & \text{VAE}, \\ L, & \text{SOLA}. \end{cases}$$

The output is reshaped to $256 \times 4 \times 4$ and passed through four transposed convolutions,

$$256 \to 128 \to 64 \to 32 \to C, \tag{170}$$

using 4×4 kernels, stride 2, and padding 1. ReLU activations follow the first three transposed convolutions, no output activation is applied after the final layer.

Thus, within a given dataset, the convolutional encoder and decoder are shared between the VAE and SOLA. Their architectural differences are restricted to those required by their respective latent representations.

Optimization protocol. For the synthetic Gaussian, dSprites, Shapes3D, and Frey Faces experiments, all models are trained for 200 epochs using Adam with learning rate 10^{-3} and mini-batch size

$$B = 32.$$

Training data are shuffled at every epoch. Reconstruction is measured using mean-squared error. SOLA minimizes only the reconstruction loss,

$$\mathcal{L}_{\text{SOLA}} = \mathcal{L}_{\text{rec}}, \tag{171}$$

whereas the standard VAE minimizes

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \beta D_{\text{KL}}. \tag{172}$$

The effective latent dimensionality of the VAE is matched to the retained rank K of SOLA on every dataset. Consequently, differences in performance are not obtained by assigning SOLA a smaller effective bottleneck than the corresponding VAE.

For SVD-based models, an incomplete final batch is retained whenever its size is at least K. It is discarded only if its size is smaller than the requested rank, for which a rank-K decomposition would not be defined.

The main quantitative experiments use five independent random seeds for the synthetic Gaussian benchmark and Frey Faces, and three seeds for dSprites, Shapes3D, CelebA, and CIFAR-10.

Table 19 summarizes the principal training configuration of each dataset.

Table 19: Main experimental configurations. L denotes the wide SOLA latent dimension and K the effective latent bottleneck dimension. The VAE latent dimension is matched to K. SOLA uses the same fixed $\sigma = 0.1$ in every experiment.

Dataset	N_{tr}	N_{val}	N_{te}	L	K	B	Epochs
Synthetic Gaussian	400	–	100	200	2	32	200
dSprites	589824	–	147456	200	5	32	200
Shapes3D	384000	–	96000	200	10	32	200
Frey Faces	1665	–	300	200	10	32	200
CelebA	162770	19867	19962	512	186	576	10
CIFAR-10	45000	5000	10000	256	64	128	50

Synthetic Gaussian benchmark. The synthetic benchmark consists of single-channel 64×64 images containing isotropic Gaussian bumps,

$$I(x, y) = \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2(0.2)^2}\right), \tag{173}$$

evaluated over the spatial domain $[-1, 1]^2$. The center (x_0, y_0) supplies two underlying degrees of freedom, motivating the choice

$$K = 2.$$

The main experiment uses 400 training images and 100 test images. Training centers are constructed from a regular Cartesian grid over $[-0.5, 0.5]^2$, whereas test centers are sampled independently from the same domain. The training and test sets are each normalized using their own empirical mean and standard deviation.

Both models use the shared architecture above. SOLA uses $L = 200$, $K = 2$, $B = 32$, and $\sigma = 0.1$, while the corresponding VAE uses a two-dimensional latent variable.

For the KL-weight sensitivity analysis, the tested VAE values are

$$\beta \in \left\{0,\, 10^{-6},\, 10^{-5},\, 10^{-4},\, 10^{-3},\, 10^{-2},\, 10^{-1},\, 1,\, 10,\, 64\right\}.$$

All values use exactly the same architecture, optimizer, and training budget.

dSprites. dSprites consists of binary 64×64 images and is therefore processed using the single-channel version of the shared architecture. The main split contains 589824 training samples and 147456 test samples.

SOLA uses

$$L = 200, \qquad K = 5, \qquad B = 32, \qquad \sigma = 0.1,$$

and is trained for 200 epochs using Adam with learning rate 10^{-3}. The corresponding VAE uses a matched five-dimensional latent variable and the same encoder, decoder, batch size, optimizer, and training duration.

The best-performing VAE configuration among the evaluated KL weights is obtained at

$$\beta^\star = 10^{-1}.$$

Shapes3D. Shapes3D contains RGB 64×64 images and therefore uses the three-channel version of the shared convolutional architecture. The main split contains 384000 training samples and 96000 test samples.

SOLA uses

$$L = 200, \qquad K = 10, \qquad B = 32, \qquad \sigma = 0.1,$$

and is trained for 200 epochs with Adam at learning rate 10^{-3}. The corresponding VAE uses the same convolutional backbone and a matched 10-dimensional latent variable.

The choice $K = 10$ is examined explicitly in Appendix A.10. As shown by the retained-rank sensitivity experiment, reconstruction changes little beyond this regime, while the generation FID is lowest at $K = 10$ among the tested ranks. The best-performing VAE configuration in the KL-weight experiment uses

$$\beta^\star = 10^{-3}.$$

Frey Faces. The Frey Faces dataset contains 1965 grayscale face images of original size 28×20. Each image is resized directly to 64×64 using bilinear interpolation. A fixed random permutation with seed 1234 is used to construct the dataset split: 300 images are reserved for testing and the remaining 1665 images are used for training. The mean and standard deviation used for standardization are computed exclusively from the training set.

The main configuration uses

$$L = 200, \qquad K = 10, \qquad B = 32, \qquad \sigma = 0.1,$$

with 200 epochs of Adam optimization at learning rate 10^{-3}. The VAE uses a matched 10-dimensional latent variable. The retained-rank sensitivity study in Appendix A.10 further examines the choice of K.

