Loss Attenuation Envelopes: A Loss-Agnostic Framework for Gradient Shaping via Value-Sensitive Thresholding

Tommy Barnes tommybarnesresearch@gmail.com

1 Aug 2025

Abstract

We introduce *Loss Attenuation Envelopes* (LAEs), a general method for shaping gradient magnitudes as a function of loss value, attenuating low-loss examples while preserving direction, thereby reducing gradient noise from low-signal samples. LAEs are loss-agnostic, easy to implement, and adapt dynamically via quantiles of the batch loss distribution, all with negligible computational overhead. We explore several envelope shapes and thresholding schemes, showing that LAEs consistently improve generalization in supervised, unsupervised, and reinforcement learning tasks. Notably, LAEs subsume the widely used *free bits* technique in variational autoencoders (VAEs), offering a principled generalization and delivering significant gains under the same training conditions.

1 Introduction

In gradient-based learning systems, each sample's gradient provides the direction toward the system's optima, but the distance from that optima is unknown. From static loss coefficients, to modified loss functions such as Huber-losses, and momentum based optimizers, choosing exactly how far to step in this gradient direction is a challenging and important decision. Loss Attenuation Envelopes provide an additional tool for tuning gradient magnitudes for better convergence and generalization by selectively attenuating low-loss gradients, preserving the direction of each sample, only modifying their magnitude.

While working on a reinforcement learning agent with a distributional latent state, choosing a free bits value for its KL-divergence to a gaussian prior proved challenging. Free bits prevent posterior collapse, but tuning the threshold was difficult, domain-dependent, and in the face of distributional shift during training, static values seemed inadequate. During this process, we began to view free bits through a different lens: as a generic mechanism that suppresses gradients for low loss elements. This perspective led to a broader question—*could the same idea be applied to any loss function, not just KL?*

This insight led to a more general framework: *Loss Attenuation Envelopes* (LAEs). Rather than modifying the loss itself, LAEs scale each individual loss element by a detached weight $w \in [0, 1]$, determined by applying a *loss shape function* to the loss value. This provides a simple, flexible mechanism for attenuating gradients from uninformative or low-loss examples—regardless of the semantics of the underlying loss.

LAEs apply a detached scalar weight $w \in [0, 1]$ to each loss element based on a *loss shape function*. This function defines how aggressively gradients should be attenuated across different loss values.

Unlike hard-coded clamping thresholds, LAEs support *adaptive envelopes*, where the parameters of the shape function are dynamically tuned using the empirical distribution of loss values in each batch. In particular, we use **exponential moving averages (EMA) over the batch-wise loss CDF** to set thresholds (e.g., the 90th percentile), allowing LAEs to remain calibrated to the evolving loss landscape. This makes LAEs far easier to tune than fixed thresholds and provides consistent gains across domains.

We propose Loss Attenuation Envelopes as a general framework for value-sensitive gradient shaping. Our key contributions include:

- A unified formulation for envelope-based loss weighting applicable to any elementwise loss
- A family of envelope shapes (step, linear, sigmoid) with static and **EMA-adaptive** thresholding
- Empirical evaluation across supervised, unsupervised, and reinforcement learning tasks
- Preliminary analysis of the effects of LAEs on generalization and optimization dynamics

2 Related Work

Loss Thresholding in VAEs: *Variational Autoencoders* (VAEs) often suffer from *posterior collapse*, where the latent code carries little information because the KL-divergence term dominates training. Prior works have addressed this by modifying the VAE training objective. Bowman et al. (2015) and Sønderby et al. (2016) proposed *KL annealing*, gradually increasing the weight of the KL term from 0 to 1 over the course of training. Kingma et al. (2016) introduced the *"free bits"* technique as an alternative: it places a lower bound on the KL term (per latent dimension) by effectively treating any KL value below a threshold as if it were at that threshold. In other words, once the KL-divergence for a latent falls below a set target, the model stops optimizing it further. This ensures each latent dimension encodes at least a minimum amount of information, preventing collapse. Our proposed *Loss Attenuation Envelopes (LAEs)* generalize this idea beyond VAEs – instead of applying a threshold only to the KL term, we apply a thresholding/attenuation mechanism to arbitrary loss components. LAEs dynamically reduce the influence of losses that fall below a certain value, analogous to free bits but for *any* loss. Moreover, whereas free bits use a fixed preset threshold, LAEs determine this cutoff **dynamically** via a quantile of the batch loss distribution. This quantile-based adaptive thresholding is, to our knowledge, novel in that it allows the "loss floor" to adjust itself to the current training dynamics rather than being manually tuned.

