Sorted eigenvalue comparison d_{Eig}: A simple alternative to d_{FID}

Jiqing Wu, Viktor H. Koelzer
Department of Pathology and Molecular Pathology
University Hospital, University of Zurich, Zurich, Switzerland
{Jiqing.Wu, Viktor.Koelzer}@usz.ch

Abstract

For $i = 1, 2$, let $\boldsymbol{S}_i$ be the sample covariance of $\boldsymbol{Z}_i$ with n_i p-dimensional vectors. First, we theoretically justify an improved Fréchet Inception Distance (d_{FID}) algorithm that replaces np.trace(sqrtm($\boldsymbol{S}_1\boldsymbol{S}_2$)) with np.sqrt(eigvals($\boldsymbol{S}_1\boldsymbol{S}_2$)).sum(). With the appearance of unsorted eigenvalues in the improved d_{FID}, we are then motivated to propose sorted eigenvalue comparison (d_{Eig}) as a simple alternative: $d_{\mathsf{Eig}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \sum_{j=1}^{p}(\sqrt{\lambda_j^1} - \sqrt{\lambda_j^2})^2$, and λ_j^i is the j-th largest eigenvalue of $\boldsymbol{S}_i$. Second, we present two main takeaways for the improved d_{FID} and proposed d_{Eig}.

(i) d_{FID}: The error bound for computing non-negative eigenvalues of diagonalizable $\boldsymbol{S}_1\boldsymbol{S}_2$ is reduced to $\mathcal{O}(\varepsilon)\|\boldsymbol{S}_1\|\|\boldsymbol{S}_1\boldsymbol{S}_2\|$, along with reducing the run time by $\sim 25\%$.
(ii) d_{Eig}: The error bound for computing non-negative eigenvalues of sample covariance $\boldsymbol{S}_i$ is further tightened to $\mathcal{O}(\varepsilon)\|\boldsymbol{S}_i\|$, with reducing $\sim 90\%$ run time. Last, we discuss limitations and future work for d_{Eig}.

1 Introduction

In the image domain, it is of great interest to analyze the distribution shift between two collections of data entries [31, 5]. On one hand, this is driven by the increasing awareness about the violation of the assumption of 'identical distribution' between training and (real-world) test datasets [33]. As for instance illustrated in the leaderboard of WILDS [22, 27], many algorithms suffer from performance degradation and fail to generalize to heterogeneous testing settings. On the other hand, the importance of assessing distribution shift has been recognized with the rise of generative adversarial nets (GAN) [13, 15]. The rapid development of GAN variants [19, 20] urges reliable and accurate metric(s) to assess the discrepancy between generated and real images [5].

To objectively assess GAN models, researchers have proposed a plethora of evaluation scores including Inception Score [29], Kernel Inception Distance (d_{KID}) [3], and Precision/Recall [24, 28] (please also see [5, 6] for in-depth review). Among various scores, Fréchet Inception Distance (d_{FID}) [15] is arguably the most widely-used metric for benchmarking GAN performance [26]. This is mainly due to the favorable theoretical property of being a mathematical metric [10] and practical property of being well-correlated with perceived image quality [28]. Meanwhile, Chong and Forsyth [7] argued that d_{FID} is a biased estimator and Kynkäänniemi *et al.* [23] observed its undesirable sensitivity towards fringe features or classes. Despite these weaknesses, d_{FID} currently remains the 'gold standard' for GAN evaluation and continuously attracts broad attention. In a recent study, Mathiasen and Hvilshøj [25] proposed to compute eigenvalues rather than square root of a matrix as in d_{FID}. We view this as a promising simplification and improvement, nonetheless a precise theoretical analysis has not been performed and therefore becomes the starting point of this paper.

Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

With the appearance of unsorted eigenvalues in the improved d_{FID}, we are motivated to propose sorted eigenvalue comparison (d_{Eig}) as a simple alternative. Our contributions are summarized as follows: For $i = 1, 2$, let $\mathbf{S}_i$ be the sample covariance of $\boldsymbol{Z}_i = (\boldsymbol{z}_1^i, \ldots, \boldsymbol{z}_{n_i}^i)$ with n_i p-dimensional vectors.

- (d_{FID}) We articulate the fact that $\mathbf{S}_1\mathbf{S}_2$ is diagonalizable and has non-negative eigenvalues. This allows us to theoretically justify an improved algorithm of d_{FID}, *i.e.*, by replacing the unique principal square root of a matrix with the element-wise square root of its eigenvalues. Therefore, the error bound for computing its eigenvalues is reduced to $\mathcal{O}(\varepsilon)\|\boldsymbol{S}_1\|\|\boldsymbol{S}_1\boldsymbol{S}_2\|$, reducing the run time by $\sim 25\%$.
- (d_{Eig}) Since $\mathbf{S}_i$ is symmetric positive semidefinite, the error bound for computing its non-negative eigenvalues is further tightened to $\mathcal{O}(\varepsilon)\|\boldsymbol{S}_i\|$, with reducing $\sim 90\%$ run time.

2 The improved d_{FID} and proposed d_{Eig}

(Linear Algebra) Notation: Lower case Roman or Greek letters (*e.g.*, $s, \epsilon, \gamma, \lambda$) denote scalars, bold lower case letters (*e.g.*, $\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{\mu}$) denote vectors, and bold upper case letters (*e.g.*, $\boldsymbol{Q}, \boldsymbol{S}, \boldsymbol{U}, \boldsymbol{Z}, \boldsymbol{\Lambda}$) denote matrices. $^\mathsf{T}$ is matrix transpose, $\|.\|$ is L^2 norm, $\lesssim$ denotes asymptotically less than.

