Investigating the Functional Role of Learned Channel Suppression in Diffusion VAEs

Eugenie Shi
Department of Computer Science, Stanford University
yqshi@stanford.edu

Herry Wang
Department of Computer Science, Stanford University
zezhiw@stanford.edu

Oleg Roshka
Department of Computer Science, Stanford University
oros@stanford.edu

Abstract

Variational Autoencoders (VAEs) in large-scale diffusion models like Stable Diffusion XL (SDXL) often exhibit extensive "channel suppression," where numerous channels become inactive. This learned behavior, primarily mediated by Group Normalization scale parameters, is distinct from traditional neuron death. This paper investigates the functional role of this suppression. We fine-tuned the SDXL-VAE on diverse datasets—ImageNette-256, CIFAR-10, and Google Fonts—and employed a methodology involving channel activity tracking, classification of suppressed channels, and a "nudging" intervention to reactivate them by modifying GroupNorm scales. Qualitative analysis included activation grids and an adapted Logit Lens technique. Our experiments reveal that the intervention's impact is data-dependent: on complex natural image datasets (ImageNette, CIFAR-10), nudging modestly improved reconstruction quality (MSE, PSNR, SSIM)[cite: 1, 2, 5, 6]. Conversely, on the low-entropy Google Fonts dataset, nudging did not improve reconstruction but significantly lowered KL divergence[cite: 3, 4], suggesting that extensive suppression might be an effective specialization for simpler data. These findings indicate that learned channel suppression is a context-dependent mechanism, potentially acting as beneficial pruning on simple data but a recoverable loss of capacity on complex data. This work offers insights into VAE training dynamics and paths toward more robust generative models.

1. Introduction

Variational Autoencoders (VAEs) are a cornerstone of modern generative modeling, serving critical roles in tasks like image synthesis and compression. The VAE from Stable Diffusion XL (SDXL-VAE) [14], a widely used high-resolution image generation model, has been observed to develop extensive regions of inactive or "suppressed" channels within its convolutional layers during training or fine-tuning [5]. As documented by Gilman and others, a substantial portion of channels, particularly in deeper layers, can become effectively disabled by the model itself. This phenomenon is primarily mediated by the learned scale parameters of preceding Group Normalization (GroupNorm) [15] layers, which can reduce the output of entire channels to near-zero values. This suggests a form of learned, structured network self-modification, which is notably different from the well-known "dying ReLU" problem where neurons become permanently inactive due to consistently non-positive inputs [12, 1].

The central research question this project addresses is: **Is this self-imposed channel suppression a beneficial optimization strategy, a detrimental artifact, or a context-dependent phenomenon?** Understanding this mechanism is important because it could reveal insights into the learning dynamics of VAEs, potentially leading to more efficient network architectures, improved training stability, or better generalization. If suppression is beneficial, it might represent an implicit form of network pruning or specialization. If detrimental, it might limit representational capacity or indicate training issues that need mitigation.

To investigate this, our project aims to:

- Characterize the conditions under which channel suppression emerges during VAE fine-tuning, particularly in response to different dataset characteristics.
- Determine the functional impact of these inactive channels on VAE performance, such as reconstruction quality and latent space properties.
- Explore methods to selectively reactivate ("nudge") these suppressed channels and evaluate the consequences on model behavior and performance.

Our approach involves fine-tuning pre-trained SDXL-VAE models on standard image datasets. We employ a pipeline that includes dynamic channel activity tracking, classification of channels based on their activity levels, targeted interventions to modify parameters of suppressed channels, and qualitative/quantitative evaluation. We present preliminary findings indicating that dataset characteristics significantly influence the extent of channel suppression, and we outline our methods for deeper investigation. Ultimately, this work aims to shed light on the functional role of this intriguing emergent behavior in large-scale VAEs.

2. Related Work

Our investigation into learned channel suppression in VAEs draws upon several areas of deep learning research:

