# A APPENDIX

This supplementary material provides in-depth information on the following topics:

- Additional Experiments.
- Experiment Details.
- Related Works.
- Training-Based Multi-modal Out-of-Distribution (OOD) Detection Methods
- Principal Component Analysis (PCA).
- Mean Variance Calculation.

Each section offers detailed insights into the respective topic for a comprehensive understanding.

# B ADDITIONAL EXPERIMENTS

## B.1 GRIC LEVERAGING RESNET-BASED CLIP MODELS

Our primary findings are based on the CLIP model featuring a Vision Transformer (ViT) image encoder. Additionally, we explore the efficacy of GRIC for models based on ResNet architecture in the context of CLIP. Specifically, we employ the ResNet model with a depth of 50 and a width multiplier of 4 (RN50x4) with 178.3 million parameters, a parameter count comparable to CLIP-B/16 (149.6 million). The results are presented in Table 4.

The outcomes demonstrate that GRIC continues to yield promising results when applied to ResNet-based CLIP models. The performance remains competitive between RN50x4 and CLIP-B/16, with AUROC values of 90.68 and 92.89, respectively.

| Method | OOD Dataset | | | | | | | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | iNaturalist (Van Horn et al., 2018) | | SUN (Xiao et al., 2010) | | Places (Zhou et al., 2017) | | Texture (Cimpoi et al., 2014) | | | |
| | **FPR95**↓ | **AUROC**↑ | **FPR95**↓ | **AUROC**↑ | **FPR95**↓ | **AUROC**↑ | **FPR95**↓ | **AUROC**↑ | **FPR95**↓ | **AUROC**↑ |
| GRIC (ours) (RN50x4) | 38.80 | 92.03 | 32.93 | 93.23 | 38.11 | 90.00 | 49.96 | 87.49 | 39.95 | 90.68 |
| GRIC (ours) (CLIP-B/16) | $10.32^{\pm}0.23$ | $98.81^{\pm}0.10$ | $20.11^{\pm}0.28$ | $97.59^{\pm}0.14$ | $24.37^{\pm}0.31$ | $96.82^{\pm}0.29$ | $26.51^{\pm}0.11$ | $93.97^{\pm}0.25$ | $20.32^{\pm}0.23$ | $96.80^{\pm}0.20$ |
| MCM (RN50x4) | 44.51 | 91.51 | 35.11 | 92.84 | 43.74 | 89.60 | 57.73 | 85.93 | 45.27 | 89.97 |
| MCM (CLIP-B/16) | 30.91 | 94.61 | 37.59 | 92.57 | 44.69 | 89.77 | 57.77 | 86.11 | 42.74 | 90.77 |

Table 4: GRIC presents outstanding performance leveraging ResNet-based CLIP model on ImageNet-1k (ID).

## B.2 EVALUATING THE SIGNIFICANCE OF $k$ IN GRIC PERFORMANCE

As elucidated in Section 3, we assign a value of zero to the $k$ most important features to derive a general representation of in-distribution (ID) data. The determination of the $k$ value is accomplished through diminishing mean-variance, with a set threshold of $1e^{-4}$ in our ImageNet-1k experiment, resulting in an optimal $k$ value of 34 to meet this threshold.

In this investigation, we systematically assess the impact of the $k$ value on the performance of GRIC by employing different $k$ values, specifically 30, 40, and 45. The experimental outcomes are detailed in Table 5. Notably, deviations from $k = 34$ demonstrate discernible effects on the method's performance. Generally, values close to $k = 34$ demonstrate discernible effects on the method's performance. Generally, values close to $k = 34$ exhibit comparable performance. However, as the deviation from $k = 34$ increases, there is a noticeable degradation in performance.

In conclusion, our analysis underscores the substantial influence of the $k$ parameter on the performance of the GRIC method. The selection of an appropriate $k$ value emerges as a critical factor in achieving optimal results across diverse out-of-distribution datasets.

