Zero-Variance Gradients for Variational Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

Training deep generative models like Variational Autoencoders (VAEs) is often hindered by the need to backpropagate gradients through the stochastic sampling of their latent variables, a process that inherently introduces estimation variance, which can slow convergence and degrade performance. In this paper, we propose a new perspective that sidesteps this problem, which we call **Silent Gradients**. Instead of improving stochastic estimators, we leverage specific decoder architectures to analytically compute the expected ELBO, yielding a gradient with zero variance. We first provide a theoretical foundation for this method and demonstrate its superiority over existing estimators in a controlled setting with a linear decoder. To generalize our approach for practical use with complex, expressive decoders, we introduce a novel training dynamic that uses the exact, zero-variance gradient to guide the early stages of encoder training before annealing to a standard stochastic estimator. Our experiments show that this technique consistently improves the performance of established baselines, including reparameterization, Gumbel-Softmax, and REINFORCE, across multiple datasets. This work opens a new direction for training generative models by combining the stability of analytical computation with the expressiveness of deep, nonlinear architecture.

1 Introduction

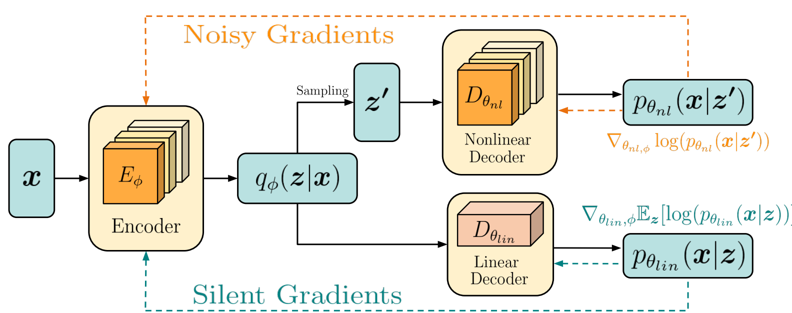


Figure 1: **Illustration of the use of Silent Gradients in training VAEs.** The encoder (E_ϕ) takes input $\boldsymbol{x}$ and infers a latent distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. These parameters are fed directly to the linear decoder (D_{lin}), which computes the analytical reconstruction log-likelihood, yielding a noise-free (Silent) gradient (dashed teal arrow) used to train the encoder. In parallel, samples $\boldsymbol{z'}$ are drawn from the latent distribution and fed to the nonlinear decoder (D_{nl}), which produces a standard, sample-based loss, resulting in a noisy gradient (dashed orange arrow). During training, we can choose to train the encoder solely with the Silent Gradients or combine it with the noisy gradient using an annealing schedule. At inference time, only the trained encoder E_ϕ and nonlinear decoder D_{nl} are used.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Training of neural networks with stochastic components, such as the sampling of latent variables in generative models, often suffers from high variance in gradient estimates. This variance can impede the optimization process, leading to slower convergence and suboptimal model performance. In Variational Autoencoders (VAEs) [9, 18], for instance, gradients must be propagated through a stochastic sampling layer. This has led to the development of several estimation techniques. For continuous latent spaces, the reparameterization trick [9] is commonly used. For discrete spaces, common approaches include the REINFORCE algorithm [25] and the Gumbel-Softmax trick [15, 7]. However, all of these sample-based techniques introduce estimation variance, and in this paper we show that this variance hinders the optimization even in a simple, controlled setting.

In this paper, we propose a fundamentally different perspective on gradient estimation for VAEs. Given the estimation variance introduced by these stochastic gradient estimators, we argue for a different paradigm. Instead of developing more sophisticated techniques to *estimate* the gradient of an expectation, we explore the possibility of first efficiently[1] computing the expectation itself in closed form, and then differentiating the resulting analytical expression. This path, when available, yields a gradient that is computed exactly and therefore has *zero variance* by definition, in terms of the latent variables.

The feasibility of this approach hinges on the decoder architecture. While it is well-known that for a linear function, the expectation of its output can be computed exactly by linearity of expectation, this does not trivially extend to the full reconstruction log-likelihood. We first show that for a Gaussian likelihood with a fixed variance, the expected loss can still be computed in closed form, as a function of the latent distribution rather than the sampled latent variables. We then empirically demonstrate that using this analytic gradient leads to superior performance and faster convergence compared to standard stochastic estimators in this setting. Furthermore, we extend this technique to a more expressive setting where the output variance is also a learnable function of the same latent variables, again providing a zero-variance gradient. This analytic gradient component can then boost the performance of existing standard stochastic gradient estimators.

Finally, to generalize our method for more complex and practical settings, we introduce a novel training dynamic, depicted in Figure 1, that combines our analytic gradient component with standard, expressive nonlinear decoders. By using Silent Gradients to guide the initial training of the encoder before annealing to a conventional estimator, our technique serves as a powerful variance reduction tool that consistently improves the performance of established methods. Our experimental results on the MNIST [5], ImageNet [4], and CIFAR-10 [12] datasets demonstrate a significant and consistent improvement in model performance. This shows that architectural choices that provide exact gradients are a powerful and general strategy for improving the training dynamics of models with stochastic layers.

2 Background

Variational Autoencoders (VAEs) [9] are a class of generative models for learning the probability distribution $p(\boldsymbol{x})$ that underlies a dataset. VAEs introduce a set of latent variables $\boldsymbol{z}$ that are assumed to generate the observed data $\boldsymbol{x}$. The model consists of two components: a prior distribution over the latent space $p(\boldsymbol{z})$ and a conditional likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, defined by the decoder D_θ. The marginal likelihood of the model given the data then is $p_\theta(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[p_\theta(\boldsymbol{x}|\boldsymbol{z})]$.

Since direct maximization of this likelihood is generally intractable, VAEs introduce a variational approximation $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, called the encoder E_ϕ, parameterized by ϕ. Instead of maximizing the log-likelihood directly, one maximizes the Evidence Lower Bound (ELBO) [8]:

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z})). \tag{1}$$

The first term is the expected reconstruction log-likelihood, which encourages the decoder to reconstruct the input $\boldsymbol{x}$ from its latent representation $\boldsymbol{z}$. The second term is the Kullback-Leibler (KL) divergence, which forces the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to be close to the prior $p(\boldsymbol{z})$. Maximizing the ELBO provides a framework to jointly train the encoder and decoder parameters, ϕ and θ, respectively. We include more details about related work in VAEs and gradient estimation methods in Appendix A.

[1] In time linear in the number of latent dimensions.

3 Exact ELBO with Linear Decoder

While the KL divergence term is often analytically tractable by employing the mean-field assumption, in which the approximate posterior factorizes across latent dimensions: $q(\boldsymbol{z}|\boldsymbol{x}) := \prod_i q(z_i|\boldsymbol{x})$, the reconstruction term, $\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$, remains intractable to compute. This difficulty comes from the complexity of the decoder function $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, which is typically a deep neural network. Consequently, estimating the gradient of this reconstruction term with respect to the encoder parameters ϕ poses a challenge since the gradient operator ∇_ϕ cannot be passed inside an expectation that depends on those same parameters.

To resolve this problem, various techniques have been introduced [25, 7, 15, 9]. However, we show that even widely-used estimators like the reparameterization trick can be far from optimal.

Table 1: **Comparison of the total gradient variance with respect to the encoder parameters. Continuous and Discrete refer to the type of latent space.** The variance is measured by repeatedly sampling gradients for a fixed input batch at different training epochs. The results show that existing gradient estimators have substantial variance. In contrast, our method (Silent Gradients) has a true variance of zero as its gradient is computed analytically.

		Epoch		
	Method	10	200	500
Continuous	Silent Gradients	0	0	0
	Reparameterization	1.08×10^5	1.78×10^4	1.37×10^4
Discrete	Silent Gradients	0	0	0
	Gumbel-Softmax	1.10×10^5	2.08×10^5	1.68×10^5
	REINFORCE	2.30×10^8	6.30×10^7	4.00×10^7

Specifically, we compute the average gradient variance of the ELBO for single samples on the MNIST dataset [5] at three different training stages (epochs 10, 200, and 500). This allows us to directly compare the gradient noise introduced by each estimator and observe how it evolves during optimization. As shown in Table 1, the variance of the gradients for standard estimators is substantial. Even for the reparameterization trick, which is considered a low-variance method, the gradient noise is significant and can hinder optimization. This naturally raises the question: how much performance is lost to this estimation variance, and what could be gained if we were able to compute the exact ELBO and its gradients? This motivates our exploration of an analytical approach.

3.1 Analytic gradient

As established, the intractability of the reconstruction term comes from the complexity of the decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, which is typically a deep neural network. This motivates an investigation into specific architectural choices where this analytical bottleneck can be resolved. We show that for a specific model structure, the expectation in the reconstruction term can be computed exactly (i.e., the first term in Eq. 1), bypassing the need for stochastic estimation entirely. Let the data $\boldsymbol{x}$ and latent variable $\boldsymbol{z}$ be column vectors of dimensions k and d, respectively. We consider a generative likelihood $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ that is a Gaussian distribution with a mean produced by a linear decoder and a fixed, scalar variance σ^2:

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \mathbf{W}_\mu \boldsymbol{z}, \sigma^2 I). \tag{2}$$

where the decoder is parameterized by the weight matrix $\mathbf{W}_\mu \in \mathbb{R}^{k\times d}$. For simplicity, we will denote the expectation $\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}$ as $\mathbb{E}$ where the context is clear.

