HARNESSING THE POWER OF DISTRIBUTIONS: PROBABILISTIC REPRESENTATION LEARNING ON HYPERSPHERE FOR MULTIMODAL MUSIC INFORMATION RETRIEVAL — SUPPLEMENTARY MATERIALS —

Takayuki Nakatsuka **Masahiro Hamasaki** **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Japan
{takayuki.nakatsuka, masahiro.hamasaki, m.goto}@aist.go.jp

1. INTRODUCTION

This PDF file provides the supplementary materials for the 25th International Society for Music Information Retrieval Conference (ISMIR2024) paper, entitled "Harnessing the Power of Distributions: Probabilistic Representation Learning on Hypersphere for Multimodal Music Information Retrieval." In the supplementary materials, we present algorithms of the spherical sliced-Wasserstein (SSW) [1] (Section 2.3 in the main paper) and the SSW-based loss function (Section 3.2 in the main paper). In addition, we show the results of additional comparison experiments. Furthermore, we demonstrate multimodal MIR using the multimodal queries.

2. ALGORITHM AND PSEUDOCODE OF THE SPHERICAL SLICED-WASSERSTEIN

2.1 Algorithm for Spherical Sliced-Wasserstein

The definition of the SSW [1] p-distance for $p \geq 1$ is written in the main paper as follows:

$$SSW_p(\mu, \nu) = \int_{\mathbb{V}_{d,2}} W_p \left(\mu \circ P^{U^{-1}}, \nu \circ P^{U^{-1}} \right) \, \mathrm{d}\sigma, \tag{1}$$

where $\mu, \nu \in \mathcal{P}_{p,ac}(S^{d-1})$ are the sets of absolutely continuous probability measures on a hypersphere S^{d-1} with a finite p-th moment, $\mathbb{V}_{d,2} = \{U \in \mathbb{R}^{d \times 2},\ U^\top U = I_2\}$ is the Stiefel manifold [2], σ is the uniform distribution over $\mathbb{V}_{d,2}$, P^U is the function that projects a point $\mathbf{z} \in S^{d-1}$ onto the great circle S^1 generated by U (for $a.e.$ $\mathbf{z} \in S^{d-1}$, P^U can be written in a practical form of $P^U(\mathbf{z}) = \frac{U^\top \mathbf{z}}{\|U^\top \mathbf{z}\|_2}$ [1]), and W_p is the optimal transport distance on S^1 [3, 4].

In our proposed loss function, we used $p = 1$ (i.e., SSW_1). Algorithm 1 presents the procedure of calculating SSW_1.

Algorithm 1 SSW_1

Input: $\zeta_n \sim p(\mathbf{z}_n^* | *_n)$, $\eta_n \sim p(\mathbf{z}_n^\star | \star_n)$ $(*, \star \in \{\mathbf{a}, \mathbf{i}, \mathbf{t}\}, * \neq \star)$
Generate a matrix $E \in \mathbb{R}^{d \times 2}$, where $E \ni e_{ij} \sim \mathcal{N}(0, 1)$
Calculate U by applying the QR factorization to E: $U = \mathrm{QR}(E)$
Project the vectors $z^\zeta \in \zeta_n$, $z^\eta \in \eta_n$ onto the great circle S^1: $\hat{z}^\zeta = \frac{U^\top z^\zeta}{\|U^\top z^\zeta\|_2}$, $\hat{z}^\eta = \frac{U^\top z^\eta}{\|U^\top z^\eta\|_2}$
Calculate the coordinates on one of the generated great circles S^1 by using the atan2 function:
$\tilde{z}^\zeta = \frac{\mathrm{atan2}(-y_{\hat{z}^\zeta}, -x_{\hat{z}^\zeta}) + \pi}{2\pi}$, $\tilde{z}^\eta = \frac{\mathrm{atan2}(-y_{\hat{z}^\eta}, -x_{\hat{z}^\eta}) + \pi}{2\pi}$, where $\hat{z}^\zeta = (x_{\hat{z}^\zeta}, y_{\hat{z}^\zeta})$, $\hat{z}^\eta = (x_{\hat{z}^\eta}, y_{\hat{z}^\eta})$
Calculate the $W_1(\sum \delta_{\tilde{z}^\zeta}, \sum \delta_{\tilde{z}^\eta})$ by using Equation (3) in the main paper
Iterate the calculation of W_1 for $\mathcal{T}$ times: $SSW_1(\zeta, \eta) \approx \frac{1}{\mathcal{T}} \sum^{\mathcal{T}} W_1(\sum \delta_{\tilde{z}^\zeta}, \sum \delta_{\tilde{z}^\eta})$
Output: $SSW_1(\zeta, \eta)$