2.1 Principal square root of a matrix

Without loss of accuracy, we discuss d_{FID} through the lens of linear algebra. More specifically, scalars, vectors and matrices discussed in the section are deterministic. For $i = 1, 2$, let $\boldsymbol{Z}_i = (\boldsymbol{z}_1^i, \ldots, \boldsymbol{z}_{n_i}^i)$ be a collection of n_i p-dimensional vectors. For simplicity, we assume sample mean $\frac{1}{n_i}\sum_{k=1}^{n_i} \boldsymbol{z}_k^i = \mathbf{0}$ throughout Sec. 2. Accordingly, $\boldsymbol{S}_i = \frac{1}{n_i}\boldsymbol{Z}_i\boldsymbol{Z}_i^\mathsf{T}$ denotes the sample covariance matrix (SCM) of $\boldsymbol{Z}_i$. We start the discussion with revisiting standard the definition(s) of d_{FID} [12], then we elaborate the properties of principal square root – the key computational challenge of d_{FID}.

2.1.1 $\mathsf{Trace}((\mathbf{S}_1^{\frac{1}{2}}\mathbf{S}_2\mathbf{S}_1^{\frac{1}{2}})^{\frac{1}{2}})$

Definition 1. *Let $\boldsymbol{S}_i$ be the SCM of $\boldsymbol{Z}_i$ and w.l.o.g. $\boldsymbol{S}_1$ is non-singular, then we define*

$$d_{\mathsf{FID}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \mathsf{Trace}(\boldsymbol{S}_1 + \boldsymbol{S}_2 - 2(\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}})^{\frac{1}{2}}). \tag{1}$$

To compute the $\mathsf{Trace}()$ of d_{FID}, we first need to clarify the symbol $^{\frac{1}{2}}$ in Eq. 1. As mentioned in [10], $^{\frac{1}{2}}$ denotes the positive (or principal [16]) square root of a matrix $\boldsymbol{S}$ such that $\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{S}^{\frac{1}{2}} = \boldsymbol{S}$, and 'principal' specifies the square root(s) $\boldsymbol{S}^{\frac{1}{2}}$ with non-negative eigenvalues. In general, the square root of a matrix may neither exist nor be unique [16]. Consider now the special case where $\boldsymbol{S}$ is symmetric positive semidefinite (PSD), then we have

Theorem 2. ***(Principal square root)*** *Let a symmetric PSD $\boldsymbol{S}$ be decomposed as $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\mathsf{T}$, where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a diagonal matrix with non-negative eigenvalues, then $\boldsymbol{S}^{\frac{1}{2}} := \boldsymbol{Q}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Q}^\mathsf{T}$ is the unique principal square root of $\boldsymbol{S}$.*

Based on the definition of $\boldsymbol{S}^{\frac{1}{2}}$, it is easy to see that $\boldsymbol{S}^{\frac{1}{2}}\boldsymbol{S}^{\frac{1}{2}} = \boldsymbol{S}$. Importantly, $\boldsymbol{S}^{\frac{1}{2}}$ is unique in the sense that if there exists another symmetric SPD $\tilde{\boldsymbol{S}}^{\frac{1}{2}}$ such that $\tilde{\boldsymbol{S}}^{\frac{1}{2}}\tilde{\boldsymbol{S}}^{\frac{1}{2}} = \boldsymbol{S}$, then $\tilde{\boldsymbol{S}}^{\frac{1}{2}} = \boldsymbol{S}^{\frac{1}{2}}$. With Thm. 2 in hand and given the fact that $\boldsymbol{S}_1^{\frac{1}{2}}$ and $\boldsymbol{S}_2$ are symmetric PSD, we make the following claim.

Corollary 3. *$\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}}$ is a symmetric PSD matrix and therefore has a unique principal square root.*

Claim. Accordingly, $\mathsf{Trace}((\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}})^{\frac{1}{2}})$ (Eq. 1) can be derived after eigenvalue decomposition suggested in Thm. 2. As a computational routine, this formulation is nevertheless undesirable. Because we need to call the eigenvalue decomposition function twice, which potentially increases computational time and error risk. Instead, we seek for another equivalent formulation of Eq. 1 [12].

2.1.2 $\mathsf{Trace}((\mathbf{S}_1\mathbf{S}_2)^{\frac{1}{2}})$

Lemma 4. *Following the specifications of $\boldsymbol{S}_i$ in Eq. 1, then we have*

$$d_{\mathsf{FID}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \mathsf{Trace}(\boldsymbol{S}_1 + \boldsymbol{S}_2 - 2(\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}}). \tag{2}$$

Because of non-singular $\boldsymbol{S}_1$, it is not difficult to see that eigenvalues of $\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}}$ and $\boldsymbol{S}_1\boldsymbol{S}_2$ are identical, and the corresponding eigenvectors are identical up to an invertible linear transformation $\boldsymbol{S}_1^{\frac{1}{2}}$ (or $\boldsymbol{S}_1^{-\frac{1}{2}}$). Due to the fact that $\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}}$ is symmetric PSD, then we have

Corollary 5. *$\boldsymbol{S}_1\boldsymbol{S}_2$ is a diagonalizable matrix with non-negative eigenvalues and therefore has a unique principal square root.*