Although this linear setup may seem restrictive, we will show in later sections that this technique forms the basis of a general method applicable to any VAEs. For now, we proceed by substituting this linear decoder structure into the ELBO reconstruction term:

$$\mathbb{E}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] = -\frac{1}{2\sigma^2}\mathbb{E}[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||_2^2] - \frac{1}{2}\log(2\pi\sigma^2). \tag{3}$$

To compute this term exactly, all we need is to find an analytical form for the expectation $\mathbb{E}[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||_2^2]$. We begin by expanding the squared L2 norm:

$$\mathbb{E}[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||_2^2] = \mathbb{E}[(\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z})^T(\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z})] = \mathbb{E}[||\boldsymbol{x}||_2^2 - 2\boldsymbol{x}^T\mathbf{W}_\mu \boldsymbol{z} + ||\mathbf{W}_\mu \boldsymbol{z}||_2^2].$$

By linearity of expectation, and because $\boldsymbol{x}$ and $\mathbf{W}_\mu$ are constants with respect to the expectation over $\boldsymbol{z}$, we simplify:

$$\mathbb{E}[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||_2^2] = ||\boldsymbol{x}||_2^2 - 2\boldsymbol{x}^T\mathbf{W}_\mu \mathbb{E}[\boldsymbol{z}] + \mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2].$$

The challenge lies in resolving the third term $\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2]$ since a naive computation of this expectation would take time $\mathcal{O}(k^2 d)$, which is quadratic w.r.t. the number of $\mathbf{X}$ variables. However, we can reduce the complexity by exploiting the fact that different variables in $\boldsymbol{z}$ are mutually independent due to the mean-field assumption.

We begin by expanding the quadratic term into a double summation $\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2] = \sum_i \sum_j \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \mathbb{E}[z_i z_j]$, where $\mathbf{w}_{\mu,i}$ is the ith column of $\mathbf{W}_\mu$. Using the identity $\mathbb{E}[z_i z_j] = \mathbb{E}[z_i]\mathbb{E}[z_j] + \text{Cov}(z_i, z_j)$, we split this summation into two parts: $\sum_{i,j} \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \mathbb{E}[z_i]\mathbb{E}[z_j] + \sum_{i,j} \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \,\text{Cov}(z_i, z_j)$. The first term, containing the expectations, can be factored into the square of a sum: $\left(\sum_i \mathbf{w}_{\mu,i} \mathbb{E}[z_i]\right)^2$, which can be computed in $\mathcal{O}(kd)$ time. The second covariance term can be simplified due to the independence of the latent variables. The covariance is zero for all z_i, z_j $(i \neq j)$ pairs, zeroing out all cross-terms. This collapses the double summation into a single sum over the variances $\text{Var}(z_i)$. The final analytical expression is:[2]

$$\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2] = ||\sum_i \mathbf{w}_{\mu,i} \mathbb{E}[z_i]||_2^2 + \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Var}(z_i). \tag{4}$$

Therefore, the expected squared error becomes:

$$\mathbb{E}[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||_2^2] = ||\boldsymbol{x}||_2^2 - 2\boldsymbol{x}^T(\sum_i \mathbf{w}_{\mu,i}\mathbb{E}[z_i]) + ||\sum_i \mathbf{w}_{\mu,i}\mathbb{E}[z_i]||_2^2 + \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Var}(z_i).$$

Finally, we get the expected reconstruction log-likelihood $\mathbb{E}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$:

$$-\frac{1}{2\sigma^2}\left[||\boldsymbol{x} - \mathbf{W}_\mu \boldsymbol{z}||^2 + \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Var}(z_i)\right] - \frac{1}{2}\log(2\pi\sigma^2).$$

This final equation is fully analytical. The expectation over the latent variable $\boldsymbol{z}$ has been entirely eliminated, and the reconstruction loss now depends only on the mean $(\mathbb{E}[z_i])$ and variance $\text{Var}(z_i)$ of the latent distribution. This allows for direct and analytic gradient computation with respect to the encoder parameters.

3.2 Do Analytic Gradients help?

As we show that the ELBO (and its gradients) can be computed exactly in an idealized setting with a linear decoder, we now investigate the practical impact of this zero-variance gradient. We conduct a controlled experiment on the MNIST dataset [5] to quantify the performance gains from eliminating gradient estimation noise. This experiment uses the same simple VAE with a linear decoder and fixed output variance ($\sigma^2 = 0.01$) (cf. Eq. (2)) as our our gradient variance analysis in Table 1, a setting designed to isolate the estimator's impact rather than to achieve state-of-the-art results. To ensure a fair comparison, we perform a separate hyperparameter search for each method. By epoch 500, all models have converged, and we report all final metrics at this point. All further details regarding the model architecture and hyperparameters can be found in Appendix C. We benchmark our method, Silent Gradients, against the reparameterization trick in the continuous latent space and against the Gumbel-Softmax estimator and the REINFORCE algorithm in the discrete space. Model performance is evaluated using Bits Per Dimension (BPD) and Mean Squared Error (MSE).

The results, summarized in Section 3.2, demonstrate the consistent advantages of our method. In the discrete latent space setting, our method achieves a substantially lower BPD than the corresponding baseline estimators. In the continuous case, while the final BPD scores are comparable, our method demonstrates significantly faster convergence; Silent Gradients reaches a BPD of 6.73 in just 45

[2] The detailed step-by-step derivation is in Appendix B.1.

Table 2: Performance comparison on MNIST using a linear decoder with a latent dimension of 200 under fixed variance $\sigma^2 = 0.01$. The best BPD and MSE values for the continuous and discrete latent space are in bold, respectively. Silent Gradients consistently outperforms stochastic gradient estimators in terms of BPD and MSE.

	Method	BPD (↓)	MSE (↓)
Continuous	Silent Gradients	**6.718**	**3.011**
	Reparameterization	6.722	3.059
Discrete	Silent Gradients	**6.900**	**6.103**
	Gumbel-Softmax	6.990	7.670
	REINFORCE	7.208	9.289

epochs, a milestone that the standard reparameterization trick requires 90 epochs to achieve. Besides BPD scores, the low MSE of our method confirms the high-fidelity image reconstruction, indicating that the reported BPD is not limited by poor reconstruction quality but is instead constrained by the fixed-variance assumption. The sharp reconstructions in Appendix D visually corroborate this conclusion.

4 More Expressive Decoders

We have shown that Silent Gradients offers a significant performance boost when the analytic gradient of the ELBO can be tractably computed. However, it is still unclear how to apply our method to VAEs with more general decoders. We address this in two steps: first, we demonstrate how to generalize the linear Gaussian decoder setting to make the variance a learnable parameter. Next, we show that this tractable linear component can be integrated with any existing VAE to guide encoder learning.

4.1 Linear Decoders with Learnable Variance

A key limitation of the fixed-variance Gaussian decoder introduced in the previous section is its inability to dynamically adjust confidence across different variables, resulting in significant performance degradation. This motivates generalizing our approach to allow variance to be a learnable, data-dependent function of the latent variable $\boldsymbol{z}$. That is, given latents $\boldsymbol{z}$, we predict both the mean $\mu(\boldsymbol{z})$ and the variance $\sigma^2(\boldsymbol{z})$ of the Gaussian distribution.

Under this parameterization, the first term of the expected reconstruction log-likelihood (i.e., Eq. (3)) is generalized to $\mathbb{E}[\frac{(\boldsymbol{x}-\mu(\boldsymbol{z}))^2}{2\sigma^2(\boldsymbol{z})}]$. Computing the expectation of a reciprocal is #P-hard for simple function classes including multilinear polynomials [24]. Furthermore, to ensure the expression is well-defined, $\sigma^2(\mathbf{z})$ must be strictly positive. While this can be enforced through techniques such as lower-bounding the variance by clipping, these approaches introduce discontinuities that complicate the analytical computation of the expectation and may hinder stable optimization. In addition, the second term in the reconstruction log-likelihood, $\frac{1}{2}\log(2\pi\sigma^2(\boldsymbol{z}))$, that involves the expectation of a logarithm, is also computationally hard [24].

To sidestep these challenges, we propose to represent the scale of the Gaussian distribution by the reciprocal of the standard deviation.[3] Formally, this quantity is called precision and is defined as $\alpha(\boldsymbol{z}) = 1/\sigma(\boldsymbol{z})$. Following Section 3, we define both the mean $\mu(\boldsymbol{z})$ and the precision as linear functions of the latent variable $\boldsymbol{z}$: $\mu(\boldsymbol{z}) = \mathbf{W}_\mu \boldsymbol{z}, \alpha(\boldsymbol{z}) = \mathbf{W}_\alpha \boldsymbol{z}$, which gives the model flexibility to assign pixel-wise uncertainty. Following eq. (2), we define the generative likelihood as a Gaussian distribution where the mean is $\mu(\boldsymbol{z})$ and the variance is the element-wise inverse square of $\alpha(\boldsymbol{z})$: $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}\left(\boldsymbol{x}; \mu(\boldsymbol{z}), \operatorname{diag}\left(\frac{1}{\alpha(\boldsymbol{z})^2}\right)\right)$. By the definition of the precision, the expected reconstruction log-likelihood from the ELBO becomes:

$$\mathbb{E}[\log p_\theta(x|z)] = -\frac{1}{2}\mathbb{E}\left[||(\boldsymbol{x} - \mu(\boldsymbol{z})) \odot \alpha(\boldsymbol{z})||_2^2\right] + \mathbb{E}[\log(\alpha(\boldsymbol{z}))] - \frac{1}{2}\log(2\pi). \tag{5}$$

The exact computation of both the first term and the second term is non-trivial. The first term involves an expectation of products of correlated functions of $\boldsymbol{z}$. The second term is hard since

[3]This parametrization aligns with the classical notion of precision, as originally defined by Gauss [6]; today the term precision is also used to denote the reciprocal of the variance.