The VAE KL-weight sweep uses the same grid as the synthetic Gaussian experiment, with the best generation performance obtained at

$$\beta^\star = 10^{-3}.$$

CelebA. CelebA is the only 64×64 benchmark for which we use a substantially larger model configuration. We use the official split containing 162770 training, 19867 validation, and 19962 test images. Each RGB image is center-cropped to a square and resized to 64×64 using bilinear interpolation. Per-channel mean and standard deviation are computed using only the training split and then applied to all splits.

The convolutional encoder retains the same structure as in the other image experiments,

$$3 \to 32 \to 64 \to 128 \to 256,$$

with 4×4 kernels, stride 2, padding 1, and ReLU activations. After flattening the $256 \times 4 \times 4$ feature map, however, the shared fully connected width is increased from 128 to 512:

$$4096 \to 512.$$

For SOLA, this representation is mapped to a wide latent space of dimension

$$L = 512,$$

and a rank

$$K = 186$$

is retained. The decoder correspondingly uses

$$d_{\mathrm{lat}} \to 512 \to 4096$$

before the same sequence of four transposed convolutions used by the other models. The VAE uses the same convolutional backbone and a matched 186-dimensional latent variable.

CelebA models are trained for 10 epochs with batch size

$$B = 576$$

using Adam with learning rate 10^{-3}. Gradient norms are clipped at 5 for numerical stability. SOLA again uses the unchanged stochastic scale

$$\sigma = 0.1.$$

For the VAE sensitivity experiment, we evaluate

$$\beta \in \left\{10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\right\},$$

with the best-performing configuration obtained at

$$\beta^\star = 10^{-4}.$$

Validation data are used for this model-selection procedure, while the final reported reconstruction and generation quantities are evaluated on the held-out test split.

CIFAR-10. CIFAR-10 is included as a broader natural-image benchmark with diverse, unaligned object classes. We use the native 32×32 RGB images. From the official 50000-image training set, a fixed split of 45000 images is used for training and 5000 for validation, while the official 10000 test images are kept unchanged for final evaluation.

The main SOLA configuration uses

$$L = 256, \qquad K = 64, \qquad B = 128, \qquad \sigma = 0.1,$$

and is trained for 50 epochs. The corresponding VAE uses a matched 64-dimensional latent variable. Final quantitative results are reported over three independent seeds.

The VAE KL weight is selected using validation FID. The sensitivity stage uses a 20000-sample subset of the training split and 30 training epochs, after which the selected value is used for the final full-training runs. The best-performing configuration is

$$\beta^\star = 10^{-4}.$$

The held-out test split is used only for the final reconstruction and generation evaluation.

Controlled comparison between SOLA and the VAE. Within each benchmark, SOLA and the VAE use the same data split, preprocessing, convolutional backbone, optimizer, learning rate, mini-batch size, and training duration. Their effective bottleneck dimensions are also matched: a SOLA model with retained rank K is compared with a VAE whose latent dimension is K.

The differences between the two architectures are therefore those inherent to the methods. The VAE predicts a mean and variance for each observation and optimizes a weighted reconstruction–KL objective. SOLA instead constructs a wide encoded representation, extracts a rank-K population-dependent geometry through the SVD, applies structured stochastic mixing, and optimizes the reconstruction loss alone. Across all datasets, no dataset-specific tuning of the SOLA noise amplitude is performed.

A.15 Generative Evaluation Quantities

We evaluate random generation differently for the image datasets and for the synthetic Gaussian benchmark. For the image datasets, we use the Fréchet Inception Distance (FID) (Heusel et al., 2017), whereas the synthetic benchmark admits a direct error measure because the analytical form of every valid sample is known.

Fréchet Inception Distance. Let μ_r, C_r and μ_g, C_g denote the empirical means and covariance matrices of the Inception features extracted from the real and generated samples, respectively. The FID is

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(C_r + C_g - 2(C_r C_g)^{1/2}\right). \tag{174}$$

Lower values indicate greater similarity between the real and generated distributions.

In our implementation, FID is computed using the 2048-dimensional Inception representation provided by the TorchMetrics implementation of FID. Images stored in normalized form are first mapped back to their original intensity range, clipped to $[0, 1]$, and converted to unsigned 8-bit images in $[0, 255]$. For grayscale datasets, the single image channel is replicated three times before being passed to the Inception network. Unless otherwise stated, the complete training set is used as the real reference and an equally sized set is sampled independently from each generative model. In the CelebA component ablation of Appendix A.8, FID is computed from 10,000 real test images and 10,000 generated images per seed. Comparisons within that ablation therefore use the same fixed FID sample count.

Synthetic Gaussian generation error. The synthetic benchmark consists of unit-amplitude two-dimensional Gaussian fields defined on a 64×64 grid over $[-1, 1]^2$,

$$I(x, y; x_0, y_0) = \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{2\sigma_g^2}\right), \qquad \sigma_g = 0.2, \tag{175}$$

where only the center (x_0, y_0) varies between samples. The training centers lie in $[-0.5, 0.5]^2$. Since the analytical form of a valid sample is therefore known, generation quality can be measured

directly by determining how closely each generated field can be represented by a Gaussian from this family.

For each generated image $\widetilde{I}$, we fit only the center $(\hat{x}_0, \hat{y}_0)$, while keeping the amplitude and width fixed to the values used to construct the dataset. The fit minimizes the pixel-wise least-squares residual

$$(\hat{x}_0, \hat{y}_0) = \arg \min_{(x_0,y_0)\in[-1,1]^2} \left\| \widetilde{I} - I_{\mathrm{norm}}(\,\cdot\,;\, x_0, y_0) \right\|_2^2, \tag{176}$$

where I_{norm} denotes the analytical Gaussian transformed using the training-set normalization associated with the generative model. The optimization is initialized at the spatial location of the maximum value of the generated image.