Remark 6. *Note that at least one of $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ should be non-singular, or the null space of $\boldsymbol{S}_1$ should be contained in that of $\boldsymbol{S}_2$ [10]. If $\boldsymbol{S}_1$ is singular, then the above discussions remain the same after switching the role of $\boldsymbol{S}_1$ and non-singular $\boldsymbol{S}_2$.*

Claim. Consequently, eigenvalues of $(\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}}$ are mathematically equivalent to the element-wise square root of eigenvalues of $\boldsymbol{S}_1\boldsymbol{S}_2$. Since $\boldsymbol{S}_1\boldsymbol{S}_2$ and $\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}}$ have identical eigenvalues and the trace of a diagonalizable matrix is the sum over its eigenvalues, then we have $\mathsf{Trace}((\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}}) = \mathsf{Trace}((\boldsymbol{S}_1^{\frac{1}{2}}\boldsymbol{S}_2\boldsymbol{S}_1^{\frac{1}{2}})^{\frac{1}{2}})$ and prove Lem. 4. Importantly, this rigorously justifies the workaround algorithm of d_{FID} that computes the element-wise square root of eigenvalues, which bypasses the expansive computation of the square root of a matrix.

2.2 Element-wise square root of eigenvalues

Before substituting the square root component of d_{FID}, let us take a step back and re-examine its widely-used implementation scipy.lingalg.sqrtm()[1]. In a nutshell, the underlying computational routine is a blocked Schur algorithm [4, 9], which includes two phases: Schur decomposition (schur()) and solving (triangular) Sylvester equation. For computing $\mathsf{Trace}()$ of $p \times p$ diagonalizable matrix $(\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}}$, we show the latter phase is redundant.

2.2.1 Numerical Error bound

Corollary 7. ***(Schur decomposition)*** *Let the diagonalizable matrix $\boldsymbol{S}_1\boldsymbol{S}_2$ be decomposed as $\boldsymbol{Q}\boldsymbol{U}\boldsymbol{Q}^{\mathsf{T}}$. Here, $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{U}$ is an upper triangular matrix. Then $\mathsf{Trace}((\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}}) = \sum_{j=1}^{p}\sqrt{u_{jj}}$, where $u_{11}, \ldots, u_{pp}$ are diagonal entries of $\boldsymbol{U}$.*

This equation is derived from the fact that diagonal entries of $\boldsymbol{U}$ are exactly (non-negative) eigenvalues of $\boldsymbol{S}_1\boldsymbol{S}_2$. As an immediate consequence, it suffices to compute schur() for obtaining $\mathsf{Trace}((\boldsymbol{S}_1\boldsymbol{S}_2)^{\frac{1}{2}})$. By default, schur() simultaneously computes $\boldsymbol{U}$ and $\boldsymbol{Q}$. In our case, we only want to compute diagonal entries of $\boldsymbol{U}$. This leads to a more speedy eigvals() that shares the same core routine as schur(). Eventually, we replace the standard pythonic implementation np.trace(sqrtm()) with np.sqrt(eigvals()).sum() (See [25] for reference). Such a series of algorithmic simplification allows us to propose a (strictly) tighter error bound compared to the original case *w.r.t.* sqrtm().

Error bound of eigvals(). As discussed in [2], for the computed eigenvalue $\hat{\gamma}_j$ and eigenvalue γ_j of $\boldsymbol{S}_1\boldsymbol{S}_2$ we have $|\hat{\gamma}_j - \gamma_j| \lesssim \mathcal{O}(\epsilon)s_j^{-1}\|\boldsymbol{S}_1\boldsymbol{S}_2\|$, where ϵ is machine epsilon. The remaining task is to compute s_j. Since if $\boldsymbol{v}_j$ is the right eigenvector for γ_j, then the left eigenvector is $\boldsymbol{v}_j^{\mathsf{T}}\boldsymbol{S}_1^{-1}$. Because of $s_j := |\boldsymbol{v}_j^{\mathsf{T}}\boldsymbol{S}_1^{-1}\boldsymbol{v}_j| = \|\boldsymbol{S}_1^{-\frac{1}{2}}\boldsymbol{v}_j\|^2$ we have $s_j^{-1} \leq \|\boldsymbol{S}_1^{\frac{1}{2}}\|^2 = \|\boldsymbol{S}_1\|$. For $j = 1, \ldots, p$, the (asymptotic) error bound for computing eigenvalue γ_j can be formulated as

$$|\hat{\gamma}_j - \gamma_j| \lesssim \mathcal{O}(\epsilon)\|\boldsymbol{S}_1\|\|\boldsymbol{S}_1\boldsymbol{S}_2\|. \tag{3}$$

Moreover, if we want to compute eigenvalues of the $p \times p$ SCM $\boldsymbol{S}_i$ that is symmetric PSD, we can utilize the eigvalsh() with lower run time and obtain a tighter error bound.

Error bound of eigvalsh(). For $j = 1, \ldots, p$, the error bound for computing eigenvalue λ_j^i of $\boldsymbol{S}_i$ can be formulated as [2]

$$|\hat{\lambda}_j^i - \lambda_j^i| \leq \mathcal{O}(\epsilon)\|\boldsymbol{S}_i\|. \tag{4}$$

2.2.2 Eigenvalue comparison

Now, we discuss an important variant of d_{FID} when $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ commute, *i.e.*, $\boldsymbol{S}_1\boldsymbol{S}_2 = \boldsymbol{S}_2\boldsymbol{S}_1$.