$\mathbb{E}[\log(\alpha(\boldsymbol{z}))] \neq \log(\mathbb{E}[\alpha(\boldsymbol{z})])$. We expand the first expectation term as follows:

$$\begin{aligned}&\mathbb{E}\left[||(\boldsymbol{x}-\mu(\boldsymbol{z}))\odot\alpha(\boldsymbol{z})||_2^2\right] = \\ &\quad \mathbf{1}^T\left(||\boldsymbol{x}||_2^2\,\mathbb{E}\left[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right] - 2\boldsymbol{x}\odot\mathbb{E}\left[\mathbf{W}_\mu\boldsymbol{z}||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right] + \mathbb{E}\left[||\mathbf{W}_\mu\boldsymbol{z}||_2^2||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right]\right).\end{aligned} \tag{6}$$

The first term in this expansion, $\mathbb{E}\left[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right]$, is identical in form to the quadratic term derived in the fixed-variance setting (i.e. Equation (4)). As we showed previously, it can be computed analytically, depending on only the mean and variance of latent distribution. For the two remaining terms, $\mathbb{E}[\mathbf{W}_\mu\boldsymbol{z}||\mathbf{W}_\alpha\boldsymbol{z}||_2^2]$, and $\mathbb{E}[||\mathbf{W}_\mu\boldsymbol{z}||_2^2||\mathbf{W}_\alpha\boldsymbol{z}||_2^2]$, we begin the derivation by separating the terms into their expected values and covariances:

$$\begin{aligned}\mathbb{E}\left[\mathbf{W}_\mu\boldsymbol{z}||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right] &= \mathbb{E}[\mathbf{W}_\mu\boldsymbol{z}]\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2] + \mathrm{Cov}(\mathbf{W}_\mu\boldsymbol{z}, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2).\\ \mathbb{E}\left[||\mathbf{W}_\mu\boldsymbol{z}||_2^2||\mathbf{W}_\alpha\boldsymbol{z}||_2^2\right] &= \mathbb{E}[||\mathbf{W}_\mu\boldsymbol{z}||_2^2]\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2] + \mathrm{Cov}(||\mathbf{W}_\mu z||_2^2, ||\mathbf{W}_\alpha z||_2^2).\end{aligned}$$

The challenge, therefore, lies in deriving the two covariance terms. Following the principles for computing the covariance of products of random variables by Bohrnstedt and Goldberger [3], these terms can be decomposed into functions of central moments of the individual latent variables z_i. This makes the tractability of the entire expression dependent on whether these underlying central moments can be computed in closed form, as shown below.

Proposition 1. *(Tractable Central Moments) Let $\boldsymbol{z} \in \mathbb{R}^d$ be a random vector with independent components z_i. The first four central moments of each component, $\mathbb{E}[\tilde{z}_i] := \mathbb{E}[(z_i - \mathbb{E}[z_i])^k]$ for $k \in \{1, 2, 3, 4\}$, can be computed in closed form of the parameters of its distribution if z_i follows a Gaussian distribution, $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, or a Bernoulli distribution, $z_i \sim$ Bern(p_i).*

Derivation Sketch. The proof follows from the definitions of the moment-generating functions for each distribution. For a Gaussian variable, the central moments can be derived to be simple functions of its variance σ_i^2 [26]. For a Bernoulli variable with probability p_i, the raw moments $\mathbb{E}[z_i^k]$ are trivial to compute, and the central moment of order k is given by $(1-p_i)(-p_i)^k + p_i(1-p_i)^k$, resulting in polynomials of p_i. The full derivations are provided in the Appendix B.

With tractability of the individual central moments established, we can now show how this allows for the analytical computation of the full covariance terms.

Theorem 1. *(Analytic Covariance of Linear Projections) Let $\mathbf{W}_\mu\boldsymbol{z}$ and $\mathbf{W}_\alpha\boldsymbol{z}$ be two linear projections of a random vector $\boldsymbol{z}$ whose components z_i are independent. The covariance terms Cov$(\mathbf{W}_\mu\boldsymbol{z}, ||\mathbf{W}_\alpha\boldsymbol{z}||^2)$ and Cov$(||\mathbf{W}_\mu\boldsymbol{z}||_2^2, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)$ can be expressed as a linear combination of the first four central moments of the components z_i. The coefficients of this linear combination are polynomials in the entries of the matrices $\mathbf{W}_\mu$ and $\mathbf{W}_\alpha$.*

Proof Sketch. The proof relies on the formula for the covariance of products of random variables. The full derivation is in Appendix B.

1. For the term $\mathrm{Cov}(\mathbf{W}_\mu\boldsymbol{z}, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)$: We decompose this covariance term into a function of third-order expectations, $\mathbb{E}[\tilde{z}_i\tilde{z}_j\tilde{z}_k]$, where $\tilde{\boldsymbol{z}} = \boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]$. Due to the independence of the latent variables z_i, these expectations are non-zero only when all indices are identical ($i = j = k$). This simplifies the expression to a function of the second and third central moments of z_i. The resulting expression is a linear combination of the second and third central moments of z_i. Its coefficients are third-degree polynomials of the weight matrices $\mathbf{W}_\mu$ and $\mathbf{W}_\alpha$.

2. For the term $\mathrm{Cov}(||\mathbf{W}_\mu\boldsymbol{z}||_2^2, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)$: Similarly, this term can be decomposed into functions of fourth-order expectations, $\mathbb{E}[\tilde{z}_i\tilde{z}_j\tilde{z}_k\tilde{z}_l]$. Under the independence assumption, these complex expectations simplify into a linear combination of the second, the third, and the fourth central moments. The coefficients are fourth-degree polynomials of the weight matrices.

While the covariance terms can be computed analytically, the full log-likelihood function as in Equation (5) still contains the intractable logarithmic term $\mathbb{E}[\log(\mathbf{W}_\alpha\boldsymbol{z})]$, To ensure the argument of the logarithm is non-negative and to maintain a tractable, zero-variance objective, we approximate the term $\mathbb{E}[\log(||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)]$ using a second order Taylor Expansion around the mean of $||\mathbf{W}_\alpha\boldsymbol{z}||_2^2$ [20]:

$$\mathbb{E}[\log(||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)] \approx \log(\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)] - \frac{\mathrm{Var}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2]}{2(\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z})^2]||_2^2}.$$

Algorithm 1 Training Dynamics for Integrating Silent Gradients

Require: Encoder E_ϕ, Linear Decoder for $\alpha(\boldsymbol{z})$ D_{lin}, Nonlinear Decoder D_{nl}
Require: Training data $\mathcal{D}$, cut-off epoch N_{cutoff}, annealing rate λ
for n_{epoch} = 1 to N_{max} **do**
  if n_{epoch} == N_{cutoff} **then**
    Freeze parameters ϕ of the encoder E_ϕ
  end if
  for batch $\boldsymbol{x}$ in $\mathcal{D}$ **do**
    $\boldsymbol{z}$, stats $\leftarrow E_\phi(\boldsymbol{x})$
    $\mathcal{L}_{lin} \leftarrow -D_{lin}(\text{stats}, \boldsymbol{x})$ ▷ Analytical ELBO component Equation (7)
    $\mathcal{L}_{nl} \leftarrow -\log p_{nl}(\boldsymbol{x}|\boldsymbol{z})$ ▷ Sampled reconstruction loss
    $w_{lin} \leftarrow \max(0, 1 - n_{epoch} \cdot \lambda)$, $w_{nl} \leftarrow 1 - w_{lin}$
    $\mathcal{L}_{total} \leftarrow w_{lin} \cdot \mathcal{L}_{lin} + w_{nl} \cdot \mathcal{L}_{nl} + D_{KL}$
    Take gradient step on $\mathcal{L}_{total}$ for all unfrozen parameters
  end for
end for

By combining the exact computations for the covariance terms with this approximation, the expected reconstruction log-likelihood can be expressed in an analytical solution:

$$\begin{aligned}\mathbb{E}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] &= \frac{1}{2}\big[||\boldsymbol{x}||_2^2\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2] - 2\boldsymbol{x}^T(\mathbf{W}_\mu\mathbb{E}[\boldsymbol{z}]\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||^2]) \\ &+ \mathrm{Cov}(\mathbf{W}_\mu\boldsymbol{z}, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2) + \mathbb{E}[||\mathbf{W}_\mu\boldsymbol{z}||_2^2]\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2] \\ &+ \mathrm{Cov}(||\mathbf{W}_\mu\boldsymbol{z}||_2^2, ||\mathbf{W}_\alpha\boldsymbol{z}||_2^2)\big] + \frac{1}{2}\log(2\pi)\left(\frac{1}{2}\log(\mathbb{E}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2]) - \frac{\mathrm{Var}[||\mathbf{W}_\alpha\boldsymbol{z}||_2^2]}{4(\mathbb{E}[||\mathbf{W}_\alpha z||_2^2])^2}\right). \quad (7)\end{aligned}$$

This expression relies only on the tractable moments of the latent distribution and the decoder weights.

4.2 Silent Gradients with General VAEs

While the preceding section demonstrates that analytical gradients are tractable for linear decoders with learnable variance, the expressive power of a purely linear model is limited. To handle more complex data distribution, we now introduce a training strategy that integrates the benefits of our tractable Silent Gradients with general, powerful nonlinear decoders.

Our approach uses a dual-decoder architecture consisting of a shared encoder, a linear decoder for computing the exact ELBO component and computing the exact Silent Gradients, and a parallel, more expressive nonlinear decoder for generating the final reconstructions. A visualization of this pipeline is presented in Figure 1. The training follows a two-stage process. In the initial stage, the encoder and both decoders are trained, but the encoder parameters are updated only using the analytic gradients from the linear decoder. After a set number of epochs, we freeze the encoder's weights. In the second stage, only the nonlinear decoder continues to train, fine-tuning its parameters on the now fixed, well-structured latent space provided by the encoder. The Silent Gradients framework

Table 3: **Performance comparison (BPD) (↓) for models with learnable variance across different datasets and methods.** The BPD score for the combined method is in bold when it is higher than its corresponding baseline. In both cases, the optimal performance is achieved by combining a standard estimator with our Silent Gradients. The results show that combining standard estimators with our Silent Gradients (SG) consistently improves performance. Additionally, our method used as a standalone estimator is competitive with and often superior to established baselines like REINFORCE.

