Audio Engineering Society

Conference Paper

Presented at the AES International Symposium on
AI and the Musician
2024 June 6–8, Boston, MA, USA

Operational Latent Spaces

Scott H. Hawley[1,2] and Austin R. Tackett[1]

[1]*Belmont University*
[2]*Hyperstate Music AI*

Correspondence should be addressed to Scott H. Hawley (`scott.hawley@belmont.edu`)

ABSTRACT

We investigate the construction of latent spaces through self-supervised learning to support semantically meaningful operations. Analogous to operational amplifiers, these "operational latent spaces" (OpLaS) not only demonstrate semantic structure such as clustering but also support common transformational operations with inherent semantic meaning. Some operational latent spaces are found to have arisen “unintentionally" in the progress toward some (other) self-supervised learning objective, in which unintended but still useful properties are discovered among the relationships of points in the space. Other spaces may be constructed “intentionally" by developers stipulating certain kinds of clustering or transformations intended to produce the desired structure. We focus on the intentional creation of operational latent spaces via self-supervised learning, including the introduction of rotation operators via a novel “FiLMR” layer, which can be used to enable ring-like symmetries found in some musical constructions.

1 Introduction

Self-supervised learning has emerged as a powerful tool for uncovering latent representations within data. These latent spaces, often high-dimensional, capture the underlying structure of the data in a way that can be surprisingly meaningful. Notably, some latent spaces exhibit the remarkable property of supporting transformations that correspond to real-world manipulations with semantic interpretations. These transformations can often be expressed as translations or scaling within the space, allowing for intuitive control over the data.

Examples of this can be found in natural language processing, where algebraic manipulations of word vectors can encode complex relationships. For instance, the well-known equation "king" - "man" + "woman" = "queen" from Word2Vec [1] exemplifies how these vectors capture semantic relationships. Similarly, it has been observed that vectors representing countries and their capitals often lie along parallel lines within the latent space, reflecting a clear geometric relationship. These geometric relationships were not necessarily intended but were later explained as having arisen due to the use of matrix factorization in the optimization objective [2]. Matrix factorization was then explicitly used as an objective to encourage semantic geometric structures in subsequent models. Matrix factorization is employed in style transfer systems [3, 4], as factorization is one mechanism for disentangling representations [5].

Disentangled representation learning, where the latent space factors correspond to independent aspects of the data, is another promising approach for achieving controllable music generation [6, 7]. Recent work has also explored leveraging relative positioning within the latent space to control audio effects [8]. In the image domain, StyleGAN and StyleGAN2 were built upon the premise of disentangling controls of image generation. It was later discovered [9] that many other types of possible controls are "latent" within StyleGAN beyond what it was originally intended for, including controls for subjective criteria such as "fluffiness." The potential to unlock new semantic controls within audio, using the latent space of *pre-trained* audio models has received some preliminary attention [10] but the spaces were found to be highly nonlinear, even for linear audio transformations high as high-pass or low-pass filtering. Thus, we may wish to modify the existing latent space of the pretrained model to support the operations we wish to perform, using projective methods such as SimCLR [11] or VICReg [12], which have proved to be powerful tools for self-supervised representation learning.

This paper investigates the potential for self-supervised learning applied to the latent spaces of pretrained audio encoding models to create interpretable latent spaces that empower music producers with fine-grained control over generative models. We present our approach, evaluate its effectiveness, and discuss the implications for fostering creative expression within music production. We consider the effects of enforcing algebraic structures onto the geometry of the latent space, applied through metric learning losses in self-supervised ways. This work bears similarity with some work on "task arithmetic" [13, 14], and the desire to exploit symmetries [**?**] to achieve musically relevant data transformations, yet offers a different set of tasks and mechanisms.

In Section 2, we seek to recover a vector space for music mixing in the latent domain. In Section 3, we go beyond translations and scaling to include rotations among the operations used to provide semantic relationships between data points. We provide supplemental materials and code via a companion website[1].

2 Example 1: Mixing in Latent Space

In typical linear mixing environments such as in the time or spectral (i.e. Fourier) domains, the "mix" is simply a weighted sum of the component musical parts. Neural network systems for audio processing, however, typically incorporate nonlinear transformations which may prevent the sums of neural activations from accurately representing the audio mix. How "nonlinear" are typical neural audio embeddings? In Figure 1, we take various stem components from the MUSDB18 dataset, sum them, and encode them into latent space using VGGish [15] and CLAP [16].

