Disentangled Latent Representations for Audio Style Transfer

July 8, 2025

Francesco Brigante Tommaso Federici Lucia Fornetti

Abstract

This paper introduces a framework for Audio Style Transfer by learning disentangled representations (Hung et al., 2019) of style and content directly from complex-valued spectral data. Traditional methods often operate on waveforms or magnitude spectrograms, discarding phase information crucial for high-fidelity audio synthesis (Shen et al., 2024).

Our approach involves a dual-encoder architecture to separate instrument timbre (style) from musical structure (content). A dedicated style encoder and content encoder process a combined Short-Time Fourier Transform (STFT) and Constant-Q Transform (CQT) representation (Brown, 1991). The disentanglement is enforced through a combination of strategies: the usage of a discriminator network, which engages the encoders in a reciprocal optimization challenge, characteristic of adversarial training (Goodfellow et al., 2014), complemented by contrastive losses such as InfoNCE (van den Oord et al., 2018) and a Margin-Based loss (Wu et al., 2017), to structure the style space, and an explicit penalty to minimize any remaining correlation between style and content representation, the Hilbert-Schmidt Independence Criterion (HSIC)(Gretton et al., 2005).

The ultimate goal is to enable high-quality audio generation where the style of one instrument can be applied to the content of another, by feeding these disentangled embeddings into an Autoregressive Decoder.

Email: Francesco Brigante <brigante.1987197@uniroma1.it>, Tommaso Federici <federici.2214368@uniroma1.it>, Lucia Fornetti <fornetti.2214370@uniroma1.it>.

Deep Learning and Applied AI 2025, Sapienza University of Rome, 2nd semester a.y. 2024/2025.

1. Introduction

Generative modeling of audio has seen significant advances, yet achieving *controllable synthesis*, such as in audio style transfer, remains challenging.

Most existing methods work either in the waveform domain (Huang et al., 2019b) or on magnitude spectrograms, reconstructing phase via Griffin–Lim (Griffin & Lim, 1984), which often introduces artifacts. To overcome this, we operate directly on complex spectrograms, jointly modeling real and imaginary parts.

Contributions

- Novel audio style transfer framework that operates in the complex spectrogram domain. Unlike typical approaches, our framework features an autoregressive decoder, along with a style encoder, a content encoder, and a discriminator. This autoregressive component processes the complex spectrograms sequentially, enhancing the coherence and naturalness of the generated audio
- Explicit disentanglement mechanism, inspired from cutting-edge vision-based style transfer techniques (Li et al., 2024). This is achieved through a combination of adversarial, contrastive, and decorrelation losses, meticulously designed to enforce the independence of style and content representations.

Code. The project's source code is available at: `https://github.com/francescobrigante/Audio-Style-Transfer/tree/main`.

2. Related Work

A common strategy involves using Variational Autoencoders (VAEs) with adversarial objectives. (Hung et al., 2019) employs this technique to separate instrumentation from musical score information, enforcing independence through adversarial training.

(Cífka et al., 2021) introduces a VQ-VAE for one-shot music style transfer: by quantizing the latent space and using self-supervised objectives, their model learns disentangled

representations of timbre and pitch without explicit labels, enabling transfer from a single target audio sample.

More recently, diffusion models have emerged as a powerful paradigm for generative tasks. (Ho et al., 2024) pioneers the use of diffusion for music style transfer, showing promising results in generating high-fidelity audio by iteratively refining a noisy signal conditioned on content and style embeddings. This mirrors trends in image style transfer (Li et al., 2024), where transformer and diffusion-based architectures are the current SOTA.

3. Method

3.1. Dataset Construction

We assembled balanced 10-second, 44kHz clips for two instruments—violin and piano—totaling 1536 examples (768 per instrument).

Violin. We sourced recordings from the Bach Violin Dataset(Dong et al., 2021) and ViolinEtudes(Anonymous, 2022), extracting 1–3 segments per file to reach 768 clips. All audios were resampled to 44kHz and gain-normalized to an RMS of 0.07. Average MFCCs (–351.2 vs. –334.9) were already compatible, so no further scaling was needed. From this analysis, we obtained the following visualization (blue points for "Bach Violin Dataset", red points for "ViolinEtudes").