	Method	MNIST		ImageNet		CIFAR-10	
		w/o SG	w/ SG	w/o SG	w/ SG	w/o SG	w/ SG
Continuous	None	–	2.41	–	5.98	–	5.82
	Reparameterization	1.91	**1.80**	5.79	**5.69**	5.70	**5.53**
Discrete	None	–	2.77	–	6.45	–	6.72
	Gumbel-Softmax	2.48	**2.37**	6.31	**6.20**	6.22	**6.19**
	REINFORCE	2.96	**2.94**	6.87	**6.77**	6.74	**6.67**

Table 4: **KL Divergence (KLD) comparison for models with learnable variance.** The KLD for the combined method is in bold when it is higher than its corresponding baseline. The results consistently show a higher KLD when a baseline estimator is combined with our Silent Gradient technique, which suggests the encoder learns a more informative latent representation.

	Method	MNIST		ImageNet		CIFAR-10	
		w/o SG	w/ SG	w/o SG	w/ SG	w/o SG	w/ SG
Continuous	None	–	330.77	–	478.43	–	550.70
	Reparameterization	155.15	**165.76**	382.87	**533.20**	427.79	**577.25**
Discrete	None	–	91.82	–	534.29	–	243.69
	Gumbel-Softmax	94.64	**96.24**	368.02	**404.57**	446.75	442.58
	REINFORCE	109.67	**128.55**	294.88	**381.27**	303.33	**367.10**

can be extended to boost the performance of existing gradient estimators. Instead of having the encoder rely solely on the linear decoders' analytical gradient, we introduce a gradient annealing schedule. In this combined approach, the gradient signal sent to the encoder E_ϕ is a weighted average: $\nabla_{\phi,\text{total}} = w_{\text{lin}}\nabla_{\phi,\text{Silent}} + w_{\text{nl}}\nabla_{\phi,\text{Noisy}}$,where $w_{\text{nl}} = 1 - w_{\text{lin}}$, $\nabla_{\phi,\text{Silent}}$ is the analytical gradient from the linear decoder, and $\nabla_{\phi,\text{Noisy}}$ is the noisy gradient from the nonlinear decoder using stochastic estimators. The training begins with the weight of the Silent Gradients, w_{lin}, at 1.0 and the weight of the baseline estimator's gradient, w_{nl}, at 0.0. As training progresses, w_{lin} is gradually annealed to 0 while w_{nl} is increased to 1.0. This dynamic allows the encoder to first learns a representation guided by the noise-free, analytical signal before fine-tuning with the sample-based gradients from the full, expressive model. The complete training dynamic is detailed in Algorithm 1.

We benchmark both our standalone Silent Gradients method and the combined approach against baselines on MNIST, ImageNet, and CIFAR-10. All models were tuned for optimal hyperparameters and trained until convergence to ensure a fair comparison. Our experimental results, presented in Section 4.2, demonstrate two key findings. First, our Silent Gradients method consistently improves the performance of existing gradient estimators. In every case, combining a standard estimator with our technique results in a lower BPD score compared to the baseline alone across all tested datasets. This shows that our analytical gradient serves as a powerful and general-purpose training aid. Second, our method used as a standalone estimator is highly competitive, even outperforming the widely used REINFORCE estimator on both MNIST and ImageNet. We defer more experiment details to Appendix Appendix C, and the visualized reconstruction output is presented in Appendix D.

An analysis of the KL Divergence (KLD) offers an explanation for these performance gains. As shown in Section 4.2, models trained with Silent Gradients consistently achieve a higher KLD, which suggests that the encoder learns a more informative latent representation and better avoids posterior collapse. We hypothesize this is because the zero-variance analytical gradient provides a cleaner, more stable training signal to the encoder than the noisy gradients from the standard stochastic estimators.

It is important to note that while these results are not state-of-the-art, they are by design; our experiments use a simple decoder architecture to isolate the impact of our gradient computation technique, rather than to achieve record-breaking BPD. Our method provides a consistent and significant performance lift across all baselines, demonstrating its broad potential as a general tool for improving the training of deep generative models.

5 Conclusion

In this work, we introduced Silent Gradients, a new approach to training VAEs without the problem brought by variance in gradient estimation. Instead of improving stochastic estimators, we leverage specific decoder architectures to analytically compute a zero-variance gradient signal. We provided a derivation for this method and demonstrated its effectiveness empirically. Our experiments show that Silent Gradients not only outperforms standard estimators in a controlled setting but also consistently improves their performance when combined through a novel training dynamic in general VAEs.

References

[1] Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. SIMPLE: A gradient estimator for k-subset sampling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=GPJVuyX4p_h.

[2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

[3] George W. Bohrnstedt and Arthur S. Goldberger. On the exact covariance of products of random variables. *Journal of the American Statistical Association*, 64(328):1439–1442, 1969. ISSN 01621459, 1537274X. URL http://www.jstor.org/stable/2286081.

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

[5] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

[6] Carl Friedrich Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, volume 7. FA Perthes, 1877.

[7] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkE3y85ee.

[8] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, November 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178. URL https://doi.org/10.1023/A:1007665907178.

[9] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[10] Frederic Koehler, Viraj Mehta, Chenghui Zhou, and Andrej Risteski. Variational autoencoders in the presence of low-dimensional data: landscape and implicit bias. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=y_op4lLLaWL.

[11] Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 reinforce samples, get a baseline for free! In *DeepRLStructPred@ICLR*, 2019. URL https://api.semanticscholar.org/CorpusID:198489118.

[12] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[13] Yin Lu, Xuening Zhu, Tong He, and David Wipf. Sparse autoencoders, again? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=wxU2LuTE74.

[14] James Lucas, George Tucker, Roger Grosse, and Mohammad Norouzi. *Don't blame the ELBO! a linear VAE perspective on posterior collapse*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[15] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=S1jE5L5gl.

[16] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1791–1799, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/mnih14.html`.

[17] Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit MLE: Backpropagating through discrete exponential family distributions. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=lR4aaWCQgB`.

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL `https://proceedings.mlr.press/v32/rezende14.html`.

[19] Lennert De Smet, Emanuele Sansone, and Pedro Zuidberg Dos Martires. Differentiable sampling of categorical distributions using the catlog-derivative trick. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=AQyqxXctsN`.

[20] Yee Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper_files/paper/2006/file/532b7cbe070a3579f424988a040752f2-Paper.pdf`.

[21] Hadi Vafaii, Dekel Galor, and Jacob L. Yates. Poisson variational autoencoder. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=ektPEcqGLb`.

[22] Arash Vahdat, Evgeny Andriyash, and William G. Macready. Dvae#: discrete variational autoencoders with relaxed boltzmann priors. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1869–1878, Red Hook, NY, USA, 2018. Curran Associates Inc.

[23] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. URL `https://arxiv.org/abs/1711.00937`.

[24] Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13189–13201. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/6e01383fd96a17ae51cc3e15447e7533-Paper.pdf`.

[25] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.

[26] Andreas Winkelbauer. Moments and absolute moments of the normal distribution, 2014. URL `https://arxiv.org/abs/1209.4340`.

A Related Work

VAEs. Variational Autoencoders (VAEs) are generative models that learn a latent representation of data through an encoder-decoder framework [9]. They can be categorized by their latent space: VAEs with continuous latent variables typically use Gaussian distributions and are widely applied to tasks like image modeling [9], while VAEs with discrete latent spaces have become an active research area, as discrete representations can offer better interpretability and computational efficiency [23, 7]. This line of work contains various architectures of the discrete latent space, such as the use of vector quantization in VQ-VAE [23] and relaxed Boltzmann priors in DAVE# [22].

Gradient Estimation Techniques. A key challenge in training VAEs is propagating gradients through stochastic sampling layers. In the continuous case, the reparameterization trick, which separates the stochasticity into a fixed noise source and a deterministic function, is widely used [9]. Although unbiased, reparameterization still introduces variance that impedes optimization.

In the discrete case, two main lines of techniques are used. The first is the use of the REINFORCE technique, or score function estimator, which provides a general and unbiased gradient estimate applicable to both discrete and continuous latent variables. It rewrites the gradient of the expectation as: $\nabla_\phi \mathbb{E}_{q_\phi(\boldsymbol{z})} f(\boldsymbol{z}) = \mathbb{E}_{q_\phi(\boldsymbol{z})}[f(\boldsymbol{z})\nabla_\phi \log q_\phi(\boldsymbol{z})]$ [25]. However, this estimator is often hindered by high variance, which has led to the development of variance reduction techniques such as control variates, [16, 11].

The second line of research strives to make discrete variables compatible with low-variance reparameterization trick. The straight-through (ST) estimator approximates the discrete sampling in the backward pass with a differentiable function, such as using the mean value for a Bernoulli variable [2]. Another approach is to relax discrete variables into a continuous distribution; the Concrete [15] or Gumbel-Softmax [7] distribution, for instance, achieves this by adding Gumbel noise to the logits of a softmax function, enabling reparameterization. More recent techniques such as SIMPLE [1], IndeCateR [19], and Implicit Maximum Likelihood Estimation (IMLE) [17] offer alternative strategies to derive low-variance gradient estimates for generative models with discrete latent variables.

Linear VAEs. Linear VAEs are a cornerstone in various contexts. First, their analytical tractability makes them an ideal setting for theoretical investigation. For example, Lucas et al. [14] used linear VAEs to show that posterior collapse can be an inherent issue of the marginal log-likelihood objective, not a problem caused by the ELBO approximation. Other work uses them to investigate the implicit bias of gradient descent, showing how training dynamics can recover the ground-truth data manifold [10]. Additionally, linear decoders are also crucial in tasks such as learning sparse and interpretable features from complex data [13, 21]. This broad utility motivates our method, which allows for analytic gradient estimation for any VAE with a linear decoder. Having demonstrated its effectiveness in image modeling, Silent Gradients could directly enhance these other applications.