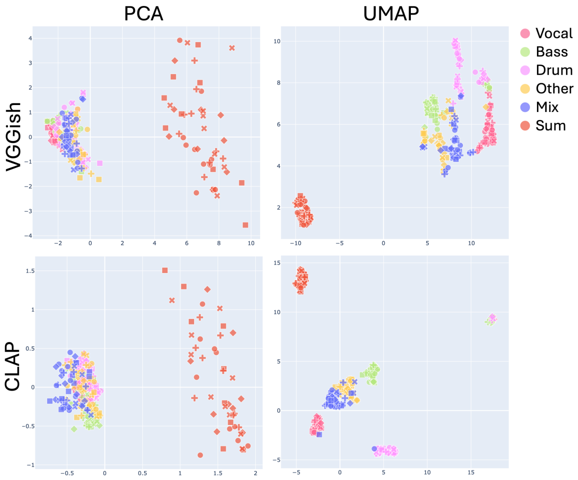

Fig. 1: Encoded stems and mixes from the MUSDB18 [17] audio dataset using the VGGish (top row) and CLAP (bottom row) pretrained encoding models, visualized using PCA (left column) and UMAP (right column). We see that while different stems encode to similar locations, their sums (brown markers) are far from the mix encodings (purple markers), illustrating the nonlinearity of these encoding models.

We consider a "toy model" of points in two dimensions, generating (neural) embeddings via some example nonlinear process, and wish to accomplish the following: *find a "projector" mapping h from the embedding domain into another domain in which the sum of the embeddings lies arbitrarily close to the embedding of the full musical mix.* We could also require that h possess an (approximate) inverse $\widehat{h^{-1}}$ which would allow the projective space to comprise a *"latent plugin"* for the pretrained given model f.

Figure 2 illustrates a schematic for the neural network architecture used, similar to the setup of VICReg [12] yet applied to a new purpose.

This preliminary toy study suggests that semantic audio transformations in latent space may be constructed