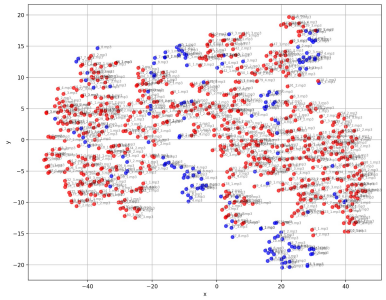

Figure 1. Sonic Diversity Distribution (t-SNE on MFCC Features)

Piano. From PianoMotion10M (Gan et al., 2025), we took one 10-second excerpt per piece to match the violin count (768 clips).

Finally, we split the combined 1536-clip set into 80% train, 10% validation, and 10% test, each containing separate folders for violin and piano.

Our model input concatenates STFT and CQT representations, leveraging STFT's temporal precision and CQT's logarithmic frequency scaling for musical content. To normalize the data, we computed mean and standard deviation for STFT and CQT features from the training set for each channel (real and imaginary) and for each frequency bin.

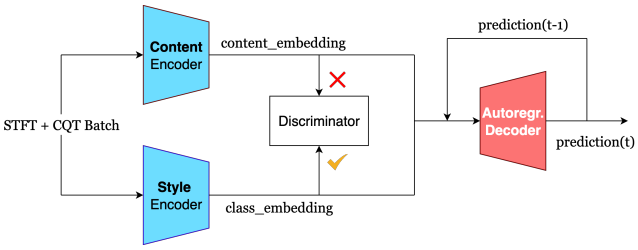

Figure 2. Model general architecture

3.2. Encoders Architecture

Both the Style and Content Encoders share a similar architectural blueprint: a deep CNN (Lee et al., 2017) followed by a Transformer Encoder(Vaswani et al., 2017). This design allows the CNN to learn hierarchical local features from the spectrogram patches, while the Transformer captures long-range temporal dependencies across these features.

Style Encoder. The Style Encoder is designed to represent the timbral identity of an instrument from a sequence of spectrogram chunks $(x_1, ..., x_S)$ into a compact representation. It produces two key outputs: an instance-level style embedding z_{style} for each input audio, and a set of class embeddings z_{class} representing the style of each instrument class in the dataset.

The architecture begins with a deep CNN composed of ResBlocks, that captures local, hierarchical patterns within the frequency and time dimensions. To the resulting sequence, we prepend a special, learnable `[CLS]` token to this sequence, which serves as a global aggregator, inspired by the success of models like BERT(Devlin et al., 2018) and the Vision Transformer (ViT)(Dosovitskiy et al., 2020a). The entire sequence is then fed into a Transformer Encoder, where the `[CLS]` token, since it has no intrinsic input information, accumulates and summarizes the most salient stylistic information through multiple layers of self-attention.

The Transformer's final output vector at the `[CLS]` token position is then taken as the instance-level style embedding, z_{style}. To obtain the general class prototypes, z_{class}, we average all the z_{style} belonging to the same instrument class within a batch. This yields a single, robust vector representing the unique timbre for each class.

Content Encoder. The Content architecture mirrors the Style Encoder but it does not use a `[CLS]` token, but instead it produces a sequence of content embeddings $z_{content} = (c_1, ..., c_S)$, preserving the temporal structure

of the input.

3.3. Adversarial Disentanglement

To enforce the separation of style and content, we employ a discriminator network, implemented as a simple Multi-Layer Perceptron (MLP). The discriminator is trained to predict the instrument class from the embeddings produced by the encoders, minimizing the cross-entropy loss when predicting the instrument class from z_{style} and z_{class}, and from $z_{content}$.