Sample Reweighting and Hard-vs-Easy Emphasis: Outside of VAEs, there is extensive work on dynamically reweighting training samples based on their loss values. A number of approaches have argued for focusing on "hard" examples (those with higher loss) to accelerate training. For instance, in image classification, methods by Loshchilov & Hutter (2015), Katharopoulos & Fleuret (2018), and Jiang et al. (2019) prioritized high-loss or high-gradient samples during mini-batch selection. By upsampling or giving more weight to high-loss cases, these methods aimed to speed up convergence on challenging examples. However, many early implementations incurred non-trivial overhead – e.g. requiring extra forward/backward passes or maintaining per-sample difficulty metrics – and were primarily designed for multi-epoch training regimes. More recent strategies have sought to make loss-based reweighting lightweight. Ren et al. (2018) used a meta-learning approach to adjust sample weights based on a validation set (effective for noisy and imbalanced data). These introduce additional complexity (nested optimization or secondary models). In contrast, *focal loss*

(Lin et al., 2017) provides a simpler *in-line* solution in supervised learning: it reshapes the cross-entropy loss to *down-weight well-classified (low-loss) examples*, focusing training on the harder misclassified examples. Our LAE approach shares a similar spirit of emphasizing harder samples, but achieves it through the aforementioned adaptive threshold on the loss values, rather than a fixed formula as in focal loss.

Very recently, research in large-scale language model training has reinforced the value of de-emphasizing already easy (low-loss) examples. Sow et al. (2025) propose a *dynamic loss-based sample reweighting* scheme for LLM pre-training that adjusts each sample's weight based on its current loss, *"allowing the model to dynamically focus on more informative samples"*. They specifically find that **deprioritizing low-loss, presumably redundant or less informative data, yields better efficiency**, as it reduces repeated training on examples the model already knows. Their methods are designed to add little or no computational overhead compared to standard training, aligning with the practical goals of our work. LAEs fit into this landscape as a *lightweight, on-the-fly* reweighting mechanism: by using a quantile-based cutoff, we *automatically* treat the lowest-loss samples in each batch as "effectively learned" (attenuating their impact) and allow the model to concentrate on higher-loss examples that contribute more learning signal. Importantly, this is done without requiring any additional passes or auxiliary models – the quantile computation is inexpensive and uses information already available from the batch's loss values.

3 Methods

3.1 Overview

A Loss Attenuation Envelope (LAE) defines a weighting function $w(\ell) \in [0, 1]$ that scales the gradient of an individual loss element ℓ. The weight is computed from the loss value itself, detached from the computation graph, meaning the envelope modifies the *magnitude* but not the *direction* of the gradient. Importantly, we require that $w(\ell)$ is monotonically non-decreasing in ℓ, ensuring that lower-loss examples receive equal or greater attenuation than higher-loss ones. This preserves the relative importance of high-loss samples and avoids reordering gradients, distinguishing LAEs from techniques like gradient clipping or robust losses.

The general form of a loss modified by a LAE is:

$$\mathcal{L}_{\text{LAE}} = \sum_i w(\ell_i) \cdot \ell_i$$

where ℓ_i is the loss for a single sample or element, and $w(\ell_i)$ is the envelope weight. LAEs can be applied to scalar losses or to vectorized/multi-element losses such as per-dimension KL divergences.