The generation error of an individual sample is then defined as the relative fitting residual

$$e_{\mathrm{gen}} = 100 \frac{\left\| \widetilde{I} - I_{\mathrm{norm}}(\,\cdot\,; \hat{x}_0, \hat{y}_0) \right\|_2}{\| I_{\mathrm{norm}}(\,\cdot\,; \hat{x}_0, \hat{y}_0) \|_2}, \tag{177}$$

and is therefore expressed as a percentage. A value of zero corresponds to an exact member of the target Gaussian family, whereas larger values indicate increasing deviations in the generated field shape. For each trained model, the reported generation error is obtained by averaging this quantity over 250 independently generated samples. The procedure is repeated across training seeds, and the values reported in the experiments are obtained from the corresponding per-seed averages.

A.16 Latent-Space Visualization on the Synthetic Gaussian Benchmark

The synthetic Gaussian benchmark provides a particularly interpretable setting for inspecting the learned latent representation. Each sample is completely determined by the two coordinates (x_0, y_0) of the Gaussian center. Accordingly, we impose a two-dimensional bottleneck for both models, allowing their learned representations to be visualized directly in the plane.

The quantities visualized are different because the two models parameterize their bottlenecks differently. For the standard VAE, the encoder defines the sample-wise approximate posterior

$$q_\theta(z \mid x_i) = \mathcal{N}\left(\mu_i, \mathrm{diag}(\sigma_i^2)\right), \tag{178}$$

and we therefore plot the two components of its posterior mean, $(\mu_{i,1}, \mu_{i,2})$. For SOLA, the intrinsic bottleneck coordinates are instead given by the columns of

$$A_K = \Sigma_K V_K^\top = [a_1, \ldots, a_B], \tag{179}$$

so the corresponding visualization uses $(a_{i,1}, a_{i,2})$. The different symbols in Figure 9 therefore reflect the different latent parameterizations of the two models rather than merely different axis names.

The VAE posterior means exhibit severe contraction toward the origin. Both mean coordinates remain extremely close to zero. In the displayed experiment, $\mu_{i,1}$ varies only on the order of 10^{-2}, and $\mu_{i,2}$ is similarly confined to approximately a few 10^{-2}. Consequently, despite the underlying data being controlled by two independent continuous variables, the encoder maps the posterior means into a very small neighborhood of the prior mean. This behavior is consistent with the collapse observed in the generation results, but the posterior means alone do not prove complete posterior collapse because the posterior variances are not shown here. What the visualization establishes directly is that sample-dependent variation carried by the mean is strongly compressed under the default $\beta = 1$ VAE objective.

This contraction is particularly clear when compared with SOLA. Its two bottleneck coordinates span an extended two-dimensional region, with values of order one rather than being concentrated near the origin. More importantly, the representation retains a regular organization of the underlying data. The points in both panels are colored according to the true horizontal coordinate x_0 of the Gaussian center. In the SOLA representation, this color varies smoothly across the latent space, from one side of the representation to the other. Samples having similar physical locations are therefore mapped to neighboring bottleneck coordinates, and the ordering of the generative factor is preserved clearly.

Some dependence on x_0 can still be observed in the VAE posterior means, but it is compressed into the small contracted region around zero and follows a strongly folded geometry. Thus, the distinction is not merely one of absolute latent scale: the VAE uses only a very restricted neighborhood of its available latent space and represents the physical factor through a distorted, compressed organization, whereas SOLA maintains a broad and smoothly ordered two-dimensional representation. This difference is consistent with the substantially lower random-generation error obtained by SOLA than by the standard $\beta = 1$ VAE on this benchmark.

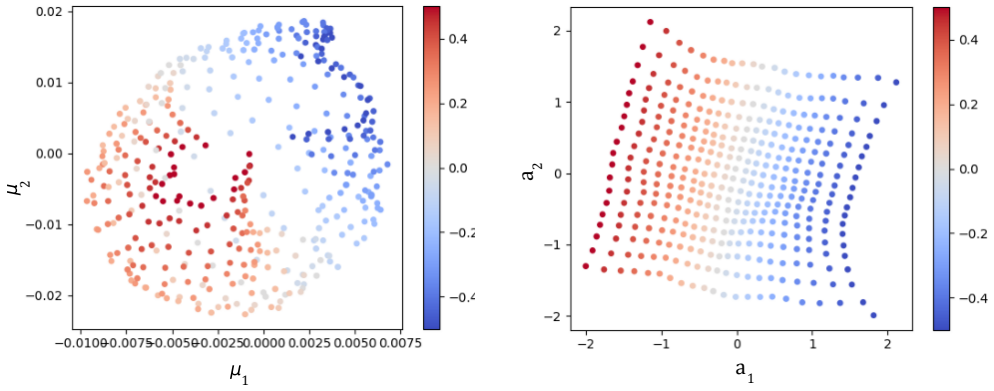


Figure 9: Learned latent representations on the synthetic Gaussian benchmark with bottleneck dimension $K = 2$. Left: VAE posterior means $(\mu_{i,1}, \mu_{i,2})$. Right: SOLA intrinsic bottleneck coordinates $(a_{i,1}, a_{i,2})$. Points are colored by the ground-truth horizontal coordinate x_0 of the Gaussian center. The VAE posterior means are strongly concentrated around the prior mean zero, with both dimensions varying only on the order of 10^{-2}, showing severe contraction of the posterior-mean representation. SOLA instead occupies an extended two-dimensional region and exhibits a smooth ordering with respect to the underlying physical coordinate.

A.17 Implementation and Hyperparameters of Alternative Baselines

This section documents the implementation details and hyperparameter configurations of the alternative methods evaluated in Section 3.2.

Our objective is to compare SOLA against representative implementations of established generative autoencoders and alternative VAE training strategies, rather than to perform exhaustive hyperparameter optimization for every competing method.

We therefore use predefined configurations for methods with established training schedules or architectural settings. For methods involving additional regularization coefficients, limited validation-based selection is performed where indicated.