The encoders are trained to fool the discriminator on the content embedding. This is achieved by maximizing the entropy of the discriminator's predictions for $z_{content}$, thereby encouraging the content embedding to be uninformative about the instrument class. At the same time, the Style Encoder is trained to produce style embeddings that are highly discriminative for the instrument class, helping the discriminator.

3.4. Autoregressive Decoder

The Decoder autoregressively reconstructs the audio signal in the form of a STFT spectrogram, conditioned on a content embedding $z_{content}$ and a class embedding z_{class}. The architecture consists of the following main components:

1. **Linear Projection Encoder:** During training, the ground truth STFT spectrogram is flattened per frame and projected into the model's hidden dimension via a linear layer. A learnable start-of-sequence token is prepended, followed by sinusoidal positional encodings and a causal attention mask to ensure autoregressive behavior.
2. **Transformer Decoder:** The core component is a multi-layer Transformer decoder that models temporal dependencies across frames through masked self-attention. Cross-attention integrates content information from $z_{content}$ and timbre conditioning from z_{class}. These embeddings are projected and concatenated before being used as memory for the decoder.
3. **Spectrogram Reconstruction:** The output embeddings of the Transformer are projected back into the original STFT space using a linear layer, reshaped into the original $(\text{real}, \text{imaginary})$ format across time and frequency.

During inference, the decoder autoregressively generates each frame by iteratively appending the most recent output to the input sequence. For style transfer, the class prototype z_{class} of the target instrument is used to condition the output, ensuring the generated audio reflects the desired timbre while preserving the input's musical content.

3.5. Loss Functions

The overall training objective is a weighted sum of several specialized loss functions designed to structure the embedding space and enforce disentanglement.

Contrastive Losses. To ensure that the style embeddings are discriminative and well-separated, we use two contrastive losses:

- **InfoNCE Loss** (van den Oord et al., 2018): Applied to the instance-level style embeddings z_{style}, this loss pulls embeddings from the same instrument class together in the latent space while pushing apart embeddings from different classes.
- **Margin Loss** (Wu et al., 2017): Applied to the class prototype embeddings z_{class}, this loss enforces a minimum distance between the representations of different instrument classes.

Adversarial Loss. As described previously, this loss governs the minimax game between the encoders and the discriminator. The total loss is a sum of the discriminator's classification loss and the generator's adversarial loss on the content embedding.

Disentanglement Loss. We also add an explicit penalty to minimize the statistical dependence between style and content embeddings. We use the Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al., 2005), a powerful kernel-based measure of independence that can capture both linear and non-linear correlations. This loss directly penalizes any remaining correlation between the two representations.

Reconstruction Loss. To optimize the Decoder during training, a composite reconstruction loss is employed to compare the predicted STFT spectrogram with the target, balancing multiple objectives for audio reconstruction and style transfer.

The primary component is the Mean Squared Error (MSE) loss, which measures the overall difference between the predicted and target spectrograms. To enhance spectral quality, a magnitude loss (Takaki & Yamagishi, 2018) computes the MSE between the spectral amplitudes of the predicted and target spectrograms. A phase loss (Takaki & Yamagishi, 2018), incorporating phase wrapping via `atan2`.

To promote temporal coherence, a temporal consistency loss (Park et al., 2019) penalizes differences in consecutive frame transitions between the predicted and target spectrograms. Similarly, a spectral consistency loss (Takaki & Yamagishi, 2018) enforces harmonic coherence by comparing spectral gradients along the frequency dimension. Finally,

a magnitude-phase consistency loss (Huang et al., 2019a) ensures that predicted magnitude and phase are coherent with the complex spectrogram representation, reducing artifacts.

This weighted combination of losses enables the Decoder to generate spectrograms that preserve musical content and target timbre, optimizing audio quality for both reconstruction and style transfer tasks.

4. Experimental Results

The model was trained for 100 epochs and the evaluation focused on two primary tasks: content reconstruction and style transfer:

Content Reconstruction Metrics

- Chroma Distance: Euclidean distance between original/reconstructed chroma features.
- Onset F1: F1 score of onset detection to gauge rhythmic alignment.
- Pitch Correlation: Pearson's r between pitch contours.
- MSE Spectrogram: MSE between original and reconstructed STFTs.