## B.3 ID CLASSIFICATION ACCURACY

To augment the precision of ID classification, we integrate the comprehensive feature representation, acknowledging its indispensable role in the identification process. Simultaneously, we incorporate informative prompts that leverage hierarchy information, aligning with the methodology employed in our out-of-distribution (OOD) detection experiments, denoting it as GRIC-IP. The outcomes of our experiments are detailed in Table 3, underscoring the exceptional performance achieved with

| Method | OOD Dataset | | | | | | | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| | iNaturalist (Van Horn et al., 2018) | | SUN (Xiao et al., 2010) | | Places (Zhou et al., 2017) | | Texture (Cimpoi et al., 2014) | | | |
| | FPR95↓ | AUROC↑ | FPR95↓ | AUROC↑ | FPR95↓ | AUROC↑ | FPR95↓ | AUROC↑ | FPR95↓ | AUROC↑ |
| GRIC, $k = 34$ | 10.32 | 98.81 | 20.11 | 97.59 | 24.37 | 96.82 | 26.51 | 93.97 | 20.32 | 96.80 |
| GRIC, $k = 30$ | 10.67 | 98.66 | 20.78 | 97.09 | 24.70 | 96.56 | 26.84 | 93.80 | 20.75 | 96.53 |
| GRIC, $k = 40$ | 13.19 | 97.01 | 23.09 | 95.98 | 28.37 | 92.47 | 29.18 | 92.11 | 23.46 | 94.39 |
| GRIC, $k = 45$ | 13.68 | 96.81 | 23.43 | 95.82 | 28.19 | 92.35 | 29.53 | 91.93 | 23.71 | 94.23 |

Table 5: Impact of $k$ in performance of GRIC.

GRIC-IP. The results indicate that the incorporation of informative prompts contributes to an enhancement in the ID classification performance. Notably, it is crucial to emphasize that MCM represents the base of our methodology, omitting both the consideration of general ID representation and the utilization of informative prompts.

| Method | ID ACC |
|---|---|
| **Training free** | |
| MCM (CLIP-B/16) | 67.01 |
| MCM (CLIP-L/14) | 73.28 |
| GRIC-IP (CLIP-B/16) | 80.29 |
| GRIC-IP (CLIP-L/14) | 85.64 |
| **w. fine-tuning** | |
| MSP (CLIP-B/16) | 79.39 |
| MSP (CLIP-L/14) | 84.12 |
| Energy (Liu et al., 2020) (CLIP-B/16) | 79.39 |
| Energy (Liu et al., 2020) (CLIP-L/14) | 84.12 |
| Fort et al. (Fort et al., 2021) (ViT-B/16) | 81.25 |
| Fort et al. (Fort et al., 2021) (ViT-L/14) | 84.05 |
| MOS (Huang & Li, 2021) (BiT) | 75.16 |

Table 6: The accuracy of ID classification on ImageNet-1k (%) demonstrates promising performance with our method, GRIC-IP, which utilizes informative prompts.

Furthermore, Table 6 presents the multi-class classification accuracy on ImageNet-1k for the methods listed in Table 2.

### B.4 GRIC Masking (GM) leads to a notable enhancement in the performance of single-modal methods:

We conducted supplementary experiments to assess the influence of incorporating the general representation of in-distribution (ID) data on single-modal out-of-distribution (OOD) detection methodologies such as Mahalanobis (Lee et al., 2018), Energy score (Liu et al., 2020), React (Sun et al., 2021a), and GradNorm (Huang et al., 2021).

We utilize the ImageNet-1k dataset as the ID dataset in our experimental setup. Firstly, we compute mask indices and general feature representations of ID data from ImageNet-1k. Subsequently, we apply these mask indices to each test sample before subjecting them to traditional single-modal OOD detection methods. This methodology enables us to assess how leveraging general ID data representations influences the performance of OOD detection algorithms.

Results and Discussion: Our experimental results, as presented in Table 7 demonstrate that leveraging general feature representations from the ImageNet-1k dataset leads to improvements in the average AUROC performance of Mahalanobis, GradNorm, Energy score, and React OOD detection methods by 3.91, 2.74, 3.76, and 0.31, respectively. These findings highlight the significance of incorporating general ID data representations in enhancing the effectiveness of traditional single-modal OOD detection algorithms.