2.2 Pseudocode for SSW-Based Loss Function

Algorithm 2 presents the pseudocode of SSW_1 for the SSW-based loss function. This pseudocode is written in PyTorch [5]. The SSW-based loss function utilizes parameters of a von Mises-Fisher distribution (the mean direction μ and the concentration κ) for each content item (music audio, image, and text). In this loss function, we first generate von Mises-Fisher distributions from the estimated parameters and then apply a rejection-sampling reparameterization trick [6] to obtain L samples from each distribution. Finally, we calculate the SSW distances between positive probability distributions using the obtained samples to derive a SSW-based loss value.

Algorithm 2 Pseudocode of SSW_1 for SSW-based loss Function

```
# a_mu, a_kappa - estimated parameters of von Mises-Fisher distributions for a mini-batch
    of audio
# i_mu, i_kappa - estimated parameters of von Mises-Fisher distributions for a mini-batch
    of image
# t_mu, t_kappa - estimated parameters of von Mises-Fisher distributions for a mini-batch
    of text

# VonMisesFisher - ``torch.distributions.Distribution'' implementation of a von Mises-
    Fisher distribution. We used the code available at ``https://github.com/nicola-decao/s-
    vae-pytorch/blob/master/hyperspherical_vae/distributions/von_mises_fisher.py'' for our
     implementation.
# L - the number of samples obtained from each probability distribution

# SSW_1 - the spherical sliced-Wasserstein (SSW) distance. Our implementation is based on
    the code available at ``https://github.com/clbonet/Spherical_Sliced-Wasserstein/blob/
    main/lib/sw_sphere.py.''

# SSW-based loss Function
SBLossFunction(a_mu, a_kappa, i_mu, i_kappa, t_mu, t_kappa):

   # generate von Mises-Fisher distributions and apply a rejection-sampling
       reparameterization trick
   p_audio = VonMisesFisher(a_mu, a_kappa).rsample(L)
   p_image = VonMisesFisher(i_mu, i_kappa).rsample(L)
   p_text = VonMisesFisher(t_mu, t_kappa).rsample(L)

   # calculate SSW distances between positive (i.e., the same indices within a mini-batch)
        probability distributions (Equation (7))
   p_distance = (SSW_1(p_audio, p_image) + SSW_1(p_image, p_text) + SSW_1(p_text, p_audio)
       ) / 3

   return p_distance
```

3. ADDITIONAL COMPARISON EXPERIMENTS MENTIONED IN THE MAIN PAPER

3.1 Training runtime

The computation of the SSW-based loss function is highly efficient as shown in Table 1. Although our proposed method additionally computes the SSW-based loss function, both the baseline method and our method were almost equivalent in terms of the training runtime. The proposed method is thus practical and useful for MIR tasks.

Table 1. Comparison for training runtime.

Dataset	Method	Runtime (sec/triplet)
YT8M-MusicVideo	Baseline	0.1050 ± 0.008
	Proposed	0.1096 ± 0.009
AS5M	Baseline	0.05624 ± 0.0002
	Proposed	0.05629 ± 0.0003

3.2 Recall@k

While we reported results for only R@1 in the main paper, we here report results for larger k of R@k to provide a deeper understanding of the performance of each method. We set k to 5, 10, and 15 in this additional comparison experiment. Tables 2–4 show the results for R@k on the YT8M-MusicVideo dataset, while Tables 5–7 show the results for R@k on the AS5M dataset. We confirmed that our methods outperformed the competitive and baseline methods in all the retrieval tasks.