Style Transfer Metrics

- Chroma Similarity: Assesses harmonic consistency between the generated and original audio by computing the correlation between chroma vectors. Lower values indicate stronger stylistic transformation.
- MFCC Distance: Measures the timbral difference between the generated audio and a reference sample of the target instrument.
- Instrumentation Similarity: Compares spectral energy distributions to determine similarity in instrumental characteristics.
- Self-Similarity Matrix Distance: Computes the distance between recurrence matrices of the generated and reference audios to assess structural alignment.

4.1. Experimental Results

All test audios (piano and violin) were resampled to 22050Hz mono.

- **Piano Reconstruction**
  - Chroma Distance: Mean = [value], Std = [value]
  - Onset Accuracy: Mean = [value], Std = [value]
  - Pitch Correlation: Mean = [value], Std = [value]
  - MSE Spectrogram: Mean = [value], Std = [value]
- **Violin Reconstruction**
  - Chroma Distance: Mean = [value], Std = [value]
  - Onset Accuracy: Mean = [value], Std = [value]
  - Pitch Correlation: Mean = [value], Std = [value]
  - MSE Spectrogram: Mean = [value], Std = [value]
- **Piano-to-Violin Style Transfer**
  - Chroma Similarity: Mean = [value], Std = [value]
  - MFCC Distance: Mean = [value], Std = [value]
  - Instrumentation Similarity: Mean = [value], Std = [value]
  - Self-Similarity Distance: Mean = [value], Std = [value]
- **Violin-to-Piano Style Transfer**
  - Chroma Similarity: Mean = [value], Std = [value]
  - MFCC Distance: Mean = [value], Std = [value]
  - Instrumentation Similarity: Mean = [value], Std = [value]
  - Self-Similarity Distance: Mean = [value], Std = [value]

5. Discussion and Conclusions

TO DO (LOOK AT GITHUB REPOSITORY TO FIND NEWEST VERSION OF REPORT)

Team separation of tasks:

- *Lucia Fornetti, 2214370:* Dataset and dataloader creation, data pre-processing, reconstruction evaluation and style transfer evaluation.
- *Francesco Brigante, 1987197:* Implementation of style encoder, content encoder, discriminator, GAN-style training loop, definition of losses and actual training on local machine.
- *Tommaso Federici, 2214368:* Implementation of Autoregressive Decoder, definition of the multi-component reconstruction loss and style transfer inference pipeline.

Appendix

Motivating the idea The initial idea was to design a system that was able to reconstruct the original audio, starting from the style and content embedding, thus proving that

these latent representations are meaningful and representative for the style transfer task.

However, the architecture itself is not enough to impone disentanglement between the two embeddings: even with perfect reconstruction there's no guarantee that the style representation is really encoding musical timbre and the content representation musical structure.

To enforce this, a series of techniques and constraints was applied: first the idea of batching audio files including balanced class representatives (e.g. for 2 instruments B/2 elements refer to instrument 1 and B/2 to instrument 2), then applying contrastive and adversarial methods utilizing the available data: since we have different songs played by the same instrument, it would be meaningful to add distances constraints to enforce content-invariance in the style encoder; also we don't have audios with same musical structure but played by different instruments, so for the style-invariance we had to use other constraints, such as the discriminator network.

For the Discriminator, I wanted to choose a simple model (that could be easily part of an Ablation study) since there were already large architectures. When training the whole model optimizing for Generators (I refer to everything that is not the discriminator: 2 encoders and decoder) the chosen optimization strategy was to minimize negative entropy given $z_{content}$, (we've also experimented with Cross-entropy), thus maximizing entropy and forcing the discriminator's predictions to be random.