[1] Demo & code: https://drscotthawley.github.io/oplas

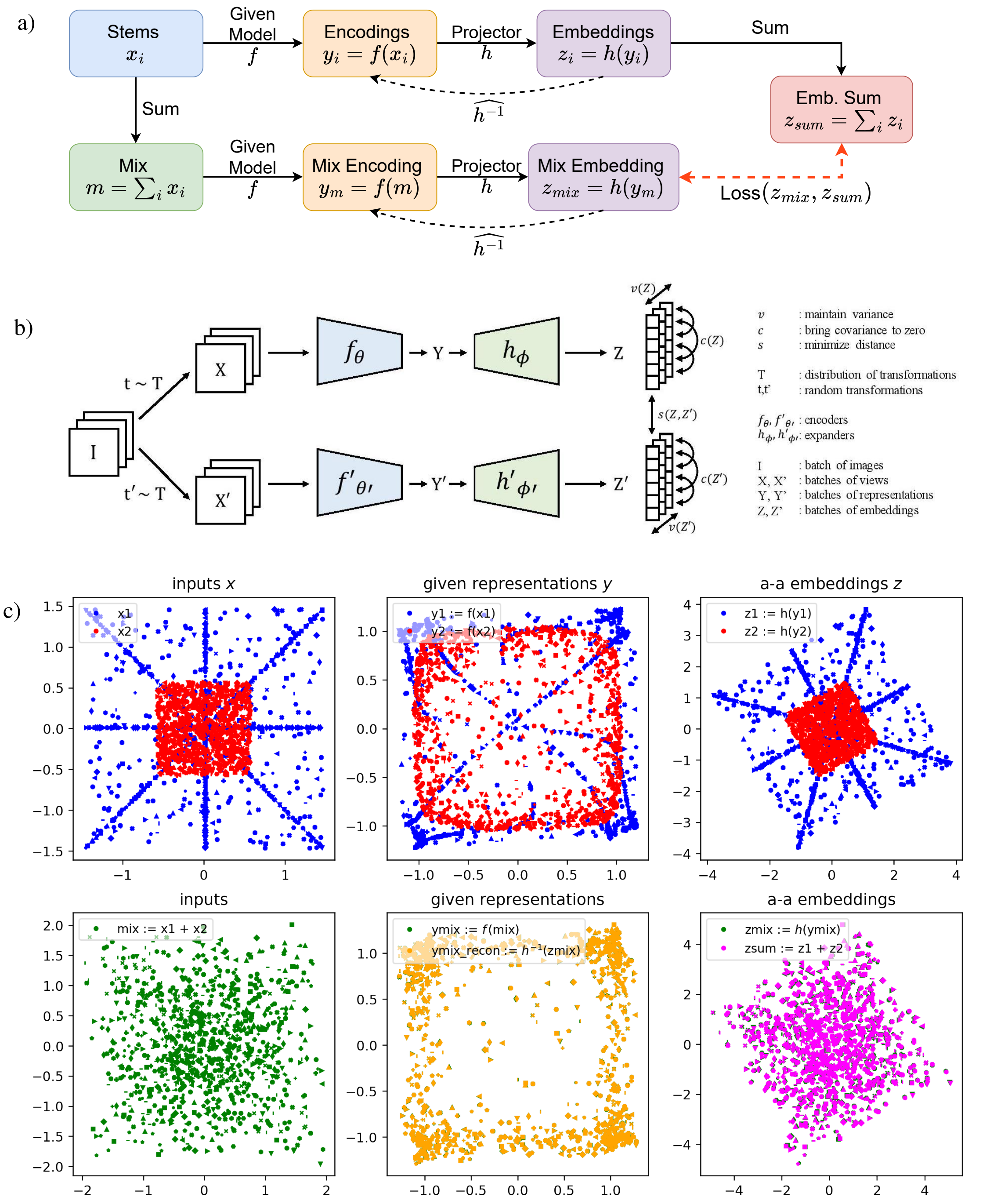

Fig. 2: Mixing with embeddings. a) Flowchart of the algorithm, inspired by a similar flowchart from the VICReg paper [12] shown in b) for comparison. c) Implementation using two classes of 2-D "dots" as proxies for audio stems. The sum of the stems x_i appears in the bottom left in green as the “mix”. In the middle column, we apply some nonlinear twisting and leveling to the “dots” in the left column. In the bottom right, the sums of the embeddings (purple shapes) lie right on top of the embeddings of the mixes (green shapes). Finally, the yellow dots in the bottom middle covering the green dots confirm that we have learned an invertible mapping.

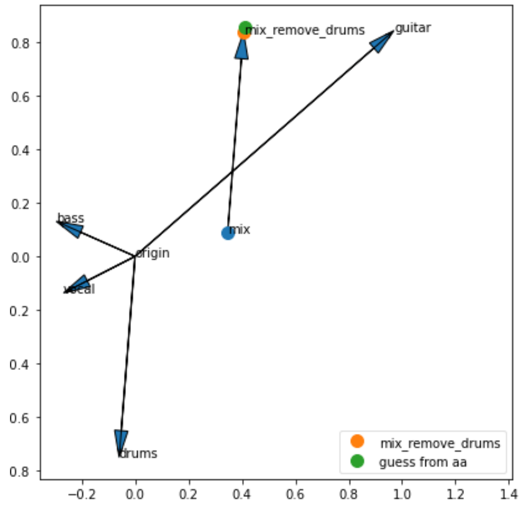

Fig. 3: Mixing in latent space: subtracting the “drums” vector. Here, the signals denoted by “vocal,” “bass,” “drums” and their time-domain sum “mix“ are first embedded in a space Y and then projected into Z. We then compare the projected vectors for the mix without the drums (in the time domain) shown in orange with the “audio algebra” result of subtracting the vector for “drums“ from the “mix“ vector. We see that these are very close to each other in the projected space Z.

explicitly using self-supervised representation learning techniques. Similar studies using real audio encoded via VGGish and/or CLAP models are currently underway but are incomplete at the time of manuscript submission.

3 Example 2: Enabling Rotations

Beyond the FiLM layers [18], which learn abelian transformations, we can try adding rotations, which may lead to additional (and perhaps powerful) semantic operations. We refer to such layers as “FiLMR” layers with the “R” denoting the inclusion of rotation transformations. To illustrate the potential utility for rotations, we pose a sample problem in two dimensions (the “Stargate Problem”, below) to compare a network using square matrices instead of FiLMR layers.

Beyond 2 dimensions, arbitrary rotations in n dimensions incur a “curse of dimensionality” since their symmetry group has a “triangular number“ of $(n^2-n)/2$ degrees of freedom. Restricting our attention to “simple rotations” in a 2-dimensional subspace, we still retain functionality. The algorithm of Aguilera and Aguila [19] provides a way to construct such a rotation operator M iteratively using the plane of 2 n-dimensional vectors $\vec{u}$ and $\vec{v}$, rotating arbitrary vectors $\vec{x}$ by twice the angular separation of $\vec{u}$ and $\vec{v}$. Algorithm 1 shows an outline of a FiLMR layer's operation.

