Contrastive Learning with Latent Tension Regularization for Tight Orbits

Ritwik Ghosal RITWIKGHOSAL1999@GMAIL.COM

Editors: List of editors' names

Abstract

In self-supervised contrastive learning, multiple augmentations of the same input naturally form a set of latent representations, or an *orbit*. Ideally, these representations should remain compact and directionally consistent under transformations. Standard methods such as *SimCLR* prioritize separating different samples but do not explicitly enforce intra-orbit coherence, allowing augmented views of the same input to drift in latent space. We propose **Orbit Regularization Loss (ORL)**, a lightweight extension to the *Normalized Temperature-scaled Cross-Entropy* (NT-Xent) loss that reweights negative pairs based on a *tension score* - a measure of alignment between the positive-pair direction and the candidate negative's displacement. This encourages augmented views to align along stable latent directions, reducing orbit spread without architectural changes or additional supervision. For now, ORL is aimed at improving the geometric structure of embeddings, rather than directly targeting downstream classification accuracy. Experiments on MNIST and CIFAR-10 show that ORL lowers intra-orbit variance, improves directional consistency, and yields a more coherent latent space geometry compared to the NT-Xent baseline.

Keywords: Self-supervised learning, Contrastive learning, Representation learning, Orbit regularization, Latent geometry, Intra-orbit variance, Tension score.

1. Introduction

Self-supervised contrastive learning has emerged as a powerful paradigm for representation learning without labels, primarily by maximizing agreement between different augmentations of the same image. In methods such as the Simple Framework for Contrastive Learning of Visual Representations (SimCLR) (Chen et al., 2020a), augmented views are treated as positive pairs whose latent embeddings are encouraged to align, while embeddings from different images are pushed apart. Collectively, the augmented views of a single image form an *orbit* in the latent space, representing the set of transformation-consistent features for that input.

While contrastive learning effectively separates orbits corresponding to different images, the *internal structure* of each orbit is often overlooked. Strong augmentations can introduce significant variability within an orbit, leading to scattered and unstable representations in the latent space. Such intra-orbit dispersion reduces transformation consistency and can degrade downstream performance.

To address this, we propose **Orbit Regularization Loss (ORL)**, a simple yet effective tension-aware variant of the standard contrastive loss (NT-Xent). ORL introduces the notion of *latent tension*, which measures the geometric alignment of negative samples relative to the transformation direction between positive pairs. By incorporating tension-aware reweighting into the NT-Xent loss, ORL softly regularizes the batch-wise similarity matrix to encourage orbit-tight consistency without compromising inter-class separation. Unlike SimCLR, which treats all negatives equally, ORL gives more importance to negatives aligned with the transformation direction and softly suppresses those in unrelated or opposite directions, leading to more consistent embeddings.

We evaluate ORL on CIFAR-10 (Krizhevsky, 2009) and MNIST (LeCun et al., 2010), comparing it with a standard SimCLR baseline trained using the NT-Xent loss. ORL consistently reduces orbit variance and improves the alignment of augmented views, while maintaining class separation in the latent space. These results are achieved without changing the model architecture or using additional supervision.

Contributions. We introduce *Orbit Regularization Loss (ORL or OR loss)*, a simple extension of NT-Xent that incorporates orbit-aware tension to modulate negative sample influence. ORL promotes consistent, low-variance latent orbits without changing the encoder architecture or requiring additional supervision. We demonstrate its effect through controlled experiments against NT-Xent baselines.

2. Related work

Contrastive learning has become a cornerstone of self-supervised representation learning. SimCLR (Chen et al., 2020a) introduced a simple yet effective framework that learns visual embeddings by maximizing agreement between augmented views using a contrastive loss, requiring large batch sizes to sample diverse negatives. Momentum Contrast (MoCo) (He et al., 2020; Chen et al., 2020b) addressed the batch-size limitation by maintaining a memory bank and a momentum encoder to store stable negative examples. Beyond traditional contrastive losses, methods such as BYOL (Grill et al., 2020) and CPC (Oord et al., 2018) demonstrated that strong representations can be learned even without explicit negatives, highlighting the flexibility of self-supervised learning.

A number of works have refined the contrastive objective to better reflect semantic similarity and mitigate sampling biases. Hardness-aware approaches (Wang et al., 2022; Robinson et al., 2020) reweight negatives based on their difficulty, while Debiased Contrastive Learning (Chuang et al., 2020) reduces the impact of false negatives. Supervised contrastive learning (Khosla et al., 2020) extends the framework by leveraging labels to promote tighter intra-class clustering. Our approach is conceptually related, as it also introduces a reweighting mechanism; however, ORL uses a purely geometric criterion - *latent tension*, to identify orbit-relevant negatives, without requiring labels or global hardness heuristics.

Our work is also connected to orbit-aware and group-equivariant representation learning. Lenc and Vedaldi (Lenc and Vedaldi, 2016) emphasized learning features that respect transformation symmetries, while multi-view or orbit-based generative models (Bouchacourt et al., 2018) exploit shared factors of variation across grouped inputs. Inspired by this perspective, ORL explicitly models the directional relationship between positive pairs and modulates the contrastive loss to preserve orbit geometry in latent space.

Complementary research has explored regularization strategies to improve latent space structure. Barlow Twins (Zbontar et al., 2021) and VICReg (Bardes et al., 2021) enforce decorrelation and variance preservation through auxiliary regularization terms applied *after* contrastive training. In contrast, ORL integrates its regularization *directly into the* *contrastive objective*, influencing the gradients that shape representation geometry. Finally, while orthogonal pretext tasks such as rotation prediction (Gidaris et al., 2018), pretext-invariant learning (Misra and Maaten, 2020), nearest-neighbor contrastive learning (Dwibedi et al., 2021), spatial ensemble smoothing (Huang et al., 2021), and relational reasoning (Patacchiola and Storkey, 2020) offer additional gains, our method focuses on enhancing the *core contrastive loss* itself without auxiliary tasks.

Key difference. ORL reweights negatives based on their alignment with the positive pair's transformation, promoting orbit consistency without labels, extra regularizers, or architectural changes.

3. SimCLR framework

SimCLR is a self-supervised contrastive learning framework that learns visual representations by maximizing agreement between different augmented views of the same image.

Let x be an input image. Two random augmentations $x_1 = t_1(x)$ and $x_2 = t_2(x)$ are generated using stochastic augmentation functions $t_1, t_2 \sim \mathcal{T}$. Each augmented image is passed through an encoder $f(\cdot)$ followed by a projection head $g(\cdot)$ to produce the latent representations:

$$z_1 = g(f(x_1)), \quad z_2 = g(f(x_2)) \tag{1}$$

SimCLR trains the model using the–Normalized Temperature-scaled Cross Entropy(NT-Xent) loss, which for a positive pair (z_1, z_2) is defined as:

$$\mathcal{L}_{\text{NTX}}(z_i, z_j) = -\log \frac{\exp\left(\frac{\text{sim}(z_i,z_j)}{\tau}\right)}{\sum\limits_{k=1}^{2N} \mathbf{1}_{k\neq i} \cdot \exp\left(\frac{\text{sim}(z_i,z_k)}{\tau}\right)} \tag{2}$$

Here, $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ denotes the cosine similarity between vectors u and v, and τ is the temperature hyperparameter controlling the sharpness of the distribution. The batch contains N original images, resulting in a total of $2N$ augmented views. In the loss formulation, the numerator measures the similarity of the positive pair, while the denominator sums over the $2N - 1$ negative pairs corresponding to the chosen anchor.

Intuition. The loss function pulls together representations of positive pairs (i.e., two augmented views of the same image) while pushing apart representations of negative pairs (i.e., views of different images). This encourages the model to form a discriminative latent space where semantically similar inputs cluster together.

4. Proposed framework

4.1. Motivation: why tension?

In SimCLR-style contrastive learning, the objective encourages augmented views of the same image (positive pairs) to align in latent space, while pushing apart views of different images (negative pairs). This is achieved with the NT-Xent loss, which applies uniform repulsion to all negatives, regardless of their geometric relation to the positive pair.

This uniform treatment has a key limitation: it ignores the *structure* induced by augmentations. Multiple transformations of the same image may produce embeddings that scatter in latent space, forming a loose *orbit* with high intra-orbit variance. Such dispersion leads to unstable and inconsistent representations under transformations.

To address this, we introduce the concept of **latent tension** - a measure of how well a negative sample aligns with the transformation direction of a positive pair. Intuitively, tension captures how "on-axis" or "off-axis" a negative is, relative to the motion between two augmented views.