3.2 Envelope Shapes

We support several shape functions $w(\ell)$ that define how the attenuation evolves with the loss value:

Step Envelope.

$$w(\ell) = \begin{cases} 0 & \ell < T \\ 1 & \ell \geq T \end{cases}$$

This enforces a hard cutoff: only samples with loss above threshold T contribute to the gradient.

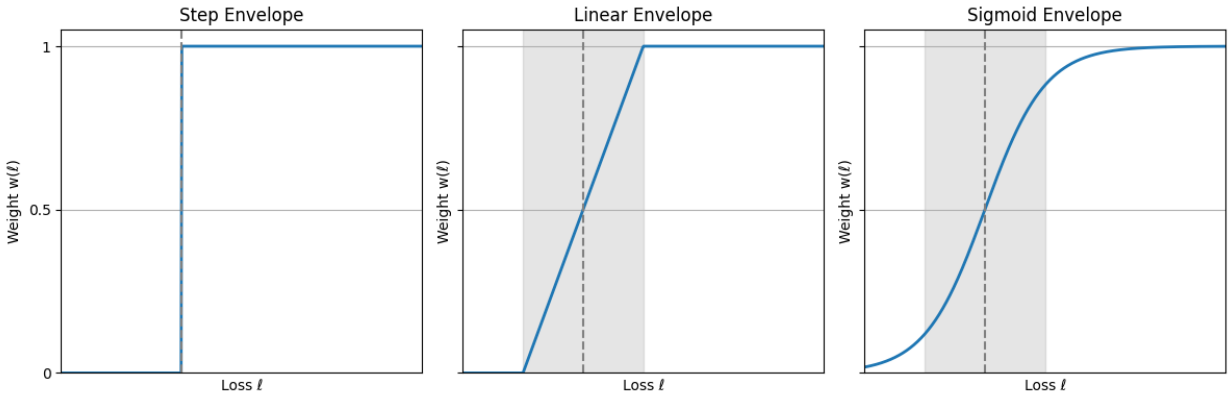

Figure 1: Illustration of three Loss Attenuation Envelope (LAE) shapes: step, linear, and sigmoid. The x-axis denotes the per-sample loss ℓ, and the y-axis shows the resulting gradient weight $w(\ell) \in [0, 1]$. The dashed vertical line indicates the threshold T, and the shaded region represents the width parameter that defines the transition region for linear and sigmoid envelopes.

Sigmoid Envelope.

$$w(\ell) = \frac{1}{1 + e^{-\beta(\ell - T)}}$$

where $\beta = \frac{4}{\text{invSlope}}$ Here, β controls the sharpness of the transition. This smooth approximation to the step envelope is differentiable and tunable. The parameter `invSlope` defines the width of the transition, with $\beta = \frac{4}{\text{invSlope}}$ setting the maximum slope.

Linear Envelope.

$$w(\ell) = \begin{cases} 0 & \ell \leq T - \frac{1}{2} \cdot \text{invSlope} \\ \frac{1}{\text{invSlope}} \cdot (\ell - T) + \frac{1}{2} & |\ell - T| \leq \frac{1}{2} \cdot \text{invSlope} \\ 1 & \ell \geq T + \frac{1}{2} \cdot \text{invSlope} \end{cases}$$

The ‘invSlope‘ defines the width of the linear region. The central point of the ramp is located at threshold T, and the gradient of the ramp is $1/\text{invSlope}$. This shape smoothly interpolates from 0 to 1 over a finite range and becomes the identity function when invSlope $\to 0$.

3.3 Static vs Adaptive Envelopes

Static Envelopes require manual selection of T and shape parameters. However, choosing appropriate static thresholds requires prior knowledge of the loss distribution. Since the distribution may shift during training or across tasks, static configurations are often brittle.

Adaptive Envelopes address this by estimating the loss distribution per batch and adjusting T accordingly. We compute a loss percentile (e.g., 90th) over the batch, and use an exponential moving average (EMA) to smooth the threshold over time. This allows LAEs to track evolving training dynamics and remain calibrated even in non-stationary environments.