[1] `https://github.com/GaParmar/clean-fid/blob/main/cleanfid/fid.py`

Corollary 8. ***(Unsorted eigenvalue comparison)*** *Let $\boldsymbol{S}_{1,2}$ be two SCMs that are simultaneously diagonalizable by an orthogonal matrix $\boldsymbol{Q}$, then*

$$d_{\mathsf{FID}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \mathsf{Trace}((\boldsymbol{S}_1^{\frac{1}{2}} - \boldsymbol{S}_2^{\frac{1}{2}})^2) = \sum_{j=1}^{p} (\sqrt{\tilde{\lambda}_j^1} - \sqrt{\tilde{\lambda}_j^2})^2, \tag{5}$$

where $\tilde{\lambda}_j^i$ is the j-th eigenvalue of $\boldsymbol{S}_i$ w.r.t. $\boldsymbol{Q}$.

Under such a special case where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ share the same eigenbasis, d_{FID} is reduced to computing the Euclidean distance between unsorted eigenvalues. Motivated by this reduction, we propose to compare sorted eigenvalues as a simple alternative to d_{FID}.

Definition 9. ***(Sorted eigenvalue comparison)*** *Let $\boldsymbol{S}_{1,2}$ be two SCMs, then we define*

$$d_{\mathsf{Eig}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \sum_{j=1}^{p} (\sqrt{\lambda_j^1} - \sqrt{\lambda_j^2})^2, \tag{6}$$

*where λ_j^i is the j-th **largest** eigenvalue of $\boldsymbol{S}_i$. Accordingly, d_{Eig} is a pseudometric on the set of SCMs with order p.*

Note that $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ in Eq. 6 do not necessarily commute. Instead of eigvals() used for computing eigenvalues of non-symmetric $\boldsymbol{S}_1\boldsymbol{S}_2$ (Eq. 2), d_{Eig} can be obtained with a more numerically stable and faster eigvalsh(), which is customized to compute λ_j^i of symmetric $\boldsymbol{S}_i$. As a pseudometric, d_{Eig} satisfies non-negativity, symmetry and triangular inequality, while SCMs need not to be indistinguishable regarding d_{Eig}. In the toy studies (App. A), we show that d_{Eig} is more reliable than d_{FID}. In the GAN studies (App. B), we demonstrate that d_{Eig} is a simple alternative to d_{FID}.

3 Limitations and Future work

3.1 Eigenvector

In contrast to the improved d_{FID} that implicitly takes eigenvectors of $\boldsymbol{S}_i$ into account via the matrix multiplication $\boldsymbol{S}_1\boldsymbol{S}_2$, the proposed d_{Eig} only measures the eigenvalue difference. Admittedly, the exclusion of eigenvectors in d_{Eig} is mainly due to the disencouraging properties such as more loose numerical error bound [2] and more strict conditions for distribution estimation [21]. Nevertheless, eigenvectors carry plausibly critical information and should be carefully examined in subsequent work.

3.2 Future studies: d_{Eig} may be more comprehensive and informative than d_{FID}.

Similar to existing measurements, d_{Eig} remains a scalar-valued score for measuring high-dimensional distribution shifts. A more comprehensive quantification is still missing for applications in both the natural and medical image domains. Due to the inherent data heterogeneity and critical implications for real-world application, facilitating in-depth analysis of distribution shifts underlying high-dimensional images (or representations) is of key importance to support the development and application of high-quality data science approaches, *e.g.*, in the medical domain [34, 8]. In such a scenario where inaccurate analysis can have severe consequences, existing scalar-valued scores including d_{Eig} is not sufficient. To resolve this issue, a direct follow-up on d_{Eig} is to individually compare the eigenvalue difference along each dimension. Naturally, the scalar-valued d_{Eig} is decomposed to a multi-dimensional vector-valued measurement and enables a more complete overview of data heterogeneity. In addition, the d_{Eig} builds the bridge between the classical principal component analysis (PCA) [1] and latent semantic understanding [30, 14].

Taking cancer studies as an example [11, 32], the fine-grained multi-dimensional analysis with d_{Eig} could pave a promising way towards precise risk stratification by validating well-established and proposing novel features with prognostic importance in complex medical images. This can be concretely supported with biologically interpretable visualization examples generated by perturbing the largest eigenvalue(s)/eigenvector(s) in a given dataset of interest. This approach could thus be used to control for inherent variance in existing data repositories and generate prototypical examples of disease states such as highly aggressive tumors in radiological or pathological time series. In combination of comprehensive quantification and biologically meaningful visualization, d_{Eig} thus adds a valuable tool for future work in the natural and medical image domains.

4 Implementation

```python
import numpy as np
from scipy.linalg import eigvals, eigvalsh

# The square of improved d_FID
def dFID(mean1, cov1, mean2, cov2):
    eigval = eigvals(cov1 @ cov2)
    # Round computational errors (if exist)
    # that lead to negative eigenvalues close to 0
    eigval[eigval < 0] = 0
    dif = mean1 - mean2
    res = dif.dot(dif) + np.trace(cov1 + cov2)
    return res - 2 * np.sqrt(eigval).sum()

# The square of proposed d_Eig
def dEig(scm1, scm2):
    # Sorted eigenvalues
    eigval1 = eigvalsh(scm1)
    eigval1[eigval1 < 0] = 0
    eigval2 = eigvalsh(scm2)
    eigval2[eigval2 < 0] = 0
    dif = np.sqrt(eigval1) - np.sqrt(eigval2)
    return dif.dot(dif)
```

Figure 1: Python codes for the square of improved d_{FID} and proposed d_{Eig}.

5 Acknowledgements

We would like to thank Prof. Yukun He and Prof. Zhigang Bao for insightful theoretical discussions.