### B.5 Masking One Class at a time

In section 4.2, we present the initial findings from our experiments, focusing on the performance evaluation of our method across various ID datasets. The summarized outcomes are presented in Table 1. For this experiment, we derived masking indices and a general representation using all classes. An intriguing aspect of our approach involves masking class-specific features for individual classes. To delve deeper into this aspect, we conducted supplementary experiments, masking one class at a time while leveraging the ImageNet10 ID dataset.

| Single-modal method | FPR95↓ | AUROC↑ | Single-modal method+GM | FPR95↓ | AUROC↑ |
|---|---|---|---|---|---|
| Mahalanobis (Lee et al., 2018) | 87.43 | 55.47 | Mahalanobis + GM | **75.26** | **61.92** |
| GradNorm (Huang et al., 2021) | 40.29 | 87.34 | GradNorm + GM | **31.16** | **91.62** |
| Energy (Liu et al., 2020) | 58.41 | 86.17 | Energy + GM | **47.08** | **89.37** |
| React (Sun et al., 2021a) | 31.43 | 92.95 | React + GM | **25.31** | **95.73** |

Table 7: **GRIC Masking (GM) improves** most Single-modal methods significantly.

| Single-modal method | FPR95↓ | AUROC↑ | Single-modal method+GM | FPR95↓ | AUROC↑ |
|---|---|---|---|---|---|
| GRIC [All] | 0.20 | 99.88 | MCM | 0.33 | 99.78 |
| GRIC [car] | 0.31 | 99.73 | GRIC [bird] | 0.34 | 99.75 |
| GRIC [cat] | 0.29 | 99.80 | GRIC [antelope] | 0.32 | 99.76 |
| GRIC [dog] | 0.30 | 99.82 | GRIC [frog] | 0.31 | 99.77 |
| GRIC [truck] | 0.43 | 99.73 | GRIC [horse] | 0.32 | 99.75 |
| GRIC [warplane] | 0.38 | 99.69 | GRIC [Ship] | 0.40 | 99.61 |

Table 8: Masking one class at a time,ImageNet10 as ID. $x$ refers to the masked class in GRIC [x].

Following the experiment reported in Table 1, we evaluated our method using four OOD datasets: iNaturalist, SUN, Places, and Textures. We report the average performance metrics over these datasets, considering FPR95 and AUROC.

The detailed experimental outcomes are presented in Table 8. As shown in Table 8, masking different classes affects the performance variably. We observed that the best performance was achieved when leveraging masking generated by considering all classes collectively. Furthermore, our results highlight the importance of specific classes, prompting further investigation into the optimal selection of classes for masking. However, this aspect falls beyond the scope of the current paper and warrants future research investigations.

# C Experiment Details

## C.1 Software and Hardware

**Software** We run all experiments with Python 3.8.0 and PyTorch 1.12.1.

**Hardware** All experiments are run on NVIDIA RTX 3090.

## C.2 Hyperparameters

As we explained in section 3.3, the formal definition of the matching score $S(x; \mathcal{Y}\text{in}, \mathcal{T}, \mathcal{I})$ is given by:

$$S(x) = \max_i \frac{e^{s_i(x)}/\tau}{\sum_{j=1}^{N} e^{s_j(x)}/\tau}, \tag{8}$$

where we set $\mathcal{T}$ to 1 in our formulation. The sole hyper parameter governing our model is the temperature scaling factor denoted as $\tau$. Our empirical investigations indicate that, our scoring function exhibits robustness to variations in the scaling factor. Specifically, across a broad range of values spanning from 0.5 to 100, the performance remains consistent.

## C.3 Datasets

**ImageNet-10** We establish ImageNet-10, designed to emulate the class distribution of CIFAR-10, while utilizing high-resolution images. This dataset encompasses the following categories, each accompanied by its respective class ID: warplane (n04552348), sports car (n04285008), brambling bird (n01530575), Siamese cat (n02123597), antelope (n02422699), Swiss mountain dog (n02107574), bullfrog (n01641577), garbage truck (n03417042), horse (n02389026), and container ship (n03095699).