3.4 Multiloss and Multitask Settings

LAEs can be applied independently to each loss term in a multi-objective setting (e.g., value and policy loss in reinforcement learning, or reconstruction and KL in a VAE). The envelope shapes, thresholds, and parameters can be customized per loss, and need not be shared.

3.5 Pseudocode

Algorithm 1 Adaptive Step Envelope

Require: Loss vector $\ell_1, \ldots, \ell_B$, quantile threshold $T \in [0, 1]$, step size $\alpha \in (0, 1]$, running threshold estimate $\hat{\tau}$

Compute $q_T \leftarrow \text{Quantile}(\{\ell_i\}_{i=1}^B, T)$
Update running threshold: $\hat{\tau} \leftarrow (1 - \alpha) \cdot \hat{\tau} + \alpha \cdot q_T$
for each sample $i = 1$ to B **do**
  if $\ell_i > \hat{\tau}$ **then**
    $w_i \leftarrow 1$
  else
    $w_i \leftarrow 0$
  end if
  $\tilde{\ell}_i \leftarrow w_i \cdot \ell_i$
end for
return Shaped losses $\tilde{\ell}_1, \ldots, \tilde{\ell}_B$ and updated $\hat{\tau}$

Note: In practice, w_i is treated as a fixed mask with no gradient flow (e.g., via `detach()` in PyTorch), so gradients propagate only through the surviving ℓ_i values.

Algorithm 2 Training Loop with Loss Attenuation Envelope

for each training step **do**
  Sample batch of inputs $x_1, \ldots, x_B$ and targets $y_1, \ldots, y_B$
  Compute model predictions $\hat{y}_1, \ldots, \hat{y}_B$
  Compute per-sample losses $\ell_i \leftarrow \text{Loss}(y_i, \hat{y}_i)$
  Apply envelope: $\tilde{\ell}_1, \ldots, \tilde{\ell}_B \leftarrow \text{Envelope}(\ell_1, \ldots, \ell_B)$
  Compute mean loss: $\tilde{L} \leftarrow \frac{1}{B} \sum_{i=1}^B \tilde{\ell}_i$
  Backpropagate and update model parameters using $\tilde{L}$
end for

3.6 Design Notes and Practical Considerations

- LAEs are computationally inexpensive. Quantile computation and envelope functions add minimal overhead.
- The threshold T is derived from the loss distribution and is interpretable.
- When applied per-element (e.g., KL per latent dimension), LAEs can treat each component independently.
- If $T = 0$, all shapes reduce to the identity envelope. Similarly, invSlope $= 0$ reduces linear/sigmoid to identity.

- Choosing envelope shapes and parameters remains an open hyperparameter selection problem. Our empirical results provide guidance, but more work is needed to automate tuning.

3.7 Terminology and Intuition: Envelopes as Signal Shaping

The term *Loss Attenuation Envelope* is inspired by envelope functions in audio and signal processing, such as ADSR envelopes (Attack, Decay, Sustain, Release). In that setting, an envelope defines the amplitude of a sound signal as a function of time, shaping how it is heard without altering its underlying content.

LAEs similarly define *gradient amplitude as a function of loss magnitude*. Each envelope shape—step, linear, or sigmoid—maps the raw per-sample loss ℓ to a weight $w(\ell) \in [0, 1]$, which modulates that sample's contribution to the overall gradient. As in the audio setting, the envelope's 2D shape directly determines how the incoming signal (here, the raw loss values) is modulated.

The envelope shape is not incidental—it *is* the algorithm. It defines which parts of the loss spectrum are preserved or attenuated during training. LAEs are loss-agnostic, self-contained modules: their shape, parameters, and internal state (e.g., EMA for threshold estimation) fully determine their behavior. Once configured, they can be applied with a single call: `loss = envelope(loss)`, encapsulating both logic and control in a compact form.