4.2. Capturing consistency across transformations

Formally, latent tension is computed using a *tension score*, which quantifies the directional alignment between the transformation direction defined by a positive pair and a candidate negative. Given a positive pair with embeddings $z_1, z_2 \in \mathbb{R}^d$, we define the normalized *transformation direction*:

$$\delta_{pos} = \frac{z_2 - z_1}{\|z_2 - z_1\|} \tag{3}$$

For an anchor z_i and any other sample z_k, the difference vector is:

$$\delta_{ik} = z_k - z_i \tag{4}$$

The *tension score* is the cosine similarity between the positive direction and this difference vector:

$$T_{ik} = \cos(\theta_{ik}) = \frac{\delta_{pos}^{\top} \delta_{ik}}{\|\delta_{ik}\|} \tag{5}$$

High T_{ik} indicates that z_k lies along the orbit direction and is therefore more "informative" as a negative, while low or negative T_{ik} indicates that z_k lies off-axis.

4.3. Orbit-aware refinement

Instead of treating all negatives equally, ORL scales their influence in the NT-Xent loss according to tension:

$$\mathcal{L}_{\mathrm{ORL}}(z_i, z_j) = -\log \frac{\exp\left(\frac{\mathrm{sim}(z_i, z_j)}{\tau}\right)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \cdot \exp\left(\frac{\mathrm{sim}(z_i, z_k) \cdot T_{ik}}{\tau}\right)} \tag{6}$$

A potential risk in scaling logits by T_{ik} is the **sign-flip pitfall**: if $T_{ik} < 0$, multiplying by it can reverse the sign of $\mathrm{sim}(z_i, z_k)$, accidentally *upweighting* far and anti-aligned negatives or *downweighting* close but anti-aligned negatives. Our implementation avoids this by clamping T_{ik} to a strictly positive range $[10^{-6}, 1]$, ensuring that scaling never changes the sign of the similarity term. This guarantees that the repulsion direction in the loss is preserved.

- **Aligned negatives ($T_{ik} > 0$)** are upweighted, forcing the model to strongly repel negatives that resemble the positive orbit.

- **Off-axis or anti-aligned negatives** (those with $T_{ik} \leq 0$ before clamping) are downweighted toward zero, reducing their influence and preventing them from dominating the repulsion term.

In our experiments we allow gradients to flow through $\tilde{T}_{ik}$, introducing an additional alignment-shaping term $s_{ik}\nabla\tilde{T}_{ik}$, that directly influences the geometric arrangement of negatives in latent space, complementing the standard contrastive repulsion. shown in (Appendix A).

4.4. Overall effect

The proposed Orbit Regularization Loss (ORL) dynamically emphasizes orbit-relevant negatives, promoting compact and transformation-consistent orbits. This reduces intra-orbit variance, improves alignment under augmentation, and enhances the geometric structure of the latent space, all without changing the encoder architecture or adding parameters. A more detailed theoretical analysis is presented in Appendix A and B.

5. Training setup

Our goal is not to compete with state-of-the-art results on downstream tasks, but to isolate and analyze the geometric impact of our proposed regularization on latent representations. To this end, we adopt standard datasets like MNIST and CIFAR-10, and a lightweight ResNet-18 encoder instead of deeper networks like ResNet-50 used in prior SimCLR work. This setup allows us to focus on understanding how the Orbit Regularization loss (ORL) influences intra-orbit structure and representation consistency under transformations, without confounding effects from large-scale architectures or datasets.

In each case of the two datasets, we trained two models: a baseline model using the standard NT-Xent loss, and a second model using the proposed OR loss. All models are trained for 50 epochs using a batch size of 256 and the Adam optimizer(Kingma and Ba, 2014) with a learning rate of 1×10^{-3}. The encoder and projection architectures were kept consistent across all experiments to ensure fair comparison.

It is true that ORL introduces additional computation compared to NT-Xent due to the calculation of T_{ik} for each negative pair. However, in our experiments, ORL consistently reduces the loss more rapidly than NT-Xent, effectively compensating for the extra cost by reaching comparable or lower loss values in fewer training steps.

5.1. Data augmentation for MNIST

Since MNIST consists of grayscale digit images, we used a customized augmentation pipeline designed to preserve digit identity while introducing structural variation. Each image was randomly cropped and resized back to its original size using a gentle range (scale: 0.9–1.0, aspect ratio: 0.9–1.1). Horizontal flips were disabled to avoid altering digit identity (e.g., $6 \leftrightarrow 9$). With 80% probability, we applied a mild affine transform consisting of a rotation in $[-10°, 10°]$ and translation up to 5% of the image size. With 80% probability, brightness and contrast were jittered within ±15%. Finally, Gaussian blur (kernel size 3) was applied with 50% probability to simulate mild noise. This setup introduces sufficient appearance diversity while preserving the semantic structure of each digit.

5.2. Data augmentation for CIFAR-10

For CIFAR-10, which contains natural RGB images across 10 classes, we used a stronger augmentation pipeline suitable for color images. Each image was randomly cropped and resized to 32×32 pixels (scale: 0.8–1.0, aspect ratio: 0.75–1.33), then horizontally flipped with 50% probability. With 80% probability, we applied color jitter to brightness, contrast, saturation, and hue. Grayscale conversion was applied with 20% probability to simulate channel information loss, and Gaussian blur (kernel size 3) with 50% probability to add mild noise. Finally, the image was normalized using the CIFAR-10 channel-wise mean and standard deviation. This setup introduces substantial appearance variation while preserving object-level semantics.

6. Results

We evaluate our Orbit Regularized loss (ORL) method against NT-Xent loss on the CIFAR-10 and MNIST test set, using a ResNet-18 backbone and identical data augmentations and optimization parameters. Both models are trained in a fully unsupervised setting, with no labels used at any point.

Table 1: Orbit structure comparison between NT-Xent and proposed ORL on CIFAR-10 test set (in projector space).

Metric	**NT-Xent**	**ORL**
Orbit Variance	0.000611	**0.000439**
Avg Cosine Similarity	0.8751	**0.9052**
Orbit Diameter	0.6351	**0.5798**
Centroid Deviation	0.2821	**0.2352**
Inter-Class Centroid Dist.	0.8144	**0.8692**

To assess orbit structure in the learned representations, we report five metrics, each corresponding to a specific algorithm:

(i) orbit variance (Algorithm 1), which measures intra-orbit dispersion by computing the mean squared deviation of augmented views from their orbit center,

(ii) average cosine similarity with an anchor view (Algorithm 2), capturing the directional consistency of augmented representations relative to a fixed reference,

(iii) orbit diameter (Algorithm 4), defined as the maximum pairwise distance between any two augmented views within an orbit,

(iv) centroid deviation (Algorithm 4), the average distance of orbit samples from their mean latent embedding, and

(v) inter-class centroid distance (Algorithm 3), the average ℓ_2 distance between class-wise latent centroids obtained from multiple augmented views.

The histogram of cosine similarities between anchors and their augmented views (Figure 1) shows a clear shift to the right for our proposed method compared to standard NT-Xent for images from the CIFAR-10 test set. This means that the augmented images

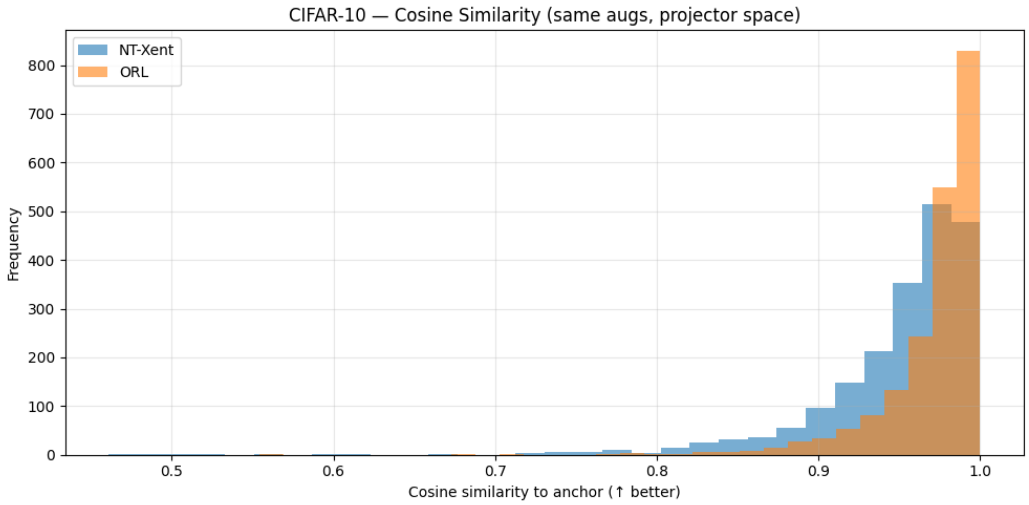
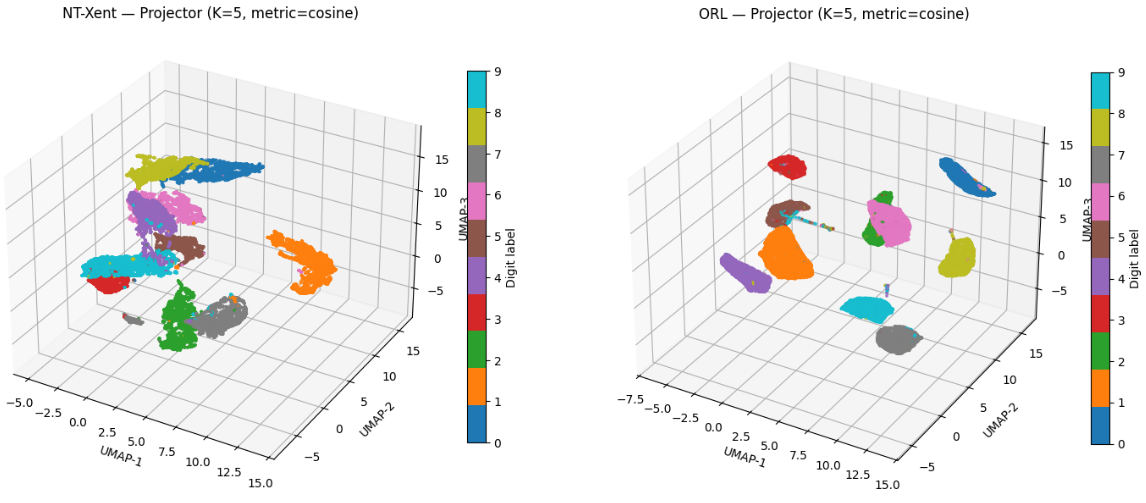