1. **Neuron Inactivity and "Dying ReLUs":** The "dying ReLU" problem describes how ReLU neurons can become permanently inactive if their input is always negative, effectively halting learning for that neuron [12, 1]. While related to inactivity, the channel suppression we observe in SDXL-VAE differs significantly. It appears to be a *learned* behavior actively modulated by the scale parameters of Group Normalization layers rather than a passive consequence of activation functions and input distributions. Furthermore, this suppression affects entire channels and is potentially reversible through targeted parameter adjustments, a key aspect our intervention methods explore.
2. **Network Pruning and Efficiency:** Significant research has focused on network pruning to create smaller, more efficient models by removing redundant weights, neurons, or channels [10, 6]. Some techniques aim for structured pruning, removing entire filters or channels [11]. The observed channel suppression might be an *implicit* form of learned pruning, where the model self-optimizes by down-weighting or disabling channels it deems unnecessary or detrimental for the given task and data. Dufort-Labbé *et al*. explore how networks can leverage neuron saturation for efficient pruning, which shares conceptual similarities with learned down-weighting [3].
3. **The Lottery Ticket Hypothesis:** Proposed by Frankle & Carbin, this hypothesis posits that dense, randomly initialized networks contain sparse subnetworks ("winning tickets") that, when trained in isolation, can achieve performance comparable to the original dense network [4]. The emergence of widespread suppressed channels could suggest that the remaining active channels form such a "winning ticket" for the VAE's objective, rendering the suppressed channels redundant for the specific data distribution. Our work explores whether these "redundant" channels can be repurposed or if their removal is indeed optimal.
4. **Normalization Layers and Controllability:** Group Normalization [15], the mechanism we identify as central to channel suppression, normalizes features within groups of channels. Its learnable scale and shift parameters (γ and β) allow the network to modulate the output of normalized features. Our focus is on how the scale parameter γ can be driven to near zero for specific channels, effectively silencing them.
5. **Artifacts and Redundancy in Generative Models:** Research into Generative Adversarial Networks (GANs) like StyleGAN has investigated sources of image artifacts, sometimes linked to network capacity or specific architectural choices [8]. Similarly, work on Vision Transformers has explored representational redundancy and the role of specific tokens or "registers" [2]. While our focus is VAEs, these studies highlight the broader themes of learned representations, redundancy, and model behavior in complex generative architectures.

The initial observations by Rudy Gilman on channel suppression in SDXL-VAE provided a direct catalyst for this project [5]. Our work aims to systematically investigate and understand this specific phenomenon within the SDXL-VAE framework.

3. Data

For this project, we utilize standard, publicly available image datasets to ensure reproducibility and to study the impact of data characteristics on channel suppression. We do not collect new datasets. The primary datasets used are:

- **ImageNette-256:** A 10-class subset of the full ImageNet dataset [7], specifically the version provided by FastAI that includes resized images. We use images processed to 256x256 pixels. ImageNette offers diverse natural image statistics, making it suitable for general VAE fine-tuning and as a benchmark for reconstruction quality on natural images. It contains approximately 13,000 training images and 500 validation images.
- **Google Fonts (`genfonts_data`):** This dataset consists of 256x256 pixel rasterizations of charac-

ters from the Google Fonts catalogue. We use the version available on Hugging Face Datasets (`rcugarte/genfonts_data`). This dataset represents a low-entropy, high-contrast domain, characterized by structured glyphs and relatively uniform backgrounds. It presents a contrasting data distribution to ImageNette, allowing us to investigate how data complexity influences channel suppression.

- **CIFAR-10:** A widely used dataset comprising 60,000 32x32 color images in 10 classes [9]. For our experiments, we resize these images to 256x256 pixels to match the input dimensions of the other datasets and the VAE's typical operating resolution, presenting a case of upscaling simpler, lower-resolution images.

Data Pre-processing. The pre-processing steps for all datasets are standardized to be compatible with the SDXL-VAE architecture. These steps include:

1. **Resizing and Cropping:** Images are resized appropriately (e.g., shorter side to 256 pixels for ImageNette/Fonts, or directly to 256x256 for CIFAR-10) using bilinear interpolation and then center-cropped to 256x256 pixels.
2. **RGB Conversion:** All images are ensured to be in RGB format.
3. **Tensor Conversion:** PIL Images are converted to PyTorch tensors. This typically maps pixel values from the [0, 255] range to [0.0, 1.0].
4. **Normalization:** Pixel values are then normalized from the [0.0, 1.0] range to [-1.0, 1.0] using the formula $(x - 0.5)/0.5$. This is a standard input range for many pre-trained VAEs, including the SDXL-VAE.

Minimal or no data augmentation is used during fine-tuning of the pre-trained VAEs. The Hugging Face `datasets` library is used for loading and initial handling, and `torchvision.transforms` for pre-processing.

4. Methods

Our methodology for investigating learned channel suppression in the SDXL-VAE integrates several components: (1) fine-tuning the VAE on different datasets, (2) tracking channel activity and model parameters, (3) classifying channels as suppressed, (4) applying targeted interventions to reactivate channels, and (5) visualizing and analyzing channel behavior. The entire pipeline is implemented in PyTorch, leveraging the Hugging Face `diffusers` and `accelerate` libraries.

4.1. Model and Baseline Training

The core model under investigation is the SDXL-VAE [14], specifically using the pre-trained weights from `stabilityai/sdxl-vae` available on Hugging Face. A dedicated model wrapper facilitates loading and, crucially, provides an interface for attaching hooks to arbitrary layers for activation capturing.