4 Experiments

4.1 Supervised Classification (MNIST)

We use a simple feedforward classifier trained on MNIST for 20 epochs, using a training batch size of 1,000 and a validation split of 5,000 held-out examples. For each run, we sweep over four envelope types (identity, step, linear, sigmoid), evaluating combinations of step size $\in \{0.01, 0.1, 1.0\}$, threshold quantiles $\in \{10\%, 50\%, 90\%\}$, and (for linear and sigmoid envelopes) transition widths $\in \{1\%, 10\%, 30\%\}$, interpreted as inverse slope or ramp width depending on envelope shape. Each configuration was trained with three random seeds. While this is sufficient to surface performance trends, we note that results may shift slightly with more extensive sampling; future versions of this paper may report 10-seed averages.

We train a shallow MLP on MNIST with and without LAEs. We measure generalization, training dynamics, and sensitivity to noisy labels. LAEs consistently improve validation accuracy and robustness.

Table 1: Best configurations by validation loss (MNIST)

Envelope	Threshold	Width	Val Loss ↓	Val Accuracy ↑
Identity	–	–	0.0686 ± 0.0023	0.9800 ± 0.0017
Step	90%	–	**0.0602 ± 0.0016**	0.9822 ± 0.0012
Sigmoid	90%	0.01	0.0607 ± 0.0023	0.9823 ± 0.0017
Linear	90%	0.01	0.0611 ± 0.0017	0.9825 ± 0.0013

Best-performing configuration (by validation loss) for each envelope type. Step size fixed at 0.1.

Table 2: Best configurations by validation accuracy (MNIST)

Envelope	Threshold	Width	Val Loss ↓	Val Accuracy ↑
Identity	–	–	0.0686 ± 0.0023	0.9800 ± 0.0017
Step	90%	–	0.0602 ± 0.0016	0.9822 ± 0.0012
Sigmoid	50%	0.1	0.0645 ± 0.0008	**0.9828 ± 0.0003**
Linear	90%	0.1	0.0611 ± 0.0017	**0.9828 ± 0.0010**

Best-performing configuration (by validation accuracy) for each envelope type. Step size fixed at 0.1.

4.2 Supervised Classification (CA Housing)

We train a small MLP regressor with a single hidden layer of 64 neurons on the CA Housing dataset using an 80%-20% train-test split and 100 training epochs with 10 seeds. As in the supervised MNIST experiment, we sweep over four envelope types (identity, step, linear, sigmoid), evaluating combinations of step size $\in \{0.01, 0.1, 1.0\}$, threshold quantiles $\in \{10\%, 50\%, 90\%\}$, and (for linear and sigmoid envelopes) transition widths $\in \{1\%, 10\%, 30\%\}$. The best envelope hyperparameters all used a 10% threshold and outperform the baseline identity envelope.

Table 3: Top configurations by validation MSE (California Housing)

Envelope Configuration	Validation MSE ↓
Identity	0.3120 ± 0.0133
Step (T=10%, ss=0.1)	**0.3086 ± 0.0090**
Linear (T=10%, invslope=0.01, ss=0.1)	0.3096 ± 0.0117
Sigmoid (T=10%, invslope=0.1, ss=0.01)	0.3087 ± 0.0114

4.3 Unsupervised Representation Learning (VAE)

We evaluate LAEs in a synthetic overparameterization regime using a shallow VAE trained on a small dataset of 200 MNIST samples, with validation on 2,000 held-out samples. The model uses a latent dimensionality of 8 and is trained for 10 epochs with batch size 50. This setting intentionally introduces overcapacity to test the envelope's ability to regularize training and prevent overfitting. We apply LAEs independently to the prior KL loss and the reconstruction loss, sweeping over step envelopes with thresholds at the 10th, 50th, and 90th percentiles and a fixed step size of 0.1.

As shown in Table X, using LAEs on either the KL prior or reconstruction loss improves validation reconstruction error compared to the baseline configuration (identity on both). The best performance is achieved when both losses are shaped: a step envelope on the prior (90th percentile) and a linear ramp on the reconstruction loss (10th percentile, narrow slope) yield the lowest reconstruction loss. This result supports our hypothesis that gradient attenuation on low-loss elements can provide meaningful regularization, even in unsupervised models trained on very small datasets.