**ImageNet-20** For rigorous out-of-distribution (OOD) evaluation using realistic datasets, we adopt ImageNet-20, a dataset introduced by MCM. ImageNet-20 is meticulously curated, comprising 20 classes that are semantically akin to those in ImageNet-10, such as dog (in-distribution) versus wolf (OOD). The selection of categories is based on the semantic distance in the WordNet synsets (Fellbaum, 2010). The dataset encompasses the following categories: sailboat (n04147183), canoe (n02951358), balloon (n02782093), tank (n04389033), missile (n03773504), bullet train (n02917067), starfish (n02317335), spotted salamander (n01632458), common newt (n01630670), zebra (n01631663), frilled lizard (n02391049), green lizard (n01693334), African crocodile (n01697457), Arctic fox (n02120079), timber wolf (n02114367), brown bear (n02132136), moped (n03785016), steam locomotive (n04310018), space shuttle (n04266014), and snowmobile (n04252077). The generation of this dataset is facilitated using the script provided by the authors of MCM.

**ImageNet-100** We compile a dataset named ImageNet-100 by selecting 100 classes from ImageNet-1k. The MCM authors randomly chose these 100 classes without adhering to specific criteria. The

dataset creation process is executed using the script provided by the MCM authors. The list of classes utilized in this dataset is accessible at https://github.com/deeplearning-wisc/MCM.

**Conventional Out-of-Distribution (OOD) Datasets** Huang et al.(Huang & Li, 2021) meticulously compile a diverse set of subsets from prominent datasets such as iNaturalist(Van Horn et al., 2018), SUN (Xiao et al., 2010), Places (Zhou et al., 2017), and Texture (Cimpoi et al., 2014), establishing expansive OOD datasets for ImageNet-1k. Importantly, the test sets for these datasets are designed such that their classes do not overlap with those in ImageNet-1k. A brief overview of each dataset is provided below.

**iNaturalist**: Comprising images captured in the natural world (Van Horn et al., 2018), iNaturalist boasts 13 super-categories and 5,089 sub-categories, spanning various domains such as plants, insects, birds, mammals, and more. For our purposes, we utilize a subset encompassing 110 plant classes that do not overlap with those present in ImageNet-1k.

**SUN**: An acronym for the Scene Understanding Dataset (Xiao et al., 2010), SUN encompasses 899 categories, encapsulating diverse indoor, urban, and natural environments, both with and without human presence. We selectively use a subset of 50 categories representing natural objects absent in ImageNet-1k.

**Places**: As a repository of large-scale scene photographs (Zhou et al., 2017), Places categorizes images into Indoor, Nature, and Urban scenes. From the larger collection, we extract a subset comprising 50 categories that are distinct from those found in ImageNet-1k.

**Texture**: Denoting the Describable Textures Dataset (Cimpoi et al., 2014), Texture consists of images featuring textures and abstracted patterns. Given the absence of category overlaps with ImageNet-1k, we utilize the entire dataset, aligning with the approach taken by Huang et al. (Huang & Li, 2021).

### C.4 Baseline Models and Model Checkpoint Sources

In our evaluation of baseline models, we rely on reported experimental results from MCM (Ming et al., 2022) and CLIPN (Wang et al., 2023). For the Mahalanobis score (Lee et al., 2018), we utilize feature embeddings without $l_2$ normalization, considering that Gaussian distributions are inherently incompatible with hyperspherical features. Alternatively, one can opt to normalize the embeddings before applying the Mahalanobis score.

In the case of Fort et al.(Fort et al., 2021), detailed in Table2, the entire Vision Transformer (ViT) model undergoes fine-tuning on the in-distribution (ID) dataset. We leverage publicly available checkpoints from Hugging Face, where the ViT model is pre-trained on ImageNet-21k and subsequently fine-tuned on ImageNet-1k. For instance, the checkpoint for ViT-B can be accessed at https://huggingface.co/google/vit-base-patch16-224.