Figure 1: Cosine similarity histogram between anchors and augmented views for NT-Xent and ORL (projector space). Evaluated on 200 anchors (images from the CIFAR-10 test set) with $K = 10$ views each ($200 \times 10 = 2000$ anchor–view pairs per model).



(a) Trained using NT-Xent

(b) Trained using ORL

Figure 2: UMAP of MNIST projector-space embeddings with cosine metric (K=5 views/image)

stay closer to their anchors, reducing the random spread within each orbit. NT-Xent has a wider spread, with more samples below 0.9, while our method produces more similarities close to 1.0. In other words, the orbits become tighter and more organized, without affecting the separation between different classes.

The results in Table 1 demonstrate that the tension-weighted contrastive loss encourages more structured latent representations. ORL reduces intra-orbit variance while simultaneously increasing the separation between classes, validating its geometric regularization effect. Notably, the improvements are achieved without any architectural changes or additional modules. All reported metrics are averaged over multiple runs to ensure robustness against randomness from augmentations and initialization.

On MNIST test images (Figure 2), ORL produces visibly tighter, transformation-consistent orbits in the projector space under identical UMAP settings (K=5 views/image, metric=`cosine`, $n_{\text{neighbors}}$=50, min_dist=0.2, fixed seed). We plot the test images and their augmentations jointly, so the class-level compactness directly reflects the compactness of individual augmentation orbits, confirmed by the explicit orbit metrics for MNIST in the Appendix C. Additional UMAP visualizations and results for encoder space (cosine) and both encoder/projector spaces (Euclidean) appear in Figures 3, 4, and 5 in Appendix C. With a frozen linear probe on encoder outputs, trained under the same augmentation policy (also with horizontal flips), ORL shows **96.13%** test accuracy versus **94.98%** for NT-Xent.

7. Discussion

The ORL formulation introduces a geometric refinement to the NT-Xent loss by incorporating the *tension score* T_{ik}, which modulates each negative sample's contribution based on its alignment with the orbit direction. A natural concern is whether this modification disrupts the core normalization or contrastive behavior of the original NT-Xent loss.

Preservation of softmax structure. In our implementation, T_{ik} is clamped to $\tilde{T}_{ik} \in [\varepsilon, 1]$ with a small $\varepsilon > 0$ to avoid sign flips and ensure numerical stability. Despite this adjustment, the denominator of the ORL loss remains a softmax over the negatives:

$$\exp\left(\frac{\text{sim}(z_i, z_k) \cdot \tilde{T}_{ik}}{\tau}\right),$$

which reduces exactly to the NT-Xent form when $\tilde{T}_{ik} = 1$. The clamping ensures that all terms remain positive scalings of their original logits, preserving the monotonicity and the relative ranking *within* the set of negatives. The softmax normalization is therefore intact, and the loss still enforces that the positive pair must be relatively more similar than the (reweighted) negatives.

Directional refinement without reversal. Since $\tilde{T}_{ik} \geq \varepsilon > 0$, the repulsive direction is never reversed. $\tilde{T}_{ik}$ acts purely as a magnitude gate: values near 1 (high alignment with the orbit direction) amplify repulsion, while values near ε (low or negative alignment) reduce it. This concentrates gradient updates along orbit-tangent directions and de-emphasizes those along orthogonal or unrelated directions. In effect, $\tilde{T}_{ik}$ functions as an *alignment-aware temperature scaler*, sharpening penalties for aligned negatives and flattening them for misaligned ones.

Maintaining contrastive intent. ORL preserves the fundamental contrastive principle: pull positives together, push negatives apart. Its novelty lies in recognizing that not all negatives are equally informative. By modulating their influence with tension, ORL filters noisy gradients that can distort orbit geometry, yielding a principled, direction-aware generalization of NT-Xent. Additional discussion is provided in Appendix A and B.

8. Conclusion

We presented Orbit Regularized Loss (ORL), a simple modification to NT-Xent that encourages more structured intra-orbit geometry in learned representations. Rather than tightly collapsing augmentations, ORL promotes geometric consistency across views. Across the two datasets tested, ORL reduces intra-orbit variance while preserving inter-class separation, indicating improved latent space organization.

While these are early findings, they suggest that orbit-aware weighting could be a useful tool for shaping latent geometry. Further work will explore its behavior on more complex datasets, assess its impact in different architectures, and evaluate potential benefits in semi-supervised learning settings.

References

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9588–9597, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Tengteng Huang, Yifan Sun, Xun Wang, Haotian Yao, and Chi Zhang. Spatial ensemble: a novel model smoothing mechanism for student-teacher framework. *Advances in Neural Information Processing Systems*, 34:15957–15968, 2021.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. Technical Report.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *European conference on computer vision*, pages 100–117. Springer, 2016.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6707–6717, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Massimiliano Patacchiola and Amos J Storkey. Self-supervised relational reasoning for representation learning. *Advances in Neural Information Processing Systems*, 33:4003–4014, 2020.

Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

Kai Wang, Yu Liu, and Quan Z Sheng. Swift and sure: Hardness-aware contrastive learning for low-dimensional knowledge graph embeddings. In *Proceedings of the ACM web conference 2022*, pages 838–849, 2022.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International conference on machine learning*, pages 12310–12320. PMLR, 2021.

Appendix A. Effect of Tension on Gradient Magnitude (with gradient flow through T)

We analyze the gradient of the tension-weighted contrastive loss while allowing gradients to flow through the tension scores. This reveals two distinct gradient channels: one through the similarity terms and one through the tension itself.

A.1. Tension-weighted loss and weights

Given a positive pair (z_i, z_j), define cosine similarities $s_{ik} = \mathrm{sim}(z_i, z_k)$ and the (raw) tension score $T_{ik} \in [-1, 1]$. The ORL objective is

$$\mathcal{L}_{\mathrm{ORL}}(z_i, z_j) = -\log \frac{\exp(s_{ij}/\tau)}{\sum_{k \neq i} \exp\big((s_{ik}/\tau)\,\tilde{T}_{ik}\big)}, \tag{7}$$

where $\tilde{T}_{ik} = \mathrm{clamp}(T_{ik};\, \varepsilon, 1) \in [\varepsilon, 1]$ is the clamped tension (with small $\varepsilon > 0$). The denominator sums over all views except i (including j).

Let the softmax weights be

$$\tilde{q}_k \;=\; \frac{\exp\big((s_{ik}/\tau)\,\tilde{T}_{ik}\big)}{\sum_{\ell \neq i} \exp\big((s_{i\ell}/\tau)\,\tilde{T}_{i\ell}\big)}. \tag{8}$$

Implementation note. Clamping makes $\nabla \tilde{T}_{ik} = 0$ whenever T_{ik} saturates at the bounds. In unconstrained regions ($\varepsilon < T_{ik} < 1$), $\nabla \tilde{T}_{ik} = \nabla T_{ik}$.