Analytic ELBO. Lucas et al. [14] derive an analytical ELBO for linear VAEs under assumptions of a fixed scalar output variance, a Gaussian latent space and a linear encoder. In contrast, our method is more general, that supports a learnable variance for output Gaussian distribution, applies to any latent distribution with tractable central moments, and makes no assumptions about the encoder architecture.

B Derivation

In this section, we provide the step-by-step derivation for Equation (4), Proposition 1, and Theorem 1.

B.1 Equation (4) ($\mathbb{E}||\mathbf{W}_\mu \boldsymbol{z}||_2^2$)

We wish to prove:

$$\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2] = ||\sum_i \mathbf{w}_{\mu,i}\mathbb{E}[z_i]||_2^2 + \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Var}(z_i). \tag{8}$$

Derivation. To begin with, we shall expand $\mathbb{E}||\mathbf{W}_\mu \boldsymbol{z}||_2^2$ using summations:

$$\begin{aligned}
\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2] &= \mathbb{E}\left[||\sum_i \mathbf{w}_{\mu,i} z_i||_2^2\right], && (9)\\
&= \mathbb{E}\left[\sum_i \sum_j (\mathbf{w}_{\mu,i} z_i)^T (\mathbf{w}_{\mu,j} z_j)\right], && (10)\\
&= \sum_i \sum_j \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \mathbb{E}[z_i z_j]. && (11)
\end{aligned}$$

where $\mathbf{w}_{\mu,i}$ is the ith column of $\mathbf{W}_\mu$. Notably, $\mathbb{E}[z_i z_j] = \mathbb{E}[z_i]\mathbb{E}[z_j] + \text{Cov}(z_i, z_j)$, by the fundamental identity relating the second moment of a random vector to its mean and covariance. Using this identity, we further split the expression into:

$$= \sum_i \sum_j \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \mathbb{E}[z_i]\mathbb{E}[z_j] + \sum_i \sum_j \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \text{Cov}(z_i, z_j). \tag{12}$$

The first term, containing the expectations, can be factored into the square of a sum: $||\sum_i \mathbf{w}_{\mu,i}\mathbb{E}[z_i]||_2^2$. The second term, involving the covariance is simplified by decomposing the summation into two cases. The first case is when the indices are equal ($i = j$), and the second is when they are not ($i \neq j$):

$$\begin{aligned} &\sum_i \sum_j \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \text{Cov}(z_i, z_j) \\ &= \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Cov}(z_i, z_i) + \sum_{i \neq j} \mathbf{w}_{\mu,i}^T \mathbf{w}_{\mu,j} \text{Cov}(z_i, z_j). \end{aligned} \tag{13}$$

By definition, the covariance of a variable with itself is its variance: $\text{Cov}(z_i, z_i) = \text{Var}(z_i)$. Additionally, we assume the components of the latent vector z are independent. A standard property of independent random variables is that their covariance is 0. Therefore, for all $i \neq j$, $\text{Cov}(z_i, z_j) = 0$, which cancels out the second summation entirely. By combining the simplified expectation and covariance terms, the final analytical expression for the quadratic term is:

$$\mathbb{E}[||\mathbf{W}_\mu \boldsymbol{z}||_2^2] = ||\sum_i \mathbf{w}_{\mu,i}\mathbb{E}[z_i]||_2^2 + \sum_i ||\mathbf{w}_{\mu,i}||_2^2 \text{Var}(z_i). \tag{14}$$

B.2 Proposition 1

Proposition 1. Let $\boldsymbol{z} \in \mathbb{R}^d$ be a random vector with independent components z_i. The first four central moments of each component, $\mathbb{E}[\tilde{z}_i] := \mathbb{E}[(z_i - \mathbb{E}[z_i])^k]$ for $k \in \{1, 2, 3, 4\}$, can be computed in closed form of the parameters of its distribution if z_i follows:

1. A Gaussian distribution, $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

2. A Bernoulli distribution, $z_i \sim \text{Bern}(p_i)$.

Derivation. 1. Let z_i be a random variable following a Gaussian distribution, $z_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$.

The kth central moment is defined as $\mathbb{E}[(z_i - \mathbb{E}[z_i])^k]$. Winkelbauer [26] introduces the formula to calculate central moments for a Gaussian distribution:

$$\mathbb{E}[(z_i - \mu_i)^k] = \begin{cases} \sigma_i^k (k-1)!! & \text{if } k \in \mathbb{N}_+ \text{ is even,} \\ 0 & \text{if } k \in \mathbb{N}_+ \text{ is odd.} \end{cases} \tag{15}$$

where $(k-1)!!$ is the double factorial. Using this formula, we can state the first four central moments:

- $k = 1$. $\mathbb{E}[z_i - \mu_i] = 0$ since k is odd.
- $k = 2$. $\mathbb{E}[(z_i - \mu_i)^2] = \sigma_i^2(2-1)!! = \sigma_i^2$ since k is even. Notably, the result is the variance of this Gaussian distribution.
- $k = 3$. $\mathbb{E}[(z_i - \mu_i)^3] = 0$ since k is odd.
- $k = 4$. $\mathbb{E}[(z_i - \mu_i)^4] = \sigma_i^4(4-1)!! = 3\sigma_i^4$ since k is even.

Therefore, all four central moments are closed-form functions of the distribution's variance σ_i^2.

2. Let z_i be a random variable following a Bernoulli distribution, $z_i \sim \text{Bern}(p_i)$. The variable z_i takes the value 1 with probability p_i and 0 with probability $1 - p_i$. The mean is $\mathbb{E}[z_i] = p_i$.

The kth central moment is defined as $\mathbb{E}[(z_i - \mathbb{E}[z_i])]$. We consider the two possible outcomes for z_i:

- If $z_i = 1$, then $(z_i - p_i)^k = (1 - p_i)^k$.
- If $z_i = 0$, then $(z_i - p_i)^k = (-p_i)^k$.

We can compute the central moments using the definition of expectation.

$$\mathbb{E}[(z_i - p_i)^k] = (1-p_i)^k \cdot p_i + (-p_i)^k \cdot (1-p_i). \tag{16}$$

Now we can compute the first four central moments:

- $k = 1$. $\mathbb{E}[z_i - p_i] = (1-p_i)^1 p_i + (-p_i)^1(1-p_i) = p_i - p_i^2 - p_i + p_i^2 = 0$.
- $k = 2$.

$$\begin{aligned}\mathbb{E}[(z_i - p_i)^2] &= (1-p_i)^2 p_i + (-p_i)^2(1-p_i), \\ &= p_i - 2p_i^2 + p_i^3 + p_i^2 - p_i^3, \\ &= p_i - p_i^2 = p_i(1-p_i).\end{aligned} \tag{17}$$

- $k = 3$.

$$\begin{aligned}\mathbb{E}[(z_i - p_i)^3] &= (1-p_i)^3 p_i + (-p_i)^3(1-p_i), \\ &= (1 - 3p_i + 3p_i^2 - p_i^3)p_i - p_i^3(1-p_i), \\ &= p_i - 3p_i^2 + 3p_i^3 - p_i^4 - p_i^3 + p_i^4, \\ &= p_i - 3p_i^2 + 2p_i^3 \\ &= p_i(1-p_i)(1-2p_i).\end{aligned} \tag{18}$$

- $k = 4$.

$$\begin{aligned}\mathbb{E}[(z_i - p_i)^4] &= (1-p_i)^4 p_i + (-p_i)^4(1-p_i), \\ &= (1 - 4p_i + 6p_i^2 - 4p_i^3 + p_i^4)p_i \\ &\quad + p_i^4(1-p_i), \\ &= p_i - 4p_i^2 + 6p_i^3 - 4p_i^4 + p_i^5 + p_i^4 - p_i^5, \\ &= p_i - 4p_i^2 + 6p_i^3 - 3p_i^4 \\ &= p_i(1-p_i)(1 - 3p_i + 3p_i^2).\end{aligned} \tag{19}$$

Therefore, all four central moments are closed-form functions of the parameter p_i.

B.3 Theorem 1

Theorem 1. Let $\mathbf{W}_\mu \boldsymbol{z}$ and $\mathbf{W}_\alpha \boldsymbol{z}$ be two linear projections of a random vector $\boldsymbol{z}$ whose components z_i are independent. The covariance terms $\text{Cov}(\mathbf{W}_\mu \boldsymbol{z}, ||\mathbf{W}_\alpha \boldsymbol{z}||^2)$ and $\text{Cov}(||\mathbf{W}_\mu \boldsymbol{z}||_2^2, ||\mathbf{W}_\alpha \boldsymbol{z}||_2^2)$ can be expressed as a linear combination of the first four central moments of the components z_i. The coefficients of this linear combination are polynomials in the entries of the matrices $\mathbf{W}_\mu$ and $\mathbf{W}_\alpha$.

Proof. For simplicity in writing, we define:

$$u_1 = u_2 = \mathbf{W}_\mu \boldsymbol{z}, \quad v_1 = v_2 = \mathbf{W}_\alpha \boldsymbol{z}, \tag{20}$$

$$\Delta u_1 = \Delta u_2 = \mathbf{W}_\mu \boldsymbol{z} - \mathbb{E}[\mathbf{W}_\mu \boldsymbol{z}] = \mathbf{W}_\mu(\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]), \tag{21}$$

$$\Delta v_1 = \Delta v_2 = \mathbf{W}_\alpha \boldsymbol{z} - \mathbb{E}[\mathbf{W}_\alpha \boldsymbol{z}] = \mathbf{W}_\alpha(\boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]). \tag{22}$$

And we denote $\tilde{\boldsymbol{z}} = \boldsymbol{z} - \mathbb{E}[\boldsymbol{z}]$, thus $\mathbb{E}[\tilde{\boldsymbol{z}}] = 0$, and $\mathbb{E}[(\tilde{\boldsymbol{z}})^2] = \text{Var}(\boldsymbol{z})$.