Regarding CLIP models, our reported results are based on checkpoints provided by Hugging Face for CLIP-B (https://huggingface.co/openai/clip-vit-base-patch16) and CLIP-L (https://huggingface.co/openai/clip-vit-large-patch14). Similar outcomes can be achieved using checkpoints available in the OpenAI codebase (https://github.com/openai/CLIP). Notably, for CLIP (RN50x4), which is not accessible via Hugging Face, we employ the checkpoint provided directly by OpenAI.

## D Related Works

**Vision-Language Models.** The usage of large-scale pre-trained vision-language models for multimodal tasks has emerged as a promising paradigm, exhibiting impressive performance (Gu et al., 2020). Typically, two architectural paradigms are prevalent: single-stream models, exemplified by VisualBERT (Li et al., 2019a) and ViLT (Kim et al., 2021), which integrate text and visual features into a single transformer-based encoder; and dual-stream models like CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021), and FILIP (Yao et al., 2021), employing separate encoders for text and image. These models optimize with contrastive objectives to align semantically similar features across different modalities. Among these, CLIP has gained widespread popularity due to its simplicity and robust performance. The success of CLIP-like models has prompted subsequent works (Li et al., 2022; Zhang et al., 2021), focusing on enhancing data efficiency and task adaptation. While our

paper centers around CLIP as the primary pre-trained model, the proposed approach can generally apply to contrastive models aiming to align vision and language features.

**OOD Detection in Computer Vision.** For open-world multi-class classification, the objective of OOD detection is to establish a binary ID-OOD classifier alongside a multi-class model tailored for visual inputs. Various methodologies have emerged for deep neural networks (Yang et al., 2021b). These approaches include generative model-based techniques (Cai & Li, 2023; Ge et al., 2017; Kirichenko et al., 2020), as well as discriminative-model based methods. Within the latter category, OOD scores are derived from the model's softmax output (DeVries & Taylor, 2018; Hein et al., 2019; Yang et al., 2021a), energy-based scores (Liu et al., 2020; Sun et al., 2021b; Sun & Li, 2022), or gradient information (Behpour et al., 2023; Huang et al., 2021). Theoretical analyses have been presented by (Morteza & Li, 2022; Fang et al., 2022; Bitterwolf et al., 2022) in the domain of OOD detection.

Recent works (Roy et al., 2022; Wang et al., 2022b) have explored OOD detection specifically in long-tailed distributions. So far, these works have primarily concentrated on task-specific models using only visual information. Our method marks a pioneering leap in training-free multi-modal OOD detection, incorporating informative textual information alongside the shared general visual representation within in-distribution data across a spectrum of diverse tasks.

# E Training-Based Multi-modal Out-of-Distribution (OOD) Detection Methods

In accordance with the discussions presented in Section D, numerous studies have explored the realm of multi-modal OOD detection, employing various training strategies.

**CLIPN: Saying "No" with CLIP**

Wang et al. (Wang et al., 2023) introduce CLIPN, an extension of CLIP (Contrastive Language-Image Pre-training) specifically designed for discerning between ID and OOD samples. CLIPN achieves this by incorporating positive semantic prompts and introducing negation-semantic prompts. The method employs a learnable "no" prompt and a dedicated "no" text encoder to capture negation semantics within images. Dual loss functions, the image-text binary-opposite loss, and the text semantic-opposite loss, are utilized to instruct CLIPN in associating images with "no" prompts, enabling it to identify unknown samples effectively.

**ZOC: Zero-Shot OOD Detection based on CLIP** Esmaeilpour et al. (Esmaeilpour et al., 2022) present the Zero-Shot OOD Detection (ZOC) method, extending the pre-trained language-vision model CLIP. ZOC incorporates a text-based image description generator trained on top of CLIP. During testing, this extended model generates candidate unknown class names for each test sample. A confidence score is then computed based on both known class names and candidate unknown class names, facilitating zero-shot OOD detection.

**CLIPood: Generalizing CLIP to OOD Test Data**

Shu et al. (Shu et al., 2023) proposed CLIPood, a fine-tuning method aimed at adapting CLIP models to out-of-distribution scenarios in downstream tasks. CLIPood addresses situations involving domain shifts and open classes in unseen test data. Introducing the margin metric softmax (MMS) as a novel training objective, CLIPood exploits semantic relations between classes from the text modality. Additionally, it incorporates a new optimization strategy, Beta moving average (BMA), for maintaining a temporal ensemble weighted by a Beta distribution. The focus of our paper centers on zero-shot multi-modal OOD detection, and thus, studies involving training a text encoder, such as those discussed above, fall outside the scope of our investigation.