Baseline models are established by fine-tuning the pre-trained SDXL-VAE on our chosen datasets (ImageNette-256 and Google Fonts) without any specific interventions targeting channel suppression. Standard VAE loss is used, comprising a reconstruction term and a KL divergence term to regularize the latent space, weighted by a factor β_{KL}:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - \beta_{KL} \cdot D_{KL}(q(z|x)||p(z))$$

Typically, $p(z)$ is a standard normal distribution $\mathcal{N}(0, I)$, and $q(z|x)$ is the encoder's output distribution (e.g., $\mathcal{N}(\mu_z, \Sigma_z)$). The reconstruction term, $-\mathbb{E}_{q(z|x)}[\log p(x|z)]$, is often implemented as an MSE loss for Gaussian observation models: $\mathcal{L}_{\text{MSE}} = ||x - \hat{x}||_2^2$, where $\hat{x}$ is the reconstructed input. The performance of these baseline models serves as a reference against which models with interventions are compared.

4.2. Tracking Channel Activity

To understand when and where channels become suppressed, we employ two main tracking mechanisms during fine-tuning:

Activation Monitoring. An activation monitoring component attaches PyTorch forward hooks to specified layers within the VAE. It can capture layer inputs or outputs. For each hooked layer, it calculates and logs metrics at a defined tracking interval. Key metrics include:

- **Mean Absolute Activation Per Channel:** For a given activation tensor $A \in \mathbb{R}^{B \times C \times H \times W}$ (Batch, Channels, Height, Width), this metric computes the mean of the absolute activation values across spatial and batch dimensions for each channel $c \in \{1, \ldots, C\}$:

$$m_c = \frac{1}{B \cdot H \cdot W} \sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{w=1}^{W} |A_{b,c,h,w}|$$

  This results in a vector $\mathbf{m} \in \mathbb{R}^C$, providing a measure of each channel's overall activity and is the primary metric used by our channel classifier.
- **Full Activation Map:** The entire activation tensor A (detached and moved to CPU) is stored. This is used for qualitative visualization, such as generating the per-channel activation grids (e.g., as depicted in Figure 2).

Other statistics like overall mean and standard deviation of activations are also logged for general monitoring. The collected data is aggregated and stored, forming the basis for subsequent classification and analysis.

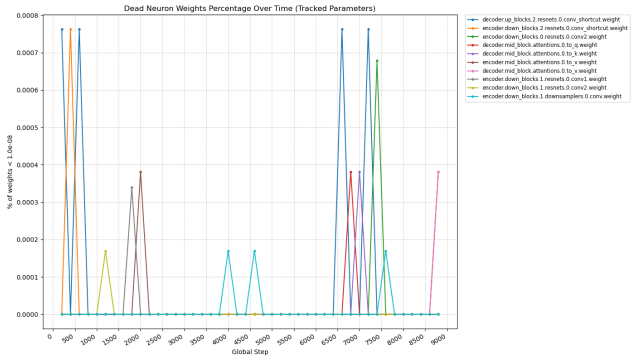


Figure 1: Dynamics of Near-Zero Channel Weights During ImageNette Training (baseline - no intervention). Percentage of weights with values below 1×10^{-8} over 30 training epochs for various convolutional layers in the SDXL-VAE. Different layers exhibit varying trends and volatility.

Weight-based Dead Neuron Tracking. While our primary focus is on GroupNorm-mediated suppression, we also include a tracking mechanism to monitor the percentage of near-zero weights in convolutional and linear layers directly. Figure 1 illustrates the dynamics of such near-zero weights during ImageNette training, showing the percentage of weights with values below 1×10^{-8} over 30 epochs for various convolutional layers. This helps distinguish the broader phenomenon of low-magnitude weights from the specific channel suppression mechanism. This tracker uses a fixed absolute threshold or a threshold relative to the mean weight magnitude to classify individual weights as "dead."

4.3. Classification of Suppressed Channels

A dedicated channel classification component identifies channels that have become suppressed, specifically targeting Group Normalization layers, as these are hypothesized to be the primary mechanism for learned channel suppression.

Its operation involves:

1. **Mapping Monitored Layers to Parameters:** On initialization, it builds a map from the activation monitor's layer identifiers (e.g., `vae.encoder.down_blocks.0.norm1.output`) to the corresponding GroupNorm module's scale parameter name (e.g., `vae.encoder.down_blocks.0.norm1.weight`) and its number of channels within the model. This allows direct access to the relevant parameters for intervention.
2. **Processing Tracked Data:** At each classification step, it receives the aggregated metrics from the activation monitor for the current global training step.
3. **Thresholding for Inactivity:** For each targeted GroupNorm layer, it uses the mean absolute activation per channel vector $\mathbf{m}$. If a channel's value m_c falls below a configurable activation threshold θ_{act} (e.g., 1×10^{-3}), that channel c is classified as "inactive" or suppressed for that specific GroupNorm layer.
4. **Outputting Targets for Intervention:** The classifier outputs a dictionary where keys are layer identifiers. Each entry contains the actual name of the GroupNorm scale parameter and a list of indices corresponding to the channels identified as inactive within that layer.