3.1 The "Stargate Problem"

As an example toy problem, we imagine giving the model the task of creating a latent space supporting a simple operation: given a data element (i.e., data point) advance to the next point, with a wrap-around boundary condition such that if the point in question is the last element in the sequence, the operation will map to the first point in the sequence. This is a classic specification of a “ring” symmetry group. Such rings occur in many fields, but especially so in musical contexts such as the basic modulo-12 arithmetic of musical keys, the Circle of Fifths, and the “matrix” of John Coltrane [**?**]. Our sample problem is very simple, but we could extend this by imagining tasks such as: What if we wanted to embed the Coltrane Matrix in a latent space and learn the geometric transformations corresponding to Coltrane's processes?

Formally, this means, given some initial data space Y, the model learns a projection $h$to a new space Z such that for points $z_i \in Z$, the model is also able to learn a transformation Tsuch that $T(z_i) = z_{i+1}$. We refer to this learning task as the “Stargate Problem” because watching the system try to “lock in” while learning the ring structure is somewhat reminiscent of “Stargate” movie and TV shows, in which getting the stargate's chevron-shaped elements to “lock” was a prerequisite for interstellar travel.

We start with points $y_i \in Y$ that lie along a horizontal line shown in blue in Figure **??**. Even though we may “know“ that the correct pair of functions h, $T$to learn are those in which points z_i in the new space Z form a circle, and that Tshould simply be a rotation of $2\pi/N$ (if N is the number of data points) One *could* apply such goals in the form of supervised learning which

Algorithm 1 FiLMR Layer in n dimensions

```
Trainable Parameters: $\gamma, \beta \sim \mathcal{N}$;   $\vec{u}, \vec{v} \sim \mathcal{N}(\mathbf{0}, I_n)$
Forward method:
    Compute rotation matrix $M(\vec{u}, \vec{v}) \in \mathbb{R}^{n \times n}$ via [19].
    Given input $\vec{x} \in \mathbb{R}^n$, transform via $\vec{x} \leftarrow (\gamma\vec{x} + \beta)M$.
```

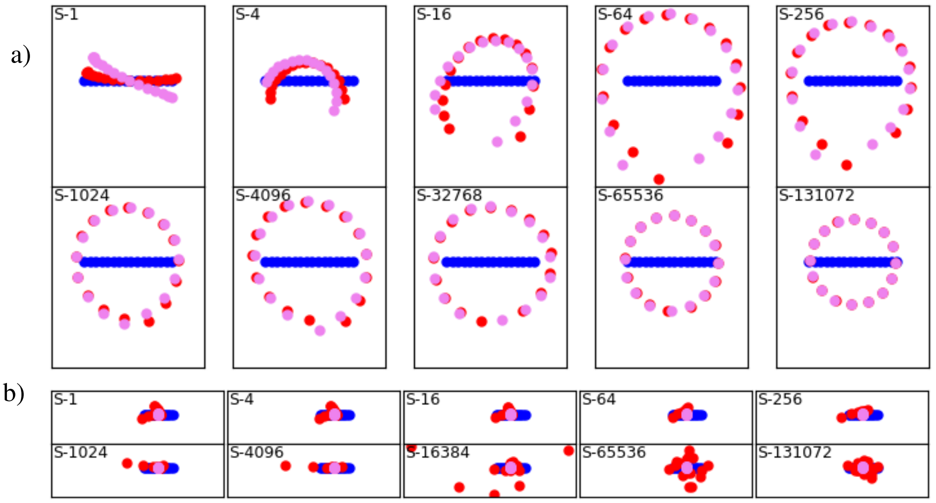

Fig. 4: a) Progress of the Stargate Problem using FiLMR layer. "S-" in the top left of each pane indicates the training step number. b) In contrast, evolution using a learned square orthogonal matrix. While such a solution should exist in theory, the neural network fails to learn the appropriate transformations, perhaps due to dynamic instability. See Figure 5 for a zoomed view of the final simulation states.