Table 4: Representative configurations (VAE overparam)

Prior Envelope	Recon Envelope	Val Recon ↓
Identity	Identity	151.2 ± 5.8
Step (T=90%)	Identity	136.6 ± 4.5
Identity	Linear (T=10%, w=1%)	136.5 ± 2.3
Step (T=90%)	Linear (T=10%, w=1%)	**125.2 ± 2.2**

Note: T=10% indicates the threshold is set to the 10th percentile of the loss distribution

4.4 Reinforcement Learning (Lunar Lander)

We evaluate LAEs in the reinforcement learning setting using the standard PPO-clip algorithm on the LunarLander-v3 environment. The agent is trained for 500,000 environment steps using 4 PPO epochs per rollout, with a rollout size of 500 and minibatch size of 100. We sweep over a grid of envelope types applied independently to both the policy and value losses: identity (no envelope), step, linear, and sigmoid. For each non-identity envelope, we vary the threshold quantile (10%, 30%) and transition width (10%, 30%, represented via the inverse slope), using a fixed step size of 0.1.

All runs were repeated with 20 random seeds, and results are reported as mean and standard deviation of the moving average of episode return (window size 100) over the final training phase.

Table X summarizes a compact selection of key configurations: baseline (identity), best-performing policy envelope with identity value loss, best-performing value envelope with identity policy loss, and best overall configuration (both policy and value envelope tuned). LAEs significantly improve reward compared to baseline.

Table 5: Top configurations by moving average return (PPO LunarLander-v3)

Policy Envelope	Value Envelope	Return MA100 ↑
Step (T=30%, ss=0.1)	Sigmoid (T=10%, invslope=0.1, ss=0.1)	**22.52 ± 55.16**
Sigmoid (T=30%, invslope=0.1, ss=0.1)	Identity	9.42 ± 49.62
Identity	Linear (T=30%, invslope=0.3, ss=0.1)	-108.99 ± 14.93
Identity	Identity	-115.35 ± 16.52

Note: T=10% indicates the threshold is set to the 10th percentile of the loss distribution.

4.5 Reinforcement Learning (Acrobot)

In a similar test, we evaluate using LAEs in the Acrobot environment using a PPO agent.

5 Analysis

5.1 Loss Attenuation and Gradient Noise

Neural networks are powerful function approximators, but in practice they operate with finite capacity and inductive bias. As established by universal approximation theory (Hornik 1989),

Table 6: Top configurations for PPO on Acrobot at 50k steps. Return is 100-episode moving average.

Policy Envelope	**Value Envelope**	**Return**
Identity	Step (T=0.3)	**-106.04 ± 9.08**
Linear (T=0.1, V=0.1)	Linear (T=0.1, V=0.01)	-121.04 ± 24.33
Sigmoid (T=0.1, V=0.01)	Identity	-132.96 ± 42.21
Identity	Identity	-189.58 ± 100.05

Table 7: Top configurations for PPO on Acrobot at 200k steps. Return is 100-episode moving average.

Policy Envelope	**Value Envelope**	**Return**
Step (T=0.3)	Linear (T=0.1, V=0.01)	**-81.73 ± 1.26**
Identity	Step (T=0.1)	-82.46 ± 2.43
Step (T=0.1)	Identity	-83.70 ± 1.20
Identity	Identity	-85.04 ± 2.48

neural networks can represent any continuous function given sufficient width, depth, and training data. However, under realistic constraints—such as limited layers, parameter count, or data—the network's function class is typically insufficient to capture the true data-generating function.

This mismatch introduces structured residuals. We model the true output y as:

$$y = \hat{y}(x; \theta^*) + \varepsilon(x)$$

Here, $\hat{y}(x; \theta^*)$ is the best approximation that the model class can express (under some objective), and $\varepsilon(x)$ captures the irreducible, input-dependent approximation error due to limited model capacity. Although $\varepsilon(x)$ may be deterministic, it is unpredictable from the model's perspective and thus functions as **structured noise**.