4.4. Intervention: Nudging Suppressed Channels

Once suppressed channels are identified, an intervention handler applies targeted modifications to attempt to reactivate them. The primary strategy is a "gentle nudge" to the GroupNorm scale parameters:

1. **Receiving Targets:** It takes the output from the channel classifier.
2. **Accessing Parameters:** Using the provided scale parameter name (e.g., `vae.encoder.down_blocks.0.norm1.weight`), it retrieves the actual `torch.nn.Parameter` object $\gamma \in \mathbb{R}^C$ from the model.
3. **Applying Nudge:** For each inactive channel index c_{inactive} identified by the classifier, it modifies the corresponding scale value $\gamma_{c_{\text{inactive}}}$. The modification typically involves:
   - Multiplying the current scale value by a nudge factor f_{nudge} (e.g., 1.1 or 1.2): $\gamma'_{c_{\text{inactive}}} = \gamma_{c_{\text{inactive}}} \cdot f_{\text{nudge}}$.
   - Alternatively, adding a small additive nudge value a_{nudge} (e.g., 0.01): $\gamma'_{c_{\text{inactive}}} = \gamma_{c_{\text{inactive}}} + a_{\text{nudge}}$.

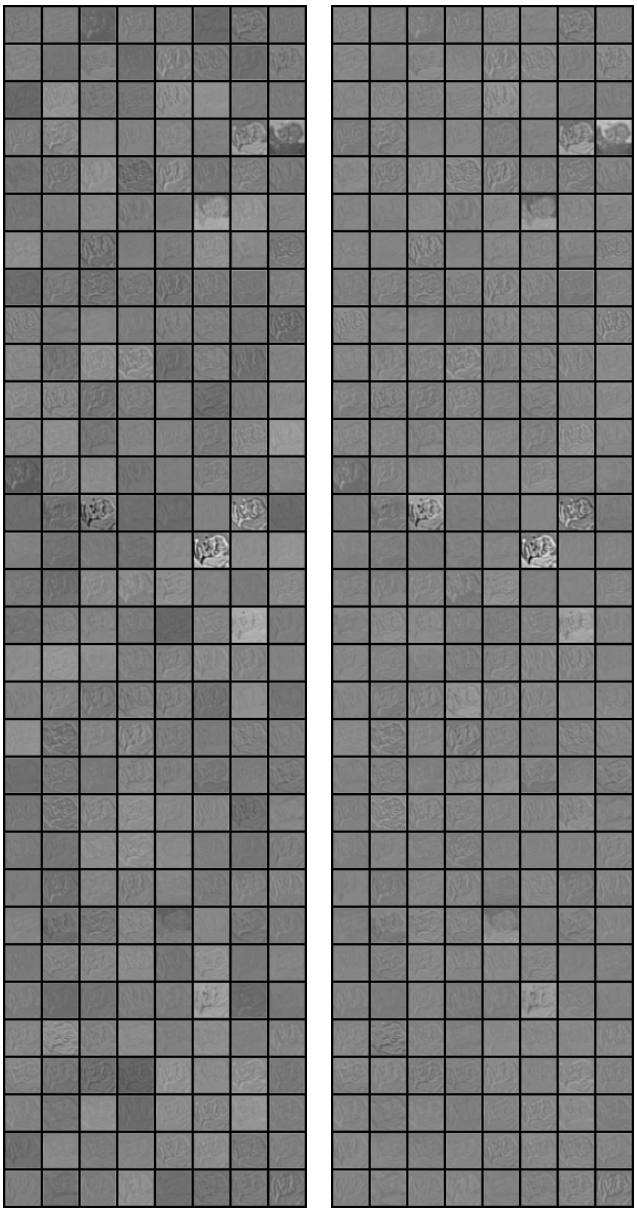
   The nudged value is then capped at a maximum scale value $\gamma_{\max}$ (e.g., 2.0): $\gamma''_{c_{\text{inactive}}} = \min(\gamma'_{c_{\text{inactive}}}, \gamma_{\max})$.
4. **Intervention Interval:** Interventions are applied periodically, controlled by an intervention interval (e.g., every 200 training steps), not at every step, to allow the network time to adapt.

An alternative strategy, "reset GroupNorm scale," simply resets $\gamma_{c_{\text{inactive}}}$ to 1.0. The goal of these interventions is to "encourage" the model to reuse these channels and observe the impact on performance and learning dynamics.