In summary, these training-based methods showcase diverse approaches to multi-modal OOD detection, each contributing unique insights and methodologies. However, our emphasis in this paper is specifically on zero-shot multi-modal OOD detection, excluding investigations that involve training a text encoder.

# F PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA aims to transform high-dimensional data into a lower-dimensional representation while retaining the maximum variance in the data. It achieves this by identifying the principal components, which are orthogonal vectors that capture the directions of maximum variance (Shlens, 2014; Jolliffe, 2002).

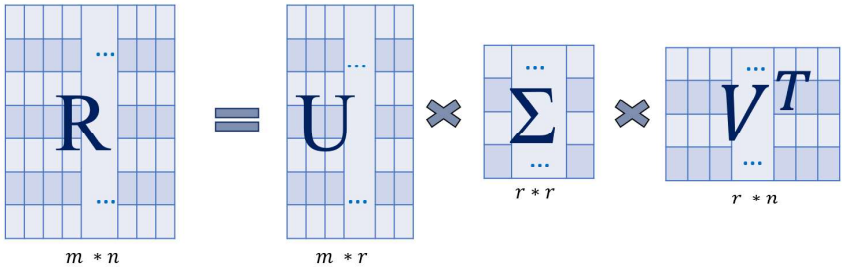


Figure 4: Singular Value Decomposition

## F.1 PROCEDURE:

In this section, we explain the procedure that is followed in PCA analysis to extract the most informative features.

**Data Standardization:**

Standardize the features of the dataset to have zero mean and unit variance.

**Covariance Matrix:** Compute the covariance matrix of the standardized data. The covariance matrix represents the relationships between different features.

**SVD of Covariance Matrix:**

Perform SVD on the covariance matrix. The singular value decomposition of the covariance matrix results in the principal components. Selecting Principal Components:

Sort the singular values in descending order. The corresponding singular vectors are the principal components. Choose the top k principal components to form a reduced-dimensional space.

**Projection:**

Project the original data onto the selected principal components to obtain the lower-dimensional representation.

**Benefits:**

Dimensionality reduction facilitates easier visualization and interpretation of data. Reduced dimensions often lead to computational efficiency. Principal components capture the most significant patterns in the data.

**More Explanation Regarding SVD Computation:** Consider an $m \times n$ matrix $\boldsymbol{R}$, where $m$ denotes the number of rows, and $n$ represents the number of columns. The primary objective of Singular Value Decomposition (SVD) is to decompose matrix $\boldsymbol{R}$ into three distinct matrices: $\boldsymbol{U}, \boldsymbol{\Sigma}$, and $\boldsymbol{V}^T$ (transpose of matrix $\boldsymbol{V}$). This decomposition is expressed as $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \in \mathbb{R}^{m \times n}$, as illustrated in Fig. 4.

$\boldsymbol{U}$: An $m \times m$ orthogonal matrix, where its columns signify the left singular vectors of $\boldsymbol{R}$.

$\boldsymbol{\Sigma}$: An $m \times n$ diagonal matrix, featuring singular values of $\boldsymbol{R}$ (non-negative and arranged in descending order).

$\boldsymbol{V}^T$: An $n \times n$ orthogonal matrix, with its columns representing the right singular vectors of $\boldsymbol{R}$.

In addition to singular values and singular vectors, eigenvalues and eigenvectors are also integral to understanding matrix properties. An eigenvalue $\lambda$ and its corresponding eigenvector $\mathbf{v}$ of a square matrix $\boldsymbol{R}$ satisfy the equation $\boldsymbol{R}\mathbf{v} = \lambda\mathbf{v}$. Eigenvectors denote directions in the vector space that are solely scaled by the matrix $\boldsymbol{R}$, while eigenvalues represent the scaling factors for these eigenvectors.