Table 2. Performance of R@k on YT8M-MusicVideo dataset for multimodal image retrieval.

	Audio → Image			Text → Image			Audio & Text → Image		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
PCME	–	–	–	2.82 ± 0.55	4.78 ± 0.69	6.73 ± 0.9	–	–	–
MPC	–	–	–	1.32 ± 0.29	2.83 ± 0.13	4.1 ± 0.07	–	–	–
Baseline	2.5 ± 0.16	4.67 ± 0.31	6.32 ± 0.18	5.85 ± 0.15	9.6 ± 0.45	12.8 ± 0.59	5.52 ± 0.12	8.65 ± 0.41	11.53 ± 0.62
Proposed	$\mathbf{3.45 \pm 0.44}$	$\mathbf{5.85 \pm 0.29}$	$\mathbf{7.82 \pm 0.52}$	$\mathbf{15.13 \pm 0.14}$	$\mathbf{21.18 \pm 0.21}$	$\mathbf{24.75 \pm 0.32}$	$\mathbf{15.45 \pm 0.67}$	$\mathbf{21.15 \pm 0.4}$	$\mathbf{25.37 \pm 0.55}$

Table 3. Performance of R@k on YT8M-MusicVideo dataset for multimodal text retrieval.

	Audio → Text			Image → Text			Audio & Image → Text		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
PCME	–	–	–	2.58 ± 0.15	4.57 ± 0.1	6.0 ± 0.52	–	–	–
MPC	–	–	–	1.2 ± 0.2	2.55 ± 0.04	3.72 ± 0.31	–	–	–
Baseline	2.93 ± 0.17	5.22 ± 0.16	7.27 ± 0.2	6.02 ± 0.2	9.88 ± 0.42	12.83 ± 0.45	6.73 ± 0.05	11.55 ± 0.29	15.15 ± 0.19
Proposed	$\mathbf{4.68 \pm 0.44}$	$\mathbf{7.73 \pm 0.42}$	$\mathbf{10.4 \pm 0.4}$	$\mathbf{15.28 \pm 0.27}$	$\mathbf{21.32 \pm 0.21}$	$\mathbf{25.18 \pm 0.3}$	$\mathbf{18.35 \pm 0.51}$	$\mathbf{24.62 \pm 0.8}$	$\mathbf{30.47 \pm 0.49}$

Table 4. Performance of R@k on YT8M-MusicVideo dataset for multimodal audio retrieval.

	Image → Audio			Text → Audio			Image & Text → Audio		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
Baseline	2.15 ± 0.23	4.27 ± 0.49	5.97 ± 0.49	3.08 ± 0.24	5.4 ± 0.37	7.58 ± 0.51	3.93 ± 0.1	6.98 ± 0.08	9.52 ± 0.2
Proposed	$\mathbf{3.15 \pm 0.21}$	$\mathbf{5.98 \pm 0.39}$	$\mathbf{8.02 \pm 0.45}$	$\mathbf{4.98 \pm 0.12}$	$\mathbf{8.92 \pm 0.09}$	$\mathbf{11.48 \pm 0.14}$	$\mathbf{6.35 \pm 0.2}$	$\mathbf{10.17 \pm 0.37}$	$\mathbf{13.52 \pm 0.36}$

Table 5. Performance of R@k on AS5M dataset for multimodal image retrieval.

	Audio → Image			Text → Image			Audio & Text → Image		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
PCME	–	–	–	9.24 ± 0.67	14.33 ± 0.81	18.0 ± 0.68	–	–	–
MPC	–	–	–	2.94 ± 0.36	5.17 ± 0.54	7.45 ± 0.58	–	–	–
Baseline	5.6 ± 0.39	9.77 ± 0.33	13.24 ± 0.47	17.43 ± 0.71	25.35 ± 0.83	30.66 ± 1.22	13.71 ± 0.74	21.04 ± 0.58	26.12 ± 0.48
Proposed	$\mathbf{9.9 \pm 0.75}$	$\mathbf{15.74 \pm 0.51}$	$\mathbf{20.22 \pm 0.61}$	$\mathbf{65.67 \pm 0.65}$	$\mathbf{72.66 \pm 0.77}$	$\mathbf{76.25 \pm 0.57}$	$\mathbf{63.15 \pm 0.6}$	$\mathbf{70.88 \pm 0.74}$	$\mathbf{75.04 \pm 0.75}$

Table 6. Performance of R@k on AS5M dataset for multimodal text retrieval.