A.2. Cosine-similarity gradients

For general (not necessarily unit-norm) embeddings,

$$\nabla_{z_i} s_{ij} = \frac{1}{\|z_i\|\,\|z_j\|}\left(z_j - s_{ij}\frac{z_i}{\|z_i\|^2}\right), \tag{9}$$

$$\nabla_{z_i} s_{ik} = \frac{1}{\|z_i\|\,\|z_k\|}\left(z_k - s_{ik}\frac{z_i}{\|z_i\|^2}\right). \tag{10}$$

Under unit-norm embeddings (common in practice), these simplify to

$$\nabla_{z_i} s_{ij} = z_j - s_{ij} z_i, \tag{11}$$

$$\nabla_{z_i} s_{ik} = z_k - s_{ik} z_i. \tag{12}$$

A.3. Tension definition and its gradient

Let the (raw) tension be the cosine between *normalized* displacement directions:

$$T_{ik} \;=\; \cos\big(u_{ij},\, u_{ik}\big) \;=\; u_{ij}^\top u_{ik}, \quad u_{ij} = \frac{\delta_{ij}}{\|\delta_{ij}\|}, \;\; u_{ik} = \frac{\delta_{ik}}{\|\delta_{ik}\|}, \tag{13}$$

with $\delta_{ij} = z_j - z_i$ and $\delta_{ik} = z_k - z_i$. Then, using standard derivatives of normalized vectors,

$$\frac{\partial u}{\partial \delta} = \frac{I - uu^\top}{\|\delta\|}, \qquad \frac{\partial \delta_{ij}}{\partial z_i} = -I, \quad \frac{\partial \delta_{ik}}{\partial z_i} = -I. \tag{14}$$

By the chain rule:

$$\nabla_{z_i} T_{ik} = \left(\frac{\partial T}{\partial u_{ij}}\right)\left(\frac{\partial u_{ij}}{\partial z_i}\right) + \left(\frac{\partial T}{\partial u_{ik}}\right)\left(\frac{\partial u_{ik}}{\partial z_i}\right) = -\,\frac{u_{ik} - (u_{ij}^{\top} u_{ik})\, u_{ij}}{\|\delta_{ij}\|} - \frac{u_{ij} - (u_{ij}^{\top} u_{ik})\, u_{ik}}{\|\delta_{ik}\|}. \tag{15}$$

Equivalently, writing $T_{ik} = u_{ij}^{\top} u_{ik}$,

$$\nabla_{z_i} T_{ik} = -\,\frac{u_{ik} - T_{ik}\, u_{ij}}{\|\delta_{ij}\|} - \frac{u_{ij} - T_{ik}\, u_{ik}}{\|\delta_{ik}\|}. \tag{16}$$

For the clamped score, $\nabla_{z_i} \tilde{T}_{ik}$ equals this expression wherever $\varepsilon < T_{ik} < 1$, and 0 when saturated.

A.4. Gradient of the loss with gradient flow through tension

Write the loss as

$$\mathcal{L}_{\mathrm{ORL}}(z_i, z_j) \;=\; -\frac{s_{ij}}{\tau} + \log Z_i, \quad Z_i = \sum_{k \neq i} \exp\!\big((s_{ik}/\tau)\, \tilde{T}_{ik}\big). \tag{17}$$

Differentiating $\log Z_i$ yields

$$\nabla_{z_i} \log Z_i = \sum_{k \neq i} \tilde{q}_k\, \nabla_{z_i}\!\left(\frac{s_{ik}}{\tau}\, \tilde{T}_{ik}\right) = \frac{1}{\tau} \sum_{k \neq i} \tilde{q}_k \left(\tilde{T}_{ik}\, \nabla_{z_i} s_{ik} + s_{ik}\, \nabla_{z_i} \tilde{T}_{ik}\right), \tag{18}$$

where

$$\tilde{q}_k \;=\; \frac{\exp\!\big((s_{ik}/\tau)\, \tilde{T}_{ik}\big)}{\sum_{\ell \neq i} \exp\!\big((s_{i\ell}/\tau)\, \tilde{T}_{i\ell}\big)}.$$

Therefore, the full gradient is

$$\nabla_{z_i} \mathcal{L}_{\mathrm{ORL}}(z_i, z_j) = -\frac{1}{\tau}\, \nabla_{z_i} s_{ij} \;+\; \frac{1}{\tau} \sum_{k \neq i} \tilde{q}_k \Big(\tilde{T}_{ik}\, \nabla_{z_i} s_{ik} \;+\; s_{ik}\, \nabla_{z_i} \tilde{T}_{ik}\Big). \tag{19}$$

For comparison, NT-Xent (no tension) has

$$\nabla_{z_i} \mathcal{L}^{\mathrm{NTX}}_{i,j} = -\frac{1}{\tau}\, \nabla_{z_i} s_{ij} \;+\; \frac{1}{\tau} \sum_{k \neq i} q_k\, \nabla_{z_i} s_{ik}, \tag{20}$$

with

$$q_k = \frac{\exp(s_{ik}/\tau)}{\sum_{\ell \neq i} \exp(s_{i\ell}/\tau)}.$$

Interpretation. Clamping $\tilde{T}_{ik} \in [\varepsilon, 1]$ ensures that the repulsive force from ∇s_{ik} always points in the correct direction. In ORL, this force has two parts - a *scaling term*, where $\tilde{T}_{ik}$ changes the strength of ∇s_{ik} so that more aligned negatives are pushed away more strongly and less aligned ones more weakly; and - an *alignment term*, $s_{ik}\, \nabla \tilde{T}_{ik}$, which can slightly adjust the alignment score by changing the relative direction between the positive and negative pairs. When T_{ik} is clamped at a bound, the alignment term becomes zero, leaving only the scaled repulsion. In this way, the method adjusts both the strength and, to a smaller extent, the orientation of updates, while keeping the basic contrastive objective intact.

A.5. Effect of the Alignment-Shaping Term

Meaning of the second term. If $\tilde{T}_{ik}$ were a fixed constant (e.g., precomputed), this term would vanish. But here,

$$\tilde{T}_{ik} = \cos(\theta_{ik})$$

is the cosine between the positive displacement direction $(z_j - z_i)$ and the negative displacement direction $(z_k - z_i)$. This means that moving z_i changes the *alignment* between the orbit direction and the negative direction. The second term

$$\frac{1}{\tau} \sum_{k \neq i} \tilde{q}_k \, s_{ik} \, \nabla \tilde{T}_{ik}$$

captures how the loss changes when these directions are *rotated*, not just when their magnitudes change.

Does it "weaken" negative repulsion? Not inherently, instead, it reallocates gradient energy:

- The first term $(\frac{1}{\tau} \sum_{k \neq i} \tilde{q}_k \, \tilde{T}_{ik} \, \nabla_{z_i} s_{ik})$ in the gradient still pushes/pulls based on similarity s_{ik}, scaled by $\tilde{T}_{ik}$.
- The second term $(\frac{1}{\tau} \sum_{k \neq i} \tilde{q}_k \, s_{ik} \, \nabla_{z_i} \tilde{T}_{ik})$ adds an extra push that tries to maintain the intended tension geometry - it can either increase or decrease the net repulsion depending on whether moving z_i increases or decreases the alignment.

Importantly:

- If a negative is on-axis ($\tilde{T}_{ik} \approx 1$), then small moves in z_i barely change $\tilde{T}_{ik}$, so $\partial \tilde{T}_{ik} / \partial z_i$ is small, making the extra term negligible.
- If a negative is off-axis ($\tilde{T}_{ik} \approx \varepsilon$), then $\partial \tilde{T}_{ik} / \partial z_i$ can be relatively larger - but here s_{ik} is usually small, so the scaling s_{ik}/τ keeps it from dominating.

Why this is a safeguard. This term ensures that negatives do not simply get pushed away in magnitude while staying tangent to the orbit direction. Instead, it actively rotates aligned negatives out of the orbit-tangent subspace when needed, improving geometric separation. Two design aspects prevent it from harming the repulsion objective:

- Clamping: $\tilde{T}_{ik} \in [\varepsilon, 1]$ ensures the scaling term never flips the sign of ∇s_{ik} and that $\nabla \tilde{T}_{ik}$ vanishes at saturation.
- Softmax weighting: Negatives with low similarity s_{ik} receive very small weights $\tilde{q}_k$, minimizing their influence.

Thus, the alignment-shaping channel can only:

1. Push hard, orbit-aligned negatives further away in both distance and direction.
2. Harmlessly reorient easy negatives to maintain a clean separation between orbits.

This extra shaping is absent in the standard NT-Xent loss, where all negatives are repelled purely along their current displacement without regard for orbit geometry.