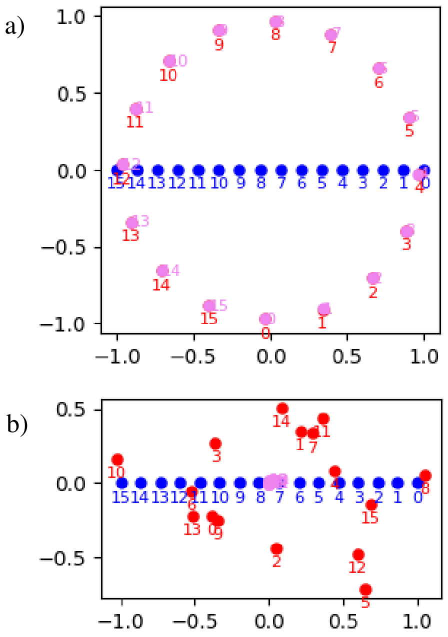

Fig. 5: a): "Final" successful state of model trying the Stargate Problem via a FiLMR layer. The red and pink colors and numbers are intended to show points lining up on top of their "targets," *i.e.,* the next points in the sequence. b): Unsuccessful result of trying to use a learned orthogonal square matrix.

would make this problem nearly trivial. Instead, we simply task the model with minimizing the objective

$$[T(z_i) - z_{i+1}]^2 \qquad (1)$$

where $z_i = h(y_i)$. Note also that the points y_i are imagined as encodings of time-domain audio x_i, encoding via $y_i = f(x_i)$.)

In theory, learning a square matrix for both $h$and Tcould produce the desired projection and transformation properties, respectively. In practice, however, we find trying to learn a full matrix doesn't work, i.e. none of the many attempts we tried ever resulted in the desired structure. Instead, the square-matrix solutions tend to extend the points z_i along a line. Figure 4 illustrates the progress of training, with final states shown in Figure 5.

Extending beyond 2 dimensions to n dimensions, we make use of the Aguilera-Perez Algorithm [19] to construct a n-dimensional rotation matrix M from two learned n-dimensional vectors $\vec{u}, \vec{v}$. The action of M will be to rotate in the plane of $\vec{u}$ and $\vec{v}$ by an angle double that of their separation. Thus, rather than needing to learn a "triangular number" of O(n^2) parameters, the system only needs to learn $2n$ parameters beyond the initial scale and translation of the FiLM layer. The n-dimensional FiLMR layer is outlined in Algorithm 1.

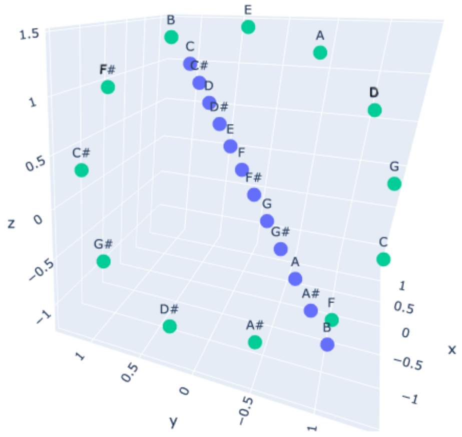

Fig. 6: PCA plot after extending the Stargate Problem to 64 dimensions and converting the sequence of notes to the musical Circle of 5ths. The inputs lie along the diagonal line of 1's (1,1,1,...), and the system learns a rotation operator to bring them into a ring. A separate process learns to rearrange the notes according to the Circle of 5ths.

4 Summary

We have shown two examples of constructing "operational latent spaces (OpLaS)," via self-supervised learning, taking the encodings from larger pretrained models and projecting them to spaces that support a desired (learned) transformation such as summation or rotation. These systems show potential for enabling "latent plugins" for larger pretrained models which by default may not support the desired transformations. The pointwise actions of the loss functions in these systems are reminiscent of inter-particle forces in physics, which typically arise via some symmetry such as energy conservation [20]. This suggests that physical symmetries may yield a fruitful set of transformations for semantic musical operations, as is suggested by recent work by Liu et al [21]. This paper serves as a preliminary feasibility study using "points" in space as proxies for audio stems and their encodings. Future work should include applying the techniques from this study to high-dimensional encodings of real audio.

5 Acknowledgements

The authors wish to thank Brad Schleben and Jonathan Rowden for stimulating discussions. Computing support was provided by equipment grants from Dell and Belmont University. The work for the "Mixing in Latent Space" section was largely completed in 2022 while S.H.H. was employed by Harmonai/Stability AI [22]. The abbreviation "OpLaS" was suggested by Pablo Rivas.