These experiments should not be interpreted as an exhaustive hyperparameter comparison. In particular, SOLA uses the same fixed stochastic scale $\sigma = 0.1$ across all datasets, without dataset-specific selection.

The alternative baselines are evaluated on the synthetic Gaussian benchmark, Frey Faces, and CelebA. The corresponding dataset configurations, preprocessing procedures, network architectures, and training budgets are documented in Appendix A.14.

Baseline configurations. Table 20 summarizes the principal implementation choices and hyperparameter-selection procedures used for the alternative baselines.

Selected hyperparameters. Table 21 reports the regularization coefficients used for the three comparison datasets.

Table 20: Implementation settings and hyperparameter-selection procedures for the alternative baselines. Fixed configurations are used without hyperparameter searches, whereas selected regularization coefficients are determined using validation data.

Method	Implementation and principal settings	Hyperparameter selection
Monotonic KL annealing	The KL weight increases linearly from 0 to 1 during the first half of training and subsequently remains equal to 1.	Fixed schedule. No hyperparameter search.
Cyclical KL annealing	Four training cycles. Within each cycle, the KL weight increases linearly from 0 to 1 during the first half of the cycle and remains equal to 1 during the second half.	Fixed schedule. No hyperparameter search.
Free Bits	The batch-averaged KL divergence of each latent dimension is subject to a minimum threshold λ_{FB}, expressed in nats per dimension.	Validation grid: $\{0.01, 0.05, 0.1, 0.25, 0.5, 1, 2\}$.
σ-VAE	The decoder noise scale is estimated at each training step from the batch reconstruction MSE, with a soft lower bound of -6 on the log noise scale. The KL weight is fixed at 1.	No hyperparameter search.
Capacity-controlled β-VAE	The target KL capacity increases linearly from 0 to $C_{\max}$ during training. The capacity penalty is controlled by γ.	Limited validation-based selection of γ and $C_{\max}$.
RAE-L2	Deterministic encoder with latent ℓ_2 regularization and decoder weight regularization. Generation uses a fitted Gaussian mixture model.	Dataset-specific coefficients reported in Table 21.
WAE-MMD	Stochastic, reparameterized encoder. The KL term is replaced by an IMQ-kernel MMD penalty matching the aggregated posterior to $\mathcal{N}(0, I)$.	Validation grid: $\{1, 10, 50, 100, 500, 1000\}$.
VQ-VAE	Codebook size 512, embedding dimension 64, latent grid 4×4, and commitment coefficient 0.25.	Fixed configuration. No hyperparameter search.

Table 21: Hyperparameters used for the alternative baselines. The Gaussian column denotes the synthetic Gaussian benchmark.

Method and parameter	Gaussian	Frey Faces	CelebA
Free Bits: λ_{FB}	2.0	2.0	1.0
Capacity-controlled β-VAE: γ	0.1	0.01	0.01
Capacity-controlled β-VAE: $C_{\max}$	5	10	93
RAE-L2: β_z	10^{-4}	10^{-4}	10^{-4}
RAE-L2: λ_{dec}	10^{-6}	10^{-5}	10^{-6}
WAE-MMD: λ_{MMD}	1.0	1.0	1.0

Free Bits. For Free Bits, the conventional VAE objective is modified by imposing a minimum contribution for the batch-averaged KL divergence of each latent dimension:

$$\mathcal{L}_{\mathrm{FB}} = \mathcal{L}_{\mathrm{rec}} + \sum_{d=1}^{K} \max\left(\overline{D}_{\mathrm{KL},d}, \lambda_{\mathrm{FB}}\right), \tag{180}$$

where $\overline{D}_{\mathrm{KL},d}$ denotes the batch-averaged KL divergence associated with latent dimension d.

The threshold is selected from

$$\lambda_{\mathrm{FB}} \in \{0.01, 0.05, 0.1, 0.25, 0.5, 1, 2\}.$$

Capacity-controlled β-VAE. For capacity-controlled β-VAE, the target KL capacity increases linearly from zero to $C_{\max}$ during training.

The penalty coefficient is selected from

$$\gamma \in \{0.01, 0.1, 0.25\}.$$

The candidate capacity limits depend on the dataset:

$$C_{\max} \in \begin{cases} \{1, 2, 5\}, & \text{Synthetic Gaussian,} \\ \{5, 10, 25\}, & \text{Frey Faces,} \\ \{93, 186, 465\}, & \text{CelebA.} \end{cases}$$

The selected configurations are reported in Table 21.

WAE-MMD. Our WAE-MMD implementation uses a stochastic encoder that predicts the mean and log-variance of a Gaussian approximate posterior. Latent vectors are obtained using the standard reparameterization trick.

Unlike the conventional VAE, the sample-wise KL term is omitted entirely. Instead, the aggregated latent distribution $q(z)$ is regularized toward the standard Gaussian prior $p(z) = \mathcal{N}(0, I)$ through maximum mean discrepancy.

The resulting objective is

$$\mathcal{L}_{\mathrm{WAE}} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{MMD}} \,\mathrm{MMD}\left(q(z), p(z)\right). \tag{181}$$

We use the inverse multiquadratic kernel

$$k(z, z') = \frac{C}{C + \|z - z'\|_2^2}, \qquad C = 2D, \tag{182}$$

where D denotes the latent dimensionality.

The MMD penalty is calculated using an unbiased U-statistic estimator. The regularization coefficient is selected from

$$\lambda_{\mathrm{MMD}} \in \{1, 10, 50, 100, 500, 1000\}.$$

Validation selects $\lambda_{\mathrm{MMD}} = 1$ for all three datasets.

Generation is performed by directly sampling from $\mathcal{N}(0, I)$ and decoding the resulting latent vectors.