A.6. Why the alignment-shaping term helps (and pushes toward orthogonality)

Setup. For anchor z_i and positive z_j, the ORL loss is

$$\mathcal{L}_i \;=\; -\frac{s_{ij}}{\tau} \;+\; \log Z_i, \qquad Z_i \;=\; \sum_{k\neq i} \exp\bigl((s_{ik}/\tau)\,\tilde{T}_{ik}\bigr),$$

with similarities $s_{ik} = \mathrm{sim}(z_i, z_k)$ and clamped tension $\tilde{T}_{ik} = \mathrm{clamp}(T_{ik};\, \varepsilon, 1)$, where $T_{ik} = \cos\theta_{ik}$ is the cosine between the positive displacement $u_{ij} = \frac{z_j - z_i}{\|z_j - z_i\|}$ and the anchor–negative direction $u_{ik} = \frac{z_k - z_i}{\|z_k - z_i\|}$. Define the softmax weights

$$\tilde{q}_k \;=\; \frac{\exp\bigl((s_{ik}/\tau)\,\tilde{T}_{ik}\bigr)}{\sum_{\ell\neq i}\exp\bigl((s_{i\ell}/\tau)\,\tilde{T}_{i\ell}\bigr)}.$$

Gradient decomposition (two channels). As, we have seen in eq.(19), differentiating w.r.t. z_i yields

$$\nabla_{z_i}\mathcal{L}_i = -\tfrac{1}{\tau}\nabla s_{ij} \;+\; \underbrace{\tfrac{1}{\tau}\sum_{k\neq i}\tilde{q}_k\,\tilde{T}_{ik}\,\nabla s_{ik}}_{\text{(1) similarity / scaling}} \;+\; \underbrace{\tfrac{1}{\tau}\sum_{k\neq i}\tilde{q}_k\,s_{ik}\,\nabla\tilde{T}_{ik}}_{\text{(2) alignment shaping}}\;.$$

Channel (1) preserves contrastive repulsion but reweights it by $\tilde{T}_{ik} \in [\varepsilon, 1]$. Channel (2) alters *orientation* via $\nabla\tilde{T}_{ik}$; it is zero when $\tilde{T}_{ik}$ is clamped at a bound.

Direct sensitivity and "culprit" targeting. The loss is monotone in the orbit-relevant logit $s_{ik}\tilde{T}_{ik}$:

$$\frac{\partial\mathcal{L}_i}{\partial s_{ik}} = \frac{1}{\tau}\tilde{q}_k\,\tilde{T}_{ik} \;>\; 0, \qquad \frac{\partial\mathcal{L}_i}{\partial\tilde{T}_{ik}} = \frac{1}{\tau}\tilde{q}_k\,s_{ik} \;>\; 0. \tag{21}$$

so negatives with larger $s_{ik}\tilde{T}_{ik}$ receive strictly stronger effective push. The shaping term *does not* remove this property; it only adjusts angles.

Angular effect: $\tilde{T}_{ik} = \cos\theta_{ik}$ with $\theta_{ik} \in [0, \pi]$. Consider the directional derivative of the loss with respect to the angle θ_{ik} (holding s_{ik} fixed):

$$\frac{\partial\mathcal{L}_i}{\partial\theta_{ik}} = \tilde{q}_k\,\frac{\partial}{\partial\theta_{ik}}\Bigl(\tfrac{s_{ik}}{\tau}\tilde{T}_{ik}\Bigr) = \tilde{q}_k\,\tfrac{s_{ik}}{\tau}\,\frac{\partial}{\partial\theta_{ik}}\cos\theta_{ik} = -\,\tfrac{1}{\tau}\,\tilde{q}_k\,s_{ik}\,\sin\theta_{ik}. \tag{22}$$

Hence, for *hard negatives* with $s_{ik} > 0$ and $0 < \theta_{ik} < \pi$, we have $\frac{\partial\mathcal{L}_i}{\partial\theta_{ik}} < 0$ and gradient descent updates

$$\theta_{ik} \;\leftarrow\; \theta_{ik} - \eta\,\frac{\partial\mathcal{L}_i}{\partial\theta_{ik}} \;=\; \theta_{ik} + \eta\,\tfrac{1}{\tau}\,\tilde{q}_k\,s_{ik}\,\sin\theta_{ik} \;>\; \theta_{ik}, \tag{23}$$

i.e., **the angle increases**: the negative is rotated *away* from the positive tangent, toward orthogonality (90°). This is precisely the geometry-cleaning effect of the alignment-shaping channel.

Where it is strongest (mid angles) and why it is safe at extremes. The magnitude satisfies

$$\left|\frac{\partial \mathcal{L}_i}{\partial \theta_{ik}}\right| = \frac{1}{\tau}\, \tilde{q}_k \, |s_{ik}| \, \sin\theta_{ik}. \tag{24}$$

Thus the angular drive is:

- *Largest for mid angles* ($\theta_{ik} \approx 45°$), where $\sin\theta_{ik}$ is large *and* s_{ik}, $\tilde{q}_k$ are still non-negligible. This is the regime where ORL performs the most useful *angular cleanup* while preserving radial repulsion.
- *Small near alignment* ($\theta_{ik} \approx 0°$): $\sin\theta_{ik} \approx 0$, so shaping vanishes; we do not perturb truly on-axis directions.
- *Small near orthogonality* ($\theta_{ik} \approx 90°$): $s_{ik} \approx 0$ and $\tilde{q}_k$ is tiny, so shaping again vanishes; we do not waste updates on irrelevant directions.

Clamp as a built-in brake. With $\tilde{T}_{ik} \in [\varepsilon, 1]$, the shaping channel is active only for $\varepsilon < \tilde{T}_{ik} < 1$. At either bound, $\nabla \tilde{T}_{ik} = 0$ (by clamping), so the angular update *shuts off* automatically. Consequently, for hard negatives with $s_{ik} > 0$, the rotation proceeds *toward* orthogonality until the clamp stops it; it cannot overshoot or reverse the repulsive direction.

Net outcome (why it helps). In the *mid-angle zone*($30° < \theta_{ik} < 60°$) , ORL enforces both radial repulsion and angular cleanup, rotating negatives off the positive tangent. At alignment or orthogonality, shaping naturally vanishes ($\sin\theta \approx 0$ or $s_{ik} \approx 0$), and the clamp halts it at saturation. Thus ORL sharpens geometry only where it matters, self-damps where it could destabilize, and *preserves* the rule of "repel big culprits more."

Appendix B. Orbit variance, subspace geometry, and tension (informal sketch)

Consider the latent orbit $\mathcal{O}_x$ of an image x under a set of augmentations. The displacement vector $\delta_{ij} = z_j - z_i$ between two augmented views can be interpreted as a step along a dominant axis of variation in the orbit.

When most augmentation-induced displacements share a similar orientation, the set of normalized displacement vectors $\{u_{ij}\}$ lies close to a low-dimensional subspace that often aligns with the principal direction(s) of variation in the orbit, as identified by the top eigenvectors of its covariance Σ_x. Positives tend to align strongly within this subspace, while negatives sampled from outside it have low alignment. This difference is quantified by the *tension score*, which measures how "on-axis" or "off-axis" a negative is relative to the displacement between two augmented views. In this way, the observed gap between positive and negative alignment emerges naturally from the geometry of the orbit, without being explicitly enforced.

Under typical, non-adversarial augmentations and assuming a smooth encoder, small changes in orbit geometry, such as slight rotations or minor shifts in aspect ratio, cause proportionally small changes in displacement directions. Since the tension score is based on cosine similarity, which varies smoothly with orientation, these perturbations produce proportionally small shifts in tension. This smooth dependence makes the loss relatively robust to random fluctuations in augmentation geometry across batches.

The variance of the orbit along its main axis reflects the typical squared displacement magnitude along that axis. Larger variance produces longer displacement steps, making it more difficult to maintain high alignment and thus increasing the loss. The tension score mitigates this by down-weighting contributions from directions orthogonal to the main axis of variation, thereby constraining unnecessary spread and encouraging a consistent alignment structure.

From this perspective, tension, geometric selectivity, and variance control are interconnected aspects of the same latent-space geometry. Therefore, the effect of the method on representation learning can be understood as shaping the orbit structure in latent space.

Statement. When the positive-pair directions $\delta_{ij} = z_j - z_i$ align with the top principal subspace of the orbit covariance Σ_x and provide adequate coverage within that subspace, the tension-weighted loss concentrates the contrastive signal along those orbit-relevant directions. Heuristically, this yields a dataset-level control of total intra-orbit variance:

$$\sum_{x \in \mathcal{D}} \mathrm{Tr}(\Sigma_x) \;\lesssim\; \frac{\tau}{\alpha\, c\, (1-\beta)} \sum_{(i,j) \in \mathcal{P}} \mathbb{E}[\mathcal{L}_{\mathrm{ORL}}(z_i, z_j)] \quad + \; \mathcal{O}(\epsilon R^2), \tag{25}$$

where α quantifies empirical coverage within the top-k subspace, $(1-\beta)$ is the tension margin separating in-orbit from out-of-orbit directions, τ is the temperature, $c > 0$ is a universal constant, and ϵ captures small misalignment.