The choice to use one single Discriminator, instead of 2 (one for style and one for content), is motivated by the fact that a single discriminator that evaluates both $z_{content}$ and z_{style}, learns both latent spaces, so when trained from the generator point of view, it's "forced to hide all tymbral clues" of $z_{content}$. And the discriminator "has already seen those clues" during training from its point of view, when it's forced to learn style from z_{style}. On the other hand, if using 2 discriminators, each network could quickly specialize on the easiest statistical cue available in its respective domain.

Dataset selection and preprocessing The choice to utilize piano and violin stems from the availability of well-established datasets and the significant timbral contrast between the two instruments. This distinction enables the model to better differentiate their respective class embeddings when processed through the content encoder. Despite the relative ease in acquiring piano data, obtaining a violin dataset that met all project requirements proved challenging. To address this, two separate violin datasets were merged. While both offered only a limited number of tracks, these tracks were of considerable duration. Consequently, multiple audio clips were extracted from each recording to expand the dataset. This preprocessing step necessitated an analysis of the extracted clips to ensure sufficient variability and to avoid segments dominated by prolonged silence, which could compromise training quality and embedding integrity.

First training issues The training of our GAN-based audio style transfer model presented significant numerical stability challenges, represented by nan values in the losses. After some debugging we were able to find the cause which was the standard initialization scheme for the models (Xavier/He), that was leading to gradient explosion and layerNorm collapse. The solution was to drastically reduce the gain in the weight initializations, reducing learning rate schedulers, adding strong gradient clipping and warmup scheduling.

It was also complex to design the GAN architecture carefully, since its multi- objective nature, which needs to alternate the training between the generators and discriminator, and also isolate gradients for the respective phases, which lead to a lot of instability during training, despite the mitigations taken.

Ablation Study on Style Encoder It would be interesting to replace our full Style Encoder for a simpler implementation: either learnable class vectors (one per instrument) or even fixed one-hot labels to see how this affects the network. The idea is to keep everything else the same and just replace the timbre extractor. It could also be interesting to replace other parts of the architecture to see their actual impact.

Experiments with Decoder For the design of the autoregressive decoder, we initially considered the use of a GRU enriched with an explicit attention mechanism. While this solution is computationally lighter compared to a Transformer-based architecture, it was quickly discarded due to concerns about its ability to effectively model long-term temporal dependencies.

We therefore opted for a Transformer Decoder, which—thanks to the self-attention mechanism combined with a causal mask—naturally supports an autoregressive setup. In the initial version, the decoder was flanked by two convolutional modules: an encoder CNN, used to project the ground truth target (y_{target}) from the space $[B, S, C, T, F]$ to $[B, S, d_{\text{model}}]$, and a decoder CNN, employed to map the decoder output back to the original STFT space. This structure also allowed for the use of teacher forcing during training.

However, due to the overall complexity of the model and the unsatisfactory results obtained during training, we decided to drastically simplify the architecture. In particular,

both CNNs were removed and replaced with a simple flattening operation applied to each target frame. During training, the ground truth STFT spectrogram is flattened frame by frame and projected into the model's latent space via a linear layer. The output of the Transformer is then mapped back to the STFT domain through an additional linear layer and finally reshaped to recover the real and imaginary components over time and frequency.

We also conducted a set of experiments using a Vision Transformer (ViT) (Dosovitskiy et al., 2020b) adapted for audio. In this case, the idea was to treat spectral windows as images and divide them into patches, which were then processed as tokens. However, the high number of patches required to represent each frame with sufficient detail led to a significant increase in computational cost, making the model extremely slow—especially during inference. Due to limited computational resources, this line of development was eventually abandoned.

Regarding the cross-attention mechanism between content and style, we adopted a deliberately simple design to avoid excessive architectural complexity. In an initial version, the Transformer Decoder's memory was obtained by concatenating the content embedding (z_{content}) and the style embedding (z_{class}), producing a sequence of shape $[B, S+1, d_{\text{model}}]$.