4.5. Visualization and Qualitative Analysis

To gain qualitative insights into what features channels are learning or how suppression manifests, we use two main visualization techniques:

Per-Channel Activation Grids. Figure 2 shows examples of activation grids from an encoder layer for a sample ImageNette validation image, processed by VAEs fine-tuned on ImageNette (left) and Google Fonts (right). This



(a) ImageNette-tuned VAE (b) Google Fonts-tuned VAE

Figure 2: Activation grids from an encoder layer for a sample ImageNette validation image, processed by VAEs fine-tuned on different datasets. Flat grey tiles denote suppressed channels. The Fonts-tuned model (right) suppresses a much larger fraction of channels compared to the ImageNette-tuned model (left).

method visualizes the activation maps captured by the activation monitor. For a given input image and layer, it plots the spatial activation map for each channel (or a subset of channels) as a separate tile in a grid. Each tile is individually normalized to the [0, 1] range for visualization (typically using a greyscale colormap such as 'viridis'). "Flat grey" tiles, prominently seen in the Fonts-tuned VAE, indicate channels with low variance or near-zero activations across their spatial dimensions, visually representing suppressed channels. This allows for direct comparison of channel activity patterns across models fine-tuned on different datasets.

Logit Lens for VAEs. Inspired by the Logit Lens technique used to interpret Language Models [13], we adapt this concept for VAEs. This approach uses a "mini-decoder" — a small stack of transposed convolutional layers — to project captured activation maps from intermediate VAE layers into an image-like space.

- **Projection Types:**
  - *Single-channel projection*: Takes a single channel's activation map (e.g., shape $1 \times 1 \times H \times W$), passes it through the mini-decoder, producing a small image patch. This helps hypothesize about the visual concept a specific channel might be sensitive to or trying to reconstruct.
  - *Full-map projection*: If the number of channels in an activation map matches the mini-decoder's input channel requirement, the entire map (e.g., $1 \times C \times H \times W$) can be projected.
- **Mini-Decoder Architecture:** The mini-decoder itself is a simple CNN, typically with a few transposed convolutional layers to upsample the feature map, interspersed with ReLU activations, and culminating in a Sigmoid to produce an output in the [0,1] range, interpretable as an image. Its input channel dimensionality is configurable based on the projection type.

This technique provides a visual hypothesis about the features encoded in specific channels or layers, complementing the quantitative analysis of channel activity. These visualizations are generated periodically during training and can also be generated during evaluation.

By combining these tracking, classification, intervention, and visualization methods, we aim to build a comprehensive understanding of the causes and consequences of learned channel suppression in the SDXL-VAE.

5. Experiments

To evaluate the functional role of suppressed channels and the impact of our intervention strategy, we conducted experiments fine-tuning the SDXL-VAE on three distinct datasets: ImageNette-256 (natural images), Google Fonts (low-entropy characters), and CIFAR-10 (upscaled low-resolution natural images). For each dataset, we compare two conditions:

- **Baseline:** Standard fine-tuning of the pre-trained SDXL-VAE.

- **Nudge Intervention:** Fine-tuning with our "gentle nudge" strategy applied periodically to the scale parameters of Group Normalization layers whose associated channels were classified as inactive.

Performance is primarily evaluated using reconstruction Mean Squared Error (MSE), KL Divergence (D_{KL}), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM). For MSE and D_{KL}, lower values are generally better, while for PSNR and SSIM, higher values indicate better performance.

5.1. Quantitative Results

Table 1 summarizes the key evaluation metrics on the test/validation splits for each dataset under both baseline and nudge intervention conditions.

Performance on Natural Image Datasets (ImageNette and CIFAR-10). For the ImageNette-256 dataset, the nudge intervention led to slight improvements across all metrics: MSE decreased from 0.010443 to 0.010396, D_{KL} from 3361.52 to 3351.61, PSNR increased from 25.833 to 25.853, and SSIM from 0.8077 to 0.8086[cite: 5, 6]. The training loss dynamics, particularly the average epoch reconstruction loss shown in Figure 3(a), also indicate that the nudge model consistently achieved slightly lower reconstruction loss during training on ImageNette; a similar trend was observed for the total training loss.

On CIFAR-10 (upscaled to 256x256), the nudge intervention similarly improved reconstruction quality: MSE decreased from 0.002879 to 0.002826, PSNR increased from 31.429 to 31.509, and SSIM from 0.9195 to 0.9210[cite: 1, 2]. The KL divergence was marginally higher for the nudge model (997.53 vs. 994.63)[cite: 1, 2]. Figure 3(b) illustrates that the nudge model maintained a lower validation reconstruction loss throughout much of the CIFAR-10 fine-tuning process. These results suggest that for datasets with diverse natural image content, reactivating potentially underutilized channels can be beneficial for representational capacity and reconstruction fidelity.