Geometric picture. For an anchor x with orbit $\mathcal{O}_x = \{z_1, \ldots, z_n\}$, let μ_x be its centroid and Σ_x its covariance with eigenpairs $(\lambda_r, \mathbf{v}_r)$. Under the *orbit–geometry alignment* assumption,

$$\mathrm{span}\{\delta_{ij} : (i,j) \in \mathcal{P}_x\} \;\approx\; \mathrm{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}, \tag{26}$$

so positive-pair differences live predominantly in the top-k subspace. The *coverage* condition (parameter α) says these directions are not degenerate within that subspace.

Parallel/orthogonal variance split. Fix a unit direction w in the top-k subspace. Any deviation decomposes as

$$(z - \mu_x)_{\parallel} = (w^{\top}(z - \mu_x))\, w, \qquad (z - \mu_x)_{\perp} = (z - \mu_x) - (z - \mu_x)_{\parallel}. \tag{27}$$

Averaging over the orbit gives the standard identity

$$\mathrm{Tr}(\Sigma_x) = \underbrace{\mathbb{E}\|(z - \mu_x)_{\parallel}\|^2}_{\text{variance along } w} + \underbrace{\mathbb{E}\|(z - \mu_x)_{\perp}\|^2}_{\text{variance orthogonal to } w} \quad . \tag{28}$$

Since w can be chosen from an orthonormal basis of the top-k subspace, *most* variance is captured by a few directions when alignment holds (leakage contributes only $\mathcal{O}(\epsilon R^2)$).

Effect of tension weighting. In ORL the denominator logits for negatives are scaled by the (clamped) tension $\tilde{T}_{ik} \in [\varepsilon, 1]$:

$$\exp\!\Big(\tfrac{s_{ik}\, \tilde{T}_{ik}}{\tau}\Big).$$

Negatives aligned with the positive direction (large $\tilde{T}_{ik}$) are *upweighted*; off-axis or opposite directions (small $\tilde{T}_{ik} \approx \varepsilon$) are *downweighted.* This concentrates repulsion along orbit-relevant directions and reduces incentive to spread orthogonally—intuitively promoting *tighter* orbits in latent space while maintaining inter-class separation. Clamping to $[\varepsilon, 1]$ ensures no sign reversals and preserves the softmax structure.

Heuristic link to variance. Two standard ingredients connect the loss to variance: (i) the pairwise–variance identity, yielding (up to $\mathcal{O}(\epsilon R^2)$)

$$\frac{1}{|\mathcal{P}_x|} \sum_{(i,j)\in\mathcal{P}_x} (w^\top \delta_{ij})^2 \;\approx\; 2\, w^\top \Sigma_x w, \tag{29}$$

and (ii) coverage within the top-k subspace: for unit w therein, $w^\top M_x w \geq \alpha$, where M_x averages $u_{ij}u_{ij}^\top$ with $u_{ij} = \delta_{ij}/\|\delta_{ij}\|$. Projection and averaging then relate $\mathrm{Tr}(\Sigma_x)$ to the mean squared distances of positive pairs in orbit x; summing over orbits yields

$$\sum_x \mathrm{Tr}(\Sigma_x) \;\lesssim\; \frac{1}{\alpha} \sum_{(i,j)\in\mathcal{P}} \mathbb{E}\|\delta_{ij}\|^2 \;+\; \mathcal{O}(\epsilon R^2). \tag{30}$$

Finally, *tension consistency* gives a margin $(1-\beta)$ between in-orbit and out-of-orbit directions, and the softmax temperature τ translates distances into loss, yielding the informal bound stated above with constant $\frac{\tau}{\alpha\, c\,(1-\beta)}$.

Contrast with NT-Xent. NT-Xent weights all negatives equally, so repulsion acts uniformly in all directions and tends to isotropically compress orbits toward class centroids. ORL redistributes gradient magnitude toward orbit-tangent directions (via $\tilde{T}_{ik}$), reducing orthogonal spread and preserving anisotropic structure - consistent with the filament-like manifolds observed in Euclidean UMAP.

Scope. The above discussion is intended to provide an informal yet coherent geometric narrative linking the tension score, orbit–geometry alignment, and variance control. Our aim is to convey a conceptual picture in which dominant latent–space variation directions concentrate the contrastive signal, while tension–weighting suppresses orthogonal spread and preserves anisotropic orbit structure. In this discussion (of Appendix B), the gradient contributions are interpreted through the combined effect of the tension factor and the similarity term, rather than by decomposing them into separate $s\nabla T$ and $T\,\nabla s$ components. This perspective is not meant as a formal theorem, but as a guiding framework for interpreting the empirical patterns we observe: stable loss, clear direction separation, and reduced off-axis variance.

A fully rigorous analysis would require additional machinery, such as spectral perturbation bounds for the orbit covariance and precise control of softmax gradient contributions under tension–weighting, which we leave for future work. Our present goal is to establish the geometric intuition and to foster a shared conceptual model for discussion, supported by the empirical validation provided in the main text as well as what follows.

Appendix C. Additional results

Table 2 reports additional metrics in the *encoder feature space* for MNIST, computed on the same set of anchors for both the vanilla NT-Xent and ORL-trained models. These metrics directly reflect the geometric properties targeted by ORL:

- **Mean class spread** (cosine): average within-class cosine spread (lower is more compact).
- **Silhouette (cosine)**: cluster separability in cosine space (higher is better).
- **Mean orbit diameter / spread**: maximum and mean cosine distances within an orbit (lower is more consistent).
- **Orbit crossing rate**: fraction of points in an orbit misclassified by a 5-NN trained on single-view features (lower is better).
- **Mean positive cosine**: average cosine similarity between an anchor and its K augmented views (higher is more stable).

Table 2: MNIST encoder-space metrics (K=10 views per anchor), evaluated using identical anchors and augmentations for both models.

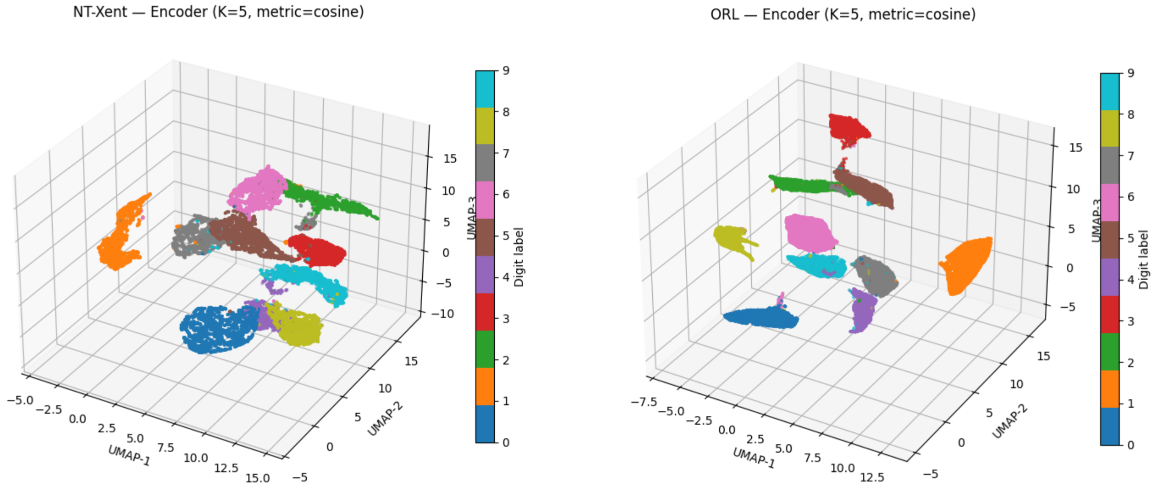
Metric	**NT-Xent**	**ORL**	**Δ** (ORL–NT-Xent)
Mean class spread ↓	0.5177	**0.0896**	-0.4282
Silhouette (cosine) ↑	0.4287	**0.7640**	+0.3352
Mean orbit diameter ↓	0.0327	**0.0108**	-0.0219
Mean orbit spread ↓	0.0083	**0.0029**	-0.0055
Orbit crossing rate ↓	0.0450	**0.0375**	-0.0075
Mean positive cosine ↑	0.9926	**0.9970**	+0.0044

These results corroborate the visual trends in Figure 2: ORL produces orbits that are markedly tighter (lower spread/diameter), more transformation-consistent (higher mean positive cosine), and better separated across classes (higher silhouette), while also reducing orbit crossings.

C.1. Euclidean UMAP: elongated class structure under ORL.