To encourage more robust learning of stylistic consistency throughout the sequence, we modified this design by repeating z_{class} S times, so that a copy of the style vector would accompany each content frame. The final memory therefore has shape $[B, 2S, d_{\text{model}}]$ and represents a sequence in which, for every content frame, there is a corresponding and identical style vector. This strategy promotes the treatment of z_{class} as a global invariant with respect to time, fostering a clearer separation between content and style in the decoding process.

Training Experiments We experimented with different training strategies: from using reconstruction loss only, to a regular pipeline with single weights emphasizing importances in losses (and here you also need to be very careful with hyperparameter selection and ideally perform finetuning on it), to implementing curriculum learning(Bengio et al., 2009), to ensure stable training convergence and prevent adversarial collapse: in the first phase, we used reconstruction loss only to establish basic STFT generation capabilities; in the second phase we introduced contrastive loss; in the third one disentanglement loss via HSIC. Finally, in the fourth phase, we gradually introduced adversarial training with a dynamic weight.

Despite our experiments, the results were not convincing enough, some losses were oscillating too much indicating instability and the model wasn't reconstructing properly. However we are strongly convinced of the foundations and mathematical proofs behind our choices and we believe that further experiments and refinements could lead to proper learning.

References

Anonymous. Violinetudes, May 2022. URL https://doi.org/10.5281/zenodo.6564408.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 41–48, Montreal, Quebec, Canada, 2009. ACM. doi: 10.1145/1553374.1553380.

Brown, J. C. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1): 425–434, 1991.

Cífka, O., Ozerov, A., Şimşekli, U., and Richard, G. Self-supervised vq-vae for one-shot music style transfer. In *ICASSP*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv*, abs/1810.04805, 2018.

Dong, H.-W., Zhou, C.-Z., Berg-Kirkpatrick, T., and McAuley, J. Bach violin dataset (v1.0), 2021. URL https://doi.org/10.5281/zenodo.4916209. [Data set].

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv*, abs/2010.11929, 2020a.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020b.

Gan, Q., Wang, S., Wu, S., and Zhu, J. Pianomotion10m: Dataset and benchmark for hand motion generation in piano performance, 2025. URL https://arxiv.org/abs/2406.09326.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *arXiv*, abs/1406.2661, 2014.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with hilbert–schmidt norms. *ALT*, 2005.

Griffin, D. and Lim, J. Signal estimation from modified short-time Fourier transform. In *ICASSP*, 1984.

Ho, J. et al. Music style transfer with diffusion model. *arXiv*, abs/2404.14771, 2024.

Huang, P.-C., Kim, M., Hasegawa-Johnson, M., and Smaragdis, P. Phase-aware speech enhancement with deep complex u-net. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2881–2885, 2019a. doi: 10.1109/ICASSP.2019.8683012.

Huang, R. et al. Timbretron: A wavenet(c) on cartoon-style spectrograms for audio timbre transfer. *arXiv*, abs/1904.11842, 2019b.

Hung, Y. et al. Musical composition style transfer via disentangled timbre representations. *arXiv*, abs/1901.01589, 2019.

Lee, J.-H., Park, J., Kim, K. L., and Nam, J. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

Li, W. et al. St2si: Image style transfer via vision transformer using spatial interaction. In *CVPR*, 2024.

Park, K., Woo, S., Kim, D., Cho, D., and Kweon, I. S. Preserving semantic and temporal consistency for unpaired video-to-video translation. *arXiv preprint arXiv:1908.07683*, 2019. URL https://arxiv.org/abs/1908.07683.

Shen, J., Li, Z., Wang, Y., Liu, Y., Wu, Z., and Meng, H. FA-GAN: Artifacts-free and Phase-aware High-fidelity GAN-based Vocoder. In *Proc. INTERSPEECH*, pp. 3884–3888, 2024. doi: 10.21437/Interspeech.2024-862.

Takaki, S. and Yamagishi, J. Stft spectral loss for training a neural speech waveform model. *arXiv preprint arXiv:1810.11945*, 2018. URL https://arxiv.org/abs/1810.11945.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv*, abs/1807.03748, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. Sampling matters in deep embedding learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 2840–2848, 2017.