RAE-L2. The RAE-L2 baseline employs a deterministic encoder together with explicit regularization of the latent representation and decoder parameters.

Its training objective is

$$\mathcal{L}_{\mathrm{RAE}} = \mathcal{L}_{\mathrm{rec}} + \frac{\beta_z}{2}\mathbb{E}\left[\|z\|_2^2\right] + \lambda_{\mathrm{dec}}\|\theta_{\mathrm{dec}}\|_2^2, \tag{183}$$

where β_z controls latent regularization and λ_{dec} controls decoder weight regularization.

The configurations used for the synthetic Gaussian benchmark, Frey Faces, and CelebA are reported in Table 21.

Since RAE uses a deterministic encoder, its training objective does not directly define a stochastic sampling distribution.

For generation, we fit a Gaussian mixture model to the learned training-set latent representations.

We use a mixture of 10 full-covariance Gaussian components. The fitted distribution is sampled after training, and the resulting latent vectors are passed through the decoder.

Unlike SOLA, this procedure requires a separate latent-density estimation stage to obtain the final generator.

VQ-VAE. The VQ-VAE baseline uses a discrete latent representation with a codebook containing 512 entries, each of embedding dimension 64.

The spatial latent representation is a 4×4 grid, and the commitment-loss coefficient is fixed at

$$\beta_{\text{commit}} = 0.25.$$

For generation, a PixelCNN prior is trained on the discrete code indices extracted from the training set.

The PixelCNN contains 64 hidden channels, one masked type-A convolutional layer, and two masked type-B convolutional layers.

Both VQ-VAE and the PixelCNN prior are optimized using Adam with learning rate

$$\eta = 10^{-3},$$

and are trained for the corresponding dataset's specified number of epochs.

No architecture or hyperparameter sweep is performed for this baseline.

A.18 Qualitative Comparison of Interpolation and Random Generation on the Synthetic Gaussian Benchmark

We provide in this appendix a qualitative comparison of latent interpolation and random generation on the synthetic Gaussian benchmark. The goal is to complement the quantitative results of Section 3 by directly visualizing the types of samples produced by the different models.

Figure 10 shows latent interpolations for three models: a standard VAE with its default setting $\beta = 1$, a tuned VAE with $\beta = 0$, and SOLA. For each model, interpolation is performed between two held-out samples by linearly traversing the corresponding latent representations. The standard VAE produces almost no meaningful variation along the interpolation path, which is consistent with the severe latent contraction observed above. In contrast, the tuned VAE and SOLA both produce structured transitions between Gaussian bumps. SOLA yields a smooth and coherent evolution of the bump position across the interpolation path, showing that the learned latent space remains both organized and usable for controlled traversal.

Figure 11 shows random generations for the same three models. The standard VAE with $\beta = 1$ again produces nearly identical samples with very limited diversity, consistent with the same collapse-like behavior. The tuned VAE already recovers meaningful variability, confirming that problem-specific tuning of β can recover strong generative behavior on this benchmark. SOLA also generates diverse and coherent Gaussian samples without introducing a KL-weight hyperparameter, which is consistent with the quantitative results reported in the main paper.

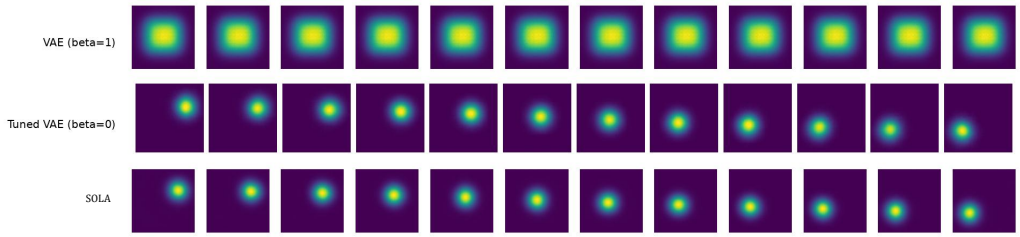


Figure 10: Qualitative comparison of latent interpolation on the synthetic Gaussian benchmark. Each row corresponds to one method: standard VAE with $\beta = 1$, tuned VAE with $\beta = 0$, and SOLA. The standard VAE shows little variation along the interpolation path, consistent with severe latent contraction, whereas the tuned VAE and SOLA produce meaningful and smooth transitions.

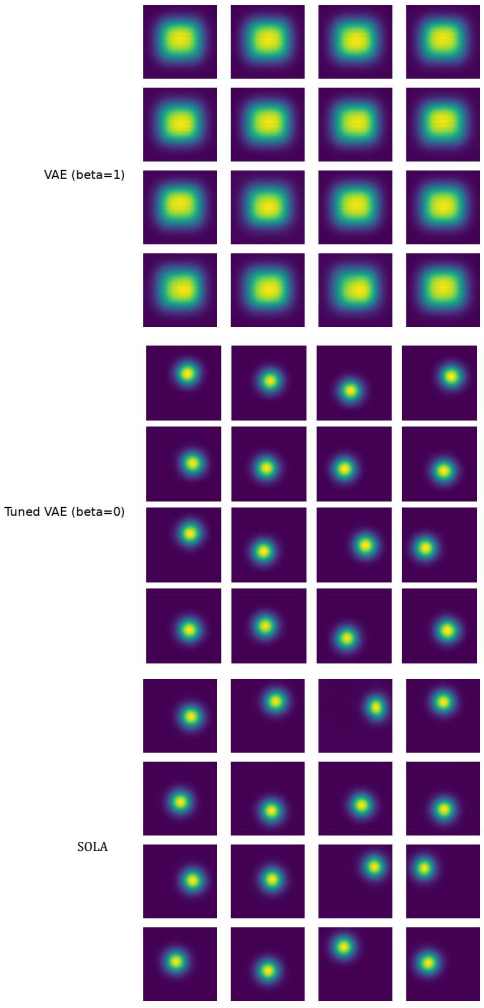


Figure 11: Qualitative comparison of random generation on the synthetic Gaussian benchmark. Each row corresponds to one method: standard VAE with $\beta = 1$, tuned VAE with $\beta = 0$, and SOLA. The standard VAE produces nearly identical samples, while the tuned VAE and SOLA generate diverse Gaussian bumps.