Under the Euclidean metric in projector space, the latent geometry learned by ORL differs qualitatively from NT-Xent not only at the level of individual augmentation orbits, but also at the level of entire class manifolds, as shown for MNIST in Figure 5(b).

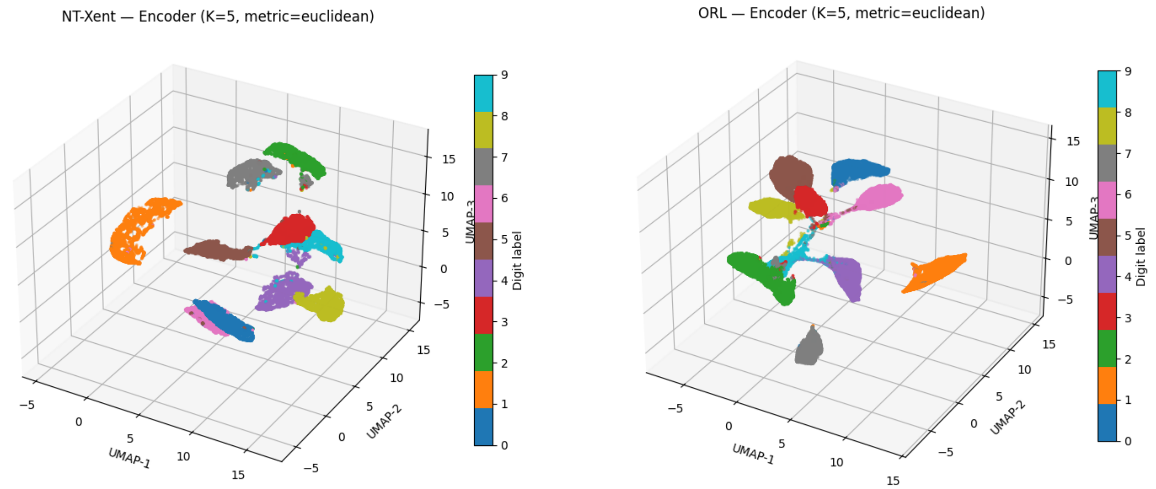
Local (orbit) scale. ORL's tension weighting T_{ik} selectively reduces repulsion from negatives that are poorly aligned with the positive-pair tangent direction $\delta_{ij} = z_j - z_i$. Hence, movement along the orbit tangent is weakly penalized, allowing augmented views of the same anchor to remain spread out along this principal direction rather than being pulled tightly toward the anchor. By contrast, NT-Xent treats all negatives equally, so even tangential displacements receive strong repulsion, collapsing augmentation orbits into compact blobs.



(a) Trained using NT-Xent

(b) Trained using ORL

Figure 3: UMAP of MNIST encoder-space embeddings with cosine metric (K=5 views/image)

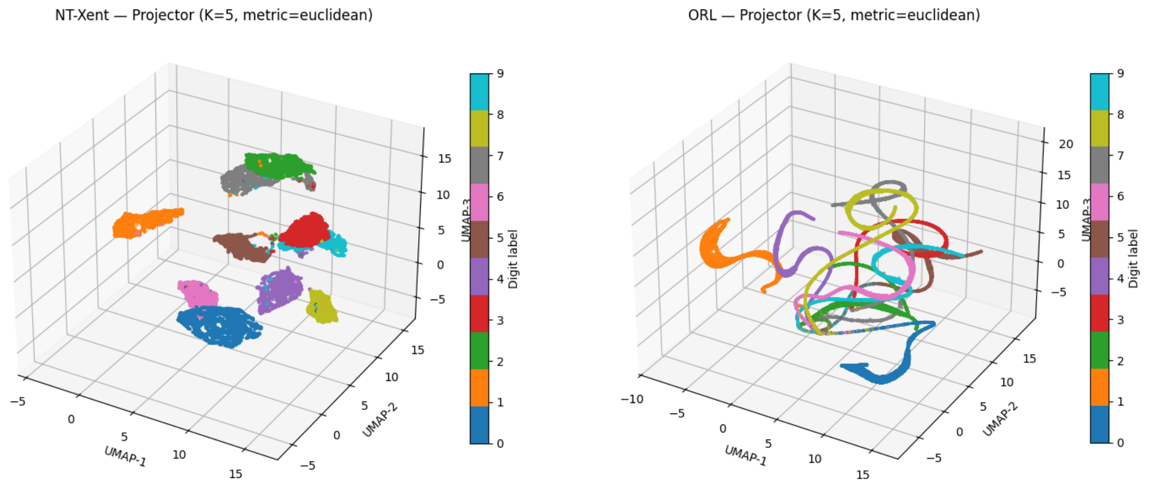


(a) Trained using NT-Xent

(b) Trained using ORL

Figure 4: UMAP of MNIST encoder-space embeddings with Euclidean metric (K=5 views/image)

Global (class) scale. Each class comprises many anchors, each with its own local tangent. Because ORL consistently preserves these local tangents across anchors, the embeddings of all test samples connect into *elongated, thread-like latent structures* at the class level. Thus, the class manifold is allowed to stretch along its main axes of variation (e.g.,



(a) Trained using NT-Xent

(b) Trained using ORL

Figure 5: UMAP of MNIST projector-space embeddings with Euclidean metric (K=5 views/image)

digit slant, stroke thickness, small geometric transforms) instead of being isotropically compressed toward a single centroid. NT-Xent's uniform repulsion removes this anisotropy, producing more isotropic, blob-shaped clusters.

Why MNIST makes it visible. MNIST exhibits low-dimensional, coherent intra-class variation; principal directions such as small translations, rotations, or elastic distortions are consistent across samples. ORL's selective repulsion preserves these directions, and Euclidean UMAP, which preserves straight-line distances, renders them as clear, continuous elongated forms. In contrast, a cosine metric ignores displacement magnitude and considers only angular differences, so points far apart along the same meaningful tangent direction in latent space can still have small cosine distance. This tends to collapse elongated structures into more compact clusters in cosine-based projections. On more complex datasets with noisier intra-class variation, the same mechanism still operates, but these forms may appear less visually coherent in low-dimensional projections.The effect is most pronounced in projector space, where embeddings are directly shaped by the contrastive objective; in encoder space, the same variation exists but is less geometrically exaggerated, so Euclidean UMAP shows weaker elongation.

Takeaway. The elongated-vs.-blob contrast indicates that ORL modifies *entire intra-class geometry*, not just local augmentation orbits, by preserving anisotropic structure in the learned space while maintaining inter-class separation.

Note:

These elongated class structures in Figure 5(b) are observed in projector space with Euclidean metric UMAP. Downstream task performance is determined from encoder representations, which retain the benefits of orbit-aware alignment while being optimized for task-specific objectives.

Appendix D. Pseudo-codes for CIFAR-10 experiments

Algorithm 1: Compute Normalized Orbit Variance for Two Models

```
1. **Input:** Trained models $\mathcal{M}_1$, $\mathcal{M}_2$; image batch $\{x_i\}_{i=1}^{B}$; number of views $k$
2. Initialize lists: var1_list $\leftarrow$ [ ], var2_list $\leftarrow$ [ ]
3. **For** $i = 1$ to $B$
   (a) Generate $k$ augmented views $\{v_1, \dots, v_k\}$ from $x_i$
   (b) **For** $j = 1$ to $k$
       i. Encode $v_j$ with $\mathcal{M}_1$: $z_j^{(1)} \leftarrow \text{Normalize}(\mathcal{M}_1(v_j))$
       ii. Encode $v_j$ with $\mathcal{M}_2$: $z_j^{(2)} \leftarrow \text{Normalize}(\mathcal{M}_2(v_j))$
   (c) Compute orbit mean: $\bar{z}^{(1)} \leftarrow \frac{1}{k}\sum_{j=1}^{k} z_j^{(1)}$, $\bar{z}^{(2)} \leftarrow \frac{1}{k}\sum_{j=1}^{k} z_j^{(2)}$
   (d) Compute orbit variance:
       $$\texttt{var}_1 \leftarrow \frac{1}{k}\sum_{j=1}^{k} \|z_j^{(1)} - \bar{z}^{(1)}\|^2, \quad \texttt{var}_2 \leftarrow \frac{1}{k}\sum_{j=1}^{k} \|z_j^{(2)} - \bar{z}^{(2)}\|^2$$
   (e) Append $\texttt{var}_1$ to var1_list and $\texttt{var}_2$ to var2_list
4. **Return:** var1_list, var2_list
```

The centroid $\bar{z}$ is used solely as a reference point in latent space for computing variance; no claim is made regarding its direct interpretability in the image domain.

Note:

Throughout, we use the terms orbit variance, spread, and pairwise distance somewhat interchangeably to capture how dispersed the orbit is. Strictly speaking, the variance (trace of the covariance) and the average squared pairwise distance are mathematically equivalent, while the orbit diameter (maximum distance) is not identical but strongly correlated. For clarity, we focus on variance as the primary measure, while occasionally using the other terms informally.