Bohrnstedt and Goldberger [3] introduces the formula to compute covariance between the products of independent variables as follows:

$$\begin{aligned}\text{Cov}(u_1, v_1 v_2) = \mathbb{E}[v_1]\text{Cov}(v_2, u_1) + \mathbb{E}[v_2]\text{Cov}(v_1, u_1) \\ + \mathbb{E}[(\Delta v_1)(\Delta v_2)(\Delta u_1)].\end{aligned} \tag{23}$$

Identical to the fixed variance case, we can derive $\text{Cov}(v_1, u_1) = \text{Cov}(v_2, u_1) = \sum_i \mathbf{w}_{\alpha,i}^T \mathbf{w}_{\mu,i} \text{Var}(z_i) = \mathbf{W}_\alpha^T \mathbf{W}_\mu \mathbb{E}[(\tilde{\boldsymbol{z}})^2]$. And to compute the last term, we expand it as follows, with $\odot$ denoting Hadamard product:

$$\begin{aligned}
&\mathbb{E}\big[(\Delta v_1)(\Delta v_2)(\Delta u_1)\big] && (24)\\
&= \mathbb{E}\big[(\mathbf{W}_\mu \tilde{\boldsymbol{z}}) \odot (\mathbf{W}_\alpha \tilde{\boldsymbol{z}}) \odot (\mathbf{W}_\alpha \tilde{\boldsymbol{z}})\big], && (25)\\
&= \mathbb{E}\big[\sum_i \sum_j \sum_k (\mathbf{w}_{\mu,i} \tilde{z}_i) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \tilde{z}_j \tilde{z}_k)\big], && (26)\\
&= \sum_i \sum_j \sum_k \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \mathbb{E}[\tilde{z}_i \tilde{z}_j \tilde{z}_k]. && (27)
\end{aligned}$$

Notably, $\mathbb{E}[\tilde{z}_i \tilde{z}_j \tilde{z}_k]$ is nonzero only if $i = k = j$. In other generic cases, for example, $i = j \neq k$, we can always separate $\tilde{z}_i \tilde{z}_j$ from $\tilde{z}_k$. In fact, because of the constraint, we can simplify the expression:

$$\begin{aligned}
\mathbb{E}[\tilde{z}_i \tilde{z}_i \tilde{z}_k] &= \mathbb{E}[(\tilde{z}_i)^2 \tilde{z}_k] && (28)\\
&= \mathbb{E}[(\tilde{z}_i)^2]\mathbb{E}[\tilde{z}_k] && (29)\\
&= 0 && (30)
\end{aligned}$$

Therefore, the only case we need to consider is when $i = j = k$, and thus we can write:

$$\begin{aligned}
\mathbb{E}\big[(\Delta v_1)(\Delta v_2)(\Delta u_1)\big] &= \sum_i \sum_j \sum_k \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \mathbb{E}[\tilde{z}_i \tilde{z}_i \tilde{z}_i] && (31)\\
&= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \mathbb{E}[(\tilde{z}_i)^3] && (32)
\end{aligned}$$

Piecing all together, we derive the expression for the covariance term $\text{Cov}(\mathbf{W}_\mu \boldsymbol{z}, (\mathbf{W}_\alpha \boldsymbol{z})^2)$:

$$\begin{aligned}
\text{Cov}(\mathbf{W}_\mu \boldsymbol{z}, ||\mathbf{W}_\alpha \boldsymbol{z}||_2^2) &= (\mathbf{W}_\alpha \mathbb{E}[\boldsymbol{z}]) \odot (\mathbf{W}_\alpha \odot \mathbf{W}_\mu \mathbb{E}[(\tilde{\boldsymbol{z}})^2])\\
&\quad + (\mathbf{W}_\alpha \mathbb{E}[\boldsymbol{z}]) \odot (\mathbf{W}_\alpha \odot \mathbf{W}_\mu \mathbb{E}[(\tilde{\boldsymbol{z}})^2])\\
&\quad \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \mathbb{E}[(\tilde{z}_i)^3],\\
&= 2(\mathbf{W}_\alpha \mathbb{E}[\boldsymbol{z}]) \odot (\mathbf{W}_\alpha \odot \mathbf{W}_\mu \mathbb{E}[(\tilde{\boldsymbol{z}})^2])\\
&\quad + \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,k} \mathbb{E}[(\tilde{z}_i)^3].
\end{aligned} \tag{33}$$

Bohrnstedt and Goldberger [3] also introduced the formula to calculate the covariance between two products of random variables:

$$\begin{aligned}
&\text{Cov}(u_1 u_2, v_1 v_2)\\
&= \mathbb{E}(u_1)\,\mathbb{E}(v_1)\,\text{Cov}(u_2, v_2) + \mathbb{E}(u_1)\,\mathbb{E}(v_2)\,\text{Cov}(u_2, v_1)\\
&\quad + \mathbb{E}(u_2)\,\mathbb{E}(v_1)\,\text{Cov}(u_1, v_2) + \mathbb{E}(u_2)\,\mathbb{E}(v_2)\,\text{Cov}(u_1, v_1)\\
&\quad + \mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_1\,\Delta v_2\big] + \mathbb{E}(u_1)\,\mathbb{E}\big[\Delta u_2\,\Delta v_1\,\Delta v_2\big]\\
&\quad + \mathbb{E}(u_2)\,\mathbb{E}\big[\Delta u_1\,\Delta v_1\,\Delta v_2\big] + \mathbb{E}(v_1)\,\mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_2\big]\\
&\quad + \mathbb{E}(v_2)\,\mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_1\big] - \text{Cov}(u_1, u_2)\,\text{Cov}(v_1, v_2).
\end{aligned} \tag{34}$$

Following the derivation earlier, we can compute the terms $\mathbb{E}(u_1)\,\mathbb{E}\big[\Delta u_2\,\Delta v_1\,\Delta v_2\big]$, $\mathbb{E}(u_2)\,\mathbb{E}\big[\Delta u_1\,\Delta v_1\,\Delta v_2\big]$, $\mathbb{E}(v_1)\,\mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_2\big]$, $\mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_1\big]$. And similarly, we wish to consider all nonzero cases in $\mathbb{E}\big[\Delta u_1\,\Delta u_2\,\Delta v_1\,\Delta v_2\big]$, and they are: $i = j = k = l, i = j \neq k =$

$l, i = k \neq j = l, i = l \neq k = j$.

$$\begin{aligned}
&\mathbb{E}\big[\Delta u_1\, \Delta u_2\, \Delta v_1\, \Delta v_2\big] && (35)\\
&= \mathbb{E}\big[\mathbf{W}_\mu \tilde{z} \odot \mathbf{W}_\mu \tilde{z} \odot \mathbf{W}_\alpha \tilde{z} \odot \mathbf{W}_\alpha \tilde{z}\big] && (36)\\
&= \mathbb{E}\big[\sum_i \sum_j \sum_k \sum_l \left[(\mathbf{w}_{\mu,i}\tilde{z}_i) \odot (\mathbf{w}_{\mu,j}\tilde{z}_j)\right] \odot \left[(\mathbf{w}_{\alpha,k}\tilde{z}_k) \odot (\mathbf{w}_{\alpha,l}\tilde{z}_l)\right]\big] && (37)\\
&= \sum_i \sum_j \sum_k \sum_j \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,k} \odot \mathbf{w}_{\alpha,l} \mathbb{E}[\tilde{z}_i \tilde{z}_j \tilde{z}_k \tilde{z}_l] && (38)
\end{aligned}$$

Consider the case where $i = j = k = l$,

$$\begin{aligned}
&\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[\tilde{z}_i \tilde{z}_i \tilde{z}_i \tilde{z}_i]\\
&= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^4] && (39)
\end{aligned}$$

And consider the case where $i = j \neq k = l$

$$\begin{aligned}
&\sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[\tilde{z}_i \tilde{z}_i \tilde{z}_j \tilde{z}_j] && (40)\\
&= \sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_i)^2 (\tilde{z}_j)^2] && (41)\\
&= \sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_i)^2] \mathbb{E}[(\tilde{z}_j)^2] && (42)\\
&= \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \mathbb{E}[(\tilde{z}_i)^2] \sum_{j, j\neq i} (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_j)^2] && (43)\\
&= \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \mathbb{E}[(\tilde{z}_i)^2] \left(\sum_j (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_j)^2] - (\mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i}) \mathbb{E}[(\tilde{z}_i)^2] \right) && (44)\\
&= \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \mathbb{E}[(\tilde{z}_i)^2] \sum_j (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_j)^2]\\
&\quad - \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i}) \mathbb{E}[(\tilde{z}_i)^2] (\mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i}) \mathbb{E}[(\tilde{z}_i)^2] && (45)
\end{aligned}$$

This holds because we know $i \neq k$. Similarly, when $i = k \neq j = l$,

$$\begin{aligned}
&\sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[\tilde{z}_i \tilde{z}_j \tilde{z}_i \tilde{z}_j] && (46)\\
&= \sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_i)^2 (\tilde{z}_j)^2] && (47)\\
&= \sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,j}) \mathbb{E}[(\tilde{z}_i)^2] \mathbb{E}[(\tilde{z}_j)^2] && (48)\\
&= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_{j, j\neq i} \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] && (49)\\
&= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2]\\
&\quad - \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] && (50)
\end{aligned}$$