A.19 SAMPLE IMAGES FOR IMAGE BENCHMARKS

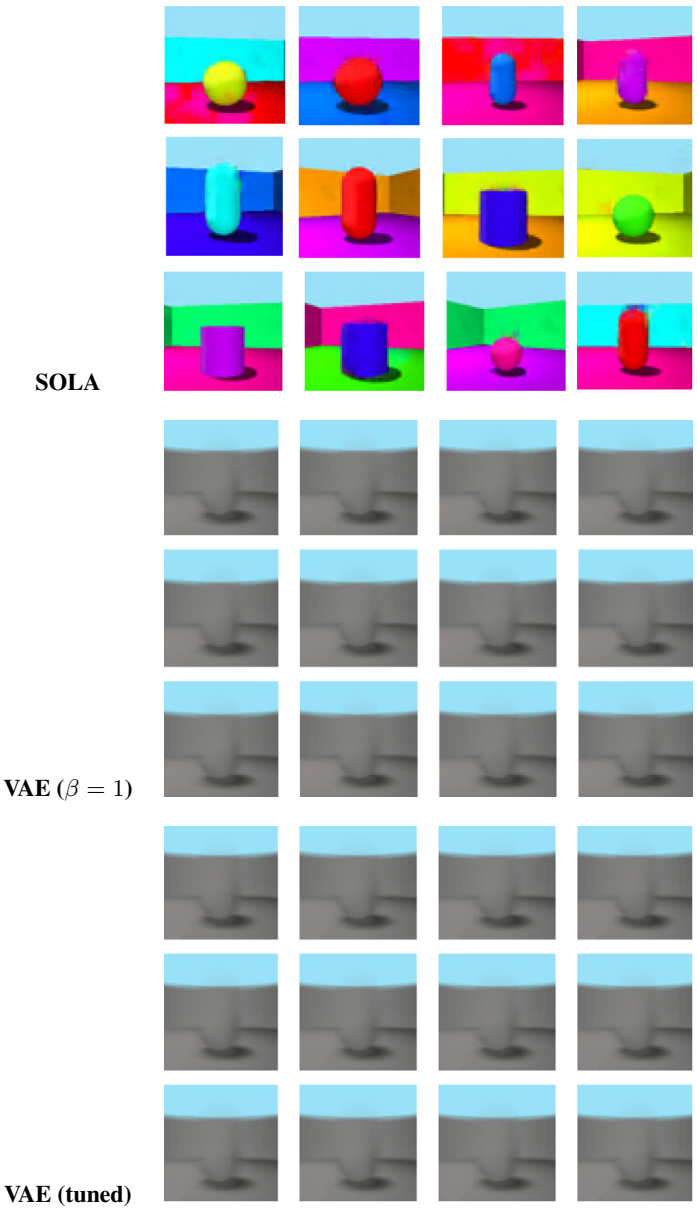


Figure 12: Additional random generations on Shapes3D. Rows correspond to SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-3}$).

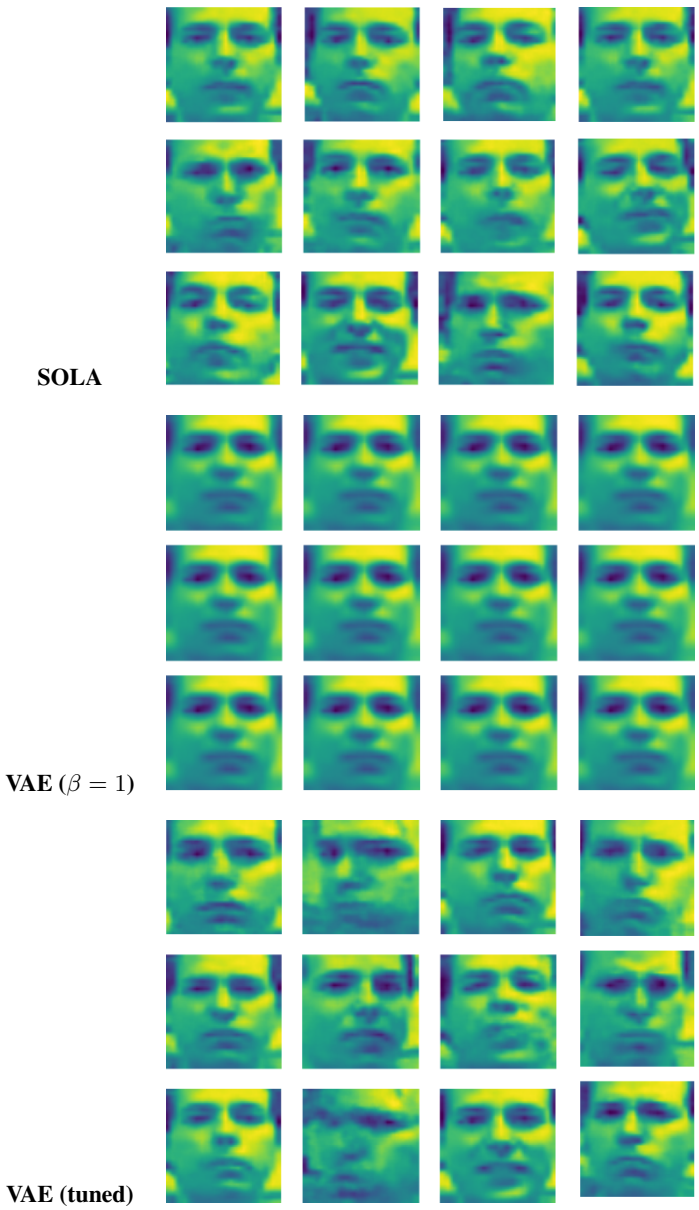


Figure 13: Additional random generations on Frey Faces. Rows correspond to SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-3}$).