	Audio → Text			Image → Text			Audio & Image → Text		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
PCME	–	–	–	8.9 ± 0.5	13.97 ± 0.75	17.78 ± 0.73	–	–	–
MPC	–	–	–	2.76 ± 0.45	5.23 ± 0.65	7.33 ± 0.7	–	–	–
Baseline	8.17 ± 0.36	13.6 ± 0.46	18.23 ± 0.7	17.7 ± 0.62	25.6 ± 0.84	31.24 ± 0.57	20.24 ± 0.98	29.88 ± 1.05	37.11 ± 0.99
Proposed	$\mathbf{15.74 \pm 0.58}$	$\mathbf{23.8 \pm 0.65}$	$\mathbf{29.82 \pm 0.99}$	$\mathbf{65.78 \pm 0.62}$	$\mathbf{72.03 \pm 0.68}$	$\mathbf{75.58 \pm 0.7}$	$\mathbf{70.24 \pm 0.79}$	$\mathbf{77.14 \pm 0.49}$	$\mathbf{80.75 \pm 0.64}$

Table 7. Performance of R@k on AS5M dataset for multimodal audio retrieval.

	Image → Audio			Text → Audio			Image & Text → Audio		
Method	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑	R@5 (%) ↑	R@10 (%) ↑	R@15 (%) ↑
Baseline	5.36 ± 0.4	9.58 ± 0.49	13.14 ± 0.52	8.8 ± 0.36	14.89 ± 0.42	19.51 ± 0.82	8.88 ± 0.55	15.2 ± 0.63	20.01 ± 0.89
Proposed	$\mathbf{9.8 \pm 0.55}$	$\mathbf{15.89 \pm 0.57}$	$\mathbf{20.08 \pm 0.78}$	$\mathbf{16.3 \pm 0.71}$	$\mathbf{24.87 \pm 0.78}$	$\mathbf{30.67 \pm 0.95}$	$\mathbf{17.88 \pm 0.75}$	$\mathbf{26.77 \pm 0.99}$	$\mathbf{33.18 \pm 0.96}$

3.3 Hyperparameter λ_S

Our contribution lies in proposing and integrating two distinct losses, multimodal probabilistic contrastive loss $\mathcal{L}_C$ and SSW-based loss (optimal transport-based loss) $\mathcal{L}_S$, for probabilistic representation learning. The proposed loss function $\mathcal{L}$ is written in the main paper as follows:

$$\mathcal{L} = \mathcal{L}_C + \lambda_S \mathcal{L}_S. \tag{2}$$

Our method does not work as expected when either loss is overly emphasized (i.e., extremely small or large λ_S value) because their roles for optimizing the encoders are quite different: $\mathcal{L}_C$ loss is designed for distancing irrelevant pairs on the shared hyper-spherical surface S^{d-1}_{shared}, and $\mathcal{L}_S$ loss is for placing positive pairs close to each other and matching their distributional representations.

We therefore conducted comparison experiments regarding the weight λ_S as described in the main paper. We here set λ_S to 0.01, 0.1, 1.0, 10.0, and 100.0. Note that we utilized $\lambda_S = 1.0$ to report the results of both the quantitative evaluations and the qualitative analysis in the main paper. The experimental results are shown in Tables 8 and 9. We found that the optimal λ_S value was 1.0, and that a larger λ_S leads to a decline in performance.

Table 8. Comparison of weight λ_S on YT8M-MusicVideo dataset for multimodal queries.