When we compute the per-sample loss $\ell_i = \mathcal{L}(\hat{y}(x_i), y_i)$ and its gradient $\nabla_\theta \ell_i$, the gradient reflects not only signal from true model error, but also the contribution of $\varepsilon(x_i)$. For samples with high loss, the signal typically dominates, and the gradient provides useful guidance. However, for samples where ℓ_i is small—i.e., the prediction is already close to the target—the gradient reflects less of the true model error and becomes increasingly influenced by the residual $\varepsilon(x_i)$.

This effect can be understood in terms of a per-sample signal-to-noise ratio (SNR):

$$\text{SNR}_i = \frac{\|\mathbb{E}_\varepsilon[\nabla_\theta \ell_i]\|^2}{\mathbb{E}_\varepsilon[\|\nabla_\theta \ell_i - \mathbb{E}_\varepsilon[\nabla_\theta \ell_i]\|^2]}$$

When ℓ_i is small, the numerator (signal) diminishes while the denominator (variance due to residuals) remains nonzero, resulting in $\text{SNR}_i \ll 1$. This situation parallels the classic bias–variance decomposition (Hastie 2009), where irreducible error leads to noise-like behavior even in deterministic data.

Loss Attenuation Envelopes (LAEs) directly target this regime by suppressing gradient contributions from low-loss samples. The mapping $\nabla_\theta \ell_i \mapsto w(\ell_i) \cdot \nabla_\theta \ell_i$ with $w(\ell_i) \in [0, 1]$ monotonically increasing in ℓ_i effectively acts as a **gradient SNR filter**, amplifying signal-dominant gradients

and attenuating noise-dominant ones. This improves both training efficiency and generalization, particularly in underparameterized regimes where structured residuals are prominent.

5.2 LAEs as Sample-Wise Regularization

While Loss Attenuation Envelopes (LAEs) can be interpreted through the lens of signal-to-noise ratio (SNR) in the presence of structured residuals, they also admit a compelling alternative explanation: LAEs act as a form of sample-wise regularization that limits overfitting to the training distribution.

In supervised learning, the training dataset is typically a subsample from a larger underlying data distribution. Even in the absence of label noise or aleatoric uncertainty, not all samples in the dataset are equally representative of the broader task. Some examples may be rare, ambiguous, or idiosyncratic, and overly adapting to these samples can lead to memorization and poor generalization.

Traditional regularization techniques like weight decay, dropout, and early stopping operate globally across the model's parameters or training epochs. In contrast, LAEs introduce a localized mechanism: they attenuate gradients based on the per-sample loss, effectively halting optimization on low-loss samples below a learned threshold. This process mirrors early stopping at the sample level.

By preventing further gradient updates on examples that the model already fits well, and which may lie in ambiguous or overfit-prone regions, LAEs limit unnecessary complexity in the learned function. This is aligned with the principle of structural risk minimization, which favors simpler models that fit the data well but avoid excess flexibility. LAEs can thus be understood as introducing a kind of per-example capacity control, dynamically allocating learning effort to the most informative examples.

This perspective helps explain the empirical observation that high-threshold LAEs often improve generalization, even in the absence of structured model inadequacy. Rather than acting only as noise suppressors, LAEs also act as strategic regularizers in data-limited or overparameterized regimes.

5.3 Envelope Behavior

Each of the three envelope shapes—step, linear, and sigmoid—performed best in at least one experimental domain, suggesting that no single shape is universally optimal. However, each has theoretical strengths and weaknesses. The step envelope is easy to tune and aggressively filters low-loss samples, but has an infinite Lipschitz constant which may introduce optimization instability. Linear and sigmoid envelopes allow for smoother transitions via an inverse slope hyperparameter. Sigmoid has the advantage of smooth gradients but never fully suppresses low-loss examples, which could permit overfitting. Linear envelopes are easy to reason about and offer strict zeroing below threshold but are less smooth.

5.4 Practical Tuning and Limitations

A practical challenge of LAEs is the introduction of new hyperparameters: the threshold percentile and the slope or width of the transition region. Fortunately, our experiments show LAEs are relatively insensitive to fine-grained tuning, and broad search over values like threshold and width

was sufficient. More sophisticated search strategies or annealing procedures could further simplify tuning.