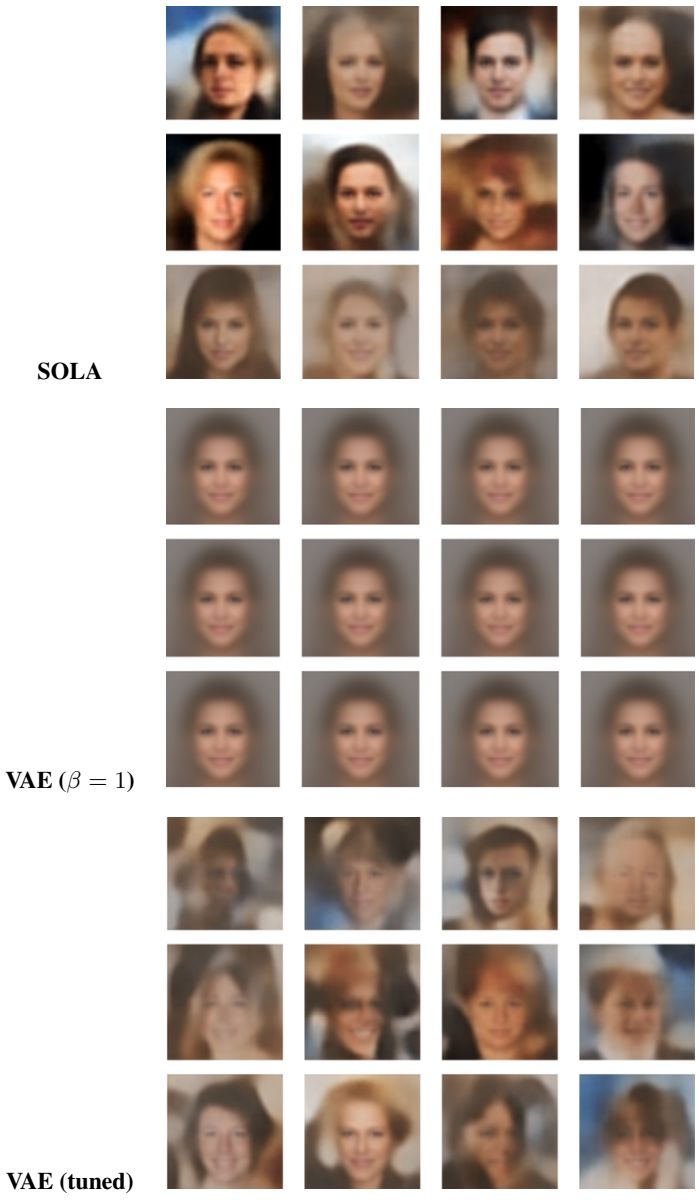


Figure 14: Additional random generations on CelebA. Rows correspond to SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-4}$).

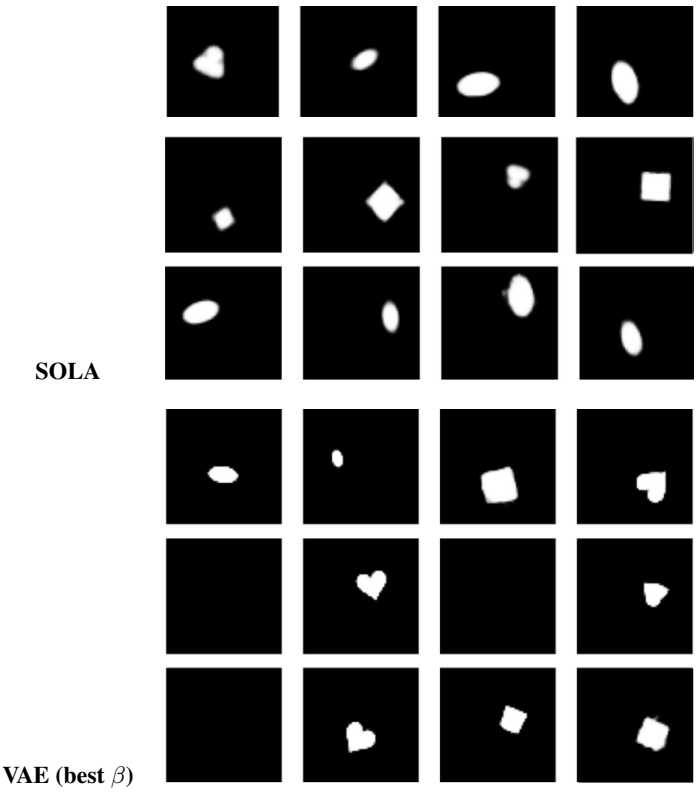


Figure 15: Additional random generations on dSprites. Rows correspond to SOLA and the best-performing VAE configuration. For dSprites, the best VAE setting is $\beta^* = 0.1$, which coincides with the standard VAE setting.