		Audio & Text → Image			Audio & Image → Text			Image & Text → Audio		
Method	λ_S	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
Proposed	0.01	0.105 ± 0.002	5.6 ± 0.19	78	0.129 ± 0.002	7.9 ± 0.56	62	0.044 ± 0.002	1.7 ± 0.23	151
Proposed	0.1	0.11 ± 0.003	5.8 ± 0.25	74	0.135 ± 0.002	7.6 ± 0.44	57	0.044 ± 0.003	1.4 ± 0.22	148
Proposed	1.0	$\mathbf{0.119 \pm 0.002}$	$\mathbf{6.8 \pm 0.29}$	**72**	$\mathbf{0.139 \pm 0.002}$	$\mathbf{7.97 \pm 0.46}$	**55**	$\mathbf{0.05 \pm 0.002}$	$\mathbf{1.75 \pm 0.25}$	**141**
Proposed	10	0.071 ± 0.001	4.37 ± 0.23	118	0.087 ± 0.003	5.87 ± 0.42	99	0.035 ± 0.003	1.08 ± 0.19	166
Proposed	100	0.077 ± 0.001	3.72 ± 0.13	122	0.082 ± 0.001	3.82 ± 0.19	112	0.017 ± 0.001	0.3 ± 0.11	329

Table 9. Comparison of weight λ_S on AS5M dataset for multimodal queries.

		Audio & Text → Image			Audio & Image → Text			Image & Text → Audio		
Method	λ_S	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
Proposed	0.01	0.504 ± 0.006	41.08 ± 0.95	**2**	0.573 ± 0.008	47.0 ± 0.93	**2**	0.116 ± 0.005	4.8 ± 0.35	41
Proposed	0.1	0.502 ± 0.005	40.78 ± 0.76	**2**	0.573 ± 0.009	46.88 ± 1.02	**2**	0.117 ± 0.005	4.9 ± 0.38	40
Proposed	1.0	$\mathbf{0.508 \pm 0.008}$	$\mathbf{41.35 \pm 1.12}$	**2**	$\mathbf{0.58 \pm 0.009}$	$\mathbf{47.75 \pm 1.19}$	**2**	$\mathbf{0.126 \pm 0.006}$	$\mathbf{5.54 \pm 0.62}$	**37**
Proposed	10	0.447 ± 0.006	35.04 ± 0.79	3	0.533 ± 0.007	42.5 ± 1.01	**2**	0.109 ± 0.004	4.5 ± 0.39	44
Proposed	100	0.397 ± 0.006	30.42 ± 0.71	5	0.457 ± 0.005	35.08 ± 0.61	3	0.065 ± 0.002	2.08 ± 0.27	82

To elucidate this further, we conducted additional experiments for the limit of λ_S values (i.e., the cases of $\lambda \to 0$ and $\lambda \to \infty$). For this experiment, we set $\lambda_S = 1.0 \times 10^{-12}$ to simulate the case of $\lambda_S \to 0$, and $\mathcal{L} = \mathcal{L}_S$ to simulate the case of $\lambda_S \to \infty$. The experimental results are shown in Table 10. The results show that, as expected, the performance is almost the same as the baseline when $\lambda_S \to 0$ because the small weight overshadows the benefit of optimal transport. The results also show that the performance is greatly degraded when $\lambda_S \to \infty$ because $\mathcal{L}_S$ loss can only take care of positive pairs according to its definition and cannot deal with negative (irrelevant) pairs at all. Here, although $\mathcal{L}_C$ loss is expected to place positive pairs close to each other while distancing irrelevant pairs, we can realize that it was not enough and $\mathcal{L}_S$ loss with the appropriate λ_S (1.0) significantly contributed to the performance improvements since it can directly match positive pairs considering their distributional shapes. These results emphasize the importance of our approach that integrates these two losses with the optimal balance.

Table 10. Multimodal retrieval performance on YT8M-MusicVideo dataset for the limit of λ_S values.