When $i = l \neq k = j$,

$$\sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,i}) \mathbb{E}[\tilde{z}_i \tilde{z}_j \tilde{z}_j \tilde{z}_i] \tag{51}$$

$$= \sum_i \sum_{j} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,i}) \mathbb{E}[(\tilde{z}_i)^2 (\tilde{z}_j)^2] \tag{52}$$

$$= \sum_i \sum_{j, j\neq i} (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,j}) \odot (\mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,i}) \mathbb{E}[(\tilde{z}_i)^2] \mathbb{E}[(\tilde{z}_j)^2] \tag{53}$$

$$= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_{j, j\neq i} \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \tag{54}$$

$$= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \left(\sum_j \mathbf{w}_{\mu,j}^T \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] - \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \right) \tag{55}$$

$$\begin{aligned} &= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad - \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \end{aligned} \tag{56}$$

$$\tag{57}$$

Since these four cases are mutually exclusive, we could rewrite the full term as:

$$\begin{aligned} &\mathbb{E}\big[\Delta u_1 \, \Delta u_2 \, \Delta v_1 \, \Delta v_2\big] \\ &= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^4] \\ &\quad + \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\alpha,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad - \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad + 2 \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad - 2 \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2])(\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2]), \end{aligned} \tag{58}$$

$$\begin{aligned} &= \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^4] \\ &\quad + \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad + 2 \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\ &\quad - 3 \sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2])(\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2]). \end{aligned} \tag{59}$$

Putting this back to Equation (34), we can write the final expression as:

$$
\begin{aligned}
&\mathrm{Cov}(u_1u_2, v_1v_2) = \mathrm{Cov}(||\mathbf{W}_\mu z||_2^2, ||\mathbf{W}_\alpha z||_2^2) \\
&= 4(\mathbf{W}_\mu \odot \mathbf{W}_\mu \odot \mathbf{W}_\alpha \odot \mathbf{W}_\alpha (\mathbb{E}[\boldsymbol{z}])^2 \mathbb{E}[(\tilde{\boldsymbol{z}})^2] \\
&\quad + \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^4] \\
&\quad + 2\sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2] \sum_j \mathbf{w}_{\mu,j} \odot \mathbf{w}_{\alpha,j} \mathbb{E}[(\tilde{z}_j)^2] \\
&\quad - 3\sum_i (\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2])(\mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^2]) \\
&\quad + 2\mathbf{W}_\mu \mathbb{E}[z] \odot \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^3] \\
&\quad + 2\mathbf{W}_\alpha \mathbb{E}[z] \odot \sum_i \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\mu,i} \odot \mathbf{w}_{\alpha,i} \mathbb{E}[(\tilde{z}_i)^3].
\end{aligned} \tag{60}
$$

Therefore, we show that the covariance terms $\mathrm{Cov}(\mathbf{W}_\mu \boldsymbol{z}, ||\mathbf{W}_\alpha \boldsymbol{z}||^2)$ (cf. Eq. 33) and $\mathrm{Cov}(||\mathbf{W}_\mu \boldsymbol{z}||_2^2, ||\mathbf{W}_\alpha \boldsymbol{z}||_2^2)$ (cf. Eq. 60) can be expressed as a linear combination of the first four central moments of the components z_i. The coefficients of this linear combination are polynomials in the entries of the weight matrices $\mathbf{W}_\mu$ and $\mathbf{W}_\alpha$.

C Experiment Details

C.1 Uniform Dequantization

Our model's decoder defines a likelihood over continuous values using a Gaussian distribution. However, the image datasets we use, such as MNIST, consist of discrete pixels values. To bridge this gap, we employ uniform dequantization. This standard technique adds a small amount of uniform noise to each discrete pixel value, transforming data into a continuous variable that is compatible with our model's likelihood function.

Specifically, for discrete data $\boldsymbol{x}_{\text{int}}$ with values in $\{0, 1, \ldots, 255\}$, the dequantized data is defined as $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{u}$, where $\boldsymbol{x} = \frac{\boldsymbol{x}_{\text{int}}}{256}$, $\boldsymbol{u} \sim \mathcal{U}[0, \frac{1}{256})$. This process maps each discrete pixel value to a unique continuous bin of width $\frac{1}{256}$ within $[0, 1)$.

The true probability of a discrete pixel value under our continuous model is thus defined as:

$$
P_{\text{model}}(X = \boldsymbol{x}_{\text{int}}) = \int_{\boldsymbol{x}_{\text{int}}/256}^{(\boldsymbol{x}_{\text{int}}+1)/256} p_{\text{model}}(\boldsymbol{y}) d\boldsymbol{y} \tag{61}
$$

By applying Jensen's inequality, we can establish a formal relationship:

$$
\begin{aligned}
\mathbb{E}_{\boldsymbol{u}}[\log p_{\text{model}}(\boldsymbol{y})] &\leq \log\left(\mathbb{E}_{\boldsymbol{u}}[p_{\text{model}}(\boldsymbol{y})]\right), && (62)\\
&= \log P_{\text{model}}(X = \boldsymbol{x}_{\text{int}}) - \log(256). && (63)
\end{aligned}
$$

Rearranging this gives us a lower bound on the discrete log-likelihood:

$$
\log P_{\text{model}}(X = \boldsymbol{x}_{\text{int}}) \geq \mathbb{E}_{\boldsymbol{u}}[\log p_{\text{model}}(\boldsymbol{y})] + \log(256). \tag{64}
$$

Therefore, to ensure we are optimizing a valid lower bound on the true log-likelihood of the discrete data, we apply a correction to the pixel-wise reconstruction log-likelihood by adding a constant $\log(256)$ to it.

C.2 Model Architecture

In this section, we detail the model architecture used in the experiments in section 3 and 4.

C.2.1 Fixed Variance Experiment

In section 3, we conduct a controlled experiment with a fixed output variance, the VAE consists of a convolutional encoder and a simple linear decoder. The encoder architecture, which is shared across both continuous and discrete latent space models, is detailed in Table 5. The decoder is a single fully-connected linear layer that maps the latent variable $\boldsymbol{z}$ directly to the flattened output image pixels. Equivalent to a learnable bias, we augment the latent vector $\boldsymbol{z}$ by concatenating it with an additional dimension fixed at a constant value of 1.

Table 5: Encoder architecture for the fixed and learnable variance experiments on MNIST.

Layer	Kernel Size	Stride	Padding	Activation
Conv2d	3x3	1	1	ReLU
Conv2d	3x3	1	1	ReLU
Conv2d	3x3	1	1	-
Flatten	-	-	-	-
Linear	-	-	-	-

C.2.2 Learnable Variance Experiment

MNIST. The encoder for the MNIST experiments is consistent with fixed-variance experiment, as presented in Table 5. The nonlinear decoder mirrors this structure with a linear layer followed by several convolutional layers. The architecture is detailed in Table 6. The linear decoder is a single fully-connected layer without any activations.

Table 6: Nonlinear Decoder architecture for the learnable variance experiments on MNIST.

Layer	Kernel Size	Stride	Padding	Activation
Linear	-	-	-	-
Reshape	-	-	-	-
Conv2d	3x3	1	1	ReLU
Conv2d	3x3	1	1	ReLU
Conv2d	3x3	1	1	ReLU
Conv2d	3x3	1	1	ReLU
Conv2d	1x1	1	0	-

ImageNet Architecture and CIFAR-10 For the more complex CIFAR-10 and ImageNet datasets, we use deeper, strided convolutional architectures for both encoder and the nonlinear decoder, with batch normalization after each convolutional layers. The linear decoder remains a single fully-connected layers. The architectures are detailed in Table 7 and Table 8.

C.3 Training Details

C.3.1 Data Preprocessing

For all experiments, the input images are first transformed into PyTorch tensors. Before being passed to the model, the image data is scaled by $\frac{255}{256}$, in preparation for uniform dequantization.

C.3.2 Baselines

For our baseline models used throughout the following experiments, we use standard implementations for the Gumbel-Softmax and the reparameterization trick in VAEs. For the REINFORCE, we implement a baseline to reduce variance. Specifically, we use the running average of the reconstruction loss.

Table 7: Encoder architecture for the learnable variance experiments on CIFAR-10 and ImageNet.

Layer	Kernel Size	Stride	Padding	Activation
Conv2d	4x4	2	1	ReLU
BatchNorm2d	-	-	-	-
Conv2d	4x4	2	1	ReLU
BatchNorm2d	-	-	-	-
Conv2d	4x4	2	1	ReLU
BatchNorm2d	-	-	-	-
Conv2d	4x4	1	0	ReLU
BatchNorm2d	-	-	-	-
Flatten	-	-	-	-
Linear	-	-	-	-

Table 8: Nonlinear Decoder architecture for the learnable variance experiments on CIFAR-10 and ImageNet.

Layer	Kernel Size	Stride	Padding	Activation
Linear	-	-	-	-
Reshape	-	-	-	-
ConvTranspose2d	4x4	1	0	ReLU
BatchNorm2d	-	-	-	-
ConvTranspose2d	4x4	2	1	ReLU
BatchNorm2d	-	-	-	-
ConvTranspose2d	4x4	2	1	ReLU
BatchNorm2d	-	-	-	-
ConvTranspose2d	4x4	2	1	-

C.3.3 Fixed Variance Experiment

All models in this experiment are trained using the AdamW optimizer with betas set to $(0.9, 0.95)$. We performed a hyperparameter search for the learning rate over the values $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-5}, 5 \times 10^{-5}\}$. For each model, the best performing rate was selected based on the final BPD score, which was evaluated on the validation set at the end of Epoch 500. All reported metrics in Section 3.2 are likewise evaluated on the validation set at this same epoch.

The output variance of the decoder was fixed for these experimented. We tested σ^2 values of $\{0.1, 0.05, 0.01\}$ and found that a fixed variance $\sigma^2 = 0.01$ yielded the best results across all models. The models are trained with a batch size of 64, and no gradient clipping was applied. Additionally, we did not use KL annealing; the β parameter for KLD is fixed at 1.0 throughout training.