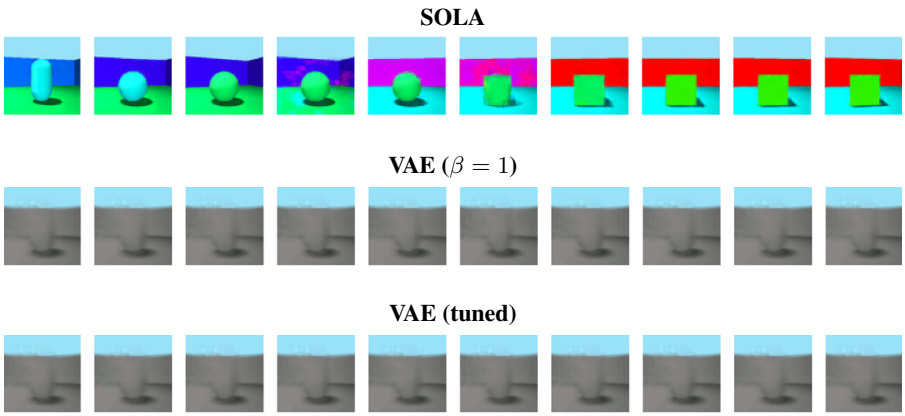


Figure 16: Latent interpolation on Shapes3D for SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-3}$).

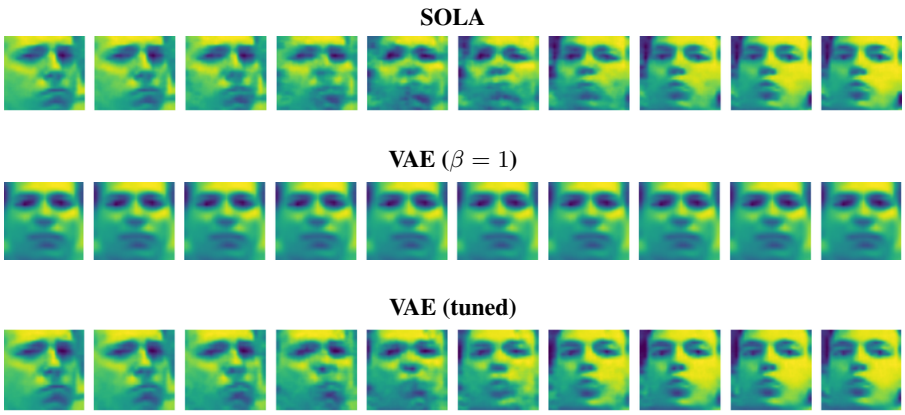


Figure 17: Latent interpolation on Frey Faces for SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-3}$).

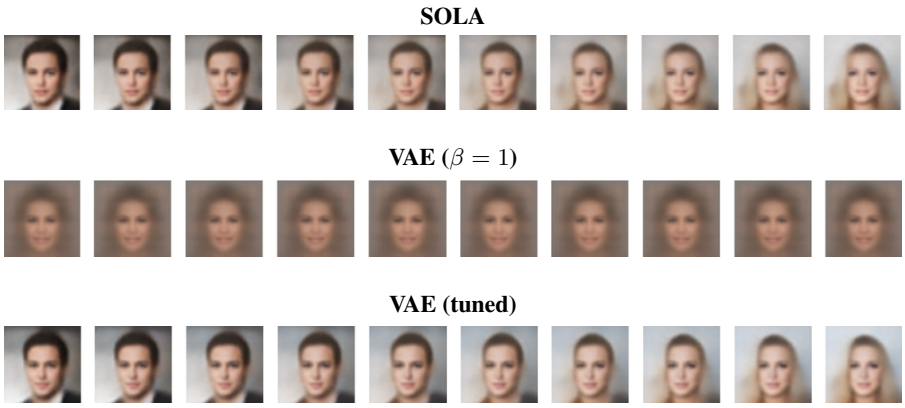


Figure 18: Latent interpolation on CelebA for SOLA, the standard VAE with $\beta = 1$, and the problem-specific tuned VAE ($\beta^* = 10^{-4}$).

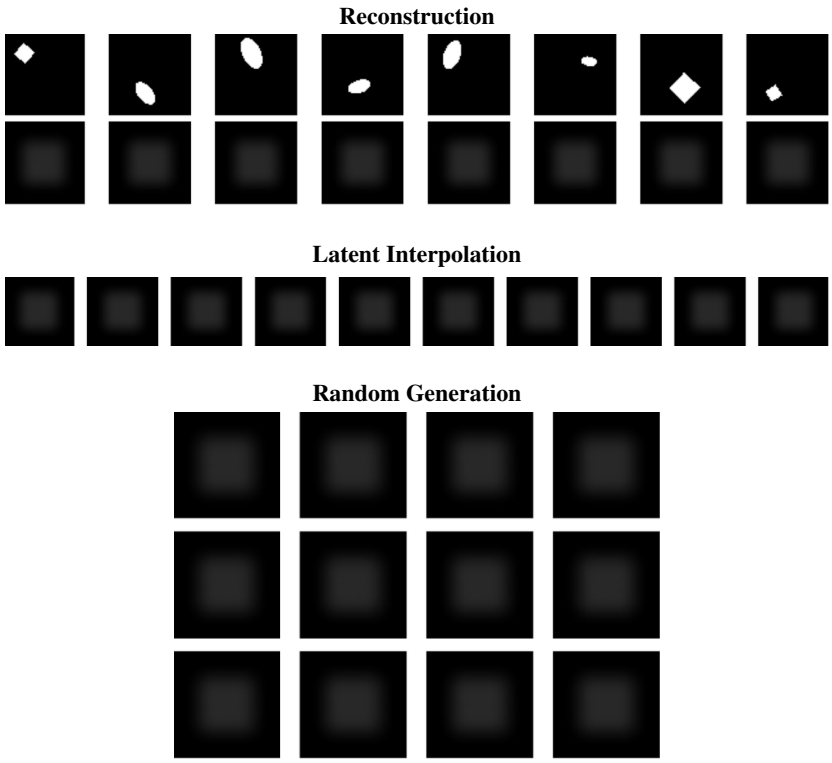


Figure 19: Qualitative example of a failed standard VAE seed on dSprites with $\beta = 1$, showing the corresponding reconstruction, latent interpolation and random generation outputs.