References

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Edward Anderson, Zhaojun Bai, Christian Bischof, L Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al. *LAPACK users' guide*. SIAM, 1999.

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018.

[4] Åke Björck and Sven Hammarling. A schur method for the square root of a matrix. *Linear algebra and its applications*, 52:127–140, 1983.

[5] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[6] Ali Borji. Pros and cons of gan evaluation measures: New developments. *Computer Vision and Image Understanding*, 215:103329, 2022.

[7] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079, 2020.

[8] Krzysztof J Cios and G William Moore. Uniqueness of medical data mining. *Artificial intelligence in medicine*, 26(1-2):1–24, 2002.

[9] Edvin Deadman, Nicholas J Higham, and Rui Ralha. Blocked schur algorithms for computing the matrix square root. In *International Workshop on Applied Parallel Computing*, pages 171–182. Springer, 2012.

[10] DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[11] Sarah Fremond, Viktor Hendrik Koelzer, Nanda Horeweg, and Tjalling Bosse. The evolving role of morphology in endometrial cancer diagnostics: From histopathology and molecular testing towards integrative data analysis by deep learning. *Frontiers in Oncology*, 12, 2022.

[12] Clark R Givens and Rae Michael Shortt. A class of wasserstein metrics for probability distributions. *Michigan Mathematical Journal*, 31(2):231–240, 1984.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33:9841–9850, 2020.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[16] Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[17] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

[18] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[21] Antti Knowles and Jun Yin. Eigenvector distribution of wigner matrices. *Probability Theory and Related Fields*, 155(3):543–582, 2013.

[22] Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[23] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. *arXiv preprint arXiv:2203.06026*, 2022.

[24] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[25] Alexander Mathiasen and Frederik Hvilshøj. Backpropagating through fr\'echet inception distance. *arXiv preprint arXiv:2009.14075*, 2020.

[26] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in gan evaluation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11410–11420, 2022.

[27] Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, et al. Extending the wilds benchmark for unsupervised adaptation. *arXiv preprint arXiv:2112.05090*, 2021.

[28] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *Advances in Neural Information Processing Systems*, 31, 2018.

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[30] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021.

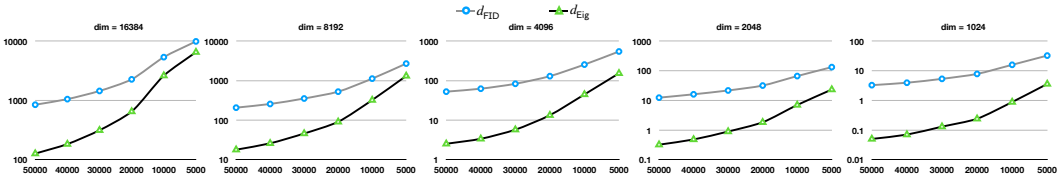
[31] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

[32] Jiqing Wu, Nanda Horeweg, Marco de Bruyn, Remi A Nout, Ina M Jürgenliemk-Schulz, Ludy CHW Lutgens, Jan J Jobsen, Elzbieta M van der Steen-Banasik, Hans W Nijman, Vincent THBM Smit, et al. Automated causal inference in application to randomized controlled clinical trials. *Nature Machine Intelligence*, 2022.

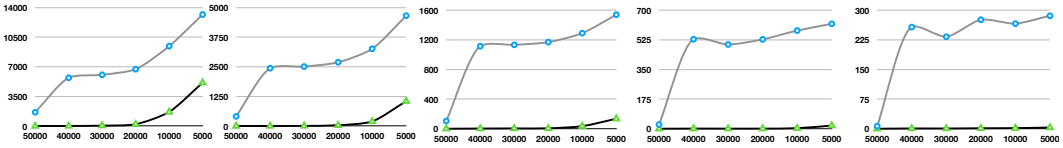
[33] Jiqing Wu, Inti Zlobec, Maxime Lafarge, Yukun He, and Viktor H Koelzer. Towards iid representation learning and its application on biomedical data. *arXiv preprint arXiv:2203.00332*, 2022.

[34] Lin Yue, Dongyuan Tian, Weitong Chen, Xuming Han, and Minghao Yin. Deep learning for heterogeneous medical data analysis. *World Wide Web*, 23(5):2715–2737, 2020.

A Toy Studies: d_{Eig} is more reliable than d_{FID}.



a. (Oracle comparison) The d_{FID} and d_{Eig} scores computed between sample covariance of 50k, 40k, ..., 5k data VS ground-truth covariance.



b. (Real-world comparison)The d_{FID} and d_{Eig} scores computed between sample covariance of 50k, 40k, ..., 5k data VS sample covariance of 50k data.

	d_{FID}			d_{Eig}	
dim	**sqrtm**	**schur**	**eigvals**	**svdvals**	**eigvalsh**
16384	1101.04 (1.21)	968.41 (2.47)	**854.16 (8.23)**	160.17 (0.96)	**66.03 (2.50)**
8192	168.72 (0.57)	148.47 (0.79)	**138.84 (2.33)**	20.29 (0.04)	**10.72 (0.68)**
4096	18.42 (0.04)	15.58 (0.07)	**14.47 (0.11)**	2.77 (0.00)	**2.73 (0.02)**
2048	2.34 (0.02)	1.80 (0.02)	**1.58 (0.03)**	0.43 (0.00)	**0.20 (0.00)**
1024	0.42 (0.01)	0.33 (0.00)	**0.32 (0.02)**	0.06 (0.00)	**0.04 (0.00)**

c. The d_{FID} and d_{Eig} run time obtained with different implementation variants.