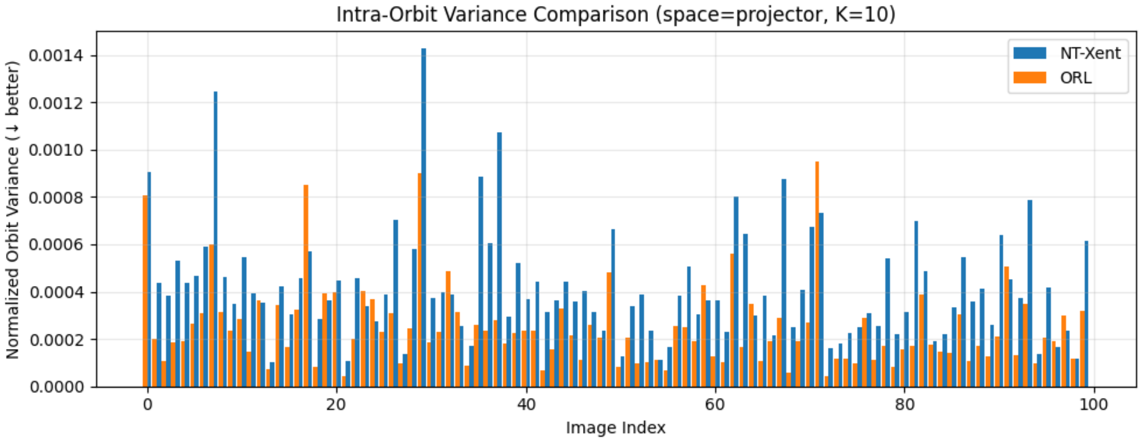


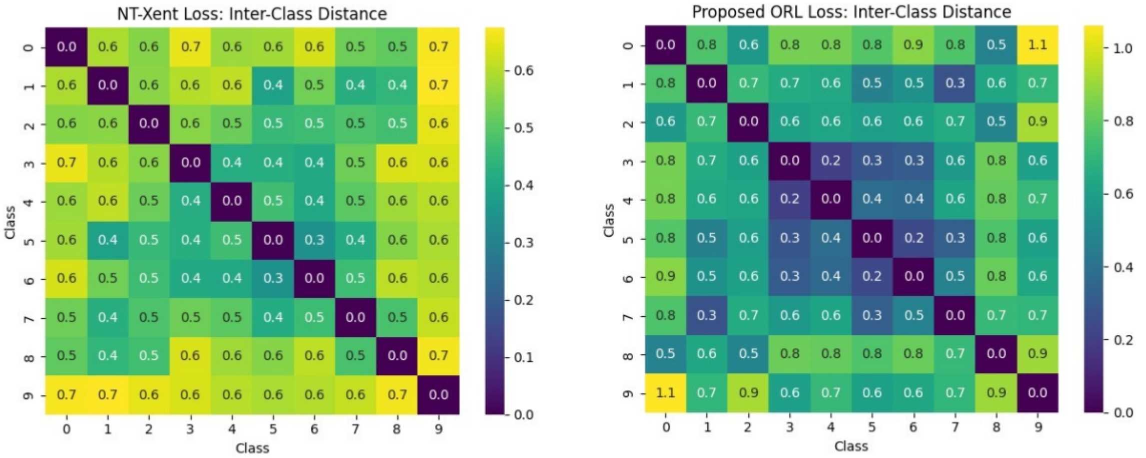
Figure 6: Intra-orbit variance for 100 test images on CIFAR-10 under NT-Xent and our proposed OR loss. Each bar represents the variance across 10 augmented views of the same image. On average, ORL yields tighter orbits with lower variance.

Algorithm 2: Compute Cosine Similarity with Anchor for Two Models

```
1. Input: Trained models $\mathcal{M}_1$, $\mathcal{M}_2$; image batch $\{x_i\}_{i=1}^{B}$; number of views $k$
2. Initialize lists $\texttt{sim1\_list} \leftarrow [\,]$, $\texttt{sim2\_list} \leftarrow [\,]$
3. For $i = 1$ to $B$
    (a) Generate one anchor and $k$ augmented views from $x_i$
    (b) Compute anchor embedding for model $\mathcal{M}_1$: $z^{(1)}_{\text{anchor}} \leftarrow \text{Normalize}(\mathcal{M}_1(\text{anchor}))$
    (c) Compute anchor embedding for model $\mathcal{M}_2$: $z^{(2)}_{\text{anchor}} \leftarrow \text{Normalize}(\mathcal{M}_2(\text{anchor}))$
    (d) For $j = 1$ to $k$
        i. Compute view embedding: $z^{(1)}_j \leftarrow \text{Normalize}(\mathcal{M}_1(v_j))$
        ii. Compute view embedding: $z^{(2)}_j \leftarrow \text{Normalize}(\mathcal{M}_2(v_j))$
        iii. Append cosine similarity to lists:
            • $\texttt{sim1\_list} \leftarrow \texttt{sim1\_list} \cup \{z^{(1)}_{\text{anchor}} \cdot z^{(1)}_j\}$
            • $\texttt{sim2\_list} \leftarrow \texttt{sim2\_list} \cup \{z^{(2)}_{\text{anchor}} \cdot z^{(2)}_j\}$
4. Compute average cosine similarities:
    • $\texttt{avg1} \leftarrow \text{Mean}(\texttt{sim1\_list})$
    • $\texttt{avg2} \leftarrow \text{Mean}(\texttt{sim2\_list})$
5. Return: $\texttt{avg1}, \texttt{avg2}$
```

Algorithm 3: Compute Class Centroids and Inter-Class Distances

```
1. Input: Trained model $\mathcal{M}$; class-wise dictionary $\mathcal{V} = \{c \mapsto \{x_1, x_2, \dots\}\}$; number of views per image $k$
2. Initialize list centroids $\leftarrow$ [ ]
3. For each class $c$ in $\mathcal{V}$
   (a) Initialize all_latents $\leftarrow$ [ ]
   (b) For each image $x$ in $\mathcal{V}[c]$
       i. For $j = 1$ to $k$
          A. Generate augmentation $v_j$ from $x$
          B. Compute latent $z_j \leftarrow \text{Normalize}(\mathcal{M}(v_j))$
          C. Append $z_j$ to all_latents
   (c) Compute centroid $\mu_c \leftarrow \frac{1}{|\texttt{all\_latents}|} \sum z_j$
   (d) Append $\mu_c$ to centroids
4. Stack centroids $C = [\mu_0, \mu_1, \dots, \mu_9]$
5. Compute pairwise $\ell_2$ distances between all pairs in $C$
6. Compute and return mean inter-class distance
```



(a) For NT-Xent loss

(b) For OR loss

Figure 7: Comparison of inter-class centroid distances under NT-Xent and ORL. Pairwise heatmaps (Fig. 7(a), 7(b)) visualize class centroid distances for models trained with NT-Xent and the proposed OR loss, respectively.

Algorithm 4: Compute Orbit Diameter and Centroid Deviation

```
1. Input: Trained models $\mathcal{M}_1$, $\mathcal{M}_2$; image batch $\{x_i\}_{i=1}^{B}$; number of views $k$
2. Initialize lists $\texttt{diam1}, \texttt{dev1}, \texttt{diam2}, \texttt{dev2} \leftarrow [\,]$
3. For $i = 1$ to $B$
    (a) Generate $k$ augmented views $\{v_1, \ldots, v_k\}$ from $x_i$
    (b) Compute normalized embeddings for model $\mathcal{M}_1$: $\{z_j^{(1)}\}_{j=1}^{k}$
    (c) Compute normalized embeddings for model $\mathcal{M}_2$: $\{z_j^{(2)}\}_{j=1}^{k}$
    (d) Compute orbit diameter:
        • $\texttt{diam}_1 \leftarrow \max_{j,l} \|z_j^{(1)} - z_l^{(1)}\|$
        • $\texttt{diam}_2 \leftarrow \max_{j,l} \|z_j^{(2)} - z_l^{(2)}\|$
    (e) Compute centroid deviation:
        • $\texttt{dev}_1 \leftarrow \frac{1}{k} \sum_j \|z_j^{(1)} - \bar{z}^{(1)}\|$
        • $\texttt{dev}_2 \leftarrow \frac{1}{k} \sum_j \|z_j^{(2)} - \bar{z}^{(2)}\|$
    (f) Append all results to their respective lists
4. Compute average values from each list:
    • $\texttt{avg\_diam1}, \texttt{avg\_dev1}, \texttt{avg\_diam2}, \texttt{avg\_dev2}$
5. Return: All average orbit metrics for comparison
```