Performance on Low-Entropy Dataset (Google Fonts). The Google Fonts dataset, characterized by simpler, high-contrast glyphs, showed a different trend. The baseline model achieved slightly better reconstruction metrics: MSE was 0.002451 for baseline versus 0.002470 for nudge, PSNR was 32.138 versus 32.105, and SSIM was 0.9917 versus 0.9916[cite: 3, 4]. However, the nudge intervention resulted in a notably lower (better) KL divergence (6313.26 compared to 6363.02 for baseline)[cite: 3, 4]. The validation total loss curves for Google Fonts, shown in Figure 3(c), are very close, with the nudge model sometimes achieving lower total loss, likely benefiting from the reduced KL term. This suggests that on such low-entropy data, extensive channel suppression might be an effective strategy for the VAE to simplify its latent space, and nudging channels might slightly impair reconstruction possibly by reintroducing complexity not essential for the task, even while improving the KL term.

5.2. Discussion

The experimental results indicate that the impact of the nudge intervention is data-dependent. For more complex, natural image datasets like ImageNette and CIFAR-10, nudging suppressed channels appears to provide modest but consistent benefits in reconstruction quality (MSE, PSNR, SSIM). This supports the hypothesis that some channels might be prematurely or overly suppressed during fine-tuning on such data, and reactivating them allows the model to capture finer details or more diverse features. The training and validation loss curves (Figure 3(a) and (b)) corroborate this by showing improved reconstruction loss for the nudge model.

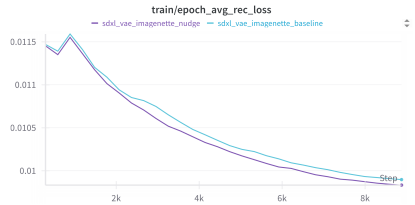
Conversely, on the highly structured and low-entropy Google Fonts dataset, the nudge intervention did not improve (and slightly worsened) reconstruction metrics, although it did lead to a better KL divergence. This suggests that for simpler data distributions, the VAE might effectively learn to prune unnecessary channels, and the baseline model's extensive channel suppression (as qualitatively observed in Figure 2) could be a form of beneficial specialization. Forcing these channels to remain active via nudging might reintroduce redundant or less optimal pathways for reconstruction, even if it aids in achieving a more compressed or regularized latent space as indicated by the lower D_{KL}.

The interplay between reconstruction quality and KL divergence is crucial. The nudge intervention can shift this balance differently depending on the dataset. The slight increase in KL for CIFAR-10 with nudge, despite better reconstruction, and the significant decrease in KL for Fonts with nudge, despite slightly worse reconstruction, highlight this complex relationship.

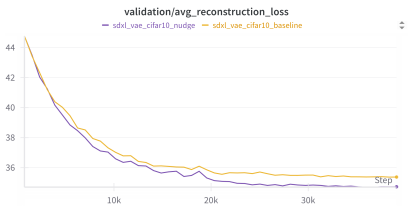
These findings open avenues for adaptive intervention strategies. For instance, interventions could be selectively applied based on dataset complexity, or the nudge factor could be dynamically adjusted. The qualitative visualizations from our Logit Lens approach (described in Section 4) are intended to further probe what features these reactivated channels begin to represent, though detailed analysis of these visualizations is part of ongoing and future work. The observed dynamics of near-zero weights (Figure 1) further confirm that significant portions of the network are subject to learning-induced inactivity, setting the stage for interventions like ours.

Table 1: Quantitative evaluation metrics for VAEs fine-tuned with and without the nudge intervention. MSE and D_{KL} are better lower; PSNR and SSIM are better higher. Best results for each metric within a dataset are bolded. Results for ImageNette are on its validation split[cite: 5, 6], others on test splits[cite: 1, 2, 3, 4].

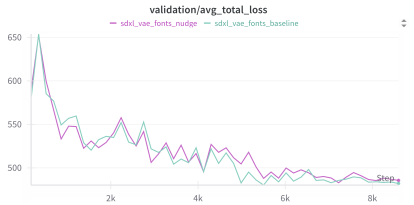
Dataset	Condition	MSE ($\times 10^{-3}$) ↓	D_{KL} ↓	PSNR (dB) ↑	SSIM ↑
ImageNette-256	Baseline	10.443 [cite: 5]	3361.52 [cite: 5]	25.833 [cite: 5]	0.8077 [cite: 5]
	Nudge	**10.396** [cite: 6]	**3351.61** [cite: 6]	**25.853** [cite: 6]	**0.8086** [cite: 6]
CIFAR-10	Baseline	2.879 [cite: 1]	**994.63** [cite: 1]	31.429 [cite: 1]	0.9195 [cite: 1]
	Nudge	**2.826** [cite: 2]	997.53 [cite: 2]	**31.509** [cite: 2]	**0.9210** [cite: 2]
Google Fonts	Baseline	**2.451** [cite: 3]	6363.02 [cite: 3]	**32.138** [cite: 3]	**0.9917** [cite: 3]
	Nudge	2.470 [cite: 4]	**6313.26** [cite: 4]	32.105 [cite: 4]	0.9916 [cite: 4]



(a) ImageNette: Train Epoch Avg. Rec. Loss



(b) CIFAR-10: Validation Avg. Rec. Loss



(c) Google Fonts: Validation Avg. Total Loss

Figure 3: Training and validation loss dynamics. (a) Average epoch reconstruction loss during ImageNette fine-tuning. (b) Average validation reconstruction loss for CIFAR-10. (c) Average validation total loss (Rec. + KL term) for Google Fonts. In (a) and (b), lower is better, showing a tendency for the nudge intervention to improve reconstruction loss on more complex datasets. In (c), the total loss curves are very close, with nudge slightly better at certain steps.