Figure 2: **The toy studies of multivariate Gaussian distribution** $(\mathbf{0}, \mathbf{\Sigma})$. Here, all the experiments are computed with Intel(R) i9-9940X CPU @ 3.30GHZ and repeated with four random seeds. Since the coefficient of variance $\mathsf{std}/\mathsf{mean} < 0.01$ for both d_{Eig} and d_{FID}, we only report the mean score in Plot a and b. Besides, sqrtm(), schur() and eigvals() achieve identical numerical results for computing d_{FID} up to negligible rounding error. So do svdvals() and eigvalsh() for computing d_{Eig}. Therefore, only two curves are presented in Plot a, b.

Mathematical equivalence between d_{Eig} and d_{FID}. For proof of principle, we conduct toy studies with multivariate Gaussian data. Concretely, we construct non-negative diagonal entries of a p-dim covariance matrix $\mathbf{\Sigma}$ with np.abs(np.random.randn(p)), while keeping the off-diagonal entries zero. By multiplying $\mathbf{\Sigma}^{\frac{1}{2}}$ and $\boldsymbol{X}_i =$ np.random.randn(p, n_i), we obtain n_i Gaussian data entries $\boldsymbol{Y}_i = \mathbf{\Sigma}^{\frac{1}{2}}\boldsymbol{X}_i$ that are drawn from $(\mathbf{0}, \mathbf{\Sigma})$. Then we compare $\boldsymbol{S}_1 = \frac{1}{n_1}\boldsymbol{Y}_1\boldsymbol{Y}_1^{\mathsf{T}}$ to ground-truth $\mathbf{\Sigma}$ (Fig. 2(a)) and to $\boldsymbol{S}_2 = \frac{1}{n_2}\boldsymbol{Y}_2\boldsymbol{Y}_2^{\mathsf{T}}$ (Fig. 2(b)). Following above theoretical discussions, we instantiate Eq. 6 of d_{Eig} with sqrtm(), schur() and eigvals(). Because $\boldsymbol{S}_i$ is a symmetric SPD, we implement Eq. 2 of d_{FID} with svdvals() and eigvalsh(). Throughout our experiments, we notice that the results of implementation variants are identical up to very small rounding errors. Therefore, we experimentally confirm the validity of improved d_{FID} and the equivalency between svdvals() and eigvalsh(). Because identical (sample) covariances are simultaneously diagonalizable, we have $d_{\mathsf{Eig}} = d_{\mathsf{FID}}$ in theory. Since $\boldsymbol{S}_1 \approx \boldsymbol{S}_2 \approx \mathbf{\Sigma}$ with sufficient amount of data, we expect $d_{\mathsf{Eig}} \approx d_{\mathsf{FID}} \approx 0$ in practice.

Numerical difference between d_{Eig} and d_{FID}. When comparing $\boldsymbol{S}_1$ to $\mathbf{\Sigma}$, Fig. 2 (a) shows that d_{Eig} and d_{FID} have a comparable trend of decreasing scores with a growing number of data entries ($5k \rightarrow 50k$). This indicates that both d_{Eig} and d_{FID} are meaningful metrics and can converge to their theoretical limit. When comparing $\boldsymbol{S}_1$ to $\boldsymbol{S}_2$, Fig. 2 (b) illustrates that d_{Eig} is more resistant to the data size difference. In contrast to d_{FID}, it suffices to use a smaller amount of data to achieve a good estimation for d_{Eig}. Arguably, d_{Eig} represents a more reliable score than d_{FID} due to the fact that 1) d_{Eig} demonstrates favorable convergence curves that are overall closer to 0, and 2) in comparison with the standard d_{FID} (Eq. 2), d_{Eig} (Eq. 6) is a more faithful routine to approximate Eq. 5 – the simplified d_{FID} for our toy setting.

Run time. When comparing different variants for implementing d_{Eig} and d_{FID}, Fig. 2 (c) shows $18\% - 32\%$ reduction of run time by replacing sqrtm() with eigvals(), and we further reduce the run time by $85\% - 94\%$ when utilizing eigvalsh(). As a result, it is beneficial to apply the improved d_{Eig} and proposed d_{FID} for computing distribution shifts, especially in the high dimensional cases such as $p = 16384$. From now on, d_{Eig} and d_{FID} are computed with eigvals() and eigvalsh() by default.

B GAN studies: d_{Eig} is a simple alternative to d_{FID}.