In some cases, very high thresholds (e.g., 90th percentile) led to degraded performance, likely due to excessive signal suppression. LAEs also introduce some dependency on batch size, especially for adaptive thresholds or step sizes. However, we observed that small default values like 0.01 or 0.1 performed well across tasks.

5.5 Comparison to Existing Methods

LAEs generalize and reinterpret the free bits technique, extending its principles to arbitrary losses. They share conceptual similarities with hard example mining and robust loss functions, but introduce a novel contribution by using adaptive, EMA-based thresholding. LAEs are also distinguished by their broad applicability across domains and compatibility with any elementwise loss without modifying model architecture.

5.6 Discussion: Broader Implications of Gradient Filtering

Beyond their role in improving signal-to-noise ratio (SNR) during training, Loss Attenuation Envelopes (LAEs) may offer broader benefits that extend to various optimization regimes and learning settings.

Momentum-based optimizers, such as SGD with momentum or Adam, aggregate historical gradients into smoothed updates. In this context, noisy gradients from low-loss samples may not only be ineffective, but actively destabilizing. Once accumulated into the momentum buffer, such gradients can introduce oscillations or delays in convergence. By suppressing these low-magnitude, low-SNR updates, LAEs may act as a form of regularization on the optimizer's internal dynamics, aligning momentum with meaningful gradient directions.

In **multi-task or multi-loss learning** scenarios, gradient interference is a well-known challenge—where updates beneficial to one objective may hinder others. This interference is often exacerbated by inconsistent loss magnitudes or noise in sub-tasks. LAEs could mitigate this issue by acting as a per-sample gate, reducing the influence of outlier or low-value samples that introduce misaligned updates. This points to a potential synergy between LAEs and techniques like gradient surgery or task weighting.

Finally, LAEs with high thresholding—especially in the step envelope case—may offer **computational benefits**. When a large portion of the batch receives zero weight, their gradients do not contribute to the backward pass. With appropriate implementation, this sparsity can be exploited to reduce unnecessary computation, particularly in large-scale or low-signal training regimes.

These connections suggest that LAEs may offer benefits beyond what is captured in current experiments. While speculative, they motivate future work in more complex training settings, such as continual learning, transfer, or meta-optimization, where gradient quality and stability are essential.

6 Conclusion

We introduced **Loss Attenuation Envelopes** (LAEs) as a simple, general-purpose technique for improving training dynamics across supervised, unsupervised, and reinforcement learning tasks.

By attenuating gradients from low-loss samples using batch-derived statistics, LAEs are computationally efficient and complementary to existing regularization and optimization methods. Our results support the view that gradient noise from low-loss examples, due to approximation error or overfitting, can be mitigated with envelope-based attenuation. Future work may explore how to automatically select or adapt envelope thresholds based on training dynamics, reducing the need for hyperparameter tuning. Overall, LAEs provide a flexible and broadly applicable approach to improving generalization and optimization in gradient-based learning.

References

[1] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating Sentences from a Continuous Space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics.

[2] Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder Variational Autoencoders. In *Advances in Neural Information Processing Systems* (NeurIPS).

[3] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improving Variational Autoencoders with Inverse Autoregressive Flow. In *Advances in Neural Information Processing Systems* (NeurIPS).

[4] Loshchilov, I., & Hutter, F. (2015). Online Batch Selection for Faster Training of Neural Networks. In *arXiv preprint arXiv:1511.06343*.

[5] Katharopoulos, A., & Fleuret, F. (2018). Not All Samples Are Created Equal: Deep Learning with Importance Sampling. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

[6] Jiang, L., Zhou, Z., Leung, T., Li, L. J., & Fei-Fei, L. (2019). MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

[7] Ren, M., Zeng, W., Yang, B., & Urtasun, R. (2018). Learning to Reweight Examples for Robust Deep Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*.

[8] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[9] Hornik, K. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5), 359–366.

[10] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.