SVD and its Relationship to Eigenvalues and Eigenvectors: SVD establishes a crucial connection between eigenvalues and eigenvectors with the singular values and singular vectors of a matrix. The singular values of $\boldsymbol{R}$ are the square roots of the eigenvalues of either $\boldsymbol{R}\boldsymbol{R}^T$ or $\boldsymbol{R}^T\boldsymbol{R}$, and the left and right singular vectors are the eigenvectors of $\boldsymbol{R}\boldsymbol{R}^T$ and $\boldsymbol{R}^T\boldsymbol{R}$, respectively.

Rank and Matrix Approximation: The rank of a matrix $\boldsymbol{R}$ is determined by the count of non-zero singular values in $\boldsymbol{\Sigma}$. By retaining only the largest singular values and their corresponding singular vectors, it becomes feasible to approximate the original matrix $\boldsymbol{R}$ with a lower-rank approximation. This technique is valuable for tasks such as dimensionality reduction and noise reduction, and we leverage this feature in our approach.

**Properties of SVD:**
The singular values in $\boldsymbol{\Sigma}$ are non-negative and arranged in descending order. The columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal, forming an orthogonal basis for their respective vector spaces. The SVD decomposition is unique, except for the sign of the singular values and the order of the singular vectors. SVD is a potent matrix factorization technique, offering a concise representation of a matrix while preserving essential structural properties. Its applications span diverse fields, including data analysis, image processing, recommendation systems, and more (Deisenroth et al., 2020).

# G Principal Component Analysis for Computing Mean Variance

In this section, we describe the process of calculating the mean variance of high-dimensional features using Principal Component Analysis (PCA). Let $X \in \mathbb{R}^{n \times d}$ be the dataset, where $n$ represents the number of samples, and $d$ is the number of features.

## G.1 Feature Standardization

PCA is sensitive to the scale of the input data, so the first step is to standardize the features, ensuring each has a mean of zero and a variance of one.

Given the dataset $X = \{X_1, X_2, \ldots, X_n\}$, where each $X_i \in \mathbb{R}^d$ represents a sample with $d$ features, we standardize the data as follows:

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} X_{ij}, \quad \forall j = 1, 2, \ldots, d, \tag{9}$$

$$X_{\text{centered}} = X - \mu, \tag{10}$$

$$X_{\text{standardized}} = \frac{X_{\text{centered}}}{\sigma}, \quad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{ij} - \mu_j)^2}. \tag{11}$$

Here, $\mu_j$ is the mean of the $j$-th feature, and $\sigma_j$ is its standard deviation.

## G.2 Covariance Matrix Calculation

After standardizing the data, we compute the covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, which captures the relationships between features. The covariance matrix is defined as:

$$\Sigma = \frac{1}{n-1} X_{\text{standardized}}^{\top} X_{\text{standardized}}, \tag{12}$$

where $\Sigma_{jk}$ represents the covariance between features $j$ and $k$.

## G.3 Eigenvalue Decomposition

We perform an eigenvalue decomposition on the covariance matrix $\Sigma$, which yields the principal components and the amount of variance explained by each. The decomposition is given by:

$$\Sigma = V \Lambda V^\top, \tag{13}$$

where $V \in \mathbb{R}^{d \times d}$ is the matrix of eigenvectors (principal components), and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_d) \in \mathbb{R}^{d \times d}$ is a diagonal matrix with the eigenvalues $\lambda_j$, representing the variance explained by the $j$-th principal component.

### G.4 Explained Variance

The eigenvalues $\lambda_j$ indicate the variance captured by each principal component. The proportion of variance explained by the $j$-th component is calculated as:

$$\text{Explained Variance Ratio} = \frac{\lambda_j}{\sum_{k=1}^{d} \lambda_k}. \tag{14}$$

### G.5 Mean Variance Calculation

The mean variance explained by all principal components is computed by averaging the explained variance ratios:

$$\text{Mean Variance} = \frac{1}{d} \sum_{j=1}^{d} \frac{\lambda_j}{\sum_{k=1}^{d} \lambda_k}. \tag{15}$$

This value represents the average variance explained by each principal component.