References

[1] Mikolov, T., Le, Q. V., and Sutskever, I., "Exploiting similarities among languages for machine translation," *arXiv preprint arXiv:1309.4168*, 2013.

[2] Pennington, J., Socher, R., and Manning, C., "GloVe: Global Vectors for Word Representation," in *Proc. Emp. Meth. NLP (EMNLP)*, pp. 1532–1543, 2014, doi:10.3115/v1/D14-1162.

[3] Gatys, L. A., Ecker, A. S., and Bethge, M., "A Neural Algorithm of Artistic Style," 2015.

[4] Srivatsan, N., Barron, J., Klein, D., and Berg-Kirkpatrick, T., "A Deep Factorization of Style and Structure in Fonts," in *Proc. Emp. Meth. NLP / Int'l Joint Conf. NLP (EMNLP-IJCNLP)*, pp. 2195–2205, 2019, doi:10.18653/v1/D19-1225.

[5] Kim, H. and Mnih, A., "Disentangling by Factorising," in *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2649–2658, PMLR, 2018.

[6] Yuan, S., Cheng, P., Zhang, R., Hao, W., Gan, Z., and Carin, L., "Improving Zero-Shot Voice Style Transfer via Disentangled Representation Learning," in *ICLR*, 2021, doi:10.48550/arXiv.2103.09420.

[7] Koo, J., Martinez-Ramirez, M. A., Liao, W.-H., Uhlich, S., Lee, K., and Mitsufuji, Y., "Music Mixing Style Transfer: A Contrastive Learning Approach to Disentangle Audio Effects," 2022, doi:10.48550/ARXIV.2211.02247.

[8] Moschella, L., Maiorca, V., Fumero, M., Norelli, A., Locatello, F., and Rodolà, E., "Relative representations enable zero-shot latent space communication," in *ICLR*, 2023.

[9] Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S., "GANSpace: Discovering Interpretable GAN Controls," *CoRR*, abs/2004.02546, 2020.

[10] Hawley, S. H. and Steinmetz, C. J., “Leveraging Neural Representations for Audio Manipulation,” in *154th AES Convention*, Audio Engineering Society, 2023.

[11] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G., “A Simple Framework for Contrastive Learning of Visual Representations,” in *ICML*, volume 119, pp. 1597–1607, PMLR, 2020.

[12] Bardes, A., Ponce, J., and LeCun, Y., “VICReg: Variance-Invariance-Covariance Regularization For Self-Supervised Learning,” in *ICLR*, 2022.

[13] Ilharco, G., Ribeiro, M. T., Wortsman, M., Schmidt, L., Hajishirzi, H., and Farhadi, A., “Editing models with task arithmetic,” in *ICLR*, 2023.

[14] Ortiz-Jimenez, G., Favero, A., and Frossard, P., “Task Arithmetic in the Tangent Space: Improved Editing of Pre-Trained Models,” in *NeurIPS*, 2023.

[15] Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., and Wilson, K., “CNN Architectures for Large-Scale Audio Classification,” in *ICASSP*, 2017.

[16] Wu, Y., Chen, K., Zhang, T., Hui, Y., Berg-Kirkpatrick, T., and Dubnov, S., “Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation,” in *ICASSP*, 2023.

[17] Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R., “The MUSDB18 corpus for music separation,” 2017, doi:10.5281/zenodo.1117372.

[18] Perez, E., Strub, F., de Vries, H., Dumoulin, V., and Courville, A., “FiLM: visual reasoning with a general conditioning layer,” in *Proc. of the 32nd AAAI Conf. on Artificial Intelligence*, New Orleans, Louisiana, USA, 2018, ISBN 978-1-57735-800-8.

[19] Aguilera, A. and Pérez-Aguila, R., “General n-Dimensional Rotations,” in *International Conference in Central Europe on Computer Graphics and Visualization*, 2004.

[20] Dawid, A. and LeCun, Y., “Introduction to Latent Variable Energy-Based Models: A Path Towards Autonomous Machine Intelligence,” 2023.

[21] Liu, X., Chin, D., Huang, Y., and Xia, G., “Learning Interpretable Low-dimensional Representation via Physical Symmetry,” in *NeurIPS*, 2023.

[22] Hawley, S. H., Evans, Z., and Baldridge, J., “Audio (vector) algebra: Vector space operations on neural audio embeddings,” *JASA*, 152(4-Supplement), pp. A178–A178, 2022, doi:10.1121/10.0015957.