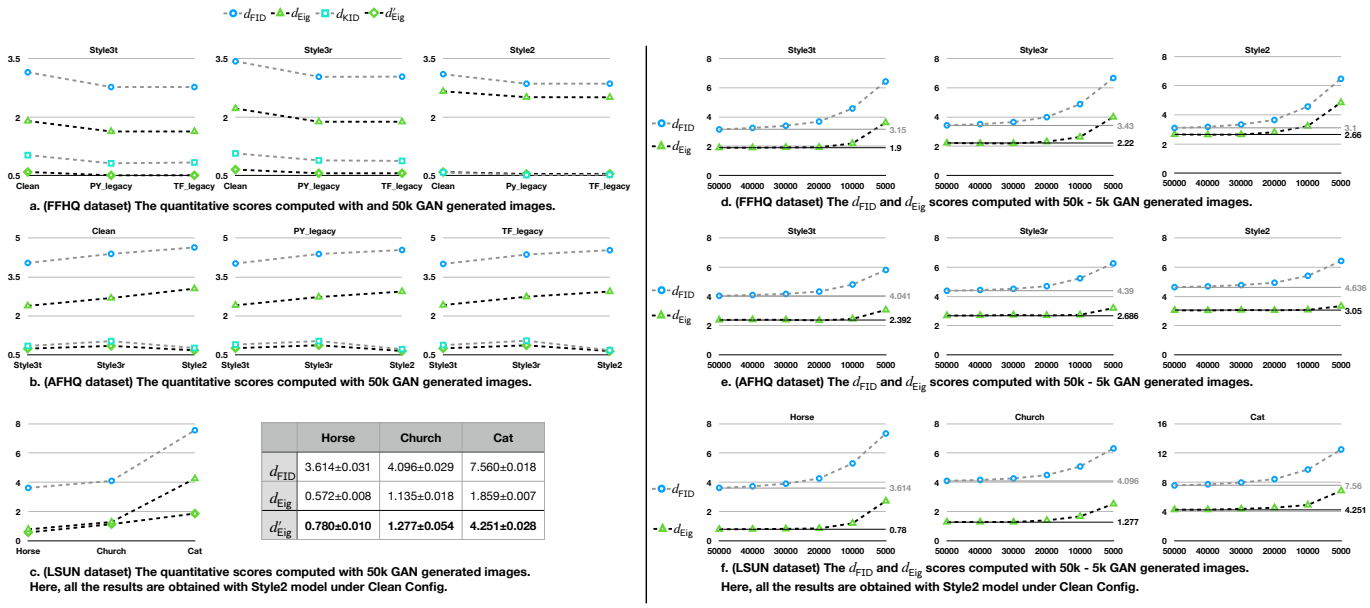
Recently, Parmar *et al*. [26] discovered surprising subtleties of image pre-processing steps for downstream GAN evaluation. To faithfully benchmark the GAN performance of state-of-the-art (sota) models, the authors published new APIs to reproduce the evaluation results. Hence, the implementation of our GAN experiments is built on top of these APIs. Next, we summarize four key aspects of GAN evaluation that we examine in this study.

4 Scores. Similar to [26], we take two widely-used scores d_{FID} and d_{KID} as baselines. Then, we investigate two variants of the proposed metric: $d_{\text{Eig}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \sum_{j=1}^{p}(\sqrt{\lambda_j^1} - \sqrt{\lambda_j^2})^2$ (Eq. 6) and $d'_{\text{Eig}}(\boldsymbol{S}_1, \boldsymbol{S}_2)^2 = \sum_{j=1}^{p}(\sqrt{\bar{\lambda}_j^1} - \sqrt{\bar{\lambda}_j^2})^2 + \|\boldsymbol{m}_1 - \boldsymbol{m}_2\|^2$. The λ_j^i of the former are eigenvalues of $\boldsymbol{S}_i$, and $\bar{\lambda}_j^i$ of the latter are eigenvalues of $\boldsymbol{S}_i - \frac{1}{n_i}\boldsymbol{m}_i\boldsymbol{m}_i^{\mathsf{T}}$, where $\boldsymbol{m}_i$ is the sample mean. Differing from toy settings of Gaussian distribution $(\mathbf{0}, \boldsymbol{\Sigma})$ that lead to $d_{\text{Eig}} \approx d_{\text{FID}} \approx 0$ with sufficient data, we do not have such a theoretical limit or ground-truth score in GAN studies. As a workaround, we consider d_{FID} to be the 'gold standard' score for analyzing d_{Eig}. Without loss of accuracy, we take $d_{\text{KID}} \times 10^3$ and $d_{\text{Eig}} \times 10$ for clearer comparisons.

3 Models. To illustrate the strength of d_{Eig} for challenging cases, we investigate three sota GAN models and probe their nuances when visual evaluations are non-trivial: StyleGAN2 with the recommended Config (**Style2**) [17], StyleGAN3 with translation equivariance Config (**Style3t**) and with translation and rotation equivariance Config (**Style3r**) [18].

3 Interpolations. Following the practice of [26], we also present results that are influenced by different image interpolations such as Clean (**Clean**), PyTorch_legacy (**Py_legacy**) and TensorFlow_legacy (**TF_legacy**).

5 Datasets. Lastly, we run thorough comparisons on commonly-used datasets including FFHQ, AFHQ, and LSUN (Horse, Church, Cat categories) for GAN model training. For each dataset, we generate 100k fake images and repeat each experiment 4 times by randomly sampling a given number of image entries from 100k fake images.



	Horse	Church	Cat
d_{FID}	3.614±0.031	4.096±0.029	7.560±0.018
d_{Eig}	0.572±0.008	1.135±0.018	1.859±0.007
d'_{Eig}	**0.780±0.010**	**1.277±0.054**	**4.251±0.028**

Figure 3: **The main results of GAN studies**. Here, 70k, 15803 and 50k real images for FFHQ, AFHQ and LSUN datasets resp. are applied to compute the reported scores.