5.3. Qualitative Analysis of Channel Dynamics and Interventions

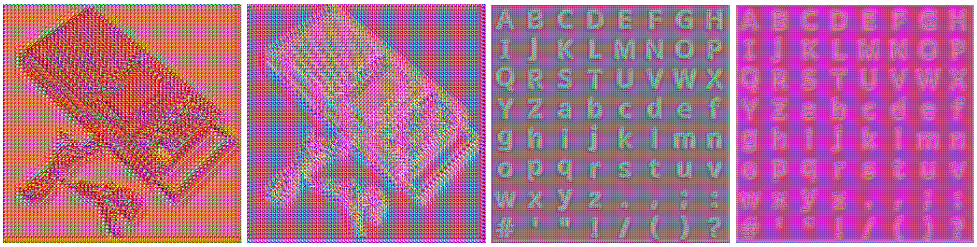
Beyond quantitative metrics, we performed qualitative analyses to better understand channel behavior.

Logit Lens Projections. Using our adapted Logit Lens technique with a full map projection through a mini-decoder, we visualized the aggregated feature representations from an intermediate encoder layer. Figure 4 presents these projections for both ImageNette and Google Fonts datasets, comparing baseline and nudge intervention models. For ImageNette (Figures 4a and 4b), the differences between baseline and nudge projections are subtle, suggesting that while quantitative improvements are observed with nudging, the overall high-level feature abstraction at this layer remains largely consistent. The projections capture abstract textural and structural information from the input image. For the Google Fonts dataset (Figures 4c and 4d), the projections clearly represent the grid of characters. Again, visual differences between baseline and nudge are not stark at this aggregated level, aligning with the observation that extensive channel suppression in the baseline Fonts model already leads to a highly specialized representation. These visualizations provide a global view of a layer's feature space; finer-grained analysis would involve single-channel projections as described in Section 4.

Intervention Dynamics. Figure 5 illustrates the dynamics of inactive channels and the application of the nudge intervention during the fine-tuning of the VAE on the Google Fonts dataset. The plot tracks the number of channels classified as inactive over training steps. The ticks indicate when interventions occurred. We observe that interventions often lead to a temporary decrease in the number of inactive channels. However, for a low-entropy dataset like Fonts, channels may become suppressed again as the model reconverges, highlighting the strong data-dependent drive towards specialization and sparsity. This visualization underscores the ongoing nature of the channel suppression phenomenon and the reactive role of the nudge intervention.

6. Conclusion

This project investigated the phenomenon of learned channel suppression in the SDXL-VAE, where channels be-



(a) ImageNette Baseline: Full Map Projection (b) ImageNette Nudge: Full Map Projection (c) Fonts Baseline: Full Map Projection (d) Fonts Nudge: Full Map Projection

Figure 4: Logit Lens full map projections from the VAE encoder layer *encoder.down_blocks[1].resnets[0].conv_shortcut* for a sample input. These visualizations project the entire activation map (all channels) from this specific layer through a mini-decoder to an image-like representation, offering a hypothesis about the layer's aggregated feature representation. Comparing baseline (left column) and nudge (right column) for ImageNette (top row) and Google Fonts (bottom row) can reveal subtle differences in learned features or attention at this stage of the encoder.

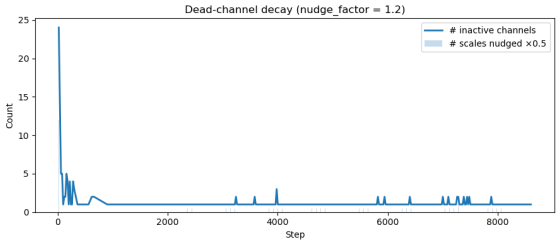


Figure 5: Dead Channel Decay and Intervention Dynamics on Google Fonts (Nudge Model). This plot shows the number of inactive channels detected over training steps (blue line) and the points at which the nudge intervention was applied (light blue ticks, scaled for visibility). It illustrates how interventions aim to reduce the count of inactive channels.

come inactive primarily due to the scaling down of Group Normalization parameters. Our core research question explored whether this suppression is a beneficial optimization or a detrimental artifact.