		Audio & Text → Image			Audio & Image → Text			Image & Text → Audio		
Method	λ_S	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓	MRR ↑	R@1 (%) ↑	MR ↓
Baseline ($\mathcal{L}_C$ only)	0	0.1 ± 0.004	4.39 ± 0.53	60	0.146 ± 0.007	6.96 ± 0.76	30	0.069 ± 0.003	2.43 ± 0.32	74
Proposed ($\lambda_S \to 0$)	1.0×10^{-12}	0.102 ± 0.006	4.48 ± 0.57	58	0.148 ± 0.011	6.94 ± 0.77	29	0.07 ± 0.004	2.43 ± 0.35	73
Proposed	1.0	0.508 ± 0.008	41.35 ± 1.12	2	0.58 ± 0.009	47.75 ± 1.19	2	0.126 ± 0.006	5.54 ± 0.62	37
$\mathcal{L}_S$ only ($\lambda_S \to \infty$)	-	0.023 ± 0.004	0.88 ± 0.41	482	0.013 ± 0.001	0.3 ± 0.13	602	0.005 ± 0.0	0.05 ± 0.02	941

4. DEMONSTRATION OF MULTIMODAL MIR

A song and its representative image (e.g., a music cover image and a thumbnail image) have the close relationship, making them the subject of extensive research in the MIR community. For example, Libeks et al. [7] found that cover images are closely related to music genres and Oramas et al. [8] applied this insight in music genre classification. Additionally, several studies have explored matching music and images in a latent space (e.g., [9]). Our proposed method enhances the development of such multimodal MIR applications.

As mentioned in the main paper, the primary advantage of probabilistic representation lies in its ability to seamlessly integrate multiple content items in a latent space as a multimodal query, whereas conventional deterministic methods requires additional networks to create such a query [10, 11]. In light of this advantage, we demonstrated multimodal MIR using the multimodal queries (i.e., multimodal image retrieval, multimodal text retrieval, and multimodal audio retrieval) with the test set of the YT8M-MusicVideo dataset as a music collection[1]. For both our method and the baseline method, the respective demonstration shows the content items closest to a different multimodal query on $S^{d-1}_{\mathrm{shared}}$ by calculating the distributional distance between the query and each target content item in the test set. By comparing the retrieved results while viewing and listening to them, we confirmed that our method was qualitatively superior to the baseline method.

5. REFERENCES

[1] C. Bonet, P. Berg, N. Courty, F. Septier, L. Drumetz, and M.-T. Pham, "Spherical sliced-wasserstein," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[2] T. Bendokat, R. Zimmermann, and P.-A. Absil, "A grassmann manifold handbook: Basic geometry and computational aspects," *arXiv preprint arXiv:2011.13699*, 2020.

[3] J. Delon, J. Salomon, and A. Sobolevski, "Fast transport optimization for monge costs on the circle," *SIAM J. Appl. Math.*, vol. 70, no. 7, pp. 2239–2258, 2010.

[4] J. Rabin, J. Delon, and Y. Gousseau, "Transportation distances on the circle," *J. Math. Imaging Vis.*, vol. 41, no. 1, pp. 147–167, 2011.

[5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019, pp. 8024–8035.

[6] T. R. Davidson, L. Falorsi, N. De Cao, T. Kipf, and J. M. Tomczak, "Hyperspherical variational auto-encoders," in *Proceeding of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018, pp. 856–865.

[7] J. Libeks and D. Turnbull, "You can judge an artist by an album cover: Using images for music annotation," *IEEE Trans. Multimed.*, vol. 18, no. 4, pp. 30–37, 2011.

[8] S. Oramas, F. Barbieri, O. Nieto Caballero, and X. Serra, "Multimodal deep learning for music genre classification," *Trans. Int. Soc. Music Inf. Retr.*, vol. 1, no. 1, pp. 4–21, 2018.

[9] B. Xing, K. Zhang, L. Zhang, X. Wu, J. Dou, and S. Sun, "Image–music synesthesia-aware learning based on emotional similarity recognition," *IEEE Access*, vol. 7, pp. 136 378–136 390, 2019.

[10] W. Zhang, F. Qiu, S. Wang, H. Zeng, Z. Zhang, R. An, B. Ma, and Y. Ding, "Transformer-based multimodal information fusion for facial expression analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2428–2437.

[11] S. Ibrahimi, X. Sun, P. Wang, A. Garg, A. Sanan, and M. Omar, "Audio-enhanced text-to-video retrieval using text-conditioned feature alignment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 12 054–12 064.

[1] https://t39nakatsuka.github.io/ISMIR2024-demo/Demo.html