In the following, we discuss the main results of our GAN studies. As displayed in Fig. 3 (a, b), d_{Eig} and d_{FID} show similar evaluation curves and correlate well with each other in terms of different combinations of models and interpolations. When observing the convergence curve with an increasing

amount of GAN generated images Fig. 3 (d, f), we observe an identical behavior as in the toy studies. That is, d_{Eig} is more favorable than d_{FID} in the sense that it suffices to use a small amount of image entries to obtain a good estimation for d_{Eig}. Similar claims can be made for the LSUN dataset. As shown in Fig. 3 (c), d_{Eig} illustrates comparably increasing scores from the Horse to the Cat category, indicating less satisfying GAN generation results for the Cat images. Meanwhile, the convergence speed remains faster for d_{Eig} compared to d_{FID} (Fig. 3 (f)). With regard to d'_{Eig}, the variant of sorted eigenvalue comparison show less consistency with the gold standard d_{FID} (See Fig. 3 (a, b)) and is less desirable in our GAN studies. Based on the investigations of the four key aspects and theoretical advantages of d_{Eig} discussed above, the proposed d_{Eig} represents a simple alternative to d_{FID}. By applying d_{Eig} in GAN model evaluation, we take a critical step towards a more comprehensive analysis of high-dim distribution shift between two collections of image entries.

FFHQ	Eigenvalue	Eigenvector
1	259.14±0.157	0.99
2	15.92±0.048	0.002
3	10.24±0.047	0.009
4	7.31±0.024	0.001
5	6.85±0.049	0.013
6	5.41±0.019	0.004
7	3.73±0.023	0.003
8	3.59±0.009	0.002
9	3.11±0.007	0.001
10	2.72±0.001	0.004

AFHQ	Eigenvalue	Eigenvector
1	133.53±0.201	0.99
2	14.43±0.037	0.015
3	9.18±0.034	0.012
4	6.43±0.070	0.009
5	5.5±0.039	0.003
6	4.25±0.016	0.003
7	3.28±0.007	0.005
8	2.82±0.009	0.0003
9	2.44±0.004	0.002
10	2.3±0.015	0.002

Horse	Eigenvalue	Eigenvector
1	237.81±0.197	0.99
2	16.46±0.025	0.011
3	6.06±0.012	0.002
4	4.14±0.013	0.006
5	4.04±0.015	0.003
6	3.10±0.013	0.0003
7	2.70±0.015	0.016
8	2.56±0.009	0.002
9	2.41±0.010	0.013
10	2.12±0.014	0.001

Church	Eigenvalue	Eigenvector
1	186.59±0.212	0.99
2	8.81±0.038	0.005
3	7.33±0.015	0.004
4	4.24±0.011	0.012
5	3.92±0.015	0.015
6	3.21±0.009	0.014
7	2.53±0.012	0.023
8	2.28±0.011	0.005
9	1.92±0.005	0.026
10	1.88±0.007	0.009

Cat	Eigenvalue	Eigenvector
1	291.90±0.094	0.99
2	11.21±0.025	0.012
3	6.27±0.022	0.010
4	4.39±0.014	0.002
5	4.12±0.009	0.001
6	3.29±0.013	0.007*
7	2.79±0.017	0.0005*
8	2.63±0.016	0.015
9	2.44±0.005	0.012*
10	2.15±0.004	0.016

Figure 4: **The eigenvalue fluctuation and eigenvector similarity for 10 largest eigenvalues of** d_{Eig}. Here, all the experiments are obtained with Style2 under Clean configuration. The eigenvalue fluctuation (standard deviation) is obtained by repeating experiments with 4 random seeds. Also, we report the largest cosine similarity between the i-th largest eigenvector of GAN images and its counterpart of real images. The * indicates that the largest cosine similarity is not obtained between the the i-th largest eigenvectors of GAN and real images.

As displayed in Fig. 4, we report the eigenvalue and eigenvector behaviors for the 10 largest eigenvalues. The reported cutoffs were determined by the dominant percentage ($> 80\%$) taken by these eigenvalues compared to the complete spectrum. Notably, the 10 largest eigenvalues present small fluctuations (std) obtained with four random seeds, which serves as complementary evidence to support the theoretical rigidity of these eigenvalues. Except for the few cases marked with *, the largest cosine similarity is mostly obtained with the i-th largest eigenvector for both GAN and real images. If we decompose the distribution shift to scale shift (eigenvalue shift) and rotation shift (eigenvector shift), such results suggest that the dominant eigenvector shift is only determined by the cosine of the angle between them, and is not influenced by eigenvector permutation. By weighing the estimation challenges of eigenvectors, d_{Eig} makes a meaningful trade-off that only takes eigenvalue differences into account.

	Clean			PY_legacy			TF_legacy		
	Style3t	Style3r	Style2	Style3t	Style3r	Style2	Style3t	Style3r	Style2
d_{FID}	3.150±0.027	3.43±0.042	3.1±0.01	2.77±0.027	3.033±0.041	2.856±0.016	2.77±0.022	3.036±0.037	2.857±0.013
d_{Eig}	**1.9±0.06**	**2.22±0.061**	**2.657±0.073**	**1.627±0.060**	**1.883±0.053**	**2.513±0.056**	**1.625±0.005**	**1.883±0.043**	**2.507±0.049**

Figure 5: **The nuance between** d_{FID} **and** d_{Eig}. Here, all the experiments are conducted with the FFHQ dataset.

Lastly, we report a nuanced case when comparing d_{FID} and d_{Eig}. Fig. 5 shows that the face generalization performance *w.r.t.* d_{Eig} tends to be improved from Style2 to Style3r and Style3t, which is not compatible with d_{FID}. By imposing the translation and rotation equivariance in StyleGAN3, [18] reported anti-aliasing improvements over StyleGAN2 by resolving the 'texture sticking' issue. Such clear visual improvements are supported by the decreasing d_{Eig} scores. However, due to the lack of ground-truth, whether such a correlation between visual improvements and d_{Eig} supports the effectiveness of d_{Eig} remains inconclusive.