Key Findings. Our experiments, involving fine-tuning on ImageNette-256, CIFAR-10, and Google Fonts, demonstrated that:

- Channel suppression is highly data-dependent, with simpler, low-entropy datasets like Google Fonts exhibiting more extensive suppression (qualitatively supported by activation grids like those in Figure 2).
- Our "gentle nudge" intervention, designed to reactivate suppressed channels, modestly improved reconstruction quality (MSE, PSNR, SSIM) on complex natural image datasets (ImageNette-256 and CIFAR-10). This suggests that for such data, some channel suppression might be premature or limit representational capacity.
- For the low-entropy Google Fonts dataset, the nudge intervention did not improve reconstruction but did lead to a better (lower) KL divergence. This implies that for simpler data, aggressive channel suppression might be an effective learned regularization or specialization strategy.

These results suggest that the functional role of channel suppression is **context-dependent**. It may be a beneficial form of learned pruning for simpler data distributions but could represent a loss of useful capacity for more complex data.

Limitations. This study is based on the SDXL-VAE architecture and a specific "gentle nudge" intervention strategy. The findings might vary with other VAE architectures or different intervention techniques. While we qualitatively analyze channel activity, a deeper quantitative analysis of the specific features learned by reactivated channels is warranted.

Future Work. Several avenues for future research emerge:

- Developing more sophisticated, adaptive intervention strategies, perhaps guided by information-theoretic measures or the specific characteristics of the dataset.
- Employing advanced interpretability methods to precisely identify the functional roles of suppressed and reactivated channels.
- Investigating the impact of channel suppression and nudging on the downstream performance of diffusion models that utilize these VAEs.
- Exploring whether these learned suppression patterns can inform explicit, structured pruning techniques for more efficient VAEs.

- Formally connecting the observed phenomena to concepts like the Lottery Ticket Hypothesis or information bottleneck principles.

In summary, our work provides initial evidence that learned channel suppression in large VAEs is a nuanced, data-dependent process. Understanding and potentially controlling this phenomenon offers a promising direction for developing more robust, efficient, and adaptable generative models.

Ethical Considerations

The computational resources for this project, including model training and experimentation, were utilized responsibly. Training was performed on an NVIDIA RTX 5090 GPU. We estimate the total energy consumption for the experiments presented to be approximately 16 kWh. This energy was sourced primarily from renewable solar panel generation, minimizing the carbon footprint associated with the computational work. All datasets used (ImageNette-256, Google Fonts, CIFAR-10) are publicly available and widely used for academic research, and do not contain personally identifiable information or inherently biased content that would raise immediate ethical concerns for the scope of this VAE architecture study. Our research focuses on understanding internal model dynamics and does not directly involve applications with immediate societal impact that would require a broader ethical review at this stage.

Author Contributions

Oleg Roshka proposed the initial research idea, developed the core training and evaluation pipeline, and implemented the channel classification, intervention handler, and Logit Lens visualization components. Herry Wang implemented the mechanisms for gathering detailed dead neuron statistics and contributed to the design and implementation of the per-channel activation grid visualizations. Eugenie Shi conducted research into various data augmentation techniques and explored alternative analytical methods that could complement our primary investigation, contributing to the breadth of our background research. All authors collaborated closely on debugging, refining the experimental setup, analyzing results, and writing this report. This project was a significant team effort, with numerous discussions leading to improvements in methodology and interpretation.

Code Availability

The source code for this project, including the implementation of the VAE wrapper, tracking mechanisms, classifier, intervention handler, and Logit Lens, is publicly available on GitHub at: https://github.com/olegroshka/vae-channel-dynamics.

References

[1] AIML.com. What is the "dead relu" problem and, why is it an issue in neural network training?, September 2023. 1, 2

[2] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski. Vision transformers need registers. ArXiv e-print, 2023. 2

[3] S. Dufort-Labbé, P. D'Oro, E. Nikishin, R. Pascanu, P.-L. Bacon, and A. Baratin. Maxwell's demon at work: Efficient pruning by leveraging saturation of neurons. ArXiv e-print, 2024. 2

[4] J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. ArXiv e-print, March 2019. 2

[5] R. Gilman. Post on x regarding sdxl vae decoder channel observations. https://x.com/rgilman33/status/1909793815411437753, April 2025. Accessed on June 4, 2025. 1, 2

[6] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[7] J. Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. 2

[8] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan, 2019. 2

[9] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 3

[10] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in neural information processing systems (NIPS)*, volume 2, pages 598–605, 1990. 2

[11] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*, 2017. 2

[12] L. Lu, Y. Shin, Y. Su, and G. E. Karniadakis. Dying relu and initialization: Theory and numerical examples. *Communications in Computational Physics*, 28(5):1671–1706, 2020. 1, 2

[13] Nostalgebraist. Logit lens. *LessWrong*, 2020. 5

[14] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 3

[15] Y. Wu and K. He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 1, 2