Gradient Variance Calculation The gradient variance with respect to the encoder parameters reported in Table 1 is measured empirically. To isolate the variance only from the latent variable sampling, we first perform a single forward pass through the encoder on a fixed batch of data to obtain the parameters of the latent distribution, which is to avoid the randomness introduced by the uniform noise we add for dequantization. With these parameters held constant, we then draw 100 latent samples from this fixed distribution. For each sample, we compute the corresponding reconstruction loss and backpropagate to get a gradient vector with respect to the encoder's parameters. The total gradient variance is computed by first calculating the variance for each individual parameter in the encoder across the 100 gradient samples. These per-parameter variances are then summed together to produce the final scalar value that is reported in Table 1.

C.3.4 Learnable Variance Experiment

The training scheme is similar to the fixed variance experiment. All models are trained using AdamW optimizer(s) with betas of $(0.9, 0.95)$, and we selected the best learning rate for each baseline from the set $\{1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-5}, 5 \times 10^{-5}\}$. For our combined methods that integrate Silent Gradients, we built upon the best baseline learning rates and introduced separate optimizers for the linear decoder's μ and α components. We perform a hyperparameter search for the annealing rate from $\{1 \times 10^{-2}, 5 \times 10^{-2}, 1 \times 10^{-3}, 5 \times 10^{-3}\}$, and the encoder freeze epoch (cut-off) from $\{50, 80, 100, 150, 200\}$.

The training duration and batch sizes varied by dataset:

MNIST. Models are trained with a batch size of 64. The REINFORCE models are trained for 300 epochs, while all others are trained for 200 epochs.

ImageNet. Models are trained for 100 epochs with a batch size of 128.

CIFAR-10. Models are trained for 2000 epochs with a batch size of 256.

At the end of training, all models are guaranteed to be converged. And like the fixed variance experiment, no KL annealing or β parameter for KLD other than 1.0 is used.

C.3.5 Computing Resources

All experiments included were conducted on 8 NVIDIA GeForce RTX 4090 GPUs.

C.3.6 Limitations

The experiment are only conducted on the three datasets mentioned above.

D Visualized Output

In this section, we provide the visualization of reconstructed images using the trained model at the epoch where the metrics are reported.

The visualizations are generated by taking a fixed batch of images from the validation set of each respective dataset. These images are passed through the train encoder to obtain a latent variable $\boldsymbol{z}$, which is then passed to the corresponding decoder to produce the reconstruction. For the learnable variable experiment in which a dual-decoder setting is introduced, only nonlinear decoder is used to generate the reconstruction. The linear decoder used for computing the Silent Gradient, in this case, is not used.

D.1 Fixed Variance Experiment

The visualizations show the original images and the corresponding reconstructed mean, as in Figure 2.

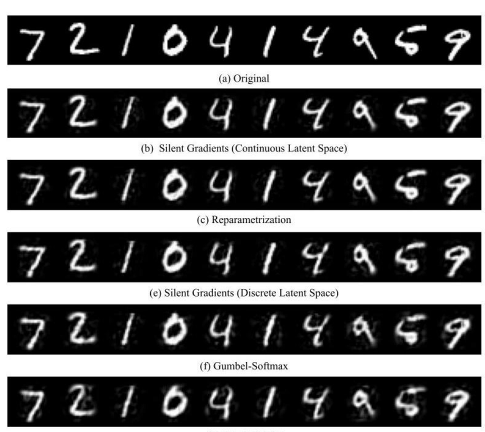


Figure 2: Visual comparison of reconstructions for the fixed variance experiment on MNIST. The top row (a) displays original images from the validation set. Subsequent rows show the reconstructed means from our Silent Gradients method and the baseline estimators for both continuous and discrete latent spaces.

D.2 Learnable Variance Experiment

In addition to the reconstructed mean, the visualizations include an additional row displaying the learned standard deviation for each pixel. For visualization purposes, the standard deviation is normalized to the range $[0, 1]$ to be displayed as an image, as shown in Figure 3, Figure 4, and Figure 5.

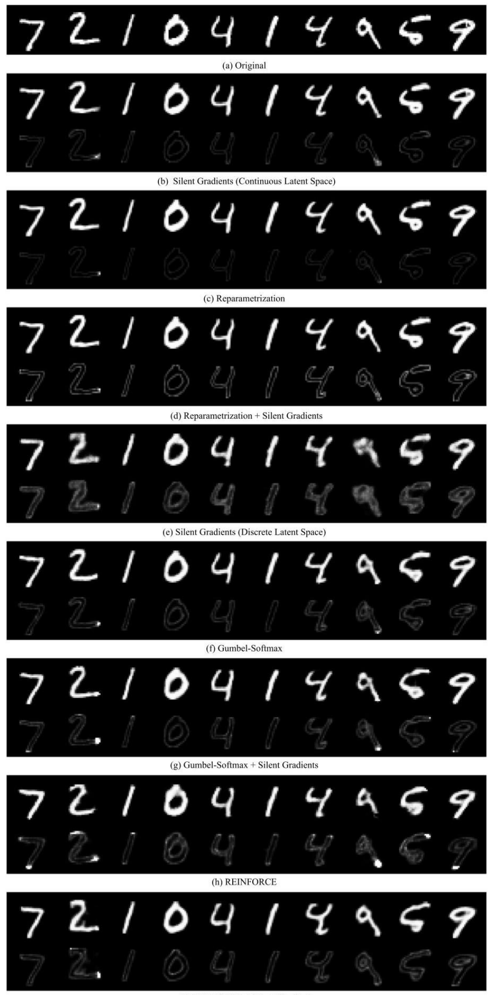
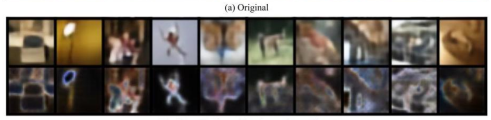
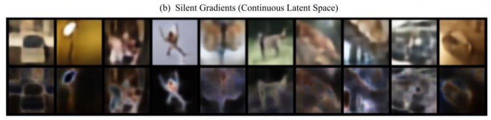


Figure 3: Reconstructions on the MNIST dataset in learnable variance experiment. The images are the output of the nonlinear decoder.

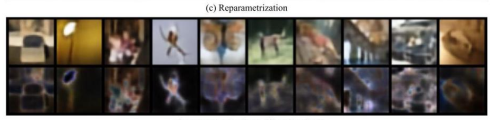
(a) Original



(b) Silent Gradients (Continuous Latent Space)

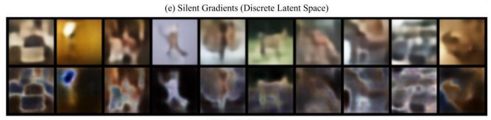


(c) Reparametrization

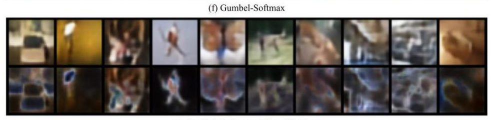


(d) Reparametrization + Silent Gradients

(e) Silent Gradients (Discrete Latent Space)

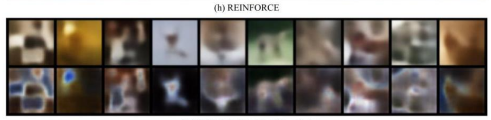


(f) Gumbel-Softmax



(g) Gumbel-Softmax + Silent Gradients

(h) REINFORCE



(I) REINFORCE + Silent Gradients

Figure 4: Reconstructions on the ImageNet dataset in learnable variance experiment.

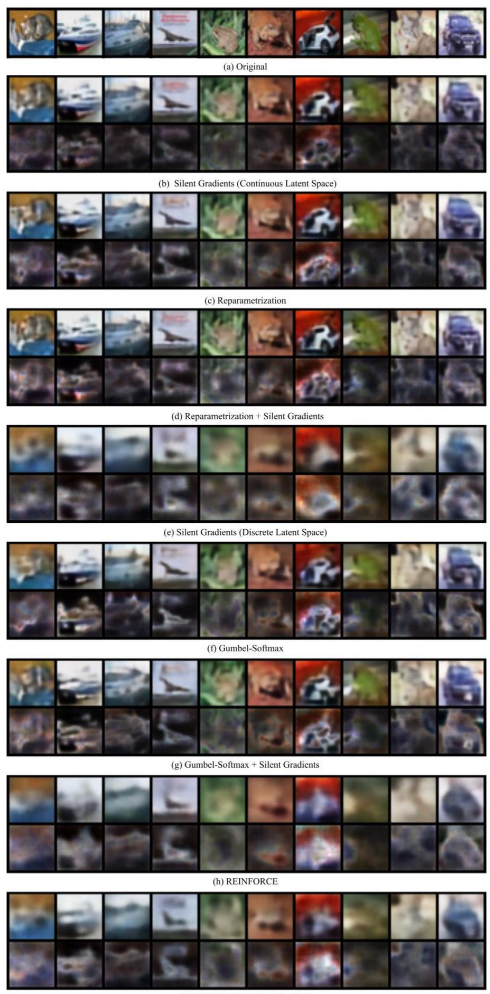


Figure 5: Reconstructions on the CIFAR-10 dataset in learnable variance experiment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are supported with the main text and appendices. The contributions and scope are clearly stated in abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the empirical results are addressed in the training detail section in Appendix C.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We state the theorem and proposition in the main text, and include the full derivation in Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include all the experimental setup and training details in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Although we do not provide the code at the moment, since the experimental setup is detailed in Appendix C, the results should be reproducible.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We include the training details, including optimizer choice, hyperparameter choosing criteria, in Appendix C

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We did not report the statistical significance in the main text, but the random seed strategy and comprehensive hyperparameter tuning in Appendix C should ensure the significance of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We list computing recourses in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm the research forms the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We address the applications of the technique in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There's no such risk in our work.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets we use in the main text